# Temporal Super-Resolution Traffic Flow Forecasting via Continuous-Time Network Dynamics

**Yi Xie**
Shanghai Key Laboratory of Data Science, Fudan University
**Yun Xiong** ( yunx@fudan.edu.cn )
Shanghai Key Laboratory of Data Science, Fudan University
**Jiawei Zhang**
IFM Lab, Department of Computer Science, University of California
**Chao Chen**
College of Computer Science, Chongqing University
**Yao Zhang**
Shanghai Key Laboratory of Data Science, Fudan University
**Jie Zhao**
College of Computer Science, Chongqing University
**Yizhu Jiao**
Shanghai Key Laboratory of Data Science, Fudan University
**Jinjing Zhao**
National Key Laboratory of Science and Technology on Information System Security
**Yangyong Zhu**
Shanghai Key Laboratory of Data Science, Fudan University

**Research Article**

**Keywords:** Temporal Super-Resolution Forecasting, Network Dynamics, Continuous-Time

**Posted Date:** September 23rd, 2022

**DOI:** https://doi.org/10.21203/rs.3.rs-2086004/v1

# Temporal Super-Resolution Traffic Flow Forecasting via Continuous-Time Network Dynamics

Yi Xie[1,2], Yun Xiong[1,2*], Jiawei Zhang[3], Chao Chen[4], Yao Zhang[1,2], Jie Zhao[4], Yizhu Jiao[1,2], Jinjing Zhao[5] and Yangyong Zhu[1,2]

[1*]Shanghai Key Laboratory of Data Science, Fudan University, Shanghai, 200433, China.
[2]School of Computer Science, Fudan University, Shanghai, China.
[3]IFM Lab, Department of Computer Science, University of California, Davis, USA.
[4]College of Computer Science, Chongqing University, Chongqing, 400044, China.
[5]National Key Laboratory of Science and Technology on Information System Security, Beijing, 100101, China.

*Corresponding author(s). E-mail(s): yunx@fudan.edu.cn;
Contributing authors: yixie18@fudan.edu.cn; jiawei@ifmlab.org; cschaochen@cqu.edu.cn; yaozhang18@fudan.edu.cn; yzjiao@fudan.edu.cn; zhjj0420@126.com; yyzhu@fudan.edu.cn;

**Abstract**

Traffic flow forecasting is a critical task for Intelligent Transportation Systems. However, the existed forecasting can only be conducted at certain timestamps, because the data, is discretely collected at these timestamps. In contrast, traffic flow evolves in real-time via a continuous manner in real world. Therefore, an ideal forecasting paradigm should be performed at arbitrary timestamps instead of only at these certain timestamps. Considering the forecasting timestamps will no longer be restricted by these timestamps, we call such paradigm as temporal super-resolution forecasting. In this paper, we incorporate the idea of neural ordinary differential equations (Neural ODEs) to handle the problem, modeling the change rate of traffic flow on the

urban road. Therefore, due to the continuous nature of ordinary differential equations, the traffic flow at arbitrary timestamps can be forecasted by performing definite integral for the change rate. The urban road is usually regarded as a network, and the change rate of which can be described by continuous-time network dynamics, we parameterize the network dynamics of the traffic flow to quantify the change rate. On these foundations, we propose **S**patial-**T**emporal **C**ontinuous **D**ynamics **N**etwork (STCDN) to complete the temporal super-resolution forecasting task. Extensive experiments on public traffic flow datasets illustrate that our model can achieve high accuracy on temporal super-resolution forecasting, while ensuring its performance on conventional experimental settings at these certain timestamps.

**Keywords:** Temporal Super-Resolution Forecasting, Network Dynamics, Continuous-Time

# 1 Introduction

With the urbanization and intelligence of human activities, traffic flow forecasting plays a fundamental role in urban governance. Such a task can significantly enhance various urban computing tasks, *e.g.*, traffic congestion control, vehicular trajectories analysis, estimating time of arrival and internet of vehicles [1–4]. In most cases, traffic flow data is usually recorded at certain timestamps, resulting in the recorded data has inherent discrete nature.

The discrete nature hinders us from handling the information within intervals among recording timestamps. Because there is no information to be contrasted within recording intervals, implying most of the information is unrecorded and intractable, resulting in the forecasting can only be conducted at these recording timestamps. For instance, when the recording interval is 5 minutes, one can't forecast the traffic flow after 30 seconds or 7 minutes, since there are no supervision signals to guide the forecasting at corresponding timestamps. Under such circumstances, the forecasting flexibility will be significantly restricted. Especially in some cases, people usually need more flexible or more frequent forecasting, *e.g.*, rescue activities, emergencies or rush hours, *etc.*, the equal-resolution forecasting under the existing paradigm, where the forecasting intervals equal to the original recording intervals, will powerless. Therefore, a new forecasting paradigm that beyonds the original recording timestamps, will be more valuable to those in need. The motivation can be illustrated in Fig. 1 (A).

This paper aims to make the traffic flow forecasting can be conducted at arbitrary timestamps, regardless of the recording intervals. In which, the forecasting intervals can be far smaller than recording intervals. We call such a forecasting paradigm as temporal super-resolution traffic flow forecasting (TSRF for short). Correspondingly, the conventional case that the

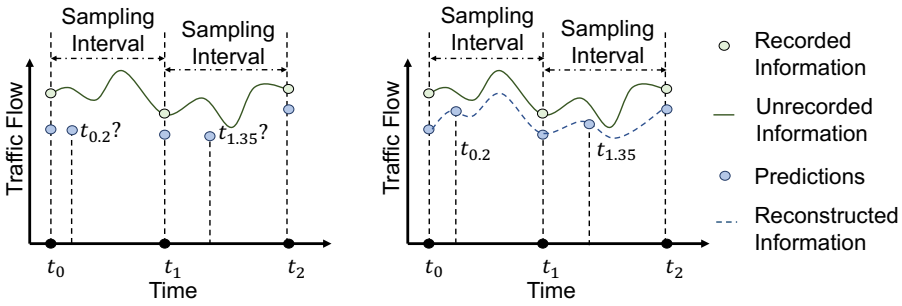


**Fig. 1** The illustration of our motivation and solution. For simplicity, we use integer timestamps to represent recording timestamps, and use float timestamps timestamps to represent any timestamps within recording intervals. (A) The motivation. Conventionally, the forecasting can only be conducted at the recording timestamps, and can't be conducted at arbitrary timestamps within recording intervals. (B) The solution. We model the change rate of traffic flow, and perform definite integral for the change rate, so that forecast the traffic flow at arbitrary timestamps regardless of the recording intervals.

forecasting are conducted at recording timestamps is referred to as temporal equal-resolution traffic flow forecasting (TERF for short) in this paper.

Despite its significance, TSRF is very challenging due to the following reasons:

- **The sparsity of supervision signals.** The coarse-grained nature results in the extreme sparsity of provided supervision signals, which means that the forecasting at most timestamps is unavailable to contrast with supervision signals. Therefore, we should make full use of the limited supervision signals to overall optimize the forecasting at all timestamps, which is not focused on in the previous studies.
- **The causality of temporal correlations.** Time series data always present strong temporal causality [5], which implies that the evolution within any tiny time intervals should meet such an abstract but actually existed temporal causality. Because the causality is hard to be quantified, it will inject extra complexity in TSRF.
- **The dependencies of spatial correlations.** Although the TSRF is conducted in the temporal dimension, the traffic flow evolution also has spatial correlations, which should be simultaneously modeled in the forecasting.

Fortunately, the idea of neural ordinary differential equations (Neural ODEs) [6] points out that the traffic flow evolution within tiny time intervals is not untraceable. We model the change rate for traffic flow, and perform the definite integral for the change rate thus forecasting the traffic flow at arbitrary timestamps due to the continuous nature of ordinary differential equations. The solution is shown as Fig. 1 (B).

The urban road is usually modeled as a network, we use continuous-time network dynamics to describe the change rate. Previous studies confirm that traffic flow of adjacent vertices in road networks are typically similar to each

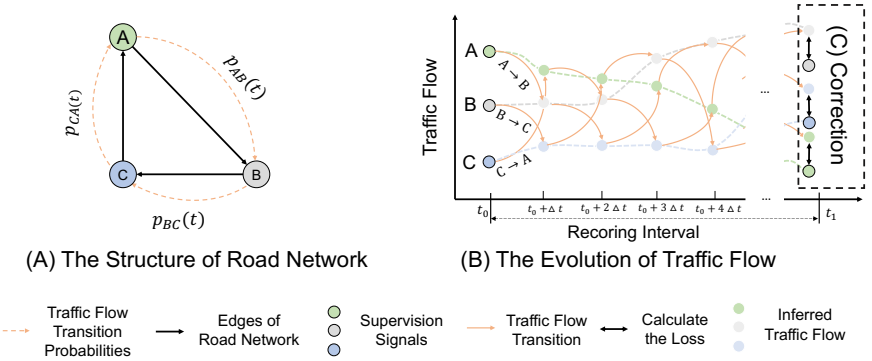


**Fig. 2** The network dynamics of traffic flow on the urban road network. (A) The topology of abstract road network, black lines represent the directed edges in the network, and the orange dotted lines are modeled transition probabilities at the moment. (B) The traffic flow evolution driven by flow transitions, where $\Delta t$ denotes a tiny time interval (mathematically infinitesimal). (C) At the corresponding timestamps, where supervision signals are available, we will use the supervision signals to correct the forecasting and optimize the network dynamics.

other because vehicles traverse between them frequently [7, 8], which can be recognized as the spatial local stationary property in statistics [9]. In light of the traffic flow will spontaneously transfer with corresponding transition probabilities to the adjacent vertices, the network dynamics can be quantified as a transition process on the road network, which will be divided into two parts: (1) real-time inference of transition probabilities for traffic flow; and (2) calculation of traffic flow transition volume based on transition probabilities. Intuitively, we take Fig. 1(B) as an example, the traffic flow of vertex $A$ continuously transfer to vertex $B$ with transition probability $p_{AB}(t)$ at timestamp $t$, causing the traffic flow to increase in $B$ but decrease in $A$. Therefore, we can model the traffic flow evolution with a tiny time interval in terms of traffic flow transitions, to better capture the spatial correlations.

By incorporating the concept of continuous network dynamics, we can well tackle the three challenges mentioned above. Firstly, the continuous nature of network dynamics can help us to infer the traffic flow at arbitrary desired timestamps. Secondly, because of the additivity of definite integrals, subsequent states are completely determined by previous states, so that the temporal causality can be well preserved. Lastly, we specify the network dynamics as the form of GNNs to capture the complex spatial correlations due to the message passing mechanism of GNNs [10]. Based on the above, we design a model, **S**patial-**T**emporal **C**ontinuous **D**ynamics **N**etwork (STCDN) [1]. In STCDN, we model the continuous-time network dynamics for traffic flow on the road network, and forecast future instantaneous traffic flow at arbitrary desired timestamps. Experiments on four public traffic flow datasets illustrate that our model can not only achieve high accuracy on temporal super-resolution traffic

[1]The source code is available at https://github.com/Xieyyyy/STCDN

flow forecasting, but also outperforms other baselines on conventional TERF tasks.

In general, the major contributions can be summarized as follows:

- We identify a common but under-explored issue, that existing traffic forecasting studies can only be conducted at certain recording timestamps, and fail to perform temporal super-resolution forecasting Therefore, we propose **S**patial-**T**emporal **C**ontinuous **D**ynamics **N**etwork (STCDN) to forecast the traffic flow at arbitrary timestamps, rather than these certain recording timestamps. We call such forecasting paradigm as temporal super-resolution forecasting (TSRF).
- To this end, we incorporate the idea of neural ordinary differential equations (Neural ODEs) to model change rate of traffic flow, and to forecast the traffic flow at arbitrary timestamp via definite integrals. Considering the urban road is usually recognized as a network, we specify the change rate of traffic flow as continuous-time network dynamics.
- Extensive experiments illustrate that in temporal super-resolution forecasting tasks, comparing with an intuitive solution, *i.e.*, incorporating interpolation algorithms, our model respectively achieves averaged 8.0% performance improvements. Meanwhile, in conventional experimental settings, our model also outperforms other baselines evaluated by 10 out of 12 metrics with averaged 2.60% improvements.

# 2 Related Work

## 2.1 Traffic Flow Forecasting

Traffic flow forecasting is a typical spatial-temporal modeling task that is important for urban computing and is of great significance for the construction of smart cities. Earlier studies focus on traditional statistical methods to analyze univariate time series, the representatives include Historical Average (HA), Vector Auto-Regression (VAR) [11, 12], Auto-Regressive Integrated Moving Average (ARIMA) [13, 14], and Support Vector Regression Machines (SVR) [15], *etc.* These shallow methods only capture the temporal dependencies and simplify the traffic flow modeling into individual time-series forecasting, the preconditions for these methods to capture complex spatial-temporal correlations are sophisticated and manually designed feature engineering.

With deep neural networks proving their superiority of powerful representation ability, subsequent studies leveraged neural networks to complete more accurate modeling [16–18], which completely ignores the spatial dependencies. Accordingly, Convolutional Neural Networks (CNN) are utilized to model spatial correlations of rasterized road networks [19–22]. Meanwhile, the temporal information can be handled by sequential models, like Recurrent Neural Networks (RNN) [20, 23] or Temporal Convolutional Networks (TCN) [24, 25].

However, CNN-based methods can only be applied to Euclidean data. Therefore, Graph Neural Networks (GNN) are incorporated to tackle massive

non-Euclidean spatial data, named spatial-temporal graph neural networks [26]. Some graph-based methods adopt prior knowledge to construct a graph structure via a pre-defined adjacency matrix [23, 24, 27, 28], in which the graph structure remains constant because its information has been determined by prior knowledge. In order to boost the representation ability of graph neural networks, various auxiliary adjacency matrices are introduced to describe spatial relationships from different aspects, such as DTW distances [29, 30] for measuring the feature relevance of vertices, or other specific functional relevance (*e.g.*, POI) [20, 31]. Recently, some studies have adopted entirely data-driven optimizable semantic adjacency matrices [25, 32, 33] to capture latent correlations among vertices. The spatial information and temporal information will be further integrated and fused in different ways, such as stacking [24, 27, 34], embedding [23, 25] or synchronization [28] *etc.*

Recently, some studies introduce Neural ODEs to to obtain spatial-temporal hidden states with continuous depth, thus greatly improving the representation ability of the model [30, 35–37]. Nonetheless, on the one hand, in technique, these studies do not introduce the view of system dynamics to associate the continuity with continuous physical time; on the other hand, in application, few studies pay attention to those actually happened but unrecorded information within recording intervals, and none of these studies focus on the significant but under-valued temporal super-resolution forecasting task. These methods above assume that the acquired temporal information is absolutely complete, resulting in inflexible forecasting. Therefore, these ODE based methods also cannot be directly utilized for temporal super-resolution forecasting.

## 2.2 Super-Resolution Reconstruction

Super-resolution reconstruction was first widely studied in computer vision. Researchers aim to reconstruct relatively higher resolution images based on lower resolution images, referred to as super-resolution reconstruction [38, 39]. Earlier studies adopt interpolation algorithms [40, 41] to reconstruct fine-grained information on images and obtain high-resolution images. However, Super-resolution reconstruction is an inherently ill-posed problem, there always exist multiple high-resolution images corresponding to one original low-resolution image. Therefore, learned-based methods are incorporated to reconstruct super-resolution images with richer semantic information [42, 43]. Subsequently, several studies referred to the idea of super-resolution reconstruction to reconstruct fine-grained information of MRI information [44], crowd flow information [45–47], and radio map information [48, 49], *etc.* Nonetheless, these studies all focus on fine-grained spatial information reconstriction. Very few studies focus on temporal information reconstriction and temporal super-resolution forecasting. A similar task is missing value imputation of time series [50–52]. Nonetheless, the missing value imputation task has fundamental differences with our task: (1) Task difference. The former is an interpolation task, which aims to impute the original incomplete time series

data. In our task, it is an extrapolation task, aiming to make the forecasting intervals independent of recording intervals; (2) Data difference. The former is usually adopted to tackle corrupted data, which might be caused by device failures and human errors, *etc.* In our task, the data could be corrupted or not, the coarse-grained and incomplete nature of which is mainly caused by the inherent recording limitations. (3) Purpose difference. The former is usually adopted to repair corrupted data, and our task is adopted to forecast future traffic flow at flexible timestamps.

# 3 Problem Satatement

## 3.1 Definition (Traffic Network)

Let $G = (V, E, \mathbf{A})$ denotes a traffic network, with the set of vertices $V$ (sensing devices), and the set of edges $E$ (geographical or semantic connections). $|V| = n$ represents the graph $G$ contains $n$ vertices. $\mathbf{A} \in \mathbb{R}^{n \times n}$ denotes the adjacency matrix of $G$. This paper will learn the adjacency matrix $\mathbf{A}$ with an end-to-end manner. Additionally, at the timestamp $t$, there are features of vertices $\mathbf{X}(t) \in \mathbb{R}^{n \times i}$, where $i$ is the original input feature dimension.

## 3.2 Temporal Super-Resolution Traffic Flow Forecasting (TSRF)

The temporal super-resolution traffic flow forecasting task (TSRF) is introduced to forecast traffic flow at arbitrary desired timestamps.

Formally, given the traffic network $G$ and $h$ historical observations from the initial timestamp $t_{-h}$ to the terminal timestamp $t_{-1}$: $\mathcal{X}(t_{-h} : t_{-1}) = [\mathbf{X}(t_{-h}), \mathbf{X}(t_{-h+1}), \cdots, \mathbf{X}(t_{-1})]$, our target is to find a mapping function $\mathcal{F}_\Theta(t, \cdot)$ to forecast the traffic flow at an arbitrary timestamp $t$. Here, $t$ can be any timestamp that satisfies $t \geq t_0$. We should minimize the error between the forecasted traffic flow and the supervision signals $\mathcal{X}(t_0 : t_{q-1}) = [\mathbf{X}(t_0), \mathbf{X}(t_1), \cdots, \mathbf{X}(t_{q-1})]$ at corresponding timestamps locate in $T = [t_0, t_1, \cdots, t_{q-1}]$:

$$\begin{aligned} \underset{\Theta}{\operatorname{argmin}} \quad & \sum_{t \in T} L(\mathbf{Y}(t), \mathbf{X}(t)) \\ \text{s.t.} \quad & \mathbf{Y}(t) = \mathcal{F}_\Theta(t, \mathcal{X}(t_{-h} : t_{-1})) \end{aligned}, \tag{1}$$

where $L$ is the objective function, $\Theta$ denotes all trainable parameters. $\mathcal{F}_\Theta(t, \mathcal{X}(t_{-h} : t_{-1})) \in \mathbb{R}^{n \times i}$ is a matrix to describe the state at timestamp $t$, $\mathcal{X}(t_{-h} : t_{-1}) \in \mathbb{R}^{h \times n \times i}$, $\mathbf{Y}(t), \mathbf{X}(t) \in \mathbb{R}^{n \times i}$ are lists of values, we stack it as matrices to facilitate parallelization. Note that we will use negative values to denote the timestamps with historical observations, while $t_0$ denotes the initial timestamp of forecasting.

### 3.3 Temporal Equal-Resolution Traffic Flow Forecasting (TERF)

In order to correspond to the aforementioned TSRF, we introduce the temporal equal-resolution traffic flow forecasting task (TERF) to represent the conventional traffic flow forecasting, where the forecasting intervals equal to the recording intervals.

Formally, given the traffic network $G$ and $h$ historical observations from the initial timestamp $t_{-h}$ to the terminal timestamp $t_{-1}$: $\mathcal{X}(t_{-h} : t_{-1}) = [\mathbf{X}(t_{-h}), \mathbf{X}(t_{-h+1}), \cdots, \mathbf{X}(t_{-1})]$, our target is to find a mapping function $\mathcal{F}_\Theta(\cdot)$ to forecast the next $q$-step traffic flow from the timestamp $t_0$ to the terminal $t_{q-1}$ as $\mathcal{Y}(t_0 : t_{q-1})$, and minimize the error between the forecasted traffic flow and the supervision signals $\mathcal{X}(t_0 : t_{q-1})$:

$$\begin{aligned} &\underset{\Theta}{\operatorname{argmin}} \quad L(\mathcal{Y}(t_0 : t_{q-1}), \mathcal{X}(t_0 : t_{q-1})) \\ &\text{s.t.} \quad \mathcal{Y}(t_0 : t_{q-1})) = \mathcal{F}_\Theta(\mathcal{X}(t_{-h} : t_{-1})) \end{aligned}, \tag{2}$$

where $\mathcal{Y}(t_0 : t_{q-1}), \mathcal{X}(t_0 : t_{q-1}) \in \mathbb{R}^{q \times n \times i}$.

The Problem 1 is what our model should tackle, and the Problem 2 is the conventional problem definition of previous studies. The difference between the both is if we can flexibly select the forecasted timestamp $t$. In the former, $t$ can be any timestamp that satisfies $t \geq t_0$, while the forecasted timestamp $t$ are fixed in the latter. Actually, the latter is a subproblem of the former, which implies that one can solve the former will certainly solve the latter, and not vice versa.

## 4 An Overall Solution

From the continuous-time dynamical system view, the continuous depth of neural networks is equivalent to continuous physical time [6, 53, 54]. Meanwhile, the discrete layer of neural networks can be converted as a continuous one in the neural ODEs, also can be recognized as the change rate of features at each moment [6]. Therefore, we impose the continuous depth on network dynamics via neural ODEs, enabling it to represent continuous physical time.

Firstly, for better representation ability, we will use a fully-connected layer to map the original input traffic flow feature $\mathbf{X}(t) \in \mathbb{R}^{n \times i}$ at timestamp $t$ into high-dimensional hidden space $\mathbf{H}(t) \in \mathbb{R}^{n \times d}$ with hidden dimension $d$. At timestamp $t$, the traffic flow hidden state is denoted as $\mathbf{H}(t) \in \mathbb{R}^{n \times d}$, we formulate the network dynamics of traffic flow on road network as a continuous-time function $f_\Theta(t, \mathbf{H}(t))$ over time $t$. The network dynamics should be the form of ordinary differential equation [6]:

$$f_\Theta(t, \mathbf{H}(t)) = \frac{d\mathbf{H}(t)}{dt}, \tag{3}$$

where $\Theta$ denotes all trainable parameters. Essentially, the network dynamics is the derivative function of hidden state $\mathbf{H}(t)$ over time $t$. Under the circumstances of our task, it can also be interpreted as the instantaneous rate of change of traffic flow.

By integrating the Eqn (3) over time $t$ from an initial hidden state $\mathbf{H}(t_0) \in \mathbb{R}^{n \times d}$ at timestamp $t_0$, Eqn (3) is actually equivalent to solving an initial value problem [55]. We can infer the continuous-time instantaneous hidden state $\mathbf{H}(t)$ at an arbitrary timestamp $t \geq t_0$ via the definite integral with variable upper bound:

$$\mathbf{H}(t) = \mathbf{H}(t_0) + \int_{t_0}^{t} f_\Theta(\tau, \mathbf{H}(\tau))d\tau. \tag{4}$$

Therefore, from the Eqn (4), we can model the traffic flow on road network as a constant coefficient dynamical system with parameter sharing over time. The larger upper bound $t$ is, the "deeper" the neural networks is, and in physical meaning, the longer the evolution time consuming of traffic flow on the road network is. Meanwhile, because the upper bound $t$ can be arbitrarily selected, as long as it satisfies $t \geq t_0$, we can infer the traffic flow at arbitrary timestamps regardless of the recording intervals, realizing the TSRF task. Also, the parameter sharing nature provides the opportunity to overall optimize the network dynamics by minimizing the loss on partial timestamps.

We will use a self-attention based Graph Neural Network to specify the network dynamics. Intuitively, in the context of traffic flow on road network, the above idea can be interpreted as a straightforward phenomenon that the ceaseless traffic flow transition in the road network will trigger the evolution of traffic flow over time. The intuition is exactly meets the conventional physical-guided traffic flow theories [7, 8, 56].

# 5 Methodology: Spatial-Temporal Continuous Dynamics Network

## 5.1 Overview

The overview of Spatial-Temporal Continuous Dynamics Network (STCDN) is shown as Fig. 3. From the illustration, we can see that STCDN is a typical encoder-decoder model [57]. The most intuitive characteristic is our model will generate a set of continuous dotted curves via the definite integral of the network dynamics, which exactly represent the continuous Subsequently, we will introduce the encoder and the decoder, respectively.

### 5.1.1 Encoder

Theoretically, if we have the initial hidden state $\mathbf{H}(t_0)$, we can inference the subsequent continuous-time hidden state via the Eqn (4). This implies that the initial hidden state should contain rich semantic information. Therefore, we design an encoder to incorporate the information from historical observations.

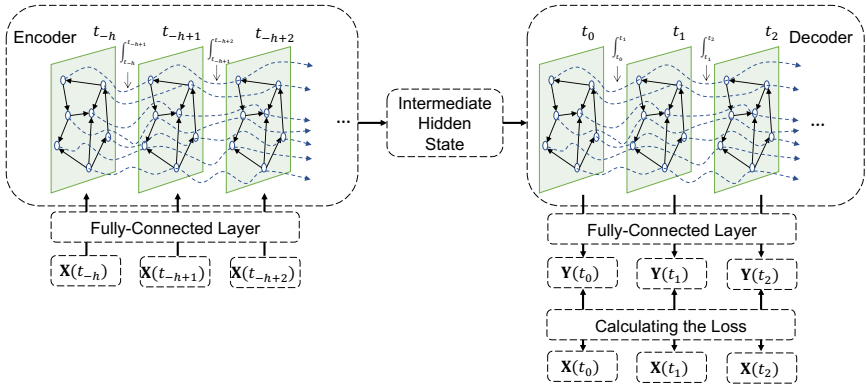


**Fig. 3** The overview of Spatial-Temporal Continuous Dynamics Network.

Formally, given the historical observations $\mathcal{X}(t_{-h} : t_{-1}) = [\mathbf{X}(t_{-h}), \mathbf{X}(t_{-h+1}), \cdots, \mathbf{X}(t_{-1})]$ from the from the initial timestamp $t_{-h}$ to the terminal timestamp $t_{-1}$, where $\mathcal{X}(t_{-h} : t_{-1}) \in \mathbb{R}^{h \times n \times i}$. We firstly use a fully-connected layer to map it into a hidden space: $\mathcal{H}(t_{-h} : t_{-1}) = [\mathbf{H}(t_{-h}), \mathbf{H}(t_{-h+1}), \cdots, \mathbf{H}(t_{-1})]$, where $\mathcal{H}(t_{-h} : t_{-1}) \in \mathbb{R}^{h \times n \times d}$. We take $\mathbf{H}(t_{-h})$ as the initial state to solve the initial value problem until the integral upper bound $t_{-h+1}$ via the network dynamics $f_{\Theta_E}(t, \mathbf{H}(t))$, and obtain the hidden state at $t_{-h+1}$:

$$\mathbf{H}(t_{-h+1}) = \mathbf{H}(t_{-h}) + \int_{t_{-h}}^{t_{-h+1}} f_{\Theta_E}(\tau, \mathbf{H}(\tau)) d\tau, \tag{5}$$

where $\Theta_E$ denotes all trainable parameters in the encoder, which implies the parameters in the encoder and the decoder are not shared. After obtaining the $\mathbf{H}(t_{-h+1}) \in \mathbb{R}^{n \times d}$, there are historical observations $\mathbf{X}(t_{-h+1}) \in \mathbb{R}^{n \times i}$ that can be used to complement the hidden state, reducing the errors accumulation. We will use a linear combination to fuse the hidden state $\mathbf{H}(t_{-h+1})$ and the historical observations $\mathbf{X}(t_{-h+1})$:

$$\mathbf{X}(t_{-h+1}) := FC(\mathbf{H}(t_{-h+1})) + FC(\mathbf{X}(t_{-h+1})), \tag{6}$$

where $FC(\cdot)$ denotes a fully-connected layer with activation function. The above process will be performed repeatly, until all historical observations are encoded into an intermediate hidden state $\mathbf{H} \in \mathbb{R}^{n \times d}$.

Although the TSRF will not be performed in the encoder, we still encode the network dynamics, since we observe the network dynamics can be interpreted as a feature augmentation, enabling the model to achieve better forecasting performance. The hypothesis will be confirmed in the ablation study part.

### 5.1.2 Decoder

The decoder takes the hidden state $\mathbf{H}$ as the initial information. Because no any complementation information should be incorporated, we directly integrate the network dynamics $f_{\Theta_D}(t, \mathbf{H}(t))$ to forecast the traffic flow $\mathbf{Y}(t) \in \mathbb{R}^{n \times i}$ at the timestamp $t$:

$$\mathbf{Y}(t) = FC(\mathbf{H} + \int_{t_0}^{t} f_{\Theta_D}(\tau, \mathbf{H}(\tau))d\tau). \tag{7}$$

Analogously, $\Theta_D$ denotes all trainable parameters in the decoder.

### 5.1.3 Optimization

STCDN provides an end-to-end manner to optimize. Formally, given the ground-truth from $t_0$ to $t_{q-1}$, $\mathcal{X}(t_0 : t_{q-1})$, as supervision signals, together with the set of timestamps where supervision signals locate in: $T = [t_0, t_1, \cdots, t_{q-1}]$. We use Mean Absolute Errors (MAE) as the loss function:

$$L = \sum_{t \in T} |\mathbf{Y}(t) - \mathbf{X}(t)| + \lambda ||\Theta||_2, \tag{8}$$

where the latter term $\lambda ||\Theta||_2$ denotes L2 regularization for avoiding overfitting.

We will only extract the forecasting results at timestamps that contained in $T$ to contrast the ground-truth and optimize. Because the network dynamics share the same parameters, we optimize the model according to the forecasting at these certain timestamps, forecasting at arbitrary timestamps will also be adjusted simultaneously, since these forecasting are all generated by the network dynamics. From this perspective, the TSRF can be regarded as a semi-supervised learning task.

## 5.2 Specifying the Form of Network Dynamics

Secondly, the form of network dynamics $f_\Theta(t, \mathbf{H}(t))$ should be specified. The network dynamics should contain two processes: (1) real-time inference of transition probabilities for traffic flow; and (2) calculation of traffic flow transition volume based on transition probabilities. We desing a self-attention mechanism to complete the both processes.

The first task is to infer the real-time transition probabilities for traffic flow. Formally, given the hidden state $\mathbf{H}(t)$ and the adjacency matrix $\mathbf{A}$ at timestamp $t$, the transition probabilities can be denoted as a transition matrix $\mathbf{M}(t) \in \mathbb{R}^{n \times n}$:

$$\mathbf{M}(t) = Softmax(Filter(\mathbf{A} \circ (\mathbf{Q}_1(t) + \mathbf{Q}_2(t))\mathbf{K}(t)^\top), \tag{9}$$

where

$$\begin{aligned}\mathbf{Q}_1(t) &= \mathbf{H}(t)\Theta_{Q1},\\ \mathbf{Q}_2(t) &= \mathbf{H}(t)\Theta_{Q2},.\\ \mathbf{K}(t) &= \mathbf{H}(t)\Theta_K\end{aligned} \tag{10}$$

$$Filter(\mathbf{A}_{ij}) = \begin{cases}\mathbf{A}_{ij} & \mathbf{A}_{ij} \neq 0\\ -\infty & \text{otherwise}\end{cases}. \tag{11}$$

$\Theta_{Q1}, \Theta_{Q2}, \Theta_K \in \mathbb{R}^{d\times d}$ are trainable parameters, $\circ$ denotes Hadamard product. The purpose of the simultaneous existence of $\mathbf{Q}_1(t)$ and $\mathbf{Q}_2(t)$ is to obtain the representations as the roles of target and source in a directed graph, respectively. $Filter(\cdot)$ is a newly defined function to set 0-value entries as the negative infinity, ensuring the transition probabilities of non-neighbor are always 0 after the $Softmax(\cdot)$.

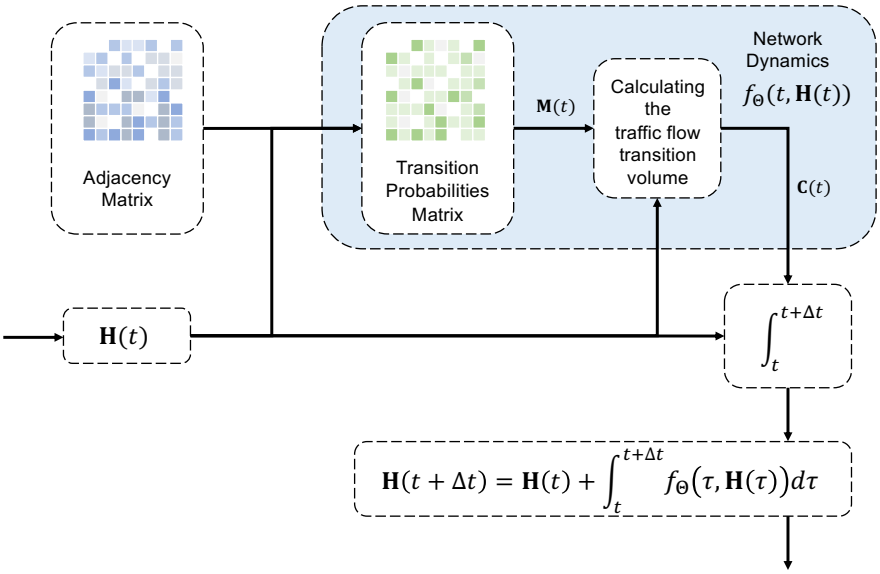


**Fig. 4** The illustration of the self-attention based network dynamics.

After obtaining the transition probabilities matrix, the next task is to calculate the traffic flow transition volume $\mathbf{C}(t) \in \mathbb{R}^{n\times d}$ based on transition probabilities:

$$\mathbf{C}(t) = \mathbf{M}(t)\mathbf{V}(t), \tag{12}$$

where

$$\mathbf{V}(t) = \mathbf{H}(t)\Theta_V, \tag{13}$$

$\Theta_V \in \mathbb{R}^{d\times d}$ is a matrix of trainable paramters. The purpose of Eqn (12) is to complete traffc flow transition and perform feature transformation. The traffic flow transition volume $\mathbf{C}(t)$ numerically equals to the network dynamics

$f_\Theta(t, \mathbf{H}(t))$. Intuitively, the traffic flow transition volume is exactly the volume of cahnge of the traffic flow at each moment, which can represent the derivative function of traffic flow on road network.

Combining Eqn (9), (10), (11), and (12), we can obtain a self-attention based Graph Neural Network to represent the network dynamics, which can be written as the following fully expanded form:

$$f_\Theta(t, \mathbf{H}(t)) = Softmax(Filter(\mathbf{A} \circ \mathbf{Q}_1(t) + \mathbf{Q}_2(t))(\mathbf{K}(t))^\top \mathbf{V}(t)) \quad (14)$$

Similar to other self-attention based algorithms [58, 59], we impose multi-head operation on the network dynamics to reduce parameters and enhance the representation ability.

## 5.3 Implementation Details

### 5.3.1 Numerical Integration

We solve the initial value problem shown in Eqn (4), (5), and (7) by numerical integration methods, such as Euler method, Runge-Kutta methood, or Dormand-Prince method [60]. These numerical integration methods can infer the continuous-time instantaneous hidden state that determined by network dynamics.

### 5.3.2 Adjacency Matrix

Instead of using geographical adjacency matrix that generated by prior geographical relationships, we adopt the trainable adaptive adjacency matrix to obtain the semantic relationships [25, 32, 33], which has been proved that can achieve better performance:

$$\mathbf{A} = TopK(Tanh(\sigma(\mathbf{M}_1 \cdot \mathbf{M}_2^\top))), \quad (15)$$

where $\mathbf{M}_1, \mathbf{M}_2 \in \mathbb{R}^{n \times n'}$ are trainable parameter matrices with $n' \ll n$, initialized by Xavier method [61]. $\sigma(\cdot)$ is the ReLU activation function. The $Tanh(\cdot)$ function here is to constrain the entries within 0 to 1 (together with the $ReLU(\cdot)$ function). $TopK(\cdot)$ function means that we will only retain a certain percentage of edges according to the weight, and other edges will be removed, the purpose of which is to ensure the sparsity of the adjacency matrix.

# 6 Evaluation

In this section, we conduct extensive experiments on four real-world datasets to answer the following questions:

- Q1. How does the STCDN performs in the temporal super-resolution forecasting (TSRF) tasks?

**Table 1** Basic information of datasets

| Datasets | #Vertices | Interval | #Timestamps |
|---|---|---|---|
| PeMSD3 | 358 | 5min | 26208 |
| PeMSD4 | 307 | 5min | 16992 |
| PeMSD7 | 883 | 5min | 16992 |
| PeMSD8 | 170 | 5min | 28224 |

- Q2. How does the STCDN performs in the conventional temporal equal-resolution forecasting (TERF) tasks?
- Q3. What the performance tendency of TSRF tasks is as the forecasting resolution magnification increases?
- Q4. Does encoding network dynamics in the encoder improve the forecasting performance?

## 6.1 Datasets

We will conduct conventional TERF on PeMSD3, PeMSD4, PeMSD7, and PeMSD8, TSRF and other experiments will be conducted on PeMSD4 and PeMSD8. Necessary information of datasets is given in Table 1. All datasets record the information of traffic flow every 5 minutes. These datasets are the benchmarks used in many existing studies [27–30, 62, 63].

According to these existing studies, for fairness, we adopt general solutions to preprocess these datasets. Firstly, we utilize the Z-score normalization to the input information $\mathcal{X}$:

$$\mathcal{X} := \frac{\mathcal{X} - mean(\mathcal{X})}{std(\mathcal{X})}, \tag{16}$$

where $mean(\cdot)$ and $std(\cdot)$ are the mean value and the standard deviation of the input information, respectively. All datasets aggregate records into 5-minute interval, and 288 timestamps per day.

In addition, following these studies, we split the training set, validation set, and test set for these datasets according to the chronological order by the corresponding ratio, *i.e.*, 60% for training set, 20% for validation set and 20% for test set.

## 6.2 Basic Experimental Introduction

We design two types of basic experiments to answer the Q1 and Q2, respectively.

### 6.2.1 Experimental Introduction of TSRF

Our model can perform TSRF tasks. Nonetheless, evaluating the quality of TSRF is a problem, since there is no information within recording intervals, thus, no ground-truth values can be provided to contrast and evaluate. In order

to generate the ground-truth values in the TSRF task, we select a compromised way: we expand the recording intervals by $k$ times for training, corresponding to the forecasting resolution magnification. Therefore, there are still extra available data that do not participate in the training within recording intervals. Meanwhile, these available data can be adopted as ground-truth to evaluate the forecasting within recording intervals. In summary, we need more ground-truth values than the supervision signals, to evaluate the super-resolution forecasting performance. Specifically, in this part, we let the forecasting resolution magnification $k = 3$, which means that the forecasting resolution in the evaluation phase will be tripled of training phase.

### 6.2.2 Experimental Introduction of TERF

The TERF task is a conventional task that widely conducted in previous studies [27–30, 62, 63]. We refer to the experimental settings in previous studies for fairness, using past 12 timestamps of historical observations to forecast traffic flow of future 12 timestamps, in which the recording intervals are 5 minutes.

## 6.3 Baselines

Up to our knowledge, there is no spatial-temporal modeling algorithm that dedicated to independently perform TSRF tasks. We divide all baselines into basic baseline models and interpolation models. The former class can complete TERF. We introduce a compromised way, imposing interpolation models on the forecasting results of these basic baseline models to simulate TSRF tasks.

### 6.3.1 Basic Baseline Models

- Vector Auto-Regression (**VAR**) [11], a time series model to capture the pairwise relationships among time series;
- Long Short-Term Memory (**LSTM**) [64], a classical variant of Recurrent Neural Networks (RNNs) for time series;
- Diffusion Convolutional Recurrent Neural Networks (**DCRNN**) [23], in which the spatial dependencies are captured by random walks, and the temporal dependencies are captured by RNNs;
- Spatial Temporal Graph Convolution Networks (**STGCN**) [24], which formulates the problem on graphs and builds the model with complete convolutional structures;
- Graph WaveNet (**GWN**) [25], a framework for Deep Spatial-Temporal Graph Modeling, which applys a learnable adaptive adjacency matrix to capture the hidden spatial dependency;
- Attention Based Spatial-Temporal Graph Convolutional Networks (**ASTGCN**) [27], which designs spatial attention and temporal attention mechanisms to model spatial and temporal dynamics, respectively;
- Spatial-Temporal Synchronous Graph Convolutional Networks (**STSGCN**) [28], which synchronously captures the spatial and temporal information by stacked graph convolutional neural networks.

- Spatial-Temporal Fusion Graph Neural Networks (**STFGNN**) [29], which adopt a data-driven temporal graph to compensate several existing correlations that spatial graph may not reflect.
- Spatial-Temporal Graph ODE Networks (**STGODE**) [30], an model that captures spatial-temporal dynamics through a tensor-based Ordinary Differential Equation.

#### 6.3.2 Interpolation Models

- Lagrange Interpolation (**LAG**), an interpolation algorithm based on polynomials [65];
- Slinear Interpolation (**SLI**), a spline interpolation of first order;
- Quadratic Interpolation (**QUA**), a spline interpolation of second order;
- Cubic Interpolation (**CUB**), a spline interpolation of third order [66];
- Linear Interpolation (**LIN**), an interpolation algorithm that the interpolation function is a polynomial of the first degree;
- Nearest Interpolation (**NEA**), an interpolation algorithm that selects the nearest information to perform interpolating.

Performing interpolating requires sampling points, which should be coarse-grained. These coarse-grained sampling points should be provided by other forecasting algorithms, referred to as basic model. In this paper, we select DCRNN [23], STGCN [24], Graph WaveNet [25], ASTGCN [27], and STGODE [30] as basic models, and perform interpolating based on the forecasting results of these models.

### 6.4 Experimental Settings

#### 6.4.1 Implementation Settings

Our experiments were conducted on the computer environments with Tesla V100 GPU cards. We implement our algorithm by PyTorch. Batch size is set as 32, the number of attention heads ($Z$) is 8, hidden dimension in our algorithm ($d$) is 128, learning rate is 0.0003. We incorporat Adam optimizer [67] to train our model. In the adjacency matrix, we retain 7.5% edges with the largest weight, and remove others. In computation, we select the 5-order Dormand-Prince method [60] for numerical integration. Notability, the numerical integration methods can be selected arbitrarily, *e,g.*, Euler method, Runge-Kutta method, *etc.*

#### 6.4.2 Evaluation Metrics

Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Errors (MAPE) are used as the evaluation metrics. Experiments on all datasets are conducted at least 5 times with different random seeds, the shown metrics are the mean value all experiments. Note that we prefer to refer to the experimental results of baselines given by authors if

any. Otherwise, we tune the hyper-parameters of baselines carefully, detailed settings of hyper-parameters for baselines are given on Github.

## 6.5 Experimental Results

### 6.5.1 Performance Comparison of TSRF

**Table 2** Performance Comparison of TSRF (**bold fonts** denote the best performance and undelines denote the second best performance)

| Datasets | Interpolations | | LAG | SLI | QUA | CUB | LIN | NEA |
|---|---|---|---|---|---|---|---|---|
| | Algorithms | Metrics | | | | | | |
| PeMSD4 | DCRNN | MAE | 32.88 | 28.63 | 30.53 | 29.13 | 30.96 | 30.67 |
| | | RMSE | 45.62 | 42.64 | 43.40 | 42.26 | 45.93 | 43.39 |
| | | MAPE(%) | 29.40 | 25.43 | 26.03 | 25.10 | 26.94 | 25.08 |
| | STGCN | MAE | 38.71 | 30.36 | 31.47 | 32.16 | 31.89 | 31.70 |
| | | RMSE | 55.93 | 48.73 | 47.52 | 48.29 | 45.28 | 47.23 |
| | | MAPE(%) | 42.02 | 30.28 | 33.28 | 34.77 | 31.16 | 31.53 |
| | GWN | MAE | 29.39 | 25.66 | 25.67 | 25.99 | 26.06 | 26.10 |
| | | RMSE | 43.86 | 38.24 | 38.10 | 38.28 | 38.50 | 38.61 |
| | | MAPE(%) | 21.34 | 20.05 | 20.01 | 19.40 | 19.99 | 19.39 |
| | ASTGCN | MAE | 29.72 | 28.85 | 28.46 | 27.60 | 28.34 | 28.36 |
| | | RMSE | 42.19 | 42.40 | 41.62 | 40.85 | 41.41 | 41.74 |
| | | MAPE(%) | 24.21 | 21.50 | 21.64 | 21.47 | 24.69 | 22.91 |
| | STGODE | MAE | 29.57 | 26.54 | 26.37 | 27.23 | 27.10 | 26.91 |
| | | RMSE | 42.60 | 38.21 | 39.44 | 39.51 | 40.71 | 40.33 |
| | | MAPE(%) | 22.34 | 19.19 | 20.22 | 19.31 | 19.52 | 19.47 |
| | STCDN | MAE | | | **24.23** (↓ 5.57%) | | | |
| | | RMSE | | | **36.15** (↓ 5.12%) | | | |
| | | MAPE(%) | | | **17.60** (↓ 12.05%) | | | |
| PeMSD8 | DCRNN | MAE | 28.07 | 23.58 | 23.03 | 23.80 | 23.94 | 23.01 |
| | | RMSE | 40.53 | 35.55 | 35.47 | 34.83 | 34.69 | 33.48 |
| | | MAPE(%) | 21.66 | 18.67 | 17.61 | 18.53 | 18.31 | 18.78 |
| | STGCN | MAE | 30.95 | 26.64 | 26.97 | 26.13 | 25.17 | 26.73 |
| | | RMSE | 46.51 | 38.21 | 40.23 | 39.52 | 41.73 | 42.43 |
| | | MAPE(%) | 31.28 | 20.33 | 21.38 | 23.17 | 22.78 | 21.12 |
| | GWN | MAE | 22.89 | 20.87 | 20.34 | 20.10 | 21.48 | 21.57 |
| | | RMSE | 34.91 | 31.14 | 31.60 | 31.70 | 31.01 | 31.20 |
| | | MAPE(%) | 20.37 | 14.81 | 15.32 | 15.41 | 14.90 | 14.69 |
| | ASTGCN | MAE | 27.83 | 25.28 | 25.66 | 25.24 | 25.48 | 25.57 |
| | | RMSE | 40.89 | 37.27 | 37.33 | 36.80 | 37.18 | 37.31 |
| | | MAPE(%) | 19.43 | 18.16 | 17.48 | 17.66 | 16.82 | 17.34 |
| | STGODE | MAE | 24.13 | 21.51 | 21.32 | 20.87 | 21.44 | 21.39 |
| | | RMSE | 36.67 | 32.58 | 32.33 | 31.89 | 33.64 | 32.70 |
| | | MAPE(%) | 20.68 | 15.31 | 14.64 | 14.01 | 14.88 | 14.41 |
| | STCDN | MAE | | | **19.05** (↓ 5.22%) | | | |
| | | RMSE | | | **29.43** (↓ 5.49%) | | | |
| | | MAPE(%) | | | **12.56** (↓ 14.50%) | | | |

The detailed comparison of TSRF is given in Table 2. Note that the comparison shown in the Table 2 is under the forecasting resolution magnification $k = 3$, other forecasting resolution magnifications will be analyzed in the subsequent section.

From the comparison, we can see that STCDN presents the best performance on TSRF tasks. Concretely, comparing STCDN with the second best performance, STCDN achieves averaged 7.58% and 8.40% performance improvements on the two datasets, respectively. Such a superiority mainly comes from the fundamental difference between STCDN and other basic baseline models.

Interpolation-based methods overally exhibit less performance. On one hand, the performance of these interpolation-based methods is highly dependent on the basic baselines that they rely on. Moreover, basic baselines that perform well on conventional TERF task are not consistent with the TSRF task. For instance, GWN can perform better on TSRF task when it is considered as a basic baseline, but it failed to present competitivity in TERF task as shown in Table 3. This results in selecting a ideal basic baseline a difficult and tricky affair. Meanwhile, Since different interpolation strategies take into account different orders, the performance of these methods also dependent on the selecting of interpolation strategies. The above two items make the construction of interpolation-based methods more like a time-consuming permutation and combination when performing the TSRF task. Even so, this approach did not achieve the desired performance.

In contrast, STCDN consideres the traffic flow on road network as a physical-guided way, *i.e.* continuous-time dynamical system. Whether classical physics-guided traditional theories [56], or human intuitive perception of traffic flow transitions [7, 8] have confirmed that comparing with conventional spatial-temporal models that directly fusing spatial information and temporal information [27–30, 62, 63], our proposed continuous-time transition of traffic flow can better reflect the nature of the road network as a complex physical system. Meanwhile, such a continuous-time nature can also more accurately approximate the real traffic flow at arbitrary timestamps in a more natural mean.

### 6.5.2 Performance Comparison of TERF

The performance comparison of TERF is given in Table 3, which has the conventional experimental settings. The comparison shows that our proposed model overall outperforms other baselines in four public traffic flow datasets on 10 out of 12 metrics with averaged 2.60% improvements. As we mentioned above, the modeling ideas of STCDN are fundamentally different the idea of directly fusing spatial information and temporal information, which more meets the natural evolution of traffic flow. Additionally, in the numerical integration methods, there will be a lot of temporary hidden states are generated, which can be regarded as feature augmentation. Therefore, we hypothesize that such a performance superiority mainly comes from the augmented features provided by the generated hidden states, which will be verified in the subsequent ablation studies.

Among these baselines, STFGNN [29] and STGODE [30] achieve the second best performance. The both, utilized DTW-augmented graphs to complete

**Table 3** Performance Comparison of TERF

| Datasets | PeMSD3 | | | PeMSD4 | | |
|---|---|---|---|---|---|---|
| Metrics / Algorithms | MAE | RMSE | MAPE(%) | MAE | RMSE | MAPE(%) |
| VAR | 23.56 | 38.26 | 24.51 | 24.54 | 38.61 | 17.24 |
| LSTM | 21.33 | 35.11 | 23.33 | 26.77 | 40.65 | 18.23 |
| DCRNN | 17.99 | 30.31 | 18.34 | 21.22 | 33.44 | 14.17 |
| STGCN | 17.55 | 30.42 | 17.34 | 21.16 | 34.89 | 13.83 |
| GWN | 19.22 | 32.77 | 18.89 | 24.89 | 39.66 | 17.29 |
| ASTGCN | 17.34 | 29.56 | 17.21 | 22.93 | 35.22 | 16.56 |
| STSGCN | 17.48 | 29.21 | 16.78 | 21.19 | 33.65 | 13.90 |
| STFGNN | 16.77 | 28.34 | 16.30 | 20.48 | 32.51 | 16.77 |
| STGODE | 16.50 | 27.87 | 16.69 | 20.84 | 32.82 | **13.77** |
| STCDN | **16.33** (↓ 1.03%) | **26.14** (↓ 6.21%) | **15.87** (↓ 2.64%) | **20.41** (↓ 0.34%) | **31.24** (↓ 3.91%) | 13.85 |
| Datasets | PeMSD7 | | | PeMSD8 | | |
| VAR | 50.22 | 75.63 | 32.22 | 19.19 | 29.81 | 13.10 |
| LSTM | 29.98 | 45.94 | 13.20 | 23.09 | 35.17 | 14.99 |
| DCRNN | 25.22 | 38.61 | 11.82 | 16.82 | 26.36 | 10.92 |
| STGCN | 25.33 | 39.34 | 11.21 | 17.50 | 27.09 | 11.29 |
| GWN | 26.39 | 41.50 | 11.97 | 18.28 | 30.05 | 12.15 |
| ASTGCN | 24.01 | 37.87 | 10.73 | 18.25 | 28.06 | 11.64 |
| STSGCN | 24.26 | 39.03 | 10.21 | 17.13 | 26.80 | 10.96 |
| STFGNN | 23.46 | 36.60 | **9.21** | 16.94 | 26.25 | 10.60 |
| STGODE | 22.59 | 37.54 | 10.14 | 16.81 | 25.97 | 10.62 |
| STCDN | **22.40** (↓ 0.84%) | **35.22** (↓ 3.77%) | 10.10 | **16.48** (↓ 1.96%) | **24.90** (↓ 4.12%) | **10.51** (↓ 0.85%) |

the feature augmentation. Therefore, the performance superiority of these two algorithms mainly comes from more powerful feature engineering. Interestingly, in previous comparison in TSRF, GWN based algorithms achieve strong performance with interpolation, but it do not perform well in TERF tasks. This also confirms the randomness and uncertainty of selecting the combination of basic baselines and interpolations when performing TSRF. Also, the experiment illustrates that STCDN can not only perform excellently in TSRF with significant performance improvements, it also present strong competivity in conventional TERF tasks.

Notability, our model significantly outperforms all baselines in metrics MAE and RMSE, but failed to achieve the best in MAPE. This is because of we modeled network dynamics is essentially a first-order ordinary differential equation mathematically. If only the first-order terms are considered, without incorporating higher-order terms, the modeled evolution trajactories of traffic flow will be smooth. Such a smooth nature will result in excellent average forecasting performance for all timestamps, but tends to be insensitive to mutations or jumps. This will be the future direction of our subsequent studies.

## 6.6 Parameter Sensitivity Analysis

To answer Q3, we will discuss the influence of forecasting resolution magnification $k$ on the accuracy of TSRF.

Based on the settings in the TSRF tasks, we will discuss the influence of forecasting resolution magnification $k$ ranges from 2 to 6 on PeMSD4 and

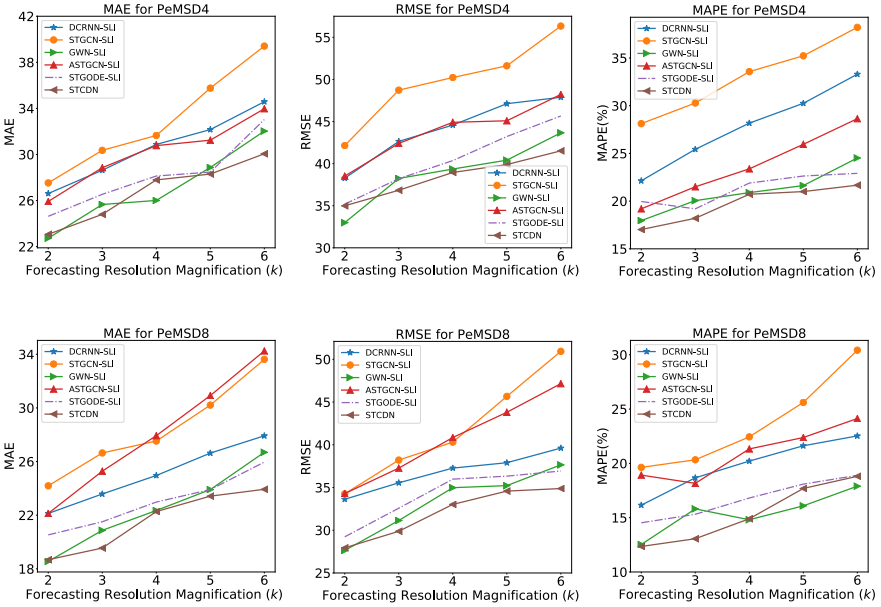


**Fig. 5** The forecasting performance comparison under different resolution magnifications.

PeMSD8 datasets, *i.e.* correspond to the cases of forecasting intervals of 150 seconds to 50 seconds respectively, under the 5-minute recording interval. As a comparison, we also illustrate the TSRF performance of basic models with Slinear Interpolation algorithms, which presents the overall best TSRF performance among all baselines.

The TSRF comparison under different resolution magnification is shown as Fig (5). Generally, the comparison illustrates that with the resolution magnification increases, the forecasting performance of all algorithms are getting worse, which is intuitive. Because a larger resolution magnification implies the more information should be forecasted, while the less information we known, thereby increasing the difficulty of accurate forecasting. Additionally, in our settings, we expand the recording intervals to simulate the case of lager intervals, which also leads to numerical increasement. But this does not prevent us from comparing the relative performance among models.

In the comparison, we can see that our model consistently achieves the best performance among these algorithms in the task. Meanwhile, we can see with the forecasting resolution magnification $k$ increases, the performance degradation of our algorithm is minimal overall. The experiment illustrate that even in the case of sparse recording data to be the supervision signals, STCDN can still perform accurate TSRF compared to other baselines.

### 6.7 Ablation Study

In this subsection, to answer the Q4, we will analyze how the existence of continuous-time hidden states that are generated by integrating the network dynamics affects forecasting performance.

To this end, we no longer regard the network dynamics, shown in Eqn (14), as dynamical information, but as a simple spatial information extractor. Therefore, we will not perform a definite integral in the temporal dimension. Insteadly, we incorporate the idea of DCRNN [23] that using an RNN block to handle the temporal information. The ablation model is referred to as STCDN(w/o dynamics). The experiment will conducted on all datasets, and all evaluation metrics are obtained via TERF tasks. Thus, STCDN follows the conventional strategy that fusing spatial and temporal information. Results are shown in Table 4.

From the comparison, we can see that the continuous-time hidden states from the view of network dynamics can enhance the forecasting performance with 4.50% improvements. We hypothesize such superiority can be explained from two aspects: (1). the continuous-time hidden states that are generated by the physical-guided way can better approximate the nature of the continuous evolution of traffic flow on the road network, enabling STCDN to have a better representation ability; (2). the continuous-time hidden states can also be regarded as a way of feature augmentation to improve forecasting performance.

**Table 4** Ablation Study of Network Dynamics

| Models | STCDN | | | STCDN(w/o dynamics) | | |
|---|---|---|---|---|---|---|
| Metrics / Datasets | MAE | RMSE | MAPE(%) | MAE | RMSE | MAPE(%) |
| PeMSD3 | **16.33** (↓ 4.00%) | **26.14** (↓ 11.12%) | **15.87** (↓ 9.83%) | 17.01 | 29.41 | 17.60 |
| PeMSD4 | **20.41** (↓ 1.93%) | **31.24** (↓ 1.21%) | **13.98** (↓ 1.62%) | 20.81 | 31.62 | 14.21 |
| PeMSD7 | **22.40** (↓ 6.75%) | **35.22** (↓ 6.11%) | **10.10** (↓ 6.31%) | 24.02 | 37.51 | 10.78 |
| PeMSD8 | **16.48** (↓ 0.36%) | **24.90** (↓ 2.28%) | **10.51** (↓ 2.41%) | 16.54 | 25.48 | 10.77 |

## 7 Conclusions and Future Work

In this paper, we notice that the widely applied recording way in traffic flow might result in the coarse-grained and incomplete nature of supervision signals, leading to the forecasting intervals being strictly limited by recording intervals. Therefore, we propose a new task named Temporal Super-Resolution Traffic Flow Forecasting to help the forecasting intervals eliminate the limitations of recording intervals. Specifically, we regard the traffic flow on the road network as a continuous-time dynamical system. By modeling the network dynamics

and incorporating the idea of ordinary differential equations, we can model the continuous-time hidden states, and further infer the traffic flow at arbitrary desired timestamps.

This is a novel attempt at making the forecasting intervals independent of the recording intervals, enabling traffic forecasting with more flexible intervals. Theoretically, such a continuous-time dynamical system based idea can be expanded to any domain that is related to time series, not just traffic flow forecasting. Therefore, it can also benefit more potential applications. On the other hand, we also need to confront some problems due to the immaturity of this solution for the novel task, such as insensitivity of mutations or jumps, and time-consuming. In the future, we will work to solve these problems, and improve the idea to better tackle such a novel problem.

## Acknowledgement

This work is funded in part by the National Natural Science Foundation of China Projects No. U1936213, and the Shanghai Science and Technology Development Fund No. 19DZ1200802. Project is also supported by Chinese National Key Laboratory of Science and Technology on Information System Security.

## References

[1] Boukerche, A.F.M., Wang, J.: Machine learning-based traffic prediction models for intelligent transportation systems. Comput. Networks **181**, 107530 (2020)

[2] Castro, P.S., Zhang, D., Chen, C., Li, S., Pan, G.: From taxi gps traces to social and community dynamics. ACM Computing Surveys (CSUR) **46**, 1–34 (2013)

[3] Chen, C., Jiao, S.H., Zhang, S., Liu, W., Feng, L., Wang, Y.: Tripimputor: Real-time imputing taxi trip purpose leveraging multi-sourced urban data. IEEE Transactions on Intelligent Transportation Systems **19**, 3292–3304 (2018)

[4] Chen, C., Ding, Y., Xie, X., Zhang, S., Wang, Z., Feng, L.: Trajcompressor: An online map-matching-based trajectory compression framework leveraging vehicle heading direction and change. IEEE Transactions on Intelligent Transportation Systems **21**, 2012–2028 (2020)

[5] Dufour, J.-M., Renault, É.: Short-run and long-rub causality in time series: Theory. (1995)

[6] Chen, T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. In: NeurIPS (2018)

[7] Zhang, Z., Li, M., Lin, X., Wang, Y.: Network-wide traffic flow estimation with insufficient volume detection and crowdsourcing data. Transportation Research Part C: Emerging Technologies **121**, 102870 (2020)

[8] Meng, C., Yi, X., Su, L., Gao, J., Zheng, Y.: City-wide traffic volume inference with loop detector data and taxi trajectories. In: Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 1–10 (2017)

[9] Berman, S.M.: Local times and sample function properties of stationary gaussian processes. Transactions of the American Mathematical Society **137**, 277–299 (1969)

[10] Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. ArXiv **abs/1704.01212** (2017)

[11] Cryer, J.D.: Time series analysis. (1986)

[12] Chandra, S.R., Al-Deek, H.: Predictions of freeway traffic speeds and volumes using vector autoregressive models. Journal of Intelligent Transportation Systems **13**, 53–72 (2009)

[13] Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L.: Predicting taxi–passenger demand using streaming data. IEEE Transactions on Intelligent Transportation Systems **14**, 1393–1402 (2013)

[14] Williams, B.M., Hoel, L.A.: Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. Journal of Transportation Engineering-asce **129**, 664–672 (2003)

[15] Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V.N.: Support vector regression machines. In: NIPS (1996)

[16] Lv, Y., Duan, Y., Kang, W., Li, Z.X., Wang, F.: Traffic flow prediction with big data: A deep learning approach. IEEE Transactions on Intelligent Transportation Systems **16**, 865–873 (2015)

[17] Koesdwiady, A.B., Soua, R., Karray, F.: Improving traffic flow prediction with weather information in connected cars: A deep learning approach. IEEE Transactions on Vehicular Technology **65**, 9508–9517 (2016)

[18] Jia, Y., Wu, J., Du, Y.: Traffic speed prediction using deep learning method. 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 1217–1222 (2016)

[19] Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: AAAI (2017)

[20] Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., Li, Z.J.: Deep multi-view spatial-temporal network for taxi demand prediction. In: AAAI (2018)

[21] Yu, H., Wu, Z., Wang, S., Wang, Y., Ma, X.: Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors (Basel, Switzerland) **17** (2017)

[22] Zonoozi, A., Kim, J.-j., Li, X., Cong, G.: Periodic-crn: A convolutional recurrent model for crowd density prediction with recurring periodic patterns. In: IJCAI (2018)

[23] Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv: Learning (2018)

[24] Yu, T., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In: IJCAI (2018)

[25] Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling. In: IJCAI (2019)

[26] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems **32**, 4–24 (2019)

[27] Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: AAAI (2019)

[28] Song, C., Lin, Y., Guo, S., Wan, H.: Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In: AAAI (2020)

[29] Li, M., Zhu, Z.: Spatial-temporal fusion graph neural networks for traffic flow forecasting. In: AAAI (2021)

[30] Fang, Z., Long, Q., Song, G., Xie, K.: Spatial-temporal graph ode networks for traffic flow forecasting. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (2021)

[31] Chai, D., Wang, L., Yang, Q.: Bike flow prediction with multi-graph convolutional networks. Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (2018)

[32] Zhao, J., Chen, C., Liao, C., Huang, H., Ma, J., Pu, H., Luo, J., Zhu,

T., Wang, S.: 2f-tp: Learning flexible spatiotemporal dependency for flexible traffic prediction. IEEE Transactions on Intelligent Transportation Systems (2022)

[33] Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., Zhang, C.: Connecting the dots: Multivariate time series forecasting with graph neural networks. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2020)

[34] Yao, H., Tang, X., Wei, H., Zheng, G., Li, Z.J.: Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In: AAAI (2019)

[35] Zhou, F., Li, L., Zhang, K., Trajcevski, G.: Urban flow prediction with spatial–temporal neural odes. Transportation Research Part C-emerging Technologies **124**, 102912 (2021)

[36] Jin, M., Zheng, Y., Li, Y., Chen, S., Yang, B., Pan, S.: Multivariate time series forecasting with dynamic graph neural odes. ArXiv **abs/2202.08408** (2022)

[37] Choi, J., Choi, H.-S., Hwang, J., Park, N.: Graph neural controlled differential equations for traffic forecasting. In: AAAI (2022)

[38] Wang, Z., Chen, J., Hoi, S.C.H.: Deep learning for image super-resolution: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**, 3365–3387 (2021)

[39] Li, J., Pei, Z., Zeng, T.: From beginner to master: A survey for deep learning-based single-image super-resolution. ArXiv **abs/2109.14335** (2021)

[40] Keys, R.: Cubic convolution interpolation for digital image processing. IEEE Transactions on Acoustics, Speech, and Signal Processing **29**, 1153–1160 (1981)

[41] Duchon, C.E.: Lanczos filtering in one and two dimensions. Journal of Applied Meteorology **18**, 1016–1022 (1979)

[42] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1874–1883 (2016)

[43] Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. 2010 IEEE Computer Society Conference on Computer Vision

and Pattern Recognition, 2528–2535 (2010)

[44] Liu, J., Yaghoobi, M.: Fine-grained mri reconstruction using attentive selection generative adversarial networks. ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1155–1159 (2021)

[45] Liang, Y., Ouyang, K., Sun, J., Wang, Y., Zhang, J., Zheng, Y., Rosenblum, D.S., Zimmermann, R.: Fine-grained urban flow prediction. Proceedings of the Web Conference 2021 (2021)

[46] Liang, Y., Ouyang, K., Jing, L., Ruan, S., Liu, Y., Zhang, J., Rosenblum, D.S., Zheng, Y.: Urbanfm: Inferring fine-grained urban flows. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2019)

[47] Shen, R., Xu, J., Bao, Q., Li, W., Yuan, H., Xu, M.: Fine-grained urban flow prediction via a spatio-temporal super-resolution scheme. In: APWeb/WAIM (2020)

[48] Li, K., Chen, J., Yu, B., Shen, Z., Li, C., He, S.: Supreme: Fine-grained radio map reconstruction via spatial-temporal fusion network. 2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 1–12 (2020)

[49] Liu, N., Ma, R., Wang, Y., Zhang, L.: Inferring fine-grained air pollution map via a spatiotemporal super-resolution scheme. Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers (2019)

[50] Luo, Y., Cai, X., Zhang, Y., Xu, J., Yuan, X.: Multivariate time series imputation with generative adversarial networks. In: NeurIPS (2018)

[51] Fortuin, V., Baranchuk, D., Rätsch, G., Mandt, S.: Gp-vae: Deep probabilistic time series imputation. In: AISTATS (2020)

[52] Luo, Y., Zhang, Y., Cai, X., Yuan, X.: E2gan: End-to-end generative adversarial network for multivariate time series imputation. In: IJCAI (2019)

[53] Zang, C., Wang, F.: Neural dynamics on complex networks. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2020)

[54] Rubanova, Y., Chen, R.T.Q., Duvenaud, D.K.: Latent odes for irregularly-sampled time series. ArXiv **abs/1907.03907** (2019)

[55] Boyce, W.E.: Elementary differential equations and boundary value problems / william e. boyce, richard c. diprima. (1996)

[56] Gerlough, D.L., Huber, M.J.: Traffic flow theory. Technical report (1976)

[57] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS (2014)

[58] Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. ArXiv **abs/1706.03762** (2017)

[59] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio', P., Bengio, Y.: Graph attention networks. ArXiv **abs/1710.10903** (2018)

[60] Dormand, J.R., Prince, P.J.: A family of embedded runge-kutta formulae. Journal of Computational and Applied Mathematics **6**, 19–26 (1980)

[61] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS (2010)

[62] Guo, K., Hu, Y., Qian, Z.S., Sun, Y., Gao, J., Yin, B.: Dynamic graph convolution network for traffic forecasting based on latent network of laplace matrix estimation. IEEE Transactions on Intelligent Transportation Systems **23**, 1009–1018 (2022)

[63] Li, Z., Xiong, G., Chen, Y.-y., Lv, Y., Hu, B., Zhu, F., Wang, F.: A hybrid deep learning approach with gcn and lstm for traffic flow prediction*. 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 1929–1933 (2019)

[64] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**, 1735–1780 (1997)

[65] Waring, E.: Vii. problems concerning interpolations. Philosophical Transactions of the Royal Society of London, 59–67

[66] Catmull, E.E., Rom, R.: A class of local interpolating splines. Computer Aided Geometric Design, 317–326 (1974)

[67] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2015)