IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Dual Graph for Traffic Forecasting

LONG WEI*[1], ZHENGXU YU*[1], ZHONGMING JIN[2], LIANG XIE[1], JIANQIANG HUANG[2], DENG CAI[1], XIAOFEI HE[1](Senior Member, IEEE), XIAN-SHENG HUA[2](Fellow, IEEE)

[1]State Key Laboratory of CAD&CG, Zhejiang University, No.388 Yuhangtang Road, Hangzhou 310058, China
[2]Alibaba Damo Academy, Alibaba Group, Hangzhou, China

Corresponding author: Deng Cai (e-mail: dengcai@cad.zju.edu.cn).

"*" means equal contribution

ABSTRACT Traffic forecasting is the task of predicting future traffic based on historical traffic data. It is challenging due to the complex spatial-temporal correlation on road networks. Most existing research works use sequential Graph Neural Networks (GNN) to model traffic inference. However, they only focus on nodes (intersections) or edges (road segments) traffic forecasting alone. As a result, they could hardly provide a complete description of future traffic on road networks. Actually, nodes and edges traffic are interrelated. Both of them are important for traffic safety and efficiency, and neither one is negligible. In this paper, we exploit nodes and edges information together and make traffic forecasting on nodes and edges simultaneously. We propose a novel dual graph framework, called DualGraph, to model the propagation behavior of traffic on road networks. Inside our framework, we develop a DualMap block to simulate the recursive interactions between nodes and edges. The interaction process is realized by a message passing mechanism of nearby information flow. We employ the Simulation of Urban MObility (SUMO) software to generate real-world traffic data to illustrate the effectiveness of our method. We also empirically evaluate our model on public traffic datasets. The results show that even for node or edge traffic forecasting alone, our model still outperforms compared ones, especially for long term (one hour) prediction.

INDEX TERMS Graph Neural Networks, Traffic Forecasting, Time Series Regression

I. INTRODUCTION

THE traffic forecasting task is to predict future volume, speed, et al. from historical traffic data on road networks. It plays a central role in the intelligent transportation systems (ITS) and has wide applications in traffic control, path planning, and navigation. The challenge of traffic forecasting is mainly caused by complex spatial-temporal dependency under a dynamic traffic environment.

Traffic forecasting is a typical structured regression problem. It has been studied for decades. Early researchers mainly take model-driven approaches [1], which are simple in form but have difficulty fitting traffic dynamics. As an alternative, data-driven approaches are more flexible and have received more studies. Among them, statistic models [2] in time series analysis, such as autoregressive integrated moving average (ARIMA) [3], are dominant in 1990s. From then on, machine learning models become primary choices for traffic forecasting. However, traditional shallow models have low model capacity and lack the ability to capture highly nonlinear spatial-temporal correlation on road networks.

Since road networks have natural graph structures, by viewing intersections as nodes and road segments between intersections as edges, traffic forecasting could be formulated as an inference task on a time series of graphs. With fast development of deep learning, popular deep neural networks have been applied to the traffic forecasting task, such as SAE [4], LSTM [5] and CNN [6]. Particularly, along with the development of graph neural networks (GNN), the recent traffic forecasting paradigm is to incorporate GNN in a sequential learning framework. As a representative of GNN, graph convolutional network (GCN) [7] has been actively used in RNN [8] or Spatio-temporal CNN networks [9] for traffic forecasting. Typically, they adopt an affinity matrix of nodes to represent traffic conditions (e.g., travel time or distance between nodes). Then the affinity matrix is used to construct GCN and characterize propagation among nearby nodes. These models show appealing performance in capturing traffic dynamics overtime on road networks.

However, they only consider forecasting on node traffic of road networks alone. They neglect edge information and fail to make predictions on edge traffic. Actually, sensors are usually located at both roadsides and intersections of modern city road networks. These sensors work together to monitor vehicle activities and provide a real-time traffic description.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2958380, IEEE Access

Neither aspect of traffic data, edges or nodes, should be ignored. Therefore, the prediction results of previous methods are inadequate for future traffic safety and efficiency. Besides, since traffic on nodes and edges are complementary and interrelated, even for the single task of node traffic forecasting, the absence of utilizing information on edges makes the prediction results less accurate.

In this paper, we broaden the scope of the traffic forecasting task. Different from previous approaches, we take both node and edge information into consideration and attempt to make predictions on nodes and edges simultaneously. Instead of studying spatial-temporal dependency only among nodes, we investigate the traffic propagation behavior between nodes and edges of the entire road networks. Consequently, the prediction of our method could provide a complete description of future traffic. Our contributions could be summarized as follows:

(1) We reformulate the traffic forecasting task as taking historical traffic data on both nodes and edges to predict future nodes and edges traffic simultaneously. We propose a novel dual graph network, called DualGraph, for traffic forecasting under this new setting. It is a unified framework that could be trained from end to end. We develop a new DualMap block to learn node features and edge features interactively. The DualMap block takes the message passing mechanism to characterize the spreading behavior of information flow between nodes and edges on road networks. As a special case, the traditional traffic forecasting with only nodes traffic inputs and predictions could still be realized by our DualGraph as a special case.

(2) We use the Simulation of Urban MObility (SUMO) software to simulate real-world traffic data, collected from sensors at nodes and edges of road networks. The effectiveness of the DualGraph is verified via an ablation study. What's more, We evaluate our method on public traffic datasets METR-LA and Pems-Bay that only contain traffic data on edges of highways. Our model achieves state-of-the-art results. Particularly, our method outperforms compared ones by a large margin for a long time (an hour) prediction.

II. RELATED WORK

A. TRAFFIC FORECASTING

Early traffic forecasting methods are based on simulations, control theory, or physical analogies [1]. Then data-driven approaches become popular. They range from statistical models, like ARIMA [3], to neural and evolutionary computational approaches [10]. More methods are referred to the survey paper [2]. Simple linear models like linear Support Vector Regression (SVR) [11] and multi-variable linear regression (MVLR) [12] are also applied to traffic forecasting, but these shallow models are less capable of capturing complex spatial-time correlation on road networks.

The development of deep learning offers an unprecedented opportunity for traffic forecasting. Such neural networks include CNN [6], [13], [14], RNN [15]–[17] and Stacked Auto-Encoders (SAEs) [4]. Regarding the graph structure of road networks, graph neural networks (GNN) facilitates time series prediction on graphs. Recently, Yu et al. propose a Spatio-Temporal Graph Convolutional Network (STGCN) [18]. Wang et al. adjust this approach to dynamic traffic conditions over time [19]. And Li et al. consider the traffic flow as a graph diffusion process and incorporate it in RNN (DCRNN) [8]. Despite their good performance in traffic forecasting, these research works only use node information and make node predictions, with the absence of edge information. Contrarily, our approach aims to make predictions on both nodes and edges to give a complete picture of future traffic.

B. GRAPH NEURAL NETWORKS

Graph Neural Networks (GNN) have received intensive studies in recent years. A comprehensive survey on GNN is given by [20]. Current GNN could be classified into two categories based on their ways of aggregation from neighbors: spectral and non-spectral approaches.

Spectral approaches extend the convolutional operator from image to graph based on spectral theory on graphs, include ChebNet [21] and GCN [7], [22]–[24]. GCN was initially developed to model the spatial dependence of nodes. Seo et al. combines ChebNet with RNN for structured sequence learning [25].

Non-spectral approaches define convolution by making direct aggregation among nearby nodes. Message passing is widely used to make inference on graphs. For example, GraphSAGE [26] generates nodes' embeddings by aggregating features from local neighborhoods. Optional aggregation strategies include self-attention [27] and shortcut connections across layers [9]. Our DualMap is a non-spectral approach. We use message passing to compute aggregation between nodes or edges interactively.

We notice that the notation "dual graph" is also introduced in the scene graph community [28]. We need to clarify the essential difference. Dual graph in scene graph research is used for static relation prediction between objects in an image. They do not make simultaneous node and edge prediction and could not be applied to temporal prediction on graphs.

III. DUAL GRAPH

A. PROBLEM DEFINITION

The traffic forecasting problem is to predict future traffic data (speed, flow volume, travel time, et al.) given a sequence of historical traffic data on a road network. In our setting, traffic data are collected from both nodes and edges. We present notations of this paper in Table 1.

Formally, the road network can be represented as a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V}$ is the set of nodes with numbers $V = |\mathcal{V}|$, $\mathcal{E}$ is the set of edges with numbers $E = |\mathcal{E}|$, and W is the weight matrix of nodes similarity. Different from previous works, we only need the topological graph matrix, i.e., $W_{ij} \in \{0, 1\}$. We denote $\mathcal{E}(v)$ as the set of (both incoming and outgoing) edges connected to a node $v \in \mathcal{V}$; and $\mathcal{V}(e)$ as the set of (both start and end) nodes connected to an $e \in \mathcal{E}$. Assume the feature size

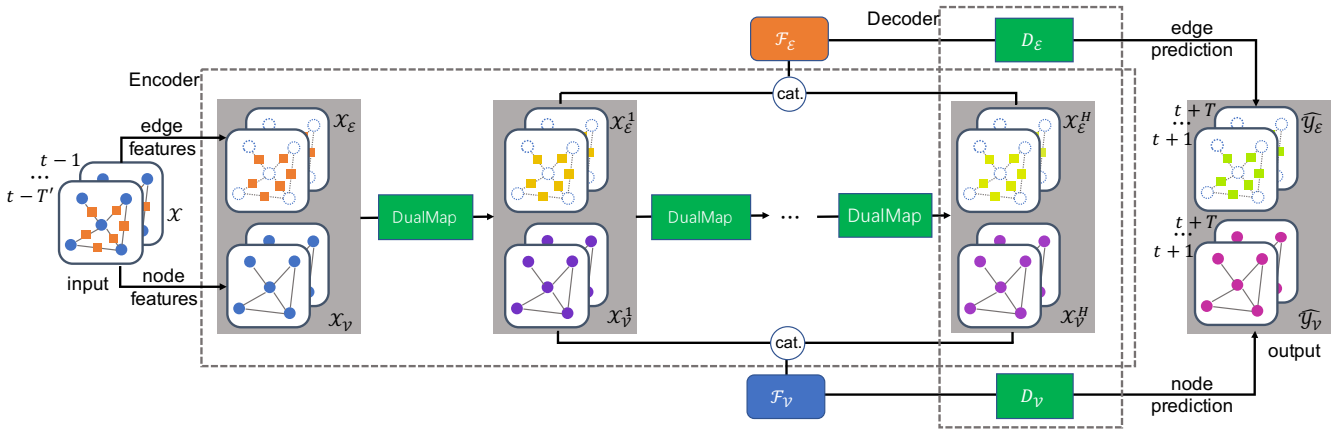


FIGURE 1. Illustration of our DualGraph network. Input is a time series of historical traffic data on graphs, consisting of both node features and edge features. Our network has an Encoder-Decoder structure. Encoder is a multi-layer network of DualMap blocks. All the intermediate hidden feature maps are concatenated as historical embeddings of nodes and edges, respectively. Finally, the Decoder is applied to make future traffic predictions on both nodes and edges. Green blocks are modules of DualGraph.

TABLE 1. Summarization of notations.

$\mathcal{G}$	a weighted directed graph
$W, \mathcal{V}, \mathcal{E}$	weight matrix, set of nodes and set of edges of $\mathcal{G}$
V, E, C	number of nodes, number of edges, feature size
$\mathcal{E}(v)$	set of edges connected to a node $v \in \mathcal{V}$
$\mathcal{V}(e)$	set of nodes connected to an edge $e \in \mathcal{E}$
T', T	number of historical and future time steps
$\mathcal{X}_\mathcal{V}, \mathcal{X}_\mathcal{E}$	historical traffic data on nodes and edges
$\mathcal{Y}_\mathcal{V}, \mathcal{Y}_\mathcal{E}$	ground-truth future traffic data on nodes and edges
$\hat{\mathcal{Y}}_\mathcal{V}, \hat{\mathcal{Y}}_\mathcal{E}$	predicted future traffic data on nodes and edges
$\mathcal{X}_v, \mathcal{X}_e$	historical feature of a node v and an edge e
$\mathcal{F}_\mathcal{V}, \mathcal{F}_\mathcal{E}$	historical embeddings of nodes and edges
H	number of DualMap layers
$\phi(\cdot)$	message passing function from nodes to edges (N2E)
$\psi(\cdot)$	message passing function from edges to nodes (E2N)
$r_\mathcal{V}(\cdot), r_\mathcal{E}(\cdot)$	reader out functions
$D = (D_\mathcal{V}, D_\mathcal{E})$	decoders

is C, the input $\mathcal{X}$ is traffic data on nodes and edges of T' historical time steps. It is a pair of tensors $\mathcal{X}_\mathcal{V} \in \mathbb{R}^{T' \times V \times C}$ and $\mathcal{X}_\mathcal{E} \in \mathbb{R}^{T' \times E \times C}$. The output $\mathcal{Y}$ is traffic data on nodes and edges of T future time steps, denoted by $\mathcal{Y}_\mathcal{V} \in \mathbb{R}^{T \times V \times C}$ and $\mathcal{Y}_\mathcal{E} \in \mathbb{R}^{T \times E \times C}$ respectively. For simplicity, we assume feature sizes are same for nodes and edges. Our method could be extended to scenarios that nodes and edges have different features with no effort. For cases where edge information is unavailable, we take values of $\mathcal{X}_\mathcal{E}$ and $\mathcal{Y}_\mathcal{E}$ as zeros. Thus it is obvious that problem settings of previous traffic forecasting research are special cases of our paper.

B. OVERVIEW OF FRAMEWORK

Our DualGraph framework is shown in Figure 1. It has an Encoder-Decoder architecture. The Encoder comprises multi-layer DualMap blocks. Each DualMap block maps the hidden features of all the nodes and edges to the next layer. Nodes and edges feature are interrelated in a DualMap block. At the end of Encoder, these hidden feature maps are concatenated to produce historical embeddings that represent all the historical information. At last, the Decoder module takes the historical embeddings as input and produces the future traffic prediction.

In our framework, we use Multi-Layer Perceptrons (MLP) as basic units. An MLP unit is a composition of fully connected layers with nonlinear activation functions. Our DualMap block is a stack of MLPs and Decoder is also an MLP. Throughout our framework, the basic unit MLP contains two hidden layers. We will specify the structures of the DualMap block in the following subsection.

C. DUALMAP BLOCK

Our DualMap block is shown in Figure 2. It is designed to model the relationship between nodes and edges. Its input consists of node features $\mathcal{X}_\mathcal{V}$ and edge features $\mathcal{X}_\mathcal{E}$ covering all historical time steps. We incorporate interaction between nodes and edges in a message passing process. We adopt message passing instead of GCN based on two aspects of considerations. First, the affinity matrix of nodes could only be used to model aggregation between nodes, rather than information flow between nodes and edges. Second, due to traffic dynamics, the affinity matrix of nodes could be variant over a long time. Thus GCN requires estimation of real-time affinity matrix, which needs extra computation cost.

In each DualMap block, we assume that the input feature size is C_1 and output feature size is C_2. For each node $v \in \mathcal{V}$, we use $\mathcal{X}_v \in \mathbb{R}^{T' \times C_1}$ to denote historical feature of the node v. Similarly, we use $\mathcal{X}_e \in \mathbb{R}^{T' \times C_1}$ to denote historical feature of an edge e. We introduce two message passing functions $\phi : \mathbb{R}^{T' \times C_1} \to \mathbb{R}^{T' \times C_1}$ and $\psi : \mathbb{R}^{T' \times C_1} \to \mathbb{R}^{T' \times C_1}$ to model spreading actions from nodes to edges (N2E), and edges to nodes (E2N), respectively. Specifically, ϕ transfers node feature $\mathcal{X}_v$ to a node message, which would be passed to connected edges of v. And similarly, ψ transfers edge feature $\mathcal{X}_e$ to an edge message, which would be passed to connected nodes of e. The message passing process models information

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2958380, IEEE Access

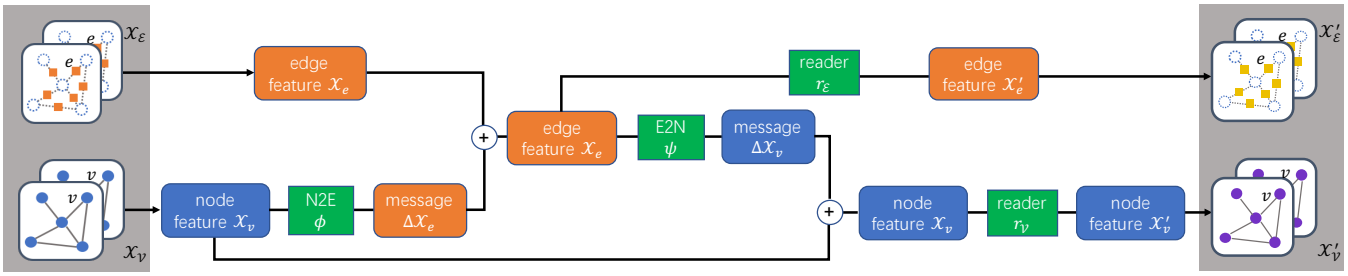


FIGURE 2. Illustration of the DualMap block. Both edge features and node features are taken as input. It computes feature increments by aggregation of messages passed from nearby nodes and edges. Then cross-layer fusion and reader out functions are applied to update edges and nodes features. Green boxes are MLPs.

flow between node and edge features. In order to control the model complexity, all the nodes share the same function ϕ and all the edges share the same function ψ. Both ϕ and ψ are MLPs.

We introduce $AGG_\mathcal{E}(\cdot)$ to represent an aggregation function of the passed message to an edge. It has a form of max or average pooling. Then for each edge $e \in \mathcal{E}$, the increment $\Delta\mathcal{X}_e$ is aggregation of messages from all its neighbor nodes:

$$\Delta\mathcal{X}_e = AGG_\mathcal{E}(\{\phi(\mathcal{X}_v)|v \in \mathcal{V}(e)\}), \tag{1}$$

Here aggregation is made on both start nodes and end nodes of e because both upstream and downstream traffic could affect traffic on an edge. Then hidden feature of each edge is summation of its original value and the incremental value. It could be regarded as a kind of residual learning [29]. Namely, we update the hidden feature by

$$\mathcal{X}_e \leftarrow \mathcal{X}_e + \epsilon_e \Delta\mathcal{X}_e, \tag{2}$$

where ϵ_e is a scalar parameter of the network to be learned. In fact, it is shown that introduction of such parameters could increase the expressive power of the network by inducing injective maps between successive layers [9].

On top of the updated hidden features of edges, we could update hidden features of nodes in a similar way. We generate message from connected edges for each node $v \in \mathcal{V}$ and aggregate them as increment on node v:

$$\Delta\mathcal{X}_v = AGG_\mathcal{V}(\{\psi(\mathcal{X}_e)|e \in \mathcal{E}(v)\}), \tag{3}$$

Here $AGG_\mathcal{V}(\cdot)$ could be chosen with the same form of $AGG_\mathcal{E}(\cdot)$. Similarly, we fuse hidden node features by shortcut connection across layers:

$$\mathcal{X}_v \leftarrow \mathcal{X}_v + \epsilon_v \Delta\mathcal{X}_v, \tag{4}$$

At last, we use two reader out functions to aggregate fused hidden features to obtain representations of edge and node:

$$\mathcal{X}'_v = r_\mathcal{V}(\mathcal{X}_v), \quad \mathcal{X}'_e = r_\mathcal{E}(\mathcal{X}_e). \tag{5}$$

Both $r_\mathcal{V}(\cdot) : \mathbb{R}^{T'\times C_1} \to \mathbb{R}^{T'\times C_2}$ and $r_\mathcal{E}(\cdot) : \mathbb{R}^{T'\times C_1} \to \mathbb{R}^{T'\times C_2}$ are two MLPs.

In summary, one DualMap block could map an input pair $(\mathcal{X}_\mathcal{V}, \mathcal{X}_\mathcal{E})$ to an output pair $(\mathcal{X}'_\mathcal{V}, \mathcal{X}'_\mathcal{E})$. We denote the DualMap block as a function $DualMap(\cdot)$, then

$$(\mathcal{X}'_\mathcal{V}, \mathcal{X}'_\mathcal{E}) = DualMap(\mathcal{X}_\mathcal{V}, \mathcal{X}_\mathcal{E}). \tag{6}$$

It is easy to see that in each DualMap block, feature of each node and edge is impacted by its neighboring edge and node, respectively. Compared with traditional approaches that only consider propagation among nodes, the spread of traffic flow is formulated in a finer way inside the DualMap block because the connecting edge between a pair of nodes are introduced. Therefore, to make graph inference over a broad receptive field, we need repetitively apply the DualMap block.

D. DUALGRAPH NETWORK

We stack H DualMap blocks to build an Encoder module of our DualGraph Network. The transformation in the h-th DualMap layer could be written as

$$(\mathcal{X}^h_\mathcal{V}, \mathcal{X}^h_\mathcal{E}) = DualMap(\mathcal{X}^{h-1}_\mathcal{V}, \mathcal{X}^{h-1}_\mathcal{E}), h = 1, \cdots, H, \tag{7}$$

where $\mathcal{X}^h_\mathcal{V} \in \mathbb{R}^{T'\times V\times C_h}$ and $\mathcal{X}^h_\mathcal{E} \in \mathbb{R}^{T'\times E\times C_h}$ are hidden features of nodes and edges in the h-th layer. Particularly, $\mathcal{X}^0_\mathcal{V} = \mathcal{X}_\mathcal{V}$ and $\mathcal{X}^0_\mathcal{E} = \mathcal{X}_\mathcal{E}$ are input historical traffic data. In a $H-$layer DualMap network, output feature of each node (resp. edge) is an aggregation of its $\lfloor H/2 \rfloor$-degree neighboring nodes (resp. edges) and $\lfloor (H+1)/2 \rfloor$-degree neighboring edges (resp. nodes) covering all historical time steps. After we get outputs of the final layer $\mathcal{X}^H_\mathcal{E}$ and $\mathcal{X}^H_\mathcal{V}$, we could generate the entire historical embeddings $\mathcal{F}_\mathcal{V}$ and $\mathcal{F}_\mathcal{E}$ by concatenation of hidden features of all layers:

$$\mathcal{F}_\mathcal{V} = [\mathcal{X}^1_\mathcal{V}; \cdots; \mathcal{X}^H_\mathcal{V}], \quad \mathcal{F}_\mathcal{E} = [\mathcal{X}^1_\mathcal{E}; \cdots; \mathcal{X}^H_\mathcal{E}]. \tag{8}$$

Here, concatenation is made along the feature dimension.

Finally, we use the Decoder $D = (D_\mathcal{V}, D_\mathcal{E})$ to make predictions of future traffic data:

$$\hat{\mathcal{Y}}_\mathcal{V} = D_\mathcal{V}(\mathcal{F}_\mathcal{V}), \quad \hat{\mathcal{Y}}_\mathcal{E} = D_\mathcal{E}(\mathcal{F}_\mathcal{E}), \tag{9}$$

where $D_\mathcal{V}(\cdot) : \mathbb{R}^{T'\times V\times(\Sigma_{h=1}^H C_h)} \to \mathbb{R}^{T\times V\times C}$ and $D_\mathcal{E}(\cdot) : \mathbb{R}^{T'\times E\times(\Sigma_{h=1}^H C_h)} \to \mathbb{R}^{T\times E\times C}$ are two MLPs.

The loss function is Mean Absolute Error (MAE) between future traffic prediction and ground-truth values:

$$L(\hat{\mathcal{Y}}_\mathcal{V}, \hat{\mathcal{Y}}_\mathcal{E}; \mathcal{Y}_\mathcal{V}, \mathcal{Y}_\mathcal{E}) = \|\hat{\mathcal{Y}}_\mathcal{V} - \mathcal{Y}_\mathcal{V}\|_1 + \|\hat{\mathcal{Y}}_\mathcal{E} - \mathcal{Y}_\mathcal{E}\|_1, \tag{10}$$

where $\|\cdot\|_1$ is ℓ_1 norm of tensors. If edge data is unavailable, we drop the error term on edges from loss function and only make predictions on nodes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2958380, IEEE Access

TABLE 2. Summarization of datasets. *#nodes* is number of nodes; C is feature size on each node and edge; *edge info.* means whether edge information is contained in the dataset; *data split* means division of training, validation and testing subsets.

Datasets	#nodes	data type	C	edge info.	data split
Synth-SUMO	21	volume	1	Yes	(8 days, 2 days, 4 days)
METR-LA	207	speed	1	No	(70%, 10%, 20%)
PeMSD7	228	speed	1	No	(34 days, 5 days, 5 days)

E. RELATION TO PREVIOUS RESEARCH

We compare DualGraph with three recent traffic forecasting methods DCRNN [8], STGCN [18] and DST-GCNN [19] as follows:

- **Problem setting**. DualGraph could make predictions on both edges and nodes, whereas DCRNN, STGCN, and DST-GCNN predict traffic only on nodes. We will see that DualGraph could make a more accurate prediction on nodes, even regardless of its additional prediction on edges.
- **Inference strategy**. STGCN and DST-GCNN could only predict traffic of one future time step in one pass. To make multiple time step predictions, STGCN needs multiple passes of the model and DCT-GCNN needs to train multiple models. DCRNN uses RNN to make a sequential prediction, incurring a risk of error propagation along time in the inference phase. DualGraph could make a multi-instance prediction simultaneously. As we will show in the following experiments, our DualGraph could alleviate error accumulation for long term prediction.
- **GNN structure**. In each GNN unit, we use message passing to replace GCN that is used in DCRNN, STGCN, and DST-GCNN. Because our model involves both node and edge prediction, GCN is no longer feasible in our model.

IV. EXPERIMENTS

We first verify the effectiveness of the DualGraph on synthetic data by SUMO simulation. Then we evaluate on public traffic datasets. All the experiments are repeated three times and the average results are reported. DualGraph are implemented by the PyTorch 0.4 framework.

A. DATASETS

We conduct experiments on a synthetic dataset and two public datasets. They are summarized in Table 2.

TABLE 3. Structure of modules in a DualMap Block.

modules		structures
Encoder	ϕ, ψ	MLP(C_hT', 64, C_hT')
	$r_{\mathcal{V}}, r_{\mathcal{E}}$	MLP(C_hT', 64, $C_{h+1}T'$)
Decoder	$D_{\mathcal{V}}$	MLP($CT'V$, 256, CTV)
	$D_{\mathcal{E}}$	MLP($CT'E$, 256, CTE)

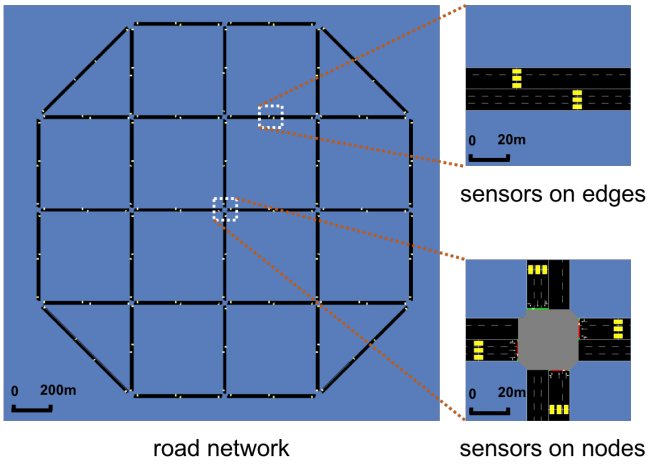


FIGURE 3. Road network by SUMO. There are 21 nodes with 72 directed edges. Each node has three or four directions. We simulate real world traffic, which contains tidal lanes, peak and off-peak flow periods.

Synth-SUMO is generated by the SUMO [1] platform. SUMO is a widely used tool for road network design and traffic volume simulation. Our data generation follows [30], simulating real-world traffic according to the given policy of traffic lights. Specifically, we construct a road network including 21 intersections (nodes) and 72 directed road segments (edges). Each intersection has three or four directions, as shown in Figure 3. Each road segment and intersection has three lanes. Arriving vehicles are generated by Poisson distribution with certain arrival rates. We generate traffic of 14 days. Training and validation data are from weekdays and testing data is from weekends. Traffic volume data is aggregated every 5 minutes. Traffic volume from all directions at a node is summed up as its input feature, thus feature size C=1 for both nodes and edges.

METR-LA contains traffic data collected from loop detectors in the highway of the Los Angeles County road network [31]. Following [8] and [19], we select 207 sensors on the road network and collect 4 months of vehicle speed data ranging from Mar 1st, 2012 to Jun 30th, 2012 for our experiments.

PeMSD7 contains traffic data across major metropolitan areas of California state highway system [32]. Following [18], we randomly select 228 stations from District 7 of California. The time range of the PeMSD7 dataset is weekdays of May and June of 2012.

[1] http://sumo.dlr.de/index.html

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2958380, IEEE Access

TABLE 4. Ablation study of DualGraph on the Synth-SUMO dataset with (✓) or without (✗) edge information. Under the condition with edge information, we report the results of traffic volume prediction on both nodes and edges. Otherwise, we only report the results of the prediction on nodes. A comparison is made under different network depths. Here for simplicity, we only present results of the MAE metric.

T	$H=1$			$H=2$			$H=3$		
	✓		✗	✓		✗	✓		✗
	node pred.	edge pred.	node pred.	node pred.	edge pred.	node pred.	node pred.	edge pred.	node pred.
15 min	6.66	7.24	6.96	6.48	6.81	6.99	6.61	6.87	6.93
30 min	7.39	7.96	7.81	7.29	7.54	7.83	7.26	7.51	7.79
60 min	9.35	9.82	9.69	9.34	9.46	9.67	9.23	9.45	9.59

In both public datasets, traffic speed data is aggregated every 5 minutes with one direction (feature size C=1). All data are viewed as speed only on nodes of traffic networks. Thus both input and output on edges of DualGraph are zeros. We apply a Z-Score normalization.

B. EXPERIMENT SETTING

TABLE 5. Comparison results with existing methods on the Synth-SUMO dataset for node traffic volume prediction only. MAE, RMSE and MAPE (%) metrics are compared for different future time steps. Bold numbers indicate the best results.

T	Metric	DCRNN	STGCN	DualGraph
15 min	MAE	14.31	11.86	**6.93**
	RMSE	22.19	19.91	**15.9**
	MAPE	13.3	20.2	**6.4**
30 min	MAE	16.04	12.14	**7.79**
	RMSE	25.84	19.44	**18.5**
	MAPE	14.7	21.0	**7.0**
60 min	MAE	19.84	14.16	**9.59**
	RMSE	32.19	**21.85**	23.0
	MAPE	17.6	24.6	**8.3**

Following [8], [18], we use three evaluation metrics: Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE). Low values mean better performance.

Different from previous approaches, we only need the topological graph for message passing. To construct the weight matrix of nodes W, we set $W_{ij} = 1$ if i and j are connected, and otherwise $W_{ij} = 0$.

Baselines. We mainly compare DualMap with three recent GCN-based methods: (1) DCRNN [8]; (2) STGCN [18]; (3) DST-GCNN [19]. The codes of DCRNN [2] and STGCN [3] are public, while codes of DST-GCNN are not available. We also include the followings as weak baselines on public dataset: (4) HA: historical Average; (5) ARIMA: Auto-Regressive Integrated Moving Average model with Kalman filter; (6) FNN: Feedforward neural network with two hidden layers and L2 regularization; (7) FC-LSTM: LSTM with fully connected hidden units [16].

Implementation. For an MLP with input dimension d_{in}, hidden dimension d_{hidden} and output dimension d_{out}, we represent it as MLP($d_{in}, d_{hidden}, d_{out}$), which is a composition of three fully connection layers $f_1 : \mathbb{R}^{d_{in}} \rightarrow \mathbb{R}^{d_{hidden}}$, $f_2 : \mathbb{R}^{d_{hidden}} \rightarrow \mathbb{R}^{2d_{hidden}}$, and $f_3 : \mathbb{R}^{2d_{hidden}} \rightarrow \mathbb{R}^{d_{out}}$. Dropout rate 0.5 is used in f_3. We adopt tanh$(\cdot)$ as the activation function between two successive layers. We list configuration of modules in our DualGraph in Table 3. Default number of layers is $H = 2$. Hidden features size are set to $2C$. On all the three datasets, we use traffic data of the past one hour ($T' = 12$) as model input. Then we predict traffic of future 15 min, 30 min, 60 min on Synth-SUMO and METR-LA; and 15 min, 30 min, 45 min on PeMSD7. We use SGD as an optimizer. The momentum in SGD is 0.9 and weight decay is 5e-4. The minibatch size is 64 and all the models run 100 epochs. In SGD, the initial learning rate is 0.001. It is lowered by a factor of 0.1 after the 50-th epoch.

TABLE 6. Performance of DualGraph on METR-LA under different depths for node prediction only. We evaluate its performance with (✓) or without (✗) edge reseting to zeros. Here for simplicity, we only present results of MAE metric.

T	H=2		H=3	
	✓	✗	✓	✗
15 min	2.60	2.64	2.63	2.67
30 min	2.83	2.85	2.83	2.86
60 min	3.13	3.15	3.14	3.16

C. EVALUATION ON SYNTH-SUMO

First, we verify the effectiveness of the DualGraph on Synth-SUMO. We evaluate DualGraph under two settings: with or without edge information and prediction. Here "with" means that input to DualGraph contains both historical edge and node data and the output covers both traffic predictions on nodes and edges. On the contrary"without" means that input only contains traffic data on nodes and data on edge are set to zeros, then we only make predictions on nodes. Evaluation is made under different depths H of the Encoder. The results are shown in Table 4.

As we can see, DualGraph achieves balanced prediction accuracy on both nodes and edges. The results vary smoothly in terms of model depth. It is appealing that simultaneous prediction on both nodes and edges also improves the accuracy of node prediction. This implies that nodes and edges are interrelated and utilization of edge information is necessary for accurate and complete traffic prediction.

Then we compare DualMap with different methods on Synth-SUMO for node traffic forecasting. The results are

[2]https://github.com/liyaguang/DCRNN

[3]https://github.com/VeritasYin/STGCN_IJCAI-18

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2958380, IEEE Access

FIGURE 4. Simultaneous node (above) and edge (below) traffic volume prediction during a test day on the Sythn-SUMO dataset. The node and edge refer to those enlarged in Figure 3. The green line means the ground truth real time traffic volume. Best viewed in color.

TABLE 7. Comparison results of node traffic speed prediction with existing methods on the METR-LA and PeMSD7 datasets. MAE, RMSE and MAPE (%) metrics are compared for different future time steps. Bold numbers indicate the best results.

Datasets	T	Metric	HA	ARIMA	FNN	FC-LSTM	DCRNN	STGCN	DST-GCNN	DualGraph
METR-LA	15 min	MAE	4.16	3.99	3.99	3.44	2.77	2.87	2.68	**2.62**
		RMSE	7.80	8.21	7.94	6.30	5.38	5.54	**5.35**	5.36
		MAPE	13.0	9.6	9.9	9.6	7.3	7.4	7.2	**7.1**
	30 min	MAE	4.16	5.15	4.23	3.77	3.15	3.48	3.01	**2.83**
		RMSE	7.80	10.45	8.17	7.23	6.45	6.84	6.23	**5.89**
		MAPE	13.0	12.7	12.9	10.9	8.8	9.4	8.5	**7.9**
	60 min	MAE	4.16	6.90	4.49	4.37	3.60	4.45	3.41	**3.15**
		RMSE	7.80	13.23	8.69	8.69	7.59	8.41	7.47	**6.67**
		MAPE	13.0	17.4	14.0	13.2	10.5	11.8	10.3	**9.1**
PeMSD7	15 min	MAE	4.01	5.55	2.74	3.57	2.37	**2.25**	-	2.35
		RMSE	7.20	9.00	4.75	6.20	4.21	**4.04**	-	4.53
		MAPE	10.61	12.92	6.38	8.60	5.54	**5.26**	-	5.48
	30 min	MAE	4.01	5.86	4.02	3.94	3.31	3.03	-	**2.89**
		RMSE	7.20	9.13	6.98	7.03	5.96	5.70	-	**5.42**
		MAPE	10.61	13.94	9.72	9.55	8.06	7.33	-	**7.21**
	45 min	MAE	4.01	6.27	5.04	4.16	4.01	3.57	-	**3.35**
		RMSE	7.20	9.38	8.58	7.51	7.13	6.77	-	**6.76**
		MAPE	10.61	15.20	12.38	10.10	9.99	8.69	-	**8.28**

shown in Table 5, where MAE results of DualGraph corresponds to the rightmost column under $H = 3$ in Table 4. We carefully tune the hyper-parameters (including learning rate, number of hidden units, et al.) in both DCRNN and STGCN to make them perform better than their default settings. Specifically, the number of RNN layers is 1 and the number of hidden units is 32 in DCRNN; the initial learning rate is 1e-4 and sigma in weight matrix is 500 in STGCN. Still, we observe that our DualGraph achieves significantly better results. The reason is that we take both edge and node information into consideration and DualMap blocks make effective simulate of interactions between them.

We illustrate the performance of simultaneous traffic volume prediction on an edge and a node of a test day in Figure 4. We could make the following observations: (1) traffic data generated by SUMO could well simulate common traffic trends, such as peak and off-peak flow periods; (2) DualGraph performs better to capture fine-grained change of the traffic trend in node prediction, compared to DCRNN and STGCN. (3) edge prediction of DualGraph also keeps good paces with ground-truth traffic.

D. EVALUATION ON METR-LA AND PEMSD7

These two public datasets only have traffic data on nodes while edge information is unavailable. So evaluation is only made on nodes. We first evaluate the effectiveness of the DualMap block on METR-LA. The results are presented in Table 6. Since METR-LA has no edge data, we test performance under two cases: set output edge features of intermediate DualMaps to zeros or not in the DualGraph (from the second DualMap layer). DualGraph shows slightly better results without setting edge output to zeros. It implies that the DualMap block could transmit useful information to the next layer even under the absence of edge information.

Comparison with State-of-the-arts Methods. We compare with different methods on METR-LA and PeMSD7 for node traffic prediction. The results are shown in Table 7. Results of all compared methods are copied from referred papers. We can see that DualGraph achieves the best prediction accuracy on these two datasets under most cases and metrics, except for 15 min prediction on PeMSD7. The results demonstrate that even for the single node prediction task, our DualGraph remains a competitive method. It is worth noting that the advantage of DualGraph becomes more obvious for long term (45 min or 60 min) prediction. This evidence shows that our framework could alleviate error propagation in the inference stage.

V. CONCLUSION

In this paper, we study the traffic forecasting problem on road networks. We extend previous single task prediction of node traffic to simultaneous node and edge prediction. In this way, we could obtain a complete landscape of future traffic. We propose a novel DualGraph framework to model the propagation behavior of traffic over road networks. As a basic block of DualGraph, DualMap is designed to learn interactions between nodes and edges features. We employ SUMO to simulate real-world traffic data and demonstrate the effectiveness of DualGraph for simultaneous node and edge traffic prediction. We also conduct experiments on public traffic datasets. The results reveal the advantage of DualGraph over existing methods on the single node prediction task. For future work, DualGraph could be extended to general spatial-temporal graph prediction tasks.

VI. ACKNOWLEDGMENT

This work was supported in part by The National Key Research and Development Program of China (Grant Nos: 2018AAA0101400), in part by The National Nature Science Foundation of China (Grant Nos: 61936006), in part by the Alibaba-Zhejiang University Joint Institute of Frontier Technologies.

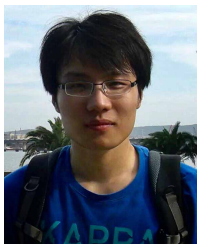
REFERENCES

[1] D. R. Drew, "Traffic flow theory and control," Tech. Rep., 1968.
[2] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," Transportation Research Part C: Emerging Technologies, vol. 43, pp. 3–19, 2014.
[3] G. A. Davis, N. L. Nihan, M. M. Hamed, and L. N. Jacobson, "Adaptive forecasting of freeway traffic congestion," Transportation Research Record, no. 1287, 1990.
[4] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 865–873, 2014.
[5] X. Dai, R. Fu, Y. Lin, L. Li, and F.-Y. Wang, "Deeptrend: A deep hierarchical neural network for traffic flow prediction," arXiv preprint arXiv:1707.03213, 2017.
[6] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[7] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in International Conference on Learning Representations (ICLR), 2017.
[8] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," arXiv preprint arXiv:1707.01926, 2017.
[9] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" arXiv preprint arXiv:1810.00826, 2018.
[10] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights," Transportation Research Part C: Emerging Technologies, vol. 19, no. 3, pp. 387–399, 2011.
[11] X. Jin, Y. Zhang, and D. Yao, "Simultaneously prediction of network traffic flow based on pca-svr," in International Symposium on Neural Networks. Springer, 2007, pp. 1022–1031.
[12] L. Li, X. Su, Y. Zhang, Y. Lin, and Z. Li, "Trend modeling for traffic time series analysis: An integrated study," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 6, pp. 3430–3439, 2015.
[13] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," Sensors, vol. 17, no. 4, p. 818, 2017.
[14] Z. Tan and R. Li, "A dynamic model for traffic flow prediction using improved drn," arXiv preprint arXiv:1805.00868, 2018.
[15] Z. Cui, R. Ke, Y. Wang et al., "Deep stacked bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction," in 6th International Workshop on Urban Computing (UrbComp 2017), 2016.
[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in neural information processing systems, 2014, pp. 3104–3112.
[17] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in 2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity). IEEE, 2015, pp. 153–158.

[18] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," arXiv preprint arXiv:1709.04875, 2017.
[19] M. Wang, B. Lai, Z. Jin, X. Gong, J. Huang, and X. Hua, "Dynamic spatio-temporal graph-based cnns for traffic prediction," arXiv preprint arXiv:1812.02019, 2018.
[20] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," arXiv preprint arXiv:1901.00596, 2019.
[21] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in Advances in neural information processing systems, 2016, pp. 3844–3852.
[22] J. Chen, T. Ma, and C. Xiao, "Fastgcn: fast learning with graph convolutional networks via importance sampling," arXiv preprint arXiv:1801.10247, 2018.
[23] W. Huang, T. Zhang, Y. Rong, and J. Huang, "Adaptive sampling towards fast graph representation learning," in Advances in Neural Information Processing Systems, 2018, pp. 4558–4567.
[24] F. Wu, T. Zhang, A. H. d. Souza Jr, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," arXiv preprint arXiv:1902.07153, 2019.
[25] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in International Conference on Neural Information Processing. Springer, 2018, pp. 362–373.
[26] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Advances in Neural Information Processing Systems, 2017, pp. 1024–1034.
[27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," arXiv preprint arXiv:1710.10903, 2017.
[28] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5410–5419.
[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[30] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018, pp. 2496–2505.
[31] H. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, "Big data and its technical challenges," Communications of the ACM, vol. 57, no. 7, pp. 86–94, 2014.
[32] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," Transportation Research Record, vol. 1748, no. 1, pp. 96–102, 2001.

LONG WEI received the B.S. degree in Math and Applied Mathematics from Zhejiang University, Hangzhou, China, in 2013. He is currently a Ph.D. candidate in computer science at Zhejiang University. His research interests include machine learning and computer vision.

ZHENGXU YU is now a doctoral candidate at Zhejiang University, and he is currently a research intern at Alibaba DAMO Academy, Hangzhou, China. His research interests include large scale machine learning and computer vision.

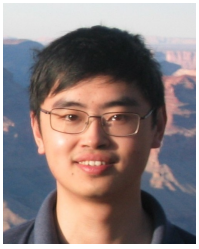


ZHONGMING JIN is now a staff algorithm engineer at Alibaba DAMO Academy. Previously, he was a researcher at Baidu Research, Beijing, China. He received his Ph.D. degree from Zhejiang University in Mar. 2015. His research interests include large scale machine learning and computer vision.

LIANG XIE is now a PhD. candidate Zhejiang University. He graduated from Huazhong University of Science and Technology, Wuhan, China, in 2018. His research interests include 3D vision and deep learning.

JIANQIANG HUANG is a director of Alibaba DAMO Academy. He received the second prize of the National Science and Technology Progress Award in 2010. His research interests focus on visual intelligence in the city brain project of Alibaba, Hangzhou, China.

DENG CAI is a Professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. He received the Ph.D. degree in computer science from the University of Illinois at Urbana Champaign in 2009. His research interests include machine learning, data mining and information retrieval.



XIAOFEI HE received a B.S. degree in Computer Science from Zhejiang University, China, in 2000 and a Ph.D. degree in Computer Science from the University of Chicago, in 2005. He is a Professor in the State Key Lab of CADCG at Zhejiang University, China. Prior to joining Zhejiang University, he was a Research Scientist at Yahoo! Research Labs, Burbank, CA. His research interests include machine learning, information retrieval, and computer vision. He is a senior member of IEEE.

XIAN-SHENG HUA (F'16) received the B.S. and Ph.D. degrees in applied mathematics from Peking University, Beijing, in 1996 and 2001, respectively. In 2001, he joined Microsoft Research Asia as a Researcher and has been a Senior Researcher of Microsoft Research Redmond since 2013. He became a Researcher and the Senior Director of the Alibaba Group in 2015. He has authored or co-authored over 250 research papers and has filed over 90 patents. His research interests have been in the areas of multimedia search, advertising, understanding, and mining, and pattern recognition and machine learning. He was honored as one of the recipients of MIT35. He received the Best Paper and Best Demonstration Awards at ACM Multimedia 2007, the Best Student Paper Award at ACM CIKM 2009, the Best Paper Award at MMM 2010, the Best Demonstration Award at ICME 2014, and the Best Paper Award of the IEEE TRANSACTIONS ON CSVT in 2014. He served as a Program Co-Chair for the IEEE ICME 2013, the ACM Multimedia 2012, and the IEEE ICME 2012, and on the Technical Directions Board of the IEEE Signal Processing Society. He is an ACM Distinguished Scientist and IEEE Fellow.