A Comprehensive Survey on Multi-View Clustering

Uno Fang, *Student Member, IEEE*, Man Li, Jianxin Li, *Senior Member, IEEE*, Longxiang Gao, *Member, IEEE*, Tao Jia, *Member, IEEE*, and Yanchun Zhang

***Abstract*—The development of information gathering and extraction technology has led to the popularity of multi-view data, which enables samples to be seen from numerous perspectives. Multi-view clustering (MVC), which groups data samples by leveraging complementary and consensual information from several views, is gaining popularity. Despite the rapid evolution of MVC approaches, there has yet to be a study that provides a full MVC roadmap for both stimulating technical improvements and orienting research newbies to MVC. In this article, we review recent MVC techniques with the purpose of exhibiting the concepts of popular methodologies and their advancements. This survey not only serves as a unique MVC comprehensive knowledge for researchers but also has the potential to spark new ideas in MVC research. We summarise a large variety of current MVC approaches based on two technical mechanisms: heuristic-based multi-view clustering (HMVC) and neural network-based multi-view clustering (NNMVC). We end with four technological approaches within the category of HMVC: nonnegative matrix factorisation, graph learning, latent representation learning, and tensor learning. Deep representation learning and deep graph learning are two technical methods that we demonstrate in NNMVC. We also show 15 publicly available multi-view datasets and examine how representative MVC approaches perform on them. In addition, this study identifies the potential research directions that may require further investigation in order to enhance the further development of MVC.**

***Index Terms*—Multi-view clustering, graph learning, nonnegative matrix factorisation, deep representation learning, contrastive learning.**

I. Introduction

CLUSTERING is a critical technique for unsupervised learning, which is essential in exploring feature patterns. Clustering can be used for various real-world data mining tasks, including social network analysis, gene expression analysis, medical effect analysis, etc. Especially, internet and communications technologies are rapidly advancing, making it easier to access and collect data via a variety of sources. In such scenarios, the collected data often contains multiple views of information. For instance, an image can be described by a set of features, including colour, histogram, texture, etc. Each feature represents a view of the data, and these different views can complement each other mutually. Thus, the multiple views of information can be integrated to bring significant benefits for unsupervised data clustering, where the underlying information embedded in the data is well exploited to improve the quality of clustering. Therefore, in recent years, multi-view Clustering (MVC) has been increasingly researched and applied to real-world scenarios, such as community detection in social networks, image annotation and recognition in computer vision, and cross-domain user modelling in recommendation systems.

Fig. 1 shows the general procedure of MVC, which firstly initialises the input images and extracts features, then clusters data according to extracted features, and acquires final clusters at the end. To handle the text data with two independent views, Bickel and Scheffer [1] applied the classic k-means and expectation maximisation (EM) clustering methods to the multi-view setting. The simplest way to exploit information from multi-view data is to concatenate all view-specific data features and then conduct classic clustering methods, such as k-means. However, this method often treats all the views equally when clustering, and does not discriminate the importance of different views. This could negatively affect the final clustering result. To maximise the utilisation of multi-view data, the later studies of MVC tend to cluster data of each view simultaneously, then integrate the clustering results according to view-based importance towards the clustering task (i.e., assigning each view a weight), such as Nie et al. [2], Wang et al. [3], they explored a Laplacian rank constrained graph for calculating the confidence of each view-based built graph in a self-weighted manner. To appropriately deal with the multi-view data and enhance the performance of MVC, all the existing research works have to address the below common technical issues:

- Constructing an effective and easily-processable consensus of multiple views efficiently;
- Assembling the multiple clustering results naturally;
- Discriminating the view-specific uniqueness for contrastive information;
- Learn how to weigh different views according to their respective importance towards clustering.

Besides the above common technical issues to be addressed in MVC, there are many other existing MVC works that focused on different lines of challenges in MVC research.

- *Underlying information utilisation:* The basic idea of Multi-view Clustering (MVC) is to maximise the use of

Manuscript received 9 July 2022; revised 16 January 2023; accepted 20 April 2023. Date of publication 25 April 2023; date of current version 8 November 2023. This work was supported by Deakin University-Southwest University Joint Research Centre on Big Data. Recommended for acceptance by B. C. M. Fung. *(Uno Fang and Man Li are co-first authors.) (Corresponding author: Jianxin Li.)*

Uno Fang, Man Li, and Jianxin Li are with the School of IT, Faculty of Science, Engineering and Built Environment, Deakin University, Geelong, VIC 3220, Australia (e-mail: uno.fang@deakin.edu.au; man.li@deakin.edu.au; jianxin.li@deakin.edu.au).

Longxiang Gao is with the Shandong Academy of Sciences, Shandong Computer Science Center National Supercomputer Center in Jinan, Qilu University of Technology, Jinan, Shandong 250316, China (e-mail: gaolx@sdas.org).

Tao Jia is with the School of Information and Technology, Southwest University, Chongqing 400700, China (e-mail: tjia@swu.edu.cn).

Yanchun Zhang is with the New Cyber Research Department, Peng Cheng Laboratory China, Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, Guangdong 510006, China (e-mail: yzhangvu@gmail.com).

Digital Object Identifier 10.1109/TKDE.2023.3270311

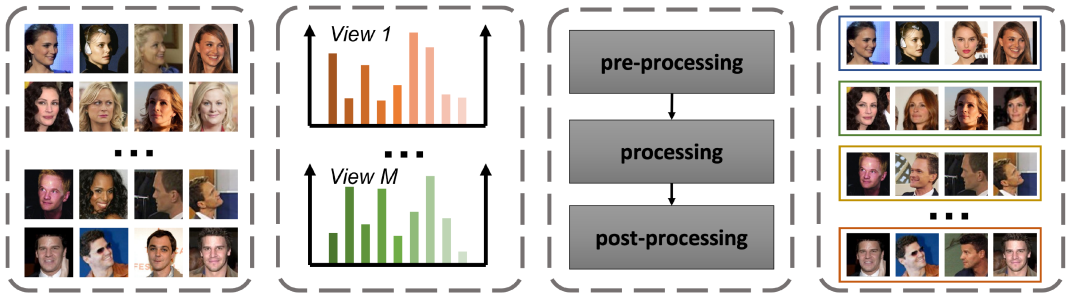


Fig. 1. General procedures of multi-view clustering (MVC). Fig. 1c provides three general steps of the clustering procedure instead of clustering details for two reasons: (1) different MVC methods cannot be generalised further; (2) technical innovations (i.e., method novelty) happen in these three steps. More details of each method will be demonstrated in Sections III and IV.

different data views. Therefore, extracting and exploiting underlying information is the main research focus of most MVC works. MVC works [4], [5], [6], [7], [8], [9], [10], [11] remarkably intensify the embedding reliability from various descriptor views for the downstream clustering tasks. To induce the discriminative improvement of features, RHMC [10] constrains the learned multi-view representations in the latent space by considering the proposed hybrid mutual information estimation. However, even though different feature views of an image are generated by different image descriptors, there could be highly overlapped features in these different views leading to inefficient representation learning. Also, compared to single-view clustering, the increase in data amount leads to a higher computational cost.

- *Large-scale clustering:* This has been a challenge since the occurrence of MVC. Clustering multi-view data results in multiple times higher computation costs than clustering single-view data. It will be a disaster when it comes to clustering large-scale multi-view data. To address the problem of large-scale MVC, some early methods are proposed in [12], [13], [14], [15], [16], which approximate multi-view data into several similar matrices. The development of MVC that can efficiently cluster large-scale data is therefore imperative, especially in light of today's explosion of data. Innovatively, [17] divides large-scale multi-view data into sub-blocks, then performs feature selection using a sharing sub-model for the general additive feature selection model. However, how to guarantee accuracy with increasing efficiency has been the pain point of large-scale MVC.
- *Incomplete or partial clustering:* Due to the complexity of data collection and transmission, some views of the data points might be missed in practice, leading to the incomplete or partial multi-view clustering problem (IMC) [18], [19], [20], [21], [22], [23], [24]. For instance, some video frames may lose their audio or visual signal during online meetings due to malfunctioning sensors. This unbalanced relationship between complete and incomplete views could result in poor performance when weighing their importance to clustering. Moreover, the incomplete views and the rest views that are complete would impact each other mutually during clustering. For instance, HCP-IMSC [24] constructs a hypergraph with a tensor factorisation regularisation in a self-weighted manner from the incomplete multi-view data, then reconstructs partial views using the hypergraph-induced hyper-Laplacian regularisation. Fortunately, we can leverage complete views to reconstruct the views with partial information via their mutual influence.
- *Noise and outlier reduction:* Real-world data inevitably contain outliers, which will degrade clustering performance. Early works [25], [26], [27], [28] construct transition probability matrices for each view as a result of combining information from multi-view data while considering the noise. Then, they create a standard low-rank transition probability matrix across various views using these constructed view-specific probability matrices. As a result of using the shared probability matrix as an essential input, the Markov chain method obtains the final clustering result. Recently, ECMC [29] generates embedded anchor graphs and leverages nonnegative matrix factorisation (NMF) while exploring correntropy to reduce noise. Yuan et al. [30] devised a sum-of-norm regularisation for determining the cluster number and employs robust statistics techniques against outliers.

Furthermore, there is an increasing number of works that attempt to address multiple tasks, such as [31], [32], [33], [34]. These research tasks not only focus on the view consistency information but also consider the uniqueness of view-specific information, which brings more research potential to MVC.

Earlier MVC surveys [35], [36], [37], [38], [39] have introduced several important aspects of MVC from different perspectives. Yang and Wang [35] illustrated how a data object is embedded into a latent data space, also demonstrated several general MVC frameworks in different styles, such as co-training style and multi-kernel style. Fu et al. [36] concluded MVC models by three categories, which are graph-based model, space-learning-based model and Binary-code-learning-based model. Chao et al. [37] proposed a novel taxonomy of MVC methods based on their different view integration procedures, i.e., common eigenvector matrix, common coefficient matrix, common indicator matrix, direct combination and combination after projection. Chen et al.

[38] took deep learning-based MVC methods into consideration to enhance the review comprehensiveness, where they built 4-layer hierarchical taxonomy to divide MVC methods from the perspective of representation learning in a binary style. Wen et al. [39] targeted and made an in-depth comparative analysis on incomplete multi-view clustering (IMC). However, in this work, we will present a more comprehensive discussion about the diversified and representative studies in MVC since we realised that existing reviews have the following shortcomings:

1) In [35], the proposed taxonomy of MVC methods is not explicit and fair. For example, spectral clustering is a broader topic than other categories (e.g., multi-view graph clustering). In addition, some categories indicate strategies instead of methods, such as co-training style and multi-task learning.
2) Three MVC categories that Fu et al. [36] introduced are intuitively basic and classic but not thorough. We can apparently further classify within these three categories for a more comprehensive understanding of MVC.
3) Chao et al. [37] summarised MVC methods into two categories of generative learning and contrastive learning. This is unique but unbalanced because most MVC methods use contrastive strategies. Also, when looking into their taxonomy, we found they excluded MVC methods using deep neural networks (DNNs).
4) Chen et al. [38] proposed 4-layer binary taxonomy, which missed to demonstrate the diversity of non-representation learning-based MVC, deep learning-base MVC, multi-view graph clustering and multi-view subspace clustering. Also, they did not follow the hierarchy of their proposed taxonomy to present MVC methods in their following sections.

Therefore, in a more comprehensive way to build the MVC roadmap, we first summarise MVC methods into two main categories: heuristic-based multi-view clustering (HMVC) methods and neural network-based multi-view clustering (NNMVC). In HMVC, we review nonnegative matrix factorisation, graph learning, latent representation learning, and tensor learning. In NNMVC, we introduce deep representation learning and deep graph learning (see Fig. 2). After familiarising MVC with the above-mentioned six methods in HMVC and NNMVC, readers will be able to seize the core mechanism ideas of MVC. Crucially, after demonstrating the theoretical methodologies of these six MVC frameworks, we will discuss the state-of-the-art technical contributions, which include input construction, regularisation, self- or auto-weighted measure and contrastive strategies. This aims to inspire researchers in the field of MVC to explore the potential of novel studies from technical perspectives.

Organisation: We provide notations and preliminaries in Section II. Section III demonstrates four HMVC methods, while Section IV introduces two NNMVC methods. To explore in-depth insights of MVC, we discuss the advancements of the state-of-the-art MVC methods in Section V. In Section VI, we introduce the most popular open multi-view datasets and summarise the experimental results of these methods. Finally, Section VII concludes this survey with a discussion of potential research directions in MVC research.

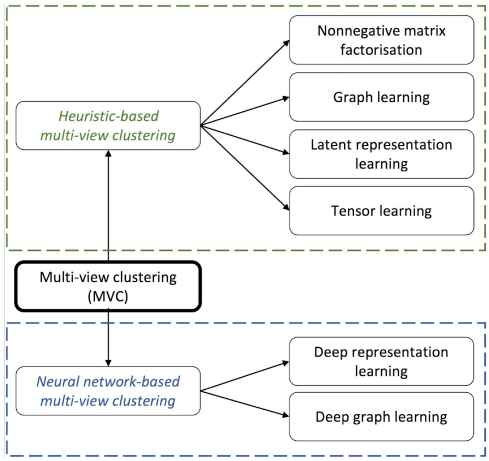


Fig. 2. The overview of our taxonomy on multi-view clustering methods .

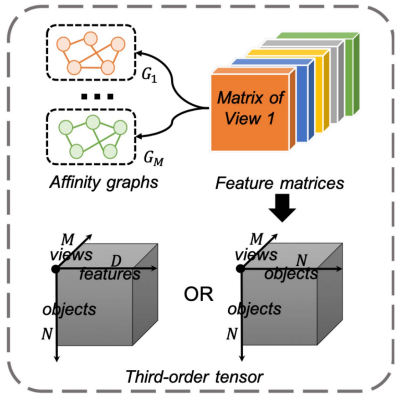


Fig. 3. The common input construction of multi-view clustering. Note that, when processing a homogeneous graph, it is also can be represented as two matrices, a feature matrix and an adjacency matrix.

II. Notations and Preliminaries

Suppose there is a dataset denoted as $\boldsymbol{V} = \{v_i\}_{i=1}^{N}$ and its feature vector set denoted as $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$, where N denotes the number of data samples. We assume the dataset has M views, then each node v_i can be transformed as $\{v_i^m\}_{m=1}^{M}$, i.e., in m-th view, v_i^m has the feature vector $\boldsymbol{x}_i^m$. The frequently used notations are summarised in Table I. In addition, we visualise the typical input construction of multi-view clustering (MVC) for intuitive understanding (see Fig. 3). Different ways of input construction will lead to different MVC frameworks. However, on the top of all methods, *multi-view spectral clustering* [25] and *multi-view subspace clustering* [40] are the fundamentals of MVC. Also, we introduce the basic definitions of *graph construction*, *tensor construction* and *regularisation*, which are the common base of MVC.

Multi-view spectral clustering makes use of the information contained in multiple graphs to learn its latent structure. Here we represent the main body of multi-view spectral clustering [41],

TABLE I
NOTATIONS AND DESCRIPTIONS

notations	descriptions
N	the total number of instances
M	the number of views
D^m	the dimensionality of the m-th view data
X	$X = \{X^1, \ldots, X^M\} \in \mathbb{R}^{\sum_{m=1}^{M} D^m \times N}$, M feature vector sets
X^m	$X^m = \{x_1^m, \ldots, x_N^m\}$, Feature vector set in the m-th view
x_i^m	$x_i^m = \{x_{ij}^m\}_{j=1}^{D_m}$, the feature vector of i-th instance in the m-th view
$\mathbf{1}$	a vector with all ones
S	$\{S^m\}_{m=1}^M = \{s_{ij}^m\}_{i=1,j=1,m=1}^{N,N,M}$, a similarity graph matrix
s_{ij}^m	the similarity score betweem v_i and v_j in the m-th view
F	a vector matrix, each column of F represents a basic vector
U	a consensus graph matrix
L_S	the Laplacian matrix of a matrix S
$\mathbf{I}$	an identity matrix, when the multiplication of two matrices equals to $\mathbf{I}$, they are called orthogonal
$G(V, E)$	a weighted homogeneous affinity graph constructed from X
$\mathbb{G}$	a graph embedding of G
$\mathbf{S}^m$	the signature matrix of m-th view
$\mathrm{Tr}(\cdot)$	a function to calculate the trace of a matrix
$\mathrm{Nor}(\cdot)$	a function of normalisation
$J(\cdot, \cdot)$	a loss function defined in each single view
$\mathrm{Reg}(\cdot)$	an regularisation term
$\mathcal{L}_h(\cdot, \cdot)$	the loss functions associated with the latent (hidden) representation
$\mathcal{L}_r(\cdot, \cdot)$	the loss functions associated with the data reconstruction
$\|\cdot\|_2$	ℓ_2-norm of a column vector
$\|\cdot\|_1$	ℓ_1-norm
$\|\cdot\|_{2,1}$	$\ell_{2,1}$-norm
$\|\cdot\|_F$	ℓ_F, Frobenius norm
$\|\cdot\|_*$	ℓ_*, nuclear norm
$\mathrm{Tr}(\cdot)$	the trace of a matrix
$\mathrm{diag}(\cdot)$	the column vector composed of diagonal elements of a square matrix
$\mathrm{rank}(\cdot)$	a ranking function
λ	$\lambda = \{\lambda_1, \ldots, \lambda_M\}$, which tunes the relative importance of different factors and the contribution of each factor to the overall result
γ	a positive regularisation parameter
τ_1, τ_2	a pair of positive balance parameters

[42] to be formulated as:

$$\min_{\mathbb{G},\mathbb{A}} \sum_{m=1}^{M} \alpha^m J\left(\mathbb{G}, W^m\right) + \gamma \,\mathrm{Reg}(\mathbb{A}),$$
$$\text{s.t.} \quad \mathbb{G}^T\mathbb{G} = \boldsymbol{I}, \mathbb{A}^T\mathbf{1} = 1, \mathbb{A} \geq \mathbf{0}, \tag{1}$$

where W^m is the similarity matrix for the m-th view. $\mathbb{G} \in \mathbb{R}^{N\times K}$ is the graph embedding across all the views, where K is the expected number of clusters. $\mathbb{G}^T$ is the transpose of $\mathbb{G}$. $\mathbb{A} = [\alpha^1, \alpha^2, \ldots, \alpha^m] \in \mathbb{R}^m$ is the weight vector set for discriminating different views. Benefiting from spectral clustering (for single-view data), $J(\mathbb{G}, W^m) = \mathrm{Tr}(\mathbb{G}^T L_W^m \mathbb{G})$ materialises the loss function in m-th view where L_W^m is the Laplacian matrix of W^m. Since the weights depend on the loss value of each view, a regularisation term $\mathrm{Reg}(\cdot)$ is required to smooth the weights, avoiding the trivial solution where the best views are assigned with 1. Accordingly, γ is inevitable to balance the regularisation.

Multi-view subspace clustering combines each data point linearly with others, and minimises the reconstruction loss to obtain the combination coefficient [43]. This coefficient is measured by using the similarity between the corresponding points. It solves the following problem:

$$\min_{S^m} \sum_{m=1}^{M} \|X^m - X^m S^m\|_F^2 + \gamma \,\mathrm{Reg}\left(S^m\right),$$
$$\text{s.t. } S^m \geq 0, S^m\mathbf{1} = \mathbf{1}, \tag{2}$$

where S^m indicates the similarity matrix of m-th view, which has size of $N \times N$, and $\gamma > 0$, and $\mathbf{1}$ is a vector with all ones. The constraints expect that all S_{ij} is nonnegative and $\sum_j S_{ij} = 1$. It is obvious that the similarity graph matrix S has size of $NM \times NM$, which faces a significant challenge concerning scalability for large data sets. With modifying the form the regularisation function $\mathrm{Reg}(\cdot)$, (2) can provide different solutions.

Graph construction is an essential step for MVC methods that involve graph learning. To construct a graph $G(V, E)$ from a feature vector set X of a dataset (see Fig. 3), the intuitive way is to consider each data point as a vertex v_i (i.e., the feature of v_i is x_i) and fully connect all vertices as a fully connected graph, where e_{ij} indicates each edge between v_i and v_j. The essential step differing each vertex from others is to calculate pair-wise similarities. The edge weight w_{ij} of e_{ij} is defined as:

$$w_{i,j} = ||x_i - x_j||^2. \tag{3}$$

However, the above method would result in the high computational cost because it needs to calculate the distance computation of all N^2 pairs of vertices . Therefore, researchers tend to utilise K nearest neighbours (KNN) or KD-tree to create affinity graphs, which links each node to its top-k nearest neighbouring nodes to form an graph affinity.

Tensor construction enables data being processed as multi-dimensional matrices or multi-way arrays [44]. This is because a third-order tensor can be seen as a collection of objects, each with a set of features, which can be viewed as different dimensions as Fig. 3 shows. The most used Tensor Rank Decomposition (TRD) was invented by [45] (canonical decomposition) and [46] (parallel factors). After compacting X to a third-order tensor $\mathcal{X} \in \mathbb{R}^{N\times D\times M}$ as an example, TRD seeks to approximate tensor $\mathcal{X}$ with R components of rank-one tensor, that is:

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{o}_r \circ \mathbf{h}_r \circ \mathbf{w}_r, \tag{4}$$

where $\mathbf{o}_r \in \mathbb{R}^N$ as objects, $\mathbf{h}_r \in \mathbb{R}^D$ as features, and $\mathbf{w}_r \in \mathbb{R}^M$ as views.

The Tucker decomposition [47] is a higher-order version of the singular value decomposition (SVD), i.e. HOSVD [48]. It allows for decomposing a tensor $\mathcal{X} \in \mathbb{R}^{N\times D\times M}$ into a core tensor $\mathcal{T} \in \mathbb{R}^{P\times Q\times R}$, surrounded by a set of lower-order tensors. This decomposition can be used to approximate a tensor or to

find patterns within the tensor. It is defined as:

$$\mathcal{X} \approx \mathcal{T} \times_1 \mathbf{O} \times_2 \mathbf{H} \times_3 \mathbf{W} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} t_{pqr}\mathbf{o}_p \circ \mathbf{h}_q \circ \mathbf{w}_r, \quad (5)$$

where $t_{pqr} \in \mathcal{T}$. *Regularisation* is exploited in almost all MVC studies to minimise the adjusted loss function and prevent overfitting or underfitting of the proposed models. According to [49], we take the matrix of the m-th view $X^m = \{x_1^m, x_2^m, \ldots, x_N^m\} \in R^{N\times D^m}$. $x_i^m \in \mathbb{R}^D$ is the i-th sample, $x_j^m \in \mathbb{R}^N$ to denote the j-th column of features in X. Therefore, x_{ij}^m indicates the i-th sample's j-th column feature in m-th view. The $\ell_{p,q}$-norm of matrix X^m is stated as:

$$\|X^m\|_{p,q} = \left[\sum_{i=1}^{D^m}\left(\sum_{j=1}^{N}|x_{ji}^m|^p\right)^{q/p}\right]^{1/q}. \quad (6)$$

Through assigning p and q different values, the ℓ_1-norm (i.e., $p = q = 1$) is formulated as:

$$\|X^m\|_1 = \sum_{i=1}^{D^m}\left\|x_{\_i}^m\right\|_1, \quad (7)$$

where $x_{\_i}$ indicates the i-th dimensional feature set of all data samples. The $\ell_{2,1}$-norm (i.e., $p = 2, q = 1$), which is leveraged by [19] to establishes a consensus basis matrix, is defined as:

$$\|X^m\|_{2,1} = \sum_{i=1}^{D^m}\left\|x_{\_i}^m\right\|_2 = \mathrm{Tr}\left(X^T\widehat{\mathbf{D}}X\right), \quad (8)$$

where $\widehat{\mathbf{D}} \in \mathbb{R}^{N\times N}$ is a diagonal matrix and the i-th element on the diagonal is denoted as $\widehat{D}_{ii} = 1/(2\|x_{\_i}\|_2)(i \in [1\cdots D^m]$. The Frobenius norm ℓ_F (i.e., $p = q = 2$), which is used by [18], [50] to reduce the influence of missing data, is written as:

$$\|X\|_F = \sqrt{\sum_{i=1}^{D^m}\sum_{j=1}^{N}\left\|x_{ji}^m\right\|^2}. \quad (9)$$

To deal with tensor regularisation, Liu et al. [51] defined the tensor nuclear norm as

$$\|\mathcal{X}\|_* = \sum_{o=1}^{O}\varepsilon_o\left\|\mathcal{X}_o\right\|_*, \quad (10)$$

where O indicates the number of tensor modes, i.e., in our case, $O = 3$ since $\mathcal{X} \in \mathbb{R}^{N\times D\times M}$. ε_k is constant and satisfies $\varepsilon_k > 0$ and $\sum_{k=1}^{K}\varepsilon_k = 1$. Huang et al. [52] utilised this norm against noise and outliers.

III. Heuristic-Based Clustering

We categorise multi-view clustering (MVC) methods that leverage and improve classic machine learning algorithms developed with heuristics as Heuristic-based MVC (HMVC). HMVC deploys theoretical algorithms and heuristics to cluster multi-view data, which can be explained and proved by mathematical theories. This provides researchers with a considerable prospect to improve MVC from the perspective of theories and mathematics.

A. Nonnegative Matrix Factorisation

Nonnegative matrix factorisation (NMF) extends the standard k-means algorithm by allowing users to relax the constraints associated with the clustering indicator matrix. It was initially proposed to tackle single-view clustering. For example, assume we have $X = \{x_1, \ldots, x_N\} \in \mathbb{R}^{D\times N}$, which is a single-view nonnegative dataset. NMF is the task of exploring a cluster centroid matrix $\mathbf{V} \in \mathbb{R}^{D\times K}$ and a coefficient matrix (i.e., a clustering indicator matrix) $\mathbf{U} \in \mathbb{R}^{N\times K}$ such that $X \approx \mathbf{VU}$. NMF learns $\mathbf{V}$ and $\mathbf{U}$ by optimising the objective function as:

$$\min_{\mathbf{V},\mathbf{U}}\left\|X - \mathbf{V}\mathbf{U}^T\right\|_F^2, \text{ s.t. } \mathbf{V} \geq 0, \mathbf{U} \geq 0. \quad (11)$$

However, (11) is not convex either in $\mathbf{V}$ or $\mathbf{U}$. As a result, finding the global optima is impractical. Generally, there are two solutions to address this issue. The first solution is to exploit the gradient descent method [22], which reduces the squared distance for $\mathbf{U}$ defined as:

$$\mathbf{U}' \leftarrow \mathbf{U}' + \frac{\mathbf{U}}{(\mathbf{V}^T\mathbf{V}\mathbf{U})}\left[\left(\mathbf{V}^T X\right) - \left(\mathbf{V}^T\mathbf{V}\mathbf{U}\right)\right]. \quad (12)$$

Another solution is to use the multiplicative method [53], [54] that leverages the rules of iterative updating as the following:

$$\mathbf{V}' \leftarrow \mathbf{V}' \odot \frac{X\mathbf{U}}{\mathbf{V}\mathbf{U}^T\mathbf{U}}, \mathbf{U}' \leftarrow \mathbf{U}' \odot \frac{X^T\mathbf{V}}{\mathbf{U}^T\mathbf{V}}. \quad (13)$$

Notably, in addition to ℓ_1 and $\ell_{2,1}$-norm, there are a number of other criteria (e.g, Kullback-Leibler divergence) that can be used to assess the dissimilarity between X and $\mathbf{V}\mathbf{U}^T$.

MVC is based on the hypothesis that the different views in the same datasets should indicate the same underlying clustering structures. As much as possible, the coefficient matrices should be consistent across different views. To gain an understanding of the coefficient matrices of various views, a soft regularisation term is employed to achieve the common consensus [29], [53], [54], [55], [56], [57]. The objective function partitions the given dataset X into K clusters. Each view in the dataset is represented by a cluster, and the below function tries to minimise the within-cluster variance:

$$\min_{\mathbf{V}^m\mathbf{U}^m}\sum_{m=1}^{M}\left\|X^m - \mathbf{V}^m\mathbf{U}^{mT}\right\|_F^2 + \sum_{m=1}^{M}\lambda_m\left\|\mathbf{U}^m - \mathbf{U}^*\right\|_F^2, \text{ s.t. } \mathbf{V}^m \geq 0, \mathbf{U}^m \geq 0, \mathbf{U}^* \geq 0, \quad (14)$$

where the consensus matrix $\mathbf{U}^*$ characterises the inherent grouping pattern of the dataset in multiple views. Due to the fact that (14) does not need all views to use the same $\mathbf{U}^*$, this model becomes more vital when facing views of lower qualities. In this case, if we set the value of λ_m to be small enough, the influence of the low-quality views will decrease to the minimum.

Besides (14), the pair-wise CoNMF model [58] is another basic multi-view NMF form for enforcing a rigid constraint on all views, which enforces a single common consensus constraint. In the pair-wise CoNMF model, each pair of views is

subject to similarity constraints. When the coefficient matrices are absorbed from the pair of views, the pair-wise regularisation should ensure that they complement each other. As a result, we can obtain the clustering results of the higher quality. The regularised objective function is written as:

$$\begin{aligned}\min_{\mathbf{V}^m\mathbf{U}^m} &\sum_{m=1}^{M} w_m \left\|X^m - \mathbf{V}^m\mathbf{U}^{mT}\right\|_F^2 \\ &+ \sum_{p,q=1}^{M} \alpha_{pq} \left\|\mathbf{U}^p - \mathbf{U}^q\right\|_F^2, \\ \text{s.t. } &\mathbf{V}^m \geq 0, \mathbf{U}^m \geq 0, \mathbf{U}^p \geq 0, \mathbf{U}^q \geq 0, \end{aligned} \tag{15}$$

where using α_{pq} as the weighting parameter, $\mathbf{U}^{(p)}$ and $\mathbf{U}^{(q)}$ are constrained for similarity. Considering that the column vectors represent clusters while taking the ℓ_2-norm based on vectors into consideration, every single component of $\mathbf{U}^T\mathbf{U}$ presents an indication of the resemblance between the pair of clusters.

Evidently, under the multi-view setting, the resemblance of various view-specific clustering results needs to keep being consistent. This leads to the cluster-wise CoNMF model [58] as the third basic multi-view NMF model. In (15), by replacing the part for regularisation in pairs with the part for regularisation in clusters, we can define the cluster-wise CoNMF as:

$$\begin{aligned}&\sum_{p,q=1}^{M} \alpha_{pq} \left\|\mathbf{U}^{pT}\mathbf{U}^p - \mathbf{U}^{qT}\mathbf{U}^q\right\|_F^2, \\ &\text{s.t. } \mathbf{U}^p \geq 0, \mathbf{U}^q \geq 0. \end{aligned} \tag{16}$$

Intuitively, we can use the pre-mentioned multiplicative update rules for single-view NMF clustering models to optimise these three fundamental models of NMF-based multi-view clustering. Furthermore, some novel methods [59], [60], [61] adopted manifold regularisation to retain the local geometrical structure of the data space or to capture hierarchical information, aiming to reach the clustering consensus.

B. Graph Learning

The classic procedure of a single-view graph learning-based clustering method is shown in Algorithm 1. If we add a constraint of $\mathbb{G}^T \operatorname{diag}(G)\mathbb{G} = \mathbf{I}$, then it turns to be the classic normalised cut [62]. The same procedure can easily be adapted to graph learning-based MVC. Here, we mainly discuss the uniqueness of multi-view graph learning-based clustering, and the related technical improvements will be demonstrated in Section V.

Graph fusion and graph partitioning have been frequently leveraged to conduct the last three steps of the advanced graph learning-based MVC methods [3], [32], [42], [63], [64], [65]. Mathematically, we can compute the consensus graph matrix $U = \{u_{ij}\}_{i=1,j=1}^{N} \in \mathbb{R}^{N\times N}$, which is known as the unified matrix, from the signature matrices $\{\mathbf{S}^1, \ldots, \mathbf{S}^M\}$ by solving the problem below:

Algorithm 1: General Procedure of Graph Learning-Based Multi-View Clustering Methods.

Input: Feature vector set X
Output: Graph embedding matrix $\mathbb{G}$
1: Constructing the homogeneous graph $G(V, E)$ from X, where each entry e_{ij} in E indicates the edge weight between data points v_i and v_j representing their similarity;
2: Computing the Laplacian matrix of the graph G as $L_G = \operatorname{diag}(G) - (G^T + G)/2$, where using function $\operatorname{diag}(\cdot)$ to obtain a diagonal matrix $\operatorname{diag}(G)$, whose i-th diagonal element is $\sum_{j=1}^{N}(e_{ij} + e_{ji})/2$;
3: Computing the graph embedding matrix $\mathbb{G} \in \mathbb{R}^{N\times K}$ by solving $\min_{\mathbb{G}\in\mathbb{R}^{N\times K}} \operatorname{Tr}(\mathbb{G}^T\mathbf{L}_G\mathbb{G})$;
4: Dividing $\mathbb{G}$ into K clusters with an supplemental clustering method ;

$$\begin{aligned}&\min_{U} \sum_{m=1}^{M} w_m \left\|U - \mathbf{S}^m\right\|_F^2, \\ &\text{s.t. } u_{ij} \geq 0, \mathbf{1}^T\mathbf{u}_i = 1, \end{aligned} \tag{17}$$

where the m-th view weight is defined as $w_m = \frac{1}{2\sqrt{\|U-\mathbf{S}^m\|_F^2}}$.

Then we can impose a rank constraint on ranking data samples within the consensus graph matrix U and its graph Laplacian matrix L_U. The following three definitions [66] are in agreement with graph theory [66], [67], given that the graph matrix U is nonnegative:

- A symmetric positive semidefinite matrix is assigned as L_U. As a result, L_U has a complete set of N extant and orthogonal eigenvalues, all of which are existent and nonnegative;
- $L_U\mathbf{1} = \mathbf{0}$. Therefore, an eigenvalue of L_U is 0, while the matching eigenvector is $\mathbf{1}$;
- Assuming that there are r connected components in U, then L_U owns r eigenvalues that equal 0. In another word, we let $\operatorname{rank}(L_U) = N - K$ as $K = r$, the homologous U is able to be grouped into K clusters straightforwardly. Next, we can further constrain $\operatorname{rank}(L_U) = N - K$ to (17).

We infer that $\vartheta_i(L_U) \geq 0$ by making $\vartheta_i(L_U)$ as the i-th lowest eigenvalue of L_U. Then, we can achieve the constraint $\operatorname{rank}(L_U) = N - K$ when $\sum_{i=1}^{K}\vartheta_i(L_U) = 0$. Conforming to [68], the objective function is defined as:

$$\begin{aligned}&\min_{U} \sum_{m=1}^{M} w_m \left\|U - S^m\right\|_F^2 + \lambda_m Tr\left(\mathbb{G}^T L_U \mathbb{G}\right), \\ &\text{s.t.} \quad s_{ii}^m = 0, s_{ij}^m \geq 0, \mathbf{1}^T\mathbf{S}_i^m = 1, \\ &\qquad u_{ij} \geq 0, \mathbf{1}^T\mathbf{u}_i = 1, \mathbb{G}^T\mathbb{G} = \mathbf{I}. \end{aligned} \tag{18}$$

When each λ_m is fit (i.e., with a proper value), (18) will keep $\sum_{i=1}^{K}\vartheta_i(L_U) = 0$. Practically, we can control the number of components to be outputted by specifying the different λ values. After that, the K clusters can be obtained by connecting

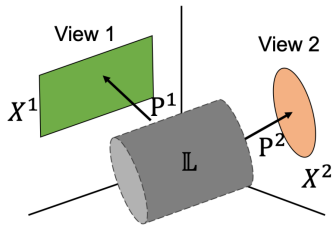


Fig. 4. Demonstration of multi-view latent representation. Observations $\{X^m\}_{m=1}^M (M \geq 2)$ corresponding to different views are partially projected by $\{\mathbf{P}^m\}_{m=1}^M$ from one underlying latent representation $\mathbb{L}$.

components with the support of the outputting consensus graph matrix U.

C. Latent Representation Learning

Latent representation learning-based MVC methods [10], [11], [22], [28], [34], [52], [69], [70], [71] aim to infer shared latent representations $\mathbb{l}$ of each data point from different views. Based on general assumptions, these various views have the same fundamental latent representation. Specifically, as Fig. 4 shows, the view-specific respective models $\{\mathbf{P}^1, \ldots, \mathbf{P}^M\}$ can reconstruct the observations on a variety of views with the shared latent representations $\mathbb{L} = \{\mathbb{l}_i\}_{i=1}^N$. Accordingly, we have $x_i^m = \mathbf{P}^m \mathbb{l}_i$. To derive the latent representation of multi-view data, we can define the object as:

$$\min_{\mathbf{P},\mathbb{L}} \mathcal{L}_h(X, \mathbf{P}\mathbb{L}), \tag{19}$$

When multiple views are combined, the resulting latent representation is more comprehensive than any of the individual views. This is because the different views complement each other, providing a more complete picture.

Then, gleaned from $\mathbb{L}$, according to the subspace clustering of (2), we can define the objective function of self-representation learning as:

$$\min_{\mathbf{U}} \mathcal{L}_r(X, \mathbb{L}\mathbf{U}) + \gamma \operatorname{Reg}(\mathbf{U}), \tag{20}$$

where $\mathbf{U}$ is the coefficient matrix of reconstruction.

Then, we can coalesce (19) and (20) as the objective function of latent representation learning-based MVC, which is written as:

$$\begin{aligned} &\min_{\mathbf{P},\mathbb{L},\mathbf{U}} \mathcal{L}_h(X, \mathbf{P}\mathbb{L}) + \tau_1 \mathcal{L}_r(\mathbb{L}, \mathbb{L}\mathbf{U}) + \tau_2 \operatorname{Reg}(\mathbf{U}), \\ &\text{s.t. } X = \mathbf{P}\mathbb{L}, \mathbb{L} = \mathbb{L}\mathbf{U} \text{ and } \mathbf{P}^T\mathbf{P} = \mathbf{I}. \end{aligned} \tag{21}$$

From the above demonstration, we know that the reasonable latent representation and the subspace reconstruction constraint can guarantee the performance of subspace clustering. By contrast, the comprehensive understanding of all views pledges the latent representation, which is enhanced by the subspace reconstruction.

D. Tensor Learning

Before conducting tensor learning-based MVC methods (i.e., tensor composition-based MVC methods), we expect to construct a tensor from given multi-view dataset X. Intuitively, we can model the multi-view dataset X naturally as objects, features, and view dimensions as a third-order tensor, or model the dataset as a similarity-based third-order tensor as shown in Fig. 3. There are several methods proposed to compact tensors for clustering. For example, we can divide X into three sets: a object set $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_R\}$, a feature set $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_R]$ and a view set $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_R\}$. Thus, a tensor can be represented as $\mathcal{X} = \{\mathbf{O}, \mathbf{H}, \mathbf{W}\} \in \mathbb{R}^{N \times D \times M}$ or a tensor $\mathcal{Y} = \{\mathbf{O}, \mathbf{O}, \mathbf{W}\} \in \mathbb{R}^{N \times N \times M}$. To process a similarity-based third-order tensor $\mathcal{Y}$, we can formulate the objective functions referring to the above-mentioned methods. In the following, we mainly present how to process $\mathcal{X}$.

In some clustering problems, it is important to consider a consecutive sequence of time points. For instance, in a video dataset containing characters, scenes and a series of time points, we would like to find out which group of characters exist during a specific period of time. When applying a constraint on successive time points, Chen et al. [72] suggested the TVCP (total variation based tensor decomposition) as a solution. To create a piece-wise constant function, the total variation regularises the time factor. The piece-wise constant function results in the reasonably compatible decomposition in intra-cluster and inter-cluster contexts. The TVCP model is defined as:

$$\min_{\mathbf{O},\mathbf{H},\mathbf{W}} \frac{1}{2} \left\| \mathcal{X} - \sum_{r=1}^{R} \mathbf{o}_r \circ \mathbf{h}_r \circ \mathbf{w}_r \right\|_F^2 + \tau \sum_{r=1}^{R} \|\mathbf{F}\mathbf{w}_r\|_1, \tag{22}$$

where $\mathbf{F}$ indicates the first-sequence difference of the matrix sized in $(D-1) \times D$, in which $\mathbf{f}_{ii} = 1$ and $\mathbf{f}_{i(i+1)} = -1$ for $i = 1, \ldots, D-1$, and the values of rest components are assigned as 0. To establish a piece-wise constant function, the first term corresponds to Tensor Rank Decomposition (TRD) [45], [46] of $\mathcal{X}$, while the time dimension factor is constrained by the second term.

Liu et al. [53] proposed a tensor decomposition-based MVC framework, specifically the Tucker decomposition. They demonstrate that a general structure of multi-view spectral clustering that is based on tracing maximisation is analogous to a Tucker decomposition problem as the following:

$$\begin{aligned} &\max_U \sum_{m=1}^{M} \operatorname{Tr}\left(U^T \operatorname{Nor}(S^m) U\right), \text{ s.t. } U^T U = \mathbf{I}, \\ &\Updownarrow \\ &\max_U \left\| \mathcal{X} \times_1 U^T \times_2 U^T \times_3 \mathbf{I}^T \right\|_F^2. \end{aligned} \tag{23}$$

We can also reformulate another general form of multi-view spectral clustering that considers view weights as a

Tucker problem:

$$\begin{aligned}&\max_{U,\mu} \operatorname{Tr}\left(U^T\left(\sum_{m=1}^{M} w_m \operatorname{Nor}(S^m)\right)U\right)\\&\text{s.t. } U^TU=\mathbf{I}, w_m\geq 0, \sum_{m=1}^{M} w_m=1.\\&\Updownarrow\\&\max_{U,\mu} \left\|\mathcal{X}\times_1 U^T\times_2 U^T\times_3 \mu^T\right\|_F^2\\&\text{s.t. } U^TU=\mathbf{I}, w_m\geq 0, \sum_{m=1}^{M} w_m=1.\end{aligned} \tag{24}$$

Using this methodology, various spectral clustering problems can be addressed by a tensor decomposition algorithm. This provides a strong connection between the different types of spectral clustering problems and highlights the ability of the tensor methodology to solve a variety of problems.

IV. Neural Network-Based Clustering

Another trending solution to handle multi-view data is to combine views, subsequently feed them into deep learning models (i.e., models of neural networks). Deep learning has illustrated extraordinary performance in many practical fields, including recognising faces, processing images, analysing natural language, detecting objects, and managing customer relationships. Deep learning exploits multiple nonlinear transformations, leading to better feature representations than shallow learning. Although multi-view clustering (MVC) using deep learning (i.e., neural network-based MVC (NNMVC)) seems to be a more practicable way to process multi-view data, there is no evidence showing that NNMVC has more apparent advantages over algorithm-based MVC (AMVC) (see Section III).

There are two challenges for NNMVC. The first is to devise different neural networks for different circumstances or real-world data views. The second is to make technical contributions to benefit NNMVC because existing deep learning models are well established. However, we can consider technology combination and framework improvement to contribute to the development of NNMVC, such as introducing contrastive learning into MVC. This part will introduce two popular NNMVC methods: deep representation learning and deep graph learning, and their recent technical innovations.

A. Deep Representation Learning

An auto-encoder that has both an encoder and a decoder is often constructed using a deep representation learning-based technique. Each data point is embedded into a representation space by the encoder using a nonlinear mapping function, and the decoder reconstructs the data from that representation. Theory-wise, deep representation learning-based MVC is comparable to multi-view spectral clustering. Since it is simple to add encoder layers and related decoder levels to an auto-encoder, it might theoretically be more adaptable.

However, deep representation learning-based MVC methods are more likely deal with bi-view data. For example, in [73], [74], [75], they assume $M=2$, and $X=\{X^1,X^2\}$ where $X^1=x_i^1{}_{i=1}^N$ and $X^2=x_i^2{}_{i=1}^N$. In order to extract shared representations, Ngiam et al. [73] reconstructed both views based on the single available view while testing. Their proposed method has a shared feature extraction network $\mathscr{f}(\cdot)$ and two reconstruction networks $\mathscr{p}(\cdot)$ and $\mathscr{q}(\cdot)$ for two different views respectively. Wang et al. [75] named this model as auto-encoder (SplitAE), shown schematically in Fig. 5(a). The objective of this model is the sum of reconstruction errors for the two views:

$$\min_{\mathcal{P}_\mathscr{f},\mathcal{P}_\mathscr{p},\mathcal{P}_\mathscr{q}} \frac{1}{N}\sum_{i=1}^{N}\left(\left\|x_i^1-\mathscr{p}\left(\mathscr{f}\left(x_i^1\right)\right)\right\|^2+\left\|x_i^2-\mathscr{q}\left(\mathscr{f}\left(x_i^1\right)\right)\right\|^2\right), \tag{25}$$

where $\mathcal{P}_\mathscr{f}$, $\mathcal{P}_\mathscr{p}$ and $\mathcal{P}_\mathscr{q}$ are learnable parameters for corresponding networks (note that, $\mathcal{P}_\mathscr{g}$ in the following two equations is the learnable parameter of $\mathscr{g}(\cdot)$).

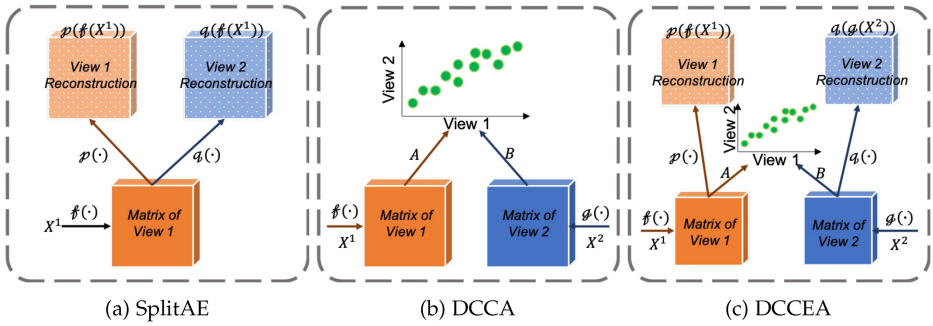
Then, Andrew et al. [74] introduced DCCA which extends neural networks with termed deep canonical correlation analysis (CCA). DCCA leverages two deep neural networks $\mathscr{f}(\cdot)$ and $\mathscr{g}(\cdot)$ to obtain view-specific nonlinear features (see Fig. 5(b)). The objective of maximising canonical correlation between the extracted features $\mathscr{f}(X^1)$ and $\mathscr{g}(X^2)$ is defined as:

$$\begin{aligned}&\max_{\mathcal{P}_\mathscr{f},\mathcal{P}_\mathscr{g},A,B} \quad \frac{1}{N}\operatorname{tr}\left(A^T\mathscr{f}\left(X^1\right)\mathscr{g}\left(X^2\right)^TB\right)\\&\text{s.t. } A^T\left(\frac{1}{N}\mathscr{f}\left(X^1\right)\mathscr{f}\left(X^1\right)^T+\gamma^1\mathbf{I}\right)A=\mathbf{I},\\&\qquad B^T\left(\frac{1}{N}\mathscr{g}\left(X^2\right)\mathscr{g}\left(X^2\right)^T+\gamma^2\mathbf{I}\right)B=\mathbf{I},\\&\qquad a_i^T\mathscr{f}\left(X^1\right)\mathscr{g}\left(X^2\right)^Tb_j=0, \quad \text{for} \quad i\neq j,\end{aligned} \tag{26}$$

where $A=\{a_1,\ldots,a_L\}$ and $B=\{b_1,\ldots,b_L\}$ are the CCA directions that forecast the deep neural network outputs and γ^1 and γ^2 are positive regularisation parameters for sample covariance estimation [76], [77]. In DCCA, testing is conducted using projection mapping function $A^T\mathscr{f}(\cdot)$. Despite the challenge of accurately reconstructing one view from the other view, CCA-based objectives provide the simple and effective way to learn a predictor of a function (or subspace) from the second view. Therefore, it is highly useful to enrich the complementary information for multi-view data even if their views are not very correlated.

To improve the canonical correlation between learned bottleneck representations and the auto-encoders' reconstruction mistakes, deep canonically correlated auto-encoders (DCCAE) integrate two auto-encoders based on both CCA and reconstruction-based goals in the form:

$$\begin{aligned}&\min_{\mathcal{P}_\mathscr{f},\mathcal{P}_\mathscr{g},\mathcal{P}_\mathscr{p},\mathcal{P}_\mathscr{q},A,B} -\frac{1}{N}\operatorname{tr}\left(A^T\mathscr{f}(X^1)\mathscr{g}(X^2)^TB\right)\\&+\frac{\beta}{N}\sum_{i=1}^{N}\left(\left\|x_i^1-\mathscr{p}\left(\mathscr{f}\left(x_i^1\right)\right)\right\|^2+\left\|x_i^2-\mathscr{q}\left(\mathscr{g}\left(x_i^2\right)\right)\right\|^2\right),\\&\text{s.t. the same constraints in (26).}\end{aligned} \tag{27}$$



(a) SplitAE (b) DCCA (c) DCCEA

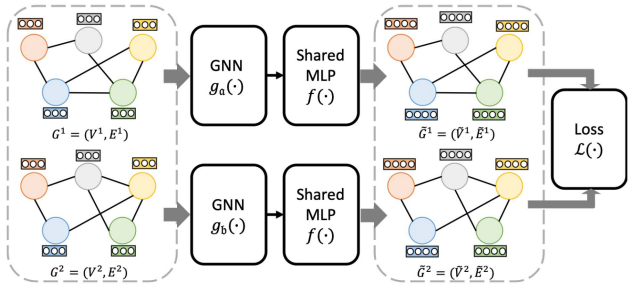
Fig. 5. Schema of deep multi-view representation learning models.

where β is a positive trade-off parameter. Another way to look at DCCEA is DCCA with an extra regularisation term. To optimise DCCA and DCCEA, we can adopt stochastic optimisation, where stochastic gradients are calculated by summing the gradients of the auto-encoder and DCCA term. The cutting-edge deep representation learning-based MVC methods, such as [9], [78], [79], [80], [81], [82], [83], [84], [85], are generally developed based on the concepts of SplitAE, DCCA and DCCEA. Also, with treating (25), (26) and (27) as functions processing pair-wise view mutual information (MI), we can easily extend them to cope with multi-view data (i.e., $M > 2$).

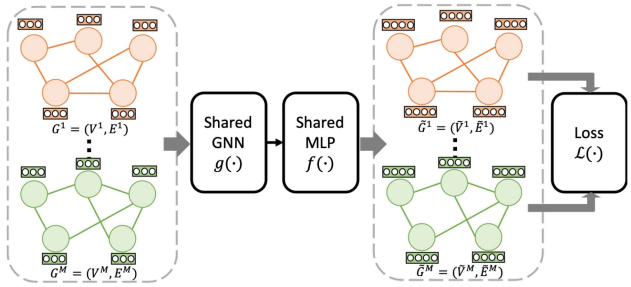
B. Deep Graph Learning

Recent years have seen a rise in the use of deep graph learning-based MVC, which offers great potential for technology fusion, e.g., there are the two representatives of the state-of-the-art deep graph learning methods [86], [87] to deal with multi-view data. We may infer from the general frameworks, as shown in Fig. 6, that deep representation learning-based MVC is the ancestor of deep graph learning-based MVC. Deep graph learning-based MVC, on the other hand, places more emphasis on the structural data (i.e., global context) of graph-based data.

We have two options to devise the graph neural network (GNN) based framework as shown in Fig. 6. The first option is for single-view datasets. We can construct an affinity graph from the dataset (if it is not a graph dataset), and then leverage data augmentations to generate two different graph views. We can use two different dedicated graph encoders to embed two different graph views, i.e., $g_{\mathfrak{a}}(\cdot)$, $g_{\mathfrak{b}}(\cdot)$ which obtains graph representations as $\mathbb{R}^{N\times D_x} \times \mathbb{R}^{N\times N} \longmapsto \mathbb{R}^{N\times D_h}$, where D_x indicates the dimensionality of node features and D_h is the dimensionality of learned representations. Then, we feed the obtained representations into a shared projection head $f(\cdot)$ which is a multilayer perceptron (MLP) learning representation as $\mathbb{R}^{N\times D_h} \longmapsto \mathbb{R}^{N\times D_h}$. Here, we omit to describe the design of encoders and projection (i.e., layers and functions) since the design differs for different datasets and problems. This results in two sets of node representations $H^1, H^2 \in \mathbb{R}^{N\times D_h}$ corresponding to two different graph views generated from the same affinity graph. We need to maximise the MI between each pair of representations of graph views using $\mathrm{MI}(\cdot)$ in order to train the encoders in an end-to-end manner and learn graph-level



(a) $M = 2$, augmenting data for two different views



(b) $M > 2$, transforming multi-view data for M affinity graphs

Fig. 6. The general frameworks of deep graph learning-based multi-view clustering. In Fig. 6a, different colours indicate different data sample, while in Fig. 6b, different colours refer to different graph views.

representations. We formulate the objective as:

$$\max_{\mathcal{P}} \frac{1}{2N} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\mathrm{MI}\left(h_i^1, h_j^2\right) + \mathrm{MI}\left(h_i^1, h_j^1\right) + \mathrm{MI}\left(h_i^2, h_j^2\right) \right), \tag{28}$$

where $\mathcal{P}$ indicates all parameters of graph encoders and projection heads in the devised framework, and $h_i^1 \in H^1$ and $h_i^2 \in H^2$.

The second option is mainly applied to predict linkage between nodes. Assume there is a single-view image dataset. We can utilise several image descriptors to generate M data views, where $M > 2$ (see Fig. 6(b)). Then, we build M affinity graphs from generated data views. After that, we leverage a shared GNN-based encoder $g(\cdot)$ and a shared projection head $f(\cdot)$ to embed each view and obtain M graph-level representations as $\{H^1, \ldots, H^M\}$. Similarly to (28), to maximise MI among

graph-level representations, we define the objective as:

$$\max_{\mathcal{P}} \frac{1}{2M} \sum_{m=1}^{M} \sum_{n=1}^{M} \left(\mathrm{MI}\left(H^m, H^n\right)\right). \tag{29}$$

In the advanced deep graph learning-based MVC methods [65], [88], [89], they mostly combine (28) and (29) while adopting contrastive learning in their proposed frameworks for exploring underlying information.

V. Multi-View Clustering Advancements

This section discusses what advancements the novel multi-view clustering (MVC) methods have generally committed. Intuitively, there are three aspects we can research in any machine learning-based problem. The three aspects are pre-processing input data, processing input data, and post-processing output data (see Fig. 1(c)). Sometimes, researchers contributed to fusing the last two aspects in their proposed methods. The following to-be-discussed four main advancements of existing MVC methods also conform to these three aspects.

A. Input Construction

In this section, we introduce the potential contributions to constructing input data. Input construction has been increasingly focused since it plays an essential role in clustering performance. In addition, for different real-world application scenarios, there expect various measures of input construction. This leads researchers to contribute to MVC technically.

1) Anchor-Based Graph Construction: Instead of constructing affinity graphs, many studies [23], [29], [41], [42], [43], [63], [90], [91], [92], [93] construct bipartite graphs or anchor graphs for learning consensus and heuristic-based partitioning, or for deep learning. They aim to reduce the computational complexity. Given a graph $G(V, E)$ transformed from X, we need to select anchors before constructing a bipartite graph or a anchor graph. Because the amount of anchors is substantially fewer than the amount of original data points, the generated anchor-based graph efficiently reduces the computational complexity of MVC methods.

Bipartite Graph Construction: The traditional framework of multi-view bipartite graph can be defined as:

$$\begin{aligned} &\min_{B^m, B} \left\|X^m - \mathbf{C}^m B^m\right\|_F^2 + \mathrm{Reg}\left(B^m, B\right), \\ &\text{s.t. } B^m \geq 0, B^{mT}\mathbf{1} = \mathbf{1}, \end{aligned} \tag{30}$$

where $\mathbf{C}^i \in \mathbb{R}^{D_i \times K}$ indicates the K selected anchors in m-th view. The produced bipartite graph $B \in \mathbb{R}^{K \times N}$ can benefit the following clustering efficiently, which reduces the computation complexity from $O(N^2)$ to $O(NK)$ $(K \ll N)$. Furthermore, there are two main ways to create anchors: the first way is to leverage the random sampling (R-sample) of column data and the second is to utilise the k-means [41], [42], [90], [91]. Since almost all the state-of-the-art studies exploit k-means to create anchors, the performance of k-means is apparently better than R-sample under the same number of anchors, which has been proved by [91].

Anchor Graph Construction: Instead of separating anchor selection and graph construction, Wang et al. [92] aimed to jointly conduct these two processes to learn a *anchor graph*, then formulates the objective as:

$$\begin{aligned} &\min_{W, J^m, A, Z} \sum_{m=1}^{M} w_m^2 \left\|X^m - J^m A Z\right\|_F^2, \\ &\text{s.t. } W^T\mathbf{1} = 1, J^{mT} J^m = \mathbf{I}_K, \\ &\quad A^T A = \mathbf{I}_K, Z \geq 0, Z^T\mathbf{1} = \mathbf{1}, \end{aligned} \tag{31}$$

where $A \in \mathbb{R}^{D_K \times K}$ is the unified anchor matrix where D_K is the selected feature dimension across all view, and Z is the unified anchor graph with $N \times K$ dimension, and J^m is the anchor projection matrix from m-th view, which could project the unified anchor to corresponding original data space. Based on (31), [93] adds $\|\mathbf{Z}\|_F^2$ as the second term for graph fusion. Also, it selects D_k as the common dimension and chooses the number of anchors $K \in \{D_k, 2D_k, 3D_k\}$. As it claimed, the common dimension and the orthogonal constraint jointly make A more discriminative.

2) Approximate Tensor Construction: After compacting X to a third-order tensor $\mathcal{X} \in \mathbb{R}^{N \times D \times M}$, TRD seeks to resemble tensor $\mathcal{X}$ using R components of rank-one tensor, that is:

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{o}_r \circ \mathbf{h}_r \circ \mathbf{w}_r, \tag{32}$$

where $\mathbf{o}_r \in \mathbb{R}^N$ as objects, $\mathbf{h}_r \in \mathbb{R}^D$ as features, and $\mathbf{w}_r \in \mathbb{R}^M$ as views.

Based on higher-order singular value decomposition (HOSVD) [48] (see (5)), the tensor Singular Value Decomposition (t-SVD) [95] is defined as:

$$\mathcal{X} = \mathcal{U} * \mathcal{G} * \mathcal{V}^T, \tag{33}$$

where $\mathcal{U} \in \mathbb{R}^{N \times N \times M}$ and $\mathcal{V} \in \mathbb{R}^{D \times D \times M}$ are orthogonal tensors, $\mathcal{G} \in \mathbb{R}^{N \times D \times M}$ is an f-diagonal tensor. Fig. 7 visualises the decomposition.

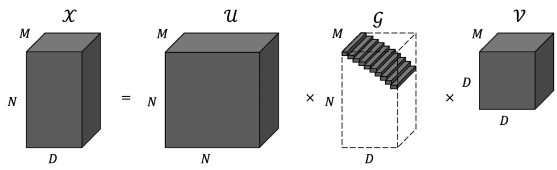


Fig. 7. The visualisation of tensor Singular Value Decomposition [94].

The above demonstrates the basic and classic methods to approximate tensors. As the development of tensor learning-based MVC, the recent studies [94], [96] propose to impose low-rank constraints to approximate tensors (i.e., *low-rank tensor approximation and multi-rank minimisation*). To extract the high order correlations among diverse views, Xie et al. [94] utilised t-SVD

to conduct tensor nuclear norm. Its objective is formulated as:

$$\begin{aligned}&\min_{\mathbf{Z},\mathbf{E}} \lambda\|\mathbf{E}\|_{2,1} + \|\mathcal{Z}\|_{\circledast},\\ &\text{s.t.} \quad X^m = X^m\mathbf{Z}^m + \mathbf{E}^m,\\ &\qquad m = \{1\cdots M\},\\ &\qquad \mathcal{Z} = \Phi\left(\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^M\right),\\ &\qquad \mathbf{E} = \{\mathbf{E}^1, \mathbf{E}^2, \ldots, \mathbf{E}^M\},\end{aligned} \tag{34}$$

where $\mathbf{Z}^m \in \mathbb{R}^{N\times N}$ is the coefficient matrix for the m-th view with each $\mathbf{z}_i^m$ being the new representation of sample x_i^m, and $\mathbf{E}^m$ is the noise in the m-th view. $\Phi(\cdot)$ indicates a function to construct the tensor $\mathcal{Z}$, where the function merges different representation $\mathbf{Z}^m$ to a 3-mode tensor, and then revolve its dimensionality from $N \times N \times M$ to $N \times M \times N$, and $\|\cdot\|_{\circledast}$ denotes the t-SVD-based tensor nuclear norm. We shall notice a significant disadvantage of [94], i.e., (34) has high computation cost, because it requires the matrix inversion on X for a multi-rank tensor, and the tensor transformation using $\Phi(\cdot)$. To tackle this issue, Wu et al. [97] proposed a linear constraint $\mathcal{P} = \mathcal{Z} + \mathcal{E}$ to formulate the objective as:

$$\begin{aligned}&\min_{\mathcal{Z},\mathcal{E}} \lambda\|\mathcal{E}\|_{2,1} + \|\mathcal{Z}\|_{\circledast}\\ &\text{s.t. } \mathcal{P} = \mathcal{Z} + \mathcal{E},\end{aligned} \tag{35}$$

where $\mathcal{P} \in \mathbb{R}^{N\times N\times M}$ is established by M similarity matrices P^m. P^m is produced by the Markov chain. Chen et al. [96] adopted (35) as their objective, but it replaces the tensor nuclear norm with a nonconvex function as:

$$\begin{aligned}&\min_{\mathcal{Z},\mathcal{E}} \lambda\|\mathcal{E}\|_{2,1} + \|\mathcal{Z}\|_{\theta}\\ &\text{s.t. } \mathcal{P} = \mathcal{Z} + \mathcal{E},\end{aligned} \tag{36}$$

where

$$\|\mathcal{Z}\|_{\theta} = \frac{1}{N}\sum_{m}^{M}\sum_{i}^{N}\psi\left(\sigma_i\left(\hat{\mathcal{Z}}^m\right), \theta\right). \tag{37}$$

where $\hat{\mathcal{Z}}$ is the fast Fourier transformation of $\mathcal{Z}$, and θ is a nonnegative parameter. $\psi(\cdot)$ indicates the nonconvex functions.

Remarks: The importance of input construction has been apparently ignored in existing MVC methods. Instead of technically promoting input construction, most MVC methods adopt existing intuitive algorithms, e.g., k-means. However, as the increase of view amount and data amount, the complexity of input instruction will increase exponentially. This definitely will bring more challenges to research MVC.

B. Regularisation

Almost all MVC studies leverage regularisation to minimise the adjusted loss function and prevent overfitting or underfitting of the proposed models. The proposed models can also make accurate predictions by incorporating these two measures. This part will introduce the basics and several technical modifications and applications of regularisation terms.

Jiang et al. [98], Kumar et al. [4] pioneered the a co-regularised MVC method, where the following cost function as a measure of disagreement $\mathrm{Dis}(\cdot)$ between the m-th view and n-th view:

$$\mathrm{Dis}\left(X^m, X^n\right) = \left\|\frac{S^m}{\|S^m\|_F^2} - \frac{S^n}{\|S^n\|_F^2}\right\|_F^2. \tag{38}$$

[49] imposes a structured sparsity penalty $\ell_{2,1}$-norm regularisation to its constructed sparse transformation matrix $\mathbf{U}$, which represents the relation between the original features and the latent low-dimensional representations. [55] modifies manifold regularisers in a pseudo-supervised style and in a discriminant style. There are several methods [24], [61], [99], [100], [101] that exploit a graph Laplacian regularisation or a hyper-Laplacian regularisation to correlate the low-dimensional consensus representation $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^N$ with the consensus similarity graph $\mathbf{S} = \{\mathbf{s}_{i,j}\}_{i=1,j=i}^{N,N}$, which is defined as:

$$\min_{\mathbf{P},\mathbf{S}} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\|\mathbf{p}_i - \mathbf{p}_j\|_2^2 s_{ij}. \tag{39}$$

In the proposed auto-encoder of [11], by leveraging an adversarial regularisation, Du et al. [11] ensured that the low-dimensional representation of each view met a desired prior distribution. They also used diversity regularisation to enforce more diverse or complementary subspace representation among multiple perspectives and layers. To determine the cluster number, Yuan et al. [30] devised a regularisation form, namely sum-of-norm regularisation, which uses ℓ_2-norm to maximise the sum of common presentations considering the features in the kernel space. Furthermore, Huang et al. [56] leveraged the merit of dual manifold regularisation to develop a Dual-regularised Multi-view Non-negative Matrix Factorisation (DMvNMF) algorithm.

Remark: Regularisation has been comprehensively leveraged in the state-of-the-art MVC methods and covers most of the clustering process. However, we can tell that there are hardly any mathematical improvements in the regularisation terms. Instead, the state-of-the-art tends to apply existing terms to enhance their novelty in MVC objectives. For example, co-regularisation is to use existing regularisers in parallel, and adversarial regularisation is to utilise existing regularisation to enhance adversarial learning.

C. Self- or Auto-Weighted Measure

As the majority of existing MVC methods consider all view weights either equal or fixed valued, there are an increasing number of studies [2], [24], [102], [103], [104], [105] introduce the auto-weighted or self-weighted measures to assign adaptive importance of different views. We can take the intuitive form of multi-view subspace clustering (see (2)) as an example to demonstrate the self-weighted MVC objective, which can be defined as:

$$\begin{aligned}&\min_{S^m}\sum_{m=1}^{M} w_m\left\|X^m - X^mS^m\right\|_F^2 + \gamma\,\mathrm{Reg}\left(S^m\right),\\ &\text{s.t. } S^m \geq 0, S^m\mathbf{1} = \mathbf{1},\end{aligned} \tag{40}$$

where the constant w_m indicates the weight of each view, which is formulated as:

$$w_m \triangleq \frac{1}{\|X^m - X^m S^m\|_F}. \tag{41}$$

We can see that the w_m refers to the m-th view weight. However, Nie et al. [106] clarified that (41) is not fair when the m-th view has a dramatic disagreement on X^m, i.e., the m-th view shall not be assigned with a significant weight w_m. To tackle with this issue, Nie et al. [106] exploited a unified evaluation, which is formulated as:

$$S^{m*} = \arg\min_{S^m} \sum_{m=1}^{M} w_m \|X^m - X^m S^m\|_F^2 + \gamma \operatorname{Reg}(S^m),$$
$$\text{s.t. } S^m \geq 0, S^m \mathbf{1} = \mathbf{1}, \tag{42}$$

where

$$w_m \triangleq \frac{1}{\|X^m - X^m S^{m*}\|_F}. \tag{43}$$

An extra hyperparameter β to maintain the smoothness of the weight assignment [63], [105], [107] is defined as:

$$w_m \triangleq \beta \frac{1}{\|X^m - X^m S^m\|_F}. \tag{44}$$

Remark: Including self- or auto-weighted measures as a technical contribution in MVC has become trending in recent years. Weighting views is essential for addressing general research problems because different views can play different roles while learning consensus, and we should not manually assign the importance to each view. However, when coping with a variety of real-life applications, such as plant disease detection and face recognition, we expect the auto-weighting to be developed as more adaptive and automatic to adjust the MVC objective towards different scenarios. For a specific example, colour features can be essential when processing face images, but they can be less important when it comes to plants. Therefore, existing self- or auto-weighted measures have the potential to be further improved and innovated.

D. Contrastive Strategy

This part will discuss the innovation in the design of MVC frameworks and their objectives, which aim to maximise the mutual agreement or information across all views achieving their greatest consensus. A contrastive technique bootstraps the grouping of various views by using prior knowledge or learning from each other. Iteratively implementing this strategy produces view-specific clustering results tending towards each other and leading to the broader consensus. We demonstrate the general procedure of the contrastive MVC framework in Fig. 8. During the development of MVC framework, many frameworks using contrastive strategies have been proposed. For example, the co-training technique has been employed in [12], [15], [63], [98] that utilised labels automatically learnt in one view in order to produce discriminative subspaces in another, by integrating the clustering algorithm with linear discriminant analysis. Also, Liu et al. [6], [108] utilised multi-kernel learning to deal with multi-view data, where they employ kernels corresponding to various views and then combine them either linearly or non-linearly.

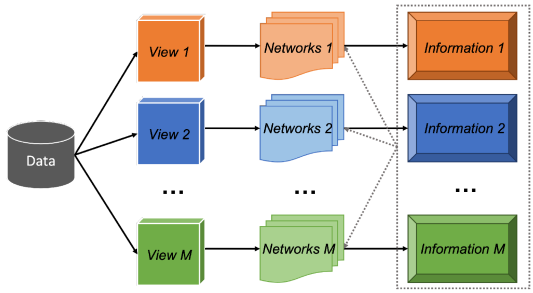


Fig. 8. The general procedure of the contrastive multi-view clustering framework. The multi-view data produces multiple data views, then process each data view to its corresponding neural networks to obtain information (e.g., clustering results, embedding, kernel, etc.), then backpropagate the consensus of all information to each encoder.

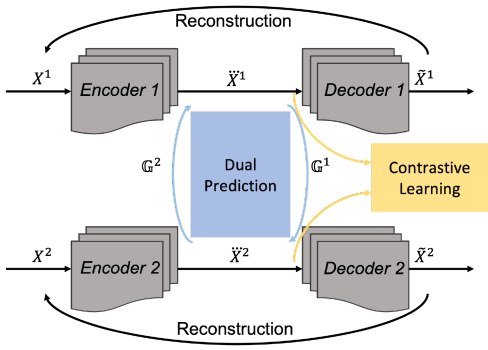


Fig. 9. The showcase of COMPLETER using bi-view. There are three joint learning objectives in COMPLETER, including within-view reconstruction, cross-view contrastive learning, and cross-view dual prediction [89].

Chen et al. [109] introduced the comprehensive concept of contrastive learning has been acknowledged by researchers. Brilliantly, Lin et al. [89] proposed COMPLETER to exploit contrastive learning for dealing with incomplete views as shown in Fig. 9. Their novelty is that COMPLETER deploys three contrastive strategies for dual prediction, contrastive learning and incomplete view reconstruction, respectively. Meanwhile, contrastive learning also derives into contrastive graph learning (CGL) for processing graphs [110], [111], which has achieved excellent performance on graph or node classification. We can see the huge potential of CGL, but it has not been widely recognised in the MVC field yet. Fig. 6 has shown the general procedure of CGL, but omitted the objective demonstration at the end. Here, we make up the diagram of the general node-level CGL objective in Fig. 10.

Zeng and Xie [110], Wan et al. [111] adopt the learning objective of CL rather straightforwardly. In doing so, they focused on node-level representations, and neglected the graph-level information. For any node v_i, its embedding generated in one view $v_i^1 \in V^1$ and the embedding in the other view $v_i^2 \in V^2$, form a positive pair, whereas embeddings of other nodes are negative samples. The pairwise objective for each positive pair

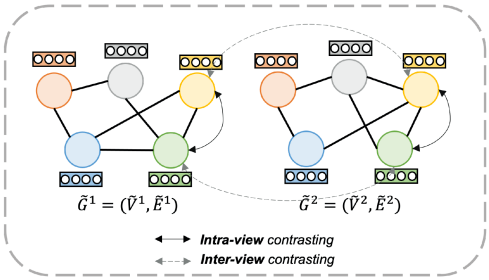


Fig. 10. The diagram of the general node-level contrastive graph learning objective following by Fig. 6.

(v_i^1, v_i^2) is defined as

$$\ell\left(v_i^1, v_i^2\right) = -\log \frac{\exp(\text{sim}\left(v_i^1, v_i^2\right)/\tau)}{\underbrace{\exp(\text{sim}\left(v_i^1, v_i^2\right)/\tau)}_{\text{positive pair}} + \mathbb{E} + \mathbb{A}}, \tag{45}$$

where sim denotes the function computing cosine similarity, τ is the temperature parameter, $\mathbb{E}$ identifies contrasting of inter-view negative pairs as $\mathbb{E} = \sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(v_i^1, v_k^2)/\tau)$, and $\mathbb{A}$ denotes the contrasting of intra-view negative pairs as $\mathbb{A} = \sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(v_i^1, v_k^1)/\tau)$, where $\mathbb{1}_{[k \neq i]} \in \{0, 1\}$ is an indicator function. The overall objective is then defined as:

$$\mathcal{J} = \frac{1}{2N} \sum_{i=1}^{N} \left[\ell\left(v_i^1, v_i^2\right) + \ell\left(v_i^2, v_i^1\right)\right]. \tag{46}$$

Remark: Before the popularity of contrastive learning, MVC methods have widely exploited the concept of mutual information maximisation among views, which is also the main idea of contrastive learning. As contrastive learning gets trending, combining contrastive learning and deep graph learning brings more research potential to MVC for exploring complementary information of multiple views, which considers both the global and local context of data. In addition, we can take the heterogeneity of multi-view data in consideration to explore more research possibility.

VI. Datasets and Experiments

A. Datasets

This section will categorise the open datasets, which are commonly adopted to evaluate multi-view clustering (MVC) methods, into three classes, i.e., multi-view image datasets, graph datasets, and other multi-view feature-based datasets. Before introducing open multi-view datasets that are well established, we will introduce *image descriptors* that generate multiple views of single-view images. Multiple different views can be constructed in a variety of feature view descriptors, and each of them has its feature focus or focuses.

In details of Fig. 11, both CLD and Intensity can produce the features of colour distribution. Scalable Colour Descriptor (SCD) generates colour-oriented features. Gabor, CENTRIST, GIST, and Local Binary Pattern (LBP) are for textural information of images. Histogram of Oriented Gradients (HOG) extracts joint histograms from images. Fuzzy Colour produces fuzzy-linking histograms. In addition, the rest image descriptors either improve or fuse the above-mentioned descriptors for comprehensive information, for example, Edge Histogram Descriptor (EHD) expresses the local edge distribution in the images. By using various image descriptors, we are able to convert a single-view image dataset to a multi-view image dataset.

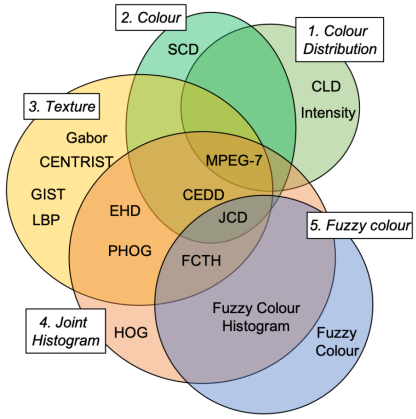


Fig. 11. The demonstration of various existing image view descriptors and their feature focuses. To note, even some image descriptors are in the same category, it does not mean they produce the same feature view. Moreover, Wavelet Moments (WM) descriptor is not included in the demonstration for its uniqueness, which describes interactions among image pixels.

1) Multi-View Image Datasets:

a) *COIL-20* [112] has 1,440 Gy-scale images of 20 classes and each class contains 72 images. Each image has been extracted under 3 views, where the first is a 1024-dimensional intensity feature, the second is a 3304-dimensional LBP feature, and the third is a 6750-dimensional Gabor feature.

b) *Caltech7* [113] contains 1,474 images of 7 classes. Each image has 6 views of features extracted. These views include 48-dimensional Gabor feature, 40-dimensional WM feature, 254-dimensional CENTRIST feature, 1984 dimensional HOG feature, 512-dimensional GIST feature, and 928-dimensional LBP feature.

c) *UCIdigits* [114] includes 2,000 data points and 10 digit classes. There are 200 handwritten digits in every class. There are six views: 216-dimensional profile correlation, the second being 76-dimensional Fourier Coefficient, the third being 64-dimensional Karhunen-Loeve Coefficient, the fourth being 47-dimensional Zernike moments, the sixth being 240-dimensional intensity averaged in two 3-column windows, and the fifth being 6-dimensional morphological features.

d) *MITIndoor* [115] consists of 5,360 images with 67 categories as a scene dataset. It has four types of features including 4096-dimensional PHOW, 3600-dimensional LBP, 1770-dimensional CENTRIST, and 1240-dimensional deep features.

e) *NUS-WIDE* [116] consists of 269,648 images, 5,018 related tags, and 81 ground-truth concepts. Each image

has 6 views: a 64-dimensional colour histogram, a 128-dimensional wavelet texture, a 123-dimensional edge direction histogram, a 225-dimensional block-wise colour moment extracted over a 5 by 5 fixed grid, and a 500-dimensional bag of words extracted from SIFT descriptions.

f) *ORL* [120] contains 40 person subjects, each of which consists of 10 different images. Each image has three feature views including 4096-dimensional intensity feature, 3304-dimensional LBP feature and 6750-dimensional Gabor.

2) Graph Datasets:

a) *DBLP* [121] is a co-authorship network composed of 10,305 authors. This network has 617 layers, each layer representing different publication categories.

b) *Facebook* [122] includes 1,640 users and multiple kinds of connections as a three-layer social network. The first layer reveals the friendship between each pair of users. The second layer indicates whether two users belong to the same group. The third layer indicates whether two users shared photos.

c) *CiteSeer* [118] contains 3312 scientific publications classified into 6 categories, namely Agents, AI, DB, IR, ML, and HCI. Scientific publications are represented by nodes, and citation relationships are represented by links. The absence/presence of keywords is represented by a 3703-dimensional one-hot encoding vector for each node.

d) *Enron e-mail* [123] has 184 users and 44 layers. In spite of being a temporal network, it can be viewed as a multi-layer network. Different layers represent different months of communication.

e) *Cora* [119] contains 2,708 documents from 7 classes. There are 5,429 edges on the graph. The first view contains a 1433-dimensional binary matrix, which indicates the presence of corresponding words. The other view is 2708-dimensional reflecting how the papers cite one another and their relationships

3) Multi-View Feature-Based Datasets:

a) *BBCSport* [124] contains 116 documents captured from the BBC sports news website. Five topical labels are manually annotated on each document in four segments.

b) *3Sources* [124] is a multi-view dataset that is inherently incomplete. There are 948 articles obtained from three reputable online news sources, which include the BBC, Reuters, and The Guardian, covering 416 different news stories. Three sources report 169 of these stories, two sources report 194, and one source reports 53.

c) *Reuters* [125] consists of six classes of 18,758 samples each, written in five different languages and translated into five different languages. There were 21,511, 24,892, 34,251, 15,506 and 11,547 translations into five different languages: English, French, German, Italian, and Spanish.

d) *HHAR* [126] contains records of human activity grouped into six categories, a total of 10,299 nodes, whose content is described by a 561-dimensional feature matrix. Top-5 neighbours are utilised to construct an undirected nearest neighbour graph.

B. Evaluation Metrics

To assess the performance of clustering methods, we can utilise three mainstream metrics. The first is the pairwise F-measure metric [127] to calculate precision (Pre), recall (Rec) and F1 score. For node v_i, v_j in the k-th cluster $\mathbb{c}_k \in \mathbb{C}$, we denote the ground truth labels by $L(v_i)$ and $L(v_j)$, and the cluster label by $C(v_i)$ and $C(v_j)$, where $i \neq j$ and each pair will only be counted once. We can formulate the pairwise correctness as

$$\mathrm{Cor}(v_i, v_j) = \begin{cases} 1, & \text{if } L(v_i) = L(v_j) \\ 0, & \text{otherwise} \end{cases} \tag{47}$$

To evaluate the clustering accuracy, ACC is formulated as:

$$\mathrm{ACC} = \frac{\sum_{i=1}^{N} \mathrm{Cor}\left(v_i, \mathrm{map}\left(v_i\right)\right)}{N}, \tag{48}$$

where $\mathrm{map}(v_i)$ maps the v_i to its corresponding cluster, which makes $\mathrm{Cor}(\cdot)$ returning its cluster label $L(\mathrm{map}(v_i))$.

Furthermore, we can employ the normalised mutual information (NMI) [128] as a metric, which is defined as:

$$\mathrm{NMI}(\Omega, \mathbb{C}) = \frac{\mathrm{I}(\Omega, \mathbb{C})}{\sqrt{\mathrm{H}(\Omega)\mathrm{H}(\mathbb{C})}}, \tag{49}$$

where Ω means the ground truth and $H(\cdot)$ is the entropy metric.

Also, for highlighting the ratio of a single class in each cluster, we introduce the purity (PUR) as:

$$\mathrm{PUR} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{k}} \max_{\mathrm{j}} \left|\mathbb{c}_{\mathrm{i}} \cap \mathbb{t}_{\mathrm{j}}\right|, \tag{50}$$

where $\mathbb{t}_j$ is the classification that has the maximum count in cluster $\mathbb{c}_i$, i.e., the highest frequent label in each cluster will be the classification of the cluster.

C. Benchmark and Discussion

We select 15 MVC methods in the benchmark to evaluate their clustering performance on the 5 multi-view image datasets (see Table II), 2 graph datasets (see Table IIIa), and 3 other feature-based multi-view datasets (see Table IIIb). Please note that we collect the clustering results from reviewed articles rather than re-conducting the experiments to reproduce these clustering results. From the big picture of these three tables, we have the below observations:

- *The adaptability of these methods is limited:* We find that some methods can cluster multi-view image datasets with a better performance, but when it comes to graph datasets and multi-view featured datasets, their performances are not outstanding, such as MVGL [7]. Similarly, we find that some methods are developed to deal with specific scenarios, thus, each of them mainly targets one kind of datasets.
- *It is unbalanced to use the datasets in recent methods:* Despite that we have chosen the most commonly applied datasets, we observe that UCIdigits [114], MITIndoor [115], NUS-WIDE [116], CiteSeer [118] and Cora [119] are generally significantly neglected in the more recent methods. These datasets have either small-sized

TABLE II
CLUSTERING PERFORMANCE ON MULTI-VIEW IMAGE DATASETS (UNDERLYING INFORMATION UTILISATION; LARGE-SCALE CLUSTERING; INCOMPLETE OR PARTIAL CLUSTERING; NOISE AND OUTLIERS REDUCTION)

	COIL-20 [112]			Caltech7 [113]			UCIdigits [114]			MITIndoor [115]			NUS-WIDE [116]		
Method	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity	ACC
MIC [50]	72.42	80.49	58.64	37.99	81.29	58.64	40.12	50.74	46.34	-	-	-	17.00	15.78	14.53
DCCAE [75]	70.58	78.00	55.51	59.14	-	41.89	-	-	-	-	-	-	-	-	-
MVGL [7]	89.20	85.84	66.08	58.81	84.47	62.95	89.05	94.20	94.20	-	-	-	-	-	-
MLAN [102]	92.50	-	84.24	63.60	-	78.00	65.63	-	65.93	40.82	-	23.20	-	-	-
MCGC [8]	83.78	70.84	**88.21**	59.47	85.06	64.51	94.22	97.55	97.55	-	-	-	-	-	-
t-SVD-MSC [94]	88.40	-	83.00	**85.80**	-	60.70	-	-	-	75.00	-	68.10	-	-	-
UEAF [20]	57.90	-	47.40	-	-	-	-	-	-	-	-	-	-	-	-
GMC [3]	34.46	55.49	45.64	60.56	**88.47**	69.20	73.60	-	85.10	20.40	-	9.9	7.88	17.87	**16.50**
RMSL [79]	94.10	-	82.19	-	-	-	51.10	-	57.80	37.20	-	27.90	-	-	-
LMVSC [43]	38.46	33.45	28.63	11.85	49.80	37.04	75.60	-	79.00	52.20	-	37.10	**12.95**	19.82	15.53
GNLTA [96]	-	-	-	-	-	-	**98.10**	-	**98.10**	**91.50**	-	**80.60**	-	-	
FMCNOF [91]	-	-	-	39.22	80.54	72.65	-	-	-	-	-	-	8.00	22.09	13.36
IMVC [23]				43.58	80.96	60.40	-	-	-	-	-	-	10.42	**22.17**	12.53
AMCGM [117]	**94.50**	-	86.74	81.60	-	**92.48**	-	-	-	-	-	-	-	-	
SMSC-NN [105]	91.70	**92.30**	85.30	59.10	60.50	53.30	-	-	-	-	-	-	-	-	-

TABLE III
CLUSTERING PERFORMANCE ON NON-IMAGE MULTI-VIEW DATASETS (UNDERLYING INFORMATION UTILISATION; LARGE-SCALE CLUSTERING; INCOMPLETE OR PARTIAL CLUSTERING; NOISE AND OUTLIERS REDUCTION)

	CiteSeer [118]			Cora [119]		
Method	NMI	Purity	ACC	NMI	Purity	ACC
MIC [50]	-	-	-	-	-	-
DCCAE [75]	**32.10**	-	**54.60**	**41.40**	-	**61.60**
MVGL [7]	8.03	-	28.16	6.31	-	23.71
MLAN [102]	2.55	31.79	31.39	1.85	22.80	22.64
MCGC [8]	10.37	-	32.04	0.38	-	30.43
t-SVD-MSC [94]	-	-	-	-	-	-
UEAF [20]	-	-	-	-	-	-
GMC [3]	-	-	-	13.89	-	36.67
RMSL [79]	-	-	-	-	-	-
LMVSC [43]	8.75	31.76	29.35	21.45	**45.86**	40.84
GNLTA [96]	-	-	-	-	-	-
FMCNOF [91]	-	-	-	-	-	-
IMVC [23]	-	-	-	-	-	-
AMCGM [117]	-	-	-	-	-	-
SMSC-NN [105]	-	-	-	-	-	-

(a) GRAPH DATASETS

	BBCSports			3Sources			Reuters		
Method	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity	ACC
MIC [50]	52.41	-	64.66	53.56	-	56.33	-	-	-
DCCAE [75]	-	-	-	54.55	-	72.98	-	-	-
MVGL [7]	34.75	-	66.20	48.10	-	45.50	27.10	46.10	35.20
MLAN [102]	77.90	-	72.10	65.57	80.47	76.33	35.50	35.98	49.00
MCGC [8]	35.47	-	66.06	42.54	-	54.44	30.29	-	44.22
t-SVD-MSC [94]	-	-	-	**91.82**	-	**97.61**	24.88	-	43.40
UEAF [20]	70.71	87.41	78.22	56.47	75.50	62.60	-	-	-
GMC [3]	62.16	-	69.23	62.16	-	69.23	23.50	-	27.70
RMSL [79]	64.38	-	83.65	48.40	-	52.19	39.10	-	57.00
LMVSC [43]	82.88	**94.67**	94.67	67.10	**81.60**	72.10	26.27	**74.47**	43.31
GNLTA [96]	**98.90**	-	**98.10**	-	-	-	**96.70**	-	**96.70**
FMCNOF [91]	-	-	-	-	-	-	15.04	-	44.70
IMVC [23]	-	-	-	-	-	-	-	-	-
AMCGM [117]	-	-	-	74.50	-	78.11	-	-	-
SMSC-NN [105]	-	-	-	-	-	-	-	-	-

(b) OTHER MULTI-VIEW FEATURE-BASED DATASETS

dimensions of features or similar view-specific features, which brings additional challenge to MVC tasks because discriminating different views often lead to the poor performance of MVC clustering.

- *Consideration of efficiency and effectiveness together has become one of the recent attentions in MVC research:* Most existing work, such as MCGC [8], only reported their effectiveness of clustering, but didn't evaluate the efficiency of their models. There are some recent studies that paid attention to the efficiency as well. For instance, when processing the dataset Caltech7, the running time of FMCNOF [91] is 0.19 seconds (72.65% accuracy), MLAN [102] obtains clustering results in 11.55 seconds (78.00% accuracy), LMVSC [43] outputs clustering results in 11.95 seconds (37.04% accuracy), and the running time of MCGC is 30259.00 seconds (88.21% accuracy). From the above results, it can be seen that the time costs of these methods are very different when they optimize the effectiveness of their clustering results. In this case, it could be an interesting research trend to consider both effectiveness and efficiency in novel MVC researchs.

Looking into the details of their clustering results, we can also observe:

- The methods that target underlying information utilisation have better clustering accuracy. The accuracy of MCGC [8] reaches 88.21% on COIL-20 [112], and the accuracy of AMCGM [117] is 92.48% on Caltech7 [113]. GNLTA [96] incredibly has 98.10% accuracy on UCIdigits, 80.60% accuracy on MITIndoor, 98.10% accuracy on BBCSports, and 96.70% om Reuters. Furthermore, DCCAE [75] proposed in 2015, keeps the best accuracy results 54.60% and 61.60% on CiteSeer and Cora.
- All methods perform poor on NUS-WIDE, CiteSeer and Cora. Especially, for the best performance on NUS-WIDE, LMVSC [43] reaches 12.95% NMI, IMVC [23] has 22.17% purity, and GMC [3] reaches 16.50% accuracy. Also, the state-of-the-art did not pay enough attention to these three datasets.
- A dramatic performance gap occurs when evaluating with Reuters. The best performance happens to GNLTA with 96.70% NMI and 96.70% accuracy, while FMCNOF [91] gets the lowest NMI 15.04% and GMC has the lowest accuracy 22.70%.

These observations can call back an essential idea of this survey, that is, every method shall be devised to handle specific applications. We have to highlight the interesting point that the more recent studies do not necessarily have more accurate clustering results, like FMCNOF and SMSC-NN, since those methods might emphasise the fast algorithms (i.e., large-scale clustering) and the capability of tackling outliers, etc., rather than the clustering accuracy. Furthermore, we can find a trend from these results that the more recent studies adopt more non-image datasets to evaluate their proposed methods.

VII. Conclusion and Discussion

In the era of Big Data and the early days of the Fourth Industrial Revolution, the proliferation of data and the dramatic procreation of real-world application problems require the advancements of clustering technologies that can explore and exploit comprehensive knowledge and underlying information. Therefore, multi-view clustering (MVC) has gained increasing attention. This paper reviewed the most recent MVC methods and technologies.

We provide a clear MVC roadmap to potential MVC researchers. Based on how each MVC method constitutes, we summarised the MVC method into two technical mechanisms: heuristic-based MVC (HMVC) based on classic machine learning algorithms in Section III, and neural networks-based MVC (NNMVC) based on deep neural networks (DNNs) in Section IV. In HMVC, we introduced four technical methods, i.e., nonnegative matrix factorisation, graph learning, latent representation learning, and tensor decomposition. In NNMVC, we presented two technical methods: deep representation learning and deep graph learning. Under the subsection of each method, besides general concepts and technologies of each method, we also illustrated state-of-the-art technical innovations in Section V, which was to inspire researchers to create their technical novelty.

Even though MVC was first proposed around two decades ago, there are no standards to determine which MVC method is best because different technical methods have their benefits and drawbacks. Concisely, HMVC methods rely on prior knowledge and certain constraints, while NNMVC methods are significantly dependent on input pre-processing and neural network parameters. Furthermore, any to-propose technical approach needs to be devised according to specific scenarios or real-world applications, such as [129]. This will benefit researchers in figuring out creative and advanced technical measures. Fortunately, all of those technologies mentioned in this paper are intimately connected, enabling us to explore the research possibility and advance MVC theoretically.

To provide convenience and inspiration for future researchers, in Section VI, we introduced how to convert a single-view image dataset to a multi-view dataset and included 15 open datasets from three aspects (i.e., multi-view image datasets, graph datasets, and multi-view feature-based datasets). Also, we demonstrated commonly used evaluation metrics for MVC. Then, we compared and discussed some available benchmarking experimental results obtained from reviewed MVC studies.

Potential Research Directions: At the end of this survey, we would like to share some emerging and highly demanding issues for further MVC research.

1) *Multi-view Graph Construction:* A multi-view dataset needs to be transformed into a graph to leverage graph learning or deep graph learning. Existing relevant methods utilise classic algorithms, e.g., KD-tree and k-neighbours, to establish either an anchor-based graph or an affinity graph from a multi-view dataset. This brings high computation complexity. It could be more interesting for researchers to develop novel and efficient graph constructing algorithms based on theories and methodologies.
2) *Adaptive Multi-view Clustering:* There is a trend of utilising multi-view data to tackle various types of real-world applications. Besides multi-view image clustering and graph representation learning, the community of natural language processing has been attracted by multi-lingual processing. The community of video processing gets interested in multi-modal learning. Therefore, It could be more interesting for researchers to devise MVC methods adaptive to multiple real-world problems.
3) *Precise Weighing and Sampling:* Multi-view data will inevitably lead to high computational costs. Despite that some of the latest MVC methods attempted to approximate multi-view by weighing and sampling, there still exists a significant room to improve the accuracy with theoretical guarantee and the computational efficiency.
4) *Contrastive Fusion:* Existing MVC methods use mainly 3 fusion strategies to enhance clustering: data fusion, projected feature fusion, and result fusion. However, there is no work involving more than one fusion strategies to conduct contrastive learning by leveraging more underlying information. Thus, this could be a reasonable research direction for researchers to advance the MVC research.
5) *Heterogeneous Graph Leaning:* Since a single-view dataset can be considered as a homogeneous graph, we can treat the multi-view data as the heterogeneity information. Thus, there is a trend of leveraging the development of heterogeneous graph learning to develop MVC research.

References

[1] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proc. Int. Conf. Des. Minings*, 2004, pp. 19–26.
[2] F. Nie et al., "Self-weighted multiview clustering with multiple graphs," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2564–2570.
[3] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1116–1129, Jun. 2020.
[4] Y. Jiang, J. Liu, Z. Li, P. Li, and H. Lu, "Co-regularized PLSA for multi-view clustering," in *Proc. Asian Conf. Comput. Vis.*, Springer, 2012, pp. 202–213.
[5] Z. Guan, L. Zhang, J. Peng, and J. Fan, "Multi-view concept learning for data representation," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3016–3028, Nov. 2015.
[6] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1888–1894.
[7] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2887–2895, Oct. 2018.
[8] K. Zhan, F. Nie, J. Wang, and Y. Yang, "Multiview consensus graph clustering," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1261–1270, Mar. 2019.
[9] Z. Huang, J. T. Zhou, H. Zhu, C. Zhang, J. Lv, and X. Peng, "Deep spectral representation learning from multi-view data," *IEEE Trans. Image Process.*, vol. 30, pp. 5352–5362, 2021.
[10] J. Li, W. Qiang, C. Zheng, and B. Su, "RHMC: Modeling consistent information from deep multiple views via regularized and hybrid multiview coding," *Knowl.-Based Syst.*, vol. 241, 2022, Art. no. 108201.
[11] G. Du, L. Zhou, K. Lü, H. Wu, and Z. Xu, "Multiview subspace clustering with multilevel representations and adversarial regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 27, 2022, doi: 10.1109/TNNLS.2022.3165542.
[12] H. Wang, F. Nie, H. Huang, and F. Makedon, "Fast nonnegative matrix tri-factorization for large-scale data co-clustering," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1553–1558.

[13] Y. Li, C. Chen, W. Liu, and J. Huang, "Sub-selective quantization for large-scale image search," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2803–2809.

[14] R. Zhang and Z. Lu, "Large scale sparse clustering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2336–2342.

[15] J. Han, K. Song, F. Nie, and X. Li, "Bilateral k-means algorithm for fast co-clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1969–1975.

[16] F. Nie, W. Zhu, and X. Li, "Unsupervised large graph embedding," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2422–2428.

[17] Q. Lin, M. Men, L. Yang, and P. Zhong, "A supervised multi-view feature selection method based on locally sparse regularization and block computing," *Inf. Sci.*, vol. 582, pp. 146–166, 2022.

[18] H. Zhao, H. Liu, and Y. Fu, "Incomplete multi-modal visual data grouping," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2392–2398.

[19] M. Hu and S. Chen, "Doubly aligned incomplete multi-view clustering," 2019, *arXiv: 1903.02785*.

[20] J. Wen, Z. Zhang, Y. Xu, B. Zhang, L. Fei, and H. Liu, "Unified embedding alignment with missing views inferring for incomplete multi-view clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5393–5400.

[21] J. Wen, Z. Zhang, Z. Zhang, L. Fei, and M. Wang, "Generalized incomplete multiview clustering with flexible locality structure diffusion," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 101–114, Jul. 2021.

[22] J. Yin and S. Sun, "Incomplete multi-view clustering with cosine similarity," *Pattern Recognit.*, vol. 123, 2022, Art. no. 108371.

[23] S. Wang et al., "Highly-efficient incomplete large-scale multi-view clustering with consensus bipartite graph," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9776–9785.

[24] Z. Li, C. Tang, X. Zheng, X. Liu, W. Zhang, and E. Zhu, "High-order correlation preserved incomplete multi-view subspace clustering," *IEEE Trans. Image Process.*, vol. 31, pp. 2067–2080, 2022.

[25] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2149–2155.

[26] M. D. Collins, J. Liu, J. Xu, L. Mukherjee, and V. Singh, "Spectral clustering with a convex regularizer on millions of images," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2014, pp. 282–298.

[27] H. Gao, F. Nie, X. Li, and H. Huang, "Multi-view subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4238–4246.

[28] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4279–4287.

[29] B. Yang, X. Zhang, B. Chen, F. Nie, Z. Lin, and Z. Nan, "Efficient correntropy-based multi-view clustering with anchor graph embedding," *Neural Netw.*, vol. 146, pp. 290–302, 2022.

[30] C. Yuan, Y. Zhu, Z. Zhong, W. Zheng, and X. Zhu, "Robust self-tuning multi-view clustering," *World Wide Web*, vol. 25, pp. 489–512, 2022.

[31] C. Xu, Z. Guan, W. Zhao, H. Wu, Y. Niu, and B. Ling, "Adversarial incomplete multi-view clustering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 3933–3939.

[32] S. Huang, I. Tsang, Z. Xu, and J. C. Lv, "Measuring diversity in graph learning: A unified framework for structured multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5869–5883, Dec. 2022.

[33] C. Xu, H. Liu, Z. Guan, X. Wu, J. Tan, and B. Ling, "Adversarial incomplete multiview subspace clustering networks," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10490–10503, Oct. 2022.

[34] X. Wang, L. Fu, Y. Zhang, Y. Wang, and Z. Li, "MMatch: Semi-supervised discriminative representation learning for multi-view classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 6425–6436, Sep. 2022.

[35] Y. Yang and H. Wang, "Multi-view clustering: A survey," *Big Data Mining Analytics*, vol. 1, no. 2, pp. 83–107, 2018.

[36] L. Fu, P. Lin, A. V. Vasilakos, and S. Wang, "An overview of recent multi-view clustering," *Neurocomputing*, vol. 402, pp. 148–161, 2020.

[37] G. Chao, S. Sun, and J. Bi, "A survey on multiview clustering," *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 146–168, Apr. 2021.

[38] M.-S. Chen et al., "Representation learning in multi-view clustering: A literature review," *Data Sci. Eng.*, vol. 7, pp. 225–241, 2022.

[39] J. Wen et al., "A survey on incomplete multiview clustering," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 53, no. 2, pp. 1136–1149, Feb. 2023.

[40] X. Fang, Y. Xu, X. Li, Z. Lai, and W. K. Wong, "Robust semi-supervised subspace clustering via non-negative low-rank representation," *IEEE Trans. Cybern.*, vol. 46, no. 8, pp. 1828–1838, Aug. 2016.

[41] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2750–2756.

[42] X. Li, H. Zhang, R. Wang, and F. Nie, "Multiview clustering: A scalable and parameter-free bipartite graph fusion method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 330–344, Jan. 2022.

[43] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 4412–4419.

[44] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[45] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[46] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis," *UCLA Working Papers Phonetics*, vol. 16, pp. 1–84, 1970.

[47] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[48] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[49] P. Jing, Y. Su, Z. Li, and L. Nie, "Learning robust affinity graph representation for multi-view clustering," *Inf. Sci.*, vol. 544, pp. 155–167, 2021.

[50] W. Shao, L. He, and P. S. Yu, "Multiple incomplete views clustering via weighted nonnegative matrix factorization with $l_{2,1}$ regularization," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Springer, 2015, pp. 318–334.

[51] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

[52] A. Huang, W. Chen, T. Zhao, and C. W. Chen, "Joint learning of latent similarity and local embedding for multi-view clustering," *IEEE Trans. Image Process.*, vol. 30, pp. 6772–6784, 2021.

[53] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proc. SIAM Int. Conf. Data Mining*, SIAM, 2013, pp. 252–260.

[54] L. Zong, X. Zhang, and X. Liu, "Multi-view clustering on unmapped data via constrained non-negative matrix factorization," *Neural Netw.*, vol. 108, pp. 155–171, 2018.

[55] J. Ma, Y. Zhang, and L. Zhang, "Discriminative subspace matrix factorization for multiview data clustering," *Pattern Recognit.*, vol. 111, 2021, Art. no. 107676.

[56] Z. Huang, Y. Ren, X. Pu, L. Pan, D. Yao, and G. Yu, "Dual self-paced multi-view clustering," *Neural Netw.*, vol. 140, pp. 184–192, 2021.

[57] J. Tan, Z. Yang, J. Ren, B. Wang, Y. Cheng, and W.-K. Ling, "A novel robust low-rank multi-view diversity optimization model with adaptive-weighting based manifold learning," *Pattern Recognit.*, vol. 122, 2022, Art. no. 108298.

[58] X. He, M.-Y. Kan, P. Xie, and X. Chen, "Comment-based multi-view clustering of web 2.0 items," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 771–782.

[59] C. Xu, Z. Guan, W. Zhao, Y. Niu, Q. Wang, and Z. Wang, "Deep multi-view concept learning," in *Proc. Int. Joint Conf. Artif. Intell.*, Stockholm, 2018, pp. 2898–2904.

[60] K. Liu, X. Li, Z. Zhu, L. Brand, and H. Wang, "Factor-bounded nonnegative matrix factorization," *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 6, pp. 1–18, 2021.

[61] G. A. Khan, J. Hu, T. Li, B. Diallo, and H. Wang, "Multi-view data clustering via non-negative matrix factorization with manifold regularization," *Int. J. Mach. Learn. Cybern.*, vol. 13, no. 3, pp. 677–689, 2022.

[62] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[63] S. Huang, Z. Xu, I. W. Tsang, and Z. Kang, "Auto-weighted multi-view co-clustering with bipartite graphs," *Inf. Sci.*, vol. 512, pp. 18–30, 2020.

[64] Z. Kang et al., "Multi-graph fusion for multi-view spectral clustering," *Knowl.-Based Syst.*, vol. 189, 2020, Art. no. 105102.

[65] Y. Wang, D. Chang, Z. Fu, and Y. Zhao, "Consistent multiple graph embedding for multi-view clustering," *IEEE Trans. Multimedia*, vol. 25, pp. 1008–1018, 2023.

[66] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2001, pp. 269–274.

[67] B. Mohar, Y. Alavi, G. Chartrand, and O. Oellermann, "The Laplacian spectrum of graphs," *Graph Theory, Combinatorics, Appl.*, vol. 2, no. 871/898, 1991, Art. no. 12.

[68] K. Fan, "On a theorem of Weyl concerning eigenvalues of linear transformations I," *Proc. Nat. Acad. Sci. USA*, vol. 35, no. 11, 1949, Art. no. 652.
[69] C. Zhang et al., "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Jan. 2020.
[70] D. Xie, Q. Gao, Q. Wang, X. Zhang, and X. Gao, "Adaptive latent similarity learning for multi-view clustering," *Neural Netw.*, vol. 121, pp. 409–418, 2020.
[71] Y. Jin, C. Li, Y. Li, P. Peng, and G. A. Giannopoulos, "Model latent views with multi-center metric learning for vehicle re-identification," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1919–1931, Mar. 2021.
[72] C. Chen, X. Li, M. K. Ng, and X. Yuan, "Total variation based tensor decomposition for multi-dimensional data with time dimension," *Numer. Linear Algebra Appl.*, vol. 22, no. 6, pp. 999–1019, 2015.
[73] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 689–696.
[74] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2013, pp. 1247–1255.
[75] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2015, pp. 1083–1092.
[76] T. De Bie and B. De Moor, "On the regularization of canonical correlation analysis," in *Proc. Int. Symp. Independent Compon. Anal. Blind Signal Separation*, 2003, pp. 785–790.
[77] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2004.
[78] J. Li and H. Liu, "Projective low-rank subspace clustering via learning deep encoder," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2145–2151.
[79] R. Li, C. Zhang, H. Fu, X. Peng, T. Zhou, and Q. Hu, "Reciprocal multi-layer subspace learning for multi-view clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8172–8180.
[80] D. J. Trosten, S. Lokse, R. Jenssen, and M. Kampffmeyer, "Reconsidering representation alignment for multi-view clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1255–1265.
[81] J. Wen, Z. Zhang, Y. Xu, B. Zhang, L. Fei, and G.-S. Xie, "CDIMC-net: Cognitive deep incomplete multi-view clustering network," in *Proc. 29th Int. Conf. Int. Joint Conf. Artif. Intell.*, 2021, pp. 3230–3236.
[82] Y. Qin, H. Wu, X. Zhang, and G. Feng, "Semi-supervised structured subspace learning for multi-view clustering," *IEEE Trans. Image Process.*, vol. 31, pp. 1–14, 2022.
[83] J. Xu, Y. Ren, G. Li, L. Pan, C. Zhu, and Z. Xu, "Deep embedded multi-view clustering with collaborative training," *Inf. Sci.*, vol. 573, pp. 279–290, 2021.
[84] Q. Wang, Z. Ding, Z. Tao, Q. Gao, and Y. Fu, "Generative partial multi-view clustering with adaptive fusion and cycle consistency," *IEEE Trans. Image Process.*, vol. 30, pp. 1771–1783, 2021.
[85] H. Fu, Y. Geng, C. Zhang, Z. Li, and Q. Hu, "RED-Nets: Redistribution networks for multi-view classification," *Inf. Fusion*, vol. 65, pp. 119–127, 2021.
[86] Y. Yang, Z. Guan, J. Li, W. Zhao, J. Cui, and Q. Wang, "Interpretable and efficient heterogeneous graph convolutional network," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1637–1650, Feb. 2023.
[87] Y. Yang, Z. Guan, W. Zhao, L. Weigang, and B. Zong, "Graph substructure assembling network with soft sequence and context attention," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4894–4907, May 2023.
[88] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," 2020, *arXiv: 2006.05582*.
[89] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, "Completer: Incomplete multi-view clustering via contrastive prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11 174–11 183.
[90] L. Li and H. He, "Bipartite graph based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, pp. 3111–3125, Jul. 2022.
[91] B. Yang, X. Zhang, F. Nie, F. Wang, W. Yu, and R. Wang, "Fast multi-view clustering via nonnegative and orthogonal factorization," *IEEE Trans. Image Process.*, vol. 30, pp. 2575–2586, 2021.
[92] S. Wang et al., "Fast parameter-free multi-view subspace clustering with consensus anchor guidance," *IEEE Trans. Image Process.*, vol. 31, pp. 556–568, 2022.
[93] M. Sun et al., "Scalable multi-view subspace clustering with unified anchors," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 3528–3536.
[94] Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, and Y. Qu, "On unifying multi-view self-representations for clustering by tensor multi-rank minimization," *Int. J. Comput. Vis.*, vol. 126, no. 11, pp. 1157–1179, 2018.
[95] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, 2013.
[96] Y. Chen, S. Wang, C. Peng, Z. Hua, and Y. Zhou, "Generalized nonconvex low-rank tensor approximation for multi-view subspace clustering," *IEEE Trans. Image Process.*, vol. 30, pp. 4022–4035, 2021.
[97] J. Wu, Z. Lin, and H. Zha, "Essential tensor learning for multi-view spectral clustering," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5910–5922, Dec. 2019.
[98] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1413–1421.
[99] J. Liu et al., "A novel consensus learning approach to incomplete multi-view clustering," *Pattern Recognit.*, vol. 115, 2021, Art. no. 107890.
[100] J. Yin and S. Sun, "Incomplete multi-view clustering with reconstructed views," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2671–2682, Mar. 2023.
[101] W. Zhao, C. Xu, Z. Guan, and Y. Liu, "Multiview concept learning via deep matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 814–825, Feb. 2021.
[102] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2408–2414.
[103] P. Ren et al., "Robust auto-weighted multi-view clustering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 2644–2650.
[104] H. Peng and H. Cai, "Multi-view clustering through self-weighted high-order similarity fusion," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2021, pp. 1–6.
[105] S. Shi, F. Nie, R. Wang, and X. Li, "Self-weighting multi-view spectral clustering based on nuclear norm," *Pattern Recognit.*, vol. 124, 2022, Art. no. 108429.
[106] F. Nie, S. Shi, and X. Li, "Auto-weighted multi-view co-clustering via fast matrix factorization," *Pattern Recognit.*, vol. 102, 2020, Art. no. 107207.
[107] S. Huang, Z. Kang, I. W. Tsang, and Z. Xu, "Auto-weighted multi-view clustering via kernelized graph learning," *Pattern Recognit.*, vol. 88, pp. 174–184, 2019.
[108] D. Guo, J. Zhang, X. Liu, Y. Cui, and C. Zhao, "Multiple kernel learning based multi-view spectral clustering," in *Proc. IEEE 22nd Int. Conf. Pattern Recognit.*, 2014, pp. 3774–3779.
[109] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020, *arXiv: 2002.05709*.
[110] J. Zeng and P. Xie, "Contrastive self-supervised learning for graph classification," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 10 824–10 832.
[111] S. Wan, S. Pan, J. Yang, and C. Gong, "Contrastive and generative graph convolutional networks for graph-based semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 10 049–10 057.
[112] S. A. Nene et al., "Columbia object image library (COIL-100)," Tech. Rep. CUCS-006-96, Feb. 1996.
[113] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, 2004, pp. 178–178.
[114] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[115] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 413–420.
[116] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "Nus-wide: A real-world web image database from national university of Singapore," in *Proc. ACM Int. Conf. Image Video Retrieval*, 2009, pp. 1–9.
[117] D. Wu, X. Dong, F. Nie, R. Wang, and X. Li, "An attention-based framework for multi-view clustering on Grassmann manifold," *Pattern Recognit.*, vol. 128, 2022, Art. no. 108610.
[118] C. L. Giles, K. D. Bollacker, and S. Lawrence, "Citeseer: An automatic citation indexing system," in *Proc. 3rd ACM Conf. Digit. Libraries*, 1998, pp. 89–98.
[119] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Inf. Retrieval*, vol. 3, no. 2, pp. 127–163, 2000.
[120] The ORL database of faces. Accessed: May 30, 2022. [Online]. Available: http://www.face-rec.org/databases/
[121] M. K.-P. Ng, X. Li, and Y. Ye, "Multirank: Co-ranking for objects and relations in multi-relational data," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2011, pp. 1217–1225.

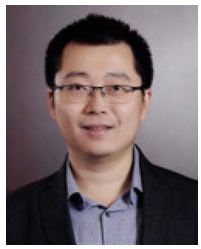
[122] K. Lewis, J. Kaufman, M. Gonzalez, A. Wimmer, and N. Christakis, "Tastes, ties, and time: A new social network dataset using facebook.com," *Social Netw.*, vol. 30, no. 4, pp. 330–342, 2008.
[123] B. W. Bader, R. A. Harshman, and T. G. Kolda, "Temporal analysis of semantic graphs using ASALSAN," in *Proc. IEEE 7th Int. Conf. Data Mining*, 2007, pp. 33–42.
[124] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 377–384.
[125] Reuters-21578 text categorization collection data set. Accessed: May 30, 2022. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection
[126] A. Stisen et al., "Smart devices are different: Assessing and mitigating-mobile sensing heterogeneities for activity recognition," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, 2015, pp. 127–140.
[127] C. Otto, D. Wang, and A. K. Jain, "Clustering millions of faces by identity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 289–303, Feb. 2018.
[128] K. Tian, S. Zhou, and J. Guan, "DeepCluster: A general clustering framework based on deep learning," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Springer, 2017, pp. 809–825.
[129] C. Xu, W. Zhao, J. Zhao, Z. Guan, X. Song, and J. Li, "Uncertainty-aware multiview deep learning for Internet of Things applications," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1456–1466, Feb. 2023.

Uno Fang (Student Member, IEEE) received the BS degree (Honours) in information technology from Deakin University, Australia, in 2017. He is currently working toward the PhD degree with the Deakin Univ.-Southwest Univ. Joint Research Centre on Big Data, Deakin University, in 2020. His current research interests include precision agriculture, computer vision and deep learning.

Man Li is currently working toward the PhD degree with the School of Information Technology, Deakin University. Her current research interests include similarity models in text data, time series based data analytics, big finance data analytics.

Jianxin Li (Senior Member, IEEE) received the PhD degree in computer science from Swinburne University of Technology, Australia, in 2009. He is an associate professor of Data Science with the School of Information Technology, Deakin University. He has published 90 high quality paper in top-tier venues, including *The VLDB Journal*, *IEEE Transactions on Knowledge and Data Engineering*, PVLDB and IEEE ICDE. He has received two competitive grants from Australian Research Council. His research interests include graph query processing, social network computing, and information network data analytics.

Longxiang Gao (Member, IEEE) received the PhD degree in computer science from Deakin University, Australia. He is currently a senior lecturer and co-founder of Deakin Blockchain Innovation Lab with the School of Information Technology, Deakin University. Before joined Deakin University, he was a post-doctoral research fellow with IBM Research & Development, Australia. His research interests include Fog/Edge computing, Blockchain, data analysis and privacy protection. He has more than 80 publications in the top venue, such as *IEEE Transactions on Mobile Computing*, *IEEE Internet of Things Journal*, etc. He has served as the TPC co-chair, publicity co-chair, organization chair and TPC member for many international conferences.



Tao Jia (Member, IEEE) received the PhD degree in physics from Virginia Tech, US in 2011. He worked a postdoctoral fellow with Northeastern University, US and Rensselaer Polytechnic Institute, US. He is now is a professor and the director with the Office of Information and Technology, Southwest University, China. His research interests include complex networks (such as network topology and brain network) and social computing.

Yanchun Zhang is an international research leader in databases, data mining, health informatics, web information systems, and web services. He has published more than 300 research papers in international journals and conferences proceedings, and authored/edited 20 books. His research has been supported by a number of Australian Research Council (ARC) linkage projects and discovery project grants. He is the editor-in-chief of *World Wide Web Journal* (Springer), and *Health Information Science and Systems* (Springer). He is chairman of the International Web Information Systems Engineering Society (WISE Society).