Learning Traffic Network Embeddings for Predicting Congestion Propagation

Yidan Sun, Guiyuan Jiang, Siew-Kei Lam, Peilan He

***Abstract*—Traffic congestion has become a global concern due to continuous increase in traffic demand and limited road capacity. The ability to predict traffic congestion propagation, which depicts the spatiotemporal evolution of the congestion scenario, is essential for developing smart traffic management systems and enabling road users to make informed route choices. In this work, we study the behavior of congestion propagation at the road segment level, and leverage this to develop a novel machine learning framework that characterizes and predicts the congestion evolution among different road segments in the traffic network. In particular, our framework can infer the likelihood of congestion propagation between any pair of road segments through single or multiple propagation paths. The proposed framework relies on a network embedding module to learn a representation for each road segment, and a propagation model which calculates the congestion propagation likelihood based on the learned representations. Specifically, an asymmetric embedding of local proximity and global tendency (AE-LPGT) is relied upon for learning low dimension embeddings of the road segments which incorporate various realistic properties of congestion propagations, such as the local proximity property, global propagation tendency, and asymmetric transitivity of congestion propagations. Experimental results with Singapore traffic data show that our method significantly outperforms the state-of-the-art, and the congestion propagation properties in our embeddings have significant impact on the prediction performance.**

***Index Terms*—Congestion propagation, network embedding, local proximity, global propagation tendency, propagation model.**

I. Introduction

Traffic congestions, especially in urban areas, is becoming increasingly severe due to the rapid growth in the number of vehicles and travel demands. This adversely impacts the quality of life and economic productivity in metropolitan areas [1], [2] due to the increase in fuel consumption, travel cost, road accidents, and carbon emissions. Generally, traffic congestions exhibit specific propagation patterns based on their spatiotemporal significance [3]. For example, different regions of a large urban road network experience distinctive traffic flow patterns depending on the significance of venues (e.g., shopping mall, school, stadium, hospital, etc.) in the region. In addition, the occurrence of congestions is associated with evolving travel demands. Typically, congestion emerges on road segments where the traffic flow exceeds the road capacity. If this persists, the congested road segments are likely to affect its neighboring road segments. If the traffic flow continues to increase (e.g., during daily peak hours), the congestion propagation may exhibit a domino effect where the congestions propagate to other spatially connected road segments. This will last until the travel demand diminishes, and a reverse propagation effect can be observed.

Researches in traffic congestion propagation have been studied at various levels. At the *macroscopic* level, congestion are represented as spatially connected road clusters. Congestion propagation are determined based on the changes in shape/size/location of the clusters instead of the individual road segments [4], [5]. Most of the research make efforts to develop efficient pattern mining algorithms, or to partition the traffic network into sub-networks (clusters) and track the dynamic changes of the clusters [5], [6]. Research at the *microscopic* level considers congestion propagation within one long road segment. Typically, a long road is partitioned into several sections, and methods [7], [8] were proposed to characterize the congestion propagation from downstream to upstream sections.

The macroscopic level models consider congestion propagations at a coarse granularity (i.e., a cluster of similar roads), while microscopic level models consider propagations among subsections within an individual road. Existing works focusing on the macroscopic and microscopic levels fail to model congestion evolution behaviors among different roads at road-level granularity (i.e., mesoscopic level). Accurate prediction of congestion propagation at mesoscopic level is essential for many downstream applications such as road-level traffic prediction [3], bottleneck identification [9], and traffic signal scheduling [10]. In addition, road users of private vehicles and public transport can make effective route decisions if they are provided with near-future road congestion predictions.

Many works [3], [11] have investigated congestion propagation at the *mesoscopic* level, which considers congestion propagation between adjacent different road segments. Existing works mainly focused on mining propagation patterns using causality trees/graphs [3], [9], [11], which are then used to predict potential propagation. As such, the quality of predictions is limited to the available patterns mined. Also, these propagation patterns fail to sufficiently capture the following realistic characteristics of true congestion propagation. Firstly, congestion propagates in an asymmetric manner (i.e.,

This research project is supported in part by the National Research Foundation Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme with the Technical University of Munich at TUMCREATE. *(G. Jiang is the corresponding author)*

Y. Sun, G. Jiang, S.K. Lam, P. He are with the School of Computer Science and Engineering, Nanyang Technological University, 639798, Singapore. (e-mail: {ysun014,phe002}@e.ntu.edu.sg, {gyjiang,assklam}@ntu.edu.sg.

Manuscript received November 15, 2020; revised May 11, 2021; accepted August 13, 2021.

the asymmetric transitivity of congestion propagation). This means the probability that road segment r_i's congestion states will propagate to $r_{i'}$ is different from the probability that $r_{i'}$'s congestion states will propagate to r_i. Secondly, the traffic state of a road segment is highly correlated with the traffic states of its local neighborhood [12] (i.e., local proximity). Also, a congestion propagation could stretch across a large area over the entire traffic network (i.e., global tendency), rather than be limited to a local area. Local proximity and global tendency reflect different ranges and granularity of the congestion propagations in the spatial dimension. The existing works have not sufficiently considered the above-mentioned properties, and hence they fail to produce good prediction performance under complex traffic conditions.

In this paper, we focus on congestion propagation at the mesoscopic level. We propose novel machine learning methods to characterize and predict congestion propagations while taking into account various realistic properties of congestion propagations as mentioned above, which have not been sufficiently considered in existing works. The main contributions of our work are summarized as follows.

- We propose a machine learning based framework to characterize and predict congestion propagations. This departs from existing methods that mostly rely on causality trees/graphs and mining propagation patterns. Our framework incorporates a novel network embedding method to learn representations of the road segments while characterizing various realistic properties of congestion propagations. With the learned representations, the proposed framework can infer the likelihood of congestion propagation between any pair of road segments that are connected through single or multiple propagation paths.
- We propose a deep learning based embedding method, called asymmetric embedding of local proximity and global tendency (AE-LPGT), to learn low dimension embedding of each road segment. AE-LPGT incorporates the local proximity property (spatiotemporal similarity in local neighborhood), and the global propagation tendency (dynamic congestion propagation tendency over the entire network). AE-LPGT also considers the asymmetric transitivity of congestion propagations by learning source embedding and target embedding separately. Moreover, the AE-LPGT model can capture the dynamic and temporal dependencies of multi-range congestion propagations in the traffic network.
- We develop a propagation model that can efficiently predict the congestion propagation between a source and target road segment. The propagation model infers the likelihood that the congestion associated with the source road segment will propagate to the target road segment through all the paths connecting them. We validate the proposed network embedding and the propagation model using real traffic data, and the results show that our method significantly outperforms the state-of-the-art.

II. Related Work

1) Macroscopic Level: Researches on congestion propagation at macroscopic level typically study the congestion behavior of a group of road segments or a region. Congestion is represented as a cluster of roads that are spatially connected and have the same congestion states. Congestion propagation is defined as the change in the shape/size of a congestion cluster. Existing works formulated road clusters by partitioning the traffic network into sub-networks [4], [5] or grids [13], [14]. Techniques have also been proposed to discover the congestion propagation patterns [15]–[17], e.g., the spatio-temporal co-location patterns.

Macroscopic level propagation models typically explain the traffic congestions as a series of events, where each event involves several spatially connected roads and stretches across multiple time slices. The related works mainly focus on mining the propagation patterns rather than predicting the propagations on large scale traffic networks. In practice, the congestion clusters are time-dependent and irregular in size, shape, duration, etc., which make accurate prediction difficult. These coarse-granularity models have tens or hundreds of roads in a cluster, which are not suitable for applications that require congestion propagation prediction of specific roads.

2) Microscopic level: Many works focused on microscopic level congestion propagations, in which a long road is partitioned into several parts (sections). The propagation considered occurs among the sections without affecting the neighboring roads. As such only downstream and upstream congestions are considered. Typically, vehicles accelerate downstream, while upstream vehicles decelerate when approaching the congestion. The work in [7] studied traffic flow diagrams at an off-ramp of a highway, which reveals congestion propagation patterns at diverge bottlenecks. [18] used traffic speed time series collected on a freeway to plot spatiotemporal images, and then extracted properties such as bottleneck strength, congestion propagation velocity, growth rate, and wave-length. These features are used to calibrate the traffic flow models of various simulators. [19] studied the highway section using time-space diagram as inputs, and utilized deep neural network with an encoder-decoder structure to predict the propagation of shock wave in the form of time-space diagram. The work in [8] studied the congestion evolution behavior within a long road, where five levels of traffic states are considered: very smooth, basic smooth, mild congestion, moderate congestions, and heavy congestion. A spatiotemporal heat diagram is used for detecting congestion evolution. Studies on microscopic level congestion propagations are limited to individual roads, and these methods cannot be extended to a traffic network with more complex traffic states.

3) Mesoscopic Level: Researches at the mesoscopic level consider congestion propagations occuring between the macroscopic and microscopic levels. Specifically, traffic congestion is associated with individual road segments, and congestion propagation occurs between adjacent road segments. The work in [20] proposed a spatiotemporal co-location congestion pattern mining method to discover an orderly set of roads with congestion propagation in urban traffic, which reveals the process of congestion propagation and uncovers the main propagation paths that lead to large-scale congestion. The work in [21] identified the locations of traffic bottlenecks by mining congestion propagations. It built congestion prop-

agation graphs to model congestion propagations, constructed maximal spanning trees to model the causal relationship, and utilized the Markov analysis to determine the probabilities of congestion propagations, based on which bottlenecks are identified. The work in [3] formulated congestion propagation between road segments as a direct edge. Then, causality trees are constructed, and propagation patterns are discovered as frequent subtree structures using association rule mining. Based on the obtained patterns, a dynamic Bayesian network is used to predict the likelihood of congestion propagation between any two adjacent road segments in the frequent subtree structures. This work was further extended in [11] to build directed acyclic graphs (DAG) based on causality trees. A spatial matrix that incorporates the DAG into a fine granularity grid map (i.e., each grid contains at most one road) is fed into a Convolutional Neural Network (CNN) to predict congestions. The work in [22] classified congestion levels of each road using fuzzy clustering and divided the traffic network into clusters using mixed integer linear programming optimization. The congestion propagation between roads within a cluster is predicted using a dynamic Bayesian network. This method has high computation overhead as it needs to perform the clustering for each time window. Also, it fails to consider the temporal dependency across different time windows. The work in [9] also described congestion propagation as causality trees and identified bottlenecks in urban traffic network by mining road segments that lead to congestion propagation. There are also simulation-based methods [23], [24] that study the traffic congestion evolution (emergence, disappearance, propagation, etc.). However, these methods rely on strong traffic assumptions (e.g., road capacity, density, number of vehicles, traffic demand, etc.), which could differ significantly from real traffic scenarios.

In general, the above works typically focus on discovering congestion propagation patterns on network structure (e.g., tree, graph. etc.) by mining frequent congested substructure using historical datasets. While some also consider congestion propagation prediction, they are limited to specific types of propagation patterns that are defined during the pattern mining stage. In this work, we also focus on congestion propagation at mesoscopic level and propose novel embedding methods for accurate prediction of congestion propagation between any road pairs in the traffic network.

A. Network embedding

Network embedding techniques have been widely investigated in various application scenarios, such as social networks, information networks, and transportation networks. Many methods have been developed for network embedding [25], i.e., learning representations of networks to support network reconstruction and network inference (e.g., link prediction, important node identification, etc.). The first-order and second-order structures [26] are usually preserved when learning the network embeddings, and high-order structure has also been considered in [27], where a method named DeepWalk was developed to preserve the neighboring structures of nodes. The work in [28] proposed a community preserving embedding method to preserve not only the first-order and second-order proximities but also the community structure (communities have dense intra-connections but sparse inter-connections) by making use of modularized non-negative matrix factorization model. The above methods typically assume symmetric proximity between nodes, which is not applicable to traffic networks. For example, traffic congestion propagates from downstream to upstream, but not the other way. These methods also typically focus on local proximity while neglecting the global tendency of congestion propagations as well as the evolving correlations of the road segments.

Some works have considered preserving asymmetric transitivity in network embedding for network inference. For example, the work in [29] proposed a method for graph embedding while preserving asymmetric transitivity, i.e., a source embedding and a target embedding are produced for each network node. For a directed edge from node i to node j, the source embedding of node i should be located closely with the target embedding of node j in the latent embedding space. They applied generalized Singular Value Decomposition (SVD) method [30] to obtain embedding by preserving high-order proximity such as Katz Index [31], Adamic-Adar [32], etc. In this paper, we also consider the asymmetric characteristics of traffic congestion propagations. Unlike previous methods, we combine the NMF method and deep neural networks to achieve efficient embedding learning while simultaneously incorporating both local proximity and global propagation tendencies.

On the other hand, network embedding techniques have been applied to various traffic prediction problems, such as travel time estimation [33], traffic flow prediction [34], traffic speed forecasting [35], etc. Network embedding techniques are developed to learn low-dimension vector representation of network nodes while incorporating various network information such as node attributes, local and global road network structure, etc. However, the existing works for traffic prediction typically focus on learning representations for individual nodes rather than the associations between nodes (i.e., links). In this paper, we investigate the propagation of congestions among different roads, which requires new embedding techniques to incorporate the unique characteristics of congestion propagations.

III. Preliminaries

The road network is represented as a graph $G = (V, E)$, where node set $V = \{r_1, ...r_N\}$ is a set of N road segments (connected by junctions in the road network), $E = \{< r_i, r_{i'} >\}$ is a set of edges, and $< r_i, r_{i'} >$ indicates a spatial connection between road segments r_i and $r_{i'}$. Each road segment $r_i \in V$ is associated with a speed time series $\hat{\mathbf{s}}_i \in \mathbb{R}^T$, where T is the number of time slices during the entire time period. The traffic speeds of all road segments in G over the T time slices are represented as a traffic speed matrix $\hat{\mathbf{S}} \in \mathbb{R}^{N \times T}$, where $\hat{s}_i^t$ represents the average traffic speed of road segment r_i during time slice t. c_i^t is a binary variable such that $c_i^t = 1$ indicates that road segment r_i is congested during time slice t, while $c_i^t = 0$ indicates that r_i

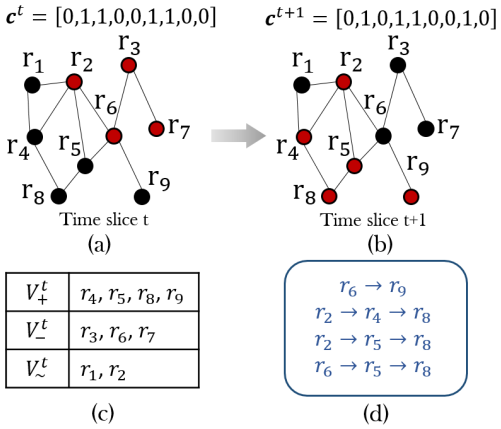


Fig. 1: (a) and (b) show G_d^t and G_d^{t+1} where red nodes are congested roads and black nodes represent congestion-free roads; (c) Sets V_+^t, V_-^t and $V_\sim^t$ of road segments whose traffic state changed from non-congested to congested, from congested to non-congested, and is unchanged, respectively. (d) Congestion propagation paths from time slices t to $t+1$.

is congestion free. Thus, $\mathbf{c}^t \in \mathbb{R}^N$ is a vector associated with each time slice, consisting of the congestion states of all road segments at time slice t.

Let $G_d^t = (V, E, \mathbf{c}^t)$ be a dynamic traffic graph that describes traffic states at time slice t. Thus, the spatiotemporal evolution of congestions over the entire traffic network can be characterized by a series of dynamic traffic graphs $G_d = \{G_d^t | G_d^t = (V, E, \mathbf{c}^t)\}$. Fig. 1 (a) and (b) show two examples of dynamic traffic graphs, i.e., G_d^t and G_d^{t+1} which correspond to time slices t and $t+1$, respectively. Based on the traffic states in G_d^t and G_d^{t+1}, we divide all road segments into 3 disjoint sets: V_+^t, V_-^t and $V_\sim^t$. Set V_+^t consists of road segments r_i whose traffic state changed from congestion-free to congested, i.e.,

$$V_+^t = \{r_i | c_i^t = 0 \wedge c_i^{t+1} = 1\} \tag{1}$$

Similarly, V_-^t consists of road segments whose traffic state changed from congested to congestion-free, i.e.,

$$V_-^t = \{r_i | c_i^t = 1 \wedge c_i^{t+1} = 0\} \tag{2}$$

$V_\sim^t$ consists of road segments whose traffic state have not changed, i.e.,

$$V_\sim^t = \{r_i | (c_i^t = 1 \wedge c_i^{t+1} = 1) \vee (c_i^t = 0 \wedge c_i^{t+1} = 0)\} \tag{3}$$

As shown in Fig. 1 (c), based on the dynamic traffic graphs shown in (a) and (b), $V_+^t = \{r_4, r_5, r_8, r_9\}$, $V_-^t = \{r_3, r_6, r_7\}$, $V_\sim^t = \{r_1, r_2\}$.

Congestion propagation behaviors are characterized by propagation paths that can be extracted from traffic graph G_d. Let $pa^t = \{pa_1^t, pa_2^t, \cdots, pa_k^t, \cdots\}$ be a set of propagation paths from time t to $t+1$. A sequence of connected road segments, $pa_k^t =< r_{i_1} \rightarrow r_{i_2} \rightarrow \cdots \rightarrow r_{i_u} \rightarrow \cdots >$, forms a propagation path if it satisfies the following conditions simultaneously: 1) $c_{i_1}^t = 1$, i.e., the road r_{i_1} is source, which is congested at time slice t; 2) $r_{i_u} \in V_+^t$ for $u = 2, 3, \cdots$, i.e., other roads are targets whose states become congested at time t+1 from non-congested state at time t; 3) pa_k^t cannot be further expanded by adding new roads at the end. If a propagation path consists of only two road segments, e.g., $pa_k^t =< r_{i_1} \rightarrow r_{i_2} >$ such that r_{i_1} propagates its congestion state to r_{i_2}, it is called a 1-hop congestion propagation. On the other hand, if $pa_k^t =< r_{i_1} \rightarrow r_{i_2} \rightarrow \cdots >$ contains more than two road segments, it is called a multi-hop congestion propagation, as the congestion state of r_{i_1} propagates to multiple road segments along the propagation path within a time slice. As shown in Fig. 1 (d), there is a single one-hop propagation path (i.e., $\langle r_6 \rightarrow r_9 \rangle$), and three multi-hop propagation paths (i.e., $\langle r_2 \rightarrow r_4 \rightarrow r_8 \rangle$, $\langle r_2 \rightarrow r_5 \rightarrow r_8 \rangle$ and $\langle r_6 \rightarrow r_5 \rightarrow r_8 \rangle$).

Congestion Propagation Prediction: Given a road network $G = (V, E)$, a set of historical traffic graphs G_d, and a pair of source and target road segment (i.e., r_i and $r_{i'}$), predict the likelihood that the congestion will propagate from r_i to $r_{i'}$ via the paths between them in the next H time slices. There may be multiple paths between the source and target road segments, which are likely to have different propagation speed and significance. For example, as shown in Fig. 1, r_2 to r_8 are connected by two distinct paths $pa_1 = \langle r_2 \rightarrow r_4 \rightarrow r_8 \rangle$ and $pa_2 = \langle r_2 \rightarrow r_5 \rightarrow r_8 \rangle$. If a congestion emerges on r_2, its propagation to r_8 could manifest through either r_4 or r_5 or both. Thus, the prediction model needs to consider multiple paths and their influence on each other.

IV. Proposed Method

Our method learns a representation for each road segment that reflects the propagation tendencies between different roads (for both 1-hop and multi-hop propagation). Two road segments with frequent propagations will have similar representations, while road segments with independent traffic states will have largely dissimilar representations. In the following, we first introduce the main framework for addressing the congestion propagation problem, then we describe the main components, i.e., the embedding model and the propagation model in Section V and Section VI, respectively.

Fig. 2 shows the proposed framework for traffic congestion propagation prediction, which consists of three major modules.

1) Detect Historical Congestion Propagations: Traffic congestions are first identified from historical traffic data. Dynamic traffic graphs are then constructed to extract the historical congestion propagation paths based on the dynamic traffic graphs defined in the previous section.

2) Learning Asymmetric Embedding of Local Proximity and Global Tendency (AE-LPGT): The propagation paths from the previous stage are used to learn embedding vectors ($\mathbf{e}_i^{src}$ and $\mathbf{e}_i^{trg}$) for each road segment r_i, where $\mathbf{e}_i^{src}$ is called source embedding and $\mathbf{e}_i^{trg}$ is called target embedding. The learned representations can reflect various propagation characteristics. The proposed AE-LPGT consists of three components: i) *Local Proximity Embedding* (LPE) incorporates the spatiotemporal correlations of traffic states in local neighborhoods into embedding. This ensures that road segments associated with a congestion propagation have similar representation

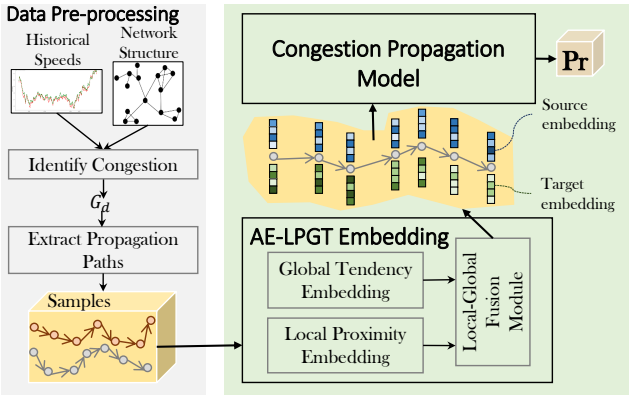


Fig. 2: Proposed framework for modeling congestion propagations.

(smaller distance in embedding space). In other words, this component preserves the same proximity of road segments in embedding space as that in the original space. ii) *Global Tendency Embedding* (GTE) captures the dynamic tendency of congestion propagations over the entire traffic network. GTE differs from LPE in that it focuses on the congestion propagation behavior over a large-scale traffic network rather than a local area. iii) *Local-Global Fusion* component fuses the LTE and GTE using an attention mechanism. Both the LPE and GTE components consider the sources and targets of the propagations separately in order to preserve asymmetric transitivity of congestion propagations.

3) Modeling Congestion Propagations: The congestion propagations are modeled in the form of propagation paths. Specifically, given a congested road segment r_i and a target road segment $r_{i'}$, the propagation model calculates the probability that the congestion associated with road segment r_i will propagate to the target road segment $r_{i'}$ through all the paths connecting them. For example, as shown in Fig. 1, r_2 and r_8 are connected through two different paths $pa_1 = \langle r_2 \rightarrow r_4 \rightarrow r_8 \rangle$ and $pa_2 = \langle r_2 \rightarrow r_5 \rightarrow r_8 \rangle$. The proposed propagation model takes into account the heterogeneous influence of both pa_1 and pa_2.

The details for data pre-processing (including both congestion detection and propagation path extraction) are discussed in Section VII-A. In the following, we introduce the second and third modules, i.e., representation learning via asymmetric embedding and modeling congestion propagations.

V. Representation Learning via Asymmetric Embedding

Network embeddings have been used in previous works [25] where low dimension representation for each network node is learned while preserving network topological and structure information, i.e., nodes with links (or have strong correlations in traffic states) in original network should locate relatively close to one another in the embedding space. These works typically assume that the embedding is symmetric, i.e., $dist(\mathbf{e}_i, \mathbf{e}_{i'}) = dist(\mathbf{e}_{i'}, \mathbf{e}_i)$ (the distance from $\mathbf{e}_i$ to $\mathbf{e}_{i'}$ equals to the distance from $\mathbf{e}_i$ to $\mathbf{e}_{i'}$), where $\mathbf{e}_i \in \mathbf{R}^d$ is $d-$ dimension vector after embeddding for node r_i. However, traffic congestion propagations are often asymmetric, i.e., the probability that road segment r_i's congestion states will propagate to $r_{i'}$ is different from the probability that $r_{i'}$'s congestion states will propagate to r_i. It is possible that r_i is always the source of $r_{i'}$'s congestion between the two road segments. To incorporate this asymmetrical property of congestion propagation, we generate the source embedding $\mathbf{e}_{i,t}^{src} \in \mathbb{R}^{K_1}$ and target embedding $\mathbf{e}_{i,t}^{trg} \in \mathbb{R}^{K_1}$ separately, for each road segment r_i at time slice t. The representation $\mathbf{e}_{i,t}^{src}$ characterizes the likelihood that r_i propagates its congestion states to other road segments; while $\mathbf{e}_{i,t}^{trg}$ captures the tendency that other road segments' congestion states will propagate to r_i. The embeddings also depend on time slice t (using historical data that are related to the time slice t), which enables it to incorporate the dynamic evolution of traffic states.

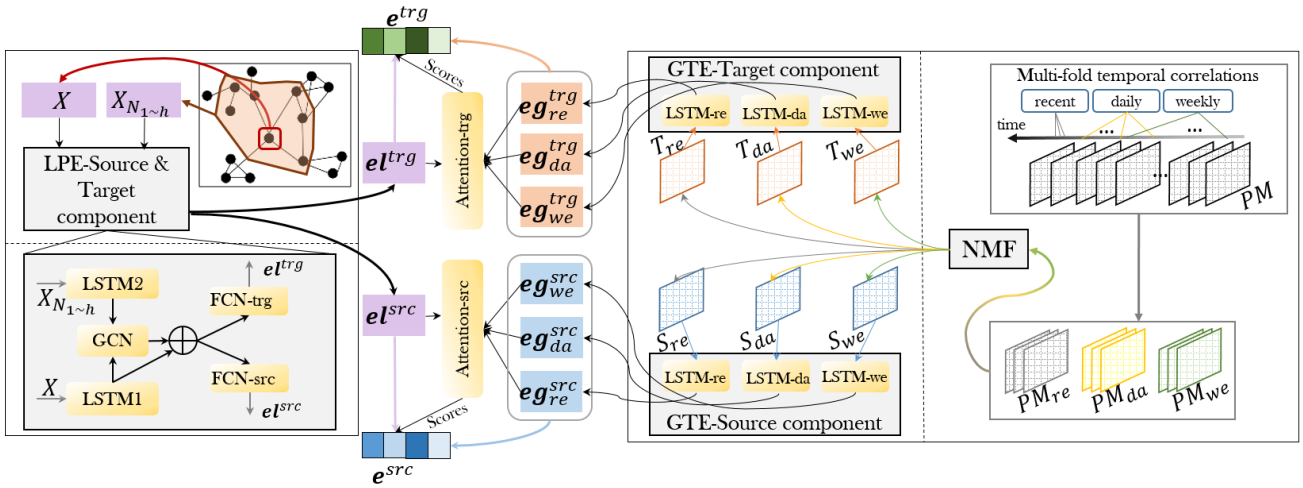
A. Local Proximity Embedding (LPE)

The core idea of network embedding is to learn low dimension vectors that are isomorphic with respect to certain network properties, e.g., proximity, community. In contrast to existing works that consider static networks, the local proximity in our work depends not only on the network structure but also on dynamic traffic states. For example, two road segments with small geographic distance should not be located close to each other in the embedding space if their traffic evolution trends differ significantly. Thus, the local proximity embedding (LPE) aims to capture the evolving local proximity that is affected by both traffic network structure and temporal traffic states. To achieve LPE, a neural network based *LPE-Source/Target* module is constructed, as shown in Fig. 3 (a), which consists of Long Short-Term Memory (LSTM) layers, Graph Convolution Network (GCN) layers, and fully-connected neural network (FCN) based fusion layers. The LPE component first relies on LSTM and GCN layers to extract local information while incorporating spatial correlations. It then simultaneously produces source embedding and target embedding using latent features learned via LSTM and GCN layers. In other words, the source embedding process and the target embedding process share the same network component to aggregate the local neighborhood information while incorporating their correlations (i.e., the LSTM layers and the GCN layers). The source embedding and target embedding are then generated by the FCN layers.

The feature vectors of the road segments in a path are fed into the LPE-Source/Target component sequentially to produce a separate source (and target) embedding for each road segment at time slice t, i.e., $\mathbf{el}_{i,t}^{src} \in \mathbb{R}^{K_1}$ ($\mathbf{el}_{i,t}^{trg} \in \mathbb{R}^{K_1}$) where K_1 is the dimension of the embedding vector. The feature vector of a road segment r_i contains two parts: a feature vector $\mathbf{x}_i^t \in \mathbb{R}^F$ of r_i itself (F is the number of features, t is time slice), and the feature vectors of r_i's up to h hop neighbors, i.e.,

$$\mathbf{X}_{N_{1\sim h}(r_i)}^t = \{\mathbf{x}_{i'}^t | r_{i'} \in N_{1\sim h}(r_i)\} \tag{4}$$

The feature vector $\mathbf{x}_i^t$ includes the following attributes: *speed values* of previous W time slices which indicate recent traffic



(a) Local Proximity (b) Fusion (c) Global Tendency

Fig. 3: Architecture of the AE-LPGT embedding model.

state, *historical daily and weekly average speed* at time slice t (which captures long term periodicity), *time indicator* such as time of day and day of week, *road characteristics* such as road category (e.g., primary road, second primary road, highway), the number of lanes and bus stops along the road. The road category is represented using one-hot encoding.

In the LPE-Source/Target component, the features $\mathbf{x}_i^t$ and $\mathbf{X}_{N_{1\sim h}(r_i)}^t$ are fed into separate LSTM layers to learn temporal dependency among features, as shown in Fig. 3 (a). First, LSTM1 (consists of three LSTM layers) takes $\mathbf{X}_i^t$ as input while LSTM2 (same structure as LSTM1) takes $\mathbf{X}_{N_{1\sim h}(r_i)}^t$ as input. Next, a GCN layer is used to convolve the information of local neighbors of r_i (i.e., $N_{1\sim h}(r_i)$), while respecting the traffic network structure. Then the learned representation of road segment r_i (i.e., the output of LSTM1) and the local neighborhood information (i.e, the output of GCN) are concatenated together as $\mathbf{h}_{local}$. To enable the resultant embeddings to distinguish whether r_i is a source or a target in a propagation process, $\mathbf{h}_{local}$ is fed into a FCN layer (i.e., FCN-src) to produce embedding of the source local proximity, while $(\mathbf{1} - \mathbf{h}_{local})$ is fed into another FCN layer (i.e., FCN-trg) to learn the target local proximity. The outputs of LPE component consist of $\mathbf{el}_{i,t}^{src}$ and $\mathbf{el}_{i,t}^{trg}$, which are local source and target embedding for given road segment r_i at time slice t. The specific structure of the model components of the LPE-Source/Target is as follows.

1) LSTM Structure: The architecture of the LSTM cell is described with the following equations:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{b}_i),\\
\mathbf{f}_t &= \sigma(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fh}\mathbf{h}_{t-1} + \mathbf{b}_f),\\
\mathbf{o}_t &= \sigma(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{b}_o),\\
\widetilde{\mathbf{C}_t} &= \tanh(\mathbf{W}_{Cx}\mathbf{x}_t + \mathbf{W}_{Ch}\mathbf{h}_{t-1} + \mathbf{b}_C),\\
\mathbf{C}_t &= \mathbf{i}_t * \widetilde{\mathbf{C}_t} + \mathbf{f}_t * \mathbf{C}_{t-1},\\
\mathbf{h}_t &= \mathbf{o}_t * \tanh(\mathbf{C}_t).
\end{aligned} \tag{5}$$

where t denotes the t-th time slice, $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ refer to the output of the input gate, forget gate and output gate respectively. $\mathbf{x}_t$, $\mathbf{c}_t$, $\mathbf{h}_t$ are the input vector, state vector and hidden vector respectively, and $\mathbf{h}_{t-1}$ is the former output of $\mathbf{h}_t$. $\widetilde{\mathbf{C}_t}$ and $\mathbf{C}_t$ are the input state and output state of the memory cell, and $\mathbf{C}_{t-1}$ is the former state of $\mathbf{C}_t$. σ is a sigmoid function. $\mathbf{W}_{ix}, \mathbf{W}_{fx}, \mathbf{W}_{ox}, \mathbf{W}_{Cx}$ are the weight matrices connecting $\mathbf{x}_t$ to the three gates and the cell input, $\mathbf{W}_{ih}, \mathbf{W}_{fh}, \mathbf{W}_{oh}, \mathbf{W}_{Ch}$ are the weight matrices connecting $\mathbf{x}_{t-1}$ to the three gates and the cell input, $\mathbf{b}_i$, $\mathbf{b}_f$, $\mathbf{b}_o$, $\mathbf{b}_C$ are the bias terms of the three gates and the cell gate.

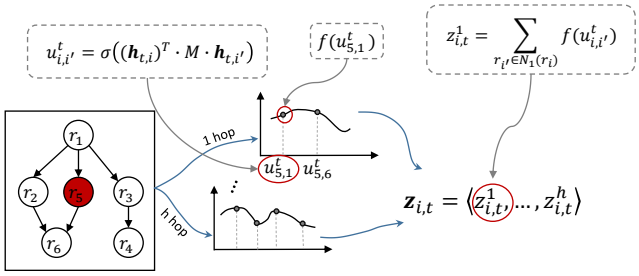


Fig. 4: Graph Convolution Network Layer.

2) GCN Layer: This component fuses the spatial correlations in the local neighborhood of up to h-hop neighbors. With the latent feature vectors obtained from the LSTM layers, a structure-aware GCN [36] is used to aggregate the local neighborhood information from both r_i and all its neighbors of up to h hops. This was shown to be efficient for capturing spatial correlations while preserving the structure consistency for network embedding problems. Fig. 4 illustrates how GCN works with an example, i.e., the traffic states of r_5's local neighborhood at time slice t. The latent feature vector $\mathbf{h}_{t,i}$ (learned through LSTM1 for r_5) and the latent feature vectors $\mathbf{h}_{t,i'}$ (learned through LSTM2 for r_5's neighbors of up to h-hops $N_{1\sim h}(r_i)$) are the inputs to the GCN component. The GCN network learns a latent feature vector $< z_{i,t}^1, z_{i,t}^2, \cdots, z_{i,t}^h >$, where $z_{i,t}^k$ incorporates the spatial

correlations within r_i's neighborhood of up to k-hops at time t. Specifically,

$$u^t_{i,i'} = \sigma((\mathbf{h}_{t,i})^T M(\mathbf{h}_{t,i'})), \tag{6}$$

$$f(u^t_{i,i'}) = \sum_{l=1}^{K_{GCN}} v_l \cdot p_l(u^t_{i,i'}), \tag{7}$$

$$z^h_{i,t} = \sum_{r_{i'} \in N_h(r_i)} f(u^t_{i,i'}), \tag{8}$$

where $\mathbf{h}_{t,i}, \mathbf{h}_{t,i'} \in \mathbb{R}^c$ are input latent features of road r_i and $N_{1\sim h}(r_i)$ (i.e., outputs of LSTM1 and LSTM2, respectively), $(\mathbf{h}_{t,i})^T$ is the transposition of $\mathbf{h}_{t,i}$, $M \in \mathbb{R}^{c\times c}$ is a matrix with $c \times c$ learnable parameters to measure the correlation between embedding vectors, $\sigma(\cdot)$ is the Sigmiod function, K_{GCN} is the number of the truncated polynomials, $\{v_1, ..., v_{K_{GCN}}\}$ are K_{GCN} coefficients corresponding to the Chebyshev polynomials $\{p_1(x), ..., p_{K_{GCN}}(x)\}$, and $N_h(r_i)$ represents the $h-$hop neighbors of r_i. The convolution filter is parameterized with learnable numerical parameters under the guidance of the function approximation theory [36]. The output of GCN module is a vector $\mathbf{z}_{i,t} = \langle z^1_{i,t}, ..., z^h_{i,t}\rangle$ for r_i, where $z^k_{i,t}$ contains the neighborhood information of r_i's neighbors of up to k-hops.

3) FCN Layers: The LPE component relies on two different FCN layers to learn the source embedding and target embedding separately. Specifically, a FCN-src layer learns the source embedding by taking $\mathbf{x}_f = \mathbf{h}_{local}$ (i.e., $\mathbf{h}_{i,t} \oplus \mathbf{z}_{i,t}$) as input ($\oplus$ is the concatenation operation), while a FCN-trg layer learns the target embedding by taking $\mathbf{x}_f = (\mathbf{1} - \mathbf{h}_{local})$ as input. The learned embedding (the output of the FCN layers) is calculated as follows.

$$\mathbf{l}^f = relu(\mathbf{W}_f(\mathbf{x}_f) + \mathbf{b}_f), \tag{9}$$

where $\mathbf{l}^f$ is the output source/target embedding, $\mathbf{W}_f$ is the weight matrix connecting neurons in FCN layer and $\mathbf{x}_f$, $\mathbf{b}_f$ are the bias terms. In particular, the output of the FCN-src is the source embedding $\mathbf{el}^{src}_{i,t} \in \mathbb{R}^{K1}$ of road r_i at time slice t, while the output of FCN-trg is the target embedding $\mathbf{el}^{trg}_{i,t} \in \mathbb{R}^{K1}$ of road r_i at time slice t. They will be combined with the global embeddings to obtain the final embedding.

B. Global Tendency Embedding (GTE)

The LPE component in the previous section only incorporated the local proximity information into embedding, which is insufficient since the traffic congestion may propagate to a large area in a short time. For example, two distant road segments could be involved in a propagation path, but the LPE fails to capture their correlations. On the other hand, even if two distant road segments always show consistent traffic states, they should not be mapped to close proximity in the embedding space if they are not involved in the same propagation path. To address these issues, a global tendency embedding (GTE) component is constructed to complement the LPE component. As such, the embeddings can capture both the proximity over local area and the congestion propagation tendency from the global view (i.e., over the entire traffic network). The GTE component includes three parts: 1) *Propagation Matrix Construction*: A congestion propagation matrix is constructed for each time slice, and then each propagation matrix is approximated with multiple matrices, each focusing on specific temporal correlations. 2) *Non-negative Matrix Factorization* (NMF). The NMF method is used to decompose a propagation matrix into a source matrix $\mathbf{S}$ and a target matrix $\mathbf{T}$ such that the inner product of the two matrices (i.e., $\mathbf{S} \cdot \mathbf{T}^T$) approximate the original matrix as much as possible. Each row in the obtained source (or target) matrix is an initial global embedding for the corresponding road segment. 3) *GTE-Source/Target component*, which learns high-order embeddings for the road segments based on their initial embeddings and further adjust their dimensions.

1) Propagation Matrix Construction: To characterize the congestion propagations from the global view, a propagation matrix PM^t is constructed to describe the congestion propagations that occurred in time slice t. Given dynamic traffic graphs G^t_d and the set of propagation paths at time slice t, the $PM^t \in \mathbb{R}^{N\times N}$ (N is the number of road segments) is constructed in the following way. $PM^t(i, i') = 1$ if there exist a propagation path *pa* containing both r_i and $r_{i'}$, and r_i appeared earlier than $r_{i'}$ in path *pa*. $PM^t(i, i') = 1$ indicates the existence of a congestion propagation from r_i to $r_{i'}$ at time slice t. The PM^t describes a sketch of all congestion propagations at time slice t from the global perspective.

The urban traffic shows various temporal correlations at different granularities. Thus, we leverage multiple propagation matrices to better capture the temporal correlations among the propagations, including near-recent dependency (similarity), daily repeatability, and weekly periodicity. The formation and dispersion of traffic congestions are gradual, indicating that the traffic states at recent time slices inevitably have influences on the near future traffics. Thus, a propagation matrix is calculated to capture the temporal dependency,

$$PM^t_{re} = \frac{1}{ls}\sum_{l=1}^{ls} PM^{t-l} \tag{10}$$

where ls is the number of time slices considered in recent period. The traffic states repeat periodically due to periodic daily travel routines of most people, thus the congestion propagations at a specific period are similar to that of the same period of the previous day or week [37]–[40]. To capture such periodicity, the daily and weekly repeatability is also taken into account by constructing the following two propagation matrices.

$$PM^t_{da} = \frac{1}{ld}\sum_{l=1}^{ld} PM^{t-Td*l} \tag{11}$$

where Td is the number of time slices within a day (e.g. $Td = 288$ if time interval is 5 minutes), and ld is the number of previous days considered. Missing values in the above calculations will not be considered.

2) Non-negative Matrix Factorization: This module relies on Non-negative Matrix Factorization (NMF) to achieve asymmetric embedding of global tendencies in congestion propagation. Specifically, NMF produces a global source matrix $\mathbf{S}^t \in \mathbb{R}^{N\times K_2}$ and a global target matrix $\mathbf{T}^t \in \mathbb{R}^{K_2\times N}$ such

that their inner product $\mathbf{S}^t(\mathbf{T}^t)^T$ approximates the original matrix $\mathbf{PM}^t$ as much as possible, where K_2 is a parameter. After NMF, the ith row of $\mathbf{S}^t$ represents the initial source embedding of r_i while the jth column of $\mathbf{T}^t$ represents the initial target embedding of r_j, at time slice t. Then, the tendency of congestion propagation from r_i to r_j at slice t (i.e., the element $\mathbf{PM}^t_{i,j}$) can be approximated as the inner product of the ith row of $\mathbf{S}^t$ and jth column of $\mathbf{T}^t$. The NMF is achieved by solving the objective function in the form of

$$\min \|\mathbf{PM}^t - \mathbf{S}^t(\mathbf{T}^t)^T\|^2,\ s.t.\ \mathbf{S}^t, \mathbf{T}^t \geq 0 \tag{12}$$

If there exists a propagation from r_i to $r_{i'}$ at time slice t, the element in the ith row and the i'th column of $\mathbf{PM}^t$ equals to 1. After matrix factorization, the inner product of the $i-th$ row in matrix $\mathbf{S}^t$ and i'th row in $(\mathbf{S}^t)^T$ will approximate to 1 as well, i.e. $\mathbf{S}^t(i)\cdot(\mathbf{T}^t)^T(i') \approx 1$. As such, the vector $\mathbf{S}^t(i)$ (or $(\mathbf{T}^t)^T(i')$) indicates the possibility that r_i (or $r_{i'}$) is a source (or a target) in a propagation relationship.

As mentioned before, we constructed three propagation matrices ($\mathbf{PM}^t_{re}, \mathbf{PM}^t_{da}$ and $\mathbf{PM}^t_{we}$) to replace the propagation matrix $\mathbf{PM}^t$ in order to better capture the multi-fold temporal correlations. Another reason for the replacement is that reliable $\mathbf{PM}^t$ may not be available in real-time, due to missing data, noises, fluctuations, and communication delays. The above matrices not only include the information in $\mathbf{PM}^t$ but also reduces the impact of the unreliability of $\mathbf{PM}^t$. Thus, we use the NMF method in [41] to decompose each of the three matrices as follows.

$$\min \|\mathbf{PM}^t_{re} - \mathbf{S}^t_{re}(\mathbf{T}^t_{re})^T\|^2,\ s.t.\ \mathbf{S}^t_{re}, \mathbf{T}^t_{re} \geq 0 \tag{13}$$

$$\min \|\mathbf{PM}^t_{da} - \mathbf{S}^t_{da}(\mathbf{T}^t_{da})^T\|^2,\ s.t.\ \mathbf{S}^t_{da}, \mathbf{T}^t_{da} \geq 0 \tag{14}$$

$$\min \|\mathbf{PM}^t_{we} - \mathbf{S}^t_{we}(\mathbf{T}^t_{we})^T\|^2,\ s.t.\ \mathbf{S}^t_{we}, \mathbf{T}^t_{we} \geq 0 \tag{15}$$

After the NMF, six matrices are obtained (i.e., $\mathbf{S}^t_{re}$, $\mathbf{T}^t_{re}$, $\mathbf{S}^t_{da}$, $\mathbf{T}^t_{da}$, $\mathbf{S}^t_{we}$, and $\mathbf{T}^t_{we}$), which incorporated the multi-fold temporal correlations in the source/target embedding for capturing the propagation behaviors. The rows in $\mathbf{S}^t$ represent the initial source embeddings of the road segments while the rows in $(\mathbf{T}^t)^T$ represent the initial target embeddings of the road segments. These initial embeddings are then fed into neural networks (denoted as GTE-Source/Target Component) to learn high-order representations (embeddings) before being fused with local embeddings.

3) GTE-Source/Target Component: The GTE-Source component consists of three separate LSTM modules, which learn the source embeddings of road segments based on the rows in matrices $\mathbf{S}^t_{re}$, $\mathbf{S}^t_{da}$, and $\mathbf{S}^t_{we}$, respectively. Specifically, the network modules `LSTM-re`, `LSTM-da`, and `LSTM-we` incorporate the recent, daily, and weekly temporal correlations, respectively, as shown in Fig. 3 (c). The three LSTM modules are of the same structure. Each LSTM module contains 3 layers, and each layer contains 16 neurons, which is optimized by a two-stage parameter tuning method. For a road segment r_i at time slice t, the ith row in matrix $\mathbf{S}^t_{re}$ is used as the input to the module `LSTM-re` to produce the global source embedding $\mathbf{g}^{src,re}_{i,t} \in \mathbb{R}^{K_1}$ that incorporated the short-term correlations of congestion propagations. Similarly, network modules `LSTM-da` and `LSTM-we` take the ith row of matrices $\mathbf{S}^t_{da}$ and $\mathbf{S}^t_{we}$ as inputs and produce embeddings $\mathbf{eg}^{src,da}_{i,t} \in \mathbb{R}^{K_1}$ and $\mathbf{eg}^{src,we}_{i,t} \in \mathbb{R}^{K_1}$ respectively. The LSTM modules for processing matrices $\mathbf{T}^t_{da}$, $\mathbf{T}^t_{we}$, and $\mathbf{T}^t_{we}$ are defined in the same way. The target embeddings $\mathbf{eg}^{trg,re}_{i,t}$, $\mathbf{eg}^{trg,da}_{i,t}$, and $\mathbf{eg}^{trg,we}_{i,t}$ are generated based on the ith rows of $(\mathbf{T}^t_{re})^T$, $(\mathbf{T}^t_{da})^T$, and $(\mathbf{T}^t_{we})^T$, respectively. The resultant 6 source and target embeddings are then fed into the Local-Global Fusion module to learn the final representations.

C. Local-Global Fusion Module

The fusion module learns the final source and target embeddings by combining the obtained local/global embeddings according to their significance. The attention mechanism is used to dynamically determine the significance of different input features. As shown in Fig. 3 (b), this module consists of two separate attention modules: `Attention-src` takes the global/local source embeddings (i.e., $\mathbf{el}^{src}_{i,t}, \mathbf{eg}^{src,re}_{i,t}, \mathbf{eg}^{src,da}_{i,t}, \mathbf{eg}^{src,we}_{i,t}$) as inputs and produces the final source embedding $\mathbf{e}^{src}_{i,t}$ for road segment r_i at time slice t; while the `Attention-trg` takes the global/local target embeddings (i.e., $\mathbf{el}^{trg}_{i,t}, \mathbf{eg}^{trg,re}_{i,t}, \mathbf{eg}^{trg,da}_{i,t}, \mathbf{eg}^{trg,we}_{i,t}$) as inputs and produces the final target embedding $\mathbf{e}^{trg}_{i,t}$ for road segment r_i at time slice t. The two attention network modules are of the same structure, and the one for source embedding is described as follows (the one for target embedding is omitted).

$$\mathbf{e}^{src}_{i,t} = \sum_j a^{src}_j \cdot (\mathbf{ex}^{src}_j) \tag{16}$$

$$a^{src}_j = \frac{\alpha^{src}_j}{\sum_l \alpha^{src}_l} \tag{17}$$

$$\alpha^{src}_j = \mathbf{W}^{src}_a(\mathbf{ex}^{src}_j) + b^{src}_a \tag{18}$$

where i indicates the road segment r_i, t indicates the time slice, $\mathbf{ex}^{src}_j \in \mathbb{R}^{K1}$ are the local/global source features learned via the LPE and GTE modules. Specifically, $\mathbf{ex}^{src}_j \in \{\mathbf{el}^{src}_{i,t}, \mathbf{eg}^{src,re}_{i,t}, \mathbf{eg}^{src,da}_{i,t}, \mathbf{eg}^{src,we}_{i,t}\}$, while $\mathbf{ex}^{trg}_j \in \{\mathbf{el}^{trg}_{i,t}, \mathbf{eg}^{trg,re}_{i,t}, \mathbf{eg}^{trg,da}_{i,t}, \mathbf{eg}^{trg,we}_{i,t}\}$. a^{src}_j ($\sum a^{src}_j = 1$) is the learned weights of corresponding feature vector (i.e., $\mathbf{ex}^{src}_j$) via an attention layer based on Eq. (17) and (18), $\mathbf{W}^{src}_a \in \mathbb{R}^{K1}$ is the weight matrix connecting neurons in attention layer and $\mathbf{ex}^{src}_j$, $\mathbf{b}^{src}_a$ are the bias terms. The final embeddings are $\mathbf{e}^{src}_{i,t}$, $\mathbf{e}^{trg}_{i,t} \in \mathbb{R}^{K_1}$.

D. Loss Function

The loss function for training AE-LPGT is defined as:

$$L = -\frac{1}{n}\sum_{k=1}^{n} \hat{y}_k \cdot \ln y_k + (1-\hat{y}_k)\cdot \ln(1-y_k) \tag{19}$$

where n is the batch size of training examples during the training process, $\hat{y}_k \in \{0, 1\}$ is the corresponding target value (label) of propagation between the source and target road segments. $\hat{y}_k = 1$ indicates that there exists a propagation from source to target road segment, and otherwise, $\hat{y}_k = 0$. y_k is the predicted value for the propagation of the corresponding

sample. The y_k is calculated based on the congestion propagation model (which will be introduced in the next section), by taking the embeddings of source and target roads as input.

VI. Congestion Propagation Model.

The congestion propagation model relies on the obtained source and target embeddings ($\mathbf{e}^{src}$ and $\mathbf{e}^{trg}$) to estimate the probability that a congestion on road segment r_i will propagate to $r_{i'}$, for any r_i, $r_{i'}$ in the traffic network. As explained earlier, congestions occur on some road segments first, and the congested road segments are likely to affect their neighbors. The newly congested road segments could also propagate the congestion states to their adjacent neighbors in a domino effect if the travel demands continue to increase. As such, the possibility that propagation occurs along a path can be described as follows. If r_i and $r_{i'}$ are adjacent road segments and r_i is congested, then the likelihood that the propagation occurs between r_i and $r_{i'}$ can be estimated as

$$Pr(r_i, r_{i'}) = \sigma(EP(r_i, r_{i'})) = \sigma(\mathbf{e}_i^{src} \cdot (\mathbf{e}_{i'}^{trg})^T) \quad (20)$$

where $\sigma(\cdot)$ is the Sigmiod function, $EP(r_i, r_{i'}) = \mathbf{e}_i^{src} \cdot (\mathbf{e}_{i'}^{trg})^T$. For a path $pa =< r_i \rightarrow ... \rightarrow r_{i'} >$, the probability that the propagation occurs along the path *pa* is calculated as

$$Pr(pa) = \sigma(EP(pa)) = \sigma(\prod_{<r_a, r_b> \in pa} EP(r_a, r_b)), \quad (21)$$

where edge $< r_a, r_b > \in pa$ indicates that r_a and r_b are consecutive road segments in path *pa*.

In practice, there could be multiple paths between two road segments r_i and $r_{i'}$ and they have heterogeneous impacts on the congestion propagation between r_i and $r_{i'}$. Let $SP(r_i, r_{i'})$ be the set of paths connecting r_i and $r_{i'}$, and there are at least one path in $SP(r_i, r_{i'})$. Assume there are k paths, i.e., $SP(r_i, r_{i'}) = \{pa_1, ..., pa_k\}$. Then, the probability that the propagation occurs between r_i and $r_{i'}$ via the multiple paths in $SP(r_i, r_{i'})$ can be calculated as follows.

$$Pr(SP(r_i, r_{i'})) = \sigma(\sum_{1 \leq j \leqslant k} w_j \cdot EP(pa_j)), \quad (22)$$

where w_j is the weight parameter indicating the importance of path pa_j. w_j is calculated based on occurence frequency of propagations along path pa_j in historical traffic data, and more frequent propagation paths have larger impacts. For example, as shown in Fig. 1 (d), there are two paths connecting r_2 and r_8, i.e., $pa_1 = \langle r_2 \rightarrow r_4 \rightarrow r_8 \rangle$ and $pa_2 = \langle r_2 \rightarrow r_5 \rightarrow r_8 \rangle$. Based on Eq. 21, let $EP(pa_1) = (\mathbf{e}_{2,t}^{src} \cdot (\mathbf{e}_{4,t}^{trg})^T) \cdot (\mathbf{e}_{4,t}^{src} \cdot (\mathbf{e}_{8,t}^{trg})^T)$, while $EP(pa_2) = (\mathbf{e}_{2,t}^{src} \cdot (\mathbf{e}_{5,t}^{trg})^T) \cdot (\mathbf{e}_{5,t}^{src} \cdot (\mathbf{e}_{8,t}^{trg})^T)$. Then, $Pr(SP(r_2, r_8)) = \sigma(w_1 \cdot EP(pa_1) + w_2 \cdot EP(pa_2))$. The congestion propagation model predicts that there will be a propagation from source r_i to target road segment $r_{i'}$ if $Pr(SP(r_2, r_8)) \geq 0.5$. Otherwise, it predicts that there is no congestion propagation between r_i and $r_{i'}$ if $Pr(SP(r_2, r_8)) < 0.5$.

VII. Experiments

A. Experiment Setup

1) Dataset: Road Network: The road network is obtained from OpenStreetMap[1]. We used a rectangle area in Downtown of Singapore (Southwest: 1.2718, 103.8002; Northeast: 1.3323, 103.8653), which contains 1858 road segments. This is used to derive topological attributes of the road segments, which include: types of road segments (e.g., primary, residential, highway), number of lanes, length, availability of traffic signals at the end of road segments, and the number of bus stops on the road segment.

Traffic Speed Data: The traffic speed is calculated based on historical bus trajectories derived from bus arrival data[2] (which contains the GPS location and timestamp of each bus service in operation, at the frequency of every minute). Similar to most works, a speed record is generated for each road segment every 5 minutes. Bus traffic data from Aug. 01 to Nov. 30, 2018, are used in our experiment. Traffic data on the first 90 days (75%) are used as the training set, and the remaining data are used as the testing set.

Traffic Congestions: Similar to [3], a road segment is detected as congested at a time slice if its traffic speed is slower than a threshold value. Specifically, the threshold p is selected as different percentiles of traffic speeds of each road segment. For example, $p = 90\%$ indicates that 90% of the road's historical speed values are higher than the threshold. With this threshold, 10% of the historical traffic speed will be identified as congested. In other words, $c_i^t = 1$ if $\hat{s}_i^t$ is lower than the threshold value. Then the traffic graphs G_d^t, the set of historical propagation paths, as well as the propagation matrix PM^t, are constructed for each time slice t. As existing works have used different thresholds, we tested the performance using a wide range of thresholds. Specifically, thresholds are set to $p = 95\%, 90\%, 85\%, 80\%, 75\%, 70\%$, 65%, and 60%, respectively. A larger threshold leads to fewer congestions detected, e.g., $p = 95\%$ will detect fewer congestions (as well as the propagations) than $p = 90\%$.

Congestion Propagation Samples: To train the model, both positive samples and negative samples are needed. The positive sample set (SS_t^{pos}) is constructed in the following way: For each road segment $r_i \in V_+^t$ at time t ($r_i \in V, 1 \leq t \leq T$), we include all the detected historical propagation paths between road pairs (single or multiple as defined in Section III) that originated from r_i into set SS_t^{pos}. Then, the negative samples are generated in the following way. For each path $pa \in SS_t^{pos}$ or path set $SP(r_i, r_{i'}) \in SS_t^{pos}$, two different types of negative samples are generated. The first type is called boundary negative sample (BNS), which captures the boundary of congestion propagations. Specifically, the BNS sample corresponding to *pa*, denoted as pa^{bnd}, is obtained by replacing the last road segment of *pa* with a non-congested road segment that is spatially connected with the second last road segment of *pa*. Similarly, the last road segment of each path in $SP(r_i, r_{i'})$ will be replaced in the same way to get corresponding negative

[1]https://www.openstreetmap.org/export

[2]https://www.mytransport.sg/content/mytransport/home/dataMall.html

sample. The second type is called inverse negative sample (INS), which indicates that the propagation has a direction and cannot simultaneously happen in the opposite direction. The INS sample corresponding to *pa*, denoted as pa^{ivs}, contains the same road segments as in *pa* but in reversed order. The inverse negative samples of $SP(r_i, r_{i'})$ are generated in the same way. The resultant BNS samples form a set SS_t^{bns} and the resultant INS samples form a set SS_t^{ins} $(1 \leq t \leq T)$.

2) Evaluation Metrics: We apply widely used metrics to evaluate the performance of the embedding model and the propagation model, which includes accuracy, F1 measurement, PR-curve, and ROC-curve. The PR-curve presents the tradeoff between precision and recall, while ROC-curve illustrates the diagnostic ability for different thresholds of a binary classifier by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. In general, large accuracy, high F1 score, and large areas of PR-curve and ROC-curve indicate high performance.

3) Baselines: The following baseline methods are used for comparison. 1) *STC* [3] utilizes frequent subtree mining methods to extract frequently appearing congestion propagation relationship as a tree structure (i.e., the node represents road segment and edge indicates a propagation). Next, a Dynamic Bayesian Network (DBN) is constructed based on these frequent patterns to predict propagations. 2) *Pro-Graph* [42] predicts congestion propagation patterns in the near future with a graph structure. Given a spatial and temporal network and congested road segments at the current time, it predicts where those congestions will propagate to. 3) *EvolveGCN* [43] combines RNN and GCN, where RNN captures graph dynamics to enable dynamic adjusting of the GCN parameters so that GCN could generate varying node embedding in non-Euclidean domain according to evolving traffic conditions. Then, link prediction is obtained using fully connected neural networks by taking the embeddings of source and target nodes as inputs. 4) *Node2Bits* [44] is a method for modeling the interactions among users in heterogeneous web networks for personalization and recommendations of web services. It represents multi-dimensional features of node contexts with binary hashcodes, and predicts whether two users correspond to the same entity (has a link) by considering temporal dimension and dynamic attributes associated with each user. 5) *ANRL* [45] proposes a network representation learning framework, which incorporates node attributes with a neighbor enhancement auto-encoder, and captures structure correlations by an attribute-aware skip-gram model. 6) *MMDNE* [46] aims to capture micro- and macro-dynamics for temporal network embedding. It proposes a temporal attention point process to incorporate micro-level dynamics; for macro-level, it uses network embeddings to parameterize a dynamic equation to impose high-level structure information. 7) *Our-woLP* is a version of our method without the local proximity embedding component. This baseline is used to verify the effectiveness of the LPE component. 8) *Our-woGT* is another version of our method without the global propagation tendency factors. The LPE component incorporates both spatial and temporal correlations in the local neighborhood, which produces a single feature vector of local proximity. As such, the output of the LPE component is directly used as the final embedding, which means that the attention-based fusion module is also removed. 9) *Our-woAT* is another ablation version of our method that does not consider the asymmetric transitivity of congestion propagation. In *Our-woAT*, each road segment has a single embedding, instead of a source embedding plus a target embedding. Specifically, the NMF in GTE module is not used and the (rows/columns in the) original propagation matrix are used to replace the rows/columns in the resultant matrices $\mathbf{S}$ and $\mathbf{T}^T$.

The first two baselines are the most recent works investigating similar problems. The subsequent four baselines are methods for link prediction based on network embedding. Then, three ablation baselines are designed to verify the impacts of model components. While there exist many efficient methods for traffic state prediction, we did not include them as baseline methods as they failed to produce good results. This is because traffic state prediction is a node prediction problem, while the congestion propagation problem is a link prediction problem. Thus, they require different techniques to boost the prediction performance. Specifically, congestion propagation has unique characteristics that are different from traffic state on individual roads, e.g., the asymmetric transitivity of congestion propagations, the local proximity in congestion propagations, and the dynamic global tendency of congestion propagations.

4) Hyperparameters: We implement our model using PyTorch framework on Intel(R) Xeon(R) CPU E5-1650 v2 @ 3.50GHz with 32G RAM. We perform a two-stage method to tune the hyperparameters: the first stage obtains an initial parameter configuration with coarse-grained parameter values, while the second stage further refines each parameter separately using a grid search method. In the first stage, we adopt the Orthogonal Array based tuning method [47] to jointly optimize hyperparameters of the AE-LPGT model. After testing the performance of each parameter combination in the Orthogonal Array table, range analysis is performed to optimize each factor and combines the optimal levels together. Hence, the optimized hyper-parameter combination is not restricted to the existing Orthogonal Array table. Specifically, 10 hyperparameters are optimized whose values are set as follows (based on the Taguchi L32 type B orthogonal array): the number of hops of local neighborhood in LPE module h (2,4), the vector dimension K_1 of the (1,2,4,8) dimension K_2 in NMF (1,2,4,8), dimension c in GCN (8, 16, 24,32), K_{GCN} in GCN (1,2,4,8), number of neurons in each LSTM layer (8, 16, 32, 64), number of neurons in each FCN layer (8, 16, 32, 64), number of neurons in each attention layer (8, 16, 32, 64), the learning rate (0.0001, 0.0005, 0.001, 0.005), the batch size (20, 40, 60, 80). After tuning, the initial parameters are set to $h = 4$, $K_1 = 4$, $K_2 = 1$, $c = 8$, $K_{GCN} =$, 16, 16, 16, 0.0001, 40, respectively. Then, the above parameters are further tuned separately using grid search based on the initial configuration, and the final configurations are as follows: $h = 5$, $K_1 = 5$, $K_2 = 1$, $c = 8$, $K_{GCN} =$, 16, 16, 12, 0.0001, 30, respectively. In addition, the parameters ls, ld, lw and W are set to 6, 7, 4, and 6, respectively, based on settings from existing works [48] as well as the time horizon of the data samples.

TABLE I: Comparison of overall performance of all methods, in terms of accuracy and F1-score.

	Accuracy								F1-score							
	95%	90%	85%	80%	75%	70%	65%	60%	95%	90%	85%	80%	75%	70%	65%	60%
STC	0.607	0.652	0.654	0.642	0.648	0.649	0.653	0.671	0.615	0.685	0.717	0.692	0.677	0.691	0.727	0.769
Pro-Graph	**0.926**	0.860	0.800	0.745	0.692	0.642	0.594	0.547	0.079	0.097	0.107	0.113	0.117	0.119	0.119	0.118
EvolveGCN	0.506	0.511	0.515	0.514	0.514	0.515	0.518	0.571	0.667	0.670	0.672	0.671	0.668	0.667	0.671	0.672
Node2Bits	0.492	0.498	0.502	0.508	0.512	0.514	0.517	0.514	0.601	0.614	0.621	0.634	0.642	0.651	0.657	0.647
ANRL	0.504	0.499	0.512	0.515	0.522	0.529	0.534	0.538	0.651	0.645	0.658	0.662	0.667	0.673	0.681	0.683
MMDNE	0.511	0.51	0.517	0.521	0.514	0.511	0.5	0.496	0.653	0.667	0.67	0.672	0.668	0.648	0.639	0.633
Our-woLP	0.861	0.810	0.786	0.750	0.751	0.745	0.744	0.737	0.824	0.760	0.720	0.659	0.660	0.645	0.642	0.616
Our-woGT	0.730	0.756	0.801	0.795	0.812	0.819	0.826	0.838	0.596	0.655	0.742	0.726	0.749	0.766	0.767	0.787
Our-woAS	0.499	0.502	0.523	0.5	0.501	0.493	0.496	0.496	0.601	0.636	0.679	0.667	0.632	0.577	0.590	0.617
Ours	0.886	**0.873**	**0.867**	**0.8691**	**0.877**	**0.876**	**0.877**	**0.887**	**0.851**	**0.836**	**0.821**	**0.825**	**0.834**	**0.833**	**0.832**	**0.846**

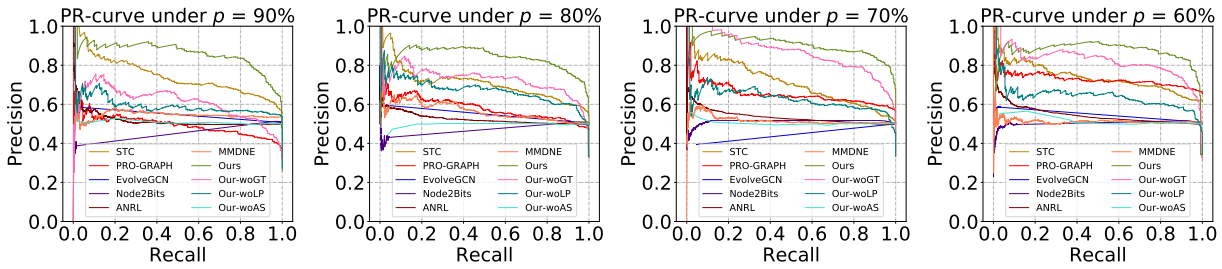


Fig. 5: The PR-curve of all methods on scenarios with congestion thresholds of $p = 90\%, 80\%, 70\%$, and 60%, respectively.

B. Results

Overall Performance: Table I compares the overall performance between our proposed method and the baselines in terms of accuracy and F1-score. The results show that `Pro-Graph` is highly sensitive to the congestion threshold of p. For example, the prediction accuracy decreases rapidly when p decreases from 95% to 60%. `Pro-Graph` obtained promising accuracy when $p = 95\%$, which is better than other methods. This is because when p is 95%, only a small amount of congestion propagations and their patterns are detected. These propagations are typically heavy congestions that frequently occur, and is relatively easier to predict due to high repeatability. `Pro-Graph` only focuses on such propagations, thus it obtains good results when p is large. When p decreases, more propagations with complex patterns are detected, which makes the prediction more challenging, and hence `Pro-Graph`'s prediction accuracy decreases. In fact, `Pro-Graph` tends to produce a large number of false-negative (FN) cases, i.e., high accuracy but very small F1-scores. On the other hand, `STC` does not seem to be affected by the congestion thresholds (as reflected by both its accuracy and F1-score values). However, its performance is much worse even when comparing to our ablation versions, i.e., `Our-woLP` and `Our-woGT`. The main reason is that, `STC` has not sufficiently considered various realistic characteristics of congestion propagations. For example, it does not take into account multi-fold temporal correlations and the global tendency of the congestion propagations.

It is worth noting that baseline Node2Bits, EvolveGCN, ANRL, and MMDNE predict links between two nodes rather than a path, and they do not incorporate asymmetry relation. For fair comparison, we only considered 2-hop propagation paths as positive samples (accounting for 70% of all propagation paths). We feed the methods that utilize node attributes (Node2Bits, EvolveGCN, and ANRL) with the same node features used in our method. The results show that our method significantly outperforms these four baselines. This could be explained by the fact that our proposed method has properly learned multiple realistic properties of congestion propagations in the low-dimensional road embeddings, i.e., local proximity, global propagation tendency, and asymmetric transitivity of congestion propagations.

The results show that any of the three ablation baselines leads to obvious performance degradation, which indicates that all three factors contribute significantly to the prediction. It also shows that both `Our-woLP` and `Our-woGT` are sensitive to the congestion threshold p. `Our-woLP` obtains better performance on scenarios with large p while `Our-woGT` obtains better performance on scenarios with small p. This indicates that the local proximity (removed from `Our-woLP`) is important for characterizing dense congestion traffics (i.e., with low p values such as 60%), while the global propagation tendency factors (removed from `Our-woGT`) is important for characterizing sparse congestion traffic (i.e., with high p such as 90%). `Our-woAS` has poor performance compare with `Our-woLP` and `Our-woGT`, which indicates that asymmetric transitivity has the largest impact. In general, neither of the three ablation methods can adequately characterize the propagations for all scenarios, i.e., with various congestion thresholds.

Fig. 5 and Fig. 6 show the PR-curve and ROC-curve of all methods. In general, our method significantly outperforms all the baselines in both PR-curve and ROC-curve, in all scenarios tested (i.e., with congestion threshold set to 90%, 80%, 70%, and 60%, respectively). Fig. 5 shows that the performances of our method, `STC` , and `Our-woGT` remain stable with the increasing density of traffic congestions (i.e., decreasing threshold p), while the other three methods (Our-woLP, Pro-Graph and ANRL) tend to obtain better perfor-

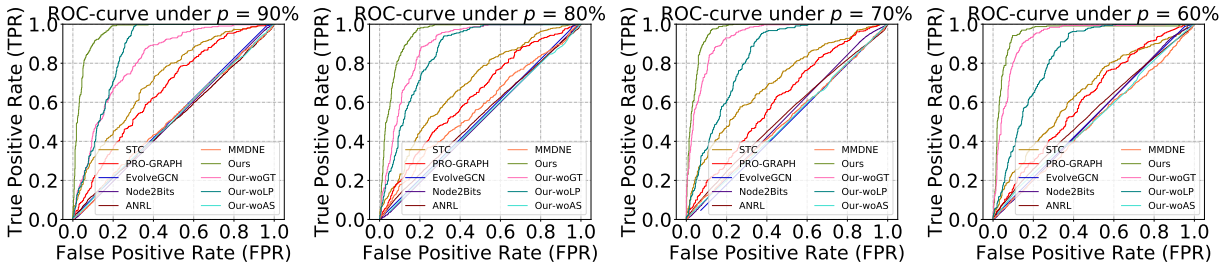


Fig. 6: The ROC-curve of all methods on scenarios with congestion thresholds of $p = 90\%, 80\%, 70\%$, and 60%, respectively.

mance. EvolveGCN and Node2Bits are not sensitive to p value, while MMDNE has a generally decreasing trend in performance as p increased. However, the differences between our method and the baselines are significant. Fig. 6 shows that both `Our-woLP` and `Our-woGT` can outperform the existing methods `STC` and Pro-Graph and the four network embedding based baselines in the ROC-curve. The performance of our methods is not sensitive to the changing p. `Our-woGT`, `ANRL` and `Pro-Grpah` tend to achieve better results on smaller p, `Our-woLP` and `MMDNE` and exhibit performance degradation with decreasing p, while the other baselines are also not very sensitive to p.

Effect of Impact Factors: There are three critical parameters that control the feature vectors, which define the size/dimension of the intermediate feature vectors at different stages of the embedding model. The three parameters include 1) $K1$ defines the size of the LPE embedding, the GT embedding, as well as the final embedding (the fusion of the LPE and GTE embeddings); 2) $K2$ defines the dimension of the latent space for NMF in GTE module; 3) c defines the dimension of feature vectors in GCN module. We also tested the impact of the number of used truncated polynomials in GCN module on the final embedding performance.

Fig. 7 (a) evaluates the impact of $K1$ by testing the performance with $K1 = 1, 2, \cdots, 10$. The results show that the performance first increases, and then decreases with increasing $K1$ values, and obtains the best results when $K1 = 5$. This is because a small $K1$ weakened the representation strength of the final embeddings, while a large $K1$ produces redundant representations that harm the performance. As such, $K1$ is further adjusted to 5 in the second stage of parameter optimization. Fig. 7 (b) shows that the performance generally decreases with increasing values of $K2$ (with some fluctuations). As such, $K2$ is set to 1 in the second stage of parameter optimization. Fig. 7 (c) tested the impact of parameter c to the final prediction performance. The results show that, $c = 8$ has the best performance during the second stage of parameter optimization. Fig. 7 (d) illustrates the impact of parameter K_{GCN}, which determines the number of the truncated polynomials to approximate the convolution filter. By varying K_{GCN} in range $\{1, 2, \cdots, 10\}$, the performance first increases then decreases, and obtains the best results when $K_{GCN} = 5$. This is because when K_{GCN} is small, increasing the number of polynomials improves the learning of the complex traffic situations of local neighbors. However, when K_{GCN} exceeds 5, the prediction performance tends to degrade if KGCN continues to increase. This is because the size of the model grows rapidly with redundant parameters, and the model overfits during training for the dataset. As such, K_{GCN} is set to 5 in the second-stage of parameter tuning.

Evaluation on Efficiency: We evaluated the proposed methods using different amounts of training data, i.e., using data of 1, 2, 3, and 4 months, respectively. The results in Fig. 8 (a) show that the performance increases with using more data, and there is a significant improvement when using the four-month dataset. This indicates that long-term temporal correlations are important, and it is necessary to employ long-term traffic data (if available) to achieve reliable prediction of congestion propagations. Fig .8 (b) shows that the training time of all methods. Since the baselines `STC` and Pro-Graph are not machine learning methods (they first mine propagation patterns and then make predictions based on the obtained patterns), the time for pattern mining is included in their training time. Our method is only obviously slower than one baseline EvolveGCN. Even though the training time is rather long, we can achieve efficient prediction during the testing phase. The prediction times for processing 1000 samples are as follows: 1.48s, 1.27s, 0.97s, 0.94s, 0.89s, 0.29s and 6.92s, for methods illustrated in Fig .8 (b).

VIII. Conclusion

This work studied the congestion propagation behavior at the individual road segment level, and propose a novel machine learning framework to characterize and predict congestion propagations in urban traffic networks. The proposed framework relies on AE-LPGT to learn representations of the road segments and a propagation model to infer the likelihood of congestion propagation based on the learned representations. AE-LPGT jointly incorporates multiple practical characteristics of congestion propagations that have been neglected by existing works. We also developed a propagation model that estimates the probability of congestion propagations between two road segments, which could be connected via multiple paths. The effectiveness of the AE-LPGT embedding model and the propagation model was verified on real traffic data.

In addition to the traffic congestion propagation problems considered in this paper, our proposed methods for traffic congestion propagation can also be applied (with slight changes) to similar problems in other domains that require network embedding techniques. These include rumor spread (propagation) in social network [49], citation trend modeling and impact prediction in citation networks [50], user preference prediction

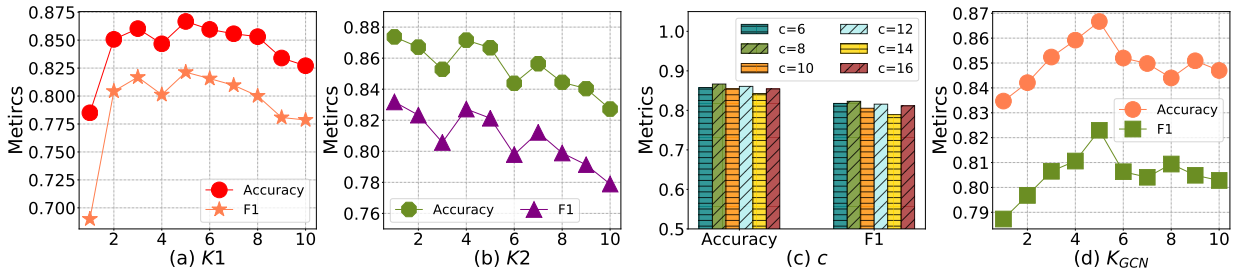


Fig. 7: The evaluation of hyperparameters $K1$, $K2$, c, and K_{GCN}

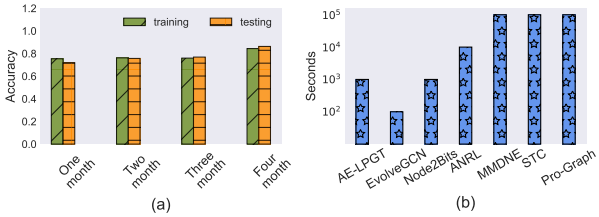


Fig. 8: (a) Results on datasets of different sizes, (b) Comparison of training time (30 epoches on four-month dataset).

via social network influence modeling (with applications of service recommendations in social media and online shopping websites) [51], etc. In these problems, the propagation of information (e.g., rumor, preference) poses similar propagation characteristics such as asymmetric transitivity, local proximity, and dynamic global tendency of propagations. The proposed embedding methods can also be applied partially to learn efficient node/link embedding for network prediction problems, e.g., utilizing LPE module to learn spatiotemporal correlations in local neighborhood, and GTE module to learn evolving multi-view global tendency in the entire network.

REFERENCES

[1] A. Elkafoury, A. M. Negm, M. H. Aly, M. F. Bady, and T. Ichimura, "Develop dynamic model for predicting traffic co emissions in urban areas," *Environ. Sci. Pollut. Res.*, vol. 23, no. 16, pp. 15 899–15 910, 2016.

[2] G. Weisbrod, D. Vary, and G. Treyz, "Measuring economic costs of urban traffic congestion to business," *Transp. Res. Record.*, vol. 1839, no. 1, pp. 98–106, 2003.

[3] H. Nguyen, W. Liu, and F. Chen, "Discovering congestion propagation patterns in spatio-temporal traffic data," *IEEE Trans. Big Data*, vol. 3, no. 2, pp. 169–180, 2016.

[4] T. Anwar, C. Liu, H. L. Vu, and M. S. Islam, "Tracking the evolution of congestion in dynamic urban road networks," in *Proceedings of the 25th ACM Intell. Conf. on Inform. and Knowl. Manag.*, 2016, pp. 2323–2328.

[5] T. Anwar, C. Liu, H. L. Vu, M. S. Islam, and T. Sellis, "Capturing the spatiotemporal evolution in road traffic networks," *IEEE Trans. Knowl. and Data Eng.*, vol. 30, no. 8, pp. 1426–1439, 2018.

[6] T. Anwar, C. Liu, H. Le Vu, and C. Leckie, "Spatial partitioning of large urban road networks." in *EDBT*, 2014, pp. 343–354.

[7] J. Sun, Z. Ma, and X. Chen, "Some observed features of traffic flow phase transition at urban expressway diverge bottlenecks," *TRANSPORTMETRICA B*, vol. 6, no. 4, pp. 320–331, 2018.

[8] N. Song, "Research on traffic congestion propagation model based on rectangular method," *Mod. Transp.*, vol. 5, no. 1, 2019.

[9] W. Yue, C. Li, and G. Mao, "Urban traffic bottleneck identification based on congestion propagation," in *2018 IEEE Intell. Conf. on Commun. (ICC)*. IEEE, 2018, pp. 1–6.

[10] J. Gregoire, X. Qian, E. Frazzoli, A. De La Fortelle, and T. Wongpiromsarn, "Capacity-aware backpressure traffic signal control," *IEEE Trans. on Control of Netw. Syst.*, vol. 2, no. 2, pp. 164–173, 2014.

[11] X. Di, Y. Xiao, C. Zhu, Y. Deng, Q. Zhao, and W. Rao, "Traffic congestion prediction by spatiotemporal propagation patterns," in *2019 20th IEEE Intell. Conf. on Mob. Data Manag. (MDM)*. IEEE, 2019, pp. 298–303.

[12] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.

[13] S. An, H. Yang, J. Wang, N. Cui, and J. Cui, "Mining urban recurrent congestion evolution patterns from gps-equipped vehicle mobility data," *Inform. Sciences*, vol. 373, pp. 515–526, 2016.

[14] S. An, H. Yang, and J. Wang, "Revealing recurrent urban congestion evolution patterns with taxi trajectories," *ISPRS Interl. J. of Geo-Inform.*, vol. 7, no. 4, p. 128, 2018.

[15] Z. Chen, Y. Yang, L. Huang, E. Wang, and D. Li, "Discovering urban traffic congestion propagation patterns with taxi trajectory data," *IEEE Access*, vol. 6, pp. 69 481–69 491, 2018.

[16] A. Keler, J. M. Krisp, and L. Ding, "Detecting traffic congestion propagation in urban environments–a case study with floating taxi data (ftd) in shanghai," *J. of location Based services*, vol. 11, no. 2, pp. 133–151, 2017.

[17] Y. He, L. Wang, Y. Fang, and Y. Li, "Discovering congestion propagation patterns by co-location pattern mining," in *APAC Web (APWeb) and Web-Age Inform. Manag. (WAIM) Joint Interl. Conf. on Web and Big Data*. Springer, 2018, pp. 46–55.

[18] M. Treiber and A. Kesting, "Calibration and validation of models describing the spatiotemporal evolution of congested traffic patterns," *arXiv preprint arXiv:1008.1639*, 2010.

[19] M. Khajeh-Hosseini and A. Talebpour, "Back to the future: Predicting traffic shockwave formation and propagation using a convolutional encoder-decoder network," in *2019 IEEE Intel. Trans. Syst. Conf. (ITSC)*. IEEE, 2019, pp. 1367–1372.

[20] L. Yang and L. Wang, "Mining traffic congestion propagation patterns based on spatio-temporal co-location patterns," *Evol. Intell.*, pp. 1–13, 2019.

[21] C. Li, W. Yue, G. Mao, and Z. Xu, "Congestion propagation based bottleneck identification in urban road networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 4827–4841, 2020.

[22] H. Gao, Y. Yang, L. Huang, Y. Wang, B. Jia, F. Yang, and Z. Zhu, "Trajectory data-driven pattern recognition of congestion propagation in road networks," in *Interl. Conf. on Algorithms and Architectures for Parallel Process.* Springer, 2018, pp. 199–211.

[23] J. Long, Z. Gao, H. Ren, and A. Lian, "Urban traffic congestion propagation and bottleneck identification," *Science in China Series F: Inform. Sciences*, vol. 51, no. 7, p. 948, 2008.

[24] C. Liu, Q.-p. Zhang, and X. Zhang, "Emergence and disappearance of traffic congestion in weight-evolving networks," *Simul. Model. Pract. Theory*, vol. 17, no. 10, pp. 1566–1574, 2009.

[25] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Trans. Knowl. and Data Eng.*, vol. 31, no. 5, pp. 833–852, 2018.

[26] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proc. 24rd Int. Conf. World Wide Web. (WWW)*, 2015, pp. 1067–1077.

[27] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (SIGKDD)*, 2014, pp. 701–710.

[28] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," in *Thirty-first AAAI Conf. Artif. Intell. (AAAI)*, 2017.

[29] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proc. 22th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (SIGKDD)*, 2016, pp. 1105–1114.

[30] C. C. Paige and M. A. Saunders, "Towards a generalized singular value decomposition," *SIAM J. Numer. Anal.*, vol. 18, no. 3, pp. 398–405, 1981.

[31] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[32] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Soc. networks*, vol. 25, no. 3, pp. 211–230, 2003.

[33] Y. Shen, C. Jin, and J. Hua, "Ttpnet: A neural network for travel time prediction based on tensor decomposition and graph embedding," *IEEE Trans. Knowl. and Data Eng.*, pp. 1–1, 2020.

[34] Z. Zheng, Y. Yang, J. Liu, H.-N. Dai, and Y. Zhang, "Deep and embedded learning approach for traffic flow prediction in urban informatics," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3927–3939, 2019.

[35] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," pp. 3634–3640, 2018.

[36] J. Chang, J. Gu, L. Wang, G. Meng, S. Xiang, and C. Pan, "Structure-aware convolutional neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 11–20.

[37] Z. Hou and X. Li, "Repeatability and similarity of freeway traffic flow and long-term prediction under big data," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1786–1796, 2016.

[38] M. Chen, X. Yu, and Y. Liu, "Pcnn: Deep convolutional networks for short-term traffic congestion prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 11, pp. 3550–3559, 2018.

[39] C.-H. Wu, J.-M. Ho, and D.-T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, 2004.

[40] J. Tang, F. Liu, Y. Zou, W. Zhang, and Y. Wang, "An improved fuzzy neural network for traffic speed prediction considering periodic characteristic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2340–2350, 2017.

[41] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2001, pp. 556–562.

[42] H. Xiong, A. Vahedian, X. Zhou, Y. Li, and J. Luo, "Predicting traffic congestion propagation patterns: A propagation graph approach," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst. (SIGSPATIAL)*, 2018, pp. 60–69.

[43] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. Schardl, and C. Leiserson, "Evolvegcn: Evolving graph convolutional networks for dynamic graphs," in *Proceedings of the AAAI Conf. on Artif. Intell. (AAAI)*, vol. 34, no. 04, 2020, pp. 5363–5370.

[44] D. Jin, M. Heimann, R. A. Rossi, and D. Koutra, "Node2bits: Compact time-and attribute-aware node representations for user stitching," in *Joint Eur. Conf. on Mach. Learn. and Knowl. Discov. in Datab. (ECML-PKDD)*. Springer, 2019, pp. 483–506.

[45] Z. Zhang, H. Yang, J. Bu, S. Zhou, P. Yu, J. Zhang, M. Ester, and C. Wang, "Anrl: Attributed network representation learning via deep neural networks." in *Int. Conf. on Adv. in Geogr. Inf. Syst. (IJCAI)*, vol. 18, 2018, pp. 3155–3161.

[46] Y. Lu, X. Wang, C. Shi, P. S. Yu, and Y. Ye, "Temporal network embedding with micro-and macro-dynamics," in *CIKM*, 2019, pp. 469–478.

[47] X. Zhang, X. Chen, L. Yao, C. Ge, and M. Dong, "Deep neural network hyperparameter optimization with orthogonal array tuning," in *Neural Inf. Process.* Springer, 2019, pp. 287–295.

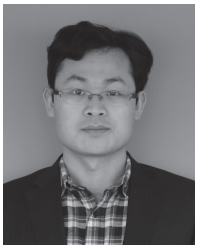
[48] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Thirty-Third AAAI Conf. Artif. Intell. (AAAI)*, vol. 33, 2019, pp. 922–929.

[49] M. Dong, B. Zheng, N. Quoc Viet Hung, H. Su, and G. Li, "Multiple rumor source detection with graph convolutional networks," in *Proceedings of the 28th ACM Int. Conf. Inf. Knowl. Manag.*, 2019, pp. 569–578.

[50] D. Cummings and M. Nassar, "Structured citation trend prediction using graph neural networks," in *ICASSP 2020-2020 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2020, pp. 3897–3901.

[51] P. K. Sharma, S. Rathore, and J. H. Park, "Multilevel learning based modeling for link prediction and users' consumption preference in online social networks," *Future Gener. Comput. Syst.*, vol. 93, pp. 952–961, 2019.

Yidan Sun received the B.Eng. degree from the School of Computer Science and Technology, Tianjin University, China. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. Her research interests include energy efficient MPSoC design, big data analytics in transport, and urban computing.



Guiyuan Jiang received the B.S. degree from Northwest University for Nationalities, China, the M.Eng. degree from Tianjin Polytechnic University, China, and the doctoral degree from Tianjin University (TJU), all in Computer Science and Technology. Currently, he works as a research fellow at School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include data analytics for citywide transport modeling and optimization, design methodologies for reconfigurable computing systems, resource optimization for datacenter and sensor networks. His research has led to over 60 publications in top venues such as IEEE TC, IEEE TPDS, IEEE TITS, IEEE TVLSI, IEEE TCAD, IEEE TIFS, IEEE-ACM TNET, IJCAI, DATE, SDM, etc.

Lam Siew Kei received his BASc, MEng and PhD from School of Computer Science and Engineering (SCSE), NTU. He was a Visiting Research Fellow in the Imperial College of London, University of Warwick, and RWTH Aachen, Germany. He is currently an Assistant Professor in SCSE, NTU, and his research focuses on developing custom computing techniques to meet the challenging demands for performance, energy-efficiency, cost, reliability, and security in edge intelligence. His current research projects include secure and reliable edge computing systems, design methodologies for domain-specific AI architectures, visual edge intelligence for autonomous systems, and distributed intelligence for smart mobility. His research has led to over 150 publications in top venues such as IEEE TC, IEEE TVLSI, IEEE TCAD, IEEE TCAS, IEEE TCSVT, IEEE TPDS, IEEE ITS, DATE, ASP-DAC, FPL, ICFPT, HASP@ISCA, etc. He serves as the Associate Editor for the IET Circuits, Devices and Systems journal and is a Senior Member of the IEEE.

Peilan He received the Bachelor degree of Management from School of Economics and Management, Hainan University, Haikou, China, and the Master degree of Engineering from School of Computer Science and Technology, Tianjin University, China. She is currently working toward the PhD degree in the School of Computer Science and Engineering, Nanyang Technological University, Singapore. Her research interests include machine learning based method for combinatorial optimization, big data analytics in transport, multi-modal journey planning.