A Self-Representation Method with Local Similarity Preserving for Fast Multi-View Outlier Detection

YU WANG, CHUAN CHEN*, and JINRONG LAI, Sun Yat-sen University, China
LELE FU, Sun Yat-sen University, China
YUREN ZHOU and ZIBIN ZHENG, Sun Yat-sen University, China

With the rapidly growing attention to multi-view data in recent years, multi-view outlier detection has become a rising field with intense research. These researches have made some success, but still exist some issues that need to be solved. First, many multi-view outlier detection methods can only handle datasets that conform to the cluster structure but are powerless for complex data distributions such as manifold structures. This overly restrictive data assumption limits the applicability of these methods. In addition, almost the majority of multi-view outlier detection algorithms cannot solve the online detection problem of multi-view outliers. To address these issues, we propose a new detection method based on the local similarity relation and data reconstruction, i.e., the Self-Representation Method with Local Similarity Preserving for fast multi-view outlier detection (SRLSP). By using the local similarity structure, the proposed method fully utilizes the characteristics of outliers and detects outliers with an applicable objective function. Besides, a well-designed optimization algorithm is proposed, which completes each iteration with linear time complexity and can calculate each instance parallelly. Also, the optimization algorithm can be easily extended to the online version, which is more suitable for practical production environments. Extensive experiments on both synthetic and real-world datasets demonstrate the superiority of the proposed method on both performance and time complexity.

CCS Concepts: • **Computing methodologies → Anomaly detection**; *Cluster analysis*; Online learning settings.

Additional Key Words and Phrases: outlier detection, multi-view data, subspace learning, adaptive similarity learning

1 INTRODUCTION

Unsupervised outlier detection aims to detect the abnormal data in a given dataset, while these abnormal data, or named outliers, are markedly inconsistent with the normal instances [2, 31]. In general, the traditional unsupervised outlier detection methods assume that normal instances are similar to each other, and most of the instances are normal. This means that the description of data is "compact" i.e., normal instances are generated by the same mechanism, while the outliers are not [22, 40]. Based on this principle, a massive number of outlier detection methods are designed in recent decades, e.g., clustering-based models, distance-based models, density-based models, probabilistic models, etc [1]. Besides these shallow models, many deep outlier detection methods have also been proposed [37, 39]. These different types of outlier detection methods are widely applied to a

*The corresponding author

Authors' addresses: Yu Wang, wangy475@mail2.sysu.edu.cn; Chuan Chen, chenchuan@mail.sysu.edu.cn; Jinrong Lai, laijr@mail2.sysu.edu.cn, Sun Yat-sen University, School of Computer Science and Engineering, Higher Education Mega Center, Panyu District, Guangzhou, China; Lele Fu, lawrencefzu@gmail.com, Sun Yat-sen University, School of Systems Science and Engineering, Higher Education Mega Center, Panyu District, Guangzhou, China; Yuren Zhou, zhouyuren@mail.sysu.edu.cn; Zibin Zheng, zhzibin@mail.sysu.edu.cn, Sun Yat-sen University, School of Computer Science and Engineering, Higher Education Mega Center, Panyu District, Guangzhou, China.

1556-4681/2022/1-ART1 $15.00
https://doi.org/10.1145/3532191

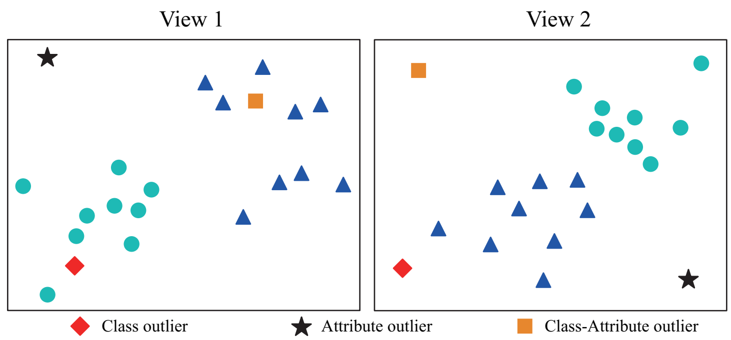


Fig. 1. An illustration of three types of multi-view outlier: (1) Class outlier (the red diamond) is closed to the green circles in the first view, but is closed to the blue triangles in the second view; (2) Attribute outlier (the black star) is remote from the other points in both views; (3) Class-Attribute outlier (the orange square) is close to the other points in the first view but is remote from the other points in the second view.

wide range of data mining scenarios, such as data cleaning, credit card fraud, network intrusion detection, web service QoS prediction, DRAM failure prediction, and even in popular federal learning in recent years [1, 3, 27, 41, 44, 49, 54].

The traditional outlier detection mainly focuses on single-source data, i.e., single-view data. However, with the constantly improving ability of data collection, mixed-type data and multi-view data are taken more seriously [12, 55]. Mixed-type data are composed of a mixture of both numerical (continuous) features and categorical (discrete) features. In recent years, many works [9, 21, 26, 52] on mixed-type outlier detection have been proposed. Among them, MIX [52] fully considers the behavior of data points in different feature spaces, and collaboratively iteratively optimizes its outlier scoring stage to get a more accurate estimate. Multi-view data is usually represented by a heterogeneous but related group of features, which are collected from diverse sources or obtained from various domains, and one of them is often called one view of the dataset [14]. Any particular single-view data cannot comprehensively represent the instance itself, so it's essential to fully utilize the abundant information in multi-view data [30, 50, 58]. Nevertheless, multi-view outlier detection is widely divergent from the single-view one, which is largely due to the two characteristics of multi-view data — consensus and complementarity [45]. Consensus means that features from different views share a part of the same information and show consistent behavior [35]. If an instance fails to meet this constraint, which is likely to happen in a multi-view scenario, e.g., the same person belongs to different social communities in different social networks, this instance is probably an outlier [20]. Interestingly, points that exhibit inconsistent behavior may also be normal points. If there is little redundancy/consistency between different views of the multi-view data (e.g. color view and weight view), then these points exhibiting inconsistent behavior may indeed not be outliers. However, the current multi-view research does not pay attention to such multi-view data, because there is no redundant information/consistent information between views, which will lead to the failure of multi-view learning [51]. Correspondingly, the current multi-view outlier detection methods only focus on the multi-view data with consistent information between the views, so those data points showing inconsistent behaviors can generally be regarded as outliers. These complex behaviors require the redefinition of multi-view outliers, and specifically, based on the characteristics of multi-view data and existing literature [16, 28], three kinds of multi-view outliers are involved in this paper, as shown in Figure 1:

- **Class outliers** exhibit inconsistent behavior, and their local neighborhood relationships are often completely various across different views, which means they have different neighborhoods in different views.

- **Attribute outliers** are the traditional single-view outliers, and they are often very dissimilar from the normal instances.
- **Class-Attribute outliers** are the mixture of the above two types of outliers. They exhibit the class outliers behaviors in some views and exhibit the Attribute outliers behaviors in the others.

Several multi-view outlier detection methods are proposed to detect one or more types of multi-view outliers in these years [16, 20, 24, 28, 29, 34, 43, 56, 57]. They have indeed made some success to some extent, but still leave some issues that need to be handled:

I1 **The arbitrary number of views**: The early methods only consider the two-views scenario and use the pair-wise constraint to model multi-view consensus. It's not easy to extend these methods to three or more views scenario.
I2 **The dataset without clustering structure**: Some methods are derived from clustering theory, so they generally show poor performance on the datasets without clustering structure.
I3 **Different types of outliers**: Some methods can not handle all three types of outliers. Even if they could, they can hardly achieve the ideal performance for different outliers proportion.
I4 **The large-scale dataset**: Some methods are suffered from high time complexity. They spend $O(\mathrm{N}^2)$ or even $O(\mathrm{N}^3)$ time complexity in each iteration, so they are not suitable for large-scale datasets.
I5 **The online scenario**: Hardly any model consider the online scenario, which is practical in many real-world problems.

We summarize the relationship between the existing methods and the proposed method for solving these issues, as shown in Table 1. In these methods, CRMOD, LDSR, MODDIS, and NCMOD handle two or more views (issue **I1**) by learning a cross-views consistent representation (multi-view consensus learning strategy). MODDIS, MUVAD, and NCMOD handle datasets without clustering structure (issue **I2**) by considering the neighborhood in the locality. Inspired by these methods, the multi-view consensus learning strategy allows the model to conveniently handle the real multi-view scenario. And taking multi-view consistent neighborhood relationships into account is more meaningful and direct than comparing the cross-view clustering membership, whereas clustering membership is regarded as a product of neighborhood relationships under the graph-based clustering methods [36, 48]. This motivates us to improve the multi-view consistent neighborhood relationship method so that it can perform fast but accurate outlier detection on all types of outliers.

In this paper, we propose a new fast multi-view outlier detection method based on the local similarity relation and data reconstruction, i.e., the Self-Representation Method with Local Similarity Preserving for fast multi-view outlier detection (SRLSP). SRLSP maintains the advantage of the multi-view consistent neighborhood relationship method, while has the superior ability to detect the different types of multi-view outliers and can be performed on online scenarios. Consequently, the main contributions of this paper are summarized as follows:

- The proposed method learns a common cross-view similarity matrix from multi-view data, which can easily extend to two or more views scenario (issue **I1**). The similarity learning and self-representation learning processes jointly capture the characteristics of different types of data (issue **I3**), making the proposed method achieve superior and stable performance on different outliers proportion by calculating outlier scores.
- The similarity learning process does not depend on the data having a distinct clustering structure (issue **I2**), and the cross-view neighbor similarity is taken into consideration to compare the consistency. Considering only a few neighbors greatly reduces the time complexity compared to considering the whole dataset. By setting the number of neighbors used in the similarity learning process as a parameter, the proposed method can complete each iteration with $O(NlogN)$ time complexity and can even calculate each instance parallelly (issue **I4**).

Table 1. Comparison of the existing methods and the proposed method. Circle ◯ means method can handle this issue and X mark × otherwise.

Method	I1	I2	I3	I4	I5
HOAD [20]	×	×	×	$O(N^2)$	×
AP [34])	×	×	×	$O(N^2 logN)$	×
DMOD [56]	×	×	◯	$O(N)$	×
CRMOD [57]	◯	×	◯	$O(N)$	×
MLRA [29]	×	×	◯	$O(N^3)$	×
LDSR [28]	◯	×	◯	$O(N^3)$	×
MODDIS[1][24]	◯	◯	◯	-	×
MUVAD [43]	×	◯	◯	$O(N^2)$	×
NCMOD[1][16]	◯	◯	◯	-	×
Proposed method (SRLSP)	◯	◯	◯	$O(NlogN)$	◯

[1] MODDIS and NCMOD are neural network method, and it's not easy to directly compare the time complaxity with the traditional methods.

- To the best of our knowledge, we are the first multi-view outlier detection method to handle online scenario. By introducing the initial normal set, the proposed method can be easily extended to an online version (issue **I5**), whereas most of the previous methods are not suitable to make this extension.

2 RELATED WORK

Outlier detection is an important topic in data analysis, which can help to accurately evaluate and make full use of the originally collected data [1, 5]. Up to now, several single-view outlier detection methods have been proposed [4, 11, 19, 32, 40, 46, 53], concentrating mainly on unsupervised and semi-supervised learning. However, there are only a few multi-view outlier detection methods in the literature. In the early stages of multi-view outlier detection research, the definition of multi-view outliers is ambiguous. [20] first studies the "inconsistent behaviors" in multi-view data and uses the result of multi-view spectral clustering to predict whether an instance performs inconsistency, which this method is named HOAD. HOAD conducts spectral clustering on the input similarity graphs of two views simultaneously with a consistency constraint. Although the framework of this method is simple and the performance remains poor, it comes up as a pioneering work and is worth studying. Another earlier work also addresses the problem of "inconsistent behaviors" and provides a preliminary consensus on the definition of class outliers [23]. [34] follows up with a multi-view outlier detection method AP, which is based on affinity propagation clustering and raises concern about the topic of multi-view outlier detection. AP has achieved a very good performance on detecting "inconsistent behavior" and is often regarded as the baseline. However, both HOAD and AP can only detect "inconsistent behaviors", or class outliers in this paper, so they can not handle other kinds of outliers perfectly.

To better connect with the single-view outlier detection, attribute outliers and class-attribute outliers are defined one after another. [56] proposes the method DMOD, which performs k-means clustering on each view and uses the pair-wise constraint to reduce the difference of the clustering indicators. In this work, the widely used $L_{2,1}$-norm [17, 47] is introduced to model the error term, which guarantees the sparsity of the error matrix in the dimension of instances. A well-defined outlier measurement criterion is proposed to help DMOD detecting both attribute and class outliers. To overcome the limitation of pair-wise constraint, [57] improves the objective of DMOD, adding a cross-views consistent clustering indicator to constrain the similarity between the view-specific clustering indicators and the cross-views consistent clustering indicator, which the model is named CRMOD. The relationship between MLRA [29] and LDSR [28] is similar to the relationship between DMOD and CRMOD, but the clustering method is changed to the low-rank representation for subspace clustering. However, all the above methods are directly related to multi-view clustering, which can not separate these methods from the clustering scenario.

Most recently, the neighborhood in the locality is considered to reveal the deeper characteristics of multi-view outliers [43], which helps to design the outlier measurement criterion in datasets without clustering structure. And neural networks are also adopted to explore the latent representation of multi-view data [24] or handle the high-dimensional data [16]. NCMOD automatically encodes the intrinsic information of each view into a comprehensive latent space with a consensus neighborhood structure and then enables multi-view outlier detection by capturing the inconsistency of neighbor structures across views. NCMOD inspires us to focus on the latent local similarities in real data.

However, these methods often have high time complexity, and none of them can solve the problem of online detection of multi-view outliers. In this paper, we pay more attention to how to design a fast multi-view outlier detection method, which can adapt easily to the online scenario.

3 THE PROPOSED METHOD

Given a specific multi-view dataset $\boldsymbol{X} = \{\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(\mathrm{V})}\}$, where $\boldsymbol{X}^{(v)} = \{\boldsymbol{X}_1^{(v)}, \dots, \boldsymbol{X}_{\mathrm{N}}^{(v)}\} \in \mathbb{R}^{\mathrm{N}\times\mathrm{D}^{(v)}}$ is the feature matrix of the v-th view. And N, V and $\mathrm{D}^{(v)}$ refer to the number of instances, views and the feature dimension of the v-th view respectively. The unsupervised multi-view outlier detection aims to score each instance according to its probability of being any kinds of multi-view outliers. Except otherwise defined, the i-th row of a matrix $\boldsymbol{A}$ is defined as $\boldsymbol{A}_i$, and given a set of indices $\mathcal{N}$, $\boldsymbol{A}_{\mathcal{N}} = \{\boldsymbol{A}_j | j \in \mathcal{N}\}$ is the submatrix of $\boldsymbol{A}$ with the rows corresponding to the indices in $\mathcal{N}$. In Section 3.1, we address the limitations of current multi-view outlier detection methods on data assumptions, and propose a neighbor-based self-representation learning submodule. In Section 3.2, we propose an adaptive similarity learning submodule for the problem of detecting inconsistent behavior of data points across views. In Section 3.3, we will fuse our proposed two submodules to obtain the final objective function. In Section 3.4, we propose our outlier score function to enable detection of multiple multi-view outliers.

3.1 The Neighbor-based Self-Representation Submodule

Now let us consider how to detect the attribute outliers. Note that attribute outliers locate far away from their neighbors in either view. Based on this observation, we consider reconstructing each sample point with its neighbors. Normal points can be reconstructed by nearby neighbors with a smaller reconstruction coefficient; while attribute outliers are reconstructed by distant neighbors (normal points), resulting in a larger scale of reconstruction coefficient. Therefore, the self-representation method can be deployed for the attribute outliers detection.

However, directly applying traditional self-expression methods to multi-view outlier detection leads to limitations in data assumptions. MLRA and LDSR simply use linear combinations of all data points in the dataset for self-expression, resulting in these methods only dealing with data distributions that conform to the assumption of global cluster structure. To accommodate more general data assumptions, we propose to limit the scope of self-expression to neighbors, proposing a neighbor-based self-expression learning submodule.

Datasets with high-dimensional features are usually strongly correlated with certain low-dimensional subspaces, rather than being uniformly distributed throughout the space [18, 28]. Specifically, in single-view dataset $\boldsymbol{X} \in \mathbb{R}^{\mathrm{N}\times\mathrm{D}}$, each data point can be reconstructed by a linear combination of other points in the same subspace, which is called the <u>self-representation</u> method:

$$\boldsymbol{X}_i = \boldsymbol{S}_i \boldsymbol{X} + \boldsymbol{E}_i, \tag{1}$$

where $\boldsymbol{S} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$ is the subspace representation matrix and $\boldsymbol{E} \in \mathbb{R}^{\mathrm{N}\times\mathrm{D}}$ models the reconstruction error. By adding the sparse or low-rank constraint on $\boldsymbol{S}$, the hidden subspaces are found in the original data from the local or global perspective. Because our basic assumption is that the dataset doesn't necessarily have a global clustering structure, and the optimization of the nuclear norm [13] is often accompanied by high time complexity, we pay more attention to the sparse constraint used from the local perspective. Inspired by LSR [33] and LLE [38], we select a few neighbors for self-representation, to ensure the sparsity of the representation matrix $\boldsymbol{S}$, and used the Frobenius norm to guarantee the smooth of the representation weight $\boldsymbol{S}$ and the reconstruction error $\boldsymbol{E}$. Thus we can learn the local structure without clustering prior by solving the following problem:

$$\begin{aligned} &\min_{S} \|\boldsymbol{X} - \boldsymbol{S}\boldsymbol{X}\|_F^2 + \gamma\|\boldsymbol{S}\|_F^2 \\ &s.t.\ \boldsymbol{S}_{i,j} = 0, \forall j \notin \mathcal{N}(i), \forall i = 1, \ldots, \mathrm{N}, \end{aligned} \tag{2}$$

where $\mathcal{N}(i)$ is the indices of the k nearest neighbors of the i-th instance, and γ is the trade-off parameter. Notice that $\boldsymbol{S}_{i,j}$ is greater than 0 only when $j \in \mathcal{N}(i)$, we use $\boldsymbol{Z} \in \mathbb{R}^{\mathrm{N}\times k}$ to represent the elements which can greater than 0 in $\boldsymbol{S}$, and take the corresponding row of $\boldsymbol{X}$ in $\mathcal{N}(i)$, i.e., $\boldsymbol{X}_{\mathcal{N}(i)}$. So the objective function (2) can be further formulated as:

$$\min_{\boldsymbol{Z}} \sum_{i=1}^{\mathrm{N}} \left(\|\boldsymbol{X}_i - \boldsymbol{Z}_i \boldsymbol{X}_{\mathcal{N}(i)}\|_2^2 + \gamma\|\boldsymbol{Z}_i\|_2^2 \right). \tag{3}$$

Attribute outliers always lie far away from the normal instances, so only the normal instances can be reconstructed with small weights in $\boldsymbol{Z}_i$. By using l_2-norm on $\boldsymbol{Z}_i$, attribute outliers' representation weights are punished, resulting in apparent large reconstruction error $\boldsymbol{E}_i$. This relationship of failing in reconstruction is mutual, which means the normal instances can not use the attribute outliers to reconstruct, nor can the attribute outliers. The proper value of k guarantees the capacity of the neighbors set, which ensures that the self-representation process of normal instances can proceed successfully, but not for attribute outliers.

3.2 Adaptive Similarity Learning Submodule with Graph Fusion

In order to further deal with class outliers, we illustrate the effectiveness of the similarity matrix in detecting class outliers, and further innovatively propose an adaptive similarity learning submodule.

When it came to the detection of the class outliers, we note that the neighbor relationship varies across varying views. Since the similarity matrix could reflect the neighbor relations among different samples, we capture the inconsistency of neighbor relations by observing the inconsistency of the similarity matrix. Specifically, normal points have similar neighbor relationships in different views, so the consistent similarity matrix learned by fusing all views should not be very different from the similarity matrix of each view. However, due to the inconsistency of the neighbor relationship between the class outliers in different views, the consensus similarity matrix obtained after the fusion should be quite different from the similarity matrix of each view.

Usually, to learn a suitable similarity matrix for a specific graph-based model (e.g., graph clustering [6, 7, 15] and graph-based information retrieval [8]) without extensive manual tuning on a specific similarity measure, the adaptive similarity learning method is often used to combine with the objective by solving the following problem [36]:

$$\begin{aligned} &\min_{S} \sum_{i,j=1}^{\mathrm{N}} \left(\boldsymbol{S}_{i,j}\|\boldsymbol{X}_i - \boldsymbol{X}_j\|_2^2 + \gamma \boldsymbol{S}_{i,j}^2 \right) \\ &s.t.\ \boldsymbol{S}_i\mathbf{1} = 1, \\ &\qquad 0 \le \boldsymbol{S}_i \le 1, \\ &\qquad \forall i = 1, \ldots, \mathrm{N}, \end{aligned} \tag{4}$$

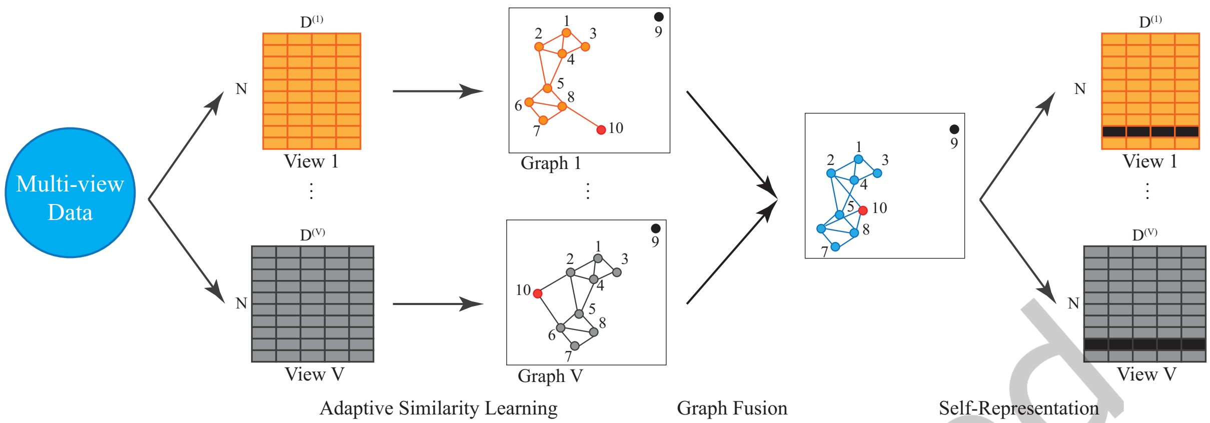


Fig. 2. Illustration of the proposed method. The adaptive similarity learning process extracts the view-specific similarity information and the graph-fusion process uniforms this information by leaning a common cross-view similarity consensus. The self-representation process further verifies the validity of this consensus by data reconstruction. For instance, the class outlier with ID 10 can not achieve cross-view consensus, and the attribute outlier with ID 9 fails to reconstruct the original data, so they can separate from the normal instances in the proposed model.

where parameter $\gamma \in (0, +\infty)$ controls the smoothness of $\boldsymbol{S}_i$: when $\gamma \to 0$, $\boldsymbol{S}_i$ becomes sharp, i.e., only the instance $\boldsymbol{X}_j$ closest to $\boldsymbol{X}_i$ has $\boldsymbol{S}_{i,j} = 1$, otherwise $\boldsymbol{S}_{i,j} = 0$; when $\gamma \to +\infty$, $\boldsymbol{S}_i$ becomes smooth, i.e., $\boldsymbol{S}_{i,j} = \frac{1}{\mathrm{N}}, \forall j = 1, \ldots, \mathrm{N}$. We extend the single-view adaptive similarity learning method Equation (4) to multi-view cases. To be specific, we assume that multi-view data share a unified similarity matrix and adopt a efficient multi-graph fusion technology [25]:

$$\begin{aligned} \min_{\boldsymbol{S}, \boldsymbol{S}^{(1)}, \ldots, \boldsymbol{S}^{(\mathrm{V})}} & \sum_{v=1}^{\mathrm{V}} \Bigg(\sum_{i,j=1}^{\mathrm{N}} \left(\boldsymbol{S}_{i,j}^{(v)} \|\boldsymbol{X}_i^{(v)} - \boldsymbol{X}_j^{(v)}\|_2^2 + \gamma {\boldsymbol{S}_{i,j}^{(v)}}^2 \right) \\ & + \lambda \|\boldsymbol{S}^{(v)} - \boldsymbol{S}\|_F^2 \Bigg) \\ s.t.\ & \boldsymbol{S}_i^{(v)} \mathbf{1} = 1, \\ & 0 \leq \boldsymbol{S}_i^{(v)} \leq 1, \\ & \forall v = 1, \ldots, \mathrm{V},\ \forall i = 1, \ldots, \mathrm{N}, \end{aligned} \tag{5}$$

where λ is the trade-off parameter. The characteristic of consensus in multi-view data indicates that normal instances have similar neighbor structures and similarities, so a common similarity can be learned. Nevertheless, the "inconsistent behaviors" indicates that class outliers learn different view-specific similarities with respect to different view data, which fail to explore the common cross-view similarity and result in large fusion error.

3.3 The Proposed Method

We argue that the fusion of the above two submodules is necessary, and also believe that detecting attribute outliers and class outliers are interdependent. There are two reasons for this practice:

- Multi-view class outliers are somewhat similar to the single-view attribute outliers when all views of features are concatenated together.
- Class-attribute outliers exhibit both the behaviors of attribute outliers and class outliers, so it's not suitable to use any single detection model above.

Given this premise, we combine Equation (3) and Equation (5), and propose our objective function:

$$\begin{aligned}\min_{Z,Z^{(1)},\ldots,Z^{(\mathrm{V})}} \; & \sum_{i=1}^{\mathrm{N}} \left(\sum_{v=1}^{\mathrm{V}} \left(\|X_i^{(v)} - Z_i X_{\mathcal{N}(i)}^{(v)}\|_2^2 + \right.\right. \\ & \left.\left. \lambda \|Z_i - Z_i^{(v)}\|_2^2 + \mu D_i^{(v)} {Z_i^{(v)}}^T \right) + \gamma \|Z_i\|_2^2 \right) \\ s.t. \; & Z_i^{(v)} \mathbf{1} = 1, \\ & 0 \le Z_i^{(v)} \le 1, \\ & \forall v = 1, \ldots, \mathrm{V}, \; \forall i = 1, \ldots, \mathrm{N}, \end{aligned} \tag{6}$$

where $\mathcal{N}(i)$ is the predefined multi-view neighbor set and

$$D_{i,j}^{(v)} = \|X_i^{(v)} - \left(X_{\mathcal{N}(i)}^{(v)}\right)_j\|_2^2, \; j = 1, \ldots, |\mathcal{N}(i)|. \tag{7}$$

This objective function Equation (6) consists of the following parts (as presented in Figure 2): Adaptive similarity learning term, i.e., $D_i^{(v)} {Z_i^{(v)}}^T$, learns the view-specific similarity $Z_i^{(v)}$ among neighbors; Graph fusion term, i.e., $\|Z_i - Z_i^{(v)}\|_2^2$, learns the common cross-view similarity Z; Self-Representation term, i.e., $\|X_i^{(v)} - Z_i X_{\mathcal{N}(i)}^{(v)}\|_2^2$, reconstructes the origin view data through the common cross-view similarity; Regularization term, i.e., $\|Z_i\|_2^2$, acts as the l_2-norm in self-representation method and the smoothness regularization in adaptive similarity learning method.

3.4 Outlier Measurement Criterion

As analyzed above, the attribute outliers have large values in self-representation term, and the class outliers have large values in the graph-fusion term, we can get the outlier scoring function as:

$$\text{outlier\_score}(i) = \sum_{v=1}^{\mathrm{V}} \left(\|X_i^{(v)} - Z_i X_{\mathcal{N}(i)}^{(v)}\|_2^2 + \lambda \|Z_i - Z_i^{(v)}\|_2^2 \right). \tag{8}$$

Notice that the parameter λ is shared in the objective function Equation (6) and Equation (8), which is because that the objective function serves the same aim as the outlier scoring function.

4 OPTIMIZATION

Problem (6) can be solved by the efficient alternating optimization method which updates Z and $Z^{(v)}$ iteratively.

4.1 Fix Z and Update $Z^{(v)}$

For the given i-th instance, keeping only the terms relevant to $Z_i^{(v)}$ in problem (6), we obtain:

$$\begin{aligned}\min_{Z_i^{(v)}} \; & \lambda \|Z_i - Z_i^{(v)}\|_2^2 + \mu D_i^{(v)} {Z_i^{(v)}}^T \\ s.t. \; & Z_i^{(v)} \mathbf{1} = 1, \\ & 0 \le Z_i^{(v)} \le 1. \end{aligned} \tag{9}$$

Algorithm 1 The enumeration method to solve problem (11).

Input: $\boldsymbol{d}$ and γ;
Output: $\boldsymbol{s}$;
```
Sort d from smallest to largest;
for p = 1, ..., D do
    Calculate λ using (14);
    if λ − d_p > 0 and λ − d_{p+1} ≤ 0 then
        Record current λ and break;
    end if
end for
for j = 1, ..., D do
    Calculate s_j using Equation (13);
end for
```

Reformulating the Equation (9) and ignoring the constant, we have:

$$\begin{aligned} \min_{Z_i^{(v)}} \; & \left(\mu \boldsymbol{D}_i^{(v)} - 2\lambda \boldsymbol{Z}_i\right) \boldsymbol{Z}_i^{(v)^T} + \lambda \|\boldsymbol{Z}_i^{(v)}\|_2^2 \\ s.t.\; & \boldsymbol{Z}_i^{(v)}\mathbf{1} = 1, \\ & 0 \leq \boldsymbol{Z}_i^{(v)} \leq 1. \end{aligned} \tag{10}$$

The optimization problem (10) can be abstracted as:

$$\begin{aligned} \min_{s} \; & \boldsymbol{d}^T \boldsymbol{s} + \gamma \|\boldsymbol{s}\|_2^2 \\ s.t.\; & \boldsymbol{s}^T \mathbf{1} = 1, \\ & 0 \leq \boldsymbol{s} \leq 1, \end{aligned} \tag{11}$$

where $\boldsymbol{d}, \boldsymbol{s} \in \mathbb{R}^{\mathrm{D}}$. The Lagrangian function of problem (11) is

$$\mathcal{L}(\boldsymbol{s}, \lambda, \boldsymbol{\nu}) = \boldsymbol{d}^T \boldsymbol{s} + \gamma \|\boldsymbol{s}\|_2^2 - \lambda(\boldsymbol{s}^T - 1) - \boldsymbol{\nu}^T \boldsymbol{s}. \tag{12}$$

According to the KKT condition [10], the optimal solution $\boldsymbol{s}$ should be

$$\boldsymbol{s}_j = \frac{1}{2\gamma}\left(\lambda - \boldsymbol{d}_j\right)_+. \tag{13}$$

Without loss of generality, assume that elements in $\boldsymbol{d}$ is sorted, i.e., $\boldsymbol{d}_1 \leq \boldsymbol{d}_2 \leq \ldots \boldsymbol{d}_{\mathrm{D}}$, the first p elements satisfy $\lambda - \boldsymbol{d}_j > 0, \forall j = 1, \ldots, p$, we have:

$$\lambda = \frac{1}{p}\left(2\gamma + \sum_{j=1}^{p} \boldsymbol{d}_j\right). \tag{14}$$

We can calculate $\boldsymbol{s}$ and λ by enumerating p, which is summarized in Algorithm 1.

4.2 Fix $Z^{(v)}$ and Update Z

For the given i-th instance, ignoring the irrelevant terms with respect to $\boldsymbol{Z}_i$ in problem (6), we obtain:

$$\min_{Z_i} \sum_{v=1}^{\mathrm{V}} \left(\|\boldsymbol{X}_i^{(v)} - \boldsymbol{Z}_i \boldsymbol{X}_{\mathcal{N}(i)}^{(v)}\|_2^2 + \lambda \|\boldsymbol{Z}_i - \boldsymbol{Z}_i^{(v)}\|_2^2\right) + \gamma \|\boldsymbol{Z}_i\|_2^2. \tag{15}$$

Algorithm 2 Self-Representation with Local Similarity Preserving for fast multi-viewoutlier detection

Input: Multi-view dataset $\boldsymbol{X} = \left\{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{\mathrm{V}}\right\}$; Parameters λ, μ, γ and the neighbors number k;
Output: outlier_score;
Use the k-nn algorithm to abtain the neighbors set in each view, and then merge them into the unified $\mathcal{N}(i)$;
Calculate $\boldsymbol{D}_i^{(v)}$ using Equation (7);
$\boldsymbol{Z} = \boldsymbol{0}$;
repeat
  for $i = 1, \ldots, \mathrm{N}$ **do**
    for $v = 1, \ldots, \mathrm{V}$ **do**
      Update $\boldsymbol{Z}_i^{(v)}$ by using Algorithm 1 to solve the problem (10);
    end for
    Update $\boldsymbol{Z}_i$ using Equation (16);
  end for
until converge
for $i = 1, \ldots, \mathrm{N}$ **do**
  Calculate outlier_score using Equation (8);
end for

Taking the derivative of Equation (15) and solving it, we have:

$$\boldsymbol{Z}_i = \left(\sum_{v=1}^{\mathrm{V}} \left(\boldsymbol{X}_i^{(v)} \boldsymbol{X}_{\mathcal{N}(i)}^{(v)}{}^T + \lambda \boldsymbol{Z}_i^{(v)}\right)\right)\left(\sum_{v=1}^{\mathrm{V}} \boldsymbol{X}_{\mathcal{N}(i)}^{(v)} \boldsymbol{X}_{\mathcal{N}(i)}^{(v)}{}^T + \lambda \mathrm{V} \boldsymbol{I} + \gamma \boldsymbol{I}\right)^{-1}. \tag{16}$$

We summarize the algorithm to solve problem (6) in Algorithm 2.

5 THEORETICAL ANALYSIS AND EXTENSION

5.1 Time Complexity

In this subsection, we analyze the time complexity of Algorithm 2. The features dimension of v-th view is $\mathrm{D}^{(v)}$, which uses D to uniformly represent all the $\mathrm{D}^{(v)}$. The cost of constructing the initial neighbors set by using KNN algorithm is $O(\mathrm{V}(\mathrm{D}+k)\mathrm{N}\log\mathrm{N})$. The cost of storing $\boldsymbol{X}_{\mathcal{N}(i)}^{(v)}$ for all instances and all views is $O(\mathrm{V}k\mathrm{DN})$. $\boldsymbol{X}_{\mathcal{N}(i)}^{(v)} \boldsymbol{X}_{\mathcal{N}(i)}^{(v)}{}^T$ and $\boldsymbol{X}_i^{(v)} \boldsymbol{X}_{\mathcal{N}(i)}^{(v)}{}^T$ can be precomputed to accelerate the update of $\boldsymbol{Z}$, which are spent $O(\mathrm{V}^2k^2\mathrm{DN})$ in total. The cost of calculate $\boldsymbol{D}^{(v)}$ for all views is $O(\mathrm{V}k\mathrm{DN})$. Therefore, the total time complexity of initialization is $O(\mathrm{V}(\mathrm{D}+k)\mathrm{N}\log\mathrm{N} + \mathrm{V}^2k^2\mathrm{DN})$. To update $\boldsymbol{Z}^{(v)}$ with Algorithm 1, we need $O(k\log k)$ for sort and enumeration in one run, and need $O(\mathrm{V}k\log k\mathrm{N})$ in total. To update $\boldsymbol{Z}$ with Equation (16), we need $O(k^3\mathrm{N})$ mainly used to solve the linear function. Overall, suppose the number of iteration is t, the total time complexity of Algorithm 2 is $O(\mathrm{V}(\mathrm{D}+k)\mathrm{N}\log\mathrm{N} + \mathrm{V}^2k^2\mathrm{DN} + (\mathrm{V}k\log k + k^3)\mathrm{N}t)$.

To make the initialization more efficient, we can collect a small normal set to process the selection of neighbors set, which means that we can use a relatively smaller time complexity than $O(\mathrm{V}(\mathrm{D}+k)\mathrm{N}\log\mathrm{N})$ for the k-nn algorithm. As the proposed method mainly aims at the low-dimensional data, D is considered to be small when analyzing the time complexity. In this way, D, k and V are relatively small compared to N and thus can be ignored, and the final time complexity of the proposed method is near $O(\mathrm{N}\log\mathrm{N})$.

Algorithm 3 Online multi-view outlier detection

Input: Normal set $\boldsymbol{A} = \left\{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{\mathrm{V}}\right\}$;
The instance that need to be detected $\boldsymbol{x}$;
Parameters λ, μ, γ and the neighbors number k;
Output: outlier_score($\boldsymbol{x}$);

```
initial: Use the k-nn algorithm to abtain the neighbors set in each view, and then merge them into the
unified N(x);
Calculate d^(v) using Equation (18);
z = 0;
repeat
    for v = 1, ..., V do
        Update z^(v) using Algorithm 1;
    end for
    Update z;
until converge
Calculate outlier_score(x);
```

5.2 Parameters

There are four parameters in the proposed method: λ, μ, γ and k. The parameter λ controls the degree of consistency between view-specific and cross-view similarity, which should be increased with a high class outliers proportion. The parameter μ controls the weight of view-specific similarity learning, which can be simply set to 1 in most cases. The parameter γ controls the weight of regularization, which should be increased with a high attribute outliers proportion. The parameter k is related to the given dataset and usually needs to be adjusted separately first.

5.3 Extension: Online Outlier Detection Algorithm

Suppose there are a given normal instances set $\boldsymbol{A}$, and review the part of objective function (6) with respect to a specific instance $\boldsymbol{x}$, we can simply get the objective function of the online algorithm:

$$\begin{aligned} \min_{\boldsymbol{z}, \boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(\mathrm{V})}} \quad & \sum_{v=1}^{\mathrm{V}} \left(\|\boldsymbol{x}^{(v)} - \boldsymbol{z}\boldsymbol{A}^{(v)}_{\mathcal{N}(\boldsymbol{x})}\|_2^2 + \lambda \|\boldsymbol{z} - \boldsymbol{z}^{(v)}\|_2^2 \right. \\ & \left. + \mu \boldsymbol{d}^{(v)} \boldsymbol{z}^{(v)T} \right) + \gamma \|\boldsymbol{z}\|_2^2 \\ s.t. \quad & \boldsymbol{z}^{(v)} \mathbf{1} = 1, \\ & 0 \le \boldsymbol{z}^{(v)} \le 1, \\ & \forall v = 1, \ldots, \mathrm{V}, \end{aligned} \tag{17}$$

where

$$\boldsymbol{d}_j^{(v)} = \|\boldsymbol{x}^{(v)} - \left(\boldsymbol{A}^{(v)}_{\mathcal{N}(\boldsymbol{x})}\right)_j\|_2^2, \; j = 1, \ldots, |\mathcal{N}(\boldsymbol{x})|. \tag{18}$$

The objective function (17) shares the same form with the origin problem (6), so we can simply summarize the algorithm to solve problem (17) in Algorithm 3. In the practical scenario, if we can not obtain the normal set, we can run Algorithm 3 and select the normal instances according to their outlier scores to build the initial normal set.

Table 2. Datasets and related description: the table data is the feature dimension of each view (with feature name in parenthesis)

View	Synthetic dataset	UCI datasets						Real multi-view datasets		
		iris	**pima**	**zoo**	**ionosphere**	**letter**	**leaf**	**MSRC-v1**	**AWA-10**	**Caltech-7**
1	2	2	4	8	17	8	7	24 (CM)	2688 (CQ)	48 (Gabor)
2	2	2	4	8	17	8	7	576 (HOG)	2000 (LSS)	40 (WM)
3	-	-	-	-	-	-	-	512 (GIST)	252 (PHOG)	254 (CENTRIST)
4	-	-	-	-	-	-	-	256 (LBP)	2000 (SIFT)	1984 (HOG)
5	-	-	-	-	-	-	-	254 (CENT)	2000 (RGSIFT)	512 (GIST)
6	-	-	-	-	-	-	-	-	2000 (SURF)	928 (LBP)
Number of instances	400	150	768	101	351	1300	340	210	800	1474
Number of categories	1	3	2	7	2	26	30	7	10	7

5.4 Compared to Autoencoder

Noting that the proposed method learns the similarity features from the original data and tries to restore the data, this model can be considered as a special autoencoder framework. In this model, the adaptive similarity learning term with regularization constitutes a nonlinear encoder, and the self-expressive learning term is a linear decoder. By introducing additional similarity information, the proposed model can effectively solve the problem that the feature learning is too free so that the model has better performance. However, while simple linear decoders bring simplicity, they also bring the problem of limited representation ability for high-dimensional data, compared with some deep architecture. We take this challenge as a direction for further research in the future.

6 EXPERIMENTS

In this part, we evaluate the proposed method on both synthetic and real datasets. For simplicity, we denote the proposed method as SRLSP in the following context. Specifically, our core code is released at Github[1].

6.1 Experiment Setting

6.1.1 Datasets and Preprocessing. One synthetic dataset is generated without clustering structure, which is used to evaluate the performance on issue **I2**. Six UCI datasets are selected with different outliers proportion settings, which are used to evaluate the performance on issue **I3**. Three real multi-view datasets with more than two views are selected, which are used to evaluate the performance on issue **I1**. Table 2 summarizes the detailed information of the experiments datasets.

The real datasets are all multi-category datasets without outliers, and the UCI datasets are all single-view datasets. For UCI datasets, we split the single-view features evenly into two subsets, generating the two-view datasets. To save the verification time, the original datasets are reduced to a certain extent, and a small of data instances are randomly and uniformly selected from each category for the experiment. We follow the outliers generation method in [28] to pre-process the data with three types of multi-view outliers. To avoid randomness, we repeat this generation method 20 times for the same dataset and calculate the mean values and standard deviations of the running results.

6.1.2 Comparison Methods and Related Setting. We compare SRLSP with seven state-of-the-art methods: OneClassSVM [42], HOAD [20], AP [34], MLRA [29], LDSR [28] , MODDIS [24] and NCMOD [16]. OneClassSVM is the representative method for single-view outlier detection, which is included to verify the detection performance of the single-view method on multi-view outliers setting. Also, the multi-view features are concatenated into one group to ensure that OneClassSVM does work. AP, HOAD, and MLRA are the pair-wise constraint methods,

[1]https://github.com/wy54224/SRLSP

Table 3. The comparison result on the synthetic dataset. AUC values (mean ± standard deviation) are reported, and the best and the second-best results are in **bold** and underline, respectively.

	AUC (mean ± std) ↑
OneClassSVM	0.842±0.018
HOAD	0.468±0.161
AP	0.660±0.042
MLRA	0.790±0.119
LDSR	0.825±0.043
MODDIS	0.975±0.011
NCMOD	0.874±0.033
SRLSP	**0.996±0.005**

which are included to expose the flaws of pair-wise constraint on multi-view data. MLRA and LDSR are also self-representation methods, which are included to show the superiority in combination with local similarity information. MODDIS is the neural network method and shows excellent performance on different multi-view outliers settings. NCMOD is the latest neural network-based method and achieves state-of-the-art performance. Both MODDIS and NCMOD are included as the representative method of state-of-the-art multi-view outlier detection methods.

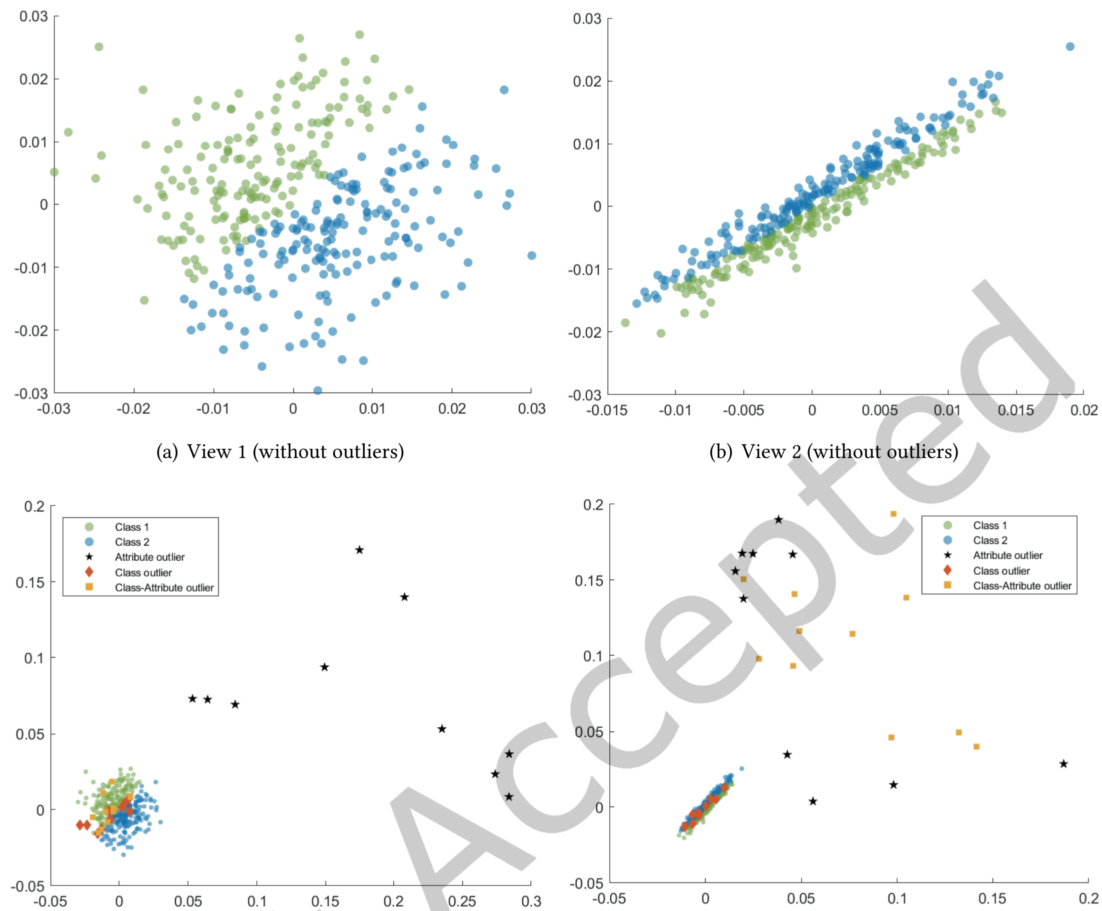
For graph-based methods, the RBF kernel is used to construct the similarity matrix. When the number of views is greater than 2, we enumerate all pairs of views and then sum up the outlier scores for those pair-wise constraint methods. For all methods, we carefully tune the parameters according to the suggestions in the paper. For SRLSP, parameter μ is simply set to 1. Only when the class outliers proportion is relatively large can we appropriately increase μ. Parameter k is searched in $\{2, 4, 7, 10, 20\}$. After setting parameters μ and k, λ and γ are combinatorial searched in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. We adopt AUC, the area under the ROC curve, to evaluate the performance of multi-view outlier detection.

6.2 Experiments on Synthetic Dataset

The synthetic dataset is generated by the python module scikit-learn, which contains 400 instances, 1 clusters and 2 features, as shown in Figure 3(a).

Another view of features is generated by a linear transformation of the original data plus a random perturbation, as shown in Figure 3(b). We process the data according to the outlier generation method in [28] to obtain our synthetic dataset containing outliers, as shown in Figure 3(c) and Figure 3(d). The proportions of attribute outliers, class outliers, and class-attribute outliers are set as $\mathrm{N_A} : \mathrm{N_C} : \mathrm{N_{CA}} = 5\% : 5\% : 5\%$. The comparison result is shown in Table 3.

Among these above methods, OncClassSVM, MODDIS, NCMOD, and the proposed SRLSP are not based on the clustering hypothesis, so these four methods show the best performances on the synthetic dataset. In these four methods, the worst-performing OneClassSVM is the single-view method, which does not fully utilize the characteristics of multi-view data. The proposed SRLSP substantially outperforms other methods, which numerically indicates that the combination of self-representation and similarity method can markedly reduce restrictive conditions on the given dataset.



(a) View 1 (without outliers) (b) View 2 (without outliers)

(c) View 1 (with outliers) (d) View 2 (with outliers)

Fig. 3. Two views of the synthetic dataset: To facilitate the generation of outliers, we divide the dataset evenly into two, assuming that each side belongs to one category. View 2 is generated by a linear transformation of the data in View 1 plus a random perturbation. We process the data according to the outlier generation method in [28], and finally obtain a synthetic dataset containing outliers, as shown in subfigures (c), (d) (For clarity of visualization, we randomly sample half of the outliers for visualization). The proportions of attribute outliers (the black star), class outliers (the red diamond), and class-attribute outliers (the orange square) are set as $N_A : N_C : N_{CA} = 5\% : 5\% : 5\%$.

6.3 Experiments on Real Datasets

6.3.1 Comparison Results on UCI Datasets. The two-views UCI datasets are applied on different outliers proportions settings, which contain only 10% of one type of outliers. The comparison results are shown in Table 4, Table 5, and Table 6. Because UCI datasets are single-view, OneCloseSVM also has good performance on some datasets. As a neural network method, if the network capacity is high, MODDIS will be over-fitting and can not distinguish attribute outliers from normal instances, so it performs very poorly on some attribute outliers datasets. AP has

Table 4. The comparison result on the UCI datasets. AUC values (mean ± standard deviation) are reported, and the best and the second-best results are in **bold** and underline, respectively.

	$N_A : N_C : N_{CA}$ = 10% : 0% : 0%					
	iris	pima	zoo	ionosphere	letter	leaf
OneClassSVM	0.995±0.002	0.028±0.015	0.002±0.005	0.454±0.023	0.364±0.005	0.881±0.006
HOAD	0.122±0.236	0.955±0.171	0.723±0.244	0.562±0.126	0.430±0.094	0.715±0.145
AP	0.037±0.036	0.001±0.002	0.396±0.101	0.483±0.037	0.487±0.012	0.171±0.032
MLRA	0.976±0.027	0.934±0.013	0.846±0.066	0.637±0.152	0.758±0.029	0.990±0.010
LDSR	0.979±0.029	0.994±0.004	0.928±0.018	0.727±0.016	0.996±0.001	**1.000±0.000**
MODDIS	0.758±0.121	0.002±0.003	0.894±0.040	0.710±0.022	0.051±0.028	**1.000±0.000**
NCMOD	0.987±0.014	**0.999±0.001**	0.859±0.045	0.637±0.023	**0.999±0.001**	0.984±0.006
SRLSP	**1.000±0.000**	0.990±0.002	**0.979±0.008**	**0.732±0.007**	0.738±0.007	0.999±0.001

Table 5. The comparison result on the UCI datasets. AUC values (mean ± standard deviation) are reported, and the best and the second-best results are in **bold** and underline, respectively.

	$N_A : N_C : N_{CA}$ = 0% : 10% : 0%					
	iris	pima	zoo	ionosphere	letter	leaf
OneClassSVM	0.500±0.080	0.505±0.035	0.479±0.093	0.533±0.047	0.496±0.025	0.497±0.043
HOAD	0.633±0.100	0.514±0.040	0.574±0.073	0.449±0.054	0.576±0.022	0.510±0.041
AP	**0.966±0.030**	0.492±0.037	**0.933±0.045**	**0.943±0.024**	0.831±0.014	0.738±0.058
MLRA	0.849±0.061	0.664±0.029	0.642±0.097	0.801±0.045	0.653±0.021	**0.821±0.045**
LDSR	0.756±0.121	0.627±0.031	0.829±0.064	0.834±0.025	0.758±0.024	0.705±0.058
MODDIS	0.819±0.104	0.694±0.030	0.782±0.096	0.784±0.034	0.852±0.028	0.711±0.048
NCMOD	0.595±0.115	0.509±0.030	0.849±0.057	0.836±0.034	0.524±0.028	0.695±0.077
SRLSP	0.946±0.043	**0.748±0.030**	0.887±0.054	0.922±0.020	**0.925±0.014**	0.810±0.022

Table 6. The comparison result on the UCI datasets. AUC values (mean ± standard deviation) are reported, and the best and the second-best results are in **bold** and underline, respectively.

	$N_A : N_C : N_{CA}$ = 0% : 0% : 10%					
	iris	pima	zoo	ionosphere	letter	leaf
OneClassSVM	0.503±0.064	0.508±0.031	0.514±0.093	0.555±0.045	0.506±0.021	0.510±0.047
HOAD	0.446±0.150	0.357±0.081	0.746±0.021	0.429±0.065	0.242±0.117	0.717±0.141
AP	0.952±0.022	0.436±0.243	0.851±0.067	**0.905±0.018**	0.785±0.021	0.828±0.043
MLRA	0.878±0.039	0.744±0.023	0.814±0.054	0.725±0.037	0.651±0.032	0.989±0.013
LDSR	0.926±0.038	0.947±0.015	0.879±0.044	0.785±0.022	0.957±0.007	**1.000±0.000**
MODDIS	0.866±0.076	0.711±0.085	0.856±0.054	0.737±0.016	0.821±0.044	0.999±0.002
NCMOD	0.958±0.034	**0.982±0.011**	0.908±0.047	0.770±0.025	**0.999±0.001**	0.976±0.012
SRLSP	**0.981±0.013**	0.809±0.038	**0.930±0.026**	0.795±0.015	0.906±0.015	0.990±0.004

Table 7. The comparison result on the real multi-view datasets. AUC values (mean ± standard deviation) are reported, and the best and the second-best results are in **bold** and underline, respectively.

	MSRC-v1	AWA-10	Caltech-7
OneClassSVM	0.680±0.042	0.800±0.024	0.789±0.016
HOAD	0.374±0.158	0.351±0.119	0.356±0.132
AP	0.508±0.030	0.455±0.024	0.454±0.020
MLRA[1]	-	-	-
LDSR	<u>0.973±0.013</u>	0.733±0.060	0.943±0.032
MODDIS	0.957±0.013	0.813±0.029	<u>0.946±0.008</u>
NCMOD	0.935±0.019	<u>0.901±0.023</u>	0.933±0.011
SRLSP	**0.985±0.006**	**0.938±0.010**	**0.976±0.004**

[1] MLRA can only run on the dataset with equal features dimension on each view.

high-quality performance in the class-outliers-only setting, but it fails to detect attribute outliers. Compared with the state-of-the-art neural network-based multi-view outlier detection method NCMOD, the proposed SRLSP has better performance. It must be emphasized that the SRLSP is not a neural network-based method, but it can achieve the same level as the most advanced neural network-based method. MLRA and LDSR are the competitive methods with SRLSP, which are all self-representation methods, and the comprehensive performance of the proposed SRLSP is significantly better than MLRA, as well as is close to or even better than LDSR in most cases. However, the proposed SRLSP has a very small time overhead, which is more competitive in real-life scenarios.

6.3.2 Comparison Results on Real Multi-View Datasets. Table 7 shows the comparison result on three real multi-view datasets. AP and HOAD have the worst performance, which further illustrates the limitation of pair-wise constraint. OneClassSVM is a single-view method, so its performance is worse than the regular multi-view methods, i.e., LDSR, MODDIS, NCMOD, and the proposed SRLSP, which shows the necessity of designing the specialized multi-view method. Therefore, the proposed SRLSP is more suitable for multi-view outlier detection in practical scenarios regardless of the views number and features dimensions.

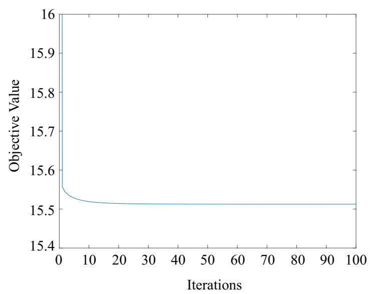
The convergence curves of the proposed SRLSP on these three datasets are shown in Figure 4. The optimization algorithm can converge within one hundred iterations in most cases, but in some complex datasets, it needs hundreds of iterations.

6.4 Ablation Experiment

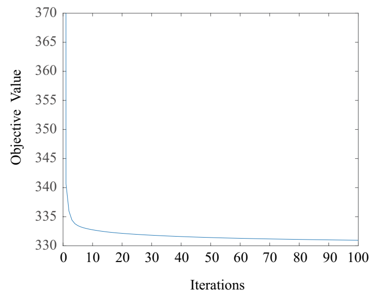
In this subsection, we discuss the two submodels in Subsection 3.1 and 3.2, which are denoted as SR (The Neighbor-based Self-Representation Submodule) and ASL (Adaptive Similarity Learning Submodule with Graph Fusion) for short. As presented in Table 8, compared with the ASL submodule, the SR submodule can detect attribute outliers more effectively, while the ASL submodule is better at detecting category outliers that exhibit inconsistent behaviors from views. And the proposed SRLSP effectively fuses these two sub-modules, achieving significant performance improvements on all types of multi-view outlier detection.

6.5 Parameter Analysis

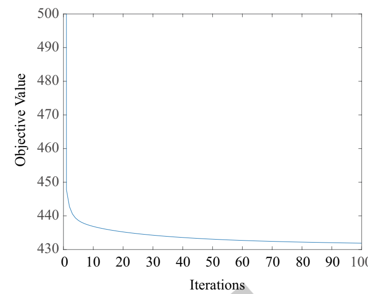
There are four parameters in the proposed SRLSP, i.e., μ, k, λ and γ. The parameter μ is not adjusted in most cases, so we mainly discuss parameters k, λ and γ. Figure 5(a) shows the evaluation result on adjusting k and Figure 5(b) shows the evaluation result on adjusting λ and γ. Fortunately, these parameters can be easily adjusted



(a) MSRC-v1



(b) AWA-10



(c) Caltech-7

Fig. 4. The convergence curve on real multi-view datasets.

Table 8. The comparison result on the pima dataset. "A", "C" and "CA" denoted the dataset contains only 10% attribute outliers, 10% class outliers, and 10% class-attribute outliers respectively. The best results are in **bold**

	pima (A)	pima (C)	pima (CA)
SR	0.973 ± 0.004	0.715 ± 0.030	0.598 ± 0.036
ASL	0.777 ± 0.018	0.729 ± 0.029	0.738 ± 0.033
SRLSP (SR + ASL)	**0.990 ± 0.002**	**0.748 ± 0.030**	**0.809 ± 0.038**

to the suitable range. The main reason for the poor performance is that most of the outlier_scores are close to 0 because of the over-strong capacity of the proposed model, or most of the outlier_scores are far above 0 because of the over-weak capacity of the proposed model. Therefore, all parameters can be easily adjusted by analyzing the distribution of outlier_scores.

6.6 Runtime Analysis

In this subsection, we discuss the relationship between the dataset sizes and algorithm runtimes. The stop criterion is defined as follows:

$$\frac{|\mathrm{obj}^{t+1} - \mathrm{obj}^{t}|}{|\mathrm{obj}^{t}|} < 10^{-2}, \tag{19}$$

where obj^t is the objective value in the t-th iteration. Figure 6 shows the comparison result, it can be seen that the proposed SRLSP has nearly linear time complexity and is more suitable to be applied to large-scale datasets.

7 CONCLUSION

In this paper, we propose a novel self-representation method with local similarity preserving, which uses the local similarity information to improve the learned common cross-view similarity consensus. The well-designed objective function and outlier measurement criterion ensure the proposed method can calculate each instance with $O(NlogN)$ time complexity parallelly. Extensive experimental results verify the superiority of the proposed method on both performance and time complexity.

The method proposed in this paper effectively solves the problem of online detection of multi-view outliers and inspires future research to focus on the online detection problem of outlier detection. There are many possible directions for future work, including improving the proposed method with fewer parameters, associating the

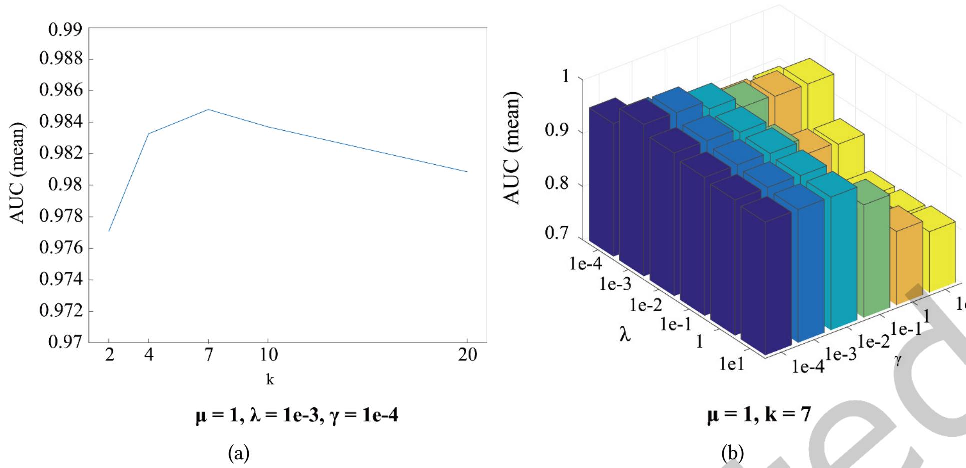


Fig. 5. Analytical experiments on the MSRC-v1 dataset. (a) The AUC values with different values of k. (b) The AUC values with different values of λ and γ.

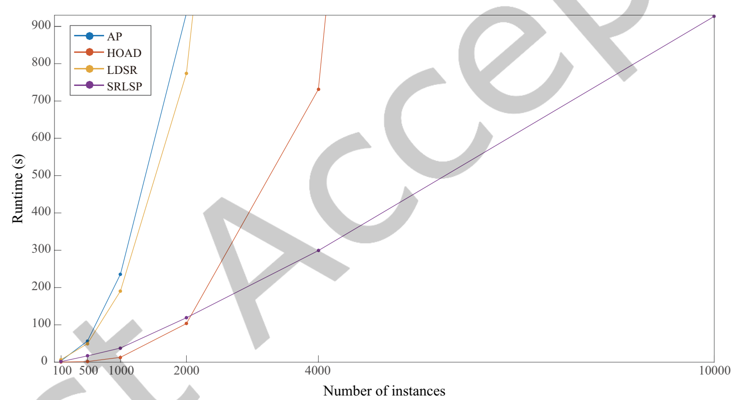


Fig. 6. The runtimes of different multi-view algorithms on the AWA dataset. MLRA can not run on this dataset as well as its runtime performance is similar to LDSR.

proposed method with an autoencoder to handle high dimensional datasets, etc. Some other applications of single-view outlier detection in neural networks will also be considered.

ACKNOWLEDGMENTS

The research is supported by the Key-Area Research and Development Program of Guangdong Province (2020B010165003), the National Natural Science Foundation of China (62176269), the Guangdong Basic and Applied Basic Research Foundation (2019A1515011043).

REFERENCES

[1] Charu C Aggarwal. 2015. Data mining. Springer.

[2] Mohiuddin Ahmed and Abdun Naser Mahmood. 2013. A novel approach for outlier detection and clustering improvement. In Proceedings of the IEEE Conference on Industrial Electronics and Applications (ICIEA). 577–582.
[3] Emin Aleskerov, Bernd Freisleben, and Bharat Rao. 1997. CARDWATCH: a neural network based database mining system for credit card fraud detection. In Proceedings of the IEEE/IAFE Computational Intelligence for Financial Engineering (CIFEr). 220–226.
[4] Fabrizio Angiulli, Stefano Basta, and Clara Pizzuti. 2005. Distance-based detection and prediction of outliers. IEEE Transactions on Knowledge and Data Engineering (TKDE) 18, 2 (2005), 145–160.
[5] Irad Ben-Gal. 2005. Outlier detection. In Data mining and knowledge discovery handbook. Springer, 131–146.
[6] Kamal Berahmand, Mehrnoush Mohammadi, Azadeh Faroughi, and Rojiar Pir Mohammadiani. 2021. A novel method of spectral clustering in attributed networks by constructing parameter-free affinity matrix. Cluster Computing (2021), 1–20.
[7] Kamal Berahmand, Elahe Nasiri, Yuefeng Li, et al. 2021. Spectral clustering on protein-protein interaction networks via constructing affinity matrix using attributed graph embedding. Computers in Biology and Medicine 138 (2021), 104933.
[8] Roi Blanco and Christina Lioma. 2012. Graph-based term weighting for information retrieval. Information retrieval 15, 1 (2012), 54–92.
[9] Mohamed Bouguessa. 2015. A practical outlier detection approach for mixed-attribute data. Expert Systems with Applications 42, 22 (2015), 8637–8649.
[10] Stephen Boyd and Lieven Vandenberghe. 2004. Convex optimization. Cambridge university press.
[11] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. 93–104.
[12] Xiao Cai, Feiping Nie, and Heng Huang. 2013. Multi-view k-means clustering on big data. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI). 2598–2604.
[13] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. 2011. Robust principal component analysis? Journal of the ACM (JACM) 58, 3 (2011), 1–37.
[14] Chuan Chen, Yu Wang, Weibo Hu, and Zibin Zheng. 2020. Robust multi-view k-means clustering with outlier removal. Knowledge-Based Systems (KBS) 210 (2020), 106518.
[15] Zitai Chen, Chuan Chen, Zong Zhang, Zibin Zheng, and Qingsong Zou. 2019. Variational graph embedding and clustering with laplacian eigenmaps. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI). 2144–2150.
[16] Li Cheng, Yijie Wang, and Xinwang Liu. 2021. Neighborhood consensus networks for unsupervised multi-view outlier detection. (2021).
[17] Miaomiao Cheng, Liping Jing, and Michael K Ng. 2020. Robust unsupervised cross-modal hashing for multimedia retrieval. ACM Transactions on Information Systems (TOIS) 38, 3 (2020), 1–25.
[18] Ehsan Elhamifar and René Vidal. 2013. Sparse subspace clustering: algorithm, theory, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 35, 11 (2013), 2765–2781.
[19] Guojun Gan and Michael Kwok-Po Ng. 2017. K-means clustering with outlier removal. Pattern Recognition Letters 90 (2017), 8–14.
[20] Jing Gao, Wei Fan, Deepak Turaga, Srinivasan Parthasarathy, and Jiawei Han. 2011. A spectral framework for detecting inconsistency across multi-source object relationships. In Proceedings of the IEEE International Conference on Data Mining (ICDM). 1050–1055.
[21] Amol Ghoting, Matthew Eric Otey, and Srinivasan Parthasarathy. 2004. Loaded: Link-based outlier and anomaly detection in evolving data sets. In Fourth IEEE International Conference on Data Mining (ICDM'04). IEEE, 387–390.
[22] Douglas M Hawkins. 1980. Identification of outliers. Springer.
[23] Vandana P Janeja and Revathi Palanisamy. 2013. Multi-domain anomaly detection in spatial datasets. Knowledge and Information Systems (KAIS) 36, 3 (2013), 749–788.
[24] Yu-Xuan Ji, Ling Huang, Heng-Ping He, Chang-Dong Wang, Guangqiang Xie, Wei Shi, and Kun-Yu Lin. 2019. Multi-view outlier detection in deep intact space. In Proceedings of the IEEE International Conference on Data Mining (ICDM). 1132–1137.
[25] Zhao Kang, Guoxin Shi, Shudong Huang, Wenyu Chen, Xiaorong Pu, Joey Tianyi Zhou, and Zenglin Xu. 2020. Multi-graph fusion for multi-view spectral clustering. Knowledge-Based Systems (KBS) 189 (2020), 105102.
[26] Anna Koufakou and Michael Georgiopoulos. 2010. A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes. Data Mining and Knowledge Discovery 20, 2 (2010), 259–289.
[27] Vipin Kumar. 2005. Parallel and distributed computing for cybersecurity. IEEE Distributed Systems Online 6, 10 (2005), 1–9.
[28] Kai Li, Sheng Li, Zhengming Ding, Weidong Zhang, and Yun Fu. 2018. Latent discriminant subspace representations for multi-view outlier detection. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI). 3522–3529.
[29] Sheng Li, Ming Shao, and Yun Fu. 2018. Multi-view low-rank analysis with applications to outlier detection. ACM Transactions on Knowledge Discovery from Data (TKDD) 12, 3 (2018), 1–22.
[30] Youwei Liang, Dong Huang, and Chang-Dong Wang. 2019. Consistency meets inconsistency: A unified graph learning framework for multi-view clustering. In Proceedings of the IEEE International Conference on Data Mining (ICDM). 1204–1209.
[31] Bo Liu, Yanshan Xiao, S Yu Philip, Zhifeng Hao, and Longbing Cao. 2013. An efficient approach for outlier detection with imperfect data labels. IEEE Transactions on Knowledge and Data Engineering (TKDE) 26, 7 (2013), 1602–1616.
[32] Hongfu Liu, Jun Li, Yue Wu, and Yun Fu. 2021. Clustering with outlier removal. IEEE Transactions on Knowledge and Data Engineering (TKDE) 33, 6 (2021), 2369–2379.

[33] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. 2012. Robust and efficient subspace segmentation via least squares regression. In Proceedings of the 12th European Conference on Computer Vision (ECCV), Vol. 7578. 347–360.

[34] Alejandro Marcos Alvarez, Makoto Yamada, Akisato Kimura, and Tomoharu Iwata. 2013. Clustering-based anomaly detection in multi-view data. In Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM). 1545–1548.

[35] Feiping Nie, Jing Li, Xuelong Li, et al. 2017. Self-weighted Multiview Clustering with Multiple Graphs. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI). 2564–2570.

[36] Feiping Nie, Xiaoqian Wang, and Heng Huang. 2014. Clustering and projected clustering with adaptive neighbors. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). 977–986.

[37] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. ACM Computing Surveys (CSUR) 54, 2 (2021), 1–38.

[38] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290, 5500 (2000), 2323–2326.

[39] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. 2021. A unifying review of deep and shallow anomaly detection. Proc. IEEE (2021).

[40] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. 2020. Deep semi-supervised anomaly detection. In International Conference on Learning Representations. https://openreview.net/forum?id=HkgH0TEYwH

[41] Felix Sattler, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. 2020. On the byzantine robustness of clustered federated learning. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 8861–8865.

[42] Bernhard Schölkopf, Robert C Williamson, Alexander J Smola, John Shawe-Taylor, John C Platt, et al. 1999. Support vector method for novelty detection. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Vol. 12. 582–588.

[43] Xiang-Rong Sheng, De-Chuan Zhan, Su Lu, and Yuan Jiang. 2019. Multi-view anomaly detection: neighborhood in locality matters. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI). 4894–4901.

[44] Clay Spence, Lucas Parra, and Paul Sajda. 2001. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. In Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA). 3–10.

[45] Jingjing Tang, Yingjie Tian, Xiaohui Liu, Dewei Li, Jia Lv, and Gang Kou. 2018. Improved multi-view privileged support vector machine. Neural Networks 106 (2018), 96–109.

[46] Chu Wang, Yan-Ming Zhang, and Cheng-Lin Liu. 2018. Anomaly detection via minimum likelihood generative adversarial networks. In Proceedings of the 24th International Conference on Pattern Recognition (ICPR). 1121–1126.

[47] Hua Wang, Feiping Nie, and Heng Huang. 2013. Multi-view clustering and feature learning via structured sparsity. In Proceedings of the 30th International Conference on Machine Learning (ICML), Vol. 28. 352–360.

[48] Hao Wang, Yan Yang, Bing Liu, and Hamido Fujita. 2019. A study of graph-based system for multi-view clustering. Knowledge-Based Systems (KBS) 163 (2019), 1009–1019.

[49] Zhiyue Wu, Hongzuo Xu, Guansong Pang, Fengyuan Yu, Yijie Wang, Songlei Jian, and Yongjun Wang. 2021. Dram failure prediction in aiops: Empirical evaluation, challenges and opportunities. arXiv preprint arXiv:2104.15052 (2021).

[50] Xijiong Xie and Shiliang Sun. 2019. Multi-view support vector machines with the consensus and complementarity information. IEEE Transactions on Knowledge and Data Engineering (TKDE) 32, 12 (2019), 2401–2413.

[51] Chang Xu, Dacheng Tao, and Chao Xu. 2013. A survey on multi-view learning. arXiv preprint arXiv:1304.5634 (2013).

[52] Hongzuo Xu, Yijie Wang, Yongjun Wang, and Zhiyue Wu. 2019. Mix: A joint learning framework for detecting both clustered and scattered outliers in mixed-type data. In 2019 IEEE International Conference on Data Mining (ICDM). IEEE, 1408–1413.

[53] Jiawei Yang, Susanto Rahardja, and Pasi Fränti. 2021. Mean-shift outlier detection and filtering. Pattern Recognition 115 (2021), 107874.

[54] Fanghua Ye, Zhiwei Lin, Chuan Chen, Zibin Zheng, Hong Huang, and Emine Yilmaz. 2020. Outlier resilient collaborative web service QoS prediction. arXiv preprint arXiv:2006.01287 (2020).

[55] Amir Zadeh, Paul Pu Liang, Navonil Mazumder, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2018. Memory fusion network for multi-view sequential learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI). 5634–5641.

[56] Handong Zhao and Yun Fu. 2015. Dual-regularized multi-view outlier detection. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI). 4077–4083.

[57] Handong Zhao, Hongfu Liu, Zhengming Ding, and Yun Fu. 2017. Consensus regularized multi-view outlier detection. IEEE Transactions on Image Processing (TIP) 27, 1 (2017), 236–248.

[58] Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun. 2017. Multi-view learning overview: Recent progress and new challenges. Information Fusion 38 (2017), 43–54.