INFORMS Journal on Data Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

# Low-Rank Robust Subspace Tensor Clustering for Metro Passenger Flow Modeling

Nurretin Dorukhan Sergin; , Jiuyun Hu; , Ziyue Li; , Chen Zhang; , Fugee Tsung, Hao Yan;

To cite this article:
Nurretin Dorukhan Sergin; , Jiuyun Hu; , Ziyue Li; , Chen Zhang; , Fugee Tsung, Hao Yan; (2025) Low-Rank Robust Subspace Tensor Clustering for Metro Passenger Flow Modeling. INFORMS Journal on Data Science 4(1):33-50. https://doi.org/10.1287/ijds.2022.0028

https://pubsonline.informs.org/journal/ijds

INFORMS JOURNAL ON DATA SCIENCE
Vol. 4, No. 1, January–March 2025, pp. 33–50
ISSN 2694-4022 (print), ISSN 2694-4030 (online)

# Low-Rank Robust Subspace Tensor Clustering for Metro Passenger Flow Modeling

**Nurretin Dorukhan Sergin,[a] Jiuyun Hu,[a] Ziyue Li,[b,c] Chen Zhang,[d] Fugee Tsung,[e,f] Hao Yan[a,*]**

[a] School of Computing and Augmented Intelligence, Arizona State University, Tempe, Arizona 85281; [b] Cologne Institute for Information Systems, University of Cologne, 50923 Cologne, Germany; [c] EWI gGmbH, University of Cologne, 50923 Cologne, Germany; [d] Department of Industrial Engineering, Tsinghua University, Beijing 100190, China; [e] Department of Industrial Engineering and Decision Analytics, Hong Kong University of Science and Technology, Hong Kong; [f] Information Hub, Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511457, China

*Corresponding author

**Contact:** dorukhansergin@fb.com (NDS); jiuyunhu@asu.edu (JH); zlibn@wiso.uni-koeln.de (ZL); zhangchen01@tsinghua.edu.cn, https://orcid.org/0000-0002-8319-524X (CZ); season@ust.hk (FT); haoyan@asu.edu, https://orcid.org/0000-0002-4322-7323 (HY)

**Received:** October 8, 2022
**Revised:** November 5, 2023
**Accepted:** March 7, 2024
**Published Online in Articles in Advance:** September 19, 2024

**https://doi.org/10.1287/ijds.2022.0028**



**Abstract.** Tensor clustering has become an important topic, specifically in spatiotemporal modeling, because of its ability to cluster spatial modes (e.g., stations or road segments) and temporal modes (e.g., time of day or day of the week). Our motivating example is from subway passenger flow modeling, where similarities between stations are commonly found. However, the challenges lie in the innate high-dimensionality of tensors and also the potential existence of anomalies. This is because the three tasks, that is, dimension reduction, clustering, and anomaly decomposition, are intercorrelated with each other, and treating them in a separate manner will render a suboptimal performance. Thus, in this work, we design a tensor-based subspace clustering and anomaly decomposition technique for simultaneous outlier-robust dimension reduction and clustering for high-dimensional tensors. To achieve this, a novel low-rank robust subspace clustering decomposition model is proposed by combining Tucker decomposition, sparse anomaly decomposition, and subspace clustering. An effective algorithm based on Block Coordinate Descent is proposed to update the parameters. Prudent experiments prove the effectiveness of the proposed framework via the simulation study, with a gain of +25% clustering accuracy over benchmark methods in a hard case. The interrelations of the three tasks are also analyzed via ablation studies, validating the interrelation assumption. Moreover, a case study in station clustering based on real passenger flow data is conducted, with quite valuable insights discovered.

**History:** Bianca Maria Colosimo served as the senior editor for this article.

**Funding:** H. Yan is partially funded by DOE [DE-EE0009354] and NSF [CMMI 2316654]. F. Tsung is partially funded with the RGC [GRF 16201718 and 16216119]. The authors appreciate the help from the Hong Kong MTR Co. research, marketing, and customer service teams.

**Data Ethics & Reproducibility Note:** The code capsule is available on Code Ocean at https://codeocean.com/capsule/6536164/tree and in the e-Companion to this article (available at https://doi.org/10.1287/ijds.2022.0028).

**Keywords:** subspace clustering • tensor decomposition • anomaly detection • spatiotemporal analysis

## 1. Introduction

Higher-order tensors have been actively used in research because they have an inclination to successfully preserve the complicated correlation structures of data. A tensor can be defined mathematically as multidimensional arrays (Kolda and Bader 2009). The order of a tensor is the number of dimensions, also known as modes. Tensor clustering is a recent generalization of the basic one-dimensional clustering problem to a high-dimensional version, and it seeks to partition an order-$K$ input tensor into coherent subtensors (e.g., slices on one mode) while optimizing some cluster-related measuring criteria (Jegelka et al. 2009).

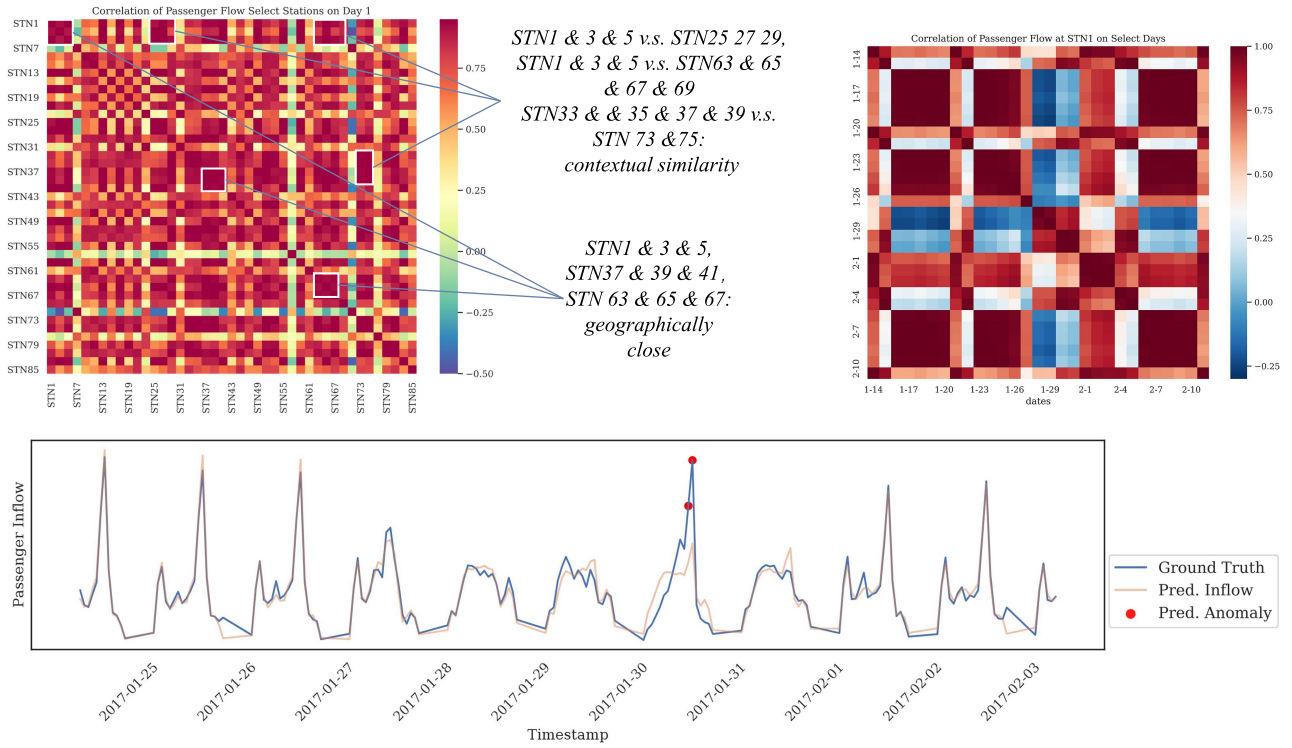
Tensor clustering has also become an important topic in spatiotemporal modeling (Bahadori et al. 2014, Sun and Li 2019, Mao et al. 2022), because of its ability to cluster complicated spatial modes (e.g., stations, road segments, etc.) and temporal modes (e.g., time of day or day of the week). For example, in the modeling of passenger flow, similarities between spatial elements are commonly found in spatiotemporal data (Li et al. 2020a). These clusters may be a result of natural geographical locations (e.g., neighboring stations) or contextual information (e.g., points of interest) in the context of where the data are collected. For example, in residential areas, stations often have a large number of inflow passengers in the morning on weekdays; in the business areas, instead, inflow peaks are expected in the afternoon on weekdays. Exploring metro station clusters can help public transportation management, such

as in operational efficiency, strategic planning, anomaly detection, policy making, and land use planning. Overall, the following two challenges present for data collected from such complex spatiotemporal systems:

1. **Complex spatiotemporal structure in high-dimensional data.** The dimensionality of the tensor data is often high (especially in the temporal domain), and traditional statistical methods may suffer from the "curse of dimensionality." Furthermore, data points in the high-dimensional tensor often present complicated and high-order variations. For the temporal dependencies, passenger flow patterns show a strong periodicity with a period of 7 days, which denotes the weekly transit patterns (Li et al. 2020b). This can also be seen in Figure 1 (right). Furthermore, on the same day, there are also complicated dependencies. For example, there are often two inflow peaks for the multifunctional regions in the daily profile during weekdays, which represent the morning and afternoon time for the daily commute, whereas a dominant morning inflow peak is usually observed in residential regions and an evening inflow peak commonly in business regions. For spatial structures, we create a correlation heatmap of passenger flows in different stations on day 1, as shown in Figure 1 (left). Figure 1 (left) demonstrates two kinds of passenger flow similarities. The geographical similarity is based on the distance between the stations, whereas the contextual similarity is based on the function of the stations.

2. **Existence of sparse anomalies.** The tensor may contain sparse outliers because of measurement errors or rare events in urban passenger flow, such as weather, special occasions, etc. The sparse outliers may refer to the sudden increase of demand; this may be related to some unusual events happening, such as concerts and festivals, or this could even be a result of maintenance or breakdowns in other stations whose spillover effect flows into the station in question. Identifying these events can also help prepare for the sudden increase in demand. To better understand the sparse anomaly feature, we create a correlation heatmap of passenger flows at station 1 on different days, as shown in Figure 1 (right). Figure 1 (right) shows that on most days, the passenger flow follows the temporal structure of the similarities of the weekday and weekend, except for four days. These days serve as the sparse anomaly in the data. Finally, to further illustrate the existence of sparse anomalies, we have included another plot on the sparse anomalies, as shown in Figure 1 (bottom). A good example is illustrated here of an unexpected peak in the afternoon of January 30, 2017, at the Fo Tan station. The peak is because of

**Figure 1.** (Color online) Correlation Heatmaps




*Notes.* (Left) Geographical and contextual similarity between stations. (Right) Temporal structure and sparse anomaly. (Bottom) Existence of sparse anomalies.

people going home from the Lunar New Year horse racing event, which takes place on the race course accessible from the Fo Tan station. Therefore, in this research, we define the "anomaly" study as the entry-level pieces of the tensor data that do not fit the underlying distribution of the main features of the data.

In the literature, most of the methods implement a three-step approach for tensor clustering. First, tensor decomposition can be used to reduce the tensor dimensionality. Then, sparse outliers can be removed by robust tensor decomposition, and lastly, clusters can be identified by existing clustering algorithms such as K-means or subspace clustering. In the first step of tensor decomposition, related to dimension reduction techniques, nonnegative tensor decomposition (Lin et al. 2018), Tucker decomposition (Kolda and Bader 2009), and CP decomposition (Kolda and Bader 2009) are widely used. In the step of anomaly detection, to handle sparse outliers, Hu and Work (2021) proposed a robust tensor decomposition method that can handle outlier data while achieving dimension reduction. Unlike standard robust tensor decomposition applications, they used the $L_{2,1}$ norm as they assumed fiber-sparse anomalies. Finally, in the step of clustering, it is quite common to apply clustering algorithms to the reduced-dimensional space. For example, Yang et al. (2019) proposed a hybrid approach that combines tensor decomposition with spectral clustering in a low-dimensional space.

However, none of these aforementioned works considers the interrelated nature of clustering, outlier removal, and dimension reduction (Aggarwal 2015), with the interrelation threefold as follows:

1. **Dimension reduction and clustering:** Clustering directly in the high dimensions will suffer from the curse of dimensionality, given that data become much more sparse and Euclidean distance is not a good distance measure in high-dimensional space. Therefore, dimension reduction techniques are often needed to project data in a low-dimensional space before clustering methods are applied (Yang et al. 2019).

2. **Outlier decomposition and clustering:** On the other hand, if outliers on stations are not correctly identified or decomposed, the results of clustering for the overall data from the normal patterns will also be compromised.

3. **Dimension reduction and outlier decomposition:** Furthermore, a direct dimension reduction on the outlier-corrupted data will also lead to the introduction of erroneous information into the lower-dimensional products, that is, decomposed latent matrices, which will hurt the downstream task performance (i.e., clustering) using the low-dimensional space.

In conclusion, we claim that the tensor clustering problem, dimension reduction, and outlier decomposition problem are intercorrelated tasks and tied to each other, and isolated treatment of them may fail to provide satisfactory performance.

In this paper, we propose a low-rank robust tensor subspace decomposition (LRTSD) technique to achieve dimensionality reduction, spatial clustering, and sparse anomaly decomposition *simultaneously*. The proposed method is inspired by the recent development in the following three fields: Tucker decomposition (Kolda and Bader 2009), tensor subspace clustering (Vidal 2011, Zhang et al. 2015), and sparse anomaly decomposition (Yan et al. 2017, Li et al. 2022a) (In this paper, we interchangeably use the term outlier and anomaly to represent the data which behave outstandingly). Tucker decomposition is used as a dimensionality reduction technique, which reduces the original high-dimensional tensor data into a lower-dimensional core tensor whereas fitting orthonormal bases (Tucker 1966). Furthermore, the proposed method is inspired by the sparse anomaly decomposition methods (Yan et al. 2017, Li et al. 2022a), which aim to decompose the sparse anomaly component from the background component, with the $L_1$ norm applied to encourage the sparsity of detected anomalies. Finally, the proposed method is also inspired by the subspace clustering method (Vidal 2011), which has achieved especially good accuracy in high-dimensional clustering problems (Parsons et al. 2004).

In this paper, we innovatively apply subspace clustering to the core tensor in the Tucker decomposition, which is inherently more resilient to corruption/anomaly. To our knowledge, this is the first instance of such integration, and our simulation studies substantiate its superior performance. The amalgamation of these three components significantly increases the complexity of the model. To address this, we have developed an efficient optimization algorithm to overcome this challenge, ensuring the generalizability, computational feasibility, and enhanced accuracy of the tensor clustering model.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of the basic tensor notations and multilinear algebra operation and subspace clustering. Section 3 provides a detailed introduction to the proposed subspace Tucker decomposition methods as well as an efficient optimization algorithm. Section 4 conducts a simulation study, and Section 5 applies the proposed method to a real data set based on Hong Kong Metro data. Section 6 concludes and presents some future works.

## 2. Literature Review

In this section, we will briefly review the methodology related to tensor decomposition and subspace clustering methods for tensor data.

### 2.1. Tensor Decomposition

Related to robust tensor decomposition methods, most existing works apply the dimension reduction approach

and tensor decomposition method for feature extraction.

Historically, all data are vectorized to the sample dimension, and the ordinary principal component analysis (PCA) is applied to the vectorized data to extract and monitor features, which is typically termed vectorized PCA (VPCA) (Nomikos and MacGregor 1994). However, such a method sacrifices the detection power, because the vectorization on high-dimensional data destroys the innate cross-dimensional relations.

Tensor has been introduced as an efficient data structure to preserve interdimensional correlations well, such as complex spatiotemporal correlations among traffic data, as mentioned before. Tensor decomposition has been developed accordingly to extract the features from the tensor data. Various tensor decomposition methods have been developed, namely, (1) CANDECOMP/PARAFAC (CP) decomposition (Hitchcock 1927), which represents a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$ as the weighted summation of a set of rank-one tensors, and (2) Tucker decomposition (Tucker 1966), which decomposes a tensor into a core tensor $\mathcal{C} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_K}$ multiplied by a mode matrix $\boldsymbol{U}^{(k)} \in \mathbb{R}^{I_k \times J_k}$ along each dimension. The proposed method is based on the Tucker decomposition, and detailed operations of the Tucker decomposition will be introduced in Section 3.1.

Yan et al. (2015) proposed to utilize several anomaly detection and monitoring schemes based on Tucker decomposition (Tucker 1966), multilinear PCA (Lu et al. 2008a), CP decomposition (Hitchcock 1927), and uncorrelated multilinear PCA (Lu et al. 2008b) with the application for image-based anomaly detection. However, such approaches do not consider the properties of sparse outliers for anomaly detection. To address this problem, tensor-based sparse anomaly decomposition techniques have been proposed (Fanaee-T and Gama 2016) and applied to video-imaging data (Yan et al. 2018), urban traffic flow prediction (Sofuoglu and Aviyente 2021), crime rate monitoring (Zhao et al. 2021), pandemic monitoring (Zhao et al. 2020), etc. However, the major limitation of these existing tensor-based anomaly decomposition methods is that they assume that the tensor follows a joint smooth and low-rank representation, which is not feasible for high-dimensional tensor data with multiple latent clusters behind the data distribution (Li et al. 2020a). However, a direct application of the decomposition method to the tensor data with outliers for clustering will render an oversimplified model.

## 2.2. Tensor Subspace Clustering

On the other hand, there is some recent literature focusing on clustering methods for tensor data. Drakopoulos et al. (2019) gave an extensive methodology review of the tensor clustering. Some methods such as topic-model-based clustering (Li 2021, Li et al. 2022b), dynamic tensor clustering (Sun and Li 2019), multiview subspace clustering (Zhang et al. 2015), dynamic subspace clustering (Zhang et al. 2020), and joint K-means and high-order singular value decomposition (SVD) (Huang et al. 2008) are proposed. Among them, subspace clustering has been a popular method for high-dimensional clustering (Parsons et al. 2004), which learns data representation in certain low-dimensional subspaces and clusters of the data points. The subspace clustering is commonly formulated based on the data's self-expression property (Gao et al. 2015), representing the original input data $\boldsymbol{X}$ by itself: $\boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z} + \boldsymbol{E}$, where $\boldsymbol{Z} \in \mathbb{R}^{n \times n}$ is the subspace representation matrix, and the nonzero elements in $\boldsymbol{Z}$ correspond to the data points from the same subspace. Different properties could be achieved by introducing regularization terms into the representation matrix. More details will be explained in Section 3.2. However, like the other tensor clustering methods mentioned above, these methods are typically used on the high-dimensional tensor directly and may suffer from the curse of dimensionality. For example, similar to tensor regression (Lock 2018, Gahrooei et al. 2021), directly applying self-expression on high-dimensional data leads to an even higher-dimensional self-expressiveness term.

Recently, there have been some works focusing on combining tensor low-rank decomposition and subspace clustering (Fu et al. 2014, 2016). Specifically, Fu et al. (2016) proposed a subspace clustering that incorporates sparse dictionary learning into Tucker decomposition, where the Tucker core tensor could be the input tensor itself (Fu et al. 2014) or the inverse matricization product of the dictionary and sparse representation (Fu et al. 2016). However, these subspace clustering methods are directly targeted at high-dimensional tensors, and are not suitable for tensors with sparse anomalies. Recently, there have been some efforts to combine subspace clustering methods with anomaly detection to handle the challenges of anomalies in multiclustered data sets.

Recently, there have been a few tensor clustering methods developed based on deep learning and neural networks. Wang et al. (2023) proposed the Tensorized Hypergraph Neural Networks (THNN) to learn the hypergraph structure on higher dimensions. In two-dimensional cases, further methods, such as spectral clustering, can be performed to cluster the nodes. However, this method only considered the hypergraph structure and did not take subspace learning and anomaly into consideration. Zhao et al. (2022a) proposed the Reinforced Tensor Graph Neural Network (RTGNN) to learn multi-view graph data. This method contains three modules, which makes it time-consuming. Tucker decomposition only takes part in the initialization, and the anomaly is neglected in this method. Huang et al. (2023) extensively

explored various methods, specifically on hyperspectral image clustering. The improvement achieved by neural network-based methods compared with tensor decomposition and self-regression-based methods is data dependent. However, compared with the tensor-based approach, neural networks tend to become time-consuming and require many training samples.

### 2.3. Sparse Anomaly Decomposition

Anomaly is usually detected based on the assumption that anomalous data do not conform to expected behavior with outstandingly different features from the homogeneous background (Chandola et al. 2009). Traditional methods detect the anomaly based on distance and density, which assume that anomalies lie far away from the background or locate in a less dense area. Various definitions of distance or density have been introduced (Du and Zhang 2014). For example, Sun et al. (2022) enhanced the CP decomposition by a deep CP decomposition neural network for feature extraction with an anomaly detection step. Nonetheless, to further achieve the separation of the anomalies, a further step such as clustering is still necessary. In this case, a one-step decomposition is favored in practice.

This paper is also motivated by the recent development of the decomposition-based methods (Yan et al. 2017, Sofuoglu and Aviyente 2021, Li et al. 2022a), which try to decompose the original input data into several components directly. For example, Yan et al. (2017) decomposed input data into the smooth background and sparse anomaly, with smoothness regularization on the background and sparsity penalty on the anomaly. Later, the extensions to spatiotemporal data (Yan et al. 2018) and deep-learning-based decomposition models (Zhao et al. 2022b) were proposed. For example, Li et al. (2022a) considered more components, namely, smooth background, sparse anomaly, sample-specific deviation, and random noise, and they introduced a transfer learning mechanism (Pan and Yang 2009) to solve the cold-start problem for anomaly decomposition. However, these methods are not developed for tensor data, and the decomposed components are predefined in a fixed manner. Recently, Shen et al. (2022) proposed a tensor-based decomposition method to decompose the static tensor background and smooth foreground. However, the method is designed for static background and cannot be directly used to learn the tensor data from different subspaces. Our method instead is to combine the sparse decomposition method with the tensor decomposition approach, and it learns the hidden components in a data-driven and unsupervised manner via tensor subspace.

To summarize, this work is the first that achieves dimensionality reduction, spatial clustering, and anomaly decomposition simultaneously.

## 3. Methodology

In this section, we will first review the basic tensor algebra in Section 3.1. Then, we will give a brief review of the recent development of subspace clustering in Section 3.2. More specifically, we will discuss the proposed formulation combining tensor subspace clustering in Section 3.3. We will then discuss how to solve the formulation efficiently in Section 3.4.

### 3.1. Basic Tensor Notation and Multilinear Algebra

In this section, we introduce basic notations, definitions, and operators in multilinear (tensor) algebra that are used in this work. Throughout the paper, scalars are denoted by lowercase italic letters, for example, $a$; vectors by lowercase boldface letters, for example, $\boldsymbol{a}$; matrices by uppercase boldface letters, for example, $\boldsymbol{A}$; and tensors by calligraphic letters, for example, $\mathcal{A}$. For example, an order-$K$ tensor is represented by $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_K}$, where $I_k$ represents the mode-$k$ dimension of $\mathcal{A}$. The mode-$k$ product of a tensor $\mathcal{A}$ by a matrix $\boldsymbol{V} \in \mathbb{R}^{P_k \times I_k}$ is defined as follows:

$$(\mathcal{A} \times_k \boldsymbol{V})(i_1, \ldots, i_{k-1}, j_k, i_{k+1}, \ldots, i_K) = \sum_{i_k} A(i_1, \ldots, i_k, \ldots, i_K) V(j_k, i_k).$$

The Frobenius norm of a tensor $\mathcal{A}$ can be defined as follows:

$$\|\mathcal{A}\|_F^2 = \sum_{i_1, \ldots, i_K} A(i_1, \ldots, i_k, \ldots, i_K)^2.$$

The $n$-mode unfold operator maps the tensor $\mathcal{A}$ into matrix $\boldsymbol{A}_{(n)}$, where the columns of $\boldsymbol{A}_{(n)}$ are the $n$-mode vectors of $\mathcal{A}$.

Tucker decomposition decomposes a tensor into a core tensor multiplied by a matrix along each mode, formulated as follows:

$$\mathcal{A} = \mathcal{C} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \cdots \times_K \boldsymbol{U}^{(K)},$$

where $\mathcal{C} \in \mathbb{R}^{J_1 \times J_2 \cdots \times J_K}$ is the core tensor, and $\boldsymbol{U}^{(k)}$ is an orthogonal $I_k \times J_k$ matrix and is a principal component mode-$k$. The definition of Kronecker product is as follows: Supposing $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{p \times q}$ are matrices, the Kronecker product of these matrices, denoted by $\boldsymbol{A} \otimes \boldsymbol{B}$, is an $mp \times nq$ block matrix, which can be formalized as follows:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \cdots & a_{1n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & \cdots & a_{mn}\boldsymbol{B} \end{bmatrix}.$$

### 3.2. Subspace Clustering

An important but often unnoticed assumption of PCA is that the data live in a *single* low-dimensional subspace. This assumption may be too crude for many

applications, including spatiotemporal problems such as motion segmentation (Vidal et al. 2008). It is more likely in such cases that the data live in a mixture of subspaces instead of a single one (see Figure 2). The field of subspace clustering deals with problems of this nature and has attracted growing interest in recent years (Guo et al. 2019, Liu et al. 2019).

The central problem of subspace clustering is that the grouping of points and their respective clusters are both unknown in the beginning. Various frameworks have been proposed in recent years, ranging from direct algebraic solutions to probabilistic approaches (Vidal 2011), whereas spectral clustering (von Luxburg 2007) dominates the approaches proposed in recent years because of its superior performance over other frameworks. In short, spectral clustering aims to build a similarity graph where nodes are individual samples and the edges are similarity links. Algorithms such as graph cut can be used later to identify the clusters.

An effective way to obtain similarity is to use the self-expressiveness property (Parsons et al. 2004). Let $\boldsymbol{D} \in \mathbb{R}^{n \times p}$ be a data matrix. The self-expressiveness can be defined with the equality $\boldsymbol{D} = \boldsymbol{Z}\boldsymbol{D}$ where the self-expression matrix $\boldsymbol{Z} \in \mathbb{R}^{n \times n}$ helps to express each point as a linear combination of other points in the data. This matrix can later be used to find clusters, as explained above.

Spectral subspace clustering methods differ in how they regulate the affinity matrix to avoid the trivial solution of identity matrix $\boldsymbol{Z} = \boldsymbol{I}$. Sparse subspace clustering (SSC) imposes sparsity on the affinity matrix (Yang et al. 2015), which is formulated as follows:

$$\min_{\boldsymbol{Z}} \quad \|\boldsymbol{Z}\|_1 + \frac{\lambda}{2}\|\boldsymbol{D} - \boldsymbol{Z}\boldsymbol{D}\|_F^2 \quad s.t. \quad \mathrm{diag}(\boldsymbol{Z}) = 0,$$

where the $\|.\|_1$ represent the L1 penalty of the self-expression matrix. By applying this penalty, the self-expression matrix $\boldsymbol{Z}$ is also assumed to be low rank. The low-rank representation method was proposed by Liu et al. (2013). Similarly, another method of low-rank subspace clustering was developed by Vidal and Favaro (2014), where the core formulation uses the nuclear norm of the self-expression matrix $\boldsymbol{Z}$ defined by the sum of its singular values as the best convex approximation of the rank of $\boldsymbol{Z}$:

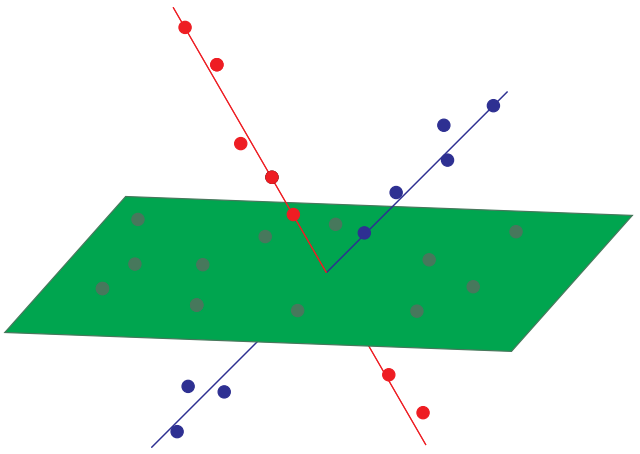
$$\min_{\boldsymbol{Z},\boldsymbol{D},\boldsymbol{A}} \|\boldsymbol{Z}\|_* + \frac{\tau}{2}\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{Z}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{D} - \boldsymbol{A}\|_F^2 \quad s.t. \quad \boldsymbol{Z}^\top = \boldsymbol{Z}.$$

Thresholding ridge regression (TRR) (Peng et al. 2015) is an improvement on SSC in the sense that it relaxes the requirement to assume the structure of the affinity matrix, whether it is low rank or sparse. The formulation is also much simpler:

$$\min_{\boldsymbol{Z}} \quad \lambda\|\boldsymbol{Z}\|_F^2 + \|\boldsymbol{D} - \boldsymbol{D}\boldsymbol{Z}\|_F^2.$$

The regularization parameter $\lambda$ represents the trade-off between the reconstruction fidelity and clustering ability, which may be set with respect to the use case. However, all these variations focus on the matrix formulation and cannot be applied to the tensor formulation.

**Figure 2.** (Color online) Data Drawn from Subspaces



*Notes.* Data are drawn from two one-dimensional subspaces (red and blue) and one two-dimensional subspace (green). Adapted from Vidal (2011).

### 3.3. Proposed LRTSD Model

Without loss of generality, we will illustrate the proposed method using a three-dimensional tensor. The methodology can be easily extended into higher-order tensors. For a three-dimensional tensor, we assume that $\mathcal{X} \in \mathbb{R}^{N \times I_2 \times I_3}$ represents the original data, where the first dimension denotes the sample dimension that requires the clustering. Furthermore, we assume that this tensor $\mathcal{X}$ can be decomposed into the normal-variation data $\mathcal{L}$ and the outlier tensor $\mathcal{A}$ (i.e., represents anomalous inflows in the passenger flow data), which is assumed to be sparse. We assume that the normal-variation data $\mathcal{L}$ has a low-rank structure, which is a common assumption for high-dimensional tensors for dimension reductions. Furthermore, without loss of generality, we assume the first mode of tensor $\mathcal{X}$ is the dimension that needs further clustering. We name the first mode of the tensor (mode 1) as the clustering mode and the other modes (modes 2 and 3) as the nonclustering modes.

For example, the passenger flow data can be represented as a tensor $\mathcal{X} \in \mathbb{R}^{N \times I_2 \times I_3}$, with the three modes representing the station (with $N$ stations), the day of the week (with $I_2 = 7$ days), and the 5-minute time intervals (with $I_3 = 247$ intervals from 5 a.m. to 1:30 a.m.) within each day for the period when the subway station is open, respectively. Each element of the tensor $X_{i,d,t}$ represents the inflow into stations $i$ at day $d$ and time interval $t$. Here, the station $i$ is the clustering mode, and $d$ and $t$ are the nonclustering modes, given that the station $s$ is often heterogeneous and may exhibit multiple clusters.

In this case, we propose to use a partial Tucker decomposition on the nonclustering modes as

$$\mathcal{L} = \mathcal{C} \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3, \tag{1}$$

where a core tensor $\mathcal{C} \in \mathbb{R}^{N \times P_2 \times P_3}$ reduces the dimensionality of the original tensor. $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ are respective orthogonal matrices for the nonclustering modes. We propose to find subspaces in the latent space on the core tensor $\mathcal{C}$ as opposed to original space, as this is computationally more efficient. Here, the notations that will be used in the paper are given in Table 1.

Using the thresholding ridge regression framework (Peng et al. 2015), we utilize the self-expressive expression $\mathcal{C} - \mathcal{C} \times_1 \boldsymbol{Z}$ by regularizing the self-expression matrix $\boldsymbol{Z} \in \mathbb{R}^{N \times N}$ using the Frobenius norm. Formally, the model is described as an optimization problem:

$$\min_{\boldsymbol{Z}, \mathcal{C}, \boldsymbol{U}_2, \boldsymbol{U}_3, \mathcal{A}} \begin{aligned} &\frac{1}{2}\|\boldsymbol{Z}\|_F^2 + \frac{\lambda_z}{2}\|\mathcal{C} - \mathcal{C} \times_1 \boldsymbol{Z}\|_F^2 + \lambda_a \|\mathcal{A}\|_1 \\ &+ \frac{\lambda_e}{2}\|\mathcal{X} - \mathcal{A} - \mathcal{C} \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3\|_F^2 \end{aligned} \tag{2}$$

$$\boldsymbol{U}_2^\top \boldsymbol{U}_2 = \boldsymbol{I} \tag{3}$$

$$\boldsymbol{U}_3^\top \boldsymbol{U}_3 = \boldsymbol{I} \tag{4}$$

In summary, each term in Equation (2) achieves the following purpose:

- $\|\mathcal{X} - \mathcal{A} - \mathcal{C} \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3\|_F^2$ is the error term between the original input data $\mathcal{X}$ and the two decomposed components, that is, anomaly $\mathcal{A}$ and low-rank structure $\mathcal{L} = \mathcal{C} \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3$;
- $\|\mathcal{C} - \mathcal{C} \times_1 \boldsymbol{Z}\|_F^2$ is the self-expression term in tensor subspace clustering and is imposed on the decomposed core tensor. $\|\boldsymbol{Z}\|_F^2$ is the ridge regularization;
- $\|\mathcal{A}\|_1$ is the sparsity regularization term for the anomaly.

To this end, this model achieves what we promised: "a simultaneous framework for dimension reduction, spatial clustering, and anomaly detection."

## 3.4. Optimization Procedure

In this section, we propose an efficient algorithm to solve the proposed optimization problem Equations (2)–(4). We will utilize the Block Coordinate Descent procedure to obtain each one of the model parameter blocks $\boldsymbol{Z}, \mathcal{C}, \boldsymbol{U}_2, \boldsymbol{U}_3$, and $\mathcal{A}$ iteratively while keeping the others fixed from the last iteration. When Algorithm 1 terminates through convergence, graph cut is applied to $\boldsymbol{L} = \boldsymbol{Z}^\top + \boldsymbol{Z}$ to achieve clusters, and $\mathcal{A}$ can be analyzed for detected point anomalies. The rationale of the proofs of the propositions is relegated to the Online Appendix.

**Table 1.** Notations for Proposed LRTSD Model

| Notation | Explanation |
|---|---|
| $\mathcal{X} \in \mathbb{R}^{N \times I_2 \times I_3}$ | $N$, number of samples; $I_2, I_3$, feature dimensions |
| $\mathcal{C} \in \mathbb{R}^{N \times P_2 \times P_3}$ | Core tensor, with reduced dimensions $P_2$, $P_3$ |
| $\boldsymbol{U}_l \in \mathbb{R}^{I_l, P_l}, l = 2,3$ | Orthogonal matrices for dimension reduction |
| $\mathcal{A} \in \mathbb{R}^{N \times I_2 \times I_3}$ | Anomaly tensor, same size as $\mathcal{X}$ |
| $\boldsymbol{Z} \in \mathbb{R}^{N \times N}$ | Self-expression matrix |

**Algorithm 1** (Low-Rank and Sparse Tensor Decomposition with Ridge Regularized Subspace Clustering)

**Data:** $\mathcal{X}$, $N$, $P_2$, $P_3$
**Result:** $\mathcal{A}$, $\mathcal{C}$, $\boldsymbol{U}_2$, $\boldsymbol{U}_3$, $\boldsymbol{Z}$
Initialize $\mathcal{A} = \boldsymbol{0}$, $\boldsymbol{Z} = \boldsymbol{I}$. Initialize $\mathcal{C}$, $\boldsymbol{U}_2$, $\boldsymbol{U}_3$ by applying Tucker decomposition on $\mathcal{X}$;
**while** *not converged* **do**
  Update $\boldsymbol{Z}$ as in Equation (6);
  Update $\mathcal{A}$ as in Equation (8);
  Update $\mathcal{C}$ as in Equation (10);
  Update $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ as in Equation (12);
**end**

### 3.4.1. Calculating the Self-Expression Matrix.

The subproblem for optimizing $\boldsymbol{Z}$ when all other parameters are fixed is

$$\min_{\boldsymbol{Z}} \frac{1}{2}\|\boldsymbol{Z}\|_F^2 + \frac{\lambda_z}{2}\|\boldsymbol{C}_{(1)} - \boldsymbol{Z}\boldsymbol{C}_{(1)}\|_F^2. \tag{5}$$

**Proposition 3.1.** *The optimization problem in Equation (5) admits a closed-form solution,*

$$\boldsymbol{Z}^* = \lambda_z(\boldsymbol{I} + \lambda_z \boldsymbol{C}_{(1)}\boldsymbol{C}_{(1)}^\top)^{-1}(\boldsymbol{C}_{(1)}\boldsymbol{C}_{(1)}^\top). \tag{6}$$

Equation (5) suggests that there are only two terms in the problem related to the self-expression matrix $\boldsymbol{Z}$. Proposition 3.1 derived a closed-form solution.

### 3.4.2. Calculating the Sparse Anomaly Tensor.

The subproblem for optimizing $\mathcal{A}$ when all other parameters are fixed is

$$\min_{\mathcal{A}} \lambda_a \|\mathcal{A}\|_1 + \frac{\lambda_e}{2}\|\mathcal{X} - \mathcal{A} - \mathcal{C} \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3\|_F^2. \tag{7}$$

**Proposition 3.2.** *The optimization problem in Equation (7) admits a closed-form solution using the soft-thresholding operator,*

$$\mathcal{A}^* = \operatorname{sgn}(\mathcal{X} - \mathcal{C} \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3) \odot \max\left(\boldsymbol{0}, |\mathcal{X} - \mathcal{C} \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3| - \frac{\lambda_a}{\lambda_e}\right). \tag{8}$$

The proof follows directly on using soft thresholding in solving the orthogonal LASSO problem and is therefore omitted here.

### 3.4.3. Calculating the Core Tensor.

The subproblem for optimizing $\mathcal{C}$ is

$$\min_{\mathcal{C}} \frac{\lambda_z}{2}\|\mathcal{C} - \mathcal{C} \times_1 \boldsymbol{Z}\|_F^2 + \frac{\lambda_e}{2}\|\mathcal{X} - \mathcal{A} - \mathcal{C} \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3\|_F^2. \tag{9}$$

**Proposition 3.3.** *Define* $\boldsymbol{M} \triangleq \boldsymbol{X}_{(1)} - \boldsymbol{A}_{(1)}$ *and* $\boldsymbol{P} \triangleq (\boldsymbol{U}_3 \otimes \boldsymbol{U}_2)^\top$. *The optimization problem in Equation* (9) *admits a closed-form solution:*

$$\boldsymbol{C}^*_{(1)} = \lambda_e(\lambda_z(\boldsymbol{I}-\boldsymbol{Z})^\top(\boldsymbol{I}-\boldsymbol{Z}) + \lambda_e\boldsymbol{I})^{-1}\boldsymbol{M}\boldsymbol{P}^\top. \tag{10}$$

*The resulting matrix* $\boldsymbol{C}^*_{(1)}$ *is folded back to the original tensor* $\mathcal{C}$ *after the calculation.*

**3.4.4. Calculating the Orthonormal Bases.** When all other parameters are fixed, the subproblem to update $\boldsymbol{U}_2$ is

$$\begin{aligned} \min_{\boldsymbol{U}_2} \quad & \frac{\lambda_e}{2}\|\mathcal{X} - \mathcal{A} - \mathcal{C}\times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3\|_F^2 \\ & \boldsymbol{U}_2^\top \boldsymbol{U}_2 = \boldsymbol{I}. \end{aligned} \tag{11}$$

**Proposition 3.4.** *Define* $\boldsymbol{Q} \triangleq (\boldsymbol{X}_{(2)} - \boldsymbol{A}_{(2)})(\boldsymbol{U}_3 \otimes \boldsymbol{I})\boldsymbol{C}^\top_{(2)}$ *and let* $\boldsymbol{Q} = \hat{\boldsymbol{U}}\boldsymbol{D}\hat{\boldsymbol{V}}^\top$ *be the singular value decomposition of* $\boldsymbol{Q}$. *Thus, the optimization problem in Equation* (11) *admits a closed-form solution for the optimal values for the first orthogonal basis* $\boldsymbol{U}^*_2$*:*

$$\boldsymbol{U}^*_2 = \hat{\boldsymbol{U}}\hat{\boldsymbol{V}}^\top. \tag{12}$$

Finally, the update of $\boldsymbol{U}^*_3$ is similar and therefore omitted. In conclusion, the updates of $\boldsymbol{U}^*_2$ and $\boldsymbol{U}^*_3$ both yield closed-form solutions.

## 4. Simulation Study and Results

In this section, we will evaluate the proposed methods via a simulation study. We will first introduce the simulation setup in Section 4.1. Then, the proposed method will be evaluated and compared with several clustering-based benchmark methods in Section 4.3.

### 4.1. Simulation Setup

In this section, we formalize the simulation procedure we use to generate clustered three-order tensors with dimensionality $N \times I_2 \times I_3$. First, the procedure calls for the parameters in Table 2. After defining these parameters and setting the random number-generating seed, we follow the following procedure to generate the data:

- For each cluster $k$ and mode of tensor $l = 2, 3$, generate random factor matrices $\boldsymbol{U}_{k,l} \in \mathbb{R}^{I_l \times P_l}$ from standard normal distribution, and then orthogonalize $\boldsymbol{U}_{k,l}$ via the QR decomposition.
- For each cluster $k$, generate a random core tensor $\mathcal{C}_k \in \mathbb{R}^{N \times P_2 \times P_3}$ from standard normal distributions.
- For each cluster $k$, generate $\mathcal{X}_k = \mathcal{C}_k \times_2 \boldsymbol{U}_{k1} \times_3 \boldsymbol{U}_{k2} + \epsilon$, where noise is added on each element of $\mathcal{X}$ as $\epsilon \sim \mathcal{N}(0, \sigma^2)$, with $\sigma^2 = 0.25$.
- Elementwise add sparse anomaly $\mathcal{X} \leftarrow \mathcal{X} + \psi\mathcal{A}_p$. Each entry of $\mathcal{A}_p$ is generated from the Bernoulli distribution with probability. Here $p$ is the anomaly ratio. In each entry where $\mathcal{A}_p$ is one, indicating anomaly is present, we add the anomaly intensity constant $\psi$ to the data.

Figure 3 provides a visual representation of the simulation data generation process. This procedure allows us to control key elements of the data-generating process, such as the number of clusters, the cluster dimensions, the prevalence of anomalies, and how gross the anomalies are. We will now experiment with these factors and their effect on how well our model clusters samples and/or detects anomalies.

### 4.2. Tuning Parameter Selection

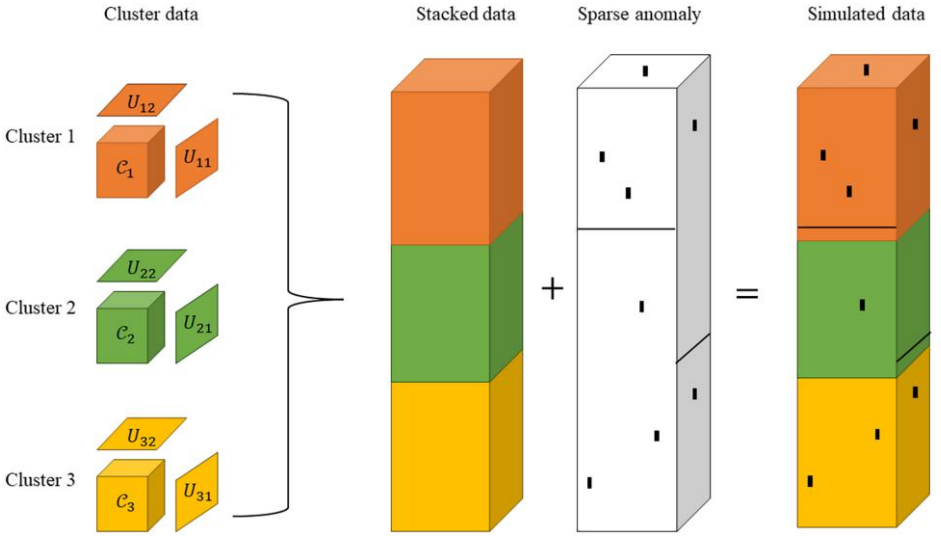
The proposed LRTSD has three tuning parameters: $\lambda_e$ penalizes the norm of the self-regression matrix, $\lambda_z$ penalizes the self-regression error of the core tensor, and $\lambda_a$ penalizes the norm of the anomaly tensor. These three tuning parameters are in comparison with the reconstruction error term, which has a coefficient of one.

Related to the anomaly tuning parameter $\lambda_a$, one can select based on the percentage of nonzero anomaly entries expected in the data set. For example, we can select the anomaly entries as a fixed percentage if we have some rough ideas of the percentage of the anomaly entries. If such a percentage is unknown, Otsu's method can be used to automatically select the tuning parameter $\lambda_a$, which aims to search for $\lambda_a$ that minimizes the intraclass variance, defined as a weighted sum of variances of the two classes: normal class and abnormal class. This procedure is discussed in detail in Yan et al. (2017). For $\lambda_e$ and $\lambda_z$, $\lambda_e$ yields smaller diagonal values in the self-regression matrix $\boldsymbol{Z}$; higher $\lambda_z$ yields a more precise self-regression. To address the challenge of unknown clustering accuracy, we propose using the normalized cut (NC) score to evaluate the affinity matrix produced by the algorithm (Zhang et al. 2023) and select the tuning parameters $\lambda_e$ and $\lambda_z$ that

**Table 2.** Required Parameters for Simulated Clustered Data Generation and Their Notation

| Notation | Explanation |
|---|---|
| $K \in \mathbb{Z}^+$ | Number of clusters |
| $N \in \mathbb{Z}^+$ | Number of samples per cluster |
| $D_l \in \mathbb{Z}^+$ | Ambient dimension of $l$th mode |
| $P_l \in \mathbb{Z}^+$ | Intrinsic dimension of $l$th mode |
| $\sigma \in \mathbb{R}^+$ | Standard deviation of random noise |
| $\mathcal{A}_p \in \mathbb{B}^{D_1 \times \cdots \times D_L}$ | Sparse anomaly index tensor with $0 < p < 1$ of entries being 1 and others 0 |
| $\psi \in \mathbb{R}$ | Sparse anomaly intensity |

**Figure 3.** (Color online) Illustration Depicting the Generation Process of Simulated Data




produce the most reasonable affinity matrix by minimizing the NC score. This measurement is defined by $NC = \sum_{c=1}^{C} W_c^{out}/(W_c^{in} + W_c^{out})$, where $c$ is the index of the cluster and $W_c^{in}$ and $W_c^{out}$ are the sum of the weights of the graph within the cluster and the weights of the graph outside the cluster. Smaller NC score means better clustering.

Some examples of clustering accuracy for different combinations of tuning parameters are shown in Figure 4. From Figure 4, we can see the following: (1) $\lambda_e$, $\lambda_z$, $\lambda_a$ cannot be too small, because we will almost lose a penalty term when any of the parameters are too small, which affects the clustering accuracy. (2) As $\lambda_e$ and $\lambda_z$ increase, the clustering precision increases. When $\lambda_e$ continues to increase, the clustering accuracy will decrease because the corresponding terms will dominate optimization. (3) $\lambda_a$ cannot be too small or too big, because we will not penalize the anomaly term when it is too small, and it will be too hard to discover any anomaly when $\lambda_a$ is too big.

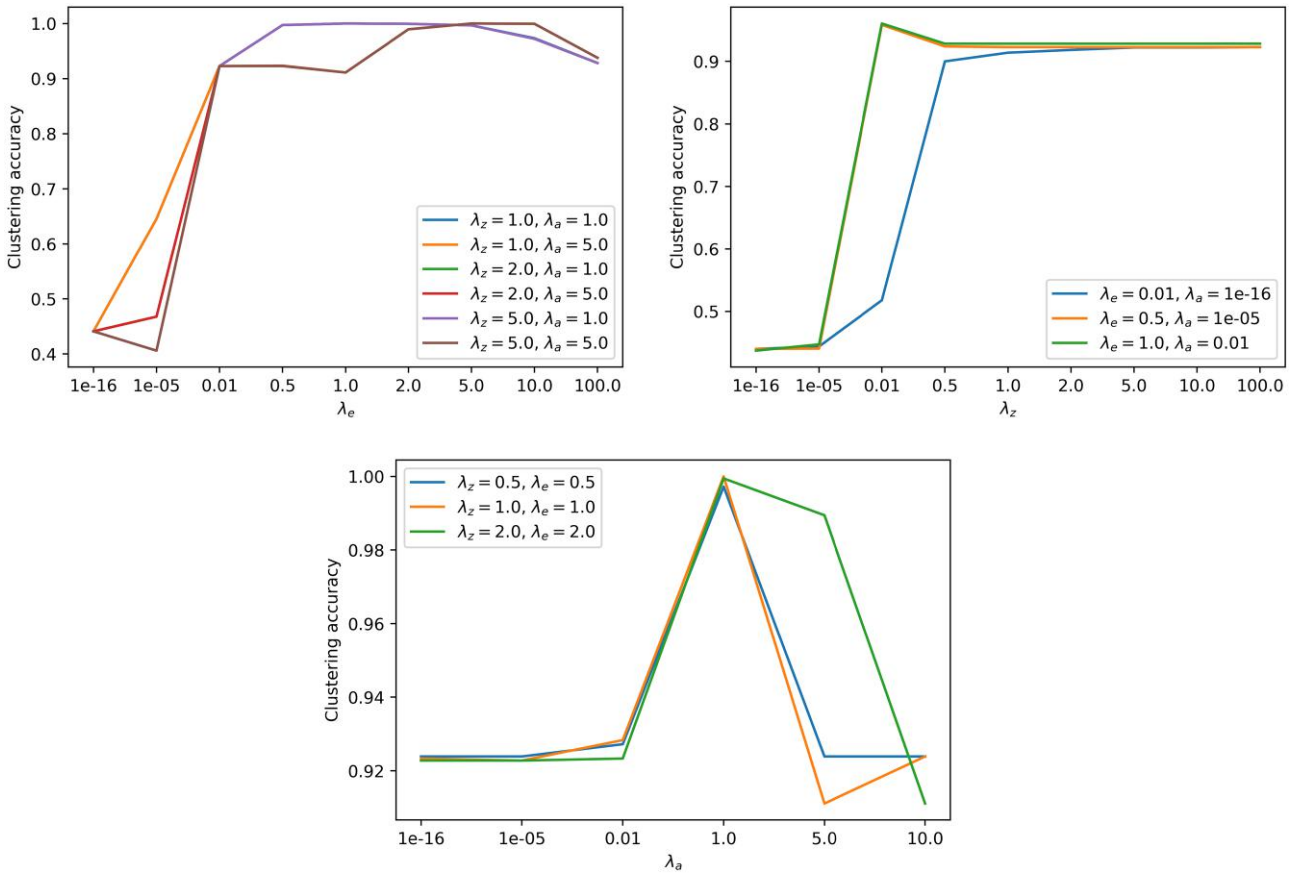
### 4.3. Performance Evaluation and Comparison

We generate a simulated three-dimensional tensor with 3 groups, 30 samples per cluster, 50 as the ambient dimensionality of each mode, and 5 as the intrinsic dimension on each mode. Therefore, the dimensionality of the data tensor is $90 \times 50 \times 50$. This can be considered representative of a spatiotemporal observation with one spatial dimension and two temporal dimensions with the inherent clustered structure and redundant information with respect to dimensionality. In this section, we compare the clustering accuracy of our models and three benchmark models under different anomaly ratios, and anomaly intensity, where anomaly ratios are defined as the ratio of anomaly entry compared with the entire tensor dimensions and anomaly intensity denotes $\psi$. The clustering accuracy is defined by the maximum proportion of matching labels of the 90 objects in the first dimension over all permutations of the labels. The benchmark methods we compared are introduced below:

- **K-means:** For each of the 90 objects, we reshape the two-dimensional $50 \times 50$ matrix to a vector of length 2,500. We make three groups using K-means based on the Euclidean distance.
- **Robust Tucker + K-means:** We first use the Robust Tucker decomposition introduced in Heng et al. (2023) and Xue et al. (2017) to reduce the dimension to $15 \times 15$ for each object. We then use K-means to generate three clusters based on the vectorized core tensor.
- **Multimode tensor space clustering** (He and Atia 2022)**:** This is a method based on low-rank tensor representation (LTRR). We will use single-mode tensor space clustering for two reasons. Firstly, all the other methods use a single mode, and thus it makes for a fair comparison. Secondly, when we increased the number of modes, the running time increased dramatically.
- **Robust subspace clustering** (Peng et al. 2015)**:** This method is based on the TRR and is compatible with the matrix data. Therefore, we first reshaped the $90 \times 50 \times 50$ tensor into the $90 \times 2{,}500$ matrix and made three clusters.

When comparing the performance of LRTSD and the benchmark methods, we use a grid search of the tuning parameters and report the best clustering precision in different settings, namely, $\lambda_e$, $\lambda_c$, $\lambda_a$ in LRTSD, representation rank $r$ in LTRR, and balance and thresholding parameters $\lambda$ and $k$ in TRR. We report the mean accuracy of 20 replications of data generation and clustering

**Figure 4.** (Color online) Clustering Accuracy of Different Tuning Parameter Combinations




*Notes.* (Left) Behavior of $\lambda_e$ with different combination of $\lambda_z$ and $\lambda_a$. (Right) Behavior of $\lambda_z$ with different combination of $\lambda_e$ and $\lambda_a$. (Bottom) Behavior of $\lambda_a$ with different combination of $\lambda_z$ and $\lambda_e$.

accuracy for each anomaly ratio and anomaly intensity combination and each method. The clustering accuracy results are shown in Table 3 with their standard deviation over the 20 replications in parentheses. In each combination of anomaly intensity and anomaly ratio, the best clustering accuracy is highlighted in bold. From the clustering accuracy result, we can see that the K-means and Robust Tucker + K-means methods yield the two lowest accuracy proportions. This is because they are not designed specifically for clustering and suffer from the curse of dimensionality when vectorized. When the anomaly ratio and intensity are not large, LRTSD, TRR, and LTRR have almost perfect clustering accuracy. As the intensity and ratio of the anomalies increase, the LTRR and TRR start to show errors. The threshold at which the two methods start to have incorrect clustering labels is smaller than that of LRTSD. Meanwhile, although LRTSD achieves the highest clustering accuracy, it does not require much more time compared with benchmark methods. The average processing times for K-means, Robust Tucker + K-means, LTRR, TRR, and LRTSD are 0.06 second, 0.68 second, 0.08 second, 0.51 second, and 0.71 second, respectively.

## 4.4. Sensitivity Analysis

More specifically, we aim to answer two major questions by comparing the proposed methods with several benchmark methods in terms of anomaly detection accuracy and clustering accuracy.

- Question 1: If sparse and gross anomalies are present, does anomaly detection lead to better clustering?
- Question 2: Does the tensor decomposition and dimension reduction lead to better clustering?

**4.4.1. The Effect of Dimensionality Reduction on Clustering Accuracy.** We generate a simulated three-dimensional tensor with 3 clusters, 30 samples per cluster, and 125 as the data dimensionality on each mode, namely, $90 \times 125 \times 125$, where the intrinsic dimensionality is 5 in each cluster. Therefore, the total intrinsic dimensionality of the tensor is actually $5 \times 3 = 15$. This can be considered representative of a spatiotemporal observation with one spatial dimension and two temporal dimensions with the inherent clustered structure and redundant information with respect to dimensionality.

For each seed, we generate some data and apply dimensionality reduction by a certain factor on each

**Table 3.** Clustering Accuracy for LRTSD and Other Three Benchmarks

| | Ratio | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Intensity | Method | 0.05 | 0.1 | 0.125 | 0.15 | 0.175 | 0.2 | 0.225 |
| 4 | K-means | 0.49 | 0.45 | 0.45 | 0.47 | 0.47 | 0.45 | 0.45 |
| | | (0.02) | (0.03) | (0.03) | (0.02) | (0.02) | (0.02) | (0.03) |
| | Robust Tucker + K-means | 0.5 | 0.5 | 0.49 | 0.45 | 0.48 | 0.47 | 0.45 |
| | | (0.03) | (0.03) | (0.03) | (0.02) | (0.02) | (0.02) | (0.02) |
| | LTRR | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| | | (0) | (0) | (0) | (0) | (0) | (0) | (0) |
| | TRR | **1** | **1** | **1** | 0.99 | 0.98 | 0.97 | 0.97 |
| | | (0) | (0) | (0) | (0.01) | (0.01) | (0.01) | (0.01) |
| | LRTSD | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| | | (0) | (0) | (0) | (0) | (0) | (0) | (0) |
| 5 | K-means | 0.47 | 0.44 | 0.46 | 0.44 | 0.46 | 0.44 | 0.44 |
| | | (0.02) | (0.02) | (0.03) | (0.03) | (0.03) | (0.03) | (0.03) |
| | Robust Tucker + K-means | 0.47 | 0.47 | 0.48 | 0.45 | 0.43 | 0.43 | 0.43 |
| | | (0.03) | (0.03) | (0.02) | (0.03) | (0.03) | (0.03) | (0.02) |
| | LTRR | **1** | **1** | **1** | **1** | 0.99 | 0.98 | 0.97 |
| | | (0) | (0) | (0) | (0) | (0.01) | (0.01) | (0.02) |
| | TRR | 0.99 | 0.99 | 0.98 | 0.96 | 0.99 | 0.96 | 0.98 |
| | | (0.01) | (0.01) | (0.01) | (0.02) | (0.01) | (0.02) | (0.01) |
| | LRTSD | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| | | (0) | (0) | (0) | (0) | (0) | (0) | (0) |
| 6 | K-means | 0.44 | 0.44 | 0.43 | 0.44 | 0.43 | 0.42 | 0.42 |
| | | (0.02) | (0.02) | (0.03) | (0.03) | (0.03) | (0.04) | (0.03) |
| | Robust Tucker + K-means | 0.46 | 0.46 | 0.45 | 0.43 | 0.41 | 0.41 | 0.4 |
| | | (0.03) | (0.02) | (0.04) | (0.02) | (0.02) | (0.03) | (0.02) |
| | LTRR | **1** | **1** | 0.99 | 0.99 | 0.96 | 0.94 | 0.87 |
| | | (0) | (0) | (0.01) | (0.01) | (0.02) | (0.02) | (0.05) |
| | TRR | 0.99 | 0.96 | 0.97 | 0.96 | 0.9 | 0.87 | 0.81 |
| | | (0.01) | (0.02) | (0.02) | (0.02) | (0.04) | (0.05) | (0.05) |
| | LRTSD | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| | | (0) | (0) | (0) | (0) | (0) | (0) | (0) |
| 7 | K-means | 0.44 | 0.41 | 0.4 | 0.41 | 0.4 | 0.39 | 0.41 |
| | | (0.02) | (0.03) | (0.03) | (0.03) | (0.04) | (0.03) | (0.03) |
| | Robust Tucker + K-means | 0.45 | 0.43 | 0.43 | 0.4 | 0.39 | 0.4 | 0.39 |
| | | (0.04) | (0.03) | (0.03) | (0.04) | (0.02) | (0.03) | (0.02) |
| | LTRR | **1** | 0.99 | 0.98 | 0.96 | 0.83 | 0.74 | 0.64 |
| | | (0) | (0.01) | (0.01) | (0.02) | (0.07) | (0.09) | (0.12) |
| | TRR | 0.97 | 0.94 | 0.9 | 0.79 | 0.76 | 0.64 | 0.63 |
| | | (0.01) | (0.02) | (0.04) | (0.1) | (0.09) | (0.11) | (0.08) |
| | LRTSD | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| | | (0) | (0) | (0) | (0) | (0) | (0) | (0) |
| 8 | K-means | 0.44 | 0.39 | 0.39 | 0.4 | 0.39 | 0.41 | 0.4 |
| | | (0.02) | (0.02) | (0.03) | (0.02) | (0.03) | (0.03) | (0.03) |
| | Robust Tucker + K-means | 0.42 | 0.41 | 0.41 | 0.4 | 0.4 | 0.39 | 0.39 |
| | | (0.03) | (0.04) | (0.03) | (0.03) | (0.02) | (0.02) | (0.03) |
| | LTRR | **1** | 0.94 | 0.86 | 0.74 | 0.58 | 0.54 | 0.53 |
| | | (0) | (0.01) | (0.04) | (0.07) | (0.09) | (0.08) | (0.08) |
| | TRR | 0.98 | 0.86 | 0.82 | 0.78 | 0.74 | 0.51 | 0.47 |
| | | (0.01) | (0.05) | (0.08) | (0.11) | (0.09) | (0.06) | (0.07) |
| | LRTSD | **1** | **1** | **1** | **1** | **1** | **1** | **0.99** |
| | | (0) | (0) | (0) | (0) | (0) | (0) | (0.01) |

*Notes.* Best accuracy is highlighted in boldface. Standard deviation in parenthesis.

mode. The dimension reduction factor $\rho_d = [(\prod_l I_l / P_l)^{1/l}]$ is defined as the geometric mean of the data dimensionality $I_l$ over the intrinsic dimension $P_l$ averaged on different mode $l$ of the tensor. For one factor of the experiment, we do not apply any dimensionality reduction (which is denoted by a factor of one on the $x$-axis in Figure 5). In other words, we directly apply subspace clustering on the high-dimensional tensor, optimizing the following objective:

$$\min_{\mathbf{Z}} \quad \lambda \|\mathbf{Z}\|_F^2 + \|\mathbf{X}_1 - \mathbf{X}_1 \mathbf{Z}\|_F^2.$$

Figure 5 shows the result of difference from minimum accuracy. For each replication, we obtain the clustering

accuracy for different dimensionality reduction factors. Then they are subtracted from the minimum clustering accuracy within that replication, and we get the difference for each dimensionality reduction factor. After 20 replications, we get the result of 20 accuracy differences for each dimensionality reduction factor. Their mean is plotted in Figure 5. Observing Figure 5, there is an optimal range of dimensionality reduction ratio around 8 to 10, which roughly corresponds to the true intrinsic dimension ratio, that is, 125/15 = 8.33. This clearly suggests that clustering is more effective at the right proportion of dimensionality reduction, supporting our claim that dimension reduction improves the tensor clustering performance.

#### 4.4.2. The Effect of Anomaly Detection on Clustering Performance.

The parameters for generating the simulation data are set to be the same as those in Section 4.3.

We factorize our experiment over the properties of the added sparse anomaly. Specifically, we alter how prevalent the anomaly is and how large its intensity is. At each factor of the experiment, we define the control group as the absence of anomaly detection, in other words, removing the term $\frac{\lambda_a}{2}\|\mathcal{A}\|_1$ from Equation (2). The treatment group is Equation (2) as is; therefore, the algorithm tries to detect the sparse anomaly. We do this over multiple seeds, where in each seed the exact same observations are used for both the control and treatment groups. We report the difference in clustering accuracy between the treatment group and the control group to identify whether anomaly detection would increase clustering accuracy or not. The clustering accuracy is defined as almost the same as the well-known clustering accuracy, except that the permutation of the labels is allowed such that it maximizes the agreement between the predicted labels and the ground truth.

**Figure 5.** (Color online) Evolution of Clustering Accuracy over Multiple Replications by Increasing Dimensionality Factor on Each Mode

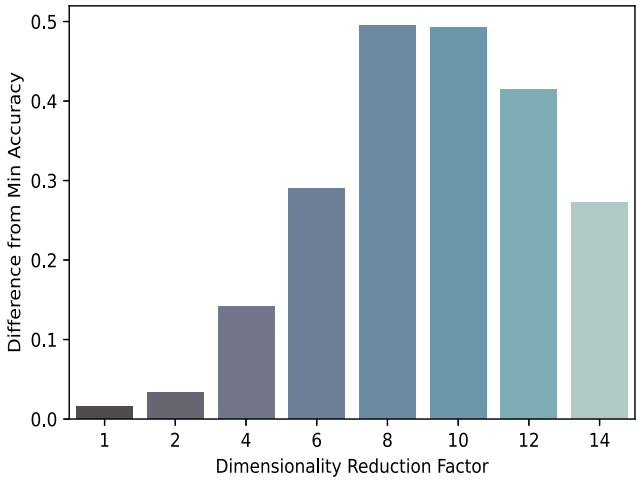
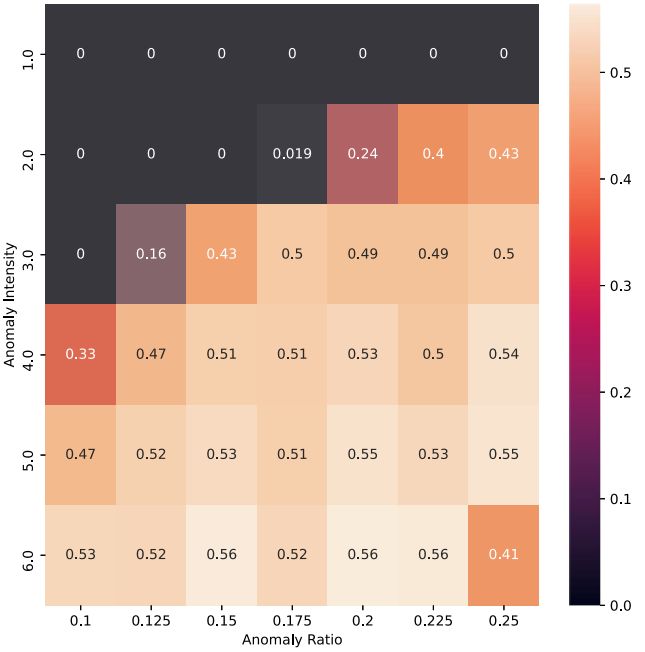



**Figure 6.** (Color online) Average of Pairwise Differences in Clustering Accuracy




*Note.* The figure shows the average of pairwise differences in clustering accuracy between when anomaly detections are allowed and when they are not, across various anomaly conditions and 10 data-generating seeds.

In Figure 6, we observe a clear relationship between the severity of the anomalies and how much anomaly detection improves clustering performance. As the intensity of the anomaly increases and/or the prevalence of anomalies increases, the application of anomaly detection becomes even more significant in terms of finding the right groupings between the samples of the data. This can be considered quite an expected finding, but it is an important sanity check of our optimization procedure, and it finalizes the discussion of why

**Figure 7.** (Color online) Eigenvalue Spectrum of the Affinity Matrix to Locate Discontinuous Segments

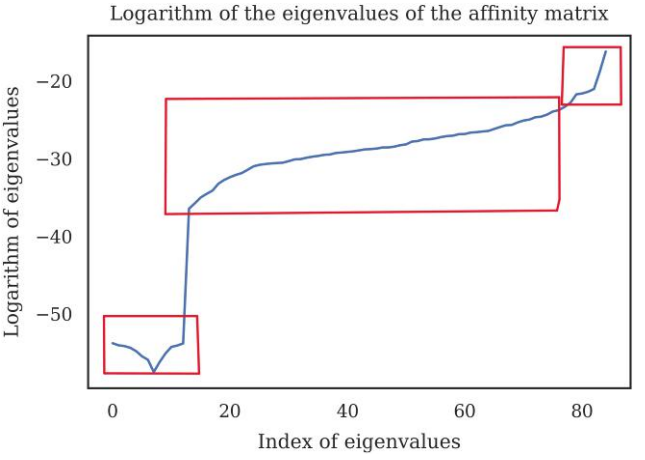
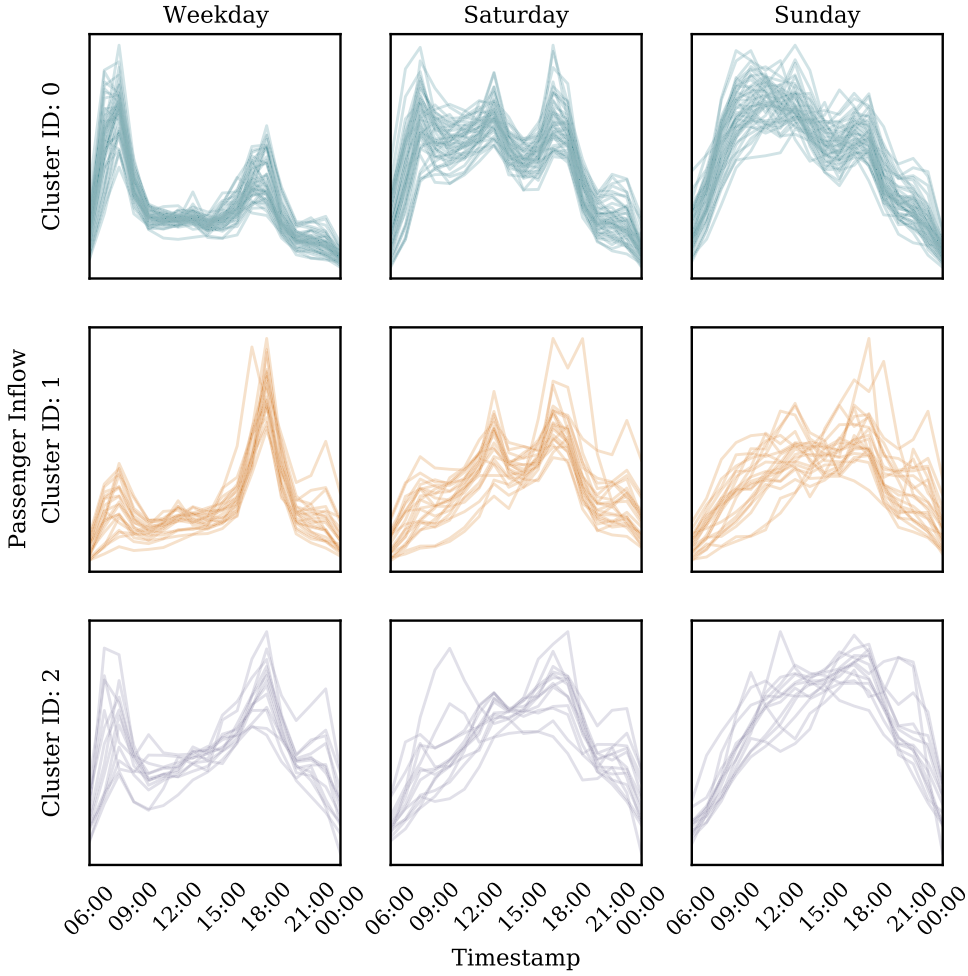


**Figure 8.** (Color online) Hourly Inflow Patterns per Cluster on a Randomly Picked Week, Across Weekdays and Weekend Days




*Note.* The color coding matches with the stations in Figure 9.

simultaneous anomaly detection is crucial for the quality of cluster discovery.

## 5. Case Study and Results

We apply our proposed methodology to the smart card data collected by MTR, the company that operates Hong Kong's subway system. Smart card data have emerged as an important ingredient for urban mobility analysis, as they can be collected cheaply and contain detailed information about how people move within a city (Pelletier et al. 2011). Using this information, decisions at the tactical or strategic level can be made in a data-driven, thus more effective, manner.

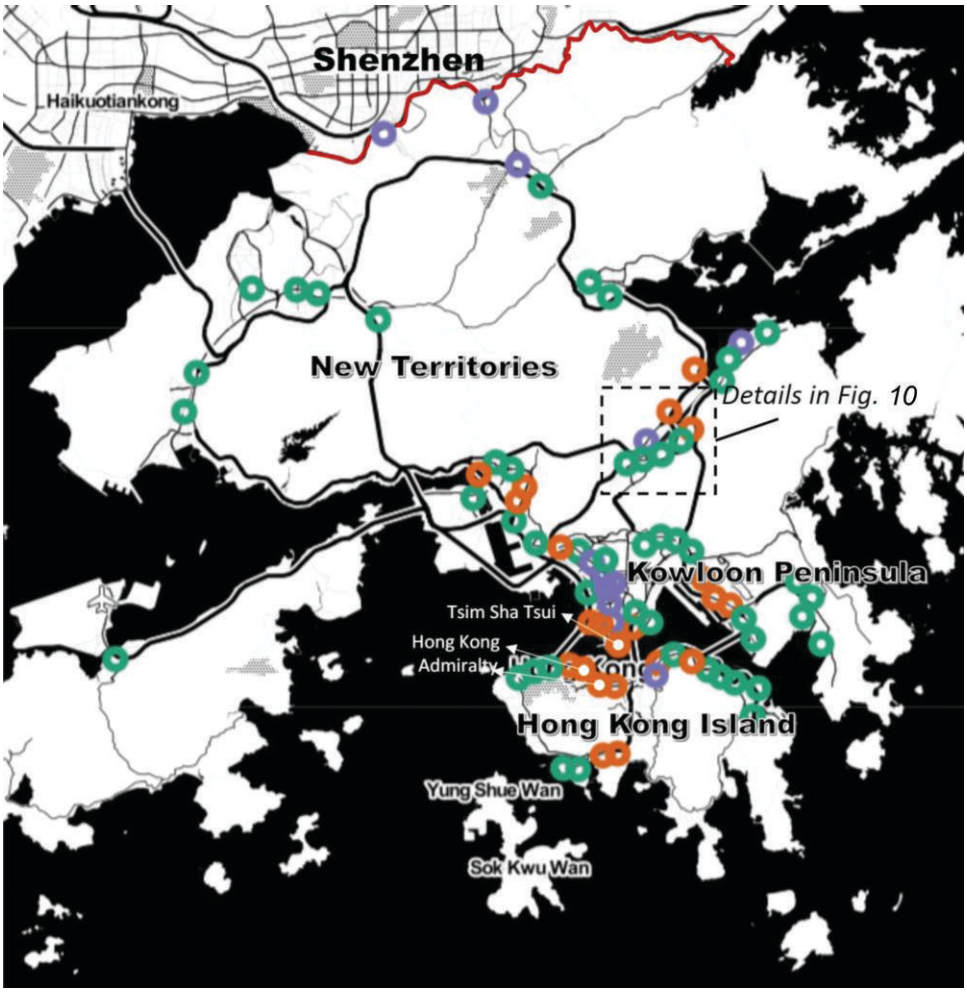
The data span an 82-day time horizon starting from January 1, 2017, and record every tap-in and tap-out event from 85 stations of the system. We resample these events hourly for each day within the operating hours of the system. Thus, we end up with the number of people entering a station every operating hour of every station.

Such data can be represented as a three-dimensional tensor. The first mode of the tensor represents the stations, whereas the second and third modes represent the day and the time of day, respectively. Thus, we expect to obtain clusters at the station level using the clustering objective and point anomalies at hourly resolution.

Understanding commonalities between stations is important from an urban mobility perspective. The stations that are similar to each other in terms of passenger inflow may reveal what kinds of features of those stations are the most influential in people's decision to take the subway on a specific day and time of day.

Detecting anomalies is also important from a managerial point of view. Generally, urban mobility shows strong predictability because of the regular weekly schedules of city dwellers. However, rare events could occur, and the subway management would rather take precautions in the face of such events. Detection of past point anomalies may reveal the underlying reasons for one-off events that lead to extreme passenger inflow, such as concerts, celebrations, or extreme weather events. Armed with this knowledge, subway management can be more accurate in predicting future rare events.

**Figure 9.** Hong Kong Subway Stations on a Map, Clustered Three Ways




*Notes.* Colors represent different clusters. Circle radius represents a typical walking distance of 400 meters around the station.

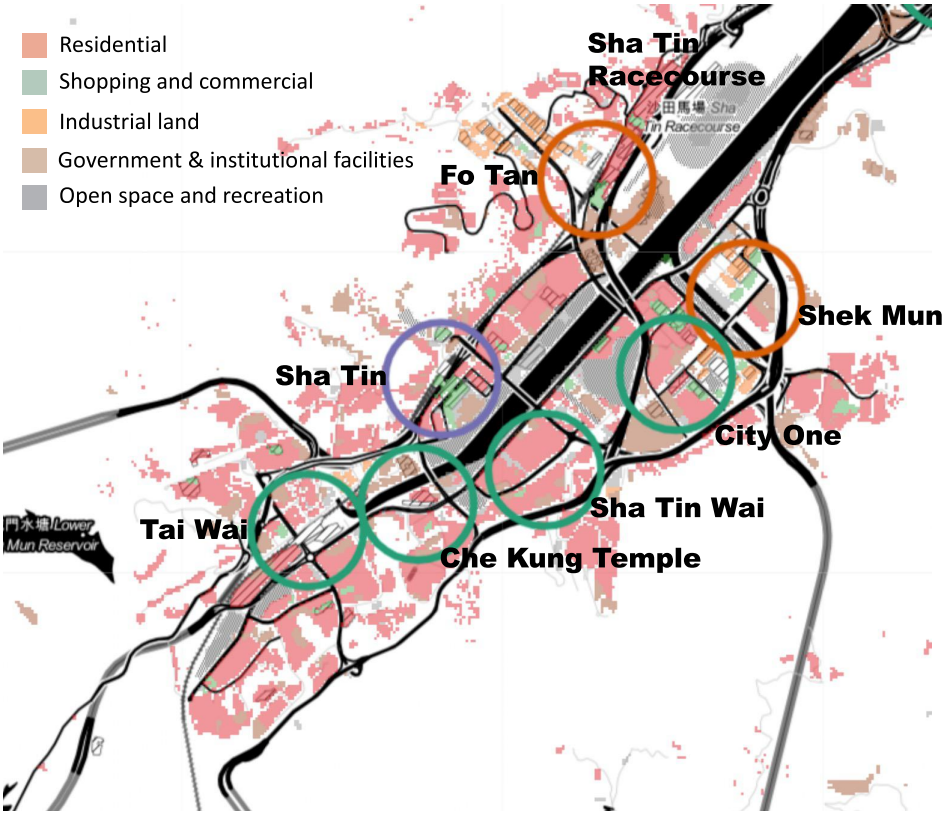
### 5.1. Station Clustering

We trained our model with the smart card data and recovered three clusters. The number of clusters was determined using the method introduced in von Luxburg (2007). We first plot the logarithm of the eigenvalue spectrum of the Laplacian of the affinity matrix and then locate the discontinuous segments, as shown in Figure 7.

Figures 8 and 9, when observed jointly, reveal important insight into how the stations are clustered. Weekdays play the primary role in determining the cluster of a station. As shown in Figure 8, the first cluster (Cluster ID: 0, colored in green) has a strong inflow peak at the morning peak hour. These stations are also mainly located in the residential areas of Hong Kong (i.e., west and east ends of Hong Kong Island, west side of New Territories, and east side of Kowloon), as shown in Figure 9, Likewise, for the second cluster (Cluster ID: 1, colored in orange), we observe a strong evening peak inflow as opposed to the mornings, and these stations are mostly located in the business regions such as the central area of Hong Kong Island and the opposite side in Kowloon, as shown in Figure 9. As we could conclude so far, for the most part, this can be explained by the land use, specifically, whether it is residential or work and office space. The third cluster (Cluster ID: 2, colored in purple) does not have a dominant peak between the morning and evening, and the rest of the inflow is rather evenly distributed throughout the day. Although not as prominent as weekdays, weekend (especially Saturday) patterns also differ from one cluster to another.

Interpretation of the third cluster requires some contextual investigation. Hong Kong draws millions of visitors each year, especially from the bordering Shenzhen City, Guangdong Province, of mainland China. The third cluster is a collection of stations where visitors enter Hong Kong, with the primary intention being cross-border shopping. The absence of taxes and duties on goods in Hong Kong makes it an appealing

**Figure 10.** Sha Tin and Tai Wai Regions with Land Use Information




*Notes.* The circle color coding of clusters matches with the stations in Figure 9. The land use color coding is as follows: red, residential; green, commercial; orange, industrial; brown, institutional or governmental; gray, parks and recreational areas.

destination for Chinese visitors to shop. Because shopping is not a strictly time-bound activity, we observe inflows that are dispersed more homogeneously around the day, on both weekdays and weekends.

We now bring in the land use information[1] to the picture. In Figure 10, the first and second clusters can strictly be separated by the prevalence of residential areas or commercial and industrial areas, respectively. The only station from the third cluster, Sha Tin station, is surrounded by many shopping malls, with an iconic New Town Plaza, which offers around 2 million square feet of more than 400 exceptional shopping, dining, and lifestyle facilities. Thus, it is a popular shopping destination for passengers visiting from mainland China.

We can analyze even more thoroughly with granular information. The key takeaway is that the clustering of stations yields important insight into the factors that determine passenger inflows.

### 5.2. Detected Point Anomalies

We now focus on the point anomalies detected by the proposed model in the data from the smart card. As we have discussed before, one-off events are an important determinant of irregular patterns in passenger inflows.

Lunar New Year celebrations in Hong Kong, which took place between the 28th and 31st of January in Victoria Harbor between Hong Kong Island and the Kowloon Peninsula, yield a number of other rare events. One of the events is the Lunar New Year fireworks display. In Figure 11, we observe the peaks related to that event in the Tsim Sha Tsui, Admiralty, and Hong Kong stations, with locations pinned in Figure 9. These stations are close to places that are considered the best places to watch the fireworks display.

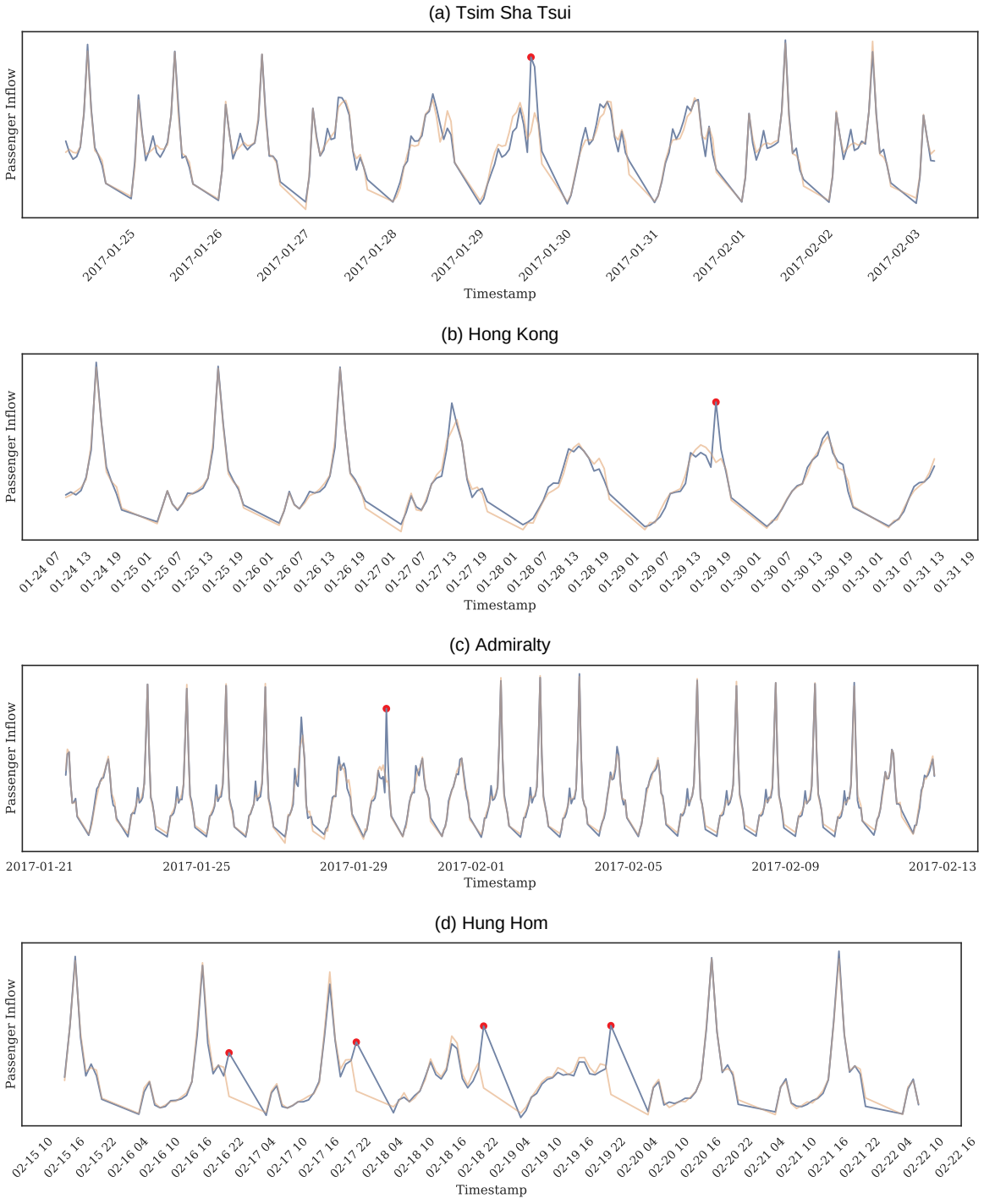
Another interesting rare event takes place at the Hung Hom station. In Figure 11, we observed an unusual peak at 8 p.m. for four consecutive days from February 16 to February 19. This suggests the scheduled end of a four-day event, perhaps a conference, given the proximity of the station to Hong Kong Polytechnic University.

In general, detected anomalies are promising in terms of the amount of insight they can provide for public transportation management.

## 6. Conclusion

In this paper, we focused on the problem of spatiotemporal modeling with tensors. Specifically, we proposed a tensor decomposition-based model that is capable of simultaneous clustering and point anomaly detection,

**Figure 11.** (Color online) Examples of Rare Events in Four Different Stations, Detected as Point Anomalies over Passenger Inflows




all of which are important challenges to be addressed in this kind of data. Our simulation study shows that, in general, point anomaly detection increases clustering accuracy when point anomalies exist in data and vice versa. As for the case study, we trained our model on a collection of smart card transactions from the subway system of Hong Kong. Our model was able to effectively recover station clusters that are semantically consistent both temporally and spatially. The model was also able to recover rare events in terms of point anomalies, which offers valuable insights and assistance for managerial analysis and operations.

## Endnote

[1] Open data via https://www.pland.gov.hk/pland_en/info_serv/open_data/landu/index.html.

## References

Aggarwal CC (2015) *Outlier Analysis* (Springer, Berlin, Heidelberg), 237–263.

Bahadori MT, Yu QR, Liu Y (2014) Fast multivariate spatio-temporal analysis via low rank tensor learning. Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger KQ, eds. *Advances in Neural Information Processing Systems*, vol. 27 (Curran Associates Inc., Red Hook, NY), 3491–3499.

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: A survey. *ACM Comput. Surveys* 41(3):1–58.

Drakopoulos G, Spyrou E, Mylonas P (2019) Tensor clustering: A review. *2019 14th Internat. Workshop Semantic Social Media Adaptation Personalization (SMAP)* (IEEE, Piscataway, NJ), 1–6.

Du B, Zhang L (2014) A discriminative metric learning based anomaly detection method. *IEEE Trans. Geosci. Remote Sensing* 52(11): 6844–6857.

Fanaee-T H, Gama J (2016) Tensor-based anomaly detection: An interdisciplinary survey. *Knowledge Based Systems* 98:130–147.

Fu Y, Gao J, Tien D, Lin Z (2014) Tensor LRR based subspace clustering. *2014 Internat. Joint Conf. Neural Networks (IJCNN)* (IEEE, Piscataway, NJ), 1877–1884.

Fu Y, Gao J, Tien D, Lin Z, Hong X (2016) Tensor LRR and sparse coding-based subspace clustering. *IEEE Trans. Neural Networks Learn. Systems* 27(10):2120–2133.

Gahrooei MR, Yan H, Paynabar K, Shi J (2021) Multiple tensor-on-tensor regression: An approach for modeling processes with heterogeneous sources of data. *Technometrics* 63(2):147–159.

Gao H, Nie F, Li X, Huang H (2015) Multi-view subspace clustering. *Proc. IEEE Internat. Conf. Comput. Vision* (IEEE, Piscataway, NJ), 4238–4246.

Guo K, Zhang T, Xu X, Xing X (2019) Low-rank tensor thresholding ridge regression. *IEEE Access* 7:153761–153772.

He Y, Atia GK (2022) Multi-mode tensor space clustering based on low-tensor-rank representation. *Proc. AAAI Conf. Artificial Intelligence* 36(6):6893–6901.

Heng Q, Chi EC, Liu Y (2023) Robust low-rank tensor decomposition with the $L_2$ criterion. *Technometrics* 65(4):537–552.

Hitchcock FL (1927) The expression of a tensor or a polyadic as a sum of products. *J. Math. Phys.* 6(1–4):164–189.

Hu Y, Work DB (2021) Robust tensor recovery with fiber outliers for traffic events. *ACM Trans. Knowledge Discovery Data* 15(1): 1–27.

Huang H, Ding C, Luo D, Li T (2008) Simultaneous tensor subspace selection and clustering: The equivalence of high order SVD and k-means clustering. Gavaldà R, Lugosi G, Zeugmann T, eds. *Proc. 14th ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (Association for Computing Machinery, New York), 327–335.

Huang S, Zhang H, Zeng H, Pižurica A (2023) From model-based optimization algorithms to deep learning models for clustering hyperspectral images. *Remote Sensing* 15(11):2832.

Jegelka S, Sra S, Banerjee A (2009) Approximation algorithms for tensor clustering. *Internat. Conf. Algorithmic Learn. Theory* (Springer, Berlin, Heidelberg), 368–383.

Kolda TG, Bader BW (2009) Tensor decompositions and applications. *SIAM Rev.* 51(3):455–500.

Li Z (2021) Tensor topic models with graphs and applications on individualized travel patterns. *2021 IEEE 37th Internat. Conf. Data Engrg. (ICDE)* (IEEE, Piscataway, NJ), 2756–2761.

Li Z, Yan H, Tsung F, Zhang K (2022a) Profile decomposition based hybrid transfer learning for cold-start data anomaly detection. *ACM Trans. Knowledge Discovery Data (TKDD)* (Association for Computing Machinery, New York).

Li Z, Yan H, Zhang C, Tsung F (2020b) Long-short term spatiotemporal tensor prediction for passenger flow profile. *IEEE Robotics Automation Lett.* 5(4):5010–5017.

Li Z, Yan H, Zhang C, Tsung F (2022b) Individualized passenger travel pattern multi-clustering based on graph regularized tensor latent dirichlet allocation. *Data Mining Knowledge Discovery* 36: 1247–1278.

Li Z, Sergin ND, Yan H, Zhang C, Tsung F (2020a) Tensor completion for weakly-dependent data on graph for metro passenger flow prediction. *Proc. AAAI Conf. Artificial Intelligence*, vol. 34 (AAAI Press, Palo Alto, CA), 4804–4810.

Lin C, Zhu Q, Guo S, Jin Z, Lin YR, Cao N (2018) Anomaly detection in spatiotemporal data via regularized non-negative tensor analysis. *Data Mining Knowledge Discovery* 32(4):1056–1073.

Liu G, Zhang Z, Liu Q, Xiong H (2019) Robust subspace clustering with compressed data. *IEEE Trans. Image Processing* 28(10): 5161–5170.

Liu G, Lin Z, Yan S, Sun J, Yu Y, Ma Y (2013) Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Machine Intelligence* 35(1):171–184.

Lock EF (2018) Tensor-on-tensor regression. *J. Comput. Graphical Statist.* 27(3):638–647.

Lu H, Plataniotis KN, Venetsanopoulos AN (2008a) MPCA: Multilinear principal component analysis of tensor objects. *IEEE Trans. Neural Networks* 19(1):18–39.

Lu H, Plataniotis KN, Venetsanopoulos AN (2008b) Uncorrelated multilinear principal component analysis through successive variance maximization. *Proc. 25th Internat. Conf. Machine Learn.* (Association for Computing Machinery, New York), 616–623.

Mao Z, Li Z, Li D, Bai L, Zhao R (2022) Jointly contrastive representation learning on road network and trajectory. Preprint, submitted September 14, https://arxiv.org/abs/2209.06389v1.

Nomikos P, MacGregor JF (1994) Monitoring batch processes using multiway principal component analysis. *AIChE J.* 40(8):1361–1375.

Pan SJ, Yang Q (2009) A survey on transfer learning. *IEEE Trans. Knowledge Data Engrg.* 22(10):1345–1359.

Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: A review. *ACM SIGKDD Explorations Newsletter* 6(1):90–105.

Pelletier MP, Trépanier M, Morency C (2011) Smart card data use in public transit: A literature review. *Transportation Res. Part C Emerging Tech.* 19(4):557–568.

Peng X, Yi Z, Tang H (2015) Robust subspace clustering via thresholding ridge regression. *Proc. AAAI Conf. Artificial Intelligence*, vol. 29 (AAAI Press, Palo Alto, CA).

Shen B, Xie W, Kong ZJ (2022) Smooth robust tensor completion for background/foreground separation with missing pixels: Novel algorithm with convergence guarantee. *J. Machine Learn. Res.* 23(217):1–40.

Sofuoglu SE, Aviyente S (2021) Low-rank on graphs plus temporally smooth sparse decomposition for anomaly detection in spatio-temporal data. *2021 IEEE Internat. Conf. Acoustics Speech Signal Processing (ICASSP)* (IEEE, Piscataway, NJ), 5614–5618.

Sun WW, Li L (2019) Dynamic tensor clustering. *J. Amer. Statist. Assoc.* 114(528):1894–1907.

Sun B, Zhao Z, Liu D, Gao X, Yu T (2022) Tensor decomposition-inspired convolutional autoencoders for hyperspectral anomaly detection. *IEEE J. Selected Topics Appl. Earth Observations Remote Sensing* 15:4990–5000.

Tucker LR (1966) Some mathematical notes on three-mode factor analysis. *Psychometrika* 31(3):279–311.

Vidal R (2011) Subspace clustering. *IEEE Signal Processing Magazine* 28(2):52–68.

Vidal R, Favaro P (2014) Low rank subspace clustering (LRSC). *Pattern Recognition Lett.* 43:47–61.

Vidal R, Tron R, Hartley R (2008) Multiframe motion segmentation with missing data using PowerFactorization and GPCA. *Internat. J. Comput. Vision* 79(1):85–105.

von Luxburg U (2007) A tutorial on spectral clustering. *Statist. Comput.* 17(4):395–416.

Wang M, Zhen Y, Pan Y, Xu Z, Guo R, Zhao X (2023) Tensorized hypergraph neural networks. Preprint, submitted June 5, https://arxiv.org/abs/2306.02560v1.

Xue N, Papamakarios G, Bahri M, Panagakis Y, Zafeiriou S (2017) Robust low-rank tensor modelling using Tucker and CP decomposition. *2017 25th Eur. Signal Processing Conf. (EUSIPCO)* (IEEE, Piscataway, NJ), 1185–1189.

Yan H, Paynabar K, Shi J (2015) Image-based process monitoring using low-rank tensor decomposition. *IEEE Trans. Automation Sci. Engrg.* 12(1):216–227.

Yan H, Paynabar K, Shi J (2017) Anomaly detection in images with smooth background via smooth-sparse decomposition. *Technometrics* 59(1):102–114.

Yan H, Paynabar K, Shi J (2018) Real-time monitoring of high-dimensional functional data streams via spatio-temporal smooth sparse decomposition. *Technometrics* 60(2):181–197.

Yang C, Robinson D, Vidal R (2015) Sparse subspace clustering with missing entries. Bach F, Blei D, eds. *Internat. Conf. Machine Learn.* (PMLR, New York), 2463–2472.

Yang S, Wu J, Xu Y, Yang T (2019) Revealing heterogeneous spatio-temporal traffic flow patterns of urban road network via tensor decomposition-based clustering approach. *Physica A Statist. Mech. Appl.* 526:120688.

Zhang C, Zheng B, Tsung F (2023) Multi-view metro station clustering based on passenger flows: A functional data-edged network community detection approach. *Data Mining Knowledge Discovery* 37(3):1154–1208.

Zhang C, Yan H, Lee S, Shi J (2020) Dynamic multivariate functional data modeling via sparse subspace learning. *Technometrics* 63(3): 370–383.

Zhang C, Fu H, Liu S, Liu G, Cao X (2015) Low-rank tensor constrained multiview subspace clustering. *Proc. IEEE Internat. Conf. Comput. Vision* (IEEE, Piscataway, NJ), 1582–1590.

Zhao Y, Yan H, Holte S, Mei Y (2021) Rapid detection of hot-spots via tensor decomposition with applications to crime rate data. *J. Appl. Statist.* 49(7):1636–1662.

Zhao X, Yan H, Hu Z, Du D (2022b) Deep spatio-temporal sparse decomposition for trend prediction and anomaly detection in cardiac electrical conduction. *IISE Trans. Healthcare Systems Engrg.* 12(2):150–164.

Zhao Y, Yan H, Holte SE, Kerani RP, Mei Y (2020) Rapid detection of hot-spot by tensor decomposition on space and circular time with application to weekly gonorrhea data. *XIIIth Internat. Workshop Intelligent Statist. Quality Control 2019*.

Zhao X, Dai Q, Wu J, Peng H, Liu M, Bai X, Tan J, Wang S, Philip SY (2022a) Multi-view tensor graph neural networks through reinforced aggregation. *IEEE Trans. Knowledge Data Engrg.* 35(4): 4077–4091.