Multi-View Adjacency-Constrained Hierarchical Clustering

Jie Yang and Chin-Teng Lin, *Fellow, IEEE*

***Abstract*—This paper explores the problem of multi-view clustering, which aims to promote clustering performance with multi-view data. The majority of existing methods have problems with parameter adjustment and high computational complexity. Moreover, in the past, there have been few works based on hierarchical clustering to learn the granular information of multiple views. To overcome these limitations, we propose a simple but efficient framework: Multi-view adjacency-Constrained Hierarchical Clustering (MCHC). Specifically, MCHC mainly consists of three parts: including the Fusion Distance matrices with Extreme Weights (FDEW); adjacency-Constrained Nearest Neighbor Clustering (CNNC); and the internal evaluation Index based on Rawls' Max-Min criterion (MMI). FDEW aims to learn a fusion distance matrix set, which not only uses complementary information among multiple views, but exploits the information from each single view. CNNC is utilized to generate multiple partitions based on FDEW, and MMI is designed for choosing the best one from the multiple partitions. In addition, we propose a parameter-free version of MCHC (MCHC-PF). Without any parameter selection, MCHC-PF can give partitions at different granularity levels with a low time complexity. Comprehensive experiments tested on eight real-world datasets validate the superiority of the proposed methods compared with the 13 current state-of-the-art methods.**

***Index Terms*—Clustering, multi-view learning, parameter-free.**

I. Introduction

MULTI-VIEW clustering has received a lot of attention in recent years as an important learning paradigm in artificial intelligence. Different from traditional clustering methods [1], multi-view clustering is exploited to process multi-view data. Multi-view data means the data is collected from different sources in diverse domains, or obtained from various feature collectors [2]. For example, an image can be described by multiple heterogeneous features, such as scale-invariant feature transform (SIFT) descriptors [3], GIST descriptors [4], local binary patterns (LBP) [5], etc. Multiple compatible and complementary features are combined in multi-view clustering algorithms to improve clustering performance.

Most existing multi-view clustering methods are essentially divided into three categories: a) multi-view spectral clustering methods; b) multi-view subspace clustering methods; and c) other multi-view clustering methods [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23].

By combining information from multiple graphs, multi-view spectral clustering can learn latent cluster structures [24], [25], [26]. Multi-view spectral clustering is based on learning an intrinsic graph that contains information from multi-view data, then using the spectral clustering approach on the learned graph to generate clustering results [23]. For example, Zong et al. introduced a weighted multi-view spectral clustering algorithm based on the spectrum perturbation theory of spectral clustering [21], which uses spectral perturbation to simulate the weights of the views. To distinguish the clustering capacity differences of different views, Nie et al. developed an adaptively weighted Procrustes spectral clustering technique, where an indicator matrix can be generated to improve the performance [14]. Tang et al. presented the one-step multi-view spectral clustering, which combines the spectral embedding and K-means into one step to decrease the information loss and reduce the running time of the total clustering procedure [22].

The main idea behind multi-view subspace clustering is to learn a uniform subspace representation of many views [27]. Zheng et al., for example, introduced a multi-view constrained bilinear factorization subspace clustering method that improves clustering results by performing constrained bilinear factorization on the low-rank representation of multiple views [28]. Zheng et al. presented a feature concatenation multi-view subspace clustering to explore the consensus information of multi-view data by concatenating multi-view features into a joint representation [29].

Furthermore, various other multi-view clustering algorithms have recently been presented [8], [18], [30], [31]. For example, by introducing a collaborative deep matrix decomposition framework, the method proposed in [8] attempts to learn the hidden representations from multi-view data. Xu et al. proposed a deep autoencoder-based method to learn the embedded representations, which take both consensus and complementary information of multiple views into account [18].

Yet despite the importance of multi-view clustering and the plethora of existing algorithms in past decades, the most current approaches suffer from the following two problems in multi-view clustering: a) parameter adjustment, and b) excessive computational cost. For most multi-view clustering, such as multi-view spectral clustering [7], [11], [16], [21] and multi-view subspace clustering [9], [28], [29], [30], parameter adjustment is critical to the final performance of the models. For example, except for the target number of clusters, the weighted multi-view spectral clustering algorithm proposed by Zong et al. has two parameters that need to be set in order to assign an optimal weight to each view [21]. The low-rank tensor-based multi-view spectral clustering proposed by Chen et al. needs to adjust v+2 parameters to get good performance, where v is the number of views [32]. This means that if the data contains four views, and each parameter needs to be adjusted 10 times, the method needs to tune 10^6 parameter combinations to get the final good result. The multi-view subspace clustering based on the joint affinity graph presented by Tang et al. has two regularization parameters to adjust to balance the weights of each component in the objective function [33]. The constrained bilinear factorization multi-view subspace clustering developed by Zheng et al. has also two prior information-related parameters to tune to obtain competitive performance [28]. Prior information, such as noise level and label information, is needed to guide the specific parameter selection procedure, which is problematic for multi-view clustering. Furthermore, the computational complexity of most existing multi-view clustering algorithms is also high; for example, the time complexities of both multi-view spectral clustering and multi-view subspace clustering are $O(n^3)$, where n is the number of data samples. Most multi-view clustering algorithms have either the above two shortcomings, or one of them [2]. These two shortcomings also greatly hinder the application of multi-view clustering in practical scenarios.

On the other hand, from the perspective of basic clustering principles, many previous multi-view clustering algorithms are based on spectral clustering or subspace clustering, which have some inherent limitations. For example, spectral clustering [34], [35] suffers from the following three problems: a) the instability of results caused by different initializations; b) the K value required to construct adjacency matrix needs to be adjusted; and c) it can only provide clustering results with a single granularity. For subspace clustering [36], a) establishing the global density threshold causes the method to perform poorly in detecting clusters with varying densities; and b) setting regularization parameters for the number of subspaces is time-consuming. Few multi-view clustering algorithms are based on hierarchical clustering [23]. Compared with spectral clustering and subspace clustering, hierarchical clustering does not need extra hyper-parameters, and a dendrogram can be generated to provide clustering results with different granularity levels.

To the best of our knowledge, there are only two related works on multi-view hierarchical clustering in the past. The multi-view hierarchical clustering (MHC) [23] proposed by Zheng et al. exploits the average cosine distance of multiple views and a conventional nearest neighbor clustering [37] to obtain the clustering results of multi-view datasets. MHC has two main drawbacks, the first is that it only considers the weight of each view equally. Obviously, MHC cannot capture best consensus information well when different views are of different importance. The second is that it uses an existing nearest neighbor clustering method [37] to get the results, which only considers the neighbor relationship between samples and ignores the manifold structure in the data, also resulting in poor clustering results. Lin et al. proposed a contrastive multi-view hyperbolic hierarchical clustering method [38], which introduces the contrastive representation learning into multi-view hierarchical clustering. An obvious disadvantage of this approach is that there are five parameters that need to be tuned for competitive performance.

The motivation of this paper is to propose a multi-view hierarchical clustering method that does not require any parameter adjustment and can obtain competitive results in a short time. Such a multi-view clustering algorithm has great application potential in practical scenarios that require fast response. To achieve this goal, in this paper we propose a Multi-view adjacency-Constrained Hierarchical Clustering algorithm (MCHC). MCHC consists of three main parts: including the Fusion Distance matrices with Extreme Weights (FDEW); adjacency-Constrained Nearest Neighbor Clustering (CNNC); and the internal evaluation Index based on Rawls' Max-Min criterion [39] (MMI). FDEW aims to learn a fusion distance matrix set, which not only uses complementary information among multiple views, but exploits the information from each single view. This is because sometimes a specific single view has a better clustering-friendly representation than the complementary (or fusion) view [40]. CNNC obeys an intuitive rule that one cluster and its nearest neighbor with higher mass (size) should be grouped into the same cluster in the clustering process, which not only considers the neighbor relationship between samples but grasps the manifold structure in the data. CNNC generates multiple partitions based on FDEW. MMI is exploited to choose the best one from the multiple partitions. MCHC just needs to be assigned a desired number of clusters, which can be estimated based on the decision graph of CNNC. In addition, we propose a parameter-free version of MCHC (MCHC-PF). Without any parameter selection, MCHC-PF can give partitions at different granularity levels. MCHC-PF has lower time complexity, which is $O(nlogn)$.

The followings are the main contributions of this paper:

- 1) Proposing a multi-view adjacency-constrained hierarchical clustering (MCHC) algorithm that can obtain promising clustering results;
- 2) Proposing a parameter-free MCHC algorithm with low computational complexity;
- 3) Proposing the fusion distance matrices with extreme weights, which not only uses complementary information among multiple views, but exploits the information from each single view;
- 4) Proposing a novel hierarchical clustering algorithm: adjacency-constrained nearest neighbor clustering (CNNC);
- 5) Proposing the internal evaluation index based on Rawls' Max-Min criterion for selecting best partition;

- 6) The proposed methods' superiority is demonstrated by experimental results on eight real-world datasets.

II. Proposed Method

Multi-view adjacency-Constrained Hierarchical Clustering (MCHC) consists of three main components: including the Fusion Distance matrices with Extreme Weights (FDEW); adjacency-Constrained Nearest Neighbor Clustering (CNNC) and the internal evaluation Index based on Rawls' Max-M criterion (MMI).

A. Fusion Distance Matrices With Extreme Weights (FDEW)

When dealing with multi-view clustering problems, there are two intuitive methods to fuse multi-view data. One is concatenating all the features of multiple views [41]. Obviously, this method increases the dimensionality of fusion data, which increases the computational complexity and may result in the curse of dimensionality. The other method is to calculate an average similarity matrix $\bar{S} = \frac{1}{v} \sum_{i=1}^{v} S^{(i)}$ [42]. However, this approach cannot represent best consensus information well when different views are of different importance.

Previous research found a very interesting phenomenon: sometimes the clustering results obtained by the best single view are even better than that obtained by the consensus information or complementary information of multiple views [40]. One of the reasons for this is the existence of conflicting views in multi-view data, making the learned consensus or complementary information poor [43]. Inspired by the phenomenon above, we propose the fusion distance matrices with extreme weights (FDEW). FDEW not only uses the complementary information among multiple views, but also exploits the information from each single view, which alleviates the poor impact of conflicting views on the clustering results.

Given multi-view data $\{X^{(i)}\}_{i=1}^{v}$ collected from *v* views, for *i*-th view, $X^{(i)} \in R^{n \times dim_i}$, where *n* and dim_i are the number of data samples and the dimensionality of the *i*-th view respectively.

On the one hand, we regard the distance matrix $D^{(i)}$ ($D^{(i)} \in R^{n \times n}$) of each view as a fusion distance matrix with extreme weights, that is

$$D^{(i)} = 1 \times D^{(i)} + \sum_{j=1, j \neq i}^{v} 0 \times D^{(j)}, \tag{1}$$

On the other hand, we define a fusion distance matrix with equal weights:

$$D^{*} = \frac{1}{v} \sum_{i=1}^{v} D^{(i)}, \tag{2}$$

$D^{(i)}$ assigns the weight of $D^{(i)}$ to 1, and assigns the weight of the distance matrix of other views to 0. D^{*} treats the distance matrix of each view equally, and assigns the same weight to the distance matrix of each view. $D^{(i)}$ only uses the information from each single view, but D^{*} exploits complementary information among multiple views. Combine $D^{(i)}$ and D^{*} to form fusion distance matrices with extreme weights (FDEW), where $\{FDEW^{(r)}\}_{r=1}^{v+1} = \{D^{(1)}, D^{(2)}, \ldots, D^{(v)}, and\ D^{*}\}$.

When calculating the distance matrix $D^{(i)}$ of each view, we exploit cosine distance. There are two main motivations for using cosine metric: First, the cosine distance between two samples on a specific view equals the cosine distance between them on the latent representation [23]. Exploiting cosine metric directly can omit the additional computational overhead of learning the latent representation. Second, cosine metric captures semantic similarity better than Euclidean distance [44].

For the cosine distance between any two samples $x_a^{(i)}, x_b^{(i)}$ in *i*-th view, it is defined as

$$d\left(x_a^{(i)}, x_b^{(i)}\right) = 1 - \frac{{x_a^{(i)}}^T x_b^{(i)}}{\sqrt{{x_a^{(i)}}^T x_a^{(i)}} \sqrt{{x_b^{(i)}}^T x_b^{(i)}}}. \tag{3}$$

where $d\left(x_a^{(i)}, x_b^{(i)}\right) \in [0, 2]$.

Theorem 1. The cosine distance between $x_a^{(i)} and\ x_b^{(i)}$ in the i-th view is equivalent to the cosine distance between them on the latent representation [23].

Next, we use a single-view clustering algorithm to get cluster partitions based on distance matrices in FDEW.

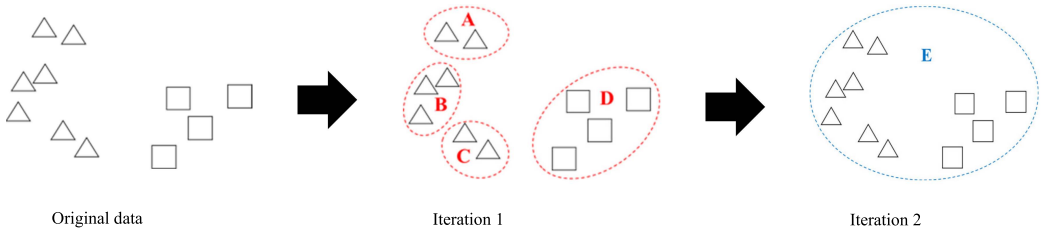
B. Adjacency-Constrained Nearest Neighbor Clustering (CNNC)

As the backbone of multi-view clustering, compared with spectral clustering or subspace clustering, hierarchical clustering does not need extra hyper-parameters and provides clustering results with different granularity levels. Recently, nearest neighbor clustering (NNC) has become a research focal point for hierarchical clustering [37], [45], [46]. Compared with traditional hierarchical clustering, such as average-link or ward-link, NNC has lower computational complexity (i.e., $O(nlogn)$) and can achieve better clustering performance. Existing NNC approaches, however, are entirely based on the statistic of nearest neighbor, i.e., the merging is done as long as the neighbor relationship is satisfied, which cannot capture the intrinsic manifolds in data well. Samples from different classes may be also merged in this fashion, lowering clustering accuracy. In this study, we propose a parameter-free adjacency-constrained nearest neighbor clustering (CNNC) algorithm, which exploits larger mass clusters to direct the merging process. This constrained way of merging captures the manifolds in data and prevents trivial wrong merging in conventional NNC approaches. The difference between the traditional NNC method and the proposed one is shown in Fig. 1.

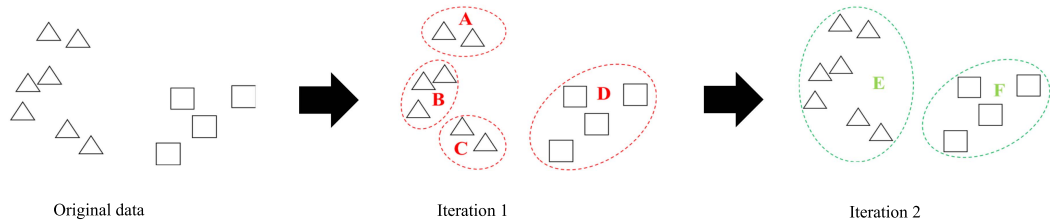
Given a single-view data $X^{(i)}$, initially, each sample is its own cluster. Given the number of samples contained in a cluster as the mass of the cluster, therefore, in the beginning, the mass of each cluster equals 1. Then, the following rule is applied to form connections between clusters:

$$\zeta_j \rightarrow \zeta_j^N,\ if\ mass\left(\zeta_j\right) \leq mass\left(\zeta_j^N\right), \tag{4}$$

where ζ_j denotes the *j*-th cluster, ζ_j^N denotes the 1-nearest cluster of ζ_j. Much research has been conducted to define the distance between two clusters. Here, we simply measure the nearest distance from any member of one cluster to any member of



(a) Conventional NNC



(b) Adjacency-constrained NNC

Fig. 1. A toy illustration of (a) Conventional NNC merging procedure and (b) the proposed CNNC algorithm. For simplification, we use dotted lines to represent clusters (i.e., A, B, C, etc.), and rectangles or triangles to represent data samples. We can see that, in iteration1, the generated clusters from conventional NNC and adjacency-constrained NNC are the same. Initially each sample is regarded as a cluster, and they all have the same mass (i.e., 1); i.e., they all satisfy the mass requirement in (4), so if the neighbor relationship is satisfied, a connection can be generated between clusters. In iteration 2, for conventional NNC, cluster A, cluster B, cluster C and cluster D are chosen to merge into one big cluster, E. because cluster A and cluster B are the nearest neighbor of each other, the nearest neighbor of cluster C is cluster B, and the nearest neighbor of cluster D is cluster C. however, in the proposed CNNC, cluster C and cluster D are not chosen to merge because mass(D)>mass(C). cluster A and cluster B, and cluster B and cluster C are both chosen to merge because mass(A)≤ mass(B) and mass(C)≤mass(B).

the other cluster as the distance between the two clusters, which is similar to the single-linkage method.$mass(\zeta_j)$ represents the mass of ζ_j (i.e, the number of samples ζ_j contains). Similarly, $mass(\zeta_j^N)$ is the mass of ζ_j^N. The symbol "$\rightarrow$" denotes a connection (i.e, merger) C_j between ζ_j and ζ_j^N. This process can be also defined in a graph *G*,

$$A\left(\zeta_j,\zeta_j^N\right)=\begin{cases}1, \text{ if } \operatorname{mass}\left(\zeta_j\right)\leq \operatorname{mass}\left(\zeta_j^N\right),\\ 0, \text{ otherwise.}\end{cases} \tag{5}$$

where *A* is the adjacency matrix of *G*. Then, new clusters can be obtained by calculating the connected components of the adjacency matrix *A*. At this point, one iteration has been completed. By repeating this merger process according to (4), all clusters will eventually merge into one cluster and form a hierarchical tree. Each layer of the hierarchical tree can be regarded as a partition under a specific granularity.

Each connection (i.e., merger) C_j has two intuitive properties. One of the properties is the product of the mass of the two clusters it connects

$$M_j = mass\left(\zeta_j\right)\times mass\left(\zeta_j^N\right), \tag{6}$$

The other is the square of the distance between the two clusters it connects

$$S_j = d^2\left(\zeta_j,\zeta_j^N\right). \tag{7}$$

Plotting all the connections on a two-dimensional graph of the two properties, called the decision graph. By observing the decision graph and finding the connections with relatively large M_j and S_j, remove these connections to get the final reasonable partition.

CNNC is parameter-free. A reasonable partition can be obtained through a certain layer (granularity) of the clustering tree, or it can be obtained by observing the decision graph and removing the connections with relatively large M_j and S_j. However, CNNC can also be assigned the desired number of clusters *K*. After simply removing *K*-1 connections with relatively large $M_j \times S_j$, then we can get a partition containing *K* clusters. On the other hand, in each iteration, CNNC only needs to find the first neighbor of each cluster. The first neighbors can be found effectively via fast approximate nearest neighbor methods, e.g., k-d tree. Therefore, the complexity of the algorithm can be reduced to $O(nlogn)$. Compared with traditional hierarchical clustering algorithms, CNNC has a lower computational overhead.

Exploiting CNNC to perform clustering based on each fusion distance matrix in FDEW, then v+1 partitions $P^{(r)}$ can be obtained, where $\{P^{(r)}\}_{r=1}^{v+1}=\{P^{(1)},P^{(2)},\ldots,P^{(v)},P^{(v+1)}\}$. So which partition is the best? This requires an evaluation index to evaluate the clustering quality of each partition.

C. Internal Evaluation Index Based on Rawls' Max-Min Criterion (MMI)

In practice, the ground-truth labels are often not known in advance [47]. Therefore, we cannot objectively judge which partition from $\{P^{(r)}\}_{r=1}^{v+1}$ is the best. A simple idea is to use internal evaluation indices to evaluate each partition to find

the best one, which is an unsupervised manner. Most of the past internal evaluation indices need to know cluster centers of partition [48]. However, CNNC does not output specific cluster centers. Here we propose a new internal clustering evaluation index based on distance matrix to select the best partition based on Rawls' max-min criterion [39], which is called Max-Min Index (MMI). MMI is based on two existing theories, the first is a common criterion in clustering internal evaluation indices: the best partition has a relatively large inter-cluster distance and a relatively small intra-cluster distance; the second is the Rawls' max-min criterion in economics: the right decision is that which maximizes the minimum outcome.

For a partition $P^{(r)}$, 1) we arbitrarily select two clusters, and then arbitrarily select a sample from each cluster, and use the distance between the two samples as the inter-class distance; 2) We again, choose a cluster arbitrarily, and use the average of the distance between any two samples in this cluster as the intra-class distance. Based on 1) and 2), we first define an initial evaluation index: $\forall x_a^{(r)} \in \forall \zeta_k^{(r)}, \forall x_b^{(r)} \in \forall \zeta_l^{(r)}$; $\forall x_c^{(r)}, x_d^{(r)} \in \forall \zeta_m^{(r)}$,

$$I^{(r)} = \frac{d\left(x_a^{(r)}, x_b^{(r)}\right)}{\frac{2}{\left|\zeta_m^{(r)}\right|\left(\left|\zeta_m^{(r)}\right|-1\right)} \sum d\left(x_c^{(r)}, x_d^{(r)}\right)}, \tag{8}$$

The larger $I^{(r)}$, $P^{(r)}$ may have a larger inter-class distance and a smaller intra-class distance, but it is not certain. This is because we randomly select clusters and samples when calculating $I^{(r)}$, which may not be representative. According to Rawls' max-min criterion, the right decision is that which maximizes the minimum outcome. Inspired by the max-min criterion, we first calculate the minimum value of $I^{(r)}$:

$$\begin{aligned} \min\left(I^{(r)}\right) &= \min\left\{\frac{d\left(x_a^{(r)}, x_b^{(r)}\right)}{\frac{2}{\left|\zeta_m^{(r)}\right|\left(\left|\zeta_m^{(r)}\right|-1\right)} \sum d\left(x_c^{(r)}, x_d^{(r)}\right)}\right\} \\ &= \frac{\min\limits_k \min\limits_l \min\limits_{\mathrm{x_a^{(r)}} \in \zeta_k^{(r)}, \mathrm{x_b^{(r)}} \in \zeta_l^{(r)}} d\left(x_a^{(r)}, x_b^{(r)}\right)}{\max\limits_m \frac{2}{\left|\zeta_m^{(r)}\right|\left(\left|\zeta_m^{(r)}\right|-1\right)} \sum_{\mathrm{x_c^{(r)}}, \mathrm{x_d^{(r)}} \in \zeta_m^{(r)}} d\left(x_c^{(r)}, x_d^{(r)}\right)}, \end{aligned} \tag{9}$$

Furthermore, we believe that $P^{(r)}$ that maximizes $min(I^{(r)})$ is the best, that is

$$\begin{aligned} s &= \underset{r}{\mathrm{argmax}} \min\left(I^{(r)}\right) = \\ & arg\max_r \frac{\min\limits_k \min\limits_l \min\limits_{\mathrm{x_a^{(r)}} \in \zeta_k^{(r)}, \mathrm{x_b^{(r)}} \in \zeta_l^{(r)}} d\left(x_a^{(r)}, x_b^{(r)}\right)}{\max\limits_m \frac{2}{\left|\zeta_m^{(r)}\right|\left(\left|\zeta_m^{(r)}\right|-1\right)} \sum_{\mathrm{x_c^{(r)}}, \mathrm{x_d^{(r)}} \in \zeta_m^{(r)}} d\left(x_c^{(r)}, x_d^{(r)}\right)}. \end{aligned} \tag{10}$$

Therefore, we can finally determine that $P^{(s)}$ is the best partition by (10). Compared with other distance matrix-based internal indices, the proposed MMI is more accurate in selecting the best partition (see Table X). The method proposed in [49] can be utilized to reduce the computational complexity of MMI.

D. Algorithm of MCHC and MCHC-PF

For Multi-view data $\{X^{(i)}\}_{i=1}^{v}$, we first calculate $\{FDEW^{(r)}\}_{r=1}^{v+1}$ according to (1)–(3). Then we use CNNC to perform clustering based on each $FDEW^{(r)}$, and get $\{P^{(r)}\}_{r=1}^{v+1}$. Finally, we select the best partition $P^{(s)}$ in $\{P^{(r)}\}_{r=1}^{v+1}$ according to (8)–(10). Fig. 2 shows the simple flowchart of MCHC, and Algorithm 1 shows the pseudo code of MCHC.

In real life, the correct number of clusters is often not known in advance. Therefore, we provide a parameter-free version of MCHC (MCHC-PF). Algorithm 2 gives the pseudo code of MCHC-PF. MCHC-PF does not require any parameters, can provide several partitions at different granularity levels, and draw a decision graph according to (6)–(7) for users to estimate a reasonable number of clusters. On the other hand, MCHC -PF only uses CNNC to perform clustering based on the fusion distance matrix with equal weights D^* from FDEW, because merging multiple similarity (or dissimilarity) matrices with equal weights is an intuitive and simple processing way for multi-view data [23], [50], [51]. Moreover, we exploit k-d tree to calculate the fusion distance matrix D^* approximately, so it has a shorter runtime than MCHC. Fig. 3 shows the decision graph of MCHC-PF on the dataset UCI-digits (this data set will be introduced in the experimental part). It can be clearly seen that there are 9 connections (mergers) with larger M_j and S_j. Remove them in the adjacency matrix, and we can get the correct 10 clusters, which matches the ground truth.

There are three main differences between MCHC and MCHC-PF. First, MCHC not only uses complementary information among multiple views, but exploits the information from each single view. However, MCHC-PF only exploits the complementary information among multiple views. Second, MCHC uses naive way to calculate distance matrix for each view, while MCHC-PF uses k-d tree to approximate the calculation to obtain a sparse distance matrix. Third, like most existing multi-view clustering methods, MCHC needs to be set target number of clusters. However, MCHC-PF does not need to set this parameter. It provides multiple clustering results at different granularity levels for users to choose according to specific scenarios.

E. Complexity Analysis

We first analyze the complexity of MCHC. According to Algorithm 1, Steps 3–7 costs $O(vn^2)$, where v is the number of views. When the distance matrix is known, the cost of CNNC is $O(n)$. Therefore, the cost of Steps 8–19 is $O((v+1)n)$. Steps 20–23 costs $O((v+1)K^2)$ for MMI calculation and best partition finding, where K is the target number of clusters. In summary, the total cost of MCHC is $O(vn^2) + O((v+1)n) + O((v+1)K^2)$, approximately $O(n^2)$. Compared to $O(n^3)$ of most multi-view spectral clustering or subspace clustering methods, the complexity of MCHC is acceptable.

For MCHC-PF, we leverage the k-d tree to compute the sparse distance matrix for each view, so Steps 3–6 costs $O(vnlogn)$. Because MCHC-PF only runs CNNC on the fusion distance

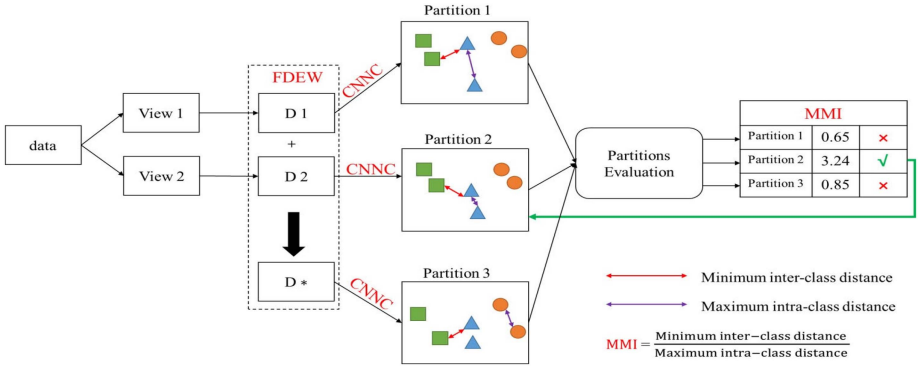


Fig. 2. The flowchart of the proposed multi-view adjacency-constrained hierarchical clustering (MCHC). We take the dataset containing two views as an example. First, we calculate the distance matrix of each view by (3) to get D1 and D2. Then we calculate $D*$ by (2). D1, D2, and $D*$ together form fusion distance matrices with extreme weights (FDEW). Next, based on each distance matrix in FDEW, we exploit adjacency-constrained nearest neighbor clustering (CNNC) to obtain three partitions (i.e., partition 1, partition 2, and partition 3). Finally, we choose the most reasonable partition (i.e., partition 2) based on MMI (i.e., (8)–(10)).

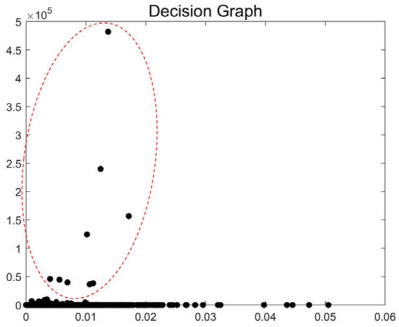


Fig. 3. Decision graph of MCHC-PF on UCI-digits dataset. The horizontal axis represents the property M_j of each connection in the CNNC, and the vertical axis represents the property S_j of each connection. The nine connections in the red circle have relatively large M_j and S_j. Remove them to leave 10 clusters, which exactly matches the ground-truth.

matrix with equal weights D^*, the cost of Steps 7–16 is approximately $O(n)$. Therefore, the total cost of MCHC-PF is $O(nlogn)$.

Algorithm 1: Algorithm of the Proposed MCHC.

Input: Multi-view data $\{X^{(i)}\}_{i=1}^{v}$ and the target number of clusters *K*.
Output: Best partition $P^{(s)}$.
for i=1:v **do**
  Calculating distance matrix $D^{(i)}$ by Eq. (3).
end
Calculating D^* by Eq. (2).
Combine $D^{(i)}$ and D^* to get FDEW.
for r=1:v+1 **do**
Initializing adjacency matrix *A*.
Constructing cluster sets $\{\zeta_j\}$ (Initially, regard each sample as a cluster).
while cluster sets $\{\zeta_j\}$ have more than two clusters **do**
Searching the nearest cluster of ζ_j with higher mass according to $FDEW^{(r)}$.
Updating *A* by Eqs. (4)–(5) (Using two nearest samples respectively from two clusters to represent these two clusters).
Calculating M_j and S_j of C_j by Eqs. (6)–(7).
Updating cluster sets $\{\zeta_j\}$ based on *A*.
end
Updating *A* by removing *K*-1 C_j with largest $M_j \times S_j$.
Getting partition $P^{(r)}$ based on *A*.
end
for r=1:v+1 **do**
  Calculating $min(I^{(r)})$ by Eqs. (8)–(9).
end
Finding best partition $P^{(s)}$ by Eq. (10).

III. Experiments and Results

In this part, we conduct several experiments to show the superiority of MCHC and MCHC-PF.

A. Datasets Description

1) **100-leaves**: There are 1600 samples in the 100-leaves dataset (https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set), divided into 100 categories. The original 100-leaves photos are also different in size. There are three views which display samples from several angles using shape descriptors, fine scale margins, and texture histogram characteristics. The full statistics of these datasets are shown in Table I.
2) **UCI-digits**: The UCI-digit dataset can be found in the UCI repository (https://archive.ics.uci.edu/ml/index.php). The digits (0–9) in this collection were extracted from Dutch utility maps in 2000 samples. Each class contains 200 samples, each of which is represented by six feature sets. We employed three feature sets following [52]: 76 character shape Fourier coefficients, 216 profile correlations, and 64 Karhunen- Loève coefficients.
3) **COIL20**: This dataset has 1440 grayscale photos of 20 different objects [53]. Each image is downscaled to 32 by

Algorithm 2: Algorithm of the Proposed MCHC-PF.

Input: Multi-view data $\{X^{(i)}\}_{i=1}^{v}$.
Output: partitions at different granularity levels $\{R_t\}$.
for i=1:v **do**
  Calculating (sparse) distance matrix $D^{(i)}$ by Eq. (3).
end
Calculating D^* by Eq. (2).
Initializing adjacency matrix A.
Constructing cluster sets $\{\zeta_j\}$ (Initially, regard each sample as a cluster).
while cluster sets $\{\zeta_j\}$ have more than two clusters **do**
Searching the nearest cluster of ζ_j with higher mass according to D^*.
Updating A by Eqs. (4)–(5) (Using two nearest samples respectively from two clusters to represent these two clusters).
Getting partition R_t at current granularity level based on A.
Calculating M_j and S_j of C_j by Eqs. (6)–(7).
Updating cluster sets $\{\zeta_j\}$ based on A.
end
Plotting decision graph by M_j and S_j.

TABLE I
STATISTICS OF MULTI-VIEW DATASETS

Datasets	#Views	#Samples	#Clusters
100-leaves	3	1600	100
UCI-digits	3	2000	10
COIL20	3	1440	20
Handwritten	2	2000	10
ORL	3	400	40
UMIST	3	575	20
CMU-PIE	3	2856	68
COIL100	3	7200	100

32 pixels for the original features scenario. Three types of features are extracted in the case of several hand-crafted features: Intensity, LBP, and Gabor. The sizes of their features are 1024, 3304 and 6750, respectively.

4) **Handwritten**: This consists of 2000 samples from 10 digits, ranging from 0 to 9. Each sample is represented by two views, the first of which is a feature vector with 240 features derived from the average of pixels in 2×3 windows, and the second of which is a Fourier coefficient vector with 76 features [54].
5) **ORL**: This is made up of 400 photos of 40 people's faces. Following [9], each image is down-sampled to 32 by 32 pixels for the original features scenario. Each image in the handcrafted features scenario is represented by three types of features (4096 Intensity, 3304 LBP and 6750 Gabor).
6) **UMIST**: This collection [55] contains 564 photos of 20 people (mixed race, gender, and appearance). Each person is depicted in a variety of poses, from profile to frontal perspectives. Each image has a resolution of about 220×220 pixels and a 256-bit greyscale. Following [56], each image is represented by three heterogeneous feature sets:30 isometric projection (ISO), 30 principal component analysis (PCA), and 30 neighborhood preserving embedding (NPE).
7) **CMU-PIE**: This dataset [57] contains 2856 frontal-face photos of 68 persons, with 42 distinct illuminations for each object. Each photograph was cropped to a size of 32×32 pixels. Three feature sets are used to express each image: 30 ISO, 30 PCA, and 30 NPE. Fig. 4 shows some sample face images from the CMU-PIE database.
8) **COIL-100**: This dataset [58] is a library of 7200 color images representing 100 different types of objects. Each image is 128×128 pixels in size. Each object has 72 distinct photos in various positions. Each image is expressed using three feature sets: 30 ISO, 30 PCA, and 30 NPE.

Fig. 4. Sample face images from the CMU-PIE database. MCHC achieves 100% accuracy on this dataset.

B. Compared Algorithms

We compare MCHC and MCHC-PF with 13 state-of-the-art multi-view clustering algorithms. They include: K-means; Multi-view low-rank sparse subspace clustering (MLRSSC) [6]; Graph-based multi-view clustering (GMC) [15]; Unified graph learning for multi-view clustering (UGLMC) [12]; Constrained bilinear factorization multi-view subspace clustering (CBF-MSC) [28]; View variation and view heredity for incomplete multi-view clustering (V3H) [59]; Large-scale multi-view subspace clustering (LMVSC) [30]; Affinity aggregation for spectral clustering (AASC) [7]; Multi-view clustering via adaptively weighted Procrustes (AWP) [14]; Co-regularized multi-view spectral clustering (CoReg) [11]; Multi-view consensus graph clustering (MCGC) [60]; Robust multi-view spectral clustering (RMSC) [61]; and Weighted multi-view spectral clustering (WMSC) [21]. Since the two multi-view hierarchical clustering methods [23], [38] related to the proposed methods do not provide open-source code, we don't include them in the comparison for the sake of objectivity. We employed three widely used external clustering validation indices to evaluate the performance of clustering algorithms: Accuracy (ACC), Normalized mutual information (NMI) [62], and F-score [63]. The best and second-best clustering results are highlighted and underlined respectively. For K-means, a single-view clustering algorithm, we reported its best clustering results of multiple views. For other multi-view clustering algorithms, the parameters are tuned as suggested in the original papers to generate the best results. For more details, please refer to the supplementary materials. All experiments were conducted on a workstation with two 14-core Intel Xeon 6132 CPUs clocked at 2.6 GHz and 3.7GHz and 96GB memory. Our code is available: https://github.com/brucejak/Multi-View-Adjacency-Constrained-Hierarchical-Clustering.

TABLE II
CLUSTERING RESULTS OF MCHC IN THE METRIC OF **ACC**

Sources	Methods	100-leaves	UCI-digits	COIL20	Handwritten	ORL	UMIST	CMU-PIE	COIL100
-	K-means	.5780	.6814	.6410	.6921	.5703	.4617	.5377	.5737
PR-18	MLRSSC	.7199	.8773	.6497	.7895	.6529	.4517	.7637	.6473
TKDE-20	GMC	.8237	.8495	.7910	.8300	.6325	.5217	.7048	.7692
ICDM-19	UGLMC	.7881	.8825	.9014	.7425	.6675	.6087	.1912	.7240
KBS-20	CBF-MSC	.7823	.9324	.7449	.8945	.6222	.4701	.7553	.5570
TAI-21	V3H	.8237	.9051	.6012	.8669	.7412	.5294	.7231	.6514
AAAI-20	LMVSC	.6575	.8935	.7569	.9005	.6300	.4696	.4769	.5622
CVPR-12	AASC	.8581	.8505	.7924	.8335	.7375	.4348	.5711	.6593
KDD-18	AWP	.7856	.8670	.6757	.9325	.6900	.5217	.8120	.7443
NeurIPS-11	CoReg	.8456	.9560	.8472	.9110	.8200	.5043	.7507	.7965
TIP-18	MCGC	.6244	.4920	.4389	.1005	.6825	.4643	.7059	.4771
AAAI-14	RMSC	.7781	.2495	.3896	.4120	.7975	.4539	.7328	.2258
AAAI-18	WMSC	.8769	.8410	.8465	.8335	.8300	.4539	.6590	.7039
-	**MCHC-PF**	.6931	.6820	.8236	.8805	.8100	.6557	.8883	.8385
-	**MCHC**	**.8888**	**.9655**	**.9250**	**.9795**	**.8700**	**.8678**	**1**	**.9087**

TABLE III
CLUSTERING RESULTS OF MCHC IN THE METRIC OF **NMI**

Sources	Methods	100-leaves	UCI-digits	COIL20	Handwritten	ORL	UMIST	CMU-PIE	COIL100
-	K-means	.7996	.7025	.8004	.7071	.7784	.6771	.7990	.8239
PR-18	MLRSSC	.8694	.8516	.7170	.7586	.8182	.6557	.9025	.8401
TKDE-20	GMC	.9296	.9013	.9410	.8767	.8590	.7373	.8892	.9371
ICDM-19	UGLMC	.9171	.9231	.9705	.8505	.8560	.8375	.4003	.9302
KBS-20	CBF-MSC	.9046	.8685	.8165	.8138	.7986	.6737	.8971	.7937
TAI-21	V3H	.9096	.8118	.7639	.7425	.8633	.6859	.8667	.8656
AAAI-20	LMVSC	.8504	.8321	.8404	.8366	.8246	.6904	.6916	.8117
CVPR-12	AASC	**.9501**	.9025	.9004	.8832	.8619	.6469	.8018	.8676
KDD-18	AWP	.8968	.8949	.9148	.9072	.8529	.7331	.9269	.9264
NeurIPS-11	CoReg	.9346	.9188	.9548	.8811	.9011	.7405	.8791	.9269
TIP-18	MCGC	.7685	.7397	.6973	.0350	.8199	.6895	.8069	.7514
AAAI-14	RMSC	.8963	.3450	.7118	.4858	.8978	.6562	.8423	.5091
AAAI-18	WMSC	.9481	.8839	.9486	.8772	.8985	.7015	.8571	.9016
-	**MCHC-PF**	.9146	.8407	.9452	.9214	.9228	.8486	.9666	.9663
-	**MCHC**	.9466	**.9276**	**.9826**	**.9535**	**.9404**	**.9262**	**1**	**.9816**

TABLE IV
CLUSTERING RESULTS OF MCHC IN THE METRIC OF **F-Score**

Sources	Methods	100-leaves	UCI-digits	COIL20	Handwritten	ORL	UMIST	CMU-PIE	COIL100
-	K-means	.4662	.6331	.6125	.6356	.4547	.4078	.4744	.5164
PR-18	MLRSSC	.6368	.8301	.5738	.7247	.5450	.3827	.7213	.5974
TKDE-20	GMC	.5042	.8426	.7943	.8113	.3599	.4620	.6171	.7195
ICDM-19	UGLMC	.7383	.8709	.8696	.7547	.5868	.5710	.0337	.5241
KBS-20	CBF-MSC	.7159	.8716	.7025	.8039	.5191	.3967	.7148	.4881
TAI-21	V3H	.7470	.8238	.5626	.7612	.6575	.4489	.6248	.5902
AAAI-20	LMVSC	.5656	.8127	.7010	.8166	.5354	.4182	.3417	.4926
CVPR-12	AASC	.6589	.8440	.7861	.8171	.6341	.3380	.3975	.4608
KDD-18	AWP	.7096	.8455	.7240	.8899	.6115	.4841	.7815	.7267
NeurIPS-11	CoReg	.7950	.9154	.8365	.8587	.7423	.4668	.6775	.7755
TIP-18	MCGC	.1103	.5298	.2278	.1810	.3654	.3442	.2412	.1056
AAAI-14	RMSC	.5667	.2161	.3083	.2887	.7328	.2780	.3912	.0497
AAAI-18	WMSC	**.8376**	.8315	.8336	.8187	.7429	.3858	.5992	.6856
-	**MCHC-PF**	.6228	.6718	.8311	.8715	.7560	.6461	.8829	.8605
-	**MCHC**	.8212	**.9323**	**.9389**	**.9591**	**.8051**	**.8509**	**1**	**.9174**

C. Experimental Results and Analysis

Tables II, III, and IV show the clustering results and Fig. 5 give the mean rankings for all multi-view clustering methods on all datasets. In essence, the proposed MCHC outperforms all other clustering methods. Whether compared with single-view clustering algorithms or multi-view clustering algorithms, MCHC shows unparalleled performance advantages. In particular, for the metric ACC, the results of our MCHC are about 4.7%, 4%, 25.9%,18.8% and 11.2% higher than the second-best (except for MCHC-PF) clustering results on Handwritten, ORL, UMIST, CMU-PIE and COIL100 dataset, respectively. For the metric NMI, the results of our MCHC are about 4.6%, 3.9%, 8.9%,7.3% and 4.5% higher than the second-best clustering results on Handwritten, ORL, UMIST, CMU-PIE and COIL100 dataset, respectively. Finally, for the metric F-score,

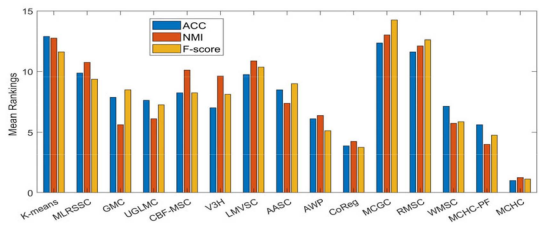


Fig. 5. Mean rankings for all multi-view clustering methods on all datasets.

TABLE V
CLUSTERING RESULTS OF MCHC-PF AT DIFFERENT GRANULARITY LEVELS. #C MEANS THE GROUND-TRUTH NUMBER OF CLUSTERS, AND NC MEANS THE NUMBER OF CLUSTERS

Datasets	#C	NC at different granularity levels	Closest NC
100-leaves	100	{391, 155, 76, 33, 14, 5, 1}	76
UCI-digits	10	{429, 134, 45, 17, 7, 3, 1}	7
COIL20	20	{414, 181, 88, 47, 24, 13, 5, 3, 1}	24
Handwritten	10	{397, 125, 41, 17, 9, 3, 2, 1}	9
ORL	40	{113, 44, 17, 6, 3, 2, 1}	44
UMIST	20	{184, 84, 42, 21, 11, 7, 4, 1}	21
CMU-PIE	68	{972, 397, 154, 78, 33, 12, 4, 3, 1}	78
COIL100	100	{2148, 924, 469, 251, 127, 52, 14, 6, 2, 1}	127

the results of our MCHC are about 6.9%, 6.9%, 6.2%, 28%, 21.9% and 14.2% higher than the second-best clustering results on COIL20, Handwritten, ORL, UMIST, CMU-PIE and COIL100 dataset, respectively.

Unlike the other 14 multi-view clustering methods (including MCHC) that require at least the ground-truth number of clusters to be set, MCHC-PF can give natural partitions at different granularity levels without any parameters. Table V shows the number of clusters obtained at different granularity levels. For most datasets, the clustering results obtained by MCHC-PF can yield a relatively accurate number of clusters. The clustering results of MCHC-PF in Tables II, III, and IV are based on the number of clusters that are closest to the ground truth. The clustering performance of MCHC-PF is worse than that of MCHC, because MCHC-PF only considers the complementary information (i.e., D^*) from multiple views, not the information from each single view. However, as Fig. 5 shows, compared with all other methods, MCHC-PF still achieves competitive results. Particularly, on the five datasets (i.e., Handwritten, ORL, UMIST, CMU-PIE and COIL100), in terms of the metric NMI, the performance of MCHC-PF is better than that of other 13 state-of-the-art multi-view clustering algorithms.

From a theoretical point of view, the reasons why the performance of most multi-view spectral or subspace clustering methods is not competitive to MCHC are mainly due to the following two aspects. First, the backbones of these methods are spectral clustering or subspace clustering, which have inherent limitations. For example, it is hard for spectral clustering to accurately capture the intrinsic manifold structure in data when constructing the k-nearest neighbor similarity graph. However, CNNC in MCHC can catch it more accurately due to its constrained way of merging. Second, when conflicting views exist in multi-view data, performing clustering on the information from a specific single view may achieve better results than that on the complementary information from multiple views [40], [43]. Most compared methods only consider the complementary information from multi-view data. However, MCHC not only uses complementary information among multiple views but exploits the information from each single view, achieving better results.

TABLE VI
RUNTIME (IN SECONDS) COMPARISON WITH THREE BEST COMPARED ALGORITHMS

Datasets	AWP	CoReg	V3H	**MCHC**	**MCHC-PF**
100-leaves	0.87	14.60	687.10	2.19	**0.60**
UCI-digits	**0.59**	11.20	876.83	2.99	0.92
COIL20	**2.47**	10.26	412.52	4.16	2.79
Handwritten	**0.40**	7.79	617.86	2.51	0.75
ORL	0.18	2.69	155.72	0.38	**0.12**
UMIST	0.27	3.72	22.67	0.53	**0.15**
CMU-PIE	**2.63**	92.72	1568.32	12.14	3.07
COIL100	26.77	1276.28	14584.9	84.74	**20.35**

The lowest runtimes are marked bold.

D. Runtime

To show the effectiveness of our method, we compare the runtime of the proposed MCHC and MCHC-PF to three best-compared algorithms from the 13 state-of-the-art multi-view clustering algorithms. The three best-compared algorithms include AWP, CoReg and V3H. According to Table VI, the proposed MCHC-PF can produce clustering results in roughly 20 seconds for all datasets. The total runtime of MCHC-PF on all datasets is the lowest, which is significantly lower than that of CoReg, V3H, and MCHC. The total runtime of AWP is close to that of MCHC-PF because it omits the computation of eigenvalue decomposition when obtaining spectral embedding [14]. While V3H performs the complete calculation and optimization of spectral embedding, resulting in excessive computational overhead. However, the clustering performance of MCHC is better than that of all other clustering algorithms (see Tables II–IV).

IV. ABLATION STUDY

A. Impact of Fusion Distance Matrices With Extreme Weights (FDEW)

According to (1)–(3), FDEW not only uses the fusion distance matrix with equal weights (i.e., D^*) containing complementary information from multiple views, but exploits the distance matrix $D^{(i)}$ of each view, which includes the information from each single view. In this section, we explore the results of CNNC on each distance matrix in FDEW to show the necessity of including these two pieces of information. As Table VII shows, partitions 1–3 denotes the results of CNNC on the distance matrix $D^{(i)}$ of each single view, and partition$*$ means the results of CNNC on the fusion distance matrix with equal weights (i.e., D^*). Here, we exploit NMI to evaluate the performance of each partition.

TABLE VII
THE RESULTS OF CNNC ON EACH DISTANCE MATRIX IN FDEW USING THE METRIC OF **NMI**

Datasets	Partition1	Partition2	Partition3	Partition$*$
100-leaves	.8020	.6550	.7577	**.9466**
UCI-digits	.7711	.7697	.9163	**.9276**
COIL20	.9555	**.9826**	.9769	.9662
Handwritten	.9186	.7697	-	**.9535**
ORL	.8175	**.9404**	.8912	.9309
UMIST	.7677	**.9262**	.8102	.8854
CMU-PIE	.8946	.9432	**1**	.9801
COIL100	.8913	**.9816**	.9159	.9782

The best and MMI-selected clustering results are highlighted and underlined respectively.

The best and MMI-selected clustering results are highlighted and underlined respectively. On the one hand, the best clustering results are from partitions 1–3 on some datasets (e.g., COIL20 and ORL), or from partition$*$ on some datasets (e.g., 100-leaves and UCI-digits).

This phenomenon is consistent with the previous studies [40]. One of the reasons for this is the existence of conflicting views in the multi-view data [43], making the complementary information of multiple views with lower discrimination for classification than a specific single view information. Therefore, it is necessary to consider both the information from each single view and the complementary information from multiple views. On the other hand, MMI can accurately select the best one from several partitions.

B. Impact of Adjacency-Constrained Nearest Neighbor Clustering (CNNC)

To show the superiority of CNNC in the proposed model, on the one hand, we explore CNNC's clustering performance advantage by comparing it with other state-of-the-art single-view hierarchical clustering methods or nearest neighbor clustering (NNC) methods. Seven well-known hierarchical clustering methods are used including single-linkage, complete-linkage, average-linkage, ward-linkage, centroid-linkage, median-linkage, weighted-linkage. Three recent NNC methods, GDL [64], SNNDPC [65] and Finch [37], are used for comparison. It is worth mentioning that MHC, one of the related works, uses the Finch method as a backbone to complete multi-view hierarchical clustering [23]. For each of the eight multi-view datasets described above, we concatenate all the features of multiple views, and then perform CNNC and other baselines directly on the concatenation. As Table VIII shows, compared with other state-of-the-art single-view clustering methods, CNNC achieves the best results on all datasets.

On the other hand, in the MCHC framework, we replace CNNC with other hierarchical clustering or NNC methods, and keep other components in the framework unchanged. The generated new multi-view hierarchical clustering (MHC) methods are named as MHC-single, MHC-complete, MHC-average, MHC-ward, MHC-centroid, MHC-median, MHC-weighted and MHC-GDL, respectively. MHC-SNNDPC and MHC-Finch have been removed, because SNNDPC and Finch need to know the coordinates of data points, and the coordinates of data points corresponding to D^* are unknown. We perform these multi-view hierarchical clustering methods on the eight multi-view datasets. According to Table IX, MCHC still achieves the best performance on all datasets.

From the above two experiments, CNNC has better performance than previous NNC methods or hierarchical clustering methods, whether processing single-view data or multi-view data. This is because the constrained merging way of CNNC can more accurately capture the manifold structure in the data.

C. Impact of Internal Evaluation Index Based on Rawls' Max-Min Criterion (MMI)

To show the validity of the Internal evaluation index based on Rawls' max-min criterion (MMI), we exploit four other distance matrix-based internal indices, including Dunn Index (DI) [66], Silhouette index (Sil) [67], Clustering Validation index based on Nearest Neighbors (CVNN) [68] and Clustering Validity index based on Density-involved Distance (CVDD) [69] to select the best partition in the MCHC framework, and keep other components in the framework unchanged. DI and Sil are two classic internal validity indices, and CVNN and CVDD are the recent ones. After that, the metric NMI is used to objectively evaluate the selected partition based on the ground-truth labels. According to Table X, the proposed MMI can select the best partition on all datasets, whereas the other four distance matrix-based internal indices cannot.

From a theoretical point of view, the other four distance matrix-based internal indices all have some inherent flaws. For example, DI exploits the distance between the two farthest points in cluster as the intra-class distance, which is obviously susceptible to outliers. Additionally, Sil does not adapt well to non-spherical datasets, and CVDD is susceptible to changing densities in clusters [68].

V. DISCUSSION

A. Theoretical Significance of the Combination of Three Components in the Proposed MCHC

The proposed MCHC consists of three main components: including FDEW, CNNC, and MMI. In this section, we will explain why FDEW, CNNC, and MMI are combined into MCHC, that is, the theoretical significance of the combination of these three components. First, this is determined by the generalized paradigm of multi-view clustering. Almost all multi-view clustering methods first learn complementary (or consensus) information from multi-view data, and then use a single-view clustering method for post-processing of the complementary information. The proposed MCHC framework follows this paradigm. Second, another significance for combining these three components is to inherit their respective advantages. For example, FDEW alleviates the poor impact of conflicting views; CNNC can capture the manifolds in data and improve clustering accuracy compared to traditional NNC methods; MMI can select the best one from several partitions in

TABLE VIII
PERFORMANCE COMPARISON FOR CNNC AND OTHER HIERARCHICAL CLUSTERING OR NNC METHODS, IN THE METRIC OF **NMI**

Methods	100-leaves	UCI-digits	COIL20	Handwritten	ORL	UMIST	CMU-PIE	COIL100
Single-linkage	.5343	.0348	.9708	.0370	.4951	.6456	.7586	.8847
Complete-linkage	.8159	.4727	.7693	.5383	.7221	.5836	.4708	.7753
Average-linkage	.8477	.4958	.7745	.6046	.7221	.6461	.4634	.7899
Ward-linkage	.8634	.6204	.8664	.7022	.7889	.6228	.5563	.8242
Centroid-linkage	.8234	.5387	.6454	.4013	.5784	.5788	.3790	.7727
Median-linkage	.8054	.4286	.6713	.1106	.6446	.5998	.4595	.7918
Weighted-linkage	.8318	.5071	.7784	.5510	.7324	.6276	.5324	.8033
GDL	.8890	.7623	.9318	.8871	.7769	.7488	.7916	.9442
SNNDPC	.6875	.6733	.8035	.8200	.6363	.6570	.4243	.6986
Finch	.8811	.6002	.8377	.8111	.5729	.6252	.3959	.8435
CNNC	**.9072**	**.7715**	**.9715**	**.9186**	**.8181**	**.9221**	**.9455**	**.9816**

TABLE IX
PERFORMANCE COMPARISON WITH OTHER MULTI-VIEW HIERARCHICAL CLUSTERING METHODS, IN THE METRIC OF **NMI**. THE AUTHOR-PROVIDED CODE FOR GDL BREAKS ON THE COIL100 DATASET

Methods	100-leaves	UCI-digits	COIL20	Handwritten	ORL	UMIST	CMU-PIE	COIL100
MHC-single	.6215	.0348	.9708	.0348	.6742	.6693	.9383	.8847
MHC-complete	.8731	.5666	.7504	.6289	.8245	.5842	.6827	.7724
MHC-average	.8885	.5878	.7854	.5878	.8611	.6524	.7171	.8506
MHC-ward	.9219	.7041	.8954	.7041	.8937	.6276	.7335	.8319
MHC-centroid	.8645	.5549	.5019	.0347	.3486	.5148	.7093	.7727
MHC-median	.8618	.3585	.6980	.3015	.4279	.5633	.6977	.7905
MHC-weighted	.8774	.6596	.8009	.6693	.8306	.6663	.7118	.8067
MHC-GDL	.9325	.9010	.9374	.9141	.8983	.7573	.8518	-
MCHC	**.9466**	**.9276**	**.9826**	**.9535**	**.9404**	**.9262**	**1**	**.9816**

TABLE X
PERFORMANCE COMPARISON WITH OTHER DISTANCE MATRIX-BASED INTERNAL INDICES, IN THE METRIC OF **NMI**

Datasets	Best partition	DI	Sil	CVNN	CVDD	MMI
100-leaves	.9466	**.9466**	**.9466**	.8020	**.9466**	**.9466**
UCI-digits	.9276	**.9276**	.7711	.7711	.7711	**.9276**
COIL20	.9826	**.9826**	.9769	.9769	.9662	**.9826**
Handwritten	.9535	**.9535**	**.9535**	.7697	**.9535**	**.9535**
ORL	.9404	**.9404**	.8912	.8175	.8175	**.9404**
UMIST	.9262	.7677	.8102	**.9262**	**.9262**	**.9262**
CMU-PIE	1	.8946	**1**	**1**	**1**	**1**
COIL100	.9816	**.9816**	.8913	.8913	**.9816**	**.9816**

an unsupervised manner, no additional manual intervention is required.

B. New Thinking for Multi-View Clustering

Firstly, most previous multi-view clustering methods focus on exploring the different importance of each view to learn an optimal clustering-friendly representation. For example, multi-view clustering methods based on graph learning exploit some optimization techniques for joint modeling the consistency and inconsistency of multiple views [12]. Unlike most previous methods, the proposed MCHC only employs several linearly combined distance matrices (i.e., FDEW) to express the consistency or inconsistency of multiple views, reducing the computational overhead simultaneously. According to (1)–(3), FDEW not only uses the fusion distance matrix with equal weights (i.e., D^*) containing the consistent information from multiple views, but exploits the distance matrix $D^{(i)}$ of each view, reflecting different kinds of view-specific characteristics (i.e., inconsistency). This approach is based on the viewpoint in [40]: sometimes the utilization of multiple views may even deteriorate the final performance, which is even worse than the performance of the best single view.

Second, most existing multi-view clustering algorithms are based on the existing backbone of spectral clustering or subspace clustering, ignoring the optimization of the clustering mechanism. Instead, this study pays more attention to the optimization of the clustering mechanism (i.e., CNNC). Even based on representations with extreme weights (i.e., FDEW), the proposed frameworks still achieve state-of-the-art performance, which provides new thinking for multi-view clustering.

VI. CONCLUSION AND FUTURE WORK

This paper proposes a Multi-view adjacency-Constrained Hierarchical Clustering (MCHC) and its parameter-free

version (MCHC-PF). By introducing the fusion distance matrices with extreme weights, adjacency-constrained nearest neighbor clustering and the internal evaluation index based on Rawls' Max-Min criterion, the promising clustering performance can be obtained by MCHC. Furthermore, without any parameter selection, MCHC-PF can provide partitions at different granularity levels with a low time complexity. Extensive experiments conducted on eight real-world datasets illustrate the superior performance of the proposed methods.

Compared to the proposed MCHC-PF, MCHC has a higher time complexity, i.e., $O(n^2)$. However, MCHC has better clustering performance than MCHC-PF. In future work, we will further combine MCHC with the bipartite graph theory of ultra-scalable spectral clustering and ensemble clustering [70], making MCHC more feasible for larger-scale datasets.

References

[1] J. Yang, Y.-K. Wang, X. Yao, and C.-T. Lin, "Adaptive initialization method for K-means algorithm," *Front. Artif. Intell.*, vol. 4, 2021, Art. no. 740817.

[2] Y. Yang and H. Wang, "Multi-view clustering: A survey," *Big Data Mining Analytics*, vol. 1, no. 2, pp. 83–107, Jun. 2018, doi: 10.26599/BDMA.2018.9020003.

[3] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, 1999, vol. 2, pp. 1150–1157, doi: 10.1109/ICCV.1999.790410.

[4] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, May 2001, doi: 10.1023/A:1011139631724.

[5] L. Wang and D.-C. He, "Texture classification using texture spectrum," *Pattern Recognit.*, vol. 23, no. 8, pp. 905–910, Jan. 1990, doi: 10.1016/0031-3203(90)90135-8.

[6] M. Brbić and I. Kopriva, "Multi-view low-rank sparse subspace clustering," *Pattern Recognit.*, vol. 73, pp. 247–258, Jan. 2018, doi: 10.1016/j.patcog.2017.08.024.

[7] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Affinity aggregation for spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 773–780, doi: 10.1109/CVPR.2012.6247748.

[8] S. Huang, Z. Kang, and Z. Xu, "Auto-weighted multi-view clustering via deep matrix decomposition," *Pattern Recognit.*, vol. 97, Jan. 2020, Art. no. 107015, doi: 10.1016/j.patcog.2019.107015.

[9] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 23–32.

[10] Y. Jia, H. Liu, J. Hou, S. Kwong, and Q. Zhang, "Multi-view spectral clustering tailored tensor low-rank representation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4784–4797, Dec. 2021, doi: 10.1109/TCSVT.2021.3055039.

[11] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, vol. 24, pp. 1413–1421.

[12] Y. Liang, D. Huang, and C. D. Wang, "Consistency meets inconsistency: A unified graph learning framework for multi-view clustering," in *Proc. IEEE Int. Conf. Data Mining*, 2019, pp. 1204–1209, doi: 10.1109/ICDM.2019.00148.

[13] J. Lv, Z. Kang, B. Wang, L. Ji, and Z. Xu, "Multi-view subspace clustering via partition fusion," *Inf. Sci.*, vol. 560, pp. 410–423, Jun. 2021, doi: 10.1016/j.ins.2021.01.033.

[14] F. Nie, L. Tian, and X. Li, "Multiview clustering via adaptively weighted procrustes," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2022–2030, doi: 10.1145/3219819.3220049.

[15] 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, doi: 10.1109/TKDE.2019.2903810.

[16] 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, doi: 10.1109/TIP.2019.2916740.

[17] W. Xia, Q. Wang, Q. Gao, X. Zhang, and X. Gao, "Self-supervised graph convolutional network for multi-view clustering," *IEEE Trans. Multimedia*, vol. 24, pp. 3182–3192, 2022, doi: 10.1109/TMM.2021.3094296.

[18] 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, Sep. 2021, doi: 10.1016/j.ins.2020.12.073.

[19] X. Yu, H. Liu, Y. Wu, and C. Zhang, "Fine-grained similarity fusion for multi-view spectral clustering," *Inf. Sci.*, vol. 568, pp. 350–368, Aug. 2021, doi: 10.1016/j.ins.2021.03.059.

[20] G.-Y. Zhang, Y.-R. Zhou, C.-D. Wang, D. Huang, and X.-Y. He, "Joint representation learning for multi-view subspace clustering," *Expert Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 113913, doi: 10.1016/j.eswa.2020.113913.

[21] L. Zong, X. Zhang, X. Liu, and H. Yu, "Weighted multi-view spectral clustering based on spectral perturbation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 4621–4628.

[22] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang, and E. Zhu, "Unified one-step multi-view spectral clustering," *IEEE Trans. Knowl. Data Eng.*, p. 1, 2022, doi: 10.1109/TKDE.2022.3172687.

[23] Q. Zheng, J. Zhu, and S. Ma, "Multi-view hierarchical clustering," Oct. 2020, doi: 10.48550/arXiv.2010.07573.

[24] F. Ye, Z. Chen, H. Qian, R. Li, C. Chen, and Z. Zheng, *New Approaches in Multi-View Clustering*. London, U.K.: IntechOpen, 2018, doi: 10.5772/intechopen.75598.

[25] B. Jiang et al., "Semi-supervised multiview feature selection with adaptive graph learning," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2022, doi: 10.1109/TNNLS.2022.3194957.

[26] B. Jiang et al., "Robust multi-view learning via adaptive regression," *Inf. Sci.*, vol. 610, pp. 916–937, Sep. 2022, doi: 10.1016/j.ins.2022.08.017.

[27] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013, doi: 10.1109/TPAMI.2012.88.

[28] Q. Zheng, J. Zhu, Z. Tian, Z. Li, S. Pang, and X. Jia, "Constrained bilinear factorization multi-view subspace clustering," *Knowl.-Based Syst.*, vol. 194, Apr. 2020, Art. no. 105514, doi: 10.1016/j.knosys.2020.105514.

[29] Q. Zheng, J. Zhu, Z. Li, S. Pang, J. Wang, and Y. Li, "Feature concatenation multi-view subspace clustering," *Neurocomputing*, vol. 379, pp. 89–102, Feb. 2020, doi: 10.1016/j.neucom.2019.10.074.

[30] 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, vol. 34, pp. 4412–4419.

[31] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2921–2927.

[32] M. S. Chen, C. D. Wang, and J. H. Lai, "Low-rank tensor based proximity learning for multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, p. 1, 2022, doi: 10.1109/TKDE.2022.3151861.

[33] C. Tang et al., "Learning a joint affinity graph for multiview subspace clustering," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1724–1736, Jul. 2019, doi: 10.1109/TMM.2018.2889560.

[34] 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, doi: 10.1109/34.868688.

[35] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., Cambridge, MA, USA: MIT Press, 2002, pp. 849–856.

[36] V. Menon, G. Muthukrishnan, and S. Kalyani, "Subspace clustering without knowing the number of clusters: A parameter free approach," *IEEE Trans. Signal Process.*, vol. 68, pp. 5047–5062, 2020, doi: 10.1109/TSP.2020.3018665.

[37] S. Sarfraz, V. Sharma, and R. Stiefelhagen, "Efficient parameter-free clustering using first neighbor relations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8926–8935, doi: 10.1109/CVPR.2019.00914.

[38] F. Lin, B. Bai, K. Bai, Y. Ren, P. Zhao, and Z. Xu, "Contrastive multi-view hyperbolic hierarchical clustering," May 5, 2022, doi: 10.48550/arXiv.2205.02618.

[39] T. Kameda et al., "Rawlsian maximin rule operates as a common cognitive anchor in distributive justice and risky decisions," *Proc. Nat. Acad. Sci.*, vol. 113, no. 42, pp. 11817–11822, Oct. 2016, doi: 10.1073/pnas.1602641113.

[40] H. Tao, C. Hou, X. Liu, T. Liu, D. Yi, and J. Zhu, "Reliable multi-view clustering," in *Proc. 32nd AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, doi: 10.1609/aaai.v32i1.11621.

[41] P. Zhao, Y. Jiang, and Z.-H. Zhou, "Multi-view matrix completion for clustering with side information," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2017, pp. 403–415, doi: 10.1007/978-3-319-57529-2_32.

[42] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2564–2570.

[43] X. He, L. Li, D. Roqueiro, and K. Borgwardt, "Multi-view spectral clustering on conflicting views," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2017, pp. 826–842, doi: 10.1007/978-3-319-71246-8_50.

[44] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Proc. Asian Conf. Comput. Vis.*, 2011, pp. 709–720, doi: 10.1007/978-3-642-19309-5_55.

[45] L. Ertoz, M. Steinbach, and V. Kumar, "A new shared nearest neighbor clustering algorithm and its applications," in *Proc. 2nd SIAM Int. Conf. Data Mining*, vol. 8, 2002.

[46] W.-B. Xie, Y.-L. Lee, C. Wang, D.-B. Chen, and T. Zhou, "Hierarchical clustering supported by reciprocal nearest neighbors," *Inf. Sci.*, vol. 527, pp. 279–292, Jul. 2020, doi: 10.1016/j.ins.2020.04.016.

[47] J. Yang, Y. Ma, X. Zhang, S. Li, and Y. Zhang, "An initialization method based on hybrid distance for k -means algorithm," *Neural Comput.*, vol. 29, no. 11, pp. 3094–3117, Nov. 2017, doi: 10.1162/neco_a_01014.

[48] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 911–916, doi: 10.1109/ICDM.2010.35.

[49] A. Bakshi and D. P. Woodruff, "Sublinear time low-rank approximation of distance matrices," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 3786–3796.

[50] S. El Hajjar, F. Dornaika, and F. Abdallah, "One-step multi-view spectral clustering with cluster label correlation graph," *Inf. Sci.*, vol. 592, pp. 97–111, May 2022, doi: 10.1016/j.ins.2022.01.017.

[51] S. Bahrami, A. Bosaghzadeh, and F. Dornaika, "Multi similarity metric fusion in graph-based semi-supervised learning," *Computation*, vol. 7, no. 1, Mar. 2019, Art. no. 15, doi: 10.3390/computation7010015.

[52] C. Lu, S. Yan, and Z. Lin, "Convex sparse spectral clustering: Single-view to multi-view," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2833–2843, Jun. 2016, doi: 10.1109/TIP.2016.2553459.

[53] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-20)," Dept. Comput. Sci., Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-005-96, 1996.

[54] X. Cai, F. Nie, and H. Huang, "Multi-view K-means clustering on big data," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2598–2604.

[55] D. B. Graham and N. M. Allinson, "Characterising virtual eigensignatures for general purpose face recognition," in *Face recognition: From theory to applications*, H. Wechsler, P. J. Phillips, V. Bruce, F. F. Soulié, and T. S. Huang, Eds., Berlin, Germany: Springer-Verlag, 1998, pp. 446–456, doi: 10.1007/978-3-642-72201-1_25.

[56] J. Han, J. Xu, F. Nie, and X. Li, "Multi-view K-means clustering with adaptive sparse memberships and weight allocation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 816–827, Feb. 2022, doi: 10.1109/TKDE.2020.2986201.

[57] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003, doi: 10.1109/TPAMI.2003.1251154.

[58] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-100)," Dept. Comput. Sci., Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-006-96, 1996.

[59] X. Fang, Y. Hu, P. Zhou, and D. O. Wu, "V^3H: View variation and view heredity for incomplete multiview clustering," *IEEE Trans. Artif. Intell.*, vol. 1, no. 03, pp. 233–247, Jul. 2020, doi: 10.1109/TAI.2021.3052425.

[60] 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, doi: 10.1109/TIP.2018.2877335.

[61] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, vol. 28, no. 1, pp. 2149–2155.

[62] A. Strehl and J. Ghosh, "Cluster ensembles — a knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, 2002.

[63] D. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[64] W. Zhang, X. Wang, D. Zhao, and X. Tang, "Graph degree linkage: Agglomerative clustering on a directed graph," in *Proc. Comput. Vis. – ECCV*, 2012, pp. 428–441, doi: 10.1007/978-3-642-33718-5_31.

[65] R. Liu, H. Wang, and X. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Inf. Sci.*, vol. 450, pp. 200–226, Jun. 2018, doi: 10.1016/j.ins.2018.03.031.

[66] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, Jan. 1973, doi: 10.1080/01969727308546046.

[67] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987, doi: 10.1016/0377-0427(87)90125-7.

[68] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, and S. Wu, "Understanding and enhancement of internal clustering validation measures," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 982–994, Jun. 2013, doi: 10.1109/TSMCB.2012.2220543.

[69] L. Hu and C. Zhong, "An internal validity index based on density-involved distance," *IEEE Access*, vol. 7, pp. 40038–40051, 2019, doi: 10.1109/ACCESS.2019.2906949.

[70] D. Huang, C.-D. Wang, J.-S. Wu, J.-H. Lai, and C.-K. Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1212–1226, Jun. 2020, doi: 10.1109/TKDE.2019.2903410.

Jie Yang is currently working toward the Ph.D. degree with the Computational Intelligence and Brain-Computer Interface Lab, Australian AI Institute, FEIT, University of Technology Sydney, Sydney, NSW, Australia. He is also a Research Assistant with Computational Intelligence and Brain-Computer Interface Lab, Australian AI Institute, FEIT, University of Technology Sydney. His current research interests include unsupervised learning, clustering, feature selection, and EEG data processing. He is a Reviewer for many international journals, such as the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and *Information Sciences*.

Chin-Teng Lin (Fellow, IEEE) received the Bachelor of Science degree from National Chiao-Tung University (NCTU), Taiwan, in 1986, and the master's and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1989 and 1992, respectively. He is currently a Distinguished Professor and the Co-Director of the Australian Artificial Intelligence Institute with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia. He is also the Honorary Chair Professor of electrical and computer engineering with NCTU.

He is the co-author of *Neural Fuzzy Systems* (Prentice-Hall) and the author of *Neural Fuzzy Control Systems with Structure and Parameter Learning* (World Scientific). He has authored or coauthored more than 425 journal papers, including more than 200 IEEE journal papers in the areas of neural networks, fuzzy systems, brain-computer interface, multimedia information processing, cognitive neuro-engineering, and human-machine teaming, that have been cited more than 30000 times. His h-index is 85, and i10-index is 372. For his contributions to biologically inspired information systems, he was awarded Fellowship with the IEEE in 2005, and with the International Fuzzy Systems Association in 2012. He was the recipient of the IEEE Fuzzy Systems Pioneer Award in 2017. He has held notable positions as the Editor-in-Chief of the IEEE TRANSACTIONS ON FUZZY SYSTEMS from 2011 to 2016, seats on Board of Governors for the IEEE Circuits and Systems (CAS) Society during 2005–2008, IEEE Systems, Man, Cybernetics (SMC) Society during 2003–2005, IEEE Computational Intelligence Society during 2008–2010, the Chair of the IEEE Taipei Section during 2009–2010, a Distinguished Lecturer with the IEEE CAS Society during 2003–2005 and the CIS Society during 2015–2017, Chair of the IEEE CIS Distinguished Lecturer Program Committee during 2018–2019, IEEE CIS Awards Chair in 2022, Deputy Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-II during 2006–2008, Program Chair of the IEEE International Conference on Systems, Man, and Cybernetics in 2005, and General Chair of the 2011 IEEE International Conference on Fuzzy Systems.