

Spatio-temporal Modeling for Large-scale Vehicular Networks Using Graph Convolutional Networks

Juntong Liu*, Yong Xiao*[‡], Yingyu Li*, Guangming Shi^{†‡}, Walid Saad[§], and H. Vincent Poor[¶]

*School of Electronic Inform. & Commun., Huazhong Univ. of Science & Technology, China

[†]School of Artificial Intelligence, Xidian University, Xi'an, China

[‡]Pazhou Lab, Guangzhou, China

[§]Bradley Department of Electrical and Computer Engineering, Virginia Tech, VT

[¶]School of Engineering & Applied Science, Princeton University, Princeton, NJ

Abstract—The effective deployment of connected vehicular networks is contingent upon maintaining a desired performance across spatial and temporal domains. In this paper, a graph-based framework, called *SMART*, is proposed to model and keep track of the spatial and temporal statistics of vehicle-to-infrastructure (V2I) communication latency across a large geographical area. *SMART* first formulates the spatio-temporal performance of a vehicular network as a graph in which each vertex corresponds to a subregion consisting of a set of neighboring location points with similar statistical features of V2I latency and each edge represents the spatio-correlation between latency statistics of two connected vertices. Motivated by the observation that the complete temporal and spatial latency performance of a vehicular network can be reconstructed from a limited number of vertices and edge relations, we develop a graph reconstruction-based approach using a graph convolutional network integrated with a deep Q-networks algorithm in order to capture the spatial and temporal statistic of feature map of latency performance for a large-scale vehicular network. Extensive simulations have been conducted based on a five-month latency measurement study on a commercial LTE network. Our results show that the proposed method can significantly improve both the accuracy and efficiency for modeling and reconstructing the latency performance of large vehicular networks.

Index Terms—Spatio-temporal modeling, Graph Convolutional Networks, latency modeling, deep Q-networks

I. INTRODUCTION

With the rapidly growing demand on intelligent vehicular services and applications, connected vehicles that rely on external communication, computation, and storage resources to facilitate decision making and driving assistance have become increasingly popular. According to the recent report [1], in 2025 over 60% of new vehicles sold globally will be connected to the Internet by wireless technologies such as 5G and beyond [2].

Despite this surge in popularity, there exists many challenges. In particular, there is a need to better understand how the achievable communication latency over spatial and temporal domains. For instance, due to the heterogeneity in services and applications as well as the diversity of service-requesting devices such as wearable devices [3], sensors [4], LiDar [5], and others, the maximum tolerable latency of different vehicular services can dynamically change across a wide range. Moreover, the latency of a large-scale vehicular network is location-dependent, closely related to the potential

signal blockage and interference caused by factors such as the surrounding environment as well as the distribution of the network infrastructure. The challenge for spatial and temporal latency modeling is further exacerbated by the fact that vehicles are consistently moving from one location to another, causing frequent service and link changes. As such, there is a need to develop a simple and effective solution to capture the performance, in terms of latency, of a large-scale vehicular network across different time and location.

According to recent observation reported in [6] [7], the instantaneous latency performance of each mobile device does not exhibit any noticeable spatial and temporal correlations. The statistical feature such as probability distribution function (PDF) however does show strong spatial and temporal dependencies. This makes it natural to develop a graph-based model to capture the statistical features of a vehicular network in which each location point can be seen as a graph vertex and each edge could represent the spatial correlation between two connected location points. Despite its potential, formulating a graphical model to characterize the interactive latency (e.g. round-trip time (RTT)) of a vehicular network faces the following novel challenges. First, it is generally impossible to constantly collect samples across a wide geographical area and keep track of temporal statistics at all locations. Second, the correlation of latency performance at different time stamps and locations can be complex and difficult to measure. There are still lacking commonly adopted metrics to quantify the correlation of the statistical distributions of interactive latency. Finally, vehicles driving at different locations may request different subsets of services, each of which may have unique service demands and requirements. Thus, due to the random nature of wireless networks, it is generally impossible to always support all the requested services with the guaranteed performance.

A. Related Works

Minimizing the latency of communication links is essential for next generation wireless technology [8]. Most existing works focused on how to maintain the latency experienced by a connected vehicle below a deterministic threshold. In particular, the authors in [6] propose AdaptiveFog, a novel framework to maximize confidence levels in LTE-based fog computing for smart vehicles. In [7], the authors propose a

spatio-temporal generative learning model to reconstruct the missing latency samples based on a variation autoencoder.

There has been a number of recent works [9]–[11] that investigated the spatial and temporal features of a vehicular networks. For examples, in [9], the authors investigate the temporal and spatial dynamics of vehicular ad-hoc networks in order to study communication properties such as adaptability, stability, and effectiveness for safety-critical applications, where latency plays a major role. In [10], a novel spatio-temporal caching strategy is proposed based on the notion of temporal graph motifs that can capture spatio-temporal communication patterns in vehicle-to-vehicle networks. In [11], the authors propose a framework based on spatio-temporal graph convolutional networks (GCN) for traffic prediction using spatio-temporal blocks to capture comprehensive spatio-temporal correlations in multi-scale traffic networks.

However, the prior art in [6], [7], [9], [10] does not consider the dynamics of the statistical features for large vehicular networks. Therefore, there is a strong need for developing a simple but effective solution that can exploit the spatio-temporal correlation for tracking the latency performance of large-scale vehicular network.

B. Contributions

The main contribution of this paper is a novel graphical modeling and reconstruction framework, called *SMART* (Spatio-temporal Modeling And ReconsTruction), that can be used to characterize the feasibility of supporting different latency-sensitive services in a vehicular network across a large geographical area. In particular, we first model a large vehicular network as a graph by dividing the entire service area into different subregions, each of which corresponds to a vertex consisting of connected location points with similar latency statistical probabilities. Any two neighboring subregions will be connected with an edge. Statistical distance measures such as the Jensen-Shannon (JS) divergence have been introduced to quantify the correlation between neighboring subregions. *SMART* adopts GCN and deep Q-networks (DQN) to capture the latency graphs’ spatial and temporal features, respectively. We show that, when some graphical features change, the captured spatial correlation is sufficient to reconstruct the complete updated graphical structure of a large vehicular network from an incomplete set of samples collected from a limited number of subregions. To accelerate the reconstruction speed of a large vehicular network, we propose an efficient graph reconstruction solution based on natural gradient descendant (NGD). We conduct extensive performance evaluation using real traces collected over a five-month measurement campaign in a commercial LTE network. Simulation results show that our proposed method can accurately recover the spatio-temporal latency performance across all the subregions in a large vehicular network.

The rest of the paper is organized as follows. In Section II, we present the preliminary observations. Section III describes the methodology used in our framework in detail. In Section IV, we describe experimental setups and present the simulation results. Finally, we conclude the paper and discuss potential future works in Section V.

II. PRELIMINARY OBSERVATION AND ARCHITECTURE OVERVIEW

A. Preliminary Observation

The latency of wireless communication systems is known to exhibit spatial and temporal variation. Here, we particularly focus on the wireless access latency between a moving vehicle and the first IP address (i.e., the first node encountered in a cellular system) of a commercial LTE network, also called the vehicle-to-infrastructure (V2I) communication latency.

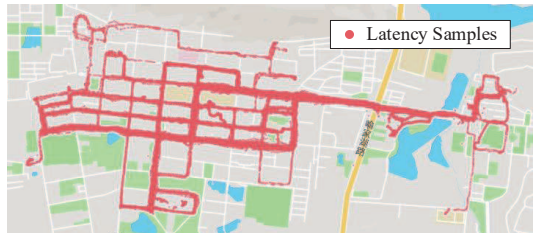


Fig. 1: Measuring routes and traces of our dataset.

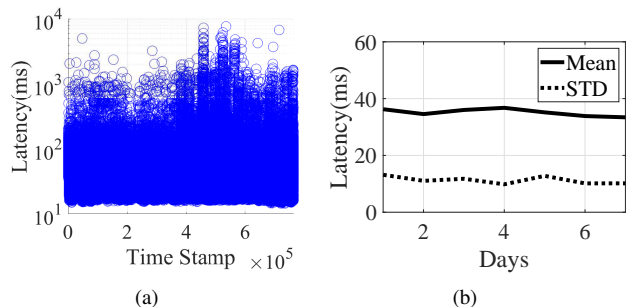


Fig. 2: (a) Latency samples recorded in a university campus listed in the sequence of time stamps. (b) Mean and STD of samples measured through a consecutive week in the same location.

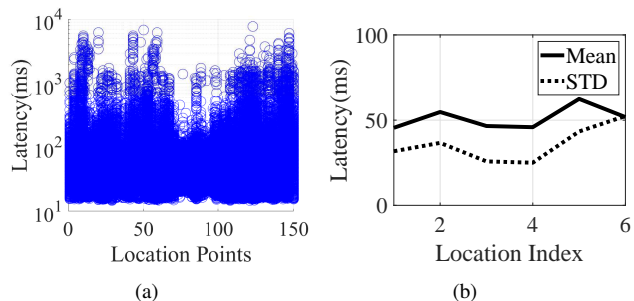


Fig. 3: (a) Latency samples recorded in a university campus listed in the sequence of location points. (b) Mean and STD of samples from 6 locations on a main driving route.

We adopt a dedicated smart phone app using Android API to periodically ping the first node and record the RTT for both data delivery and acknowledgment. Fig. 1 shows the measurement routes and traces of our dataset. We consider the RTT as the main metric for interactive wireless access latency for an LTE-supported connected vehicular system. Existing works as well as our own observation have already shown that even two consecutive measurements of the RTT at the same location can vary significantly. In addition, the temporal and spatial correlation of the instantaneous RTTs are often negligible. Fortunately, the statistical features such as mean and standard deviation (STD) remain relatively stationary. In Figs. 2 and 3, we present the mean and STD of RTT

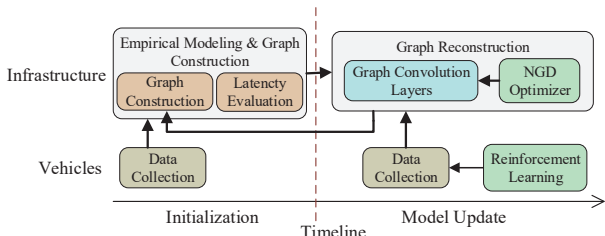


Fig. 4: Model Architecture.

samples collected at different location points throughout the main driving route of a university campus (see Fig. 3(b)) as well as these collected at the same university lab location over a consecutive week (see Fig. 2(b)). We can observe that compared to instantaneous latency samples, the mean and STD vary relatively slow according to different time and location. Also, the spatial variation causes a more noticeable impact on the mean and STD, compared to the temporal variation.

In this work, we plan to exploit the spatial and temporal correlation of the statistics of RTTs and model the vehicular network as a graph in which each vertex corresponds to a specific subregion and the edge connecting two vertices represents the statistical distance of the PDFs of RTTs between two connecting locations.

To characterize the temporal correlation of the graph, we consider a slotted process and assume the graphical model of latency statistics within each time slot can be considered to be fixed. Motivated by the fact that the temporal variation of the statistical features of the RTT at the same location often changes in a much slower pace than the statistical difference between different locations, we adopt a reinforcement learning-based approach to sequentially select a subset of subregions at the beginning of each time slot to collect samples and then exploit a GCN-based approach to reconstruct the update statistical features of all the location points in the new time slot. The reconstructed model will then be evaluated and compared with the real RTTs collected during the rest of the entire time slot. The evaluation results will then be used to update the model in future time slots.

B. Architecture Overview

We propose *SMART*, a novel architecture for modeling and keeping track of spatial and temporal statistics of wireless access latency between connected vehicles and wireless infrastructure across a large geographical area. The proposed architecture consists of three major components: data collection, empirical modeling and graphical model construction, model update and reconstruction, as illustrated in Fig. 4. We give a more detailed exposition of each components as follows.

Data Collection: We consider a connected vehicular system in which each vehicle is connected to a commercial LTE network owned by a mobile network operator while driving throughout an area of consideration. The RTTs of data packet delivered to the first wireless infrastructure node as well as receiving the feedback are recorded and reported to a central server. Note that it is not necessary for the central server to constantly collect RTT samples from all the vehicles. The server only needs to request a limited number of RTT samples

from vehicles located in a carefully selected subset of regions at the beginning of each time slot.

Empirical Modeling and Graph Construction: As observed in Section II-A, the statistical features of RTTs collected at different locations can exhibit strong spatial correlation. We adopt a statistical distance that can be used to calculate the confidence of the latency performance at each location. We can therefore establish an empirical graphical model.

Graph Reconstruction: To deal with the temporal variance and maintain a real-time update of the graphical model, we consider a slotted process. We use a reinforcement learning-based method to select a small subset of locations that collect samples at the beginning of each time slot so as to give more accurate graph reconstruction based on these samples. Then, we exploit a GCN-based approach to reconstruct the confidence of the whole graph vertices in the new time slot from s selected subset of subregions.

III. METHODOLOGY

In this section, we consider a commercial LTE network for connected vehicles across a university campus as an example to describe how to apply *SMART* to model and construct a spatial and temporal statistical modeling of latency performance of a V2I network. To make our description clearer, we first introduce the following notations and concepts. We model the roadways in the campus as an undirected graph $G = \langle \mathcal{V}, \mathcal{E}, \mathbf{X} \rangle$ where $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ is the set of vertices representing N locations; $e_{ij} \in \mathcal{E}$ is the edge between vertices with the weight characterized by the similarity (JS divergence) of latency distributions between vertex v_i and v_j , which will be discussed more thoroughly later in Section III-B. $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}^{N \times F}$ is the vertex feature matrix of the graph and F is the dimension of the feature vector of each vertex. $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacent binary matrix of graph G , i.e., for any $v_i, v_j \in V$, $A_{ij} = A_{ji} \in \{0, 1\}$ where 1 means that there is an edge between v_i and v_j and 0 otherwise. \mathbf{D} is the degree matrix whose element is $D_{ii} = \sum_j A_{ij}$.

A. Data Collection

Data collection will be conducted at the beginning of both initialization and the model updating process as shown in Fig. 4. We adopt a smartphone app, called *Delay Explorer*, that periodically pings the first node and record RTTs every 500 ms. It can also record data such as time stamps, GPS coordinates, and driving speed, among others. We next explain, in detail, the data collection procedure of data collection and model updating processes:

1) During the initialization process, RTT samples will be first collected throughout each considered area. The latency data is collected constantly until there are enough samples for each location to establish an empirical PDF. The PDFs will then be used in initial graph construction and latency performance evaluation.

2) During the model updating process, we consider a slotted process to cope with the temporal variation of latency performance and update the model slot by slot. At the beginning of each time slot, the model will carefully choose a subset

of locations to collect instantaneous RTT samples as will be discussed in Section III-C. We will evaluate the latency performance of the selected locations as well as their labels based on these data in the graph reconstruction procedure.

B. Empirical Modeling and Graph Construction

After collecting a sufficient number of samples, we can establish an empirical PDF for each location vertex in the graph. We adopt JS divergence to characterize the statistical correlation between two neighboring locations, i.e., weight of the edge connecting two vertices. Let $P_i(x)$ and $P_j(x)$ be the PDFs of latency at location i and j , respectively. The JS divergence can be written as follows:

$$\text{JS}(P_i(x)||P_j(x)) = \frac{1}{2}\text{KL}\left(P_i(x)||\frac{P_i(x)+P_j(x)}{2}\right) + \frac{1}{2}\text{KL}\left(P_j(x)||\frac{P_i(x)+P_j(x)}{2}\right), \quad (1)$$

where $\text{KL}(P_i(x)||P_j(x)) = \sum_{x \in X} P_i(x) \log \frac{P_i(x)}{P_j(x)}$. If the JS divergence $\text{JS}(P_i(x)||P_j(x))$ falls below a pre-defined threshold η , we can add an undirected link between locations i and j . In this way, we can then model the campus roadway network as an undirected graph.

As already discussed, we consider the time-varying statistical features of RTTs within a slotted time duration. The statistical feature we mainly focus on is the confidence, defined as the probability of a certain latency threshold can be satisfied, of the latency performance at each location in a multi-service scenario. We divide the locations into different classes according to their confidence for supporting some key vehicular services and give each location a preliminary label. To make our model more general, the number of classes could be flexibly assigned so that the evaluation can be adopted to various scenarios. We use the following statistical requirement as the latency performance metric:

$$\Pr(d_i \leq \tau) = \int_0^\tau P_i(x) dx \leq \varepsilon, \quad (2)$$

where $\Pr(\cdot)$ represents the probability of an event, and ε is the maximum confidence that must be guaranteed at the i th location. τ represents maximum tolerable latency for some specific vehicular services. According to the 5GAA [12], the latency requirement of major vehicular services can be roughly classified into 5 major use cases listed in Table I. The latency requirements of these services can be roughly divided into three classes according to Eq. (2):

- 1) Service Level 1: τ is 100 ms and ε is 99.99%;
- 2) Service Level 2: τ is 100 ms and ε is 99%;
- 3) Service Level 3: τ is 120 ms and ε is 99%.

It should be noticed that the above three service levels exhibit an inclusion relation: service satisfying level 1 (or level 2) requirement can also meet the requirement of level 2 (or level 3). For example, a location in Level 2 could satisfy the service requirement of a hazardous location warning, but it cannot satisfy the needs of a intersection assistance movement at a crossroad.

C. Graph Reconstruction

In a practical system, latency performance can be time-varying as shown in Section II. Thus, we need to keep the

TABLE I: Latency Requirement of 5 Different Services

Service Type	Service Level Latency	Reliability
Intersection movement	100 ms	99.99%
Awareness of the presence of vulnerable road user	100 ms	99.9%
Hazardous location warning	100 ms	99%
Cross-traffic left-turn assist	100 ms	90%
Emergency break warning	120 ms	99%

entire graph updated whenever some locations' latency performance as well as their supported service level change. To characterize the temporal correlation of the graph, we consider a slotted process and assume that the graphical model within each time slot can be assumed to be fixed. Due to the location proximity, the latency variation of two neighboring locations may experience similar changing patterns. We then introduce a GCN-based approach to recover the complete graph from a limited number of RTT samples collected throughout a subset of edges and vertices

1) *Graph Convolutional Network*: The standard convolution in CNN is not applicable to graphs due to their non-Euclidean structure. We therefore adopt GCN to reconstruct the graphical model whenever some parts of the graph change. According to [13], the computational complexity of spectral GCN is $O(n)$ where n is the number of graph edges. Hence, GCN-based approach can be directly applied into large graphs.

The propagation process of the stacking layers can be written as:

$$H^{(0)} = \mathbf{X} \text{ and } H^{(l+1)} = \sigma(\Delta \mathbf{A} H^{(l)} W_l), \quad (3)$$

where $\Delta \mathbf{A} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$ is a renormalized matrix with $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $H^{(l)}$ is the output of layer l , $\sigma(\cdot)$ is an activation function which is normally set as ReLU function. W_l is the learnable parameter matrix which can be obtained using gradient descent. The input feature vector \mathbf{x}_i can be some normalized latency samples of location point i .

To accomplish the semi-supervised classification task, we use the softmax activation function, defined as $\text{softmax}(\mathbf{x}_i) = \exp(\mathbf{x}_i) / \sum_i \exp(\mathbf{x}_i)$, on the output Z of the last convolutional layer and the cross-entropy error and the loss function L can be formulated as:

$$Z' = \text{softmax}(Z), \quad (4)$$

$$L = - \sum_{l \in y_l} \sum_{f=1}^F Y_{lf} \ln Z'_{lf}. \quad (5)$$

Z'_{lf} is entry f of the vertex's hidden representation labeled l . Y_{lf} is the ground truth of the corresponding label.

In order to accelerate the convergence speed of GCN for a large graph, we introduce a second-order gradient descent method called NGD [14] to optimize the parameter matrices in the training process of GCN. NGD transforms gradients into so-called *natural gradients* that have proved to be much faster compared to the stochastic gradient descent (SGD). Recently, the work in [15] used NGD for a semi-supervised classification task in GCN, and it showed encouraging results in both accuracy and convergence speed on some benchmark

Algorithm 1 Preconditioning using NGD

Input: Gradient of parameters ∇W_l for $l = 1, \dots, m$, adjacency matrix \mathbf{A} , degree matrix \mathbf{D} , training mask z , regularization hyper-parameters λ, ϵ

- 1: Derive the numbers of labeled and unlabeled vertices via $\bar{n} = \sum(z)$ and $n = \dim(z)$. And let $[\Delta a_{ij}]$ represent the entry of $\Delta \mathbf{A}$.
 - 2: **for** $l = 1, \dots, m$ **do**
 - 3: Formulate the feature aggregation process of each layer via $\tilde{x}_{l-1,i} = \sum_{j=1}^n \Delta a_{i,j} x_{l-1,j}$.
 - 4: Approximate matrices \mathbf{V}_l and \mathbf{U}_l via:
 $\mathbf{u}_{l-1,i} = \partial L / \partial \mathbf{x}_l \odot \sigma_l(W_l \tilde{x}_{l-1,i})$,
 $\mathbf{U}_l = \sum_{i=1}^n (z_i + (1 - z_i) \lambda) \mathbf{u}_{l-1,i} \mathbf{u}_{l-1,i}^\top / (n + \lambda \bar{n})$,
 $\mathbf{V}_l = \sum_{i=1}^n (z_i + (1 - z_i) \lambda) \tilde{x}_{l-1,i} \tilde{x}_{l-1,i}^\top / (n + \lambda \bar{n})$.
 - 5: **Output:** $(\mathbf{V}_l + \epsilon^{-1/2} \mathbf{I})^{-1} \nabla W_l (\mathbf{U}_l + \epsilon^{-1/2} \mathbf{I})^{-1}$
 - 6: **end for**
-

datasets.

Preconditioning is inspired by the idea that capturing the relation between the gradient of parameters before optimization will help with convergence. For example, the traditional optimizer, such as Adam [16], uses diagonal preconditioner which neglects the pair-wise relation between gradients. However, any extra information about gradients is often impossible or hard to obtain. Motivated by NGD, we introduce a preconditioning algorithm that uses the second moment of gradient to approximate the parameters' Fisher information matrix in the prediction distribution [15].

Algorithm 1 shows the detailed preconditioning process for modifying gradients of each layer at any iteration. The gradients are first transformed using two matrices, \mathbf{V}_l^{-1} and \mathbf{U}_l^{-1} , then sent to the optimization algorithm for parameter updating. Let m be the number of the network layers and \odot be element-wise multiplication operation. $x_{l-1,i}$ represents the output feature vector of v_i in layer $l - 1$ and is updated into $\tilde{x}_{l-1,i}$ using a renormalization trick for $i = \{1, \dots, N\}$. λ is a hyper-parameter that controls the cost of predicted labels and ϵ is a regularization hyper-parameter to evaluate \mathbf{V}_l^{-1} and \mathbf{U}_l^{-1} .

2) *Deep Q-Networks:* As mentioned earlier, the latency performance (i.e. the label of vertices in graph) of each location point can change at different time slots. Always collecting sufficient numbers of samples across all the possible locations is generally impossible. Due to the spatial correlation of the latency graph, collecting a subset of locations will be sufficient to reconstruct the complete updated latency graph. Motivated by the fact that the reconstruction accuracy of a graph model can vary significantly with different sets of selected vertices, in the rest of this section, we formulate the vertices selection for graph reconstruction as a Markov decision process (MDP) defined as follows.

State Space \mathcal{S} is a finite set of possible service levels that can be supported at each location. $\mathbf{S}_t \in \mathbb{R}^{K \times N}$ is a $K \times N$ matrix in time slot t where column vector \mathbf{S}_t^i represents the probability for each K labels in next time slot $t + 1$ of vertex v_i . Both prior and conditional probability can be obtained

from historical data.¹

Action Space \mathcal{A}_v is the possible selection of location subsets for requesting latency samples. We write $a_v^t = \{v_1, v_2, \dots, v_m\}$ as an instance of action in time slot t for $a_v \in \mathcal{A}_v$. $\{v_1, v_2, \dots, v_m\}$ are m vertices selected from all N vertices in the graph.

State Transition function $T : \mathcal{S} \times \mathcal{A}_v \times \mathcal{S} \rightarrow [0, 1]$ denotes the probability of state transiting from one state to another. When the selection of m vertices in slot t (i.e., the action a_v^t) is determined, we can observe the actual label c_i^t for each m locations. We can then establish as a mapping function $f : \mathcal{S}_{t+1} = f(\mathbf{S}_t, a_v^t)$, where the m column vectors with probability $[\Pr(c_1^{(t+1)} | c_i^t), \Pr(c_2^{(t+1)} | c_i^t), \dots, \Pr(c_K^{(t+1)} | c_i^t)]^T$ for state \mathcal{S}_t to transit into \mathcal{S}_{t+1} .

Reward Function R : We try to maximize the graph construction accuracy, defined as the percentage of the correctly predicted vertices among all the reconstructed graph, i.e., we have $R_t(\mathbf{S}_t, a_v^t) = \frac{1}{N-m} \sum_{v \in \mathcal{V} \setminus a_v^t} \mathcal{I}(c(v) = c_v)$ where $c(v)$ is the predicted label and c_v is the true label of vertex v . $\mathcal{I}(\cdot)$ is an indicator function to count the correct prediction.

We focus on maximizing the long-term reconstruction accuracy including both current and future rewards defined as $Q(\mathbf{S}_t, a_v^t)$ when action a_v^t is taken at state \mathcal{S}_t :

$$Q(\mathbf{S}_t, a_v^t) = R_t(\mathbf{S}_t, a_v^t) + \beta Q(\mathbf{S}_{t+1}, a_v^t), \quad (6)$$

where β is the learning rate.

Following the standard procedure of DQN, we can write the optimal policy π^* as:

$$\pi^* = \arg \min_{a_v^t \in \mathcal{A}_v^{\mathcal{S}_t}} Q(\mathbf{S}_t, a_v^t) \quad (7)$$

where $Q(\mathbf{S}_t, a_v^t)$ can be pre-calculated and pre-stored in a look-up table (i.e., Q-table) for finding the expected reward under all possible state and action pairs which needs to be stored in each time slot which leads to enormous storage and computational complexity. To address the above problems, DQN uses deep neural networks to estimate the Q-table. The transition $(\mathbf{S}_t, a_v^t, R_t(\mathbf{S}_t, a_v^t), \mathbf{S}_{t+1})$ is stored in the experience relay pool for learning process. During each epoch, the predict network will choose an action which will be evaluated in the target network.

IV. SIMULATION RESULTS AND ANALYSIS

In this section, we evaluate the performance of *SMART* through extensive simulations using the dataset collected in a university campus. Our simulations are performed mainly using two open-source Python libraries, Pytorch and Pytorch Geometric, on a workstation with an Intel(R) Core(TM) i9-9900K CPU@3.60GHz, 64.0 GB RAM@2133 MHz, 2 TB HD, and two NVIDIA Corporation GP102 [TITAN X] GPUs.

We consider 150 subregions across the university campus and randomly choose 30 samples in each subregion to construct the feature vector for each vertex. We train GCN models

¹We use a one-month latency collection dataset measured at a university campus. For example, we divide the latency data d_i at location i into $\{d_i^1, d_i^2, \dots, d_i^p\}$ sequentially according to p time slots and set the label in each slot based on the confidence of latency performance. We can then obtain the conditional probability $\Pr(c_j^{(t+1)} | c_i^t)$ for each location where c_i^t means the location is in label c_i at time slot t .

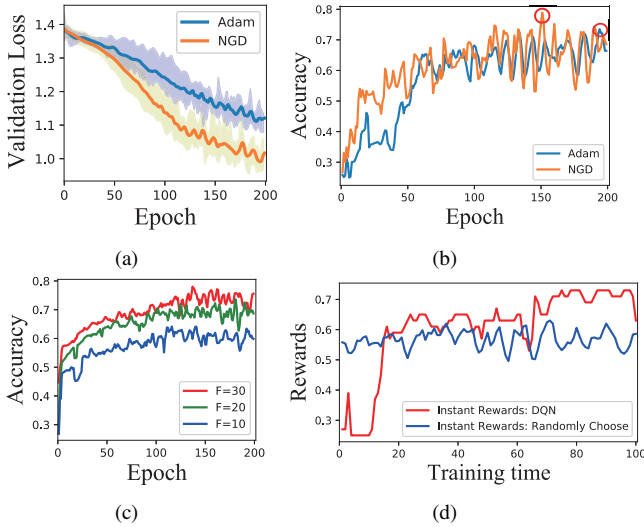


Fig. 5: (a) Validation costs. (b) model accuracy based on test dataset, (c) model accuracy with different input feature vectors, and (d) rewards under different number of training iterations.

in 200 epochs (training iterations) using both Adam [16] and NGD with learning rate 0.01. The Adam is used with the weight decay of 5×10^{-4} and the momentum of 0.9. A 2-layer GCN with a 16-dimension hidden variable is used in all simulations. The first layer is followed by a drop out function at the rate of 0.5. The training process stops if the validation loss (i.e., the value of loss function on validation set which is used to determine the hyper-parameters in the model) does not decrease for 10 consecutive epochs and the loss function is evaluated using the negative log-likelihood in equation. (5).

We compare the validation loss and testing accuracy of two optimization methods over 10 rounds in Fig. 5(a). The blue and yellow zones are confidence intervals of Adam and NGD, respectively. We can observe that the validation loss in NGD drops faster than Adam and could result in a lower validation loss. In Fig. 5(b), we compare the model accuracies of Adam and NGD based on our testing dataset. We can observe that GCN with an NGD optimizer can reach a maximum accuracy at 78.86% over 200 epochs (the red circles in the figure) which outperforms the maximum accuracy of Adam optimizer at 74.43%. This result demonstrates that the NGD offers faster processing speed compared to traditional optimizer such as standard SGD to reconstruct graph model.

We present model accuracy with different numbers of dimensions F of the input feature vectors. We can observe that the larger dimensions of feature vectors could result in a higher accuracy as shown in Fig. 5(c). This is because the feature vector with respect to latency samples can offer more information about the latency performance. Fig. 5(d) presents the rewards achieved by the actions selected by DQN under different number of training iterations. We can observe that the reward achieved by selecting locations decided by DQN is always higher than that achieved by adopting random selection of locations at each time slot.

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed *SMART*, a novel framework for modeling and keeping track of spatial and temporal

statistics of vehicle-to-infrastructure communication latency across a large geographical area. *SMART* can be directly applied to characterize the feasibility of supporting different latency-sensitive services across a large geographical area during different time periods. Specifically, *SMART* first formulates the spatio-temporal performance and correlations of a vehicular network as a graphical structure and then adopt GCN and DQN to reconstruct the spatial and temporal latency performance in a slotted process. Simulation results show that the proposed method can improve both the modeling accuracy and reconstruction efficiency for large vehicular networks.

Our work opens several potential directions that worth further investigating. In particular, it will be promising to extend *SMART* into a more general setting. In addition, it is also interesting to consider some other information to be included into the edge weights and input feature vector of GCN that can capture more complex correlations between vertices.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 62071193 and 61632019, the Key R & D Program of Hubei Province of China under Grant 2020BAA002, and China Postdoctoral Science Foundation under Grant 2020M672357.

REFERENCES

- [1] IHS Markit, "A day in the life of the "car of the future"." [Online]. Available: <https://cdn.ihsmarkit.com/www/pdf/1219/414806070-1019-YJ-AUT-CarOfTheFuture-I>
- [2] Y. Xiao and M. Krunz, "Distributed optimization for energy-efficient fog computing in the tactile internet," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2390–2400, Nov. 2018.
- [3] B. Warwick, N. Symons, X. Chen, and K. Xiong, "Detecting driver drowsiness using wireless wearables," in *IEEE MASS*, Dallas, TX, Oct. 2015.
- [4] F. Kortmann, H. Peitzmeier, N. Meier, J. Heger, and P. Drews, "Enabling road condition monitoring with an on-board vehicle sensor setup," in *IEEE SENSORS*, Montreal, QC, Canada, Oct. 2019.
- [5] R. W. Wolcott and R. M. Eustice, "Visual localization within lidar maps for automated urban driving," in *IEEE/RSJ IROS*, Chicago, IL, Sept. 2014.
- [6] Y. Xiao, M. Krunz, H. Volos, and T. Bando, "Driving in the fog: Latency measurement, modeling, and optimization of lte-based fog computing for smart vehicles," in *IEEE SECON*, Boston, MA, Jun. 2019.
- [7] R. Xia, Y. Xiao, Y. Li, M. Krunz, and D. Niyato, "A generative learning approach for spatio-temporal modeling in connected vehicular network," in *IEEE ICC*, Dublin, Ireland, Jun. 2020.
- [8] Y. Xiao, G. Shi, Y. Li, W. Saad, and H. Vincent Poor, "Toward self-learning edge intelligence in 6g," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 34–40, Dec. 2020.
- [9] B. Toghi, M. Saifuddin, M. O. Mughal, and Y. P. Fallah, "Spatio-temporal dynamics of cellular v2x communication in dense vehicular networks," in *IEEE CAVS*, Honolulu, HI, Oct. 2019.
- [10] T. Zeng, O. Semiari, and W. Saad, "Spatio-temporal motifs for optimized vehicle-to-vehicle (v2v) communications," in *ICNC*, Maui, HI, Jun. 2018.
- [11] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting," Jul. 2018. [Online]. Available: <http://arxiv.org/abs/1709.04875>
- [12] G. WhitePaper, "5gaa c-v2x use cases methodology, examples and service level requirements," 5GAA Whitepaper, June 2019.
- [13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, Toulon, France, April. 2017.
- [14] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.
- [15] M. R. Izadi, Y. Fang, R. Stevenson, and L. Lin, "Optimization of graph neural networks with natural gradient descent," Aug. 2020. [Online]. Available: <https://arxiv.org/abs/2008.09624>
- [16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv*, Dec. 2017. [Online]. Available: <https://arxiv.org/abs/1412.6980>