Edge Computing-Empowered Large-scale Traffic Data Recovery Leveraging Low-rank Theory

Chaocan Xiang, Zhao Zhang, Yuben Qu, Dongyu Lu, Xiaochen Fan, Panlong Yang, and Fan Wu

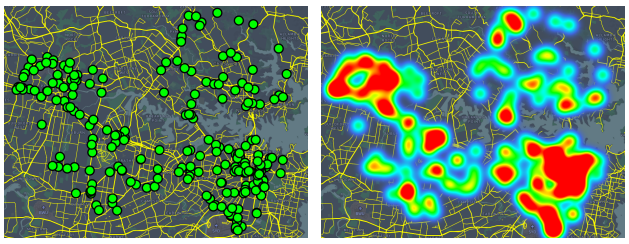
Abstract—Intelligent Transportation Systems (ITSs) have been widely deployed to provide traffic sensing data for a variety of smart traffic applications. However, the inevitable and ubiquitous missing data potentially compromises the performance of ITSs and even undermines the traffic applications. Therefore, accurate and real-time traffic data recovery is crucial to ITSs and its related services especially for large-scale traffic networks. To leverage the characteristics in transportation networks for data recovery, we first conduct experimental explorations on a large-scale traffic dataset of an ITS and further quantify the spatiotemporal correlations of traffic data. Inspired by the observation results, we propose *GTR*, an edGe computing-empowered system for large-scale Traffic data recovery with low-Rank theory. *GTR* leverages decentralized computing power of edge nodes to process massive traffic data from hundreds of traffic stations for accurate and real-time recovery. Specifically, we first propose a suboptimal edge node deployment algorithm with theoretical performance guarantee, by exploiting the supermodularity in the NP-hard joint-optimization problem. Furthermore, to leverage the low-rank nature of traffic data, we transform the data recovery problem into a low-rank minimization problem, and exploit fixed point continuation iterative scheme to capture spatiotemporal dynamics for accurate data recovery. Finally, the extensive trace-driven evaluations show that, *GTR* only needs at most 6% extra total cost compared to the optimal deployment, while outperforming three other baseline methods by 62.1% improvement in terms of traffic data recovery accuracy.

Index Terms—Edge computing; Intelligent transportation system; Edge node deployment; Traffic data recovery; Low-rank theory.

1 Introduction

With the rapid development of urbanization, cities are facing many challenges in dealing with their growing populations and vehicles, especially in transportation [1]. Hence, large numbers of Intelligent Transportation Systems (ITSs)–such as Advanced Traffic Management System and Adaptive Traffic Control System–are developed in recent years to solve the transportation issues [2]. For example, as illustrated in Fig. 1, the transportation agency of New South Wale (NSW), Australia built a Traffic Volume Viewer System (TVVS) [3]. More than 600 traffic collection stations are deployed in TVVS to monitor real-time traffic volume at most of main roads in NSW [4]. However, according to the experimental observations on the TVVS in Sec. 2.1, this system is subject to a highly serious issue of missing traffic data, *e.g.*, more than 25% missing rate for 60% stations. Indeed, this issue exists widely in many existing ITSs systems [5], [6]. Thus, accurate and real-time recovery of traffic data in large-scale ITSs is essential for realization of intelligent transportation in smart cities.

To address the problem of traffic data recovery, we conduct the experimental explorations based on a large-scale traffic volume dataset of TVVS in Sec. 2.2. The results indicate that the traffic data[1] has both temporal and spatial correlations at different time and stations, thus providing a promising opportunity for traffic data recovery. Hence, it is auspicious to jointly exploit large amounts of traffic data from multiple stations on much time for accurate real-time recovery. However, it poses a difficult dilemma of practical implementation with three following reasons. First, it needs large overhead of computation and storage for resource-intensive traffic data, such as real-time traffic videos [8]. Second, owing to massive deployments with limited budget, any individual station with limited capabilities of computation and storage cannot undertake such a heavy responsibility [9]. At last, if offloading all traffic data to the remote powerful cloud, the incurred latency would be



(a) Locations of traffic stations (b) Heat map of distribution

Fig. 1: Illustrations of the traffic volume viewer system with 600 traffic collection stations deployed in New South Wale.

- *Chaocan Xiang and Zhao Zhang are with the College of Computer Science, Chongqing University, Chongqing, China, 400044. E-mail: xiang.chaocan@gmail.com; vvfgfg@cqu.edu.cn.*
- *Yuben Qu, Dongyu Lu and Fan Wu are with Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. Email: quyuben@sjtu.edu.cn; sjtuldy@sjtu.edu.cn; fwu@cs.sjtu.edu.cn.*
- *Xiaochen Fan is with the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. E-mail: xiaochen.fan@student.uts.edu.au.*
- *Panlong Yang is with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China, 230026. E-mail: panlongyang@gmail.com.*

- *This research is supported by NSF China under Grants No. 61872447, 61702525. Chaocan Xiang and Yuben Qu are the corresponding authors.*

1. In this work, we take the recovery of traffic volume data as a typical example, which is a greatly fundamental traffic data for most applications of ITSs [7]. Hence, in the remaining paper, we will use the terms '*traffic data*' and '*traffic volume data*' interchangeably unless otherwise stated.

intolerable, as a result of long distance communication and huge traffic volume in large-scale ITSs [10].

To resolve this dilemma, we propose an edge computing-empowered large-scale traffic data recovery system, by deploying edge nodes in physical proximity to traffic stations for real-time recovery [9], [11]. Leveraging the decentralized computing power of edge nodes, it tackles not only the insufficient capability issue when recovering traffic on individual station, but also the high latency for centralized computation on cloud server [10]. Nevertheless, it is non-trivial to realize this system with two following challenges.

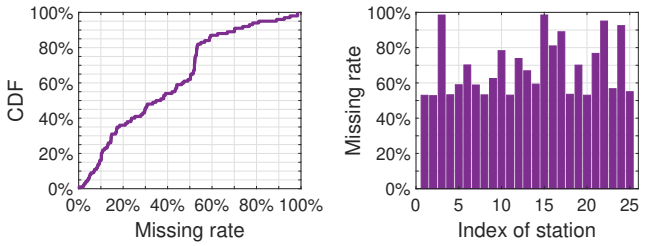
- *Optimal deployment of edge nodes for ITSs*: the edge node deployment should be jointly optimized with the traffic data collection in ITSs, while minimizing the overall cost of edge deployment and traffic collection. It is a Mixed Integer Linear Program (MILP) problem, which is proven to be NP-hard in Sec. 4.1.
- *Accurate traffic recovery with spatiotemporal dynamic correlations*: Though the experimental observations illustrate traffic data has the spatiotemporal correlations, such relationships are non-linear and space-time-varying, hence rendering accurate data recovery extremely tough even given the optimal edge deployment.

To address these two challenges, we propose an edGe computing-empowered large-scale Traffic data recovery system leveraging low-Rank theory, called ***GTR***[2]. It consists of two key modules as follows. 1) *Suboptimal deployment of edge nodes with performance guarantee*. We first find that, given any fixed edge node deployment, the traffic data collection is a linear program problem (LP). Therefore, we reformulate the optimal deployment problem as a set function optimization one, subject to only the variable of edge node deployment. Second, despite the implicit expression of the objective function in this set function optimization, we theoretically prove it is non-negative supermodular. Finally, we propose a local search-based edge node deployment algorithm, exploiting the supermodularity theory to obtain guaranteed suboptimal solution. 2) *Accurate traffic data recovery based on low-rank theory*. We conduct Singular Value Decomposition (SVD) based on experiments to investigate whether the matrix of traffic data is approximately low rank in terms of the spatiotemporal dimensions. Moreover, based on this positive result, we equivalently transform the intractable problem of traffic data recovery into a low-rank minimization one, then transform it to a convex optimization problem. At last, we use the Fixed Point Continuation (FPC) iterative scheme to achieve accurate recovery with minimal low rank. Both theoretical analyses and trace-based evaluations are conducted to evaluate the performance of *GTR*.

In summary, this paper makes four main contributions:

1) We make the experimental explorations based on a large-scale traffic dataset of massive traffic stations. Inspired by the observations of spatiotemporal correlations, we propose the traffic data recovery system empowered by edge computing, thereby resolving the dilemma between traffic stations and cloud in large-scale ITSs.
2) We present a suboptimal edge node deployment scheme with theoretical performance guarantee, leveraging the equivalent reformulation and the supermodularity to effectively decouple the NP-hard problem of joint optimization.
3) We propose a low rank-based traffic data recovery algorithm based on the experimental observations of SVD, exploiting the low-rank minimization to successfully tackle the spatiotemporal dynamic of correlations.
4) We conduct extensive experiments based on large-scale traffic dataset with 100 traffic stations. The results show that *GTR* only needs at most 6% extra total cost compared to the optimal deployment, while outperforming three other baseline methods by 62.1% improvement in terms of traffic data recovery accuracy.

2. Similar to Nissan GT-R vehicle with powerful engine and high reliability, our system can provide powerful computing capability empowered by edge computing, and achieve highly accurate data recovery based on low-rank theory in ITSs.



(a) CDF of 100 stations (b) Result of each station

Fig. 2: Analysis of missing traffic data in 100 stations of TVVS. The missing rate is between 25% and 98% for 60% stations.

The rest of this paper is organized as follows. First, the motivations based on experimental explorations are introduced in Sec. 2. We then state the system model and formalize the problem in Sec. 3. We also propose an edge computing-based large-scale traffic volume recovery system called *GTR* along with theoretical analyses in Sec. 4. In Sec. 5, we conduct traces-driven evaluations, followed by reviewing the related work in Sec. 6. Finally, we conclude this work in Sec. 7.

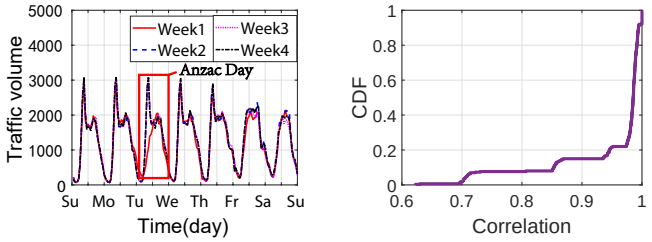
2 Motivations

In this section, we first conduct experiments to explore the issue of missing traffic data, followed by the spatiotemporal correlations. Finally, we analyze the dilemma of implementations for traffic data recovery in large-scale ITSs.

2.1 Uncovering Missing Data Issue in Large-scale ITSs

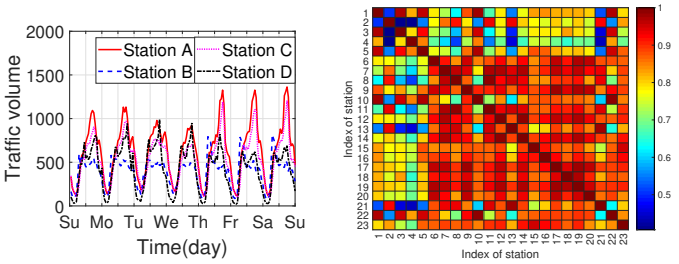
Traffic volume monitoring is of fundamental importance for ITSs, as it is essential for road navigation, congestion management and vehicles' emission monitoring [7], [12], [13]. For instance, as shown in Figs. 1a and 1b, the Roads & Maritime Services of New South Wale (NSW) established a Traffic Volume Viewer System (TVVS) by deploying over 600 traffic collection stations [3]. This system is monitoring the traffic volume at most of main roads across NSW from 2006 up to now [4].

Although a number of real systems are deployed for monitoring traffic volume, such as TVVS, most of them suffer from the severe issue of missing data, due to detector malfunction, loss of data in transmission and power outage,



(a) Comparison of different weeks (b) CDF of correlations

Fig. 3: Analysis of temporal correlation on traffic volume data in different weeks for the same traffic station.



(a) Comparison on various stations (b) Confusion matrix of correlations

Fig. 4: Analysis of spatial correlation on traffic volume data in different traffic stations.

etc [6], [14]. For example, we randomly select 100 traffic stations of TVVS and conduct statistics based on their traffic volume data. As illustrated in Fig. 2a, the missing rate is more than 5% for 90% of traffic stations, while it is beyond 25% for more than 60% of stations. Even worse, the missing rate of more than 10% stations is above 70%. Moreover, Fig. 2b shows the missing rate of 25 traffic stations, and the results indicate that the missing rate of several stations is up to 98%. Also, this issue of missing traffic data is very common in real ITSs, such as about 10% missing rate in the ITSs of Beijing city [5].

To sum up, *traffic volume data is fundamentally important for the ITSs, while many existing systems are subject to a greatly serious issue of missing traffic volume data. Thus, accurate real-time traffic volume recovery in large-scale ITSs is crucial to the realization of intelligent transportation.*

2.2 Experimental Explorations of Spatiotemporal Correlations on Traffic Data

To address the issue of missing traffic data in ITSs, we make extensive experiments to explore both temporal and spatial correlations on traffic volume data. Specifically, we collect a dataset of traffic volume from TVVS [3] in 25 traffic stations of Sydney for one year (*i.e.,* Jan.-Dec., 2018). The sampling interval is 1 hour.

1) **Analysis of temporal correlation**: we analyze the correlation of traffic volume data in terms of temporal dimension on a traffic station. More specifically, we divide one year into 52 weeks, while analyzing the correlations among the traffic volume of different weeks in one station. As illustrated in Fig. 3a, we only plot the traffic volume data of four weeks, due to the size limitation of the figure and similar results of other weeks. The experimental results indicate that the traffic volume exhibits a similar pattern in each week. Moreover, the patterns of weekends are different from those of weekdays, since the commuting activities of most citizens on weekdays (such as working) are distinguished from the ones on weekdays (*e.g.,* shopping) [15]. The above results demonstrate that the traffic volume data has the temporal correlation in the period of not one day but one week.

Furthermore, by using the Pearson correlation coefficient, we quantify the correlations of traffic volume data between any two of these 52 weeks. As shown in Fig. 3b, their temporal correlations are more than 0.6 for 100%, and above 0.95 for about 80%. As a result, *the traffic volume data has a strong temporal correlation on each week for one traffic station.* However, as shown in Fig. 3a, there exist abnormal patterns on some days, such as the Tuesday of the first week in April. It is because this Tuesday is a special holiday in Australia (*i.e.,* the Anzac Day of 2018) and the citizens are off duty on that weekday [16]. Thus, *the periodicity of traffic volume data on the temporal dimension is always affected by the social events, holidays and extreme weather conditions, etc.*

2) **Analysis of spatial correlation**: we analyze the spatial correlations of traffic volume data in different traffic stations. Firstly, we compare the traffic volume data in different traffic stations. As illustrated in Fig. 4a, we only show the data of four stations on the first week owing to the similar results. It demonstrates that the traffic volume of different stations has approximately similar pattern even on the special holiday (*e.g.,* the Tuesday).

Further, we exploit the Pearson correlation coefficient to quantify the spatial correlations of traffic data between different stations. As shown in Fig. 4b, we use the confusion matrix to represent the correlations between any two of the 23 traffic stations. The experimental results demonstrate that the correlations among most stations are more than 0.8. As a result, *the traffic volume has spatial correlations among different traffic stations*. However, Fig. 4b illustrates that a few stations (*e.g.,* stations 1, 2, 3 and 4) have lower correlations with others. The reason is that the traffic stations in the city are interconnected by the roads. Hence, the traffic volume of different stations are suffered from the same influences, *e.g.,* the rush hours, holidays, social events and weathers, *etc.* Moreover, the longer the distance of the road network between two stations is, the lower the correlation between them is [15].

In summary, *the traffic volume data has both the temporal correlation in the period of one week and the spatial correlations among different traffic stations*.

2.3 Dilemmas of Implementation for Large-scale Traffic Recovery

The experimental explorations in Sec. 2.2 indicate that the traffic volume data has the temporal and spatial correlations on different time and stations. Thus, we can jointly utilize the traffic data of massive stations across the same time to recover missing data. Nevertheless, the implementation of the traffic data recovery in large-scale ITSs is facing a dilemma for the following three challenges.

1) *Large computation & storage overhead for resource-intensive traffic data*. Many ITSs use traffic cameras to monitor the traffic volume on the roads in real-time [17]. Hence, the real-time and resource-intensive traffic video data

incurs numerous overhead on its computation and storage [18].

2) *Limited capability of computation & storage in individual station*. As the ITSs should cover a large-scale area, such as a big city, a large number of traffic stations need to be deployed at different roads, *e.g.,* more than 600 stations in TVVS [4]. Hence, most stations have a limited capability of computation and storage, owing to large-scale deployments with a limited budget.
3) *High latency of data transmission in large-scale ITSs*. The transmission delay is very long due to the resource-intensive traffic data (such as videos) and the limited bandwidth of communication network (*e.g.,* wireless network and cellular one). Even worse, this issue will be significantly exacerbated, as a result of highly long communication distance from the traffic stations to the center server in the large-scale ITSs.

In brief, it is difficult to recover missing traffic data in individual station, due to the large overhead of computation and storage for resource-intensive traffic data as well as the insufficient capability of single station. On the other hand, recovering on the cloud server is also challenging, owing to the conflict between high transmission latency and real-time requirements in ITSs. As a result, *regarding the computation, storage and transmission, it is greatly challenging to conduct traffic data recovery on both the individual station and the center server in large-scale ITSs*.

3 System Model & Problem Formulation

3.1 System Model of Edge Computing

According to the experimental observations in Sec. 2.1, as the ITSs have large amounts of missing traffic data, its accurate recovery is essentially critical for the ITSs. Moreover, the experimental explorations in Sec. 2.2 reveal that the traffic volume data has strong temporal-spatial correlations, which can be harnessed for accurate recovery of traffic data. Accordingly, we jointly use the traffic data of massive roads across time for accurate data recovery. However, it requires numerous latency for transmission and computing due to large-scale coverage and large volume of traffic data (*e.g.,* real-time traffic videos) in the ITSs. As a result, as shown in Fig. 5, we propose a traffic volume recovery framework based on edge computing to achieve low-delay, highly-accurate recovery in the large-scale ITSs. Specifically, this framework is mainly comprised of traffic stations, edge nodes and a central server as follows.

- **Traffic stations**: In the ITSs, traffic monitoring systems are deployed on each traffic station for traffic sensing of one road segment, such as traffic cameras [19]. Let r_i denote the i-th traffic station, *i.e.,* $i \in \{1, \ldots, N\}$. We assume the sensing interval is T, and the vector of sensing times is represented as $\mathbf{T} = \{1, \ldots, T\}$. Moreover, we use $v_i(t)$ to denote the traffic data of r_i at time t, *i.e.,* $t \in \mathbf{T}$. Accordingly, $\mathbf{v}_i = \{v_i(t) | 1 \leq t \leq T\}$ denotes the set of traffic data on r_i, while its data size is represented by w_i. Let $\mathbf{V} = \{\mathbf{v}_i | 1 \leq i \leq N\}$ denote the set of the traffic data from all the traffic stations (*i.e.,* $r_i, \forall i \in \{1, \ldots, N\}$) within T.
- **Edge nodes**: All the traffic data is transmitted into nearby edge nodes deployed on certain traffic stations for real-time recovery. We assume that there are S edge nodes, and let e_s denote the s-th edge node, *i.e.,* $s \in [1, \ldots, S]$. Furthermore, x_{js} indicates whether e_s is deployed on r_j, *i.e.,* $x_{js} = 1$ if yes, otherwise $x_{js} = 0$. The different edge nodes have different capacities of computation and storage due to device diversity. Hence, we let c_s denote the capacity of e_s. Also, the deployment cost of edge nodes changes with the deployed traffic stations. Let d_{js} denote the cost of e_s deployed on r_j. y_{ij} denotes the proportion of the traffic data of r_i assigned to the edge node deployed at r_j, *i.e.,* $\forall i, j \in [1, \ldots, N]$, $y_{ij} \in [0, 1]$. As the communication in different paths induces different cost, we let b_{ij} denote the communication cost of the unit traffic data transmitted from r_i to r_j.
- **Cloud server**: It plays two critical roles in this edge computing-based system as follows. First, it is connected to all traffic stations and sets up a control management to provide flexible and efficient communications, including both traffic data and control information among multiple traffic stations as well as edge nodes [11]. Second, it can provide further analysis after traffic data recovery over a large scale, which needs more powerful capacity of computation and storage than the edge nodes [9], [10].

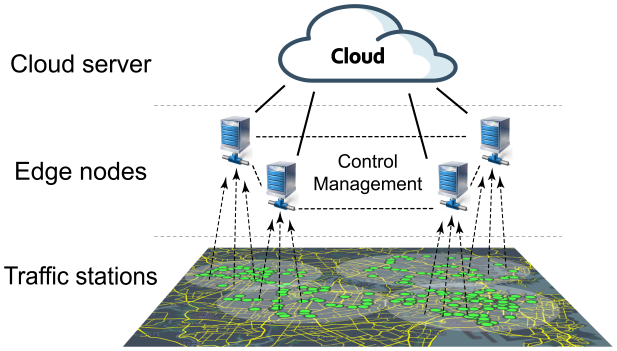


Fig. 5: Framework of edge computing-based traffic data recovery system in large-scale ITSs

3.2 Problem Formalization

Based on the above system model of edge computing-based traffic data recovery, the research problem is chiefly composed of two following sub-problems.

1) **Sub-problem A**: (*Optimal deployment of edge nodes*) Given the capacities of the edge nodes (*e.g.,* c_s), i) how to place these S edge nodes (*e.g.,* e_s) on the traffic stations (*e.g.,* r_j), *i.e.,* $\mathbf{x} = \{x_{js} | 1 \leq j \leq N, 1 \leq s \leq S\}$; and ii) how to allocate traffic data of r_i into the edge nodes deployed at r_j, *i.e.,* $\mathbf{y} = \{y_{ij} | \forall i, j \in \{1, \ldots, N\}\}$, so as to serve all the traffic data (*i.e.,* $\mathbf{V}$), while minimizing the overall cost $\Omega(\mathbf{x}, \mathbf{y})$ of data communication and edge deployment. Formally,

$$\min_{\mathbf{x},\mathbf{y}} \quad \Omega(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i y_{ij} b_{ij} + \sum_{s=1}^{S}\sum_{j=1}^{N} x_{js} d_{js}, \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{N} y_{ij} = 1, \forall i \in \{1,\ldots,N\}, \tag{2}$$

$$\sum_{j=1}^{N} x_{js} \leq 1, \forall s \in \{1,\ldots,S\}, \tag{3}$$

$$\sum_{i=1}^{N} w_i y_{ij} \leq \sum_{s=1}^{S} x_{js} c_s, \forall j \in N, \tag{4}$$

$$y_{ij} \leq l_{ij}, \forall i,j \in \{1,\ldots,N\}, \tag{5}$$

$$x_{js} \in \{0,1\}, y_{ij} \in [0,1], \forall i,j \in \{1,\ldots,N\}, \tag{6}$$

where $\sum_{i=1}^{N}\sum_{j=1}^{N} w_i p_{ij} b_{ij}$ in Eq. (1) denotes the communication cost for transmitting all the data from $r_i (\forall i \in [1,\ldots,N])$ to the edge nodes deployed on $r_j (\forall j \in [1,\ldots,N])$. $\sum_{s=1}^{S}\sum_{j=1}^{N} x_{js} d_{js}$ represents the deployment cost of all the edge nodes. Eq. (2) indicates that all the traffic data of r_i is completely served by the edge nodes. Eq. (3) constrains that an edge server is deployed on at most one traffic station. Eq. (4) limits the capacity of computation and storage in the s-th edge node. Note that, since the spatial correlation of the traffic data is very weak when the traffic stations are far from each other [15], the data of these traffic stations is not helpful for data recovery. As a result, we only use the data of the traffic stations within coverage. Specifically, l_{ij} in Eq. (5) denotes whether r_i is within the coverage of r_j. If it is, $l_{ij} = 1$. Otherwise, $l_{ij} = 0$.

2) **Sub-problem B**: (*Accurate traffic recovery based on edge computing*) Given the optimal deployment of edge nodes in the sub-problem A, we study how to accurately recover the missing traffic data of a traffic station, leveraging its remaining data based on temporal correlation and the data of its nearby traffic stations based on spatial correlation. Formally, we assume the traffic station with incomplete traffic data is r_i, $i \in \{1,\ldots,N\}$. Let $\mathbf{T}^{\mathrm{m}} = \{t_1, t_2, \ldots, t_n\}$ denote the sequence of missing data points with no traffic records, while $\mathbf{T}^{\mathrm{s}}$ represents the corresponding time intervals, *i.e.*, $\mathbf{T}^{\mathrm{s}} = \mathbf{T}/\mathbf{T}^{\mathrm{m}}$. The recovery value of $v_i(t)$ is represented by $\hat{v}_i(t)$. We let $\mathbf{r}_i^{\mathrm{c}}$ denote the set of traffic stations within the coverage of r_i, *i.e.*, $\mathbf{r}_i^{\mathrm{c}} = \{r_j | \forall j \in \{1,\ldots,N\}, l_{ij} = 1\}$. Hence, the problem is formalized as:

$$\min_{\Phi} \quad \frac{1}{T}\sum_{t=1}^{T} |v_i(t) - \hat{v}_i(t)|, \tag{7}$$

$$\text{s.t.} \quad \hat{v}_i(t) = \Phi(\mathbf{v}_i^{\mathrm{s}}, \mathbf{v}_i^{\mathrm{c}}), \forall t \in \mathbf{T}^{\mathrm{m}}, \tag{8}$$

$$\hat{v}_i(t) = v_i(t), \forall t \in \mathbf{T}^{\mathrm{s}}, \tag{9}$$

$$\mathbf{v}_i^{\mathrm{s}} = \{v_i(t) | \forall t \in \mathbf{T}^{\mathrm{s}}\}, \tag{10}$$

$$\mathbf{v}_i^{\mathrm{c}} = \{\mathbf{v}_j | \forall j \in \mathbf{r}_i^{\mathrm{c}}\}, \tag{11}$$

where Eq. (7) represents the error measurement between the recovery value and the ground truth, *e.g.*, Mean Absolute Error [20]. $\mathbf{v}_i^{\mathrm{s}}$ in Eq. (10) and $\mathbf{v}_i^{\mathrm{c}}$ in Eq. (11) denote the set of sensing traffic data on r_i and that from all the traffic stations within r_i's coverage, respectively. In Eq. (8), the missing traffic data of r_i is estimated based on $\mathbf{v}_i^{\mathrm{s}}$ and $\mathbf{v}_i^{\mathrm{c}}$, exploiting the recovery function $\Phi(\cdot)$. Meanwhile, as Eq. (9), the recovery values (*e.g.*, $\hat{v}_i(t)$) at the sensing time $t \in \mathbf{T}^{\mathrm{s}}$ are required to be the same as the ground truth $v_i(t)$.

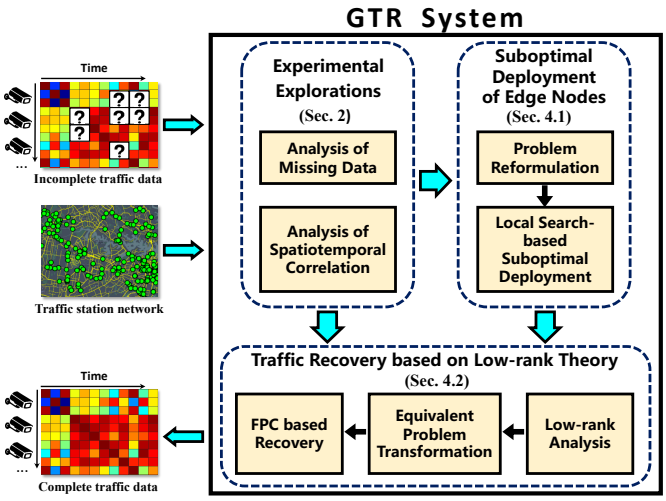


Fig. 6: Overview of *GTR*, an edge computing-based large-scale traffic data recovery system leveraging Low-rank theory.

4 System Design

To address the above sub-problems A and B in Sec. 3.2, we propose *GTR*, a large-scale traffic data recovery system, leveraging edge-computing and low-rank theory to achieve accurate, real-time traffic data recovery in large-scale ITSs. As demonstrated in Fig. 6, as the inputs of *GTR*, the ITSs provide large-scale traffic data of massive stations with spatiotemporal dynamic workload and numerous missing data. Moreover, they offer the topology of the communication network among these traffic stations in ITSs. Finally, *GTR* yields the accurate traffic data for large numbers of traffic stations in real time. Specifically, *GTR* consists of three main components as follows.

1) *Experimental explorations* (Sec. 2). We first conduct experiments to explore the issue of missing traffic data based on large-scale traffic datasets in Sec. 2.1. The results indicate that this issue is greatly serious due to the high missing rate and its pervasiveness. Further, extensive experiments are carried out to investigate the spatiotemporal correlations of traffic data in Sec. 2.2. These experimental observations are fed back to design the edge nodes deployment scheme and traffic data recovery algorithm in the following components.
2) *Suboptimal deployment of edge nodes* (Sec. 4.1). To resolve the issues of incomplete data, large-scale coverage and resource-intensive traffic data, we present the edge computing-empowered large-scale traffic data recovery system. Specifically, we focus on the optimal deployment problem of edge nodes, which is an intractable NP-hard problem. Thus, we leverage the problem reformulation and the supermodular theory to achieve a sub-optimal solution with a performance guarantee.
3) *Traffic recovery based on low-rank theory* (Sec. 4.2). Based on the experimental analysis of low rank by SVD, we present the accurate traffic data recovery algorithm

based on low-rank theory. It jointly leverages both the temporal and spatial correlations of traffic data at different time and stations to achieve accurate data recovery.

4.1 Suboptimal Deployment of Edge Nodes

In this subsection, we study how to solve the sub-problem A for traffic data recovery exploiting submodularity/supermodularity. The key idea of our approach is as follows. First, we notice that in the sub-problem A, given any edge node placement scheme $\mathbf{x}$, the traffic data allocation problem is a simple linear programming (LP) problem, whose optimal solution $\mathbf{y}^*(\mathbf{x})$ can be efficiently obtained. By substituting $\mathbf{y}$ with $\mathbf{y}^*(\mathbf{x})$ in the sub-problem A, it is equivalent to a Binary Integer Programming (BIP) problem for the edge node placement variable $\mathbf{x}$ only. Second, we reformulate the aforementioned BIP problem as a set function optimization problem. Although the explicit form of the objective function is difficult to obtain, we prove that it is supermodular and the constraints in the problem form a matroid constraint. Last, we design a suboptimal algorithm for sub-problem A with a theoretical performance guarantee.

4.1.1 Problem Reformulation

For the sub-problem A, it has the following properties about the traffic data allocation optimization.

Lemma 1. *In the sub-problem A, given any fixed edge node placement scheme, the optimal traffic data allocation can be obtained in polynomial time.*

Proof. Given any edge node placement $\mathbf{x^0} = \{x_{ij}^0\}$, the sub-problem A turns into a traffic data allocation problem with respect to $\mathbf{y}$ only in the following:

$$\begin{aligned}(\mathbf{P0}) \quad \min_{\mathbf{y}} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} w_i y_{ij} b_{ij} \\ \text{s.t.} \quad & (2),(5),(6), \\ & \sum_{i=1}^{N} w_i y_{ij} \le \sum_{s=1}^{S} x_{js}^0 c_s, \forall j \in \mathcal{N}, \end{aligned} \tag{12}$$

which is a simple LP problem and can be solved in polynomial time by many classical LP methods [21]. □

Based on Lemma 1, we reformulate the sub-problem A as a set function optimization problem. Formally, denote the objective function of the sub-problem A as $\Omega(\mathbf{x}, \mathbf{y})$. First, Lemma 1 indicates that, given any $\mathbf{x}$, we can obtain the optimal value of $\mathbf{y}$, denoted as $\mathbf{y}^*(\mathbf{x})$. Although the explicit expression of $\mathbf{y}^*(\mathbf{x})$ is hard to obtain, it implies that solving the sub-problem A is equivalent to solving the problem with respect to $\mathbf{x}$ only, by substituting $\mathbf{y}$ with $\mathbf{y}^*(\mathbf{x})$. Thus, the objective function can be transformed into $\Omega(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$.

Second, let $\mathcal{G} := \{(j,s) | \forall j \in \mathcal{N}, s \in \mathcal{S}\}$, which establishes a one-one mapping between an edge node placement variable x_{js} and the element $e = (j,s) \in \mathcal{G}$. Specifically, $x_{js} = 1$ implies choosing element (j,s) from $\mathcal{G}$, while $x_{js} = 0$ means not choosing element (j,s) from $\mathcal{G}$. Let $\mathcal{A} \subseteq \mathcal{G}$ represent the set of selected pairs of edge node and traffic station, that is, $\mathcal{A} = \{(j,s) | x_{js} = 1, j \in \mathcal{N}, s \in \mathcal{S}\}$. For a feasible set $\mathcal{A} \subseteq \mathcal{G}$, we define $f(\mathcal{A}) := \Omega(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$, where for each x_{js} in $\mathbf{x}$, $x_{js} = 1$ iff $(j,s) \in \mathcal{A}$. Then, by introducing $\mathbb{1}$ as the indicator function, we reformulate the sub-problem A in the following:

$$\begin{aligned}(\mathbf{P0}') \quad \min_{\mathcal{A} \subseteq \mathcal{G}} \quad & f(\mathcal{A}) \\ \text{s.t.} \quad & \sum_{j:(j,s)\in\mathcal{A}} \mathbb{1}_{(j,s)\in\mathcal{A}} \le 1, \forall s \in \mathcal{S}. \end{aligned} \tag{13}$$

Next, we reveal some desirable properties of the problem $\mathbf{P0}'$. We first provide the basic definitions of matroid, non-negativity, monotonicity, and submodularity as follows.

Definition 1. *(Non-negativity, Monotonicity, Submodularity [22]) A set function $f : 2^{\mathcal{G}} \to \mathbb{R}$ ($\mathcal{G}$ is a finite ground set) is non-negative if $f(\emptyset) = 0$ and $f(\mathcal{A}) \ge 0$ for $\forall \mathcal{A} \subseteq \Omega$. $f(\cdot)$ is monotone if for $\forall \mathcal{A}_1 \subseteq \mathcal{A}_2 \subseteq \mathcal{G}, f(\mathcal{A}_1) \le f(\mathcal{A}_2)$. And $f(\cdot)$ is submodular, if and only if $\forall \mathcal{A}_1 \subseteq \mathcal{A}_2 \subseteq \mathcal{G}$ and $\forall e \in \mathcal{G} \setminus \mathcal{A}_2$, $f(\mathcal{A}_1 \cup \{e\}) - f(\mathcal{A}_1) \ge f(\mathcal{A}_2 \cup \{e\}) - f(\mathcal{A}_2)$.*

Any function $f(\cdot)$ is *supermodular* if $-f(\cdot)$ is *submodular*. Submodularity has a decreasing returns property while supermodularity captures an increasing returns property, which implies that the added value of an extra element to a bigger set is no less than that to a smaller set [22].

Definition 2. *(Matroid [23]) Consider a finite ground set $\mathcal{G}$, and a non-empty collection of subsets of $\mathcal{G}$, represented by $\mathcal{I}$. The pair $(\mathcal{G}, \mathcal{I})$ is called a matroid, if and only if the following conditions hold: 1) If $\mathcal{A} \subseteq \mathcal{B} \in \mathcal{I}$, then $\mathcal{A} \in \mathcal{I}$; 2) If $\mathcal{A}, \mathcal{B} \in \mathcal{I}$ and $|\mathcal{A}| < |\mathcal{B}|$, then there exists $b \in \mathcal{B}$ such that $\mathcal{A} \cup \{b\} \in \mathcal{I}$.*

Definition 3. *(Partition matroid [23]) A matroid $(\mathcal{G}, \mathcal{I})$ is a partition matroid, if there exist disjoint sets $\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_m$ and positive integers $i_1, i_2, ..., i_m$ for a positive integer m, such that $\mathcal{G} := \mathcal{G}_1 \cup \mathcal{G}_2 \cup ... \cup \mathcal{G}_m$ and $\mathcal{I} := \{\mathcal{A} : \mathcal{A} \subseteq \mathcal{G}, |\mathcal{A} \cap \mathcal{G}_j| \le i_j, j = 1, 2, ..., m\}$ hold.*

Lemma 2. *The objective function $f(\mathcal{A})$ ($\mathcal{A} \subseteq \mathcal{G}$) in problem $\mathbf{P0}'$ is non-negative and supermodular.*

Proof. First, the objective function $f(\mathcal{A})$ is non-negative, since if $\mathcal{A} = \emptyset$, the corresponding optimal data allocation $y_{ij}^* = 0$ for $\forall i, j \in \mathcal{N}$ and accordingly $f(\emptyset) = 0$. And $f(\mathcal{A}) \ge 0$ for all $\mathcal{A} \subseteq \mathcal{G}$ due to the non-negative expression of the objective function in problem $\mathbf{P0}$. This is reasonable since no placement cost will be incurred and no traffic data transfer will happen if no edge node is placed.

Second, according to Definition 1, to prove the supermodularity, we need to show that, for any feasible $\mathcal{A}_1, \mathcal{A}_2 \subseteq \mathcal{G}$ and any $(j_1, s_1) \in \mathcal{G} \setminus \mathcal{A}_2$ satisfying that $\mathcal{A}_1 \subseteq \mathcal{A}_2$ and $\mathcal{A}_2 \cup \{(j_1, s_1)\})$ is feasible, it holds:

$$f(\mathcal{A}_1 \cup \{(j_1,s_1)\}) - f(\mathcal{A}_1) \le f(\mathcal{A}_2 \cup \{(j_1,s_1)\}) - f(\mathcal{A}_2). \tag{14}$$

Suppose that $\mathbf{y}^{(1)} = \{y_{ij}^{(1)}\}_{i\in\mathcal{N}, j\in\mathcal{N}}$ and $\widehat{\mathbf{y}}^{(1)} = \{\hat{y}_{ij}^{(1)}\}_{i\in\mathcal{N}, j\in\mathcal{N}}$ is the optimal traffic data allocation solution obtained by solving $\mathbf{P0}$ under the edge node placement variable $\mathcal{A}_1$ and $\mathcal{A}_1 \cup \{(j_1, s_1)\}$ respectively. Similarly, we can get the optimal traffic data allocation $\mathbf{y}^{(2)} = \{y_{ij}^{(2)}\}_{i\in\mathcal{N}, j\in\mathcal{N}}$ and $\widehat{\mathbf{y}}^{(2)} = \{\hat{y}_{ij}^{(2)}\}_{i\in\mathcal{N}, j\in\mathcal{N}}$ under the edge node placement variable $\mathcal{A}_2$ and $\mathcal{A}_2 \cup \{(j_1, s_1)\}$ respectively. Thus, we can rewrite the objective values of the sub-problem

A under $\mathcal{A}_1, \mathcal{A}_1 \cup \{(j_1, s_1)\}, \mathcal{A}_2, \mathcal{A}_2 \cup \{(j_1, s_1)\}$, respectively, as follows:

$$f(\mathcal{A}_1) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i y_{ij}^{(1)} b_{ij} + \sum_{(j,s)\in\mathcal{A}_1} d_{js}, \tag{15}$$

$$f(\mathcal{A}_1 \cup \{(j_1, s_1)\}) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i \hat{y}_{ij}^{(1)} b_{ij} + \sum_{(j,s)\in\mathcal{A}_1} d_{js} + d_{j_1 s_1}, \tag{16}$$

$$f(\mathcal{A}_2) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i y_{ij}^{(2)} b_{ij} + \sum_{(j,s)\in\mathcal{A}_2} d_{js}, \tag{17}$$

$$f(\mathcal{A}_2 \cup \{(j_1, s_1)\}) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i \hat{y}_{ij}^{(2)} b_{ij} + \sum_{(j,s)\in\mathcal{A}_2} d_{js} + d_{j_1 s_1}, \tag{18}$$

Then, we obtain:

$$\text{LHS of (14)} = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i(\hat{y}_{ij}^{(1)} b_{ij} - y_{ij}^{(1)} b_{ij}) + d_{j_1 s_1}, \tag{19}$$

$$\text{RHS of (14)} = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i(\hat{y}_{ij}^{(2)} b_{ij} - y_{ij}^{(2)} b_{ij}) + d_{j_1 s_1}. \tag{20}$$

The first items in (19) and (20) are the negative decrement values in communication cost before/after deploying the edge node s_1 on traffic station r_{j_1}, respectively. After adding (j_1, s_1) to $\mathcal{A}_1$, the communication cost reduction can be divided into two parts. First, consider some r_{i_2} in the coverage of r_{j_1}. If the data at r_{i_2} is originally transmitted to r_{j_2} in data allocation $\mathbf{y}^{(1)}$ and the communication cost $b_{i_2 j_1}$ to r_{j_1} is lower than $b_{i_2 j_2}$, the communication cost can be reduced by redirecting the data at r_{i_2} to r_{j_1}. Second, the edge nodes in r_{j_2} may be fully-loaded in $\mathcal{A}_1$ and after the redirection, we may redirect the data at other traffic stations to r_{j_2}, which can reduce the communication cost. The operation in the second part may be iteratively completed.

Consider some data transmission from $r_{i'}$ to $r_{j'}$ in $\mathbf{y}^{(1)}$ that is redirected after adding (j_1, s_1) to $\mathcal{A}_1$. Since that $\mathcal{A}_2 \setminus \mathcal{A}_1$ may contain some (j_3, s_3) and the communication cost $b_{i' j_3}$ may be lower than $b_{i' j'}$, cost reduction caused by (j_1, s_1) can be lower in $\widehat{\mathbf{y}}^{(2)}$ than $\widehat{\mathbf{y}}^{(1)}$. Thus, the communication cost decrement caused by (j_1, s_1) under $\mathcal{A}_2$ is lower than the cost decrement under $\mathcal{A}_1$. So $\sum_{i=1}^{N}\sum_{j=1}^{N} w_i(\hat{y}_{ij}^{(1)} b_{ij} - y_{ij}^{(1)} b_{ij}) \le \sum_{i=1}^{N}\sum_{j=1}^{N} w_i(\hat{y}_{ij}^{(2)} b_{ij} - y_{ij}^{(2)} b_{ij})$. The second item in (19) and (20) is the same. Therefore, (14) holds and the lemma is proved. □

We note that whether function $f(\mathcal{A})$ ($\mathcal{A} \subseteq \mathcal{G}$) is monotone or not is unknown, and the reason is as follows. On one hand, if adding more elements into a feasible set $\mathcal{A} \subseteq \mathcal{G}$, the only affected constraint (12) in problem $\mathbf{P0}$ will be relaxed and the feasible solution region of the LP problem for $\mathbf{y}$ will be expanded, which results in the decreased optimal value. This implies that the first part of the objective function of the sub-problem A decreases with the increase of a bigger set. On the other hand, adding more elements into $\mathcal{A}$ will inevitably incur greater placement cost, which leads to an increase of the second part in the objective function of the sub-problem A. As the increase of a feasible set has an opposite trend in the two parts, the monotonicity of the function $f(\mathcal{A})$ is difficult to determine.

Remark 1. *The monotonicity of function $f(\mathcal{A})$ is unknown.*

Lemma 3. *Let $\mathcal{G} := \{(j,s)|\forall j \in \mathcal{N}, s \in \mathcal{S}\}$ and $\mathcal{I} := \{\mathcal{A}|\mathcal{A} \subseteq \mathcal{G}, \forall a_1 := (j_1, s_1), a_2 := (j_2, s_2) \in \mathcal{A}, s_1 \neq s_2\}$. Then, the constraint (13) in problem $\mathbf{P0}'$ is a partition matroid constraint.*

Proof. We first prove that the constructed pair $(\mathcal{G}, \mathcal{I})$ is a matroid. We assume that there are at least two traffic stations and two available edge nodes in the problem, *i.e.*, $N \ge 2$ and $S \ge 2$; otherwise, solving the problem is trivial. In the following, we prove the three properties of a matroid by Definition 2 one by one. First, the nonempty property of $\mathcal{I}$ is easy to validate due to the previous assumption. Second, if $\mathcal{A} \subseteq \mathcal{B} \in \mathcal{I}$, we have $\mathcal{A} \in \mathcal{I}$. If not, there exist at least two different elements $a_1, a_2 \in \mathcal{A}$ that share the same second component. Since $\mathcal{A} \subseteq \mathcal{B}$, $a_1, a_2 \in \mathcal{B}$ holds, which obviously contradicts with $\mathcal{B} \in \mathcal{I}$.

Third, suppose that $\mathcal{A}, \mathcal{B} \in \mathcal{I}$ and $|\mathcal{A}| < |\mathcal{B}|$. If there does not exist an element $a' \in \mathcal{B}$ such that $\mathcal{A} \cup \{a'\} \in \mathcal{I}$, we have for any element $a' \in \mathcal{B}$, $\mathcal{A} \cup \{a'\} \notin \mathcal{I}$ holds. Since $\mathcal{A} \in \mathcal{I}$, each element in $\mathcal{B}$ shares the same second component with some element in $\mathcal{A}$. Due to $|\mathcal{A}| < |\mathcal{B}|$, there exist at least two different elements $a_1, a_2 \in \mathcal{B}$ and an element $a' \in \mathcal{A}$, whose second component is exactly identical. This means that a_1, a_2 share the same second component, which contradicts with $\mathcal{B} \in \mathcal{I}$. Therefore, $(\mathcal{G}, \mathcal{I})$ is indeed a matroid.

We further prove that $(\mathcal{G}, \mathcal{I})$ under the constraint (13) is a partition matroid. Since set $\mathcal{G}$ captures all possible traffic station-edge node pairs, we have $\mathcal{G} = \bigcup_{s=1}^{S} \mathcal{G}_s$ where $\mathcal{G}_s := \{(j,s)|j = 1, 2, ..., N\}$. Combing with the meaning of the constraint (13) and the definition of $\mathcal{I}$, we have, for $\forall \mathcal{A} \in \mathcal{I}$, $|\mathcal{A} \cap \mathcal{G}_s| \le 1$ holds for $s = 1, 2, ..., S$. To summarize, $(\mathcal{G}, \mathcal{I})$ is a partition matroid. The lemma is thus proved. □

Theorem 1. *Both the sub-problem A and $\mathbf{P0}'$ are NP-hard.*

Proof. We prove the theorem by proving the NP-hardness of problem $\mathbf{P0}'$ only, due to the equivalence of problem $\mathbf{P0}'$ and the sub-problem A. Based on Lemma 2 and Lemma 3, problem $\mathbf{P0}'$ is a non-negative supermodular minimization problem with a single matroid constraint, *i.e.*, $\text{Min}_{\forall \mathcal{A} \subseteq \mathcal{G}, \mathcal{A} \in \mathcal{I}} f(\mathcal{A})$. Note that $\text{Min}_{\forall \mathcal{A} \subseteq \mathcal{G}, \mathcal{A} \in \mathcal{I}} f(\mathcal{A})$ is equivalent to $\text{Max}_{\forall \mathcal{A} \subseteq \mathcal{G}, \mathcal{A} \in \mathcal{I}} - f(\mathcal{A})$, which is a submodular maximization problem. As is known to all, different from submodular minimization, submodular maximization is NP-hard, including the case with a matroid constraint [24]. Therefore, problem $\mathbf{P0}'$ is NP-hard, which also establishes the NP-hardness of the sub-problem A. □

4.1.2 Local Search-based Suboptimal Deployment

In light of Lemma 2 and Remark 1, $-f(\mathcal{A})$ is negative and submodular with unknown monotonicity. There exists a $(\frac{1}{4+\epsilon})$-approximation algorithm for maximizing a non-negative submodular but not necessarily monotone function subject to a single matroid constraint [24]. To apply that approach, we need to first transform the objective function into an appropriate non-negative function, *i.e.*, $\tilde{f}(\mathcal{A}) := f_{\max} - f(\mathcal{A})$, where $f_{\max} := \sum_{i=1}^{N}\sum_{j=1}^{N} w_i b_{ij} + \sum_{s=1}^{S}\sum_{j=1}^{N} d_{js}$. Inspired by [24], we design a local search-based suboptimal edge node placement algorithm with performance guarantee as illustrated in Algorithm 1.

Algorithm 1: Local Search-based Suboptimal Edge Node Deployment Algorithm.

Input: Set $\mathcal{G} = \{(j,s)|\forall j \in \mathcal{N}, s \in \mathcal{S}\}$, matroid $(\mathcal{G}, \mathcal{I})$, value access to function $\tilde{f}(A)$, and constant $\epsilon > 0$.
Output: Edge node placement $\mathbf{x} = \{x_{js}\}$, traffic data allocation $\mathbf{y} = \{y_{ij}\}$.

1 Initialize $\mathbf{x} = \mathbf{0}$, $\mathbf{y} = \mathbf{0}$.
2 Initialize a feasible set $\mathcal{A} \subseteq \mathcal{G}$.
3 **while** 1 **do**
4 **if** there exists $e \in \mathcal{A}$ such that $\tilde{f}(\mathcal{A} \setminus \{e\}) \geq (1 + \frac{\epsilon}{N^4 S^4})\tilde{f}(\mathcal{A})$ **then**
5 $\mathcal{A} \leftarrow \mathcal{A} \setminus \{e\}$.
6 **else if** there exist $e \in \mathcal{G} \setminus \mathcal{A}$, $e' \in \mathcal{A} \cup \{\emptyset\}$ such that $(\mathcal{A} \setminus \{e'\}) \cup \{e\} \in \mathcal{I}$ and $\tilde{f}((\mathcal{A} \setminus \{e'\}) \cup \{e\}) > (1 + \frac{\epsilon}{N^4 S^4})\tilde{f}(\mathcal{A})$ **then**
7 $\mathcal{A} \leftarrow (\mathcal{A} \setminus \{e'\}) \cup \{e\}$.
8 **else**
9 break
10 Set all $x_{js} = 1$ if $u = (j,s) \in \mathcal{A}$.
11 Solve problem $\mathbf{P0}$ with the input of $\mathbf{x}$ to obtain $\mathbf{y}$, and return $\mathbf{x}$ and $\mathbf{y}$.

We introduce the steps of Algorithm 1 in detail as follows. First, line 1 initializes $\mathbf{x}$ and $\mathbf{y}$ as 0, and corresponding set $\mathcal{A}$ as an empty set, respectively. Second, line 2 finds a feasible set $\mathcal{A} \subseteq \mathcal{G}$. Third, in lines 3-7, we employ local search on $\mathcal{G}$ running both deletions (lines 4-5) and exchanges (line 6-7) to obtain a set $\mathcal{A} \subseteq \mathcal{G}$, $\mathcal{A} \in \mathcal{I}$, such that the value of $\tilde{f}(\mathcal{A})$ can be increased by a factor of at least $(1 + \frac{\epsilon}{N^4 S^4})$ at each iteration. Last, in lines 10-11, the algorithm outputs the edge node placement decision $\mathbf{x}$ whose value of each element is determined based on the chosen set $\mathcal{A}$, and the traffic data allocation decision $\mathbf{y}$ by solving problem $\mathbf{P0}$ with the input of $\mathbf{x}$.

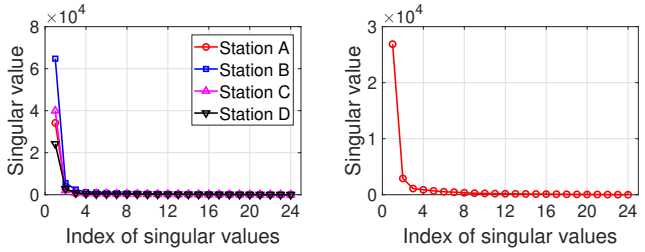
The performance guarantee of Algorithm 1 as well as its time complexity are theoretically analyzed as follows.

Theorem 2. *Let $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}^*, \mathbf{y}^*)$ be the output of Algorithm 1 and the optimal solution of the sub-problem A, respectively. Then, we have*

$$\Omega(\mathbf{x}, \mathbf{y}) \leq \frac{1}{4+\epsilon}\Omega(\mathbf{x}^*, \mathbf{y}^*) + \frac{3+\epsilon}{4+\epsilon} f_{\max}, \tag{21}$$

where $\Omega(\cdot,\cdot)$ is the objective function of the sub-problem A, $\epsilon > 0$ is the parameter determined by Algorithm 1, and $f_{\max} = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i b_{ij} + \sum_{s=1}^{S}\sum_{j=1}^{N} d_{js}$. Furthermore, the time complexity of Algorithm 1 is polynomial.

Proof. This theorem is a corollary of Theorem 2.6 in [24], which proves that there exists a $\frac{1}{4+\epsilon}$-approximation algorithm for maximizing any non-negative submodular set function subject to a matroid constraint. By applying the local search algorithm following that theorem, Algorithm 1 actually designs an approximation algorithm for the problem $\text{Max}_{\forall \mathcal{A} \subseteq \mathcal{G}, \mathcal{A} \in \mathcal{I}} f_{\max} - f(\mathcal{A})$, where set $\mathcal{A}$ is obtained by Algorithm 1. In light of that theorem, we have $f_{\max} - f(\mathcal{A}) \geq \frac{1}{4+\epsilon}[f_{\max} - f^*]$, where f^* is the optimal objective value of problem $\mathbf{P0}'$. Remember that the relationship between $f(\cdot)$ and $\Omega(\cdot,\cdot)$ and by some operations, we have the inequality as presented in the theorem. For the time complexity, it can be analyzed similar to Theorem 2.6 in [24], which is omitted here. The theorem is thus proven. □



(a) Temporal dimension (b) Spatial dimension

Fig. 7: Low-rank analysis of traffic volume matrix based on SVD in terms of temporal and spatial dimensions.

4.2 Accurate Traffic Data Recovery based on Low-rank Theory

4.2.1 Experimental Analysis of Low Rank

1) According to the experimental explorations in Sec. 2.2, the traffic data in the ITSs has the temporal correlation at different time and the spatial correlation on different traffic stations. Thus, we further evaluate whether the rank of traffic volume matrix ($\mathbf{V}$) is low in terms of temporal-spatial dimensions, by using Singular Value Decomposition (SVD) as Def. 4.

Definition 4. ***(Singular Value Decomposition, SVD)*** *For any $m \times n$ matrix denoted by $\mathbf{V}$, the SVD is a factorization of $\mathbf{V}$ as:*

$$\mathbf{V} = U_{m\times m}\Sigma_{m\times n}\Xi^*_{n\times n} \tag{22}$$

$$\text{where } \Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_i & \\ & & & \ddots \end{bmatrix}.$$

Note that Σ in Eq. (4) is a diagonal matrix. Moreover, σ_i $(\forall i \in \{1,\ldots,\min(m,n)\}, \sigma_i \geq 0)$ is named as the singular value of $\mathbf{V}$, and $\sigma_i \leq \sigma_j$ if $\forall i,j \in \{1,\ldots,\min(m,n)\}$ and $i < j$.

According to Def. 4, we conduct experiments to analyze the low rank of traffic volume matrix at different time and traffic stations, respectively. Firstly, we analyze its property of low rank in terms of temporal dimension. Specifically, we make the SVD of the traffic volume matrix on the stations A-D, where this matrix represents the data of all the weeks on a traffic station and each row denotes that of each week. Fig. 7a illustrates that the singular values mainly concentrate on a very limited number of elements. For example, few singular values (no more than 2) are far larger than others for all of these four stations. Furthermore, we analyze the low-rank property of the traffic volume matrix in the spatial dimension. This matrix represents the traffic volume of 25 stations, where each row denotes the data of one station at different time. As illustrated in Fig. 7b, similar to Fig. 7a, the weights of singular values also focus on a few elements in terms of spatial dimension.

In summary, the above experimental results indicate that the weights of singular values mainly concentrate on a very limited number of elements, in terms of both the temporal and spatial dimensions. According to the theorem of matrix rank [25], [26], if the weights of singular values for a matrix focus on very few elements, this matrix is approximately low rank. As a result, *the matrix of traffic volume in terms of both temporal and spatial dimensions is roughly low-rank.*

4.2.2 Accurate Traffic Recovery based on Low-rank Theorey

Based on the experimental analysis in Sec. 4.2.1, the traffic volume matrix $\mathbf{V}$ is approximately low-rank in terms of the temporal-spatial dimensions.According to the theorem of low-rank theory [27], if the matrix $\mathbf{M}$ is approximately low-rank and satisfies that the number of randomly sampled entries is large enough, we can find a low-rank decision matrix $\mathbf{\Theta}$ to approximately replace $\mathbf{V}$. Thus the sub-problem B in Eqs. 7-11 transform will be transformed into the low-rank minimization problem as

$$\min \quad \mathrm{rank}(\mathbf{\Theta}) \tag{23}$$
$$\text{s.t.} \quad \mathbf{\Theta}_{ij} = \mathbf{V}_{ij}, (i,j) \in \Omega. \tag{24}$$

where $\mathrm{rank}(\cdot)$ denotes the function of computing the matrix rank, *i.e.,* the number of non-zero singular values, and Ω is the set of index pairs for both $\mathbf{\Theta}$ and $\mathbf{V}$.

This low-rank minimization problem in Eq. 23 is NP-hard, due to the combinational property of the function $\mathrm{rank}(\cdot)$. Thus, it can be equivalently relaxed as Eq. 25 by using the nuclear norm $\|\cdot\|_*$ and the ℓ_2-norm $||\cdot||_2$ [28].

$$\min \quad \lambda\|\mathbf{\Theta}\|_* + \frac{1}{2}||\mathcal{A}(\mathbf{\Theta}) - b||_2^2, \tag{25}$$
$$\|\mathbf{\Theta}\|_* = \sum_{i=1}^{\min(m,n)} \sigma_i(\mathbf{\Theta}), \tag{26}$$

where λ denotes the weight factor to trade off between the nuclear norm and equality constraint Eq. 24. The linear map $\mathcal{A} : \mathbb{R}^{m\times n} \rightarrow \mathbb{R}^p$ and vector $b \in \mathbb{R}^p$ describe that the observed elements in $\mathbf{\Theta}$ are equal to the elements having same positions in $\mathbf{V}$. $\|\mathbf{\Theta}\|_*$ in Eq. 25 denotes the sum of all the non-zero singular values of $\mathbf{\Theta}$. The minimization of the nuclear norm and square of ℓ_2-norm is a convex optimization problem [29]. Thus, we equivalently transform the accurate traffic recovery problem, which is an intractable NP-hard one, into a tractable convex optimization problem as in Eq. 25.

To address this convex optimization problem, we exploit Fixed Point Continuation (FPC) iterative scheme [30] to achieve the optimal solution via limited iterations. FPC algorithm is comprising of two key ideas: the fixed-point-based iterative scheme and the continuation-based accelerated convergence strategy [31].

- **Fixed-point-based iteration**. It iteratively searches the fixed point, which is also the optimal solution of the convex optimization problem [29]. Specifically, for any $\varepsilon > 0$, let the matrix shrinkage operator $s_\varepsilon(\upsilon) = \mathbf{x}$, where $x_i = \upsilon_i - \varepsilon$ if $\upsilon_i - \varepsilon > 0$; otherwise, $x_i = 0$. In each iteration, it updates the new solution based on the previous one, using the matrix shrinkage operator. Let $\mathbf{\Theta}^k$ denote the current solution of the k-th iteration. Then, the new solution of the $(k+1)$-th iteration (*i.e.,* $\mathbf{\Theta}^{k+1}$) is:

$$\mathbf{\Theta}^{k+1} = U_Y \mathrm{Diag}(s_{\tau\lambda}(\sigma))\Xi_Y^T,, \tag{27}$$
$$\text{where} \quad \mathbf{Y}^k = \mathbf{\Theta}^k - \tau\mathbf{g}(\mathbf{\Theta}^k). \tag{28}$$

  Note that τ is a positive constant, and U_Y, Ξ_Y, σ come from the SVD of $\mathbf{Y}$, *i.e.,* $\mathbf{Y} = U_Y\mathrm{Diag}(\sigma)\Xi_Y^{\mathrm{T}}$. $\mathbf{g}(\mathbf{\Theta^k})$ represents the gradient of $\frac{1}{2}||\mathcal{A}(\mathbf{\Theta}) - b||_2^2$ at $\mathbf{\Theta}^k$.

- **Continuation-based convergence acceleration**. According to the convergence analysis [30], the speed of convergence is determined by the acceleration factor ζ, *i.e.,* $\lambda_{k+1} = max\{\zeta_k\lambda_k, \bar{\lambda}\}$. The smaller ζ is, the faster λ reduce. Thus, the Continuation-based convergence strategy is used for accelerating the convergence. Specifically, in the outer iteration, we iteratively select the λ in the ascending sequence, which is then used to search the fixed point in the inner iterations.

Algorithm 2: FPC-based Accurate Traffic Data Recovery Algorithm.

```
Initialize: Given v_i^c, select ζ_1 > 0, ζ_2 > 0, ..., ζ_n > 0, λ̄ > 0,
  λ_1 > λ_2 > ... > λ_n = λ̄. Set Θ = v_i^c.
for λ = λ_1, λ_2, ..., λ_n and λ_{k+1} = max{ζ_k λ_k, λ̄}, do
    while NOT converged, do
        Select τ > 0;
        Compute Y = Θ − τg(Θ) and SVD of Y, where
          g(Θ) = ∇(1/2 ||A(Θ) − b||_2^2);
        Compute Θ = U_Y Diag(s_{τλ}(σ)) Ξ_Y^T.
Return v̂_i^c.
```

5 Traces-based Evaluations

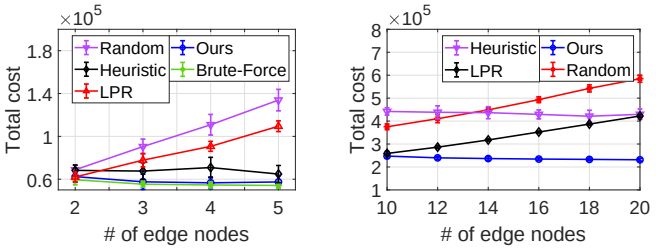
In this section, we conduct extensive experiments based on an empirical traffic dataset from a large-scale, real-world ITS. Specifically, we evaluate the performance of *GTR* from two important perspectives, *i.e.,* the cost of edge deployment and the accuracy of traffic data recovery. In the following subsections, we first describe the traffic dataset and experimental methodologies, including experimental settings, baseline methods, and evaluation metrics. Then, we present the experimental results with analysis on edge nodes deployment and traffic data recovery.

5.1 Experimental Methodology and Settings

5.1.1 Datasets and Experimental Methodology

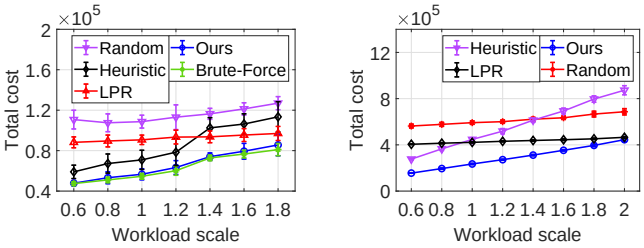
1) **Description of large-scale ITS dataset**: This traffic volume dataset is collected from an online Traffic Volume Viewer System (TVVS), which is established by the Transportation Department of New South Wales, Australia [3]. The data of traffic volume is generated by permanent and temporary roadside collection stations that monitor the number of passing vehicles on each road with calculation on the one-hour interval [4]. As shown in Fig. 1, the whole dataset covers more than 600 traffic stations that are distributed across most areas in the state of New South Wales. For our experimental studies, we collect 12-month (*e.g.,* from January 2018 to December 2018) traffic data from 100 major traffic stations.

2) **Experimental methodology and settings**: To evaluate the performance for edge node deployment, we extract the workload and navigation distance between different traffic stations from the traffic dataset. For the number of stations N, we consider two network scenarios, *i.e.,* a large network with $N = 100$ and a small network with $N = 10$. In the large network, we deploy 20 edge nodes and the capacity of each node is drawn uniformly from the interval $[100k, 140k]$. In the small network, we deploy 4 edge nodes



(a) Small-scale scenario (b) Large-scale scenario

Fig. 8: Total cost with different number of edge nodes in the small-scale and the large-scale scenarios.



(a) Small-scale scenario (b) Large-scale scenario

Fig. 9: Total cost with different workload scale in the small-scale and the large-scale scenarios.

and the capacity of each node is drawn uniformly from the interval $[60k, 70k]$. We assume that the communication cost is proportional to the navigation distance between different traffic stations. The deployment cost d_{js} is drawn uniformly from the interval $[17888, 35776]$. Besides, the coverage of each traffic station is 2.4 km, which indicates that the traffic data at each traffic station can only be allocated to other traffic stations within this coverage.

To evaluate the performance of traffic data recovery, we randomly select N stations from the traffic dataset to form a traffic volume matrix for data recovery. Thereby, we set one random station as the target station for data recovery and generate missing values (with a length of L) in its traffic volume matrix. By applying *GTR* and other baseline methods to recover the incomplete traffic volume matrix, we compare their performance with different experimental settings.

5.1.2 Baseline Methods and Evaluation Metrics

1) **Baseline methods**: To make a comprehensive study on the performance of *GTR* in edge node deployment, we make comparisons with four baseline methods as follows.

- **Brute-Force**: This algorithm finds the optimal deployment solutions by exhaustive search over all the possible deployment decisions. Nevertheless, its computational complexity is extremely high owing to NP-hard, making it impossible for large-scale scenarios.
- **Random**: It randomly selects a traffic station from all possible stations to deploy an edge node.
- **Heuristic**: It greedily deploys the server with largest capacity at traffic stations that can cover the most traffic data.
- **LPR**: This algorithm uses the LP Relaxation to get the sub-optimal and fractional solution [32].

To compare the performance in traffic data recovery, we further employ three baseline methods in evaluations of recovery accuracy as follows.

- **LR(T)**: As the simplest baseline, LR(T) method uses the linear regression scheme to recover the missing data of a station by using its remaining data with temporal correlation.
- **LR(TS)**: This method exploits temporal and spatial correlations between a target station and its nearby stations. In other word, it collaborates multiple stations to recover missing data of a target station by using linear regression.
- **SVT** [33]: The Singular Value Thresholding algorithm (SVT) is based on low-rank minimization, which iteratively conducts soft-thresholding operations on the singular values of the target matrix until convergence.

2) **Evaluation metrics**: For the above experimental studies, we adopt four metrics to evaluate the algorithm performance,*i.e.,Total Cost*, *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE), and *Root Mean Squared Error* (RMSE). In specific, **Total Cost** is the sum of deployment cost and communication cost as defined in Eq. (1). We use the Total Cost to evaluate the cost-efficiency of proposed *GTR* in edge node deployment. **MAE** is a measurement of the average absolute error between recovery results and ground truth of traffic data, as defined in Eq. (29). **MAPE** expresses the accuracy as a percentage ratio by measuring the average ratio of the recovery error to the ground truth, as defined in Eq. (30). **RMSE** is the square root of the average squared error between the recovered values and ground truth of traffic data, as defined in Eq. (31). Note that MAE, RMSE, and MAPE have been widely employed to evaluate the recovery accuracy [34]. Moreover, both of MAE and RMSE are scale-dependent metrics, while MAPE is scale-independent.

$$\mathbf{MAE} = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left|v_i(t) - \hat{v}_i(t)\right|, \tag{29}$$

$$\mathbf{MAPE} = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left|\frac{v_i(t) - \hat{v}_i(t)}{v_i(t)}\right|, \tag{30}$$

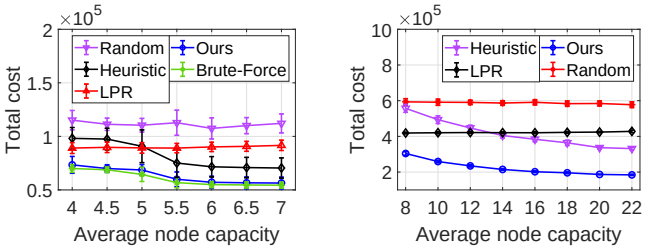
$$\mathbf{RMSE} = \sqrt{\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left(v_i(t) - \hat{v}_i(t)\right)^2}, \tag{31}$$

where N denotes the number of stations with incomplete data, and T represents the total length of missing data.

5.2 Experimental Results

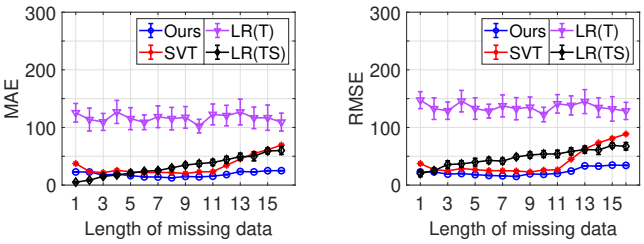
1) **Evaluations of edge nodes deployment**: We conduct the traces-based simulations to validate the deployment performance of *GTR* in different number of edge nodes, the workload scale, and the edge capacity.

First, we evaluate the impact of the number of edge nodes on the deployment performance in both small-scale and large-scale network scenarios. As shown in Fig. 8, our algorithm outperforms other baseline methods in terms of deployment cost. In contrast to Brute-Force method performs computation-intensive search, *GTR* employs efficient local search and still achieves a sub-optimal results by increasing the cost just 5.7% above the optimal one. In the small network scenario, our algorithm achieves more than



(a) Small-scale scenario (b) Large-scale scenario

Fig. 10: Total cost with different node capacity in the small-scale and the large-scale scenarios.



(a) MAE (b) RMSE

Fig. 11: Evaluations of recovery accuracy in different number of stations, in terms of MAE and RMSE.

94.3% of Brute-Force. The result shows that in small networks, our algorithm outperforms Random, Heuristic and LPR on average by 37.90%, 13.71%, and 27.63% in Fig. 8a, respectively. In the large network scenario, Fig. 8b shows that our algorithm outperforms Random, Heuristic and LPR by 48.80%, 45.21%, and 27.44% on average, respectively.

Second, as illustrated in Fig. 9, we vary the workload scale (*i.e.* the scaling ratio of while keeping the workload distribution as the same). In the small networks, Fig. 9a shows our algorithm outperforms Random, Heuristic, and LPR on average by 43.24%, 22.40%, and 29.37%, respectively. In the large networks, as demonstrated in Fig. 9b, our algorithm outperforms Random, Heuristic and LPR by 52.99%, 47.90%, and 32.78%, respectively. Furthermore, as shown in Fig. 9a, our algorithm achieves approximation performance of more than 94.6% of Brute-Force in the small scenario.

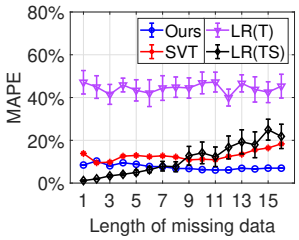
Finally, we change the capacity of edge nodes in both network scenarios, as illustrated in Fig. 10. The results in Fig. 10a show that our algorithm achieves more than 94% of optimal performance in the small network scenario. Also, *GTR* outperforms Random, Heuristic, and LPR on average by 43.31%, 22.61%, and 29.73% in the small networks, respectively, while 62.12%, 46.24%, and 47.20% in the large networks, respectively, as shown in Fig. 10a and Fig. 10b.

2) **Evaluations of traffic recovery**: We further conduct traffic data recovery experiments based on TVVS dataset, to evaluate the recovery performance of proposed *GTR* with different numbers of selected stations and different lengths of missing traffic data. In specifical, we randomly set one station (for example, station 13) as the recovery target. Considering the geospatial factor, we further select N adjacent traffic stations together with the target station to form the traffic volume matrix. By placing the traffic data of the target station in the first row of this matrix, we randomly generate L missing data points in the target station for recovery. Subsequently, we implement and test *GTR* and three baselines methods for data recovery by using using different numbers of stations to recovery target station. We run each experiment for 20 times and use the averaged performance as the final results.

First, as shown in Fig. 11, we evaluate the impact of missing data (*i.e.,* length of missing data points in the target station) on the recovery accuracy. We set a random station as recovery target and select 10 adjacent traffic stations to form the traffic volume matrix. Differently, we vary the length of missing data L from 1 to 16 and use all 10 stations to recover these missing data. As illustrated in Fig. 11a, Fig. 11b and Fig. 12a, the values of all evaluation metrics by different methods show a general ascending trend. This is the evidence to suggest that the recovery difficulty increases with the growing length of missing data. Despite that, the proposed *GTR* still preserves a robust performance in recovering traffic data, with the averaged MAE at 20, MAPE at 25 and MAPE at 10%. In contrast, SVT only maintains the high-accuracy performance before the length of missing data reaches 11. Similarly, LR(TS) shows linear descent in recovery accuracy when the number of missing data is larger. LR(TS) initially outperforms *GTR* and SVT when the length of missing data is below 7, as this method jointly consider spatial and temporal correlations in data recovery. However, the performance of LR(TS) is highly affected and degrades by the length of missing data, as its MAE, RMSE, and MAPE linearly grow with greater length of missing data. At last, the LR(T) method performs the worst in recovering traffic data across all conditions of missing data, resulting in an averaged MAE of 150, RMSE of 175 and MAPE of 60%.

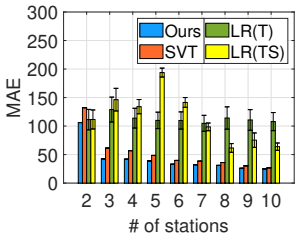
Second, we evaluate the impact of the number of stations on the recovery accuracy. Empirically, more sufficient nearby traffic volume data would help algorithms to recover missing data more accurately. As shown in Figs. 13a, 13b and 13c, the recovery accuracy in MAE, RMSE and MAPE change with different methods with different numbers of stations. Overall, with a larger number of traffic stations involved in data recovery, the recovery accuracy of all methods gradually improves. In specific, as LR(T) only performs first-degree polynomial fitting, it shows no significant enhancement with an unsatisfactory accuracy with MAE around 125, RMSE around 140 and MAPE around 45% across Fig. 13a to Fig. 13c. Meanwhile, LR(TS) exploits temporal-spatial correlations in traffic data and shows gradual improvement. With the increasing number of stations, LR(TS) outperforms LR(T) with the final MAE at 55, RMSE at 70 and MAPE at 25%. However, LR(TS) shows poor convergence with huge random variations and its performance is instable especially when the number of selected stations is below 5.

The proposed *GTR* and SVT algorithm are based on low rank minimization and they show significant superiority over linear regression methods in recovery accuracy and efficiency. For instance, by only using traffic data from two more stations for recovery (total number of stations as 3), *GTR* and SVT can achieve remarkable high-accuracy in data recovery, reducing both MAEs and RMSEs to around 50 and MAPE to below 20%. Moreover, both *GTR* and SVT exhibit notable convergence in data recovery, with all three evaluation metrics gradually decrease to lower values.

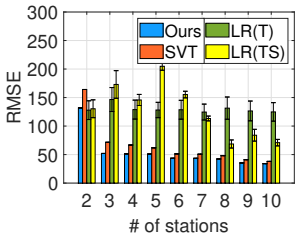


(a) MAPE

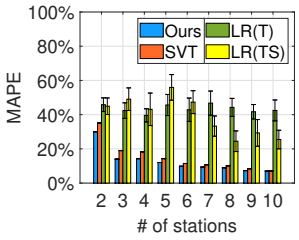
Fig. 12: Accuracy (MAPE) vs missing length.



(a) MAE



(b) RMSE



(c) MAPE

Fig. 13: Evaluations of recovery accuracy in different number of traffic stations, in terms of MAE, MRSE and MAPE.

Ultimately, when the number of traffic station is 10, *GTR* achieves the best recovery accuracy with MAE at 25, RMSE at 33 and MAPE at 8%.

In summary, the above experimental results further validate the effectiveness and efficiency of *GTR* in recovering traffic data. In comparison with baseline methods, *GTR* achieves remarkable improvement in recovery accuracy by only using 3 selected nearby stations, which could essentially reduce the computation overhead. Also, the performance of *GTR* is stable and robust, making it scalable to recover traffic data in different conditions.

6 Related Work

6.1 Traffic Data recovery in ITSs

With the pervasive deployment of large-scale intelligent transportation systems, missing data has become a ubiquitous and serious issue that directly influences the performance and integrity of ITSs. Therefore, numerous research works have devoted to recover accurate and complete traffic data with different methods [35]. For instance, Tak *et al.* [36] presented a modified k-Nearest-Neighbor method to impute the missing data in sectional units of road links. Moreover, Tang *et al.* [37] proposed a joint modeling framework to infer citywide traffic volume with GPS trajectory data and traffic counting data generated by surveillance cameras. Likewise, Chen *et al.* [38] leveraged parallel data paradigm (using real data and synthetic data) with Generative Adversarial Networks (GANs) to enhance traffic data mining and recovery. As traffic data is naturally spatial and temporal correlated across transportation networks, the spatio-temporal patterns have been further exploited for data recovery. As an example, Wang *et al.* [39] reconstructed the missing traffic data with low-rank matrix factorization, and further added a Laplacian regularization constraint to capture the spatiotemporal characteristics in the traffic data. Besides, Chen *et al.* [40] formulated the traffic data recovery as a high-dimensional problem of tensor completion and they adopted singular value decomposition to capture latent features to achieve robust recovery. More recently, multi-view learning methods have been proposed to fuse different data-driven algorithms and multiple data sources for traffic data estimation [41], [42].

Distinguished from above existing works, we are motivated to deliver real-time city-wide traffic data recovery system, thereby integrating the edge computing technique for traffic data processing. Our solution not only addresses the recovery accuracy of traffic data but also targets the optimal deployment of edge nodes for high-efficiency data processing.

6.2 Nodes Deployment in Edge Computing

Overall, our proposed system investigates the problem of how to deploy the edge nodes in the edge computing environment for traffic data management and recovery. There are many studies about the edge node deployment in edge computing, in which the most relevant studies are about cost minimization for edge node deployment in Mobile Edge Computing (MEC) [32], [43]–[48]. Caselli *et al.* [32], [43] studied how to deployment edge nodes for mobile networks to minimize the overall deployment cost by jointly optimizing edge node placement and routing schedule. Moreover, the references [46]–[48] focused on how to minimize the edge node deployment cost under the capacity and latency constraints, while Fan *et al.* [44] investigated how to tradeoff between the deployment cost and end-to-end latency for users. However, most of the existing studies consider the edge node deployment cost only, without considering the communication cost in ITSs. In our work, the incurred communication cost is non-negligible, since the traffic data collection in ITSs may consume large amounts of communication resources such as bandwidth. Therefore, the existing works cannot be applied to solve our problem.

7 Conclusion

In this paper, we propose *GTR*, an edge computing-empowered traffic data recovery system leveraging low-rank theory. First, we conduct experimental explorations based on large-scale traffic volume dataset of ITSs. The results uncover the serious issue of missing traffic data, while revealing its spatiotemporal correlations. Inspired by these observations, we propose a suboptimal edge node deployment algorithm with performance guarantee, and an accurate traffic data recovery scheme based on low-rank theory. Extensive theoretical analyses and traces-based evaluations demonstrate the performance of *GTR* outperform five baseline methods. In the future, we will explore the impacts of the data recovery performance on the edge node deployment, then improve the system design.

References

[1] Chuishi Meng, Xiuwen Yi, Lu Su, Jing Gao, and Yu Zheng. City-wide traffic volume inference with loop detector data and taxi trajectories. In *Proc. ACM SIGSPATIAL*, pages 1–10, 2017.

[2] Yanyan Xu, Qing-Jie Kong, Reinhard Klette, and Yuncai Liu. Accurate and Interpretable Bayesian MARS for Traffic Flow Prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2457–2469, 2014.

[3] System of traffic volume viewer. Website. https://www.rms.nsw.gov.au/about/corporate-publications/statistics/traffic-volumes/aadt-map/index.html.

[4] Traffic volume viewer. Website. https://www.rms.nsw.gov.au/about/corporate-publications/statistics/traffic-volumes/index.html.

[5] Qu Li, Li Li, Zhang Yi, and Jianming Hu. Ppca-based missing data imputation for traffic flow volume: A systematical approach. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):512–522, 2009.

[6] Daiheng Ni, John D Leonard, Angshuman Guin, and Chunxia Feng. Multiple imputation scheme for overcoming the missing values and variability issues in its data. *Journal of transportation engineering*, 131(12):931–938, 2005.

[7] Xianyuan Zhan, Yu Zheng, Xiuwen Yi, and Satish V Ukkusuri. Citywide traffic volume estimation using trajectory data. *IEEE Transactions on Knowledge and Data Engineering*, 29(2):272–285, 2016.

[8] Abbas Mehrabi, Matti Siekkinen, and Antti Yla-Jaaski. Edge computing assisted adaptive mobile video streaming. *IEEE Transactions on Mobile Computing*, 18(4):787–800, 2019.

[9] Pavel Mach and Zdenek Becvar. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *IEEE Communications Surveys and Tutorials*, 19(3):1628–1656, 2017.

[10] Nasir Abbas, Yan Zhang, Amir Taherkordi, and Tor Skeie. Mobile Edge Computing: A Survey. *IEEE Internet of Things Journal*, 5(1):450–465, 2018.

[11] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B Letaief. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4):2322–2358, 2017.

[12] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, Linhong Zhu, Rose Yu, and Yan Liu. Latent Space Model for Road Networks to Predict Time-Varying Traffic. In *Proc. ACM KDD*, pages 1525–1534, 2016.

[13] Zimu Zheng, Dan Wang, Jian Pei, Yi Yuan, Cheng Fan, and Fu Xiao. Urban traffic prediction through the second use of inexpensive big data from buildings. In *Proc. ACM CIKM*, pages 1363–1372, 2016.

[14] Teresa Pamuła. Impact of data loss for prediction of traffic flow on an urban road using neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):1000–1009, 2018.

[15] Ankur Sarker, Haiying Shen, and John A Stankovic. Morp: data-driven multi-objective route planning and optimization for electric vehicles. *Proc. ACM UbiComp*, 1(4):162, 2018.

[16] Public Holidays Global. Australia public holidays. Website. https://publicholidays.com.au/zh/anzac-day/.

[17] Tsuyoshi Idé, Takayuki Katsuki, Tetsuro Morimura, and Robert Morris. City-wide traffic flow estimation from a limited number of low-quality cameras. *IEEE Transactions on Intelligent Transportation Systems*, 18(4):950–959, 2016.

[18] Wei Yu, Fan Liang, Xiaofei He, William Grant Hatcher, Chao Lu, Jie Lin, and Xinyu Yang. A survey on the edge computing for the internet of things. *IEEE Access*, 6:6900–6919, 2017.

[19] Kapileswar Nellore and Gerhard P Hancke. A survey on urban traffic management system using wireless sensor networks. *Sensors*, 16(2):157, 2016.

[20] Suining He and Kang G. Shin. Spatio-temporal Adaptive Pricing for Balancing Mobility-on-Demand Networks. *ACM Transactions on Intelligent Systems and Technology*, 10(4):1–28, 2019.

[21] Katta G Murty. *Linear programming*. Springer, 1983.

[22] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-I. *Mathematical Programming*, 14(1):265–294, 1978.

[23] J. G. Oxley. *Matroid theory*. Oxford University Press, 1992.

[24] Jon Lee, Vahab S. Mirrokni, Viswanasth Nagarajan, and Maxim Sviridenko. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM Journal on Discrete Mathematics*, 23(4):2053–2078, 2009.

[25] D.C. Lay. *Linear Algebra and Its Applications*. Pearson, 5th edition, 2012.

[26] Xuangou Wu, Zhaobin Chu, Panlong Yang, Chaocan Xiang, Xiao Zheng, and Wenchao Huang. Tw-see: Human activity recognition through the wall with commodity wi-fi devices. *IEEE Transactions on Vehicular Technology*, 68(1):306–319, 2018.

[27] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[28] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Sparse and Low-Rank Matrix Decompositions. *IFAC Proceedings Volumes*, 42(10):1493–1498, 2009.

[29] Stephen Boyd, Stephen Boyd, and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

[30] Shiqian Ma, Donald Goldfarb, and Lifeng Chen. Fixed point and bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1-2):321–353, 2011.

[31] Simon Foucart and Holger Rauhut. An invitation to compressive sensing. In *A mathematical introduction to compressive sensing*, pages 1–39. Springer, 2013.

[32] Alberto Ceselli, Marco Premoli, and Stefano Secci. Mobile edge cloud network design optimization. *IEEE/ACM Transactions on Networking (TON)*, 25(3):1818–1831, 2017.

[33] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982, 2010.

[34] Xiaochen Fan, Chaocan Xiang, Liangyi Gong, Xiangjian He, Chao Chen, and Xiang Huang. Urbanedge: deep learning empowered edge computing for urban iot time series prediction. In *Proc. ACM TURC*, page 12, 2019.

[35] Ibai Laña, Ignacio Iñaki Olabarrieta, Manuel Vélez, and Javier Del Ser. On the imputation of missing data for road traffic forecasting: New insights and novel techniques. *Transportation research part C: emerging technologies*, 90:18–33, 2018.

[36] Sehyun Tak, Soomin Woo, and Hwasoo Yeo. Data-driven imputation method for traffic data in sectional units of road links. *IEEE Transactions on Intelligent Transportation Systems*, 17(6):1762–1771, 2016.

[37] Xianfeng Tang, Boqing Gong, Yanwei Yu, Huaxiu Yao, Yandong Li, Haiyong Xie, and Xiaoyu Wang. Joint modeling of dense and incomplete trajectories for citywide traffic volume inference. In *Proc. WWW*, pages 1806–1817, 2019.

[38] Yuanyuan Chen, Yisheng Lv, and Fei-Yue Wang. Traffic flow imputation using parallel data and generative adversarial networks. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[39] Yang Wang, Yong Zhang, Xinglin Piao, Hao Liu, and Ke Zhang. Traffic data reconstruction via adaptive spatial-temporal correlations. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1531–1543, 2018.

[40] Xinyu Chen, Zhaocheng He, and Jiawei Wang. Spatial-temporal traffic speed patterns discovery and incomplete data recovery via svd-combined tensor decomposition. *Transportation research part C: emerging technologies*, 86:59–77, 2018.

[41] Linchao Li, Jian Zhang, Yonggang Wang, and Bin Ran. Missing value imputation for traffic-related time series data based on a multi-view learning method. *IEEE Transactions on Intelligent Transportation Systems*, 20(8):2933–2943, 2018.

[42] Rong Du and Shudong Chen. Multi-view low rank representation for multi-source traffic data completion. *International Journal of Intelligent Transportation Systems Research*, 17(3):200–211, 2019.

[43] Alberto Ceselli, Marco Premoli, and Stefano Secci. Cloudlet network design optimization. In *Proc. IEEE IFIP Networking*, 2015.

[44] Qiang Fan and Nirwan Ansari. Cost Aware cloudlet Placement for big data processing at the edge. In *Pro. IEEE ICC*, pages 1–6, 2017.

[45] Xiaochen Fan, Xiangjian He, Deepak Puthal, Shiping Chen, Chaocan Xiang, Priyadarsi Nanda, and Xunpeng Rao. CTOM: Collaborative task offloading mechanism for mobile cloudlet networks. In *Proc. IEEE ICC*, pages 1–6, 2018.

[46] Sourav Mondal, Goutam Das, and Elaine Wong. CCOMPASSION: A Hybrid Cloudlet Placement Framework over Passive Optical Access Networks. In *Pro. IEEE INFOCOM*, pages 216–224, 2018.

[47] Feng Zeng, Yongzheng Ren, Xiaoheng Deng, and Wenjia Li. Cost-effective edge server placement in wireless metropolitan area networks. *Sensors*, 19(1):32–44, 2019.

[48] Sourav Mondal, Goutam Das, and Elaine Wong. Cost-optimal cloudlet placement frameworks over fiber-wireless access networks for low-latency applications. *Journal of Network and Computer Applications*, 138:27–38, 2019.