HetETA: Heterogeneous Information Network Embedding for Estimating Time of Arrival

Huiting Hong[*]
Yucheng Lin[*]
honghuiting@didiglobal.com
linyucheng@didiglobal.com
AI Labs, Didi Chuxing

Xiaoqing Yang
Zang Li
yangxiaoqing@didiglobal.com
lizang@didiglobal.com
AI Labs, Didi Chuxing

Kun Fu
fukunkunfu@didiglobal.com
AI Labs, Didi Chuxing

Zheng Wang
wangzhengzwang@didiglobal.com
AI Labs, Didi Chuxing

Xiaohu Qie
tiger.qie@didiglobal.com
Technology Ecosystem &
Development, Didi Chuxing

Jieping Ye
yejieping@didiglobal.com
AI Labs, Didi Chuxing

ABSTRACT

The estimated time of arrival (ETA) is a critical task in the intelligent transportation system, which involves the spatiotemporal data. Despite a significant amount of prior efforts have been made to design efficient and accurate systems for ETA task, few of them take structural graph data into account, much less the heterogeneous information network. In this paper, we propose HetETA to leverage heterogeneous information graph in ETA task. Specifically, we translate the road map into a multi-relational network and introduce a vehicle-trajectories based network to jointly consider the traffic behavior pattern. Moreover, we employ three components to model temporal information from recent periods, daily periods and weekly periods respectively. Each component comprises temporal convolutions and graph convolutions to learn representations of the spatiotemporal heterogeneous information for ETA task. Experiments on large-scale datasets illustrate the effectiveness of the proposed HetETA beyond the state-of-the-art methods, and show the importance of representation learning of heterogeneous information networks for ETA task.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; *Data mining*; • **Computing methodologies → Neural networks**.

KEYWORDS

Estimated time of arrival; traffic prediction; graph neural networks

ACM Reference Format:
Huiting Hong, Yucheng Lin, Xiaoqing Yang, Zang Li, Kun Fu, Zheng Wang, Xiaohu Qie, and Jieping Ye. 2020. HetETA: Heterogeneous Information Network Embedding for Estimating Time of Arrival. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3394486.3403294

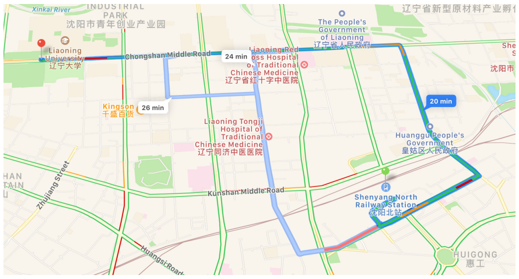


Figure 1: Application of estimated time of arrival (ETA) to route planning. The origin is in green and the destination is in red. The color of roads indicates the traffic congestion level. The darker the color, the more congested.

1 INTRODUCTION

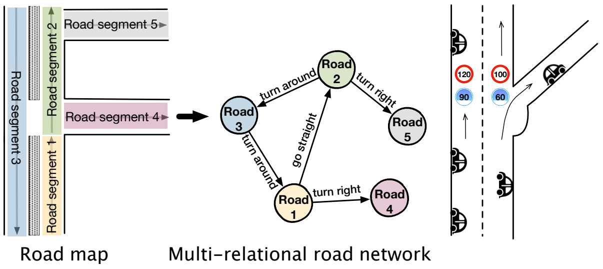
With the growing number of vehicles and travel demands of people, intelligent transportation systems have become the key role to make safer, more coordinated, and more efficient use of traffic networks. The estimated time of arrival (ETA), a core functionality in the intelligent transportation system, measures the travel time when a vehicle[1] is expected to arrive at a certain destination from origin. An accurate travel time estimation can save user time [8] and optimize vehicle dispatching [35] via mining complicated spatiotemporal information. A simple ETA method is to average the historical travel time between the pair of origin and destination [23, 30]. Naturally, these historical mean based methods produce low accuracy, due to the sparseness problem of short-term data for the same route. In addition, these approaches often fail to meet the needs of applications such as route planning [18]. As shown in Figure 1, three routes between the same pair of origin and destination

[*] Both authors contributed equally to this research.

[1] ETA also refers to time estimation for aircraft, ship, computer file, et. al. to reach the place it is directed to. In this paper, we focus on the movement of vehicles.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403294



(a) Translation from a road map to a multi-relational road network.

(b) A freeway with a right hand lane exit.

Figure 2: The road network information for ETA task.

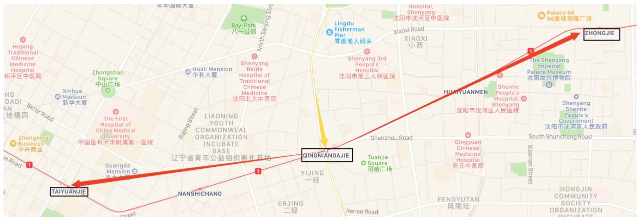


Figure 3: An example of the traffic flow pattern in Shenyang. The yellow line is Qingniandajie and the red lines on the left and right are Taiyuanjie and Zhongjie respectively.

are provided with the travel time in the route planning software, which requires the use of route-based ETA solutions.

The route-based ETA solution turns the ETA problem into traffic prediction task, that is, predicting transit time of each road segment firstly, then calculating the total time of a given trajectory path. However, this line of research focuses on collecting various kinds of raw data [10, 11, 20] or optimizing the concatenation of road segments [12, 13, 31]. On the contrary, the bulk of congestion estimation methods [21, 33, 34] are more concerned with representation learning of spatiotemporal data, especially the use of graph neural networks (GNNs) [7, 17]. These models are mainly applied to sensor network data instead of real urban road network data, which is the essential geospatial information for ETA task.

In this paper, we employ GNN to embed road network data for ETA task. This problem is very challenging since 1) the relationship of connection in road networks/graphs is more complicated than single-relation sensor networks 2) and the links between road segments are very sparse. Considering road segments as vertices in the road network, the relationships between vertices could be “go straight”, “turn right” or “turn around” and so on [Figure 2(a)]. Different relationships imply different traffic patterns. For example, Figure 2(b) shows that vehicles usually move at high speed when going straight on the highway while they usually slow down when turning right to the exit lane. Therefore, it is important to take various relationships between road segments into account during road network embedding. In addition, the road network is constructed as a large-scale network with low density. Take the road network of Shenyang (detailed in Section 4.1) as an example: there are 74, 685 vertices with an average number of 2.52 neighbors. It is much sparser than sensor networks like METR-LA [14] (207 nodes with an average degree of 13.69) or PEMS-BAY[3] (325 nodes with an average degree of 13.79). Such sparse network makes it difficult to collect sufficient messages from neighbors via GNNs.

To tackle the above challenges, we introduce heterogeneous information network (HIN) [28] to ETA task. Specifically, we translate the road map into a multi-relational network (a type of HIN), as shown in Figure 2(a), where edges indicate the directions of connection between road segments. And we also construct a vehicle-trajectories based network, where vertices are the same as the translated road network and an edge from vertex i to j indicates that vehicles travel from road segment i to j frequently. The vehicle-trajectories based network implicitly incorporates the traffic flow pattern of a city, for instance, vehicles at Qingniandajie are most likely going to Taiyuanjie (a transportation hub with railway station and a passenger transport station in there) or Zhongjie (a famous shopping street in Shenyang). Besides, the vehicle-trajectories based network can supplement some information that road network can not mine. For example, drivers may find out new roads before map update and some may know smoother/better routes by experience.

Moreover, heterogeneity also exists in temporal data. Figure 4 shows changes in traffic condition on a road segment over two weeks. Obviously, the peak hours of weekdays appear around 8:00 AM and 6:00 PM, while the rush hours on weekend last from 8:00 AM to 6:00 PM. In view of this, we class the temporal information into three categories: recent periods which are closely related to traffic condition to predict, daily periods referring to daily-periodic patterns and weekly periods implying weekly-periodic patterns.

In this paper, we propose a fused framework called HetETA to incorporate the above-mentioned spatiotemporal heterogeneous information for ETA task. Specifically, HetETA models temporal heterogeneous information from recent periods, daily periods and weekly periods respectively with three components. Each component has the structure of a double-stuffed sandwich, consisting of two graph neural networks, to embed spatial heterogeneous information from the multi-relational road network and vehicle-trajectories based network respectively, placed between three convolutional neural networks used to process information on time axis. Furthermore, we develop an attention-based graph network with fast localized spectral filtering to learn better representations when the HIN is sparse. In a nutshell, the key innovations of this paper are:

- To the best of our knowledge, this is the first time that HINs and graph neural network technology are applied to ETA task. We extract heterogeneous information both in the view of space and time and propose a framework HetETA to fuse them and learn representations toward ETA task.
- We design an attention-based graph network with fast localized spectral filtering, named Het-ChebNet, to embed sparse heterogeneous information network under the space requirement proportional to the number of edges.
- Extensive experiments are conducted on four real-world vehicle-trip datasets in a large-scale urban road network. Our model significantly outperforms other methods. The ablation studies verify the efficacy of vehicle-trajectories

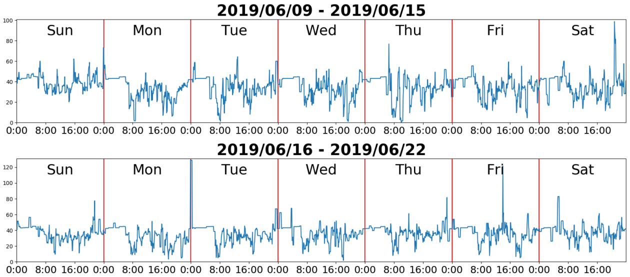


Figure 4: The temporal pattern of a road in Shenyang. The x- and y-axis represent time and speed (km/h) respectively.

based network introduced in this paper. In addition, we import representations learned by HetETA as extra features of Wide-Deep-Recurrent (WDR) [32] which is a well-designed feature system for ETA on Didi Chuxing's platform, and the experimental results demonstrate the power of heterogeneous information network embedding.

2 RELATED WORK

2.1 Estimated Time of Arrival

The estimated time of arrival (ETA) or travel time estimation is one of the key topics in the intelligent transportation system. There are mainly two categories of existing solutions. The first category is the route-based methods [6, 15, 27, 31, 32], which estimate the travel time by considering the road segments and intersections in the trip. SMA [15] models the correlation between different road segments in terms of their historical patterns. WDR [32] formulates ETA as a regression problem and proposes a Wide-Deep-Recurrent (WDR) architecture, in which, the recurrent module is specially designed to handle sequential features. The second category is the route-free methods [16, 19, 30], which focus on the origin and destination locations to make predictions. TEMP [30] estimates the time of a queried trip by calculating the weighted average of neighboring trips with similar origin and destination locations. ST-NN [16] predicts the travel distance and time jointly by designing a unified feed-forward neural network. However, these efforts do not fully mine the spatiotemporal data, and fail to exploit the rich semantic information contained in road network structures.

2.2 Traffic Forecasting with Spatiotemporal Graph Convolution

In the past two years, GNNs for traffic forecasting problem have received much attention. These works try to model spatial and temporal dependencies of the road traffic through spatiotemporal graph convolution. DCRNN [21] treats the traffic flow as a diffusion process [1] on a directed graph and proposes diffusion convolution to capture the spatial dependency. On the other hand, it captures the temporal dependency by using sequence-to-sequence architecture with gated recurrent units (GRUs) [4]. In consideration of time efficiency, instead of RNN structure, STGCN [34] employs a 1D convolution layer followed by gated linear units (GLUs) [5] to extract temporal features. Inspired by WaveNet [25], Graph WaveNet [33] stacks dilated casual convolutions to handle long-range temporal sequences. In addition, Graph WaveNet introduces a self-adaptive graph to better extract the hidden spatial features. ASTGCN [9] models temporal dependencies of three time scales, i.e., recent, daily-periodic and weekly-periodic. Spatial attention and temporal attention are performed to help dynamically capture the correlations between different time scales.

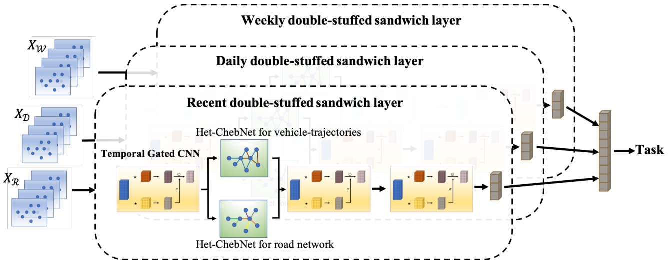

Weekly double-stuffed sandwich layer
Daily double-stuffed sandwich layer
Recent double-stuffed sandwich layer
$X_{\mathcal{W}}$
$X_{\mathcal{D}}$
$X_{\mathcal{R}}$
Het-ChebNet for vehicle-trajectories
Temporal Gated CNN
Het-ChebNet for road network
Task

Figure 5: HetETA consists of three components for learning spatiotemporal heterogeneous information of recent periods, daily periods and weekly periods respectively. Three components are connected by a fully connected layer and output the prediction value.

However, the above-mentioned methods show common weaknesses as follow: 1) They treat the road network as a homogeneous graph, and the differences between edges are neglected. 2) Most experiments are performed on public datasets like META-LA [14] or PEMS [3], where traffic conditions are collected from loop detectors in highways. In fact, situations in urban road network are much more complicated. There are streets, elevated roads, private roads and so on, and they usually show quite distinct traffic properties to highways. Obviously, it is impractical to set up loop detectors for the entire urban road network. 3) Many researches [9, 33, 36] fail to take model scalability into consideration. Their proposed models work well on small networks but are hard to scale up for large-scale networks due to the dense weight matrix of size $|V| \times |V|$ involved.

3 METHODOLOGY

3.1 Problem Statement

In this paper, we aim to produce meaningful representations for arrival time estimation. Existing ETA solutions have introduced a variety of features, such as weather information, personalized information, traffic information and temporal information. And the system architecture such as WDR [32] based on these features is set up, representing all of information with the form of a row. However, such type of features can not fully mine the spatial correlations in geospatial information which is in the form of structural networks. Therefore, we define the estimated time of arrival task as the spatiotemporal network embedding problem:

Definition 3.1 (Spatiotemporal network embedding for ETA). Given a departure query $q = (o_q, d_q, t_q, P_q)$ at time t_q, from origin o_q to destination d_q via the path route P_q, our goal is to estimate the travel time y_q by embedding spatiotemporal traffic networks in history $\{G^{(t_q-\tau+1)}, G^{(t_q-\tau+2)}, \ldots, G^{(t_q)}\}$, where $t_q - \tau + 1$ denotes the time period and τ is the number of previous time periods. A spatiotemporal traffic network at time t is represented as a directed multi-relational graph $G^{(t)} = (V, E, R, X^{(t)})$, where $V = \{v_i\}_{i=1}^{|V|}$ denotes the set of vertices (i.e., road segments) and E denotes the set of edges. An edge $e_{ijk} = (v_i, v_j, r_k) \in E$ indicates the vertex $v_i \in V$ links to the vertex $v_j \in V$ with a relation type $r_k \in R$, where $R = \{r_i\}_{i=1}^{|R|}$ is the set of relation types. The features of vertices $X^{(t)} = \{x_i^{(t)} \in \mathbb{R}^n\}_{i=1}^{|V|}$ contain static features that characterize road segments (e.g., length, width) and dynamic features at time t (e.g., traffic speed, volume). Here, n is the total dimension of features.

This problem is challenging because of two aspects: 1) complex heterogeneous structure of graphs, which is hard to represent and preserve in a low-dimensional space; 2) intricate correlation that exists among road segments and time periods. To tackle these challenges, we propose a HIN embedding framework for ETA task, namely HetETA, which is detailed in the next subsection.

3.2 HetETA

As the graph neural network (GNN) has proven to be a successful graph structure based model to learn vectorized node representations, we employ GNNs to learn spatial correlations and extract meaningful representation on the heterogeneous information road network and vehicle-trajectories based network. Recent research shows convolution neural networks (CNNs) have the advantages of parallelization, trainability and inference speed compared with recurrent neural networks (RNNs) [24, 34]. Thus, we employ CNNs to analyze temporal correlations on time-series axis. The overall architecture of HetETA is illustrated in Figure 5. GNNs and CNNs work together under the structure of the double-stuffed sandwich in three components, to learn the correlations in spatiotemporal heterogeneous information of recent periods, daily periods and weekly periods respectively. Layer normalization is implemented within the double-stuffed sandwich to deal with overfitting. To train HetETA to predict the travel time y_q given a query $q = (o_q, d_q, t_q, P_q)$ conditioned on the graph $G^{(t_q-\tau+1):(t_q)}$, we minimize the loss function as follows:

$$\mathcal{L}\left(\hat{y}_q; G^{(t_q-\tau+1):(t_q)}, \Theta\right) = \frac{|\hat{y}_q - y_q|}{y_q} + \gamma||\Theta||, \tag{1}$$

where the prediction value $\hat{y}_q = \sum_{p \in P_q} \hat{y}_p$, and $\hat{y}_p$ denotes the predicted transit time of road segment p. We train the parameters of HetETA Θ by the AMSGrad [26] optimizer.

3.2.1 *Three Components for Recent, Daily and Weekly Periods.*

As shown in Figure 5, we intercept three sequences in length of $L_{\mathcal{R}}$, $L_{\mathcal{D}}$ and $L_{\mathcal{W}}$, respectively, as the input of the recent, daily-period and weekly-period components. For the recent component, $\mathbf{X}_{\mathcal{R}} = [X^{(t_q-L_{\mathcal{R}}+1)}, X^{(t_q-L_{\mathcal{R}}+2)}, \ldots, X^{(t_q)}] \in \mathbb{R}^{L_{\mathcal{R}} \times |V| \times n}$ represents current traffic status. To learn periodic and trend patterns, we utilize historical traffic data in the last $L_{\mathcal{D}}/L_{\mathcal{W}}$ days/weeks, which has the same time period as the next future period. That is, $\mathbf{X}_{\mathcal{D}} = [X^{(t_q+1-L_{\mathcal{D}}*T_D)}, X^{(t_q+1-(L_{\mathcal{D}}-1)*T_D)}, \ldots, X^{(t_q+1-T_D)}] \in \mathbb{R}^{L_{\mathcal{D}} \times |V| \times n}$ for daily-period component, and weekly-period component $\mathbf{X}_{\mathcal{W}} = [X^{(t_q+1-L_{\mathcal{W}}*T_D*7)}, X^{(t_q+1-(L_{\mathcal{W}}-1)*T_D*7)}, \ldots, X^{(t_q+1-T_D*7)}] \in \mathbb{R}^{L_{\mathcal{W}} \times |V| \times n}$, where T_D is the number of time periods in one day. Note that all of graphs $G^{(t)}$ have the same vertices and edges, and only the features of the vertices $X^{(t)}$ change over time. We construct three components with the same deep network structure, i.e., three gated convolutional layers and a double-stuffed graph convolution layer in between. In the following, we omit the subscript $\mathcal{R}/\mathcal{D}/\mathcal{W}$ which denotes the identity of the component for simplicity.

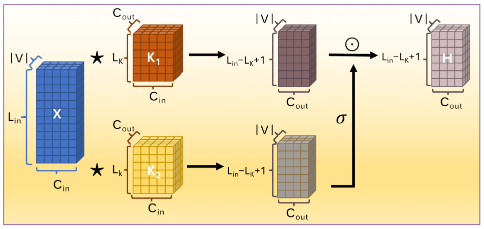


Figure 6: Layer structure of temporal gated CNN.

3.2.2 *Gated CNNs for Temporal Correlations.*

Inspired by [34], we employ the causal convolution with C_{out} kernels of size $L_K \times 1 \times C_{\text{in}}$ to sequentially convolve information along time axis, where C_{in} and C_{out} are the number of input channels and output channels, respectively. To adapt the dynamics of temporal correlations, we control temporal information flow in CNN by means of gating mechanisms, which is crucial in recurrent neural networks. As shown in Figure 6, a gated CNN layer outputs the hidden state $\mathbf{H}$ given the input $\mathbf{x} \in \mathbb{R}^{L_{\text{in}} \times |V| \times C_{\text{in}}}$ following the convolution rule:

$$\mathbf{H} = (\mathbf{K}_1 \star \mathbf{x}) \odot \sigma(\mathbf{K}_2 \star \mathbf{x}) \in \mathbb{R}^{(L_{\text{in}} - L_K + 1) \times |V| \times C_{\text{out}}}, \tag{2}$$

where $\mathbf{K}_1$ and $\mathbf{K}_2$ are convolution kernels with the same size of $[L_K \times 1 \times C_{\text{in}}, C_{\text{out}}]$. And $\odot$ denotes the element-wise Hadamard product. The sigmoid function $\sigma(\cdot)$ plays the role of gate to control the ratio of information flowing to the next layer.

For the first gated CNN layer in the structure of double-stuffed sandwich, $\mathbf{x} = \mathbf{X} \in \mathbb{R}^{L \times |V| \times n}$ and the output state $\mathbf{H} = \mathbf{H}_1 \in \mathbb{R}^{(L - L_K + 1) \times |V| \times C_1}$. The Het-ChebNet we designed in interlayer operates graph convolution and keep the output in the same dimension as the input. That is the output of the double-stuffed graph convolution layer, which consists of two Het-ChebNets, is $\mathbf{H}_2 \in \mathbb{R}^{(L - L_K + 1) \times |V| \times 2 * C_1}$. Thereby, the next gated CNN layer, with $\mathbf{x} = \mathbf{H}_2$ as the input, outputs $\mathbf{H} = \mathbf{H}_3 \in \mathbb{R}^{(L - 2*(L_K - 1)) \times |V| \times C_3}$. To obtain a comprehensive hidden state for the given spatiotemporal information, a gated CNN layer with a kernel size of $[(L - 2 * (L_K - 1)) \times 1 \times C_3, C_3]$ is used as the last layer of the double-stuffed sandwich, i.e., $\mathbf{x} = \mathbf{H}_3$ and $\mathbf{H} = \mathbf{H}_4 \in \mathbb{R}^{1 \times |V| \times C_3}$.

3.2.3 *Het-ChebNet for Spatial Correlations.*

We model the spatial correlations by considering the traffic flow as spatial information propagation process on the road network. We propose Het-ChebNet, which is primarily motivated as an adaptation of ChebNet [7] performing localized spectral filtering by Chebyshev polynomials:

$$g_\theta \star_G \mathbf{x} = g_\theta(\mathbf{L})\mathbf{x} = \sum_{z=0}^{Z} \theta_z \Gamma_z(\tilde{\mathbf{L}})\mathbf{x}, \tag{3}$$

where the vector of polynomial coefficients $\theta \in \mathbb{R}^{Z+1}$ is trainable parameters. $\mathbf{L}$ is the normalized Laplacian matrix formulated as $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \in \mathbb{R}^{|V| \times |V|}$, in which $\mathbf{I}$ denotes the identity matrix. The weighted adjacency matrix $\mathbf{A}$ records the connection

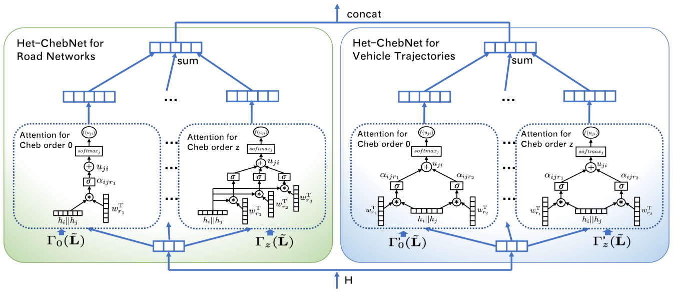

concat
Het-ChebNet for Road Networks
sum
Attention for Cheb order 0
$f(u_{ji})$
$softmax_j$
u_{ji}
α_{ijr_1}
$h_i||h_j$
$w_{r_1}^{\mathrm{T}}$
Attention for Cheb order z
$w_{r_2}^{\mathrm{T}}$
$w_{r_3}^{\mathrm{T}}$
$\Gamma_0(\tilde{\mathbf{L}})$
$\Gamma_z(\tilde{\mathbf{L}})$
Het-ChebNet for Vehicle Trajectories
α_{ijr_2}
$\Gamma'_0(\tilde{\mathbf{L}})$
$\Gamma'_z(\tilde{\mathbf{L}})$
H

Figure 7: The structure of double-stuffed graph convolution layer, where two Het-ChebNets are employed for road network and vehicle trajectories.

weight between two vertices in the graph, and the diagonal degree matrix $\mathbf{D}$ sums up the weights of each vertex, i.e., $\mathbf{D_{ii}} = \sum_j \mathbf{A}_{ij}$. $\Gamma_z(\tilde{\mathbf{L}}) \in \mathbb{R}^{|V|\times|V|}$ is the Chebyshev polynomial, which performs spectral filtering on order z evaluated at the scaled Laplacian $\tilde{\mathbf{L}} = 2\mathbf{L}/\lambda_{\max} - \mathbf{I}$, where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{L}$. Chebyshev polynomial $\Gamma_z(\tilde{\mathbf{L}})$ of order z can be computed by the recurrence formula $\Gamma_z(\tilde{\mathbf{L}}) = 2\tilde{\mathbf{L}}\Gamma_{z-1}(\tilde{\mathbf{L}}) - \Gamma_{z-2}(\tilde{\mathbf{L}})$ with $\Gamma_0(\tilde{\mathbf{L}}) = \mathbf{I}$ and $\Gamma_1(\tilde{\mathbf{L}}) = \tilde{\mathbf{L}}$.

Though ChebNet has proven its power in processing graph structure data, it suffers an inadequate characterization of multiple relationships in the road network, which is important to learn behaviors of traffic flow as we mentioned in Section 1. In pursuit of better representation, we introduce attention mechanism to ChebNet and rewrite the graph convolution operation in Eq. (3) as:

$$g_\theta \star_G \mathbf{x} = g_\theta(\mathbf{L})\mathbf{x} = \sum_{z=0}^{Z} \theta_z \left(\Gamma_z(\tilde{\mathbf{L}}) \odot f(\mathbf{U}_z) \right) \mathbf{x}, \tag{4}$$

where $\odot$ denotes the element-wise Hadamard product. $\mathbf{U}_z \in \mathbb{R}^{|V|\times|V|}$ is an attention sparse matrix encoding the score of edges in the multi-relational graph. Given an edge $e_{ijk} = (v_i, v_j, r_k)$, its attention score can be computed in the form as:

$$\alpha_{ijk} = \sigma\left({w_{r_k}}^{\mathrm{T}} \left[h_i || h_j \right] \right), \tag{5}$$

where σ is an activation function implemented by LeakyReLU($\cdot$) [22], and $||$ refers to the concatenation between vectors $h_i \in \mathbb{R}^d$ and $h_j \in \mathbb{R}^d$, denoting the low-dimensional representations of v_i and v_j respectively. $w_{r_k} \in \mathbb{R}^{2d}$ is the trainable parameter to adapt the relation type r_k. Note that vertex v_i may link to v_j with multiple relation types in a graph, so an element in the attention matrix $\mathbf{U_z}$ is the summation of attention scores of edges that v_i links to v_j:

$$u_{ji} = \sum_{k,(v_i,v_j,r_k)\in E} \alpha_{ijk}. \tag{6}$$

Then we apply a softmax function $f(\cdot)$ over the in-coming edges of each vertex to normalize the attention matrix:

$$f(u_{ji}) = \exp(u_{ji}) / \sum_{k,(v_k,v_j)\in E} \exp(u_{jk}). \tag{7}$$

Recall that the ChebNet conducts graph Fourier transform on a symmetric adjacency matrix $\mathbf{A}$, which is usually an undirected graph. However, the road network is defined as a directed graph, i.e., a vehicle can travel from road segment v_i to v_j while it may not be able to travel from v_j to v_i due to traffic regulations. To better deal with ETA task involved with directed road networks, we add reverse edges of out-going edges and set them with reverse relations. For instance, we link v_i to v_j with relation "out-going when turn left" when v_j links to v_i with relation "turn left" in the road network. As a result, the road network becomes a bidirectional graph which presents a symmetric adjacency matrix, and it offers attention mechanism more semantic information to capture spatial correlations both on upstream and downstream traffic. In addition to the turning direction relationships between road segments, we exploit vehicles historical trajectories and establish a new type of relationship named "likely going to", which indicates vehicles at the road segment v_i are most likely going to the road segment v_j with β-hop[2]. More details of Het-ChebNet are provide in Appendix A.

A double-stuffed graph convolution layer. We divide the multi-relational graph $G^{(t)}$ into two networks, i.e., the road network (denoted as $G^{(t)}_{\text{road}}$) and the vehicle-trajectories based network (denoted as $G^{(t)}_{\text{vehicle}}$). Both of them share the same vertices and the same input signals of vertices. The edges of road network are constructed according to the turning direction relationships between road segments. And the vehicle-trajectories based network consists of edges with the relationship "likely going to" (including its reverse relation "out-going when likely going to"). As shown in Figure 7, the double-stuffed graph convolution layer consists of two Het-ChebNets to model the spatial correlations of road network and vehicle-trajectories based network, respectively. In the structure of double-stuffed sandwich, the input signal of the double-stuffed graph convolution layer is $\mathbf{H}_1 \in \mathbb{R}^{(L-L_K+1)\times|V|\times C_1}$. We generalize Het-ChebNet to 3-D variables by performing C_1 spectral filtering operations in Eq. (4) on graph $G^{(t)}$ with parameters $\{\theta^{(t)}_1, \theta^{(t)}_2, \ldots, \theta^{(t)}_{C_1}\}$, where $t = 1, 2, \ldots, L - L_K + 1$. Let a time slice of $\mathbf{H}_1$, i.e., $\mathbf{H}^{(t)}_1 \in \mathbb{R}^{|V|\times C_1}$ be the low-dimensional representations of vertices, the attention scoring function Eq. (5) becomes as follows:

$$\alpha^{(t)}_{ijk} = \sigma\left({w_{r_k}}^{\mathrm{T}} \left[\mathbf{H}^{(t)}_{1i} || \mathbf{H}^{(t)}_{1j} \right] \right), \tag{8}$$

where $\mathbf{H}^{(t)}_{1i}$ is i-th row of $\mathbf{H}^{(t)}_1$. Conducting such convolution both on $G^{(t)}_{\text{road}}$ and $G^{(t)}_{\text{vehicle}}$, we obtain $\mathbf{H}_{\text{road}}$ and $\mathbf{H}_{\text{vehicle}}$ with the same size of $(L-L_K+1)\times|V|\times C_1$. Then we concatenate them along the channel axis and use it as input into the next gated CNN layer as follows:

$$\mathbf{H}_2 = [\mathbf{H}_{\text{road}} || \mathbf{H}_{\text{vehicle}}] \in \mathbb{R}^{(L-L_K+1)\times|V|\times 2*C_1}. \tag{9}$$

3.2.4 *Fusion Layer for Prediction.*

With the output of three components $\mathbf{H}_{\mathcal{R}4}$, $\mathbf{H}_{\mathcal{D}4}$ and $\mathbf{H}_{\mathcal{W}4}$, which have the same shape of $|V| \times C_3$, HetETA predicts the transit time of road segments by a full connected layer:

$$\hat{Y} = \frac{\mathbf{S}}{\sigma\left(\left[\mathbf{H}_{\mathcal{R}4} || \mathbf{H}_{\mathcal{D}4} || \mathbf{H}_{\mathcal{W}4} \right] \mathbf{W} + b \right) * 120}, \tag{10}$$

where $\mathbf{W} \in \mathbb{R}^{3*C_3\times 1}$ is the learnable weight to fuse the outputs of three components, and $b \in \mathbb{R}^{|V|}$ is a vector of biases. $\mathbf{S} \in \mathbb{R}^{|V|\times 1}$

[2] β-hop means that vehicles pass $\beta - 1$ road segments from v_i to v_j, i.e., vertex v_i is β-hop reachable to vertex v_j in vehicles historical trajectories.

Table 1: Statistics of Datasets

Dataset	Split	Date	Pickup	Trip
SY_6	training set	6.9-6.15 (2019)	687K	800K
	validation set	6.16-6.22 (2019)	700K	820K
	test set	6.23-6.29 (2019)	611K	709K
SY_7	training set	7.9-7.15 (2019)	715K	816K
	validation set	7.16-7.22 (2019)	756K	853K
	test set	7.23-7.29 (2019)	665K	746K

Table 2: Comparison Performance on ETA Task

Dataset	SY_6_Trip			SY_6_Pickup		
Metric	MAPE	MAE	RMSE	MAPE	MAE	RMSE
GRU	13.84%	129.99	216.52	24.89%	52.37	91.86
DCRNN	13.21%	124.01	208.10	24.09%	49.62	85.69
STGCN	12.88%	119.96	200.76	23.33%	47.55	**82.24**
GWN*	12.89%	121.60	205.39	23.64%	48.96	85.54
ASTGCN*	12.57%	117.53	119.17	23.39%	48.50	86.13
HetETA	**12.32%**	**116.44**	**197.26**	**22.96%**	**47.16**	82.77
Dataset	SY_7_Trip			SY_7_Pickup		
Metric	MAPE	MAE	RMSE	MAPE	MAE	RMSE
GRU	13.99%	123.27	193.06	23.77%	51.75	83.28
DCRNN	13.28%	116.80	183.48	23.04%	49.09	77.62
STGCN	12.94%	113.48	178.42	22.29%	46.59	74.01
GWN*	13.01%	114.08	179.25	22.60%	47.66	75.98
ASTGCN*	12.66%	111.26	175.95	22.28%	47.22	76.27
HetETA	**12.39%**	**109.17**	**173.03**	**21.89%**	**46.23**	**73.78**

denotes the length of each road segment and the division refers to element-wise division. The combination of sigmoid $\sigma(\cdot)$ and multiplication is used to scale prediction values which is favorable to model training, where 120 indicates the speed limit of 120km/h.

4 EXPERIMENTS

In this section, we present the experimental evaluations of HetETA and the competing baselines over four large-scale offline datasets. The heart of HetETA is to learn informative representations for ETA task by embedding spatiotemporal networks. To make fair and impartial comparisons, we evaluate comparison models not only on ETA task, but also on traffic speed prediction task for which GNN-based spatiotemporal graph embedding methods were originally designed. Whereafter, we incorporate the low-dimensional representations learned by HetETA with the hand-crafted features in Wide-Deep-Recurrent (WDR) [32], which is the state-of-the-art method for ETA task, to verify the effectiveness of HIN embedding.

4.1 Datasets

We perform our experiments on Shenyang, capital city of Liaoning Province in China. We build a road network of Shenyang city according to the commercial map provided by Didi Chuxing. It is a multi-relational graph with $74,685$ vertices and $94,127$ edges, where the relation type indicates the turning direction between road segments, including straight forward, slight left, left, sharp left, slight right, right, sharp right and turn around. The node features are divided into static and dynamic ones. The static features, including road type, segment width, segment length, speed limits, lane number, etc., do not change over time. On the contrary, the dynamic features change by period of 5 minutes. Here, we calculate the average speed of passing cars in each road segment and use it as the dynamic feature. Note that, because of the data sparsity, there are still some road segments without any vehicle passing by in certain periods. For these road segments, we set default values instead, according to their historical speed or the average speed of the same road type (like highway, local street, etc.). We collect the floating-car data in Shenyang from May 1st to July 31st, 2019 on DiDi platform. The floating-car data can be divided into two types: pickup data and trip data. A pickup sample records the car trajectory from the moment the driver receives the request until he/she picks up the passenger. A trip sample records the trajectory from the passenger is on board until the car arrives at the destination.

With the historical floating-car information provided, we design a vehicle-trajectories based network to catch the co-occurrence between road segments, which is not provided in road network. It is a directed and weighted graph, where each vertex represents a road segment in the same way and each edge from vertex i to j indicates that some vehicles traveled from i to j in history. In addition, to avoid the constructed graph being dense, we add constraints on β-hop and frequency. Specifically, a directed edge from i to j in a β-hop vehicle-trajectories network means that there are $\beta - 1$ road segments between i and j in some trajectories. Edge weight indicates the frequency of co-occurrence and we only keep the top κ frequent adjacency edges for each vertex. In our experiments, β is set to 3 and κ is set to 5, and we construct vehicle-trajectories based network by historical data in May. The final graph contains $727,666$ edges in total. In the preprocessing stage, we remove some inappropriate cases from the data: 1) short travel time less than 30 seconds; 2) abnormal trajectories caused by bad GPS signals; 3) pickup samples of reserving ride-hailing orders. After preprocessing, the path route of per query involves 20 road segments in the pickup data and 97 road segments in the trip data. Finally, we construct two datasets SY_6 and SY_7 in June and July 2019, respectively. Each of them contains pickup and trip records of 3 weeks, with details in Table 1.

4.2 Comparison Methods and Metrics

The experimental settings of our proposed model HetETA and the baseline methods for comparison are listed as following:

- **HetETA**: We implement HetETA in Python based on Tensorflow toolbox[3]. The sequences length of three components are $L_{\mathcal{R}} = L_{\mathcal{D}} = L_{\mathcal{W}} = 4$. The size L_K of the causal convolution kernels in the gated CNN is set to 2, with the output channels $C_1 = 8$ and $C_3 = 11$ respectively. Both of two Het-ChebNets, convolving on the road network and the vehicle-trajectories based network respectively, have $Z = 2$ Chebyshev polynomials. The regularization factor $\gamma = 0.0001$ and the batch size of input queries is set as the number of queries in the

[3] Available at https://github.com/didi/heteta

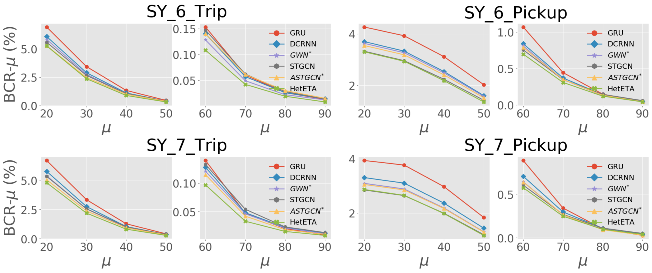


Figure 8: BCR-μ comparison on ETA task.

same period. We train HetETA with an initial learning rate 0.01 and drop learning rate by 0.15 every 5 epochs.

- **GRU [4]**: Gated Recurrent Unit Network used to process recent sequential information. We input recent 4 historical periods, i.e., $X = X_{\mathcal{R}}$ for each query, into GRU, which does not involve the spatial information such as road network.
- **DCRNN [21]**: Diffusion Convolutional Recurrent Neural Network incorporates both spatial and temporal dependency into a seq2seq framework for traffic flow prediction. We input recent 4 periods as well as the road network as the spatiotemporal information to DCRNN. Note that DCRNN requires a single-relational network so we offer the single-relational road network, where edges indicate the connection between two road segments but not the turning direction.
- **STGCN [34]**: Spatio-Temporal Graph Convolutional Networks combines graph convolution with 1-D convolution to process spatiotemporal information for traffic prediction. The spatiotemporal information provided as the input is the same as DCRNN. Again, we employ the single-relational road network for STGCN, which uses a conventional GCN to learn graph data.
- **GWN* [33]**: Graph WaveNet (GWN) stacks dilated casual convolutions and graph convolutions to handle spatiotemporal graph data. We remove the self-adaptive adjacency matrix in GWN for large-scale problems due to the multiplication of two matrices of size $|V| \times d$ involved (producing a dense matrix of size $|V| \times |V|$). Again, we provide the same spatiotemporal information as DCRNN in GWN*.
- **ASTGCN* [9]**: Attention Based Spatial-Temporal Graph Convolutional Network (ASTGCN) includes both spatial attention and temporal attention layer. However, the spatial attention operation is based on a dense matrix of size $|V| \times |V|$, which is hard to scale up for our road network. To make a comparison in our experiment, we revise the attention operation in ASTGCN as the attention mechanism in GAT [29], reducing the space complexity to $O(|E|)$. ASTGCN* accepts temporal information from three different time scales: recent, daily-periodic and weekly-periodic information, same as HetETA, i.e., $L_{\mathcal{R}} = L_{\mathcal{D}} = L_{\mathcal{W}} = 4$.

All comparison methods are evaluated by metrics below:

- **MAPE**: Mean Absolute Percentage Error computes the percentage between predicted values $\hat{y}$ and ground truth y: $MAPE = \frac{1}{m}\sum_{i=1}^{m}|\hat{y}_i - y_i|/y_i$. MAPE is the most popular metric for ETA task since the percentage is easier for people to conceptualize and it is robust against the outliers.
- **MAE**: Mean Average Error calculates the absolute residual for each data point: $MAE = \frac{1}{m}\sum_{i=1}^{m}|\hat{y}_i - y_i|$. A smaller MAE suggests the model is better at prediction.
- **RMSE**: Root Mean Squared Error describes the spread of residuals: $RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2}$.
- **Bad Case Rate**: The bad cases refer to the absolute percentage error of the prediction $|\hat{y}_i - y_i|/y_i > \mu\%$ and its absolute residual $|\hat{y}_i - y_i| > \nu$ seconds. In this paper, we set $\nu = 300$ and $\nu = 180$ for the trip dataset and the pickup dataset respectively, and present the rate of bad cases when varying μ in $\{20, 30, \ldots, 80, 90\}$, denoted as BCR-$\mu$.

Table 3: Comparison Results on Traffic Speed Prediction

Dataset	SY_6 (5min/15min/30min)		
Metric	MAPE	MAE	RMSE
GRU	35.07%/38.26%/40.55%	2.21/2.35/2.48	3.07/3.24/3.39
DCRNN	33.18%/36.08%/38.06%	2.15/2.30/2.39	2.99/3.18/3.28
STGCN	29.63%/31.94%/33.40%	1.93/2.05/2.11	2.79/2.96/3.07
GWN*	30.57%/32.99%/34.65%	1.98/2.07/2.19	2.81/2.93/3.10
ASTGCN*	29.58%/31.66%/32.48%	1.92/2.00/2.04	2.74/2.87/2.91
HetETA	**29.00**%/**30.93**%/**31.92**%	**1.89**/**1.96**/**2.01**	**2.73**/**2.85**/**2.90**
Dataset	SY_7 (5min/15min/30min)		
Metric	MAPE	MAE	RMSE
GRU	33.88%/37.02%/39.05%	2.10/2.24/2.30	2.94/3.12/3.21
DCRNN	31.95%/34.61%/36.38%	2.06/2.15/2.29	2.88/3.01/3.18
STGCN	28.67%/30.81%/32.27%	1.86/1.94/2.03	2.70/2.82/2.95
GWN*	29.59%/31.88%/33.49%	1.91/2.00/2.10	2.71/2.84/2.98
ASTGCN*	28.63%/30.47%/31.43%	1.82/1.92/1.97	2.62/2.75/**2.82**
HetETA	**28.12**%/**29.85**%/**30.73**%	**1.80**/**1.90**/**1.95**	**2.60**/**2.74**/**2.82**

All experiments are conducted on a 64-bit machine with Nvidia Tesla P100 GPU. Detailed parameter settings for baselines are available in Appendix B. For each method, we select the model with the best MAPE on validation set and report its performance on test set.

4.3 Results

4.3.1 Comparison Results. We train all models with the objective function Eq. (1) for ETA task. Table 2 lists the comparison of performance among HetETA and baseline models on four datasets. As expected, GRU with only the temporal sequences information performs worst. As spatial structural information is combined, DCRNN and STGCN achieve much lower MAPE, MAE and RMSE. It shows the benefit of structural information and the validity of GNN to embed graph data. GWN* does not perform better than STGCN, owing to the exclusion of the self-adaptive adjacency matrix. Besides, GWN* takes the advantages of the dilated causal convolution to deal with long-range temporal sequences, which is ineffective to short-range sequences used in our experiments. With the use of daily and weekly periodicity information, ASTGCN* shows competitive results but still performs worse than HetETA in the help of attention mechanism optimization by us[4]. HetETA outperforms all

[4] Our revision to the attention mechanism of ASTGCN somehow protects ASTGCN from noises caused by the full attention operation.

Table 4: Ablation Results of HetETA on ETA Task

Dataset	SY_6_Trip			SY_6_Pickup		
Metric	MAPE	MAE	RMSE	MAPE	MAE	RMSE
HetETA	**12.32**%	**116.44**	197.26	**22.96**%	47.16	82.77
w/o vehicle info.	12.50%	117.85	200.36	23.05%	**46.70**	**81.21**
w/o $\mathcal{D}\&\mathcal{W}$ cmpt.	12.46%	116.82	**197.11**	23.18%	47.72	83.28
w/o direction info.	12.43%	117.94	201.73	22.99%	47.00	82.12
w/o attention idea	12.45%	118.29	202.22	22.97%	47.69	83.92
Dataset	SY_7_Trip			SY_7_Pickup		
Metric	MAPE	MAE	RMSE	MAPE	MAE	RMSE
HetETA	**12.39**%	109.17	173.03	**21.89**%	**46.23**	**73.78**
w/o vehicle info.	12.40%	**108.43**	**171.70**	22.04%	46.37	74.26
w/o $\mathcal{D}\&\mathcal{W}$ cmpt.	12.64%	111.36	175.69	22.11%	46.98	75.16
w/o direction info.	12.44%	109.35	172.77	21.91%	46.24	73.82
w/o attention idea	12.41%	108.60	171.92	21.92%	46.31	74.04

Table 5: Effect of HetETA in Cooperation with WDR

Dataset	Metric	Pickup		Trip	
		WDR	WDR+HetETA	WDR	WDR+HetETA
SY_6	MAPE	21.13%	**20.74**%	12.08%	**11.94**%
	MAE	49.1	**46.5**	101.7	**100.1**
	RMSE	77.9	**72.9**	161.2	**158.5**
	BCR	0.90%	**0.87**%	0.89%	**0.83**%
SY_7	MAPE	20.23%	**19.99**%	12.21%	**11.97**%
	MAE	51.6	**49.7**	104.0	**101.1**
	RMSE	80.0	**75.1**	160.7	**156.0**
	BCR	0.95%	**0.77**%	0.92%	**0.83**%

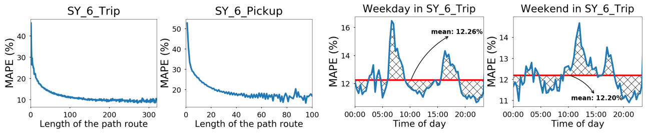
competing baselines in terms of MAPE and MAE on four datasets, verifying the gain of introducing heterogeneous information. In particular, HetETA decreases MAPE by 1.99%, 1.59%, 2.13% and 1.79% respectively over the most competitive baseline on four datasets. Note that it is tough to improve performance of ETA task and a slight decrease of MAPE usually means high commercial applications value of ETA task. The superiority of HetETA in the trip dataset is larger than the pickup dataset, where the path route is usually shorter than that of trip data. One possible reason is that the vehicle-trajectories based network can helps HetETA to model the behavior of traffic flow in a long-term view, which is more valuable for the trip data. Figure 8 depicts the bad case rate of each model with different threshold value μ on four datasets. It is evident that HetETA consistently achieves substantial gains over baselines by 3.40% ~ 46.67% and 0.69% ~ 28.33% in the trip dataset and the pickup dataset respectively. The improvement of BCR-μ becomes more significant when μ is larger.

Table 3 reports the performance results of comparison models in the traffic speed prediction task. All models are trained directly to predict the next term (5-30 minutes) traffic speeds for road segments by minimizing MAPE loss between prediction speeds and the ground truth. Again, HetETA achieves the best results in terms of all evaluation metrics. We can observe that HetETA and ASTGCN* tend to provide more superior performance for longer-term prediction, which is likely due to the utilization of periodicity information. Benefiting from the vehicle-trajectories based network, HetETA is superior to ASTGCN* in most cases.

4.3.2 Ablation Test. In this section, we further conduct ablation tests to study the effects of different heterogeneous information used in HetETA, including (a) the daily and weekly periodicity information, (b) the vehicle-trajectories based network, (c) multiple relations in the road network and (d) attention mechanism assisting in Het-ChebNet. Table 4 shows performance of HetETA against its ablations on ETA task, and we can observe that: (a) When the daily-period component and weekly-period component are removed (w/o $\mathcal{D}\&\mathcal{W}$ cmpt.), the model performs worst compared with other variants of HetETA but still performs better than baseline models. It indicates that the periodicity information plays an importance role for drawing trend patterns in ETA task; (b) Without the vehicle-trajectories based network (w/o vehicle info.), MAPE increases while MAE and RMSE decrease in some cases. It is a sign that the model may be subject to overfitting due to outliers, and the traffic flow behavior maintained in the vehicle-trajectories based network serves more reliable information and improves generalization of HetETA; (c) Replacing the multi-relational road network with a single-relational road network (no more turning direction information in the road network, w/o direction info.), the performance becomes worse than that of HetETA in most cases. It demonstrates that the turning direction between road segments has an effect on traffic patterns and it should be well considered in the model for ETA task. (d) Compared with the ablation of attention mechanism (w/o attention idea), HetETA performing graph convolution with Eq. (4) has advantage over the model that performs graph convolution with Eq. (3). It verifies the effect of our designed Het-ChebNet to adapt multiple relations in the graph. And it shows again that the consideration of relations between vertices, including "in-coming"/"out-going" or multiple types of relationship, is beneficial to the graph representation learning. All in all, HetETA achieves obvious performance gain by embedding various heterogeneous information, validating the significance of spatiotemporal heterogeneous information embedding.

4.3.3 Cooperation with WDR. In this experiment, we want to check whether HetETA can help to improve the performance of some state-of-the-art ETA frameworks applied in the industry, like WDR [32]. We first output the dynamic representations from the HetETA models trained for ETA task in Section 4.3.1. Then, the learned representations are used as additional input features for the Recurrent Channel of WDR model. Together with hand-crafted features designed by WDR, we train the WDR models and report the results in Table 5. As we can observe, with the incorporation of representations learned by HetETA, WDR+HetETA achieves better performances by a large margin (decreased by 1.19% ~ 1.94%, 1.57% ~ 5.30%, 1.67% ~ 6.42% and 3.33% ~ 18.50% in terms of MAPE, MAE, RMSE and BCR, respectively), which further proves the power of HetETA in industrial applications. Here again, many efforts have been made to improve performance of ETA task. Therefore, the superior performance of HetETA indicates tremendous potential of heterogeneous network embedding for ETA tasks.

4.3.4 Quantitative Analysis. To analyze the impact of the length of path route, we present the MAPE results of HetETA over the



(a) MAPE vs. Length of the path route (b) MAPE vs. Time of day

Figure 9: The trend of MAPE on SY_6.

length of path route in Figure 9(a). In general, HetETA performs better when the path route is longer, verifying the ability of our model to capture long-term patterns. Furthermore, we studied the performance of HetETA over different time periods in the day. As shown in Figure 9(b), HetETA performs better during off-peak hours compared to peak hours. This is because of the well-known intractability problem of traffic congestion in peak hours, which is worthy of exploration in future studies.

5 CONCLUSION

In this paper, we propose HetETA to learn the representation of spatiotemporal heterogeneous information networks for travel time estimation. HetETA combines gated convolution neural networks and graph neural networks to capture the correlations in spatiotemporal information. To tackle the different types of relationships among vertices, we design an attention-based Het-ChebNet and construct a double-stuffed graph convolution layer to embed our induced networks, including the multi-relational road network and vehicle-trajectories based network. Comprehensive empirical studies on four large-scale datasets show that HetETA achieves state-of-the-art results and demonstrate the effectiveness of the heterogeneous information network embedding in ETA task. We plan to apply other heterogeneous information, such as various types of vertices, to ETA task in the future work.

REFERENCES

[1] George M Beal and Joe M Bohlen. 1956. *The diffusion process.* Technical Report.

[2] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral Networks and Locally Connected Networks on Graphs. In *2nd International Conference on Learning Representations.*

[3] Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. 2007. Freeway Performance Measurement System: Mining Loop Detector Data. *Transportation Research Record Journal of the Transportation Research Board* (2007).

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[5] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning.* 933–941.

[6] Corrado De Fabritiis, Roberto Ragona, and Gaetano Valenti. 2008. Traffic estimation and prediction based on real time floating car data. In *2008 11th International IEEE Conference on Intelligent Transportation Systems.* IEEE, 197–203.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems.* 3844–3852.

[8] Jan-Willem Grotenhuis, Bart W Wiegmans, and Piet Rietveld. 2007. The desired quality of integrated multimodal travel information in public transport: Customer needs for time and effort savings. *Transport Policy* 14, 1 (2007), 27–38.

[9] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

[10] Anthony Harrington and Vinny Cahill. 2004. Route profiling: putting context to work. In *Proceedings of the 2004 ACM symposium on Applied computing.* ACM.

[11] Ryan Herring, Aude Hofleitner, Saurabh Amin, T Nasr, A Khalek, Pieter Abbeel, and Alexandre Bayen. 2010. Using mobile phones to forecast arterial traffic through statistical learning. In *Transportation Research Board Annual Meeting.*

[12] Aude Hofleitner and Alexandre Bayen. 2011. Optimal decomposition of travel times measured by probe vehicles using a statistical traffic flow model. In *14th International IEEE Conference on Intelligent Transportation Systems (ITSC).* IEEE.

[13] Aude Hofleitner, Ryan Herring, Pieter Abbeel, and Alexandre Bayen. 2012. Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network. *IEEE Transactions on Intelligent Transportation Systems* (2012).

[14] H. V. Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M. Patel, Raghu Ramakrishnan, and Cyrus Shahabi. 2014. Big data and its technical challenges. *Communications of the Acm* 57, 7 (2014), 86–94.

[15] Erik Jenelius and Haris N Koutsopoulos. 2013. Travel time estimation for urban road networks using low frequency probe vehicle data. *Transportation Research Part B: Methodological* 53 (2013), 64–81.

[16] Ishan Jindal, Tony Qin, Xuewen Chen, Matthew S. Nokleby, and Jieping Ye. 2017. A unified neural network approach for estimating travel time and distance for a taxi trip. *arXiv preprint arXiv:1710.04350* (2017).

[17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR, 2017.*

[18] Yaguang Li, Dingxiong Deng, Ugur Demiryurek, Cyrus Shahabi, and Siva Ravada. 2015. Towards fast and accurate solutions to vehicle routing in a large-scale and dynamic environment. In *International Symposium on Spatial and Temporal Databases.* Springer, 119–136.

[19] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. 2018. Multi-task representation learning for travel time estimation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* ACM, 1695–1704.

[20] Yanying Li and Mike McDonald. 2002. Link travel time estimation using single GPS equipped probe vehicle. In *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems.* IEEE, 932–937.

[21] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations (ICLR '18).*

[22] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, Vol. 30. 3.

[23] Santa Maiti, Arpan Pal, Arindam Pal, Tanushyam Chattopadhyay, and Arijit Mukherjee. 2014. Historical data based real time prediction of vehicle arrival time. In *17th International IEEE Conference on Intelligent Transportation Systems.*

[24] John Miller and Moritz Hardt. 2018. When recurrent models don't need to be recurrent. *arXiv preprint arXiv:1805.10369* 4 (2018).

[25] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv:1609.03499* (2016).

[26] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. 2018. On the convergence of adam and beyond. In *6th International Conference on Learning Representations.*

[27] Raffi Sevlian and Ram Rajagopal. 2010. Travel time estimation using floating car data. *arXiv preprint arXiv:1012.4249* (2010).

[28] Yizhou Sun and Jiawei Han. 2012. Mining heterogeneous information networks: a structural analysis approach. *SIGKDD Explorations* 14, 2 (2012), 20–28.

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations* (2018).

[30] Hongjian Wang, Xianfeng Tang, Yu-Hsuan Kuo, Daniel Kifer, and Zhenhui Li. 2019. A simple baseline for travel time estimation using large-scale trip data. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 19.

[31] Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 25–34.

[32] Zheng Wang, Kun Fu, and Jieping Ye. 2018. Learning to estimate the travel time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* ACM, 858–866.

[33] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI).*

[34] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence.*

[35] Nicholas Jing Yuan, Yu Zheng, Liuhang Zhang, and Xing Xie. 2012. T-finder: A recommender system for finding passengers and vacant taxis. *IEEE Transactions on knowledge and data engineering* 25, 10 (2012), 2390–2403.

[36] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. GMAN: A Graph Multi-Attention Network for Traffic Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34.

A MORE DETAILS ON HET-CHEBNET

Traced back to spectral graph convolutional neural networks [2], it first defines convolution on graph data in the context of spectral graph theory, taking the form as:

$$\mathbf{x}' = \mathbf{Q}\mathbf{\Theta}\mathbf{Q}^{\mathrm{T}}\mathbf{x}, \tag{11}$$

where $\mathbf{x}', \mathbf{x} \in \mathbb{R}^{|V|}$ are the output signal and input signal on vertices (a scalar for each vertex). $\mathbf{Q}$ comprises eigenvectors of graph Laplacian matrix $\mathbf{L}$, and the diagonal matrix $\mathbf{\Theta} = \mathbf{\Theta}(\mathbf{\Lambda}) \in \mathbb{R}^{|V|\times|V|}$ is learnable filters, where $\mathbf{\Lambda}$ denotes eigenvalues of $\mathbf{L}$. However, the calculation of full eigenvectors is time-consuming for large-scale graphs. To reduce the computational complexity, ChebNet [7] proposes a polynomial filter to approximate the spectral filter by Chebyshev expansion:

$$\mathbf{\Theta}(\mathbf{\Lambda}) = \sum_{z=0}^{Z} \theta_z \Gamma_z(\tilde{\mathbf{\Lambda}}), \tag{12}$$

where eigenvalues are rescaled as $\tilde{\mathbf{\Lambda}} = 2\mathbf{\Lambda}/\lambda_{\max} - \mathbf{I}$ for the orthonormal basis of Chebyshev polynomials. $\Gamma_z(\cdot)$ is the Chebyshev polynomial of order z with recursive relation:

$$\Gamma_z(x) = 2x\Gamma_{z-1}(x) - \Gamma_{z-2}(x). \tag{13}$$

Substitute Eq. (12) into Eq. (11), we obtain Eq. (14) as follows:

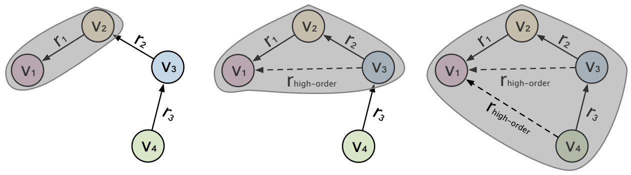
$$\mathbf{x}' = \mathbf{Q}\mathbf{\Theta}(\mathbf{\Lambda})\mathbf{Q}^{\mathrm{T}}\mathbf{x} = \sum_{z=0}^{Z} \theta_z \mathbf{Q}\Gamma_z(\tilde{\mathbf{\Lambda}})\mathbf{Q}^{\mathrm{T}}\mathbf{x} = \sum_{z=0}^{Z} \theta_z \Gamma_z(\tilde{\mathbf{L}})\mathbf{x}, \tag{14}$$

where $\tilde{\mathbf{L}} = 2\mathbf{L}/\lambda_{\max} - \mathbf{I}$ and $\mathbf{L}^z = \mathbf{Q}\mathbf{\Lambda}^z\mathbf{Q}^{\mathrm{T}}$.

Het-ChebNet. In this paper, we generalize ChebNet to the multi-relational graph/network by incorporating attention mechanism into Eq. (14):

$$\mathbf{x}' = \sum_{z=0}^{Z} \theta_z \left(\Gamma_z(\tilde{\mathbf{L}}) \odot f(\mathbf{U}_z)\right) \mathbf{x}. \tag{15}$$

where $\mathbf{U}_z$ is an attention matrix for each pair of vertices within the z-hop connection of the graph. That is, when the connection weight from v_i to v_j in $\Gamma_z(\tilde{\mathbf{L}})$ is not zero, Het-ChebNet computes an attention score for this connection according to its relation type. In addition to the original relation types in the graph, we add "high-order" relationship for high-order adjacency linkages (see Figure 10). When the attention matrix $\mathbf{U}_z$ is obtained, the softmax function $f(\cdot)$ would be performed on the last dimension of $\mathbf{U}_z$. Figure 11 depicts the computational process of a value $f(u_{ji}) \in f(\mathbf{U}_z)$.



(a) 1-localized Het-ChebNet (b) 2-localized Het-ChebNet (c) 3-localized Het-ChebNet

Figure 10: Toy example of convolutions for target vertex v_1 in Z-localized Het-ChebNet. Vertices within the gray shadow are convolved to compute the output signal of v_1.

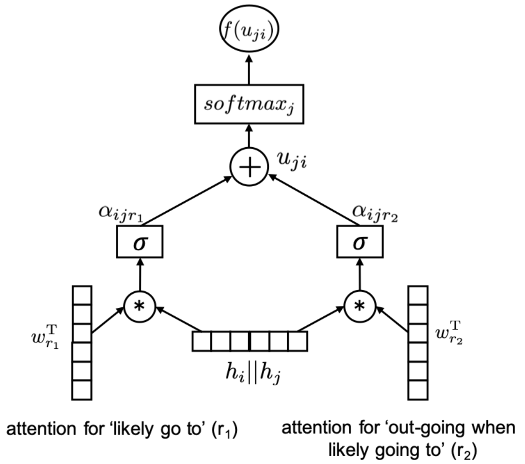


Figure 11: Attention mechanism employed by Het-ChebNet.

B DETAILED PARAMETER SETTINGS ON BASELINES

All comparison models are trained based on the best hyper-parameters chosen by using grid search until convergence within 100 epochs. In this section, we list the detailed hyper-parameters for comparison methods mentioned in our experiments. Note that, we use the same settings for either ETA task or traffic prediction task.

- **GRU [4]**: Gated Recurrent Unit Network. We apply multi-cell RNNs with 2 GRU layers. Each GRU layer contains 11 units. The initial learning rate is set to 0.02 and reduced by 15% every 5 epochs. The max training epoch number is 100 and early stopping on the validation dataset is used.
- **DCRNN [21]**: Diffusion Convolutional Recurrent Neural Network, whose implementation is available in *https://github.com/liyaguang/DCRNN*. Seq2seq architecture is used, in which the encoder accepts sequence of length= 4 and the decoder outputs sequence of length= 1. The maximum steps of diffusion convolutions, i.e., K, is set to 2. Each recurrent layer in DCRNN contains 11 units. The initial learning rate is set to 0.05 and reduced by 15% every 5 epochs. The max training epoch number is 100 and early stopping on the validation dataset is used.
- **STGCN [34]**: Spatio-Temporal Graph Convolutional Network, whose implementation is available in *https://github.com/VeritasYin/STGCN_IJCAI-18*. The channels of three hidden layers in ST-Conv block are [9, 8, 11]. Chebyshev polynomials kernels are applied with $Ks = 3$, and kernel size of temporal convolution is $Kt = 2$. The initial learning rate is set to 0.05 and reduced by 15% every 5 epochs. The max training epoch number is 100 and early stopping on the validation dataset is used.
- **Graph WaveNet [33]**: Graph WaveNet, which is originally implemented in PyTorch available at *https://github.com/nnzhan/Graph-WaveNet*. We re-implement it in Tensorflow and due to the large number of vertices in our network, we remove

the self-adaptive adjacency matrix. In the dilated casual convolutions, the kernel size is set to 2 and the dilation factors are set to 1 and 2 for the first and second layer respectively. The maximum diffusion step number is 2, same as DCRNN. The dimension of output layer is also set to 11. The initial learning rate is set to 0.02 and reduced by 15% every 5 epochs. The max training epoch number is 100 and early stopping on the validation dataset is used.

- **ASTGCN [9]**: Attention Based Spatial-Temporal Graph Convolutional Network, which is originally implemented in MxNet available at *https://github.com/Davidham3/ASTGCN*. We re-implement it in Tensorflow. Note that, the attention matrix $S \in \mathbb{R}^{|V| \times |V|}$ in ASTGCN obtained from the spatial attention mechanism is based on all the nodes. It causes OOM (Out of Memory) error in Shenyang's road network. Thus we replace it by the sparse spatial attention layer in GAT [29], whose code is available at *https://github.com/PetarV-/GAT*. Same as HetETA, the sequences length of three components (recent, daily, weekly) are $L_{\mathcal{R}} = L_{\mathcal{D}} = L_{\mathcal{W}} = 4$. Chebyshev polynomials kernels are applied with $Ks = 3$, and kernel size of temporal convolution is $Kt = 2$. The output dimension of each component is set to 8. Similarly, the initial learning rate is set to 0.05 and reduced by 15% every 5 epochs. The max training epoch number is 100 and early stopping on the validation dataset is used.