Transportation Research Part C 128 (2021) 103185

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Multistep traffic forecasting by dynamic graph convolution: Interpretations of real-time spatial correlations

Guopeng Li *, Victor L. Knoop, Hans van Lint

Delft University of Technology, Stevinweg 1, 2628 CN Delft, the Netherlands

ARTICLE INFO

Keywords:
Traffic forecasting
Deep learning
Graph convolution
Network spatial correlation

ABSTRACT

Accurate and explainable short-term traffic forecasting is pivotal for making trustworthy decisions in advanced traffic control and guidance systems. Recently, deep learning approach, as a data-driven alternative to traffic flow model-based data assimilation and prediction methods, has become popular in this domain. Many of these deep learning models show promising predictive performance, but inherently suffer from a lack of interpretability. This difficulty largely originates from the inconsistency between the static input–output mappings encoded in deep neural networks and the dynamic nature of traffic phenomena. Under different traffic conditions, such as freely-flowing versus heavily congested traffic, different mappings are needed to predict the propagation of congestion and the resulting speeds over the network more accurately. In this study, we design a novel variant of the graph attention mechanism. The major innovation of this so-called dynamic graph convolution (DGC) module is that local area-wide graph convolutional kernels are dynamically generated from evolving traffic states to capture real-time spatial dependencies. When traffic conditions change, the spatial correlation encoded by DGC module changes as well. Using the DGC, we propose a multistep traffic forecasting model, the *Dynamic Graph Convolutional Network* (DGCN). Experiments using real freeway data show that the DGCN has a competitive predictive performance compared to other state-of-the-art models. Equally importantly, the prediction process in the DGCN and the trained parameters are indeed explainable. It turns out that the DGCN learns to mimic the upstream–downstream asymmetric information flow of typical road traffic operations. Specifically, there exists a speed-dependent optimal receptive field – which governs what information the DGC kernels assimilate – that is consistent with the back-propagation speed of stop-and-go waves in traffic streams. This implies that the learnt parameters are consistent with traffic flow theory. We believe that this research paves a path to more transparent deep learning models applied for short-term traffic forecasting.

1. Introduction

Accurate and reliable short-term traffic forecasting is one of the core functions in Intelligent Transportation Systems (ITS). Predicting the dynamic evolution of traffic has been a popular research topic for many decades, both on a single corridor (e.g. Van Lint et al. (2005)) and on large road networks (e.g. Fusco et al. (2016)). For broad and recent overviews of methods and challenges in this active research field we refer the readers to Lana et al. (2018) and Ermagun and Levinson (2018). The discussion hereafter focuses on

* Corresponding author.
E-mail addresses: G.Li-5@tudelft.nl (G. Li), V.L.Knoop@tudelft.nl (V.L. Knoop), J.W.C.vanLint@tudelft.nl (H. van Lint).

https://doi.org/10.1016/j.trc.2021.103185
Received 17 November 2020; Received in revised form 27 April 2021; Accepted 27 April 2021
Available online 25 May 2021

three key methodological challenges related to real-time traffic forecasting for travel information provision and traffic management: interpretability, observability, and uncertainty.

1.1. Background: network-level traffic prediction

The evolution of traffic network conditions can be viewed as a superposition of various intertwined dynamical processes, including origin–destination (O-D) traffic demand, route choice patterns, queuing and congestion backward propagation, driving behaviors that govern emerging characteristics, etc. Traffic condition forecasting requires either implicitly or explicitly considering the dynamics in both demand and supply. In the literature many different methods are proposed to tackle short-term traffic prediction tasks. They range from simple approaches such as conditional averaging (Davis and Nihan, 1991; Smith and Demetsky, 1997), auto-regression and time series models (Ahmed and Cook, 1979; Castro-Neto et al., 2009), to sophisticated machine learning approaches (e.g. Hamner (2010), Huang et al. (2014), Polson and Sokolov (2017)), and simulation-based approaches in conjunction with sequential data assimilation methods (Wang et al., 2005; Van Hinsbergen et al., 2011). One of the biggest advantages of data-driven approaches – which encompass all methods except those using simulation-based models – is that explicitly disentangling the demand and supply dynamics is not required. Available data can be fully explored regardless of whether the model simulates the physical process between the input data (speeds, flows, travel times, trajectories, and other information) and the desired output quantity. Instead, data-driven methods aim to learn correlations between inputs and outputs. Their performances mainly depend on the type of data, model structures, and the optimization strategy, whether these mappings are explainable in traffic science. However, the interpretability of data-driven models becomes a new problem. Taking the example of deep neural networks, neither the complex model containing numerous learnable parameters nor the internal states of hidden layers are easy to explain. If a traffic prediction model is only used as a "black-box" inference engine, it may suffice for many purposes, but certainly not for traffic management and control.

Simulation-based methods offer an alternative for explainable network-level traffic forecasting, since these methods explicitly delineate the prediction into constituent sub-problems and compute the resulting traffic state by "white box" physical and behavioral models. The simulation-based approach gives a comprehensive solution for traffic state estimation, prediction, and for control optimization (Wang et al., 2008). However, this transparent method also comes with serious challenges.

The first challenge is the limited observability of some constituent parameters and variables. There are much more unknowns that govern the traffic dynamics than the limited bits of information derived from sensors, let alone project them in the short-term future. Particularly, demand and route choice patterns are difficult to be fully observed (Viti et al., 2014; Castillo et al., 2012; Krishnakumari et al., 2019). Some other key variables and parameters, such as vehicle densities and road capacities, are also difficult or costly to observe with or derive from sensor data. Although with vehicle counting, accumulation and thus density is available in principle, inevitable small errors may lead to unbounded biases (Xie et al., 2018; Nantes et al., 2016; Bhaskar et al., 2014).

Secondly, even if all required quantities were observable, because of the stochastic nature of traffic dynamics, the prediction is still influenced by uncertainty. The uncertainty comes from the predicted boundary conditions (e.g. demand and capacity constraints) and the traffic flow process itself. Small errors in the projected inflow at a certain position, or in the expected capacity of an intersection, may cause large prediction errors in the resulting traffic states. Additionally, where and when errors are made also matters. These errors may have larger consequences at some specific locations and during some time slots. For example, the spreading of congestion in a road network largely depends on whether queues spill over some junctions and off-ramps, and these differences can be significant (van Lint et al., 2012; Knoop et al., 2015).

To address these challenges, it is necessary to combine the best of both data-driven and simulation-based worlds: an adaptive learning solution that uses whatever available incomplete data and produces predictions that are at least partly explainable.

1.2. Contributions and outline

In this paper, we investigate deep learning models that are able to learn explainable dynamic spatial correlations, which can be defined as traffic-condition-dependent interactions among adjacent links. In deep learning models, spatial correlation is generally implicitly modeled by stacked convolutional or graph convolutional layers. To make this process more explainable, we propose a novel graph attention variant in the form of Dynamic Graph Convolution (DGC) modules. DGC modules assume that the spatial correlation depends on the road network connectivity, locations, and traffic states in a receptive field. In other words, it learns to mimic the location-specific and condition-dependent propagation of information. DGC modules are implemented in a recurrent encoder-decoder model to realize multistep speed forecasting. We call this prediction model a *Dynamic Graph Convolutional Network* (DGCN). By tracking the generated dynamic kernels, we explain what spatial correlations are learnt by this model in different scenarios. The key contributions of this paper are:

- **Model construction**: The dynamic convolution is extended to graphs by considering the properties of traffic networks. Based on this new DGC module, a novel multistep traffic forecasting model, the DGCN, is proposed.
- **Performance comparison**: Validated on real freeway network datasets, the DGCN shows competitive predictive accuracy compared to a selection of state-of-the-art models.
- **Model interpretation**: The generated dynamic convolutional kernels offer insights into what the model has learnt from the data. It turns out that DGC modules encode an explainable proxy to the basic traffic flow theory.

The rest of this paper is organized as follows: Section 2 describes the details of the DGCN. Next, Section 3 compares the

performances of DGCN against some baseline models using real freeway-network speed datasets. Section 4 presents model interpretations. Finally, Section 5 draws conclusions and gives several potential research directions.

2. Method: deep learning models

In this section, we first give a brief overview of deep learning models applied to short-term traffic prediction. How spatial correlations are implemented in different models is our focus. Then, specific to our purpose, some notations and the problem formulation are defined in Section 2.2. Next, we introduce the basic building block, the DGC module, and combine it with a RNN encoder-decoder to construct the novel DGCN prediction model in Section 2.3.

2.1. Overview of related works

Recently, deep learning, especially deep neural networks (DNN), brought new possibilities to the short-term traffic forecasting domain. DNNs capture spatiotemporal features by a large amount of organized trainable parameters. Many papers propose a variety of DNN models to improve predictive performance. For example, Huang et al. (2014) proposes deep belief networks (DBN) for traffic flow prediction. Polson and Sokolov (2017) combines a linear model with $L1$ regularization and a layer activated by a *tanh* function to predict the sharp transition of traffic flow. Increasingly, standardized modules, such as convolutional layers and recurrent layers, are used to construct new DNN based predictors. For example, Ma et al. (2015) uses long-short term memory (LSTM) networks to predict speed evolution on a corridor. Ma et al. (2017) converts traffic dynamics to heat map images and employs deep convolutional neural networks (CNN) for speed prediction. To apply CNN on non-Euclidean structures, the convolutional operator is extended to graph convolution (Kipf and Welling, 2017; Hamilton et al., 2017). Such graph convolutional networks (GCN) are suitable for network-level traffic forecasting because a road network can be naturally represented by a graph. For example, Yu et al. (2018) combines spectral-domain graph convolution and temporal gated convolution to predict network-level traffic speed, so-called spatio-temporal graph convolutional networks (STGCN). In many of these studies, the central focus is on improving forecasting precision and reliability, and much less on model interpretation. To improve the interpretability of deep learning models, an increasing number of studies attempt to explain what neural networks learn from data, and particularly spatial correlations.

In the literature there are two main methods to achieve this. The first one is implementing predefined components to represent spatial correlations. These components stay *static* after training. The idea is similar to those variants of regression models considering spatial interactions (e.g. Kamarianakis and Prastacos (2005), Min and Wynter (2011)). For example, Li et al. (2018) considers congestion as a bidirectional diffusion process described by random walks. Diffusion convolution is embedded into gated recurrent units (GRU) resulting in a so-called diffusion convolutional RNN (DCRNN). Cui et al. (2019) proposes traffic graph convolution (TGC) to extract rich spatial features on small graphs. The design of this TGC considers the maximum speed of vehicles and results show that statistically larger kernel weights emerge at frequently congested intersections. Zhang et al. (2019) combines a similar graph convolutional layer, a GRU encoder-decoder, and a temporal attention layer in one model, and calls this an attention graph convolutional-seq2seq (AGC-seq2seq) model. The authors define the average matrix of graph convolutional kernels as spatial dependencies and draw similar conclusions as in Cui et al. (2019). This approach can give a *static* mapping that encodes spatial dependencies that works best on average. However, spatial dependencies are fundamentally *dynamic* in traffic flows. This inconsistency restrains these models' interpretability.

The second option is letting the model learn *dynamic* spatial correlations from data. The graph attention mechanism (Velickovic et al., 2018) is a promising solution in this direction. In graph attention networks (GAT), graph convolutional kernels are no longer static, but are calculated from real-time node features. For example, Zhang et al. (2018) proposes gated attention networks (GaAN) to predict traffic states. An extra soft gate is added based on GAT. Do et al. (2019) uses a simplified pairwise spatial attention layer to predict traffic volumes and directly produce dynamic spatial correlations in the form of heat maps. However, some important features of traffic flows are ignored when designing these models. First, the propagation of information is asymmetric in traffic networks. In freely-flowing areas, kinematic waves move with the driving direction; in congested traffic they move against the traffic flow. In the former case upstream information suffices for prediction; in congestion the model needs to combine up- and downstream information. Second, how far up- or downstream the model would have to "look for information" to accurately track the evolution of congestion depends on the data granularity and the network topology. Third, some unavoidable "unseen" factors, such as an on-ramp branch that is not covered by sensors, can also influence the prediction. The ability to explain how a model gives reliable predictions by adjusting dynamic spatial correlations for incomplete data is critically important for decision-making in traffic control. Implementing the properties above may yield a more accurate and more explainable mapping.

To develop dynamical mappings for these spatio-temporal relationships, we need a spatial attention mechanism that adapts in real time to the prevailing traffic conditions. This study aims to fill this gap by designing a new spatial attention variant considering the upstream–downstream asymmetry of traffic dynamics and the influence of incomplete data.

2.2. Mathematical formulations and preliminaries

Traffic dynamics on a road network can be written as a spatio-temporal graph

$$\mathcal{G}(\mathcal{V}_N, \mathcal{E}, \mathbf{A}_{N\times N}; \mathbf{X}_{T\times N\times C}), \tag{1}$$

in which the subscripts indicate the dimension of a tensor. $\mathscr{V}_N$ is the set of N nodes, $\mathscr{E}$ is the set of edges (connecting these nodes), and $\mathbf{A}$ is the adjacency matrix. For T discrete time steps of duration Δt, the evolution of traffic states is represented by the feature tensor $\mathbf{X}$. The feature vector of node i at time t is denoted as $\mathbf{X}_i^t \in \mathbb{R}^C$, which can be composed of C traffic variables, such as speed, flow, etc. Short-term traffic forecasting is formulated as a sequence-to-sequence regressive task in this study. On a road network $\mathscr{G}$, the input is the feature tensor observed in the past m timesteps $\mathbf{X}_{m\times N\times C}^{\text{obs}}$, the output is the predicted feature tensor of the next p timesteps $\widehat{\mathbf{X}}_{p\times N\times C}^{\text{pred}}$ that maximizes the following conditional probability:

$$\widehat{\mathbf{X}}_{p\times N\times C}^{\text{pred}} = \underset{\mathbf{X}_{p\times N\times C}^{\text{real}}}{\text{argmax}} \; Pr\left(\mathbf{X}_{p\times N\times C}^{\text{real}} | \mathbf{X}_{m\times N\times C}^{\text{obs}};\ \mathscr{G}\right) \tag{2}$$

We denote $m\Delta t$ as the observation window and $p\Delta t$ as the prediction horizon. On a spatiotemporal graph $\mathscr{G}$, we denote the set of all nodes within k hops from a central node v_i as $\mathscr{N}_i^k$ (including the central node itself), in which k is the so-called *receptive field*. If the latent representation of v_i is a learnable function of all feature vectors within the receptive field, we have:

$$\overrightarrow{y}_i = \varphi_{\text{agg}}\left(\overrightarrow{x}_{\mathscr{N}_i^k};\ \mathscr{G}\right) \tag{3}$$

in which $\varphi_{\text{agg}}()$ is a *k-walk graph aggregator*.

2.3. Dynamic graph convolutional networks

One of the earliest dynamic convolutional networks is proposed by Brabandere et al. (2016), the so-called Dynamic Filters Network (DFN). In that model, pixel-wise dynamic convolutional kernels are used to capture the movement of objects and to predict the next several frames of a video. Based on a similar idea, we design a dynamic graph convolution to capture the propagation of congestion. The structure is shown in Fig. 1. The input of the DGC is split into two tensors, feature maps $\mathbf{X}^{'}$ and traffic states $\mathbf{X}$. Each DGC module has three parts: (1) a filters generation network (FGN) computes dynamic graph convolutional kernels from varying feature maps; (2) the generated kernels are then applied in a local-wide graph convolution with traffic states; (3) post-processing adjusts the dimension of outputs (if necessary).

The DGC module differs from the DFN (Brabandere et al., 2016) in three ways. First, it is extended from Euclidean space to graphs by involving an adjacency matrix that encodes the connectivity. This extension is necessary because the number of neighbor nodes is arbitrary on a graph. Another method is converting graphs to Euclidean images by stitching all roads in order and applying CNN (Ma et al., 2017). However, the neighbors around an intersection on the graph may locate far away on the image. For extracting explainable spatial correlations from DNN, GCN is a better choice because the real geographical distances among roads are conserved. Second, the FGN uses a more computationally efficient one-step process instead of the heavy convolutional encoder-decoder. Third, in the FGN of the DFN, convolutional kernels are shared among all pixels. Considering the fact that each node is unique in a road network in terms of infrastructure, capacity, and link connectivity, we add position-specific parameters in the FGN of the DGC. The mathematical formula of FGN is given below:

$$\begin{cases} \mathbf{A}_{\mathbf{k}_{N\times N}} = Ci[(\mathbf{A}_{N\times N})^k] \\ \mathbf{S}_{N\times N}(t) = (\mathbf{A}_{\mathbf{k}_{N\times N}} \odot \mathbf{B}_{N\times N})\mathbf{X}_{N\times C}^{'}(t)\mathbf{\Gamma}_{C\times N} + \mathbf{b}_N \\ \mathbf{W}_{N\times N}(t) = \sigma_1(\mathbf{A}_{\mathbf{k}_{N\times N}} \odot \mathbf{S}_{N\times N}(t)) \end{cases}, \tag{4}$$

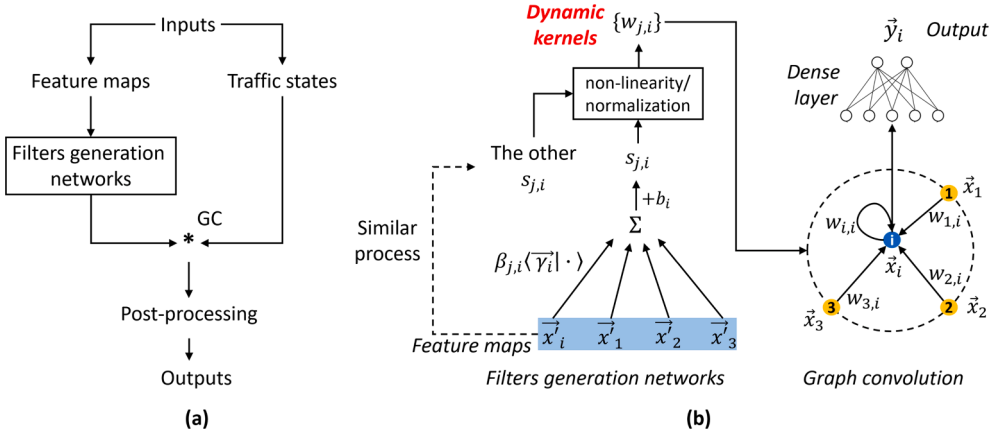

Fig. 1. DGC module: (a) the structure of a DGC module; (b) the details of filters generation networks and the DGC graph aggregator.

where $\mathbf{B}, \mathbf{\Gamma}$ and $\mathbf{b}$ are trainable parameters. $Ci[]$ pins all non-zero elements in a matrix to 1. $\mathbf{A_k}$ is the k-hop adjacency matrix. $\sigma_1()$ is the nonlinear activation or normalization function. Because the input feature map $\mathbf{X}^{'}(t)$ varies at each instant, the generated graph convolutional kernel $\mathbf{W}(t)$ also evolves with time. Then the generated dynamic kernels $\mathbf{W}(t)$ are applied in a space-domain graph convolution and a shared node-wise fully-connected layer is used for post-processing:

$$\begin{cases} \mathbf{H}_{N\times D} = (\mathbf{W}_{N\times N}(t) \odot \mathbf{A}_{\mathbf{k}_{N\times N}})\mathbf{X}_{N\times D} \\ \mathbf{Y}^{\text{out}}_{N\times C'} = \sigma_2\left(\mathbf{H}_{N\times D}\mathbf{V}_{D\times C'} + \mathbf{b}^{'}_{C'}\right) \end{cases}, \tag{5}$$

in which $\mathbf{V}$ and $\mathbf{b}^{'}$ are trainable parameters. $\sigma_2()$ is an activation function. To better clarify this process and for easier readability, Eq. (6) depicts the graph aggregator in (4) and (5) for a single node i:

$$\begin{cases} s_{j,i}(t) = \left(\sum_{j\in\mathscr{N}_i^k} \beta_{j,i}\left\langle \overrightarrow{\gamma}_i, \overrightarrow{x^{'}}_j\left(t\right)\right\rangle\right) + b_i \\ w_{j,i}(t) = \sigma_1\left(\{s_{j,i}(t) \mid j \in \mathscr{N}_i^k\}\right) \\ \overrightarrow{h}_i = \sum_{j\in\mathscr{N}_i^k} w_{j,i}(t)\overrightarrow{x}_j \\ \overrightarrow{y}_i = FC_{\theta_{c'}}\left(\overrightarrow{h}_i\right) \end{cases} \tag{6}$$

The first equation in (6) describes how the initial dynamic graph convolutional kernels $s_{j,i}$ are generated from feature vectors. It also indicates the parameter sharing strategy: $\overrightarrow{\gamma}_i$ is the shared parameter in the very receptive field. $\beta_{j,i}$ represents the speciality of each node when generating initial weights. Each $s_{j,i}$ is a learnable k-walk graph aggregator of $\overrightarrow{x^{'}}_{\mathscr{N}_i^k}$. For simplification, in the following parts we denote a DGC module with $N \times C^{'}$ output dimension and k receptive field on a spatiotemporal graph $\mathscr{G}$ as follows:

$$\mathbf{Y}^{\text{out}}_{N\times C'}(t) = \text{DGC}_k\left(\mathbf{X}^{'}(t), \mathbf{X}(t), C^{'}; \mathscr{G}\right) \tag{7}$$

The DGC proposed here is essentially a variant of the graph spatial attention module (Zhu et al., 2019). In this section hereafter, we compare it with another variant, the *Graph Attention Network* (GAT) depicted by (8) and proposed in (Velickovic et al., 2018). We only discuss the single-head attention variant here, but the conclusions also hold for multi-head attention.

$$\begin{cases} \phi_w\left(\overrightarrow{x}_i, \overrightarrow{x}_j\right) = \left\langle FC_{\theta_{va}}\left(\overrightarrow{x}_i\right), FC_{\theta_{vb}}\left(\overrightarrow{x}_j\right)\right\rangle \\ w_{j,i} = \dfrac{\exp\left(\phi_w\left(\overrightarrow{x}_i, \overrightarrow{x}_j\right)\right)}{\sum_{j\in\mathscr{N}_i^k}\exp\left(\phi_w\left(\overrightarrow{x}_i, \overrightarrow{x}_j\right)\right)} \\ \overrightarrow{y}_i = FC_{\theta_o}\left(w_{i,i}FC_{\theta_{va}}\left(\overrightarrow{x}_i\right) + \sum_{j\in\mathscr{N}_i^k, j\neq i} w_{j,i}FC_{\theta_{vb}}\left(\overrightarrow{x}_j\right)\right) \end{cases} \tag{8}$$

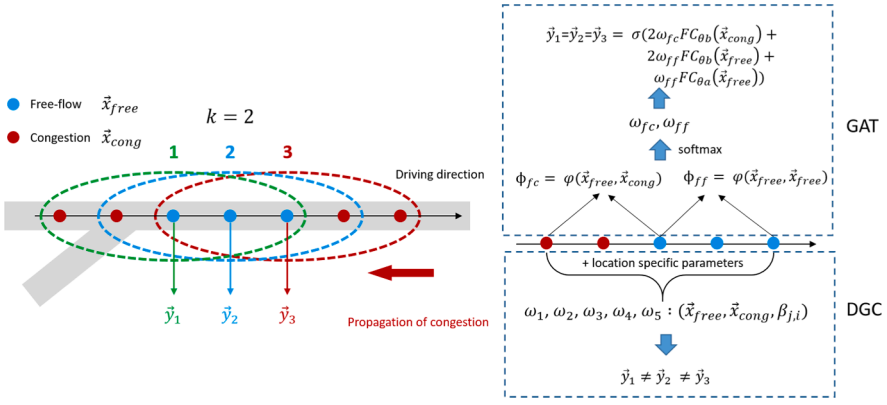

Fig. 2. An example of directional effects: if $k = 2$, each receptive field contains three free-flow nodes and two congested nodes. GAT and GaAN give the same hidden representation for the three central nodes, but DGC can distinguish them.

Different from DGC, the GAT module computes pairwise similarities between a central node and its neighbors, and then applies *softmax* normalization in each receptive field. Rearranging the feature vectors of neighbor nodes does not change the output. Although GAT can distinguish most types of node orders by overlapping multiple adjacent receptive fields, there do exist traffic phenomena where GAT may fail to learn the right spatial correlations. One example is the upstream–downstream asymmetry. For a given road segment x_i it matters whether an upstream road segment or a downstream segment is congested. In the first case (upstream congestion) this may not affect the segment x_i at all, whereas in the other case we can expect this congestion to propagate upstream and thus affect the conditions on x_i. The GAT module cannot distinguish these two cases. Other models, such as GaAN (Zhang et al., 2018), have the same limitation. At least in theory, the DGC module can explicitly learn these asymmetries.

Fig. 2 presents a clarifying example. Consider a small corridor represented by seven nodes as shown on the left. Our focus is on the outputs of the three most central nodes here, $\overrightarrow{y_1}, \overrightarrow{y_2}, \overrightarrow{y_3}$. For simplicity sake, we assume that all nodes can exhibit only two traffic states, a freely-flowing state represented by $\overrightarrow{x}_{\text{free}}$ versus a congested state $\overrightarrow{x}_{\text{cong}}$. We also consider a fixed size of receptive field, $k = 2$. From a traffic flow theory perspective the short-term evolution of these three nodes are related but quite different. Congestion propagates against the direction of flow (from right to left). Only in the case node 3 indeed becomes congested, the probability that node 2 becomes congested later in time increases dramatically. Only in that case node 2 also becomes congested, and later in time still, node 1 may become congested as well. But the three nodes are not likely to be influenced by the upstream (left) congestion because there may be a stationary bottleneck, such as an on-ramp. Thus, for short-term prediction purposes, these three nodes need to be treated differently. However, GAT modules cannot achieve this. Because the initial weight ϕ is computed from the central node and one neighbor node, only two dynamic weights, ω_{ff} (for free-flow neighbor nodes) and ω_{fc} (for congested neighbor nodes) will be generated. No matter how these 2 congested nodes and 2 freely-flowing nodes are ranged, the last equation in (8) gives the same output, see Fig. 2 right top. The DGC module, in contrast to GAT, caters for these directional dynamics *by making the local weights conditional to the relative location ($\beta_{j,i}$) of all neighbor nodes*, as shown in Fig. 2 right bottom. Each node has its unique weight. We refer to Eqs. (6) versus (8) for comparing DGC and GAT for a single receptive field.

The RNN encoder-decoder (Kumar and Asger, 2018) is widely used for seq2seq regression tasks, including applications for traffic prediction (e.g. Van Lint et al. (2005)). Each RNN cell receives the current input and the hidden state from last time step $(\mathbf{X}^t, \mathbf{h}^{t-1})$, generates the new output and hidden state $(\widehat{\mathbf{X}}^{t+1}, \mathbf{h}^t)$. The encoder saves the last hidden state as a context vector $\mathbf{C}$. The context vector is copied as the first hidden state to initialize the decoder. Usually, two RNN cells of the same type but with different parameters are used in the encoder and the decoder respectively. Based on DGC modules, we propose a novel multistep traffic forecasting model, the *Dynamic Graph Convolutional Network* (DGCN). The model architecture and the inner structure of the novel RNN cell are shown in Fig. 3. It combines a DGC module with a regular graph convolutional GRU-like (GCGRU) cell, where gates in GRU are replaced by different space-domain static graph convolutions. For each time step, the current traffic state is concatenated with the hidden state to form feature maps. The feature maps and the current traffic state are fed into DGC module to give the prediction. Then the prediction and the hidden state are put into GCGRU to produce new hidden state. Mathematical formulas are given in (9). Scheduled sampling (Bengio et al., 2015) is an effective curriculum learning strategy to mitigate the discrepancy between training and inference phases. In training, the decoder uses either the ground-truth or the prediction given by last time step. The probability of using the ground-truth, ϵ, is gradually reduced to 0 with iterations during training so the model can start with learning from the correct answers and ends with giving the predictions by itself. In inference and testing, ϵ is reset as 0.

$$\begin{cases} \mathbf{X}^{t+1} = \mathrm{DGC}_k\left(\left[\mathbf{X}^t, \mathbf{h}^{t-1}\right], \mathbf{X}^t, 1; \mathscr{G}\right) \\ \mathbf{r}^t = \sigma\left(\mathrm{GC}_{r,k'}\left(\left[\mathbf{X}^{t+1}, \mathbf{h}^{t-1}\right], \mathscr{G}\right)\right) \\ \mathbf{u}^t = \sigma\left(\mathrm{GC}_{u,k'}\left(\left[\mathbf{X}^{t+1}, \mathbf{h}^{t-1}\right], \mathscr{G}\right)\right) \\ \mathbf{c}^t = \tanh\left(\mathrm{GC}_{c,k'}\left(\left[\mathbf{X}^{t+1}, \left(\mathbf{r}^t \odot \mathbf{h}^{t-1}\right)\right], \mathscr{G}\right)\right) \\ \mathbf{h}^t = \left(1 - \mathbf{u}^t\right) \odot \mathbf{h}^{t-1} + \mathbf{u}^t \odot \mathbf{c}^t \end{cases} \tag{9}$$

To make the values of dynamic kernels more stable and to give clear congestion boundaries, the adjacency matrix masking process (the 3rd equation in (4)) is replaced by Eq. (10) to realize *softmax* normalization in each receptive field (Vaswani et al., 2017). So the model is encouraged to explore the most important one link in each receptive field and the kernel values are bounded between 0 and 1:

$$\mathbf{W}_{N\times N}(t) = softmax\left(\mathbf{S}_{N\times N}(t) - 10^{15}\left(1 - \mathbf{A}_{\mathbf{k}_{N\times N}}\right)\right) \tag{10}$$

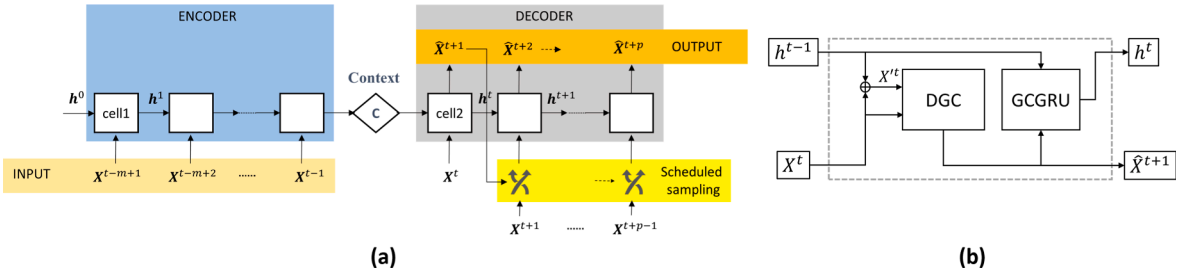

Fig. 3. Architecture of DGCN model: (a) RNN encoder-decoder; (b) The inner structure of a DGCN cell.

In summary, we propose a DGC module to capture directional dynamics in traffic data. Subsequently, we implement these DGC modules in a RNN encoder-decoder to realize multistep traffic forecasting. In the next section we will benchmark this new model against existing multi-step forecasting models.

3. Experiments

In this section, the proposed model is tested on real-world freeway networks. All data used in this study are provided by *National Data Warehouse for Traffic Information* (NDW, The Netherlands). Raw speed data are collected from loop detectors on freeways. The well-known adaptive smoothing method (ASM) (Treiber and Helbing, 2002; van Lint and Hoogendoorn, 2010) is used to fill the missing points in the sensor dataset.

3.1. Data description and benchmark models

The freeways around two big cities, Rotterdam and Amsterdam (Netherlands), are selected as research targets. The major clock-wise and counter clock-wise beltways around Rotterdam are respectively named as "`RotCL`" and "`RotCC`". The two beltways are uniformly partitioned into 200 m links. `RotCL` contains 199 links and `RotCC` contains 208 links. The more complex freeway network around Amsterdam which contains multiple intersections is noted as "`AMSnet`". `AMSnet` covers a larger area so it is uniformly partitioned into 400 m links to reduce complexity. `AMSnet` has 201 links in total. These networks are shown in Fig. 4. Each link is represented by a node to construct graphs. Speed data of the entire year of 2018 are prepared. All holidays and weekends are removed due to the lack of congestion. 27 highly-congested weeks without severe sensor malfunction are selected. The chosen weeks are shuffled and partitioned into 3 groups: 18 weeks for training, 4 weeks for validation, and 5 weeks for testing. Only afternoon-evening peak hours between 14:00–19:00 are included in the dataset because this time period contains the most diverse and richest patterns, which makes this traffic forecasting task more challenging.

A traffic forecasting formulation is determined by 4 parameters: time interval Δt, spatial resolution l, observation steps m, and prediction steps p. For the two beltways we fix the average length of each link $l = 200$ m and define two different tasks:

- *Task-1: shorter time interval*: Data is aggregated every 2 min ($\Delta t = 2$ min), observation window is 30 min ($m = 15$) and the maximum prediction horizon is 20 min ($p = 10$).
- *Task-2: longer time interval*: Data is aggregated every 5 min ($\Delta t = 5$ min), observation window is 60 min ($m = 12$) and the maximum prediction horizon is 30 min ($p = 6$).

The four corresponding datasets are noted as `RotCC2`, `RotCC5`, `RotCL2`, and `RotCL5` (network + time interval). For `AMSnet`, we only consider *Task-1*. The DGCN has two hyperparameters: the receptive field of the DGC module k, and the GCGRU module k'. We fix $k' = 2$ and tune k for different predictive tasks and networks by grid-searching. For the two ring freeways, the optimal receptive fields of *Task-1* and *Task-2* are $k = 4$ and $k = 9$, respectively. The optimal receptive field for `AMSnet` is $k = 2$. To evaluate the performance of the DGCN, it is compared with the following benchmark models:

- **Historical average** (HA): HA is a simple model that calculates the average speed of historical data as predictions. If the training set is fixed, HA gives the same prediction everyday and the performance does not change with prediction horizon.
- **K-nearest neighbors** (KNN): K-nearest neighbors regression (Denoeux, 1995) calculates the predictions based on the K most similar patterns in the training set. The similarity is measured by their Euclidean distance (RMSE) and predictions are computed through the weighted average of the corresponding future evolution. Weights are normalized by the inverse of Euclidean distance. The optimal hyperparameter $K = 25$ is chosen by grid search and cross validations.
- **DCNN**: Here we use the depth-4 model proposed in Ma et al. (2017) as a regular deep CNN baseline model. The beltways are cut off at a position and treated as a corridor. The roads in `AMSnet` are stitched together to form an image. Compared to GNN-based models, some adjacent pixels on these images may represent two geographically distant positions.
- **GAT**: The GAT module proposed by Velickovic et al. (2018) is implemented in GRU cells. This model also has a recurrent encoder-decoder structure. To guarantee a fair comparison, its receptive field is set the same as the DGCN. The feature enhancement layer in GAT has 5 output units.
- **DCRNN**: This predictor proposed by Li et al. (2018) is one of the state-of-the-art multistep traffic forecasting model. Here we use the same hyperparameters of the dual-random walk model in their paper. This is a reasonable choice because the number of nodes and the scale of our road networks are comparable to theirs. Like the DGCN, this model also has a recurrent encoder-decoder structure.
- **STGCN**: This model proposed in Yu et al. (2018) is another state-of-the-art single-step traffic forecasting model combining spatial graph convolution and gated temporal convolution. When giving multistep predictions, the observation window moves with the updated new prediction step by step. Here we use the variant that shows the best predictive accuracy in Yu et al. (2018), called STGCN(Cheb).
- **Graph Wavenet**: This model proposed by Wu et al. (2019) combines the diffusion convolution in DCRNN and the fully-convolutional structure of STGCN for multistep predictions. An additional learnable self-adaptive adjacency matrix is used to capture dependencies among links. But this self-adaptive adjacency matrix is static after training.

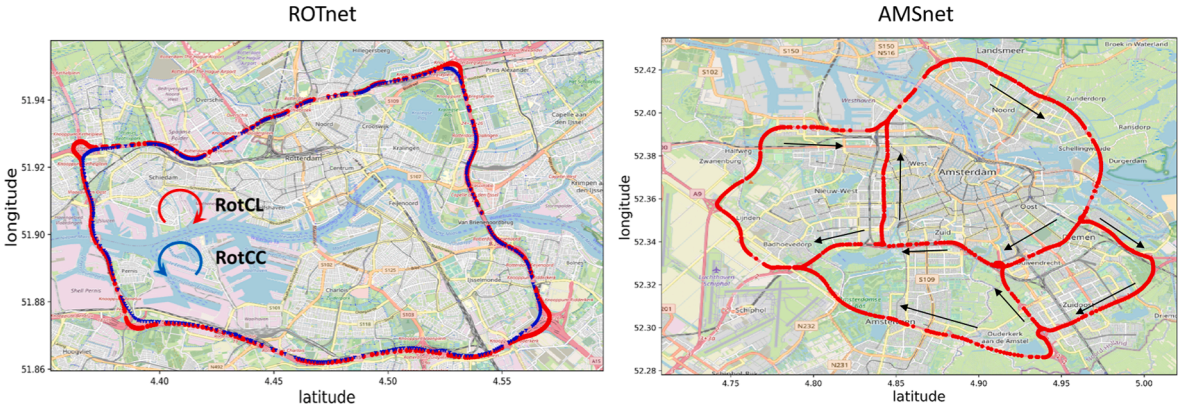

Fig. 4. The freeway networks around Rotterdam and Amsterdam. The arrows represent driving directions.

When training the models, all speed data is firstly normalized between 0 and 1 by dividing by the speed limit 120 km h^{-1}, then normalized by z-score function. For testing, normalized data are used as input, but the output is fed into a reverse process to give true-value predictions. The errors are calculated by taking several distance measures between predictions and the corresponding ground-truth. We choose three error measures: MAE, MAPE and RMSE. The mean average error (MAE) measures the overall accuracy; the mean average percentage error (MAPE) is particularly sensitive to errors in the contours of low-speed areas (i.e. whether the models accurately track the spatio-temporal boundaries of congestion patterns); and the root mean square error (RMSE) provides a combined measure (bias + variance) for the uncertainty in the predictions. The three metrics are used to evaluate the performance from different aspects. In training, RMSE is chosen as the loss function because of the z-score normalization. The *Adam* optimizer (Kingma and Ba, 2017) is chosen to minimize the loss function. Initial learning rate, decay rate, and scheduled sampling parameters are tuned for each model. Early stopping on the validation set is used to mitigate overfitting. Our experimental platform has one GPU (NVIDIA GeForce GTX 1070, 16 GB). All deep learning models are trained and tested through parallel computation on GPU. The source code, datasets, hyperparameters setting, visualization and model interpretation tools are available online: [1].

3.2. Results

Tables 1 and 2 list the overall average predictive errors of the proposed model and benchmark models on different datasets. Because the sizes of two beltways are close, we do not distinguish their training time but give the average value. Fig. 5 further shows the relation between each single-step errors and the prediction horizon on `RotCC` and `AMSnet`. They present how errors accumulate with prediction steps. The tendency is the same for another freeway `RotCL`. Because DNN models show significantly higher predictive accuracy than HA and KNN on `AMSnet`, we only compare deep learning models in Fig. 5. We observe the following phenomena from the results:

- DGCN or STGCN(Cheb) achieves the best overall multistep predictive accuracy in terms of the three metrics in most forecasting tasks. They show comparable performances on the two beltways and `AMSnet`. Especially, DGCN outperforms all the other models in terms of RMSE, which is the optimization target in training. It means that DGCN is better at tackling rare congestion patterns with higher uncertainty.
- In Fig. 5, for the beltway `RotCC`, the slope of KNN curve is the smallest in terms of the three accuracy metrics. The predictive errors of STGCN(Cheb) increase the fastest with prediction steps, especially for RMSE. The other models are between them. The reason is that STGCN(cheb) is a one-step prediction model employing a fully-convolutional structure. But all the other GNN-based deep learning models need to balance the multistep predictive errors and the prediction horizon. DCNN also has a regular fully-convolutional structure, but all predictive steps are generated together as an image so the multistep errors are also balanced. Although the overall average errors of DGCN and STGCN(Cheb) are close, STGCN(Cheb) performs better for shorter-horizon prediction while DGCN is more accurate for longer-horizon prediction.
- Having the similar GRU-like structures containing graph attention modules, DGCN consumes less running time than GAT but consistently gives better predictions. The computational efficiency originates from the parameters sharing in DGC module: the calculation of pairwise similarities is replaced by collective matrix operations.
- Generally speaking, deep learning models give better predictions than HA and KNN. This advantage is more significant on `AMSnet` than on the two ring freeways because the congestion patterns on a net-like graph are more complex due to spatial correlations among roads. DNN models consume longer training time than HA and KNN. However, the running time can be further reduced by using more powerful GPU clusters.

[1] https://github.com/RomainLITUD/DGCN_traffic_forecasting.

Table 1
Multistep predictive performances comparison on two ring freeways during peak hours 14:00–19:00.

Model	MAE(km h^{-1})		MAPE(%)		RMSE(km h^{-1})		Time(s)
freeway	RotCC	RotCL	RotCC	RotCL	RotCC	RotCL	Average
shorter time interval: $\Delta t = 2$ min; $p = 10$							
HA	9.83	9.81	12.92	13.74	16.39	16.02	72
KNN	6.95	6.29	17.66	14.90	12.94	11.38	240
DCNN	5.84	5.18	12.07	10.94	10.42	9.04	600
GAT	6.04	5.39	13.44	11.73	10.98	9.34	1100
DCRNN	5.41	5.28	12.53	10.84	9.93	9.00	350
STGCN(Cheb)	5.16	**4.56**	11.28	**9.23**	10.69	8.90	750
Graph Wavenet	5.35	4.91	12.02	10.93	9.92	8.73	900
DGCN	**5.07**	4.68	**11.23**	9.30	**9.78**	**8.53**	650
longer time interval: $\Delta t = 5$ min; $p = 6$							
HA	9.60	9.61	**12.62**	13.39	16.26	15.85	62
KNN	7.51	6.84	18.32	15.49	13.78	12.09	27
DCNN	7.25	6.28	15.07	12.62	12.63	10.74	280
GAT	7.59	5.76	13.96	11.50	14.31	10.15	500
DCRNN	6.26	5.70	14.44	12.89	11.40	10.10	250
STGCN(Cheb)	**5.66**	5.34	12.79	10.97	11.40	10.03	400
Graph Wavenet	6.46	5.62	14.61	10.83	11.79	9.63	500
DGCN	5.90	**5.15**	12.70	**10.31**	**11.12**	**9.57**	350

Table 2
Multistep predictive performances on AMSnet: $\Delta t = 2$ min, $m = 15$, $p = 10$.

Model	MAE(km h^{-1})	MAPE(%)	RMSE(km h^{-1})	Time(s)
HA	5.98	17.74	11.61	65
KNN	4.65	14.42	11.57	330
DCNN	4.14	11.02	8.52	610
GAT	3.86	10.15	8.17	1500
DCRNN	3.82	9.36	8.05	550
STGCN(Cheb)	3.76	9.48	8.03	820
Graph Wavenet	**3.67**	9.31	8.11	1040
DGCN	**3.67**	**8.98**	**7.83**	680

To summarize, we showed that DGCN is able to give satisfactory predictions on real-world datasets. The results support the theoretical discussion in Section 2: DGC performs better than GAT when tackling complex directional patterns. Next we will focus on the most important advantage of the proposed model: DGCN has high interpretability.

4. Model interpretation

Now we explain what DGCN learns from data by looking inside of DGC module. Section 4.1 relates the optimal receptive field to the back-propagation speed of stop-and-go waves, and shows again that the upstream–downstream asymmetry of models has significant influence on traffic forecasting performance. Section 4.2 directly gives the relation between dynamic graph convolutional kernels and speed data. It shows how DGC "understands" traffic dynamics in different situations.

4.1. Optimal receptive field

To explore how predictive errors change with receptive field, we formulate a series of forecasting tasks: the observation step $m = 10$ and prediction step $p = 3$ are fixed. The time interval changes from 1 min to 5 min by 1 min. For each of the five tasks above, the receptive field k gradually increases from 1 to 12. Thus, we can obtain five MAE-k curves. In Fig. 6, the five curves are overlapped

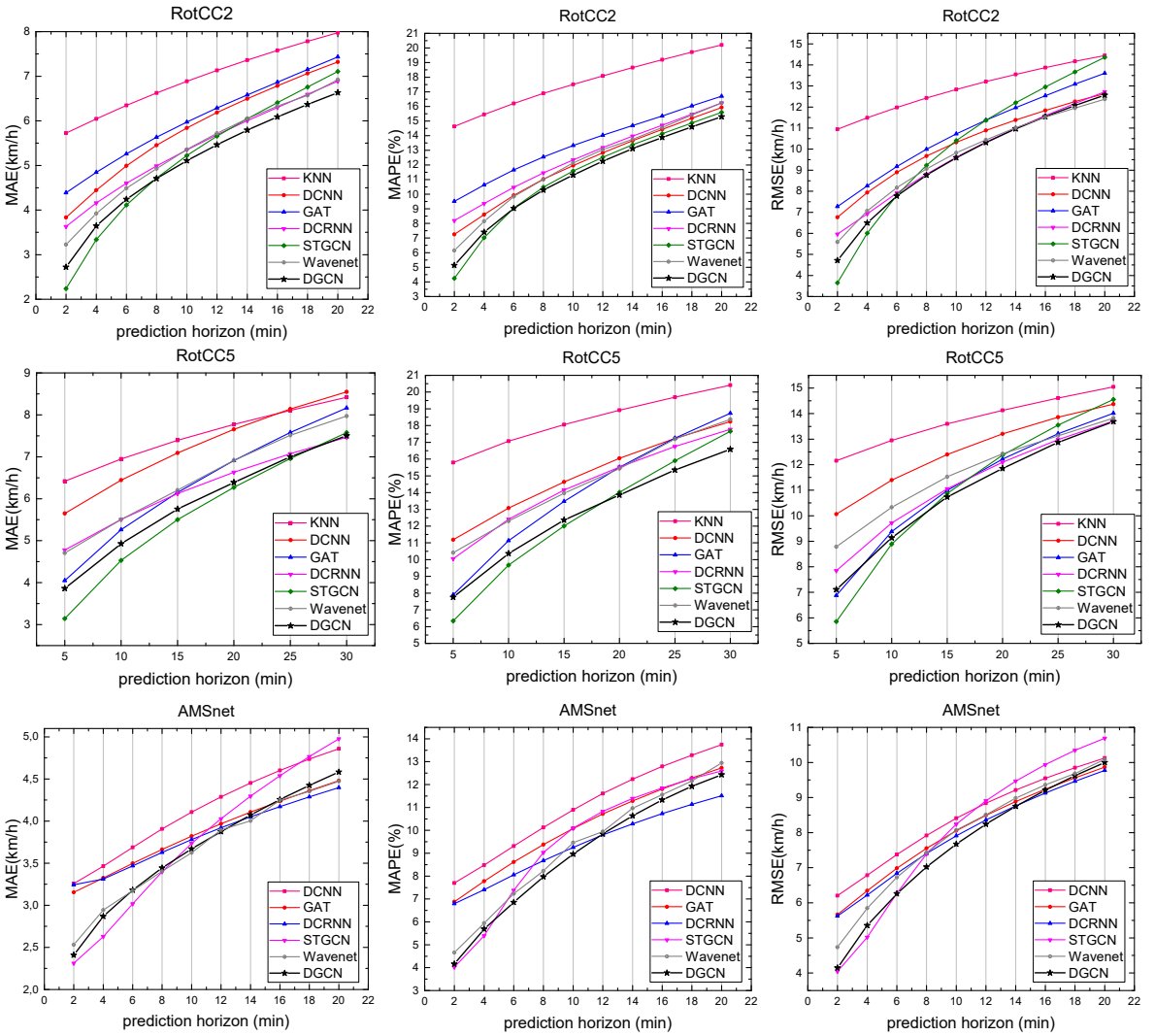

Fig. 5. The relations between MAE/MAPE/RMSE and the prediction horizon for each single time step.

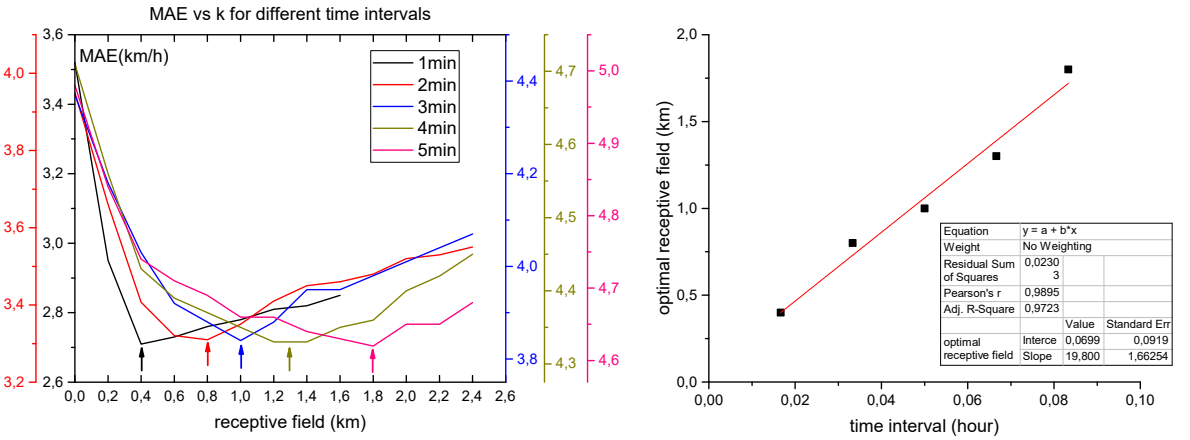

Fig. 6. Left: MAE-receptive field relations for different time intervals on `RotCC`; Right: the linear relation between the optimal receptive field and the time interval.

together. Notice that the x-axis is the real physical receptive field, $k \times l$. The left figure shows that the five curves have the similar tendency: with the increasing of receptive field, MAE drops rapidly until it reaches an optimal point, then increases slowly. The longer the time interval is, the larger the optimal receptive field is. The right figure shows that the relation between Δt and k_{opt} can be approximated by a linear equation:

$$k_{opt} \times l \approx 19.8\text{km}\,\text{h}^{-1}\Delta t \tag{11}$$

To explain this relation and visualize the effect of receptive field directly, we set different k values, re-train the DGCN models on `RotCC2`, and compare their predictions with the ground-truth. The results are shown in Fig. 7 ($\Delta t = 2$ min). For a small receptive field $k = 2$, the model tends to predict the back propagation of stop-and-go waves but the constrained receptive field results in mismatching patterns. The speed of stop-and-go wave measured on this prediction is about 13 km h^{-1}, which is lower than the true value. As for $k = 4$, the receptive field is large enough so the model can give precise predictions with clear stop-and-go wave boundaries. Even the merging of congestion (see the area in the circle) is successfully predicted because DGC considers the specialty of each location. DGC learns from historical data that congestion frequently stops spreading at that position because there is an off-ramp, even this off-ramp is not included in the dataset. When k is too large, for example $k = 8$, each receptive field contains multiple stop-and-go waves so the model is confused about which wave will cause the incoming congestion. So the congestion boundary becomes blur. Fig. 7 clearly illustrates how DGC mimics the directional flow of information. To predict the traffic state at a position after Δt, only the most adjacent stop-and-go wave should be included in the receptive field. Thus, we can estimate the maximum speed of stop-and-go waves by an empirical inequality:

$$k_{\text{opt}} \geqslant |v_s|_{\max} \frac{\Delta t}{l} \tag{12}$$

Comparing (12) with (11), we can estimate the upper bound of the speed of stop-and-go waves on `RotCC` is about 19.8 km h^{-1}. In Fig. 7, this estimated average speed is about 18 km h^{-1}. Because the receptive field is discretized by links' length $l = 0.2$km in our model, 19.8 km h^{-1} is a reasonable upper bound. This analysis explains why we choose $k = 4, k = 9$ for the two ring freeways and $k = 2$ for `AMSnet` in our experiments. It also gives a hyperparameter tuning principle in a traffic forecasting model. For a road network, if the time interval is small enough to show the fine structure of stop-and-go waves, the shorter links are, or the higher v_s is, the bigger receptive field should be chosen.

4.2. Dynamic spatial correlations

In many deep learning based traffic forecasting models, temporal dependencies and spatial correlations are modeled separately by different modules (e.g. Cui et al. (2019), Zhang et al. (2019), Yu et al. (2020)). But in DGCN, the spatial correlations depend on historical and current traffic states. For simplification, we take the first decoder cell as an example. The encoder encrypts observed traffic conditions in the past $m-1$ time steps into the context vector $\mathbf{C}$, and $\mathbf{C}$ is concatenated with the current input $\mathbf{X}^t$. Thus, each

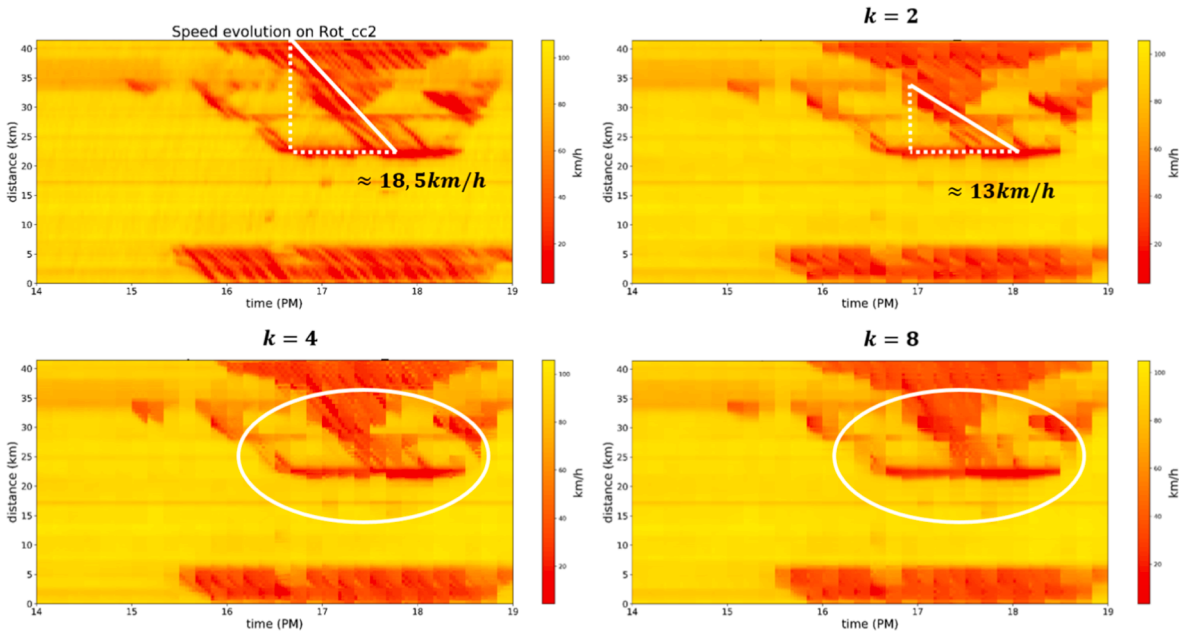

Fig. 7. Ground-truth (top left) and 10 min predictions given by DGCN with different receptive field k during peak hours on `RotCC2`: an example on 15-01-2018.

node is associated with two features, (c_i, x_i). They represent the historical information and the current traffic condition respectively. Because of the *softmax* normalization in DGC, all weights are non-negative and the sum of weights in each receptive field equals 1. So we can interpret generated dynamic graph kernels $\mathbf{W}(t)$ as relative spatial correlations. For example, $W_{i,j}(t)$ represents the "influence" of node-*j* on node-*i* for the prediction of next time step $t+1$ (see Fig. 1). We call it an "influence coefficient". To help visualizing multi-dimensional dynamic filters in a single figure, we define a directional distance vector on the ring freeways:

$$\overrightarrow{d}_k = \left(-k, -k+1, -k+2, \ldots, -1, 0, 1, \ldots, k-2, k-1, k\right) \tag{13}$$

For node *i*, the spatial correlation vector is defined by:

$$\overrightarrow{J}^t_i = \left(W^t_{i,i-k}, W^t_{i,i-k+1}, W^t_{i,i-k+2}, \ldots, W^t_{i,i+k-2}, W^t_{i,i+k-1}, W^t_{i,i+k}\right) \tag{14}$$

The filter value is defined by their inner product divided by *k*:

$$f^t_i = \frac{1}{k}\left\langle \overrightarrow{d}_k, \overrightarrow{J}^t_i \right\rangle, \quad f^t_i \in \left[-1, 1\right] \tag{15}$$

For more complex graphs, the definition is exactly the same, but the numbers of downstream/upstream adjacent nodes (the length of $\overrightarrow{d_k}$ and $\overrightarrow{J}^t_i$) depend on locations. Following the concept of spatial attention, f^t_i is called an *attention coefficient*. It is a variable of space and time. More negative f^t_i means that upstream links are more important, while more positive f^t_i indicates that traffic states of downstream links dominate the prediction. In the following parts, spatial correlation vectors $\overrightarrow{J}^t_i$ and attention coefficients f^t_i extracted from decoder cells are compared with real-time predictions v^{t+1}_i to seek model interpretations.

Fig. 8 shows the relations between the average attention coefficient $\bar{f}$ and the predicted speed *v* for different datasets. $\bar{f}$ is calculated over all links in the same speed range, which is uniformly aggregated every 5 km h^{-1}, from 0 km h^{-1} to 120 km h^{-1}. In Fig. 9, each column of the heat map represents the average spatial correlation vector $\overrightarrow{\bar{J}}$ in the corresponding speed range. Here we only show `RotCC` as two clarifying examples in Fig. 9 because the number of adjacent links is arbitrary on `AMSnet`. Fig. 10 further directly compares the ground-truth of speed and the corresponding evolution of dynamic attention coefficient on `RotCC2` during the peak hour of a randomly selected workday. In Fig. 11, we select two representative scenarios on `AMSnet`: an on-ramp and an off-ramp. The speed is compared with the real-time spatial attention distribution to precisely show how DGC gives predictions under these more complex situations. The following conclusions can be drawn from the results:

1. Given that congestion is defined by a speed value lower than 70 km h^{-1}, then Fig. 8 shows that the average attention coefficient $\bar{f}$ gradually shifts to higher positive value with the decreasing of *v*. It means that DGC tends to credit higher spatial importance to downstream links in low-speed regions. We call this phenomenon *attention transition*.
2. In Fig. 9, the heat map shows that in freely-flowing areas the average spatial attention distributes evenly around central links, while for congested areas spatial attention gradually converges to farther downstream links with the decreasing of speed (the top-left red spot). We call this phenomenon *attention convergence*.
3. Attention transition and convergence statistically explain how DGC "understands" traffic dynamics. In case all adjacent links within the receptive field of a specific link *i* are freely flowing, the model infers that link *i* will remain in a free-flow state in the next time step. But when congestion occurs, the model switches attention to downstream links to assimilate information that describes

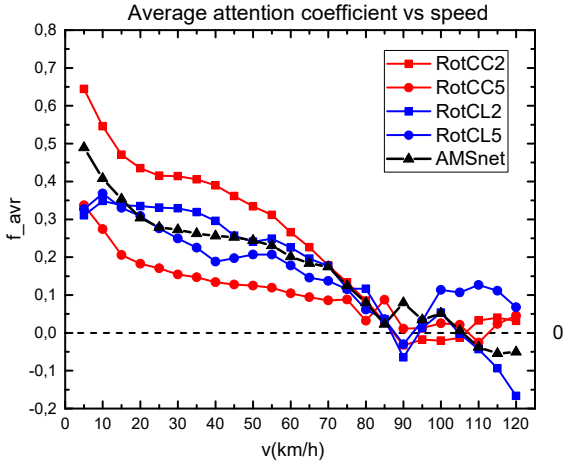

Fig. 8. $\bar{f}$-*v* relations on different datasets.

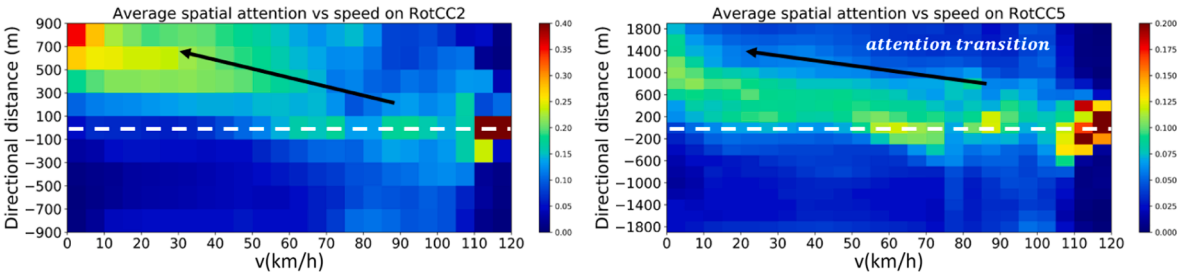

Fig. 9. $\vec{J}$-v relations on `RotCC`: each column is the average $\vec{J}$ in that speed range. y-axis is the directional distance.

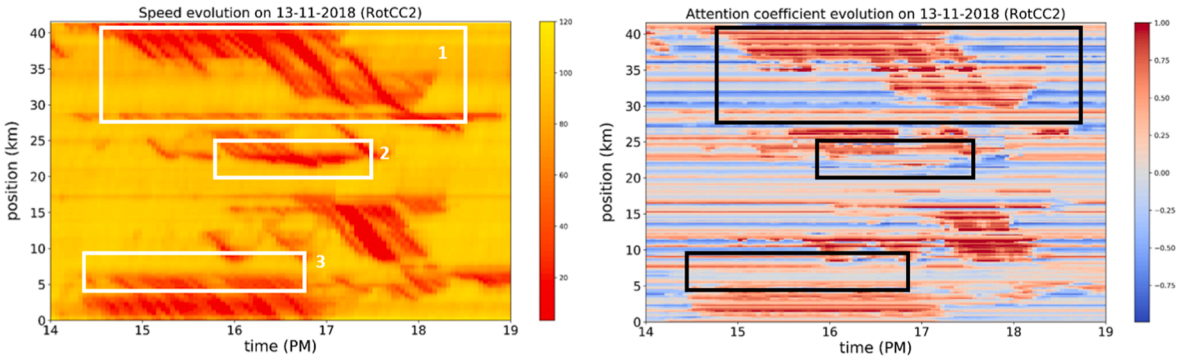

Fig. 10. Comparison between the speed ground-truth and the dynamic attention coefficient.

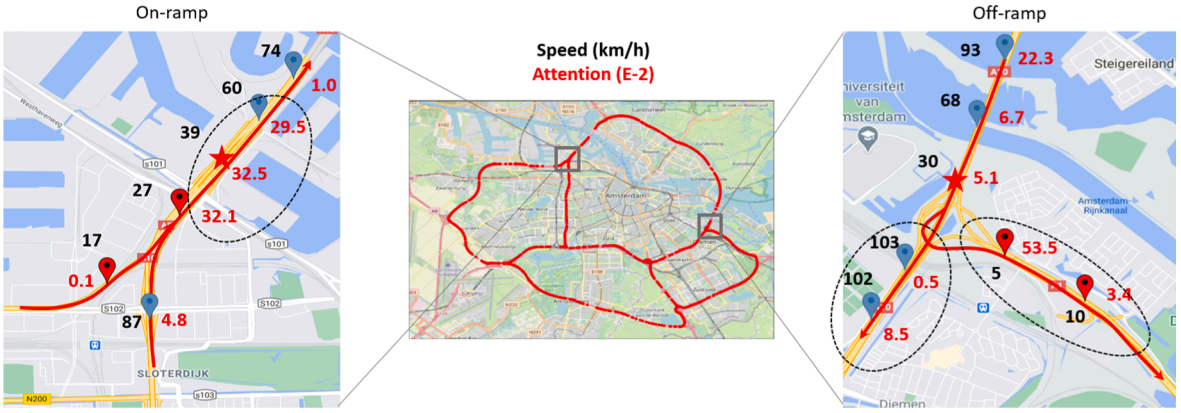

Fig. 11. Comparison between the speed ground-truth and the dynamic attention on `AMSnet` ($k = 2$): The left figure is an on-ramp and the right one is an off-ramp. The arrows represent driving directions (downstream). Stars represent the central nodes in the receptive field. Black numbers are speed, red numbers are dynamic attention values. Red nodes and blue nodes respectively represent congested and freely-flowing traffic states.

backward propagating jam waves and other congestion dynamics. The more speed decreases, the farther away downstream the attention is shifted. This mechanism is consistent with basic traffic flow theory and in line with the analysis in Section 4.1.

4. Fig. 10 presents how DGCN learns the unique property of each location. In general, as explained above, the dynamic graph convolutional kernels pay more attention to downstream links in low-speed congested areas, like area-1. But at some special positions, like in area-2 and 3, even the speed is very low, the attention still keeps neutral (close to 0) to stop the back propagation of stop-and-go waves and to resist the change of traffic states. These positions are important on-ramps causing bottlenecks or off-ramps stopping the shock waves to spill-back. DGCN learns these infrastructure differences solely from incomplete data. DFN and GAT cannot do this.
5. Fig. 11 shows similar but more complicated phenomena. For the on-ramp on the left, the left branch from the west is congested but another branch from south and the downstream links are free-flowing. The DGC deducts that there will be a standing bottleneck due to the merging of two traffic streams. So the spatial attention focuses on the central node (neutral). For the off-ramp shown in the right figure, the downstream branch to southeast is congested but another branch and upstream links are not congested. The DGC

automatically increases its attention to the congested downstream branch but credits very low attention to the other one. It predicts that the jam wave will spill back and continue spreading. The analysis shows that DGC is able to treat more complex graphs and to give explainable real-time spatial correlations. It allows delicately studying the role of intersections in a network.

4.3. Discussion on model interpretability

So far we have explained what dynamic spatial correlations the DGC has learned from traffic data and the resulting statistical relationships between trainable parameters and speed. We emphasize that the attention convergence and transition phenomena highlighted in the results section are *emerging* results of the learning process. In this final subsection we return to how this relates to traffic dynamics, which can be compactly described by the LWR kinematic wave model (Lighthill and Whitham, 1955):

$$\begin{aligned} &\frac{\partial \rho(x,t)}{\partial t} + \nabla_x q\left(x,t\right) = s\left(x,t\right) \\ \Rightarrow &\frac{\partial \rho(x,t)}{\partial t} + c\left(\rho\right)\nabla_x \rho\left(x,t\right) = s\left(x,t\right) \end{aligned} \tag{16}$$

in which ρ is density, q is flow, $c(\rho) = \partial q/\partial \rho$ is the kinematic wave speed, and s is the source term governing in- and outflows at on-ramps and off-ramps. Once traffic density increases beyond some critical density ρ_c, the wave speed $c(\rho) < 0$, which means kinematic waves move against the direction of flow. In flows with densities below ρ_c, disturbances move *with* the flow ($c(\rho) \geqslant 0$). Recall that this *directional* dynamics is used to justify the location-specific design of the DGC module. It turns out this choice really pays off. Using the spatial attention mechanism, the DGC module indeed seems to have encoded traffic state-dependent wave speeds which determine which up- and downstream nodes offer the most relevant information for the prediction. In freely-flowing conditions, it uses a mix of up- and downstream information; in congestion, it assimilates more information from downstream. This dynamic state-dependent behaviour cannot be achieved by the benchmark models.

Another tentative possibility is to consider the DGC model as an alternative to real-time macroscopic simulation models (e.g. Wang et al. (2005), Van Hinsbergen et al. (2011)), which combine continuous traffic flow models as described by (16) with sequential Bayesian estimators (Kalman filters, particle filters, etc). Li et al. (2018) explains that diffusion graph convolution is equivalent to the discrete form of a Laplacian operator (a second order differential) on graphs. Similarly, extended spatial-domain graph convolution can also be treated as the discrete form of an unknown nonlinear spatial operator with a source term parameterized inside:

$$\frac{\partial \rho}{\partial t} + \mathscr{S}_x\left(\rho, s\right) = 0 \tag{17}$$

To approximate this equation and the implicit mapping in $\mathscr{S}_x$ by neural networks, some basic rules need to be fulfilled. For one, the kinematic wave model in (16) is a conservation equation. Nothing is conserved in the relation (17) learnt by DGCN, unless one would explicitly add such conservation rules. A related issue is that the maximum speed of jam waves should constrain the adaptive receptive field. In many practical cases, however, flow data are not available, whereas speeds are increasingly available, both through infrastructure-based sensing and through probe vehicle data. The dynamics of average speed can be described as the superposition of a convective process and other higher order effects due to location specific reasons. Our results suggest that these dynamics can be more or less learnt by DGCN just from data.

5. Conclusion and outlook

In this paper, we demonstrated a multistep short-term traffic forecasting model using dynamic graph convolution. The key feature of the proposed model is that it combines the road graph embedded in their structures and the RNN component which allows learning explainable traffic dynamics.

Experiments on freeway networks show that the proposed model, DGCN, can give satisfactory short-term predictions. The core innovation, the dynamic graph convolution (DGC) module, is a "grey-box" that allows us to unravel what heuristic dynamic spatial correlations the model has learnt. Two points are demonstrated. First, designing suitable directional features extraction module and choosing a so-called adaptive receptive field are critically important for predicting the fine structures of congestion patterns. Second, we observed two phenomena, graph attention transition and convergence, that show how the DGC learns dynamic spatial correlations that make sense from a traffic flow perspective. The model has learnt speed-and-location-dependent, kinetic wave speeds so as to assimilate the right data under the right conditions. In practice, this model can give both network-level traffic state prediction and dynamic spatial dependencies among links. The predicted states can be used for prediction-based services, such as estimated time of arrival and congestion forewarning. In traffic control, the spatial correlations can help practitioners to diagnose where the predicted congestion spreads from, which is important for transparent decision-making.

This study leads to several potential research directions. For the deep learning technique itself, arguably the dynamic graph convolution module is not deep enough. The mechanism can be easily extended to multi-head attention, or multi-scale DGC modules with different receptive fields to capture hierarchical information flow. This may improve both predictive performance and interpretability. For example, this study focuses on continuous traffic streams on freeway networks. Urban road networks may have more bottlenecks because of traffic lights so the traffic flows are less continuous. Spatial correlations in these scenarios need further studies.

Second, the DGC in this paper is myopic by design. It considers traffic dynamics from a localized and short-term view. Some other

factors resulting in emerging congestion patterns or long-term variations, such as sudden demand peaks, traffic accidents, and weather, are not considered. Whereas our model can predict the dynamics of congestion once it has started, it is not suitable for predicting the onset of congestion, which is highly uncertain. This problem restricts the prediction horizon to 30 min or less during busy peak hours. For multi-step forecasting, it is very sensitive to previous predictions and unexpected new bottleneck activation quickly increases prediction errors.

A third interesting path of investigation is to add flows to the inputs. We hypothesize that in that case a DGC-based model may be designed such that it is explainable as a real-time macroscopic traffic flow model with adaptable state variables and parameters. The big difference with classic simulation-based traffic data assimilation methods is that DGC-based approaches may be better able to learn the dynamics from incomplete data, than say an LWR-like model in combination with a sequential Monte Carlo method. On the other hand, we also may be overly optimistic about the level of potential interpretability.

Finally, and this combines all previous points, maybe it is possible to redesign these dynamic graph convolution models such that they are able to learn the dynamics of multiple demand and supply processes, such as the dynamics of traffic state itself, the dynamics of key parameters like capacity, and the dynamics of demand. We believe it is better to develop methods that use the best of both worlds of data driven and simulation-based models. These results can help designing similar frameworks and developing more explainable deep learning based traffic forecasting models for decision-making support.

CRediT authorship contribution statement

Guopeng Li: Conceptualization, Methodology, Software, Validation, Writing - original draft. **Victor L. Knoop:** Methodology, Formal analysis, Supervision, Writing - review & editing. **Hans van Lint:** Conceptualization, Resources, Supervision, Writing - review & editing, Project administration, Funding acquisition.

Acknowledgements

This research is sponsored by by the NWO/TTW project MiRRORS with grant agreement number 16270. We thank them for supporting this study.

References

Ahmed, M.S., Cook, A.R., 1979. Analysis of freeway traffic time-series data by using Box-Jenkins techniques. 722.
Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N., 2015. Scheduled sampling for sequence prediction with recurrent neural networks. Adv. Neural Informat. Process. Syst. 1171–1179.
Bhaskar, A., Tsubota, T., Chung, E., et al., 2014. Urban traffic state estimation: Fusing point and zone based data. Transport. Res. Part C: Emerg. Technol. 48, 120–142.
Brabandere, B.D., Jia, X., Tuytelaars, T., Gool, L.V., 2016. Dynamic filter networks. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, pp. 667–675.
Castillo, E., Rivas, A., Jiménez, P., Menéndez, J.M., 2012. Observability in traffic networks. plate scanning added by counting information. Transportation 39, 1301–1333.
Castro-Neto, M., Jeong, Y.S., Jeong, M.K., Han, L.D., 2009. Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Syst. Appl. 36, 6164–6173.
Cui, Z., Henrickson, K., Ke, R., Wang, Y., 2019. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. In: IEEE Trans. Intell. Transp. Syst..
Davis, G.A., Nihan, N.L., 1991. Nonparametric regression and short-term freeway traffic forecasting. J. Transport. Eng. 117, 178–188.
Denoeux, T., 1995. A k-nearest neighbor classification rule based on dempster-shafer theory. IEEE Trans. Syst., Man, Cybernet. 25, 804–813.
Do, L.N., Vu, H.L., Vo, B.Q., Liu, Z., Phung, D., 2019. An effective spatial-temporal attention based neural network for traffic flow prediction. Transport. Res. Part C: Emerg. Technol. 108, 12–28.
Ermagun, A., Levinson, D., 2018. Spatiotemporal traffic forecasting: review and proposed directions. Transport Rev. 38, 786–814.
Fusco, G., Colombaroni, C., Isaenko, N., 2016. Short-term speed predictions exploiting big data on large urban road networks. Transport. Res. Part C: Emerg. Technol. 73, 183–201.
Hamilton, W.L., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs. Adv. Neural Informat. Process. Syst. 1024–1034.
Hamner, B., 2010. Predicting travel times with context-dependent random forests by modeling local and aggregate traffic flow. In: 2010 IEEE International Conference on Data Mining Workshops, IEEE, pp. 1357–1359.
Huang, W., Song, G., Hong, H., Xie, K., 2014. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. 15, 2191–2201.
Kamarianakis, Y., Prastacos, P., 2005. Space-time modeling of traffic flow. Comput. Geosci. 31, 119–133.
Kingma, D.P., Ba, J., 2017. Adam: A method for stochastic optimization doi:arXiv:1412.6980v9, arXiv:1412.6980.
Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks. In: ICLR 2017: International Conference on Learning Representations 2017.
Knoop, V.L., Van Lint, H., Hoogendoorn, S.P., 2015. Traffic dynamics: Its impact on the macroscopic fundamental diagram. Physica A 438, 236–250.
Krishnakumari, P., van Lint, H., Djukic, T., Cats, O., 2019. A data driven method for od matrix estimation. Transport. Res. Part C: Emerg. Technol..
Kumar, S., Asger, M., 2018. A study of state-of-the-art neural machine translation approaches. Int. J. Sci. Res. Comput. Sci., Eng. Informat. Technol. 4, 135–139.
Lana, I., Del Ser, J., Velez, M., Vlahogianni, E.I., 2018. Road traffic forecasting: Recent advances and new challenges. IEEE Intell. Transp. Syst. Mag. 10, 93–109.
Li, Y., Yu, R., Shahabi, C., Liu, Y., 2018. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In: ICLR 2018: International Conference on Learning Representations 2018.
Lighthill, M., Whitham, G., 1955. On kinematic waves ii: A theory of traffic flow on long crowded roads. Proc. R. Soc A 229, 317–345.
van Lint, H., Miete, O., Taale, H., Hoogendoorn, S., 2012. Systematic framework for assessing traffic measures and policies on reliability of traffic operations and travel time. Transport. Res. Rec. 2302, 92–101.
van Lint, J.W.C., Hoogendoorn, S.P., 2010. A robust and efficient method for fusing heterogeneous data from traffic sensors on freeways. Comput.-Aided Civil Infrastruct. Eng. 25, 596–612. https://doi.org/10.1111/j.1467-8667.2009.00617.x.
Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., Wang, Y., 2017. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors 17, 818.

Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transport. Res. Part C: Emerg. Technol. 54, 187–197.
Min, W., Wynter, L., 2011. Real-time road traffic prediction with spatio-temporal correlations. Transport. Res. Part C: Emerg. Technol. 19, 606–616.
Nantes, A., Ngoduy, D., Bhaskar, A., Miska, M., Chung, E., 2016. Real-time traffic state estimation in urban corridors from heterogeneous data. Transport. Res. Part C: Emerg. Technol. 66, 99–118.
Polson, N.G., Sokolov, V.O., 2017. Deep learning for short-term traffic flow prediction. Transport. Res. Part C: Emerg. Technol. 79, 1–17.
Smith, B.L., Demetsky, M.J., 1997. Traffic flow forecasting: comparison of modeling approaches. J. Transport. Eng. 123, 261–266.
Treiber, M., Helbing, D., 2002. Reconstructing the spatio-temporal traffic dynamics from stationary detector data. Cooper@ tive Tr@ nsport@ tion Dyn@ mics 1, 3–1.
Van Hinsbergen, C.P., Schreiter, T., Zuurbier, F.S., Van Lint, J., Van Zuylen, H.J., 2011. Localized extended kalman filter for scalable real-time traffic state estimation. IEEE Trans. Intell. Transport. Syst. 13, 385–394.
Van Lint, J., Hoogendoorn, S., van Zuylen, H.J., 2005. Accurate freeway travel time prediction with state-space neural networks under missing data. Transport. Res. Part C: Emerg. Technol. 13, 347–369.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Informat. Process. Syst. 5998–6008.
Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2018. Graph attention networks. In: ICLR 2018: International Conference on Learning Representations 2018.
Viti, F., Rinaldi, M., Corman, F., Tampère, C.M., 2014. Assessing partial observability in network sensor location problems. Transport. Res. Part B: Methodol. 70, 65–89.
Wang, Y., Papageorgiou, M., Messmer, A., 2005. A real-time freeway network traffic surveillance tool. IEEE Trans. Control Syst. Technol. 14, 18–32.
Wang, Y., Papageorgiou, M., Messmer, A., 2008. Real-time freeway traffic state estimation based on extended kalman filter: Adaptive capabilities and real data testing. Transport. Res. Part A: Policy Practice 42, 1340–1358.
Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C., 2019. Graph wavenet for deep spatial-temporal graph modeling. In: International Joint Conference on Artificial Intelligence 2019, Association for the Advancement of Artificial Intelligence (AAAI), pp. 1907–1913.
Xie, X., van Lint, H., Verbraeck, A., 2018. A generic data assimilation framework for vehicle trajectory reconstruction on signalized urban arterials using particle filters. Transport. Res. Part C: Emerg. Technol. 92, 364–391. https://doi.org/10.1016/j.trc.2018.05.009.
Yu, B., Lee, Y., Sohn, K., 2020. Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (gcn). Transport. Res. Part C: Emerg. Technol. 114, 189–204.
Yu, B., Yin, H., Zhu, Z., 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 3634–3640.
Zhang, J., Shi, X., Xie, J., Ma, H., King, I., yan Yeung, D., 2018. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. In: UAI 2018: The Conference on Uncertainty in Artificial Intelligence (UAI), pp. 339–349.
Zhang, Z., Li, M., Lin, X., Wang, Y., He, F., 2019. Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. Transport. Res. Part C: Emerg. Technol. 105, 297–322.
Zhu, X., Cheng, D., Zhang, Z., Lin, S., Dai, J., 2019. An empirical study of spatial attention mechanisms in deep networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 6688–6697.