Fair Clustering Ensemble With Equal Cluster Capacity

Peng Zhou, Rongwen Li, Zhaolong Ling, Liang Du, and Xinwang Liu

***Abstract*—Clustering ensemble has been widely studied in data mining and machine learning. However, the existing clustering ensemble methods do not pay attention to fairness, which is important in real-world applications, especially in applications involving humans. To address this issue, this paper proposes a novel fair clustering ensemble method, which takes multiple base clustering results as inputs and learns a fair consensus clustering result. When designing the algorithm, we observe that one of the widely used definitions of fairness may cause a cluster imbalance problem. To tackle this problem, we give a new definition of fairness that can simultaneously characterize fairness and cluster capacity equality. Based on this new definition, we design an extremely simple yet effective regularized term to achieve fairness and cluster capacity equality. We plug this regularized term into our clustering ensemble framework, finally leading to our new fair clustering ensemble method. The extensive experiments show that, compared with the state-of-the-art clustering ensemble methods, our method can not only achieve a comparable or even better clustering performance, but also obtain a much fairer and better capacity equality result, which well demonstrates the effectiveness and superiority of our method.**

***Index Terms*—Clustering ensemble, clustering with equal capacity, fairness.**

I. Introduction

CLUSTERING is a fundamental unsupervised machine learning task and has been widely used in real-world applications such as social networks [1] and crime analysis [2]. Since clustering is an unsupervised task, most clustering methods may suffer from stableness and robustness problems [3]. To address these issues, clustering ensemble is proposed [4]. Clustering ensemble aims to integrate multiple weak base clustering results into a consensus one to achieve a more robust or stable clustering result. In recent years, due to its robustness and stableness, clustering ensemble methods have been widely studied [5], [6], [7], [8], [9], [10].

As introduced above, since clustering is often used in real-world applications involving humans such as social networks and crime analysis, we should guarantee that the clustering result is fair enough to help humans to make decisions. Mainstream fairness has two forms: group-level fairness which focuses on the fairness of some specific groups, and individual-level fairness which focuses on the fairness of every individual [11]. In this paper, we focus on group-level fairness. In some real-world applications, some specific groups should be protected, such as females, which are called protected groups. The clustering with group-level fairness wishes that there are no clusters that have a disproportionately small number of instances in some specific protected groups [12], [13]. Although the conventional clustering ensemble methods can improve the clustering performance to some extent, none of them consider the fairness of the consensus result, and thus may obtain unfair consensus results if the base clustering results are unfair. Notice that there exist some fair clustering methods, such as fair k-means [14], fair spectral clustering [15], and fair deep clustering [16]. Fair clustering ensemble has some essential differences from fair clustering. Firstly, the conventional fair clustering methods are designed for some particular clustering methods such as k-means and spectral clustering, but fair clustering ensemble does not care how to generate the base results, and only takes the base results (which are often unfair) as inputs and obtain a fair consensus result. Therefore, the fair clustering ensemble is a more general post-processing framework that can follow any fair or unfair clustering methods. Secondly, the fair clustering ensemble does not need to access the original features or attributes of data, which can protect the privacy of the data [6].

To address the fairness problem in clustering ensemble, in this paper, we propose a novel group-level fair clustering ensemble method. Our method is based on a widely-used definition of group-level fairness [17], which makes the partition not biased towards or against some specific groups in the population. However, we observe that this definition of fairness ignores the capacity of each cluster, and may cause some extremely large or small clusters. For example, the results will be very fair if we put most or all instances into one cluster, according to their definition. Fig. 1 shows a simple example of clustering of humans. In this example, we have 20 people (10 males and 10 females) who are denoted as triangles and squares and we wish to partition them into two clusters. We have two protected

Received 1 March 2024; revised 8 November 2024; accepted 24 November 2024. Date of publication 28 November 2024; date of current version 5 February 2025. This work was supported by the National Natural Science Foundation of China under Grant 62176001 and Grant 62306002. This work is supported by the National Science Fund for Distinguished Young Scholars of China under Grant 62325604. This work is also supported by the Natural Science Project of Anhui Provincial Education Department under Grant 2023AH030004. Recommended for acceptance by M. Salzmann. *(Corresponding author: Xinwang Liu.)*

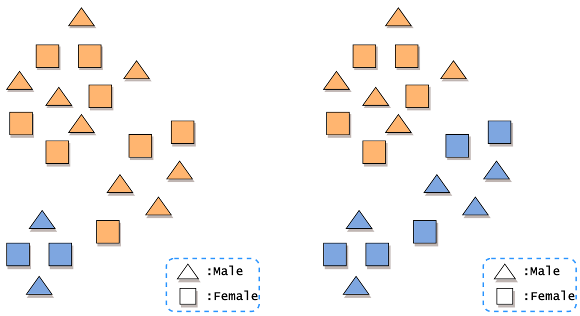
Peng Zhou, Rongwen Li, and Zhaolong Ling are with the Anhui Provincial International Joint Research Center for Advanced Technology in Medical Imaging, School of Computer Science and Technology, Anhui University, Hefei 230601, China (e-mail: zhoupeng@ahu.edu.cn; E22301284@stu.ahu.edu.cn; zlling@ahu.edu.cn).

Liang Du is with the School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China (e-mail: duliang@sxu.edu.cn).

Xinwang Liu is with the College of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: xinwangliu@nudt.edu.cn).

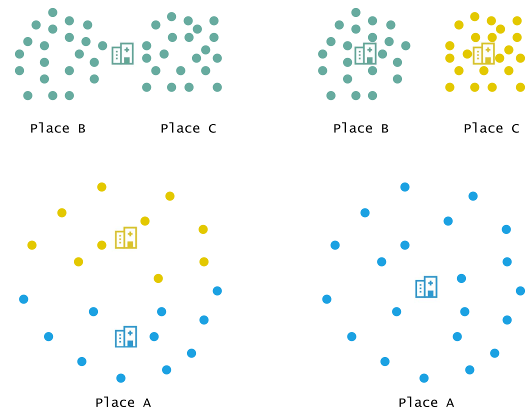
This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3507857, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3507857



(a) Fair result with inequal cluster capacity. (b) Fair result with equal cluster capacity.

Fig. 1. A simple example to show the fairness and cluster capacity equality. There are two protected groups, i.e., males and females. The triangles denote the males and the squares denote the females. (a) and (b) show two clustering results. Each result divides the individuals into two clusters, where one is denoted by the yellow color and the other is denoted by the blue color. The result in (a) is fair but imbalanced and the result in (b) is fair and has an equal cluster capacity.



(a) The partition result of k-means, which is imbalanced and unfair. (b) A balanced and fair partition result.

Fig. 2. An example of hospital location for three places A, B, and C. Places B and C are two dense urban centers close to each other, whose radius is both small. Place A is a big suburb far away from B and C, whose radius is large. We want to construct three hospitals for them. (a) shows the result of traditional k-means, which is imbalanced and unfair. (b) shows the balanced and fair result. In each result, the points with the same color form a cluster and own a hospital.

groups, i.e., male and female, where the triangles denote the males and the squares denote the females. Fig. 1(a) shows a clustering result, where the yellow color denotes one cluster and the blue color denotes the other cluster. According to the fairness definition in [17], this result is perfectly fair. However, the result is imbalanced because most people are divided into the yellow cluster. Fig. 1(b) shows another clustering result, which is also fair but has an equal cluster capacity. If we do not consider the cluster capacity explicitly, although conventional fair clustering methods have some objectives w.r.t. fairness and accuracy, they may still put most or all instances into one cluster. We take a Reverse MNIST [18] data set in our experiments as an example. Considering a state-of-the-art fair clustering method SFD [19], it achieves good fairness, i.e., 0.709 on Bal and 0.979 on MNCE (Bal and MNCE are metrics for fairness, and the larger the fairer). However, we observe that SFD puts most data into one cluster, which obtains low values on the cluster capacity equality metrics, i.e., 0.029 on CCE and 0.558 on NE (CCE and NE are metrics for cluster capacity equality, and the larger the better). More experimental details are shown in Section IV-D. It shows that although SFD can achieve good fairness since it does not consider the cluster capacity explicitly, it may cause results with inequality cluster capacity.

Besides, in many real-world applications, clusters with equal capacity are often required. For example, when a school divides the students into several classes, there should be nearly the same number of students in each class. Another example is the energy load balance of wireless sensor networks. The inequal capacity clusters may cause energy consumption and shorten the network lifetime [20]. Clustering with equal capacity can also avoid extremely large or small clusters that are often undesirable in clustering tasks. Moreover, for the consideration of fairness, equal capacity is often still helpful. Chen et al. provided an example of resource allocation in [21] to show the effects of equal capacity on fairness. In this example, there are three places A, B, and C, where each place has almost the same population. Places B and C are two dense urban centers close to each other, whose radius is both small. Place A is a big suburb far away from B and C, whose radius is large. Now we hope to build three hospitals in these places, and thus we need to partition the people in these places into three clusters. If we use the k-means clustering, since the radius of A is much larger than B and C and k-means tries to minimize the overall distances between each person and his/her cluster center, k-means is prone to divide A into two clusters and merge B and C to the same cluster, which is shown as Fig. 2(a). Then, we will build two hospitals in place A and let the people in B and C share one hospital, which seems unfair because the hospital in places B and C must serve four times as many people as the hospitals in A (notice that the population in A, B and C is almost the same). If we partition them into equal capacity clusters, which means we build one hospital in each of places A, B, and C, respectively, we can obtain a fairer result. The result is shown in Fig. 2(b).

To address this equal capacity issue, we propose a new definition of fairness considering the capacity of clusters. Based on this definition, we propose an extremely simple yet effective regularized term to simultaneously achieve fairness and equal cluster capacity. Then, we plug this fairness and equal cluster capacity regularized term into a clustering ensemble framework, leading to our Fair Clustering Ensemble (FCE) method. At last, we provide an effective iterative algorithm to optimize the introduced objective function to obtain the final clustering result. The experimental results on some benchmark data sets show that our method can achieve a fairer and more capacity equal clustering result than the state-of-the-art clustering ensemble methods. It well demonstrates the effectiveness of our fair clustering ensemble method.

It is worth clarifying that clustering may have different goals, such as fairness, accuracy, and cluster capacity equality. These goals may be consistent or at odds with each other. In this paper, we observe that fairness and cluster capacity equality can be consistent with each other and we design a regularized term, which can simultaneously improve fairness and cluster capacity equality. However, the accuracy may be at odds with others. Notice that, when measuring the accuracy, we often use external indicators, which need ground truth labels. In practice, the ground truth itself may be unfair or imbalanced, and thus the accuracy may be at odds with the fairness or the cluster capacity equality. If the applications involve humans which needs fairness, and we also want to partition the data into some groups with similar sizes, we can use the proposed method. If accuracy is at odds with fairness, we should pay more attention to fairness, or we may cause some bad social effects, such as sexism or other discrimination. If the application does not need fairness, and we just want to partition the data into some groups with similar sizes, we can also use the proposed method, which will be discussed in Section III-D. If we need neither fairness nor cluster capacity equality, we can use the traditional methods instead of the proposed one.

The main contributions of this paper are summarized as:

- To the best of our knowledge, this is the first work of clustering ensemble considering both the fairness and equal cluster capacity.
- We provide a new fairness definition by fully considering the fairness and equal cluster capacity.
- We design a novel, simple, and effective regularized term that can achieve fairness and equal cluster capacity simultaneously, and plug it into our clustering ensemble framework to form a new fair clustering ensemble method.
- The experimental results demonstrate the effectiveness of our method on fairness, equal cluster capacity, and accuracy.

II. Related Work and Preliminaries

In this section, we briefly review some related works and preliminaries about clustering ensemble, fair clustering, and clustering with equal capacity.

A. Clustering Ensemble

Clustering ensemble was first proposed in [4], aiming to integrate multiple weak base clustering results to obtain a more robust consensus clustering result. Given a data set with n instances $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, we first generate m base clustering results $\mathcal{C}^1, \ldots, \mathcal{C}^m$, where the j-th base result $\mathcal{C}^j$ contains c clusters $\pi_1^j, \ldots, \pi_c^j$ and $\mathcal{X} = \bigcup_{i=1}^{c} \pi_i^j$. Clustering ensemble aims to learn a consensus partition $\mathcal{C}^*$ by ensembling the m base clusterings $\mathcal{C}^1, \ldots, \mathcal{C}^m$.

One kind of mainstream clustering ensemble method is based on the co-association (CA) matrix, which is the matrix containing the number of times two data appear in the same cluster in multiple basic clusterings [22], [23], [24], [25], [26], [27]. For example, Tao et al. learned the robust representation from the CA matrix through the low-rank constraint to remove noises [26]; Jia et al. generated an enhanced CA matrix by propagating the high-reliability information in the CA matrix to achieve better clustering performance [27]. Because the CA matrix can also be viewed as an adjacency matrix or a similarity matrix, there are also many graph-based methods [28], [29], [30], [31], [32], [33]. For example, Liu et al. applied spectral clustering on the CA matrix and proved its theoretical equivalence with weighted k-means clustering [31]; Zhou et al. proposed a graph-based tri-level robust clustering ensemble method [32]; Zhou et al. applied the self-paced learning to learn a more robust CA matrix for ensemble [34]; Chen et al. refined multiple connection matrices through substantial rank recovery and graph tensor learning [33]; Zhou et al. proposed a clustering ensemble method on a multiplex graph [35].

Although clustering ensemble methods based on the CA matrix or graph have shown good performance, the high space and time complexity hinders their applications on large-scale data sets. Therefore, many methods attempt to ensemble base clustering using other data structures [6], [36], [37], [38], [39], [40], [41]. For example, Bai et al. developed an information theory framework to maintain the consistency of basic clustering results [39]; Zhou et al. proposed an alignment method to ensemble multiple k-means [42]; Huang et al. proposed a clustering ensemble method based on sparse graph representation and probabilistic trajectory analysis [40]; Li et al. developed a clustering ensemble method based on sample stability [6]; Zhou et al. proposed a partial clustering ensemble method that simultaneously filled in missing values and ensembled them [41]; Zheng et al. obtained a more reliable clustering indicator matrix by weighting on clusters and performed non-negative matrix factorization [43].

Previous clustering ensemble methods focus on improving clustering performance, whereas ignoring the fairness of the result. In this paper, we develop a new fair clustering ensemble method that can improve not only the clustering performance but also the fairness.

B. Fair Clustering

In recent years, the fairness of clustering has attracted increasingly more attention [44]. Chierichetti et al. provided the first definition of cluster fairness and proposed a fair decomposition method by first decomposing data into small subsets with fair properties, and then running off-the-shelf clustering on these subsets [12]. Then, Rösner et al. proposed a fair clustering method that can handle more than two protected groups [45]. Backurs et al. proposed a linear-time fair decomposition algorithm for clustering [19]. Kleindessner et al. extended spectral clustering by recasting the fairness as a linear constraint [15].

There are several definitions for fairness in existing works. One of the most widely used is proposed in [17], which is shown as follows:

Definition 1. (Fairness) [17]: Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ denote n instances with d attributes, which are partitioned into c disjoint clusters $\mathcal{C} = \{\pi_1, \ldots, \pi_c\}$. Given T disjoint protected groups $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_T$, let $\eta_i = \frac{|\mathcal{G}_i|}{n}$ and $\eta_i(k) = \frac{|\pi_k \cap \mathcal{G}_i|}{|\pi_k|}$ denote the proportion of group $\mathcal{G}_i$ in the whole data and cluster π_k, respectively. The fairness of π_k can be defined as:

$$fairness\,(\pi_k) = \min\left(\frac{\eta_i}{\eta_i(k)}, \frac{\eta_i(k)}{\eta_i}\right),\ \forall i \in \{1, \ldots T\}$$

The fairness of the whole clustering result $\mathcal{C}$ is defined as:

$$fairness(\mathcal{C}) = \min_{k \in \{1,\ldots c\}} fairness(\pi_k) \quad (1)$$

Remark 1: $fairness(\mathcal{C}) \in [0,1]$. The larger $fairness(\mathcal{C})$ is, the fairer the clustering result is. Therefore, $fairness(\mathcal{C}) = 1$ means the results are perfectly fair. Definition 1 shows that a fair clustering result requires that the proportion of $\mathcal{G}_i$ in each cluster (i.e., $\eta_i(k)$) should be close to the proportion of $\mathcal{G}_i$ in the whole data (i.e., η_i).

Based on this definition, several non-deep fair clustering methods and deep fair clustering methods have been proposed. For non-deep fair clustering methods, Ziko et al. proposed a variational fair clustering framework by combining fairness terms with three different clustering objectives [46]; Chen et al. defined a notion of proportionally fair clustering where all possible groups of reasonably large size are entitled to choose a center for themselves [21]; Mohsen et al. proposed adding weights to all sample points during the clustering process to achieve fairness in clustering [47]; Li et al. proposed a reweighting method to achieve the group-level fairness [48]; Ghadiri et al. presented a fair k-means method to choose cluster centers providing similar costs for different groups [49]; Jessica et al. used antidote data in clustering to improve group level fairness [50].

For deep fair clustering methods, Wang et al. embedded this fairness into deep clustering by learning a differentiated and fair clustering allocation function [51]; Chhabra et al. provided a robust deep fair clustering method by considering the fairness attack [52]; Zeng et al. embedded fairness constraints into deep clustering by maximizing and minimizing mutual information [16].

Although these works have demonstrated promising performance on fairness, they are designed for some specific clustering methods, which are not general enough. Moreover, these methods need to access all the original features of data, which may cause privacy leakage. To address these problems, this paper focuses on the fair clustering ensemble, which is one of the post-processings for fair clustering.

C. Clustering With Equal Capacity

To avoid extremely large or small clusters, or to handle the data whose original distribution is balanced, sometimes we hope that clusters should contain similar numbers of instances. Therefore, some clustering algorithms with equal capacity have been proposed, which can be roughly divided into two types: hard-equality and soft-equality methods. Notice that, in previous literature, clustering with equal capacity is often called "balance clustering". However, the term "balance" is also often used in fairness clustering to refer the fairness. To avoid confusion, in this paper, we use the term "clustering with equal capacity" to refer to the "balance clustering" used in the previous literature.

Hard-equality methods hope that the number of instances in each cluster is strictly the same. To achieve this, Bradley et al. and Malinen et al. considered imposing some constraints on the k-means clustering so that they can strictly control the size of the clusters in [53] and [54], respectively. Then, Costa et al. designed the equal capacity constraint for the minimum sum-of-squares clustering [55].

The soft-equality methods only apply the equal capacity constraint as a penalty to the clustering objective function to obtain less strict results compared with the hard-equality methods. For example, Banerjee et al. designed effective equal cluster capacity regularization terms and plugged them into a clustering method [56], [57]. Liu et al. designed a lasso-liked term to make the clustering results of least square regression achieve equal cluster capacity [20]. Liu et al. used the variance of cluster size as a penalty for clustering and adopted a fast optimization process to handle large-scale data sets [58]. Zhou et al. proposed new k-means and spectral clustering methods with an equal capacity regularized term and applied them to the feature selection task [59], [60].

Clustering with equal cluster capacity can prevent discrimination against minority groups and thus is helpful to fair clustering [21]. Therefore, we also plug this property into our fair clustering ensemble framework.

III. Clustering Ensemble With Fairness

Most clustering methods cannot guarantee the fairness of the clustering result. To address this issue, clustering ensemble can be used as a post-processing to obtain a fair consensus clustering result from multiple unfair base clustering results, leading to a fair clustering ensemble. The problem setting of the fair clustering ensemble is as follows:

Problem Setting (Fair Clustering Ensemble): Given m base clustering results $\mathcal{C}^1,\ldots,\mathcal{C}^m$ of n instances, and T protected groups $\mathcal{G}_1,\ldots,\mathcal{G}_T$, fair clustering ensemble aims to obtain a consensus clustering result $\mathcal{C}^*$ which is fair w.r.t. the protected groups in $\mathcal{G}_1,\ldots,\mathcal{G}_T$.

Remark 2: The base clustering results $\mathcal{C}^1,\ldots,\mathcal{C}^m$ can be generated by any clustering methods which are no matter fair or unfair methods. Since the fair clustering ensemble does not require the fairness of the base clustering methods and base results, it is a more general framework to achieve fairness compared to the fair clustering methods.

Remark 3: The inputs of the fair clustering ensemble are the multiple base results $\mathcal{C}^1,\ldots,\mathcal{C}^m$ together with several protected groups $\mathcal{G}_1,\ldots,\mathcal{G}_T$. It does not need to access any original features of data, and thus it can protect the privacy of data compared to fair clustering methods. Notice that to obtain the base results $\mathcal{C}^1,\ldots,\mathcal{C}^m$ need the original features, for example, we run k-means on the original features to obtain the base results. However, the clustering ensemble does not care about this process and does not care about how to obtain the base results, either. Considering some popular scenarios of clustering ensemble, which are the distributed computing scenario or the federated learning scenario, the local clients can run off-the-shelf clustering methods on their private data locally to generate the base results. Then, the local clients upload their base results to the cloud server without their private data, and the cloud server runs the clustering ensemble method on these base results without accessing the private data. In these scenarios, clustering ensemble can indeed generate a consensus result which protects the privacy of data.

A. Fairness Regularize

In this paper, we use Definition 1 to measure the fairness of the clustering result. However, we observe that although Definition 1 can well characterize the fairness, it may cause another problem of *equal cluster capacity*.

Considering an extreme case if we put all instances into one cluster in $\mathcal{C}$, according to Definition 1, we have $fairness(\mathcal{C}) = 1$, which means it easily achieves the perfect fairness. Therefore, if we directly use Definition 1 as the objective, we may obtain an inequal capacity clustering result. However, according to many previous works [54], [55], [60], [61], [62], in many real-world applications, we wish that the numbers of data in each cluster are about the same, and should not have the extremely large or small clusters. Hence, besides fairness, the equal cluster capacity also needs to be considered sometimes in practice.

To measure the equality of cluster capacity, we also need a metric of it just like fairness. Similar to Definition 1, we provide the following definition of cluster capacity equality of a clustering result $\mathcal{C} = \{\pi_1, \ldots, \pi_c\}$:

Definition 2. (Cluster Capacity Equality): Let $|\pi_k|$ indicates the number of samples in cluster π_k. The Cluster Capacity Equality (CCE) of $\mathcal{C}$ can be defined as:

$$CCE\left(\mathcal{C}\right) = \min\left(\frac{|\pi_i|}{|\pi_j|}\right),\ \forall i, j \in \{1, \ldots, c\}. \tag{2}$$

Remark 4: $CCE(\mathcal{C}) \in [0, 1]$. The larger $CCE(\mathcal{C})$ is, the more equal the cluster capacity is. It is easy to verify that when $|\pi_i| = |\pi_j| = \frac{n}{c}$, it is the most equal result.

With this definition of cluster capacity equality, we can provide a new metric to simultaneously measure the fairness and cluster capacity equality of a clustering result $\mathcal{C} = \{\pi_1, \ldots, \pi_c\}$. Suppose we have T protected groups $\mathcal{G}_1, \ldots, \mathcal{G}_T$. Then we give the following definition of $fairness\_CCE$:

Definition 3: (fairness_CCE) Let $\gamma_i(k) = \frac{|\pi_k \cap \mathcal{G}_i|}{|\mathcal{G}_i|}$ be the proportion of cluster π_k in the group $\mathcal{G}_i$. The fairnesss_CCE of π_k can be defined as:

$$fairnesss\_CCE\left(\pi_k\right) = \min_{i \in \{1, \ldots T\}} \left(c\gamma_i(k), \frac{1}{c\gamma_i(k)}\right).$$

The fairness_CCE of a clustering result $\mathcal{C}$ is defined as:

$$fairness\_CCE(\mathcal{C}) = \min_{k \in \{1, \ldots c\}} fairness\_CCE(\pi_k)$$

Remark 5: $fairness\_balance(\mathcal{C}) \in (0, 1]$. The larger $fairness\_balance(\mathcal{C})$ is, the fairer and more cluster capacity equal the clustering result $\mathcal{C}$ is.

Remark 6: Definition 3 is following the classical definition of fairness, i.e., Definition 1. Strictly speaking, they are more like fairness metrics. If we want to define what is "fair" and what is "unfair" explicitly, we should give a threshold δ. If $fairness\_CCE \geq \delta$, we can tell that the result is fair.

Remark 7: Here we show why $fairness\_CCE(\mathcal{C})$ can be used to measure the fairness and cluster capacity equality. It is easy to verify that the closer $c\gamma_i(k)$ is to 1, the larger $fairness\_CCE(\pi_k)$ is. Take a closer look at $c\gamma_i(k)$:

$$c\gamma_i(k) = \frac{c|\pi_k \cap \mathcal{G}_i|}{|\mathcal{G}_i|} \tag{3}$$

If $c\gamma_i(k)$ is close to 1, which means $\frac{c|\pi_k \cap \mathcal{G}_i|}{|\mathcal{G}_i|} \approx 1$, and further we obtain

$$|\pi_k \cap \mathcal{G}_i| \approx \frac{|\mathcal{G}_i|}{c} \tag{4}$$

Summing the left-hand side and right-hand side of (4) w.r.t. i, we have

$$\sum_{i=1}^{T} |\pi_k \cap \mathcal{G}_i| \approx \sum_{i=1}^{T} \frac{|\mathcal{G}_i|}{c} \Rightarrow |\pi_k| \approx \frac{n}{c}. \tag{5}$$

Equation (5) shows that the results have more equal cluster capacity according to Definition 2.

Then, dividing (4) by (5), we have

$$\frac{|\pi_k \cap \mathcal{G}_i|}{|\pi_k|} \approx \frac{|\mathcal{G}_i|}{n} \tag{6}$$

Notice that $\frac{|\pi_k \cap \mathcal{G}_i|}{|\pi_k|}$ is exact $\eta_i(k)$ in Definition 1 and $\frac{|\mathcal{G}_i|}{n}$ is η_i in Definition 1. Therefore, according to Definition 1, the results are fair. To sum up, the larger $fairness\_balance$ is, the better the fairness and cluster capacity equal of the clustering result is.

Based on the above discussion, we find that (4) is the key to simultaneously achieving fairness (i.e., (6)) and equal cluster capacity (i.e., (5)). Let us take a closer look at (4). Equation (4) means that we should divide group $\mathcal{G}_i$ averagely to each cluster. Based on this idea, we can design a simple regularized term to simultaneously achieve fairness and cluster capacity equality as follows.

We first construct a one-hot matrix $\mathbf{G} \in \{0, 1\}^{n \times T}$ for all instances from $\mathcal{G}_1, \ldots, \mathcal{G}_T$, where n is the number of instances and T is the number of protected groups. If the i-th instance belongs to the j-th protected group, $G_{ij} = 1$, and $G_{ij} = 0$ otherwise.

Then, given a clustering result with c clusters, we can construct a one-hot result matrix $\mathbf{Y} \in \{0, 1\}^{n \times c}$, where if the i-th instance belongs to the j-th cluster, then $Y_{ij} = 1$ and $Y_{ij} = 0$ otherwise. Given $\mathbf{G}$ and $\mathbf{Y}$, we can construct $\mathbf{A} = \mathbf{G}^T\mathbf{Y}$. Notice that the (i, j)-th element in $\mathbf{A}$, which is denoted as A_{ij}, is $A_{ij} = |\pi_j \cap \mathcal{G}_i|$.

Now, consider the i-th protected group. According to (4), we wish $\mathcal{G}_i$ be divided equally into each cluster, which means $A_{i1}, A_{i2}, \ldots, A_{ic}$ should be close to each other. Notice that, the summation of $A_{i1}, A_{i2}, \ldots, A_{ic}$, which is $\sum_{j=1}^{c} A_{ij} = |\mathcal{G}_i|$, is a constant. Now consider the following optimization problem:

$$\min_{A_{i1}, \ldots, A_{ic}} \quad \sum_{j=1}^{c} A_{ij}^2,$$
$$\text{s.t.} \quad \sum_{j=1}^{c} A_{ij} = |\mathcal{G}_i|. \tag{7}$$

It is easy to verify that the optima of (7) is $A_{i1} = A_{i2} = \cdots = A_{ic} = \frac{|\mathcal{G}_i|}{c}$, which means we divide $\mathcal{G}_i$ equally into each cluster. Similarly, for any other protected groups $\mathcal{G}_k$, we can also minimize $\sum_{j=1}^{c} A_{kj}^2$ to achieve fairness and cluster balance.

To sum up, we obtain the following term to simultaneously achieve fairness and cluster capacity equality:

$$\min_{\mathbf{A}} \sum_{k=1}^{T}\sum_{j=1}^{c} A_{kj}^2 = \min_{\mathbf{A}} \|\mathbf{A}\|_F^2 \Leftrightarrow \min_{\mathbf{Y}} \|\mathbf{G}^T\mathbf{Y}\|_F^2. \tag{8}$$

Given a one-hot protected group matrix $\mathbf{G}$, we wish to learn a cluster partition matrix $\mathbf{Y}$. If $\mathbf{Y}$ satisfies (8), then the clustering result can satisfy the fairness and cluster capacity equality. In the following, we will plug this regularized term into a clustering ensemble framework to obtain our fair clustering ensemble method.

B. Objective Function

In clustering ensemble, we first construct multiple one-hot clustering result matrix $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)} \in \{0,1\}^{n\times c}$ from $\mathcal{C}^1, \ldots, \mathcal{C}^m$ as introduced before. Then we try to ensemble $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)}$ to obtain a consensus result $\mathbf{Y} \in \mathbb{R}^{n\times c}$.

Notice that we cannot directly average $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)}$ to obtain $\mathbf{Y}$ because the clusters in each base result are not aligned. For example, the first cluster in $\mathbf{Y}^{(i)}$ is not necessarily the same as the first cluster in $\mathbf{Y}^{(j)}$. To tackle this problem, we can introduce a learnable rotation matrix $\mathbf{R}^{(i)} \in \mathbb{R}^{c\times c}$ for each base result, where $\mathbf{R}^{(i)T}\mathbf{R}^{(i)} = \mathbf{I}$, to align the clusters in each base result. Then, when we learn an appropriate rotation matrix $\mathbf{R}^{(i)}$, $\mathbf{Y}^{(i)}\mathbf{R}^{(i)}$ can be seen as the aligned base result, which is ready for the ensemble.

We first learn a consensus orthogonal embedding $\mathbf{H} \in \mathbb{R}^{n\times c}$ from $\mathbf{Y}^{(i)}\mathbf{R}^{(i)}$ by minimizing $\sum_{i=1}^{m} \alpha_i^2 \|\mathbf{H} - \mathbf{Y}^{(i)}\mathbf{R}^{(i)}\|_F^2$, where $0 \le \alpha_i \le 1$ is the weight of the i-th base result. Larger α_i represents a more important base result. Notice that each column of $\mathbf{H}$ is a representation of a cluster. In the conventional clustering setting, each instance should belong to only one cluster, and thus clusters should be far away from each other. This is the reason why we wish $\mathbf{H}$ to be orthogonal as spectral clustering did.

Since clustering aims to learn a hard partition of data instead of an embedding, we need to obtain a 0/1 matrix $\mathbf{Y}$ by discretizing $\mathbf{H}$. To avoid using any post-processing methods, we design a one-stage clustering ensemble method, which learns the 0/1 matrix $\mathbf{Y}$ directly without any post-processing. To this end, inspired by the spectral rotation [63], we add a discretizing term $\|\mathbf{Y} - \mathbf{H}\mathbf{R}\|_F^2$ to the objective function, where $\mathbf{R} \in \mathbb{R}^{c\times c}$ is also a rotation matrix. Then, we obtain our clustering ensemble objective function:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{m} \alpha_i^2 \|\mathbf{H} - \mathbf{Y}^{(i)}\mathbf{R}^{(i)}\|_F^2 + \lambda_1 \|\mathbf{Y} - \mathbf{H}\mathbf{R}\|_F^2 \\ \text{s.t.} \quad & \mathbf{H}^T\mathbf{H} = \mathbf{I},\ \mathbf{R}^T\mathbf{R} = \mathbf{I},\ \mathbf{R}^{(i)T}\mathbf{R}^{(i)} = \mathbf{I} \\ & \mathbf{Y} \in \{0,1\}^{n\times c},\ \sum_{j=1}^{c} Y_{ij} = 1,\ 0 \le \alpha_i \le 1,\ \sum_{i=1}^{m} \alpha_i = 1, \end{aligned} \tag{9}$$

where $\boldsymbol{\theta} = \{\alpha_i, \mathbf{H}, \mathbf{R}^{(i)}, \mathbf{R}, \mathbf{Y}\}$ is the set of learnable parameters in the objective function, and λ_1 is a hyper-parameter. Since the second term is only for discretization instead of ensemble, we do not wish it to affect the ensemble too much. To this end, we fix λ_1 as a small constant 0.001.

Then, we plug our fairness regularized term (8) into (9), leading to our final objective function:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \ & \sum_{i=1}^{m} \alpha_i^2 \|\mathbf{H} - \mathbf{Y}^{(i)}\mathbf{R}^{(i)}\|_F^2 + \lambda_1 \left(\|\mathbf{Y} - \mathbf{H}\mathbf{R}\|_F^2 + \lambda_2 \|\mathbf{G}^T\mathbf{Y}\|_F^2\right), \\ \text{s.t.} \ & \mathbf{H}^T\mathbf{H} = \mathbf{I},\ \mathbf{R}^T\mathbf{R} = \mathbf{I},\ \mathbf{R}^{(i)T}\mathbf{R}^{(i)} = \mathbf{I} \\ & \mathbf{Y} \in \{0,1\}^{n\times c},\ \sum_{j=1}^{c} Y_{ij} = 1,\ 0 \le \alpha_i \le 1,\ \sum_{i=1}^{m} \alpha_i = 1, \end{aligned} \tag{10}$$

where λ_2 is a trade-off hyper-parameter to balance the clustering performance and fairness.

C. Optimization

We optimize one variable by fixing other variables.

1) Optimizing $\mathbf{H}$*:* When fixing other variables, we can rewrite the subproblem w.r.t. $\mathbf{H}$ as:

$$\begin{aligned} \min_{\mathbf{H}} \quad & -tr(\mathbf{H}^T\mathbf{B}) \\ \text{s.t.} \quad & \mathbf{H}^T\mathbf{H} = \mathbf{I}, \end{aligned} \tag{11}$$

where $\mathbf{B} = \sum_{i=1}^{m} \alpha_i^2 \mathbf{Y}^{(i)}\mathbf{R}^{(i)} + \lambda_1 \mathbf{Y}\mathbf{R}^T$. Equation (11) can be solved by using the Singular Value Decomposition (SVD) on $\mathbf{B}$. Here the following Theorem gives a closed-form solution for the problem in (11).

Theorem 1: Supposing the SVD of $\mathbf{B}$ is $\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\boldsymbol{\Sigma}$ is a diagonal matrix, the closed-form solution of $\mathbf{H}$ in (11) is $\mathbf{H} = \mathbf{U}\mathbf{V}^T$.

Proof: Minimizing (11) is equivalent to maximizing $tr(\mathbf{H}^T\mathbf{B})$. We have that the SVD of $\mathbf{B}$ is $\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$. Since $\mathbf{H}$ is column orthogonal, its SVD is $\mathbf{H} = \mathbf{H} * \mathbf{I} * \mathbf{I}$. According to Von Neumann's trace inequality, we have

$$\begin{aligned} tr(\mathbf{H}^T\mathbf{B}) &\le tr(\mathbf{I}\boldsymbol{\Sigma}) \\ &= tr(\mathbf{U}^T\mathbf{B}\mathbf{V}) \\ &= tr((\mathbf{U}\mathbf{V}^T)^T\mathbf{B}) \end{aligned}$$

The equality holds when $\mathbf{H} = \mathbf{U}\mathbf{V}^T$. Therefore, the global optima of (11) is $\mathbf{H} = \mathbf{U}\mathbf{V}^T$. □

2) Optimizing $\mathbf{R}^{(i)}$*:* The subproblem w.r.t. $\mathbf{R}^{(i)}$ is

$$\begin{aligned} \min_{\mathbf{R}^{(i)}} \quad & -tr(\mathbf{R}^{(i)T}\mathbf{C}) \\ \text{s.t.} \quad & \mathbf{R}^{(i)T}\mathbf{R}^{(i)} = \mathbf{I}, \end{aligned} \tag{12}$$

Algorithm 1: Fair Clustering Ensemble.

Input: Multiple base clustering results $\mathcal{C}^1, \ldots, \mathcal{C}^m$, protected groups $\mathcal{G}_1, \ldots, \mathcal{G}_T$, hyper-parameters λ_1 and λ_2.
Output: Final consensus clustering matrix $\mathbf{Y}$
1: Construct base clustering result matrices $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)}$ and the one-hot protected attribute matrix $\mathbf{G}$.
2: Initialize $\mathbf{R} = \mathbf{I}$, $\mathbf{R}^{(i)} = \mathbf{I}$, and $\alpha_i = \frac{1}{m}$. Initialize $\mathbf{H}$ by minimizing $\sum_{i=1}^m \|\mathbf{H} - \mathbf{Y}^{(i)}\mathbf{R}^{(i)}\|_F^2$.
3: **while** not converges **do**
4: Update $\mathbf{Y}$ by solving (14).
5: Update $\mathbf{R}$ by solving (13).
6: Update $\mathbf{R}^{(i)}$ by solving (12).
7: Update $\mathbf{H}$ by solving (11).
8: Update α_i by (16).
9: **end while**

where $\mathbf{C} = \mathbf{Y}^{(i)T}\mathbf{H}$. It can also be solved by Theorem 1, the solution is the SVD of $\mathbf{C}$.

3) Optimizing $\mathbf{R}$*:* The subproblem w.r.t. $\mathbf{R}$ is

$$\min_{\mathbf{R}} \quad -tr(\mathbf{R}^T\mathbf{E}), \qquad \text{s.t.} \quad \mathbf{R}^T\mathbf{R} = \mathbf{I}, \tag{13}$$

where $\mathbf{E} = \mathbf{H}^T\mathbf{Y}$. The solution is also similar to the one of (11), which is the SVD of $\mathbf{E}$.

4) Optimizing $\mathbf{Y}$*:* When optimizing $\mathbf{Y}$, we have the following formula

$$\min_{\mathbf{Y}} \quad \|\mathbf{Y} - \mathbf{H}\mathbf{R}\|_F^2 + \lambda_2\|\mathbf{G}^T\mathbf{Y}\|_F^2 \qquad \text{s.t.} \quad \mathbf{Y} \in \{0,1\}^{n\times c}, \ \sum_{j=1}^c Y_{ij} = 1. \tag{14}$$

Notice that in each row of $\mathbf{Y}$ there is only one 1 and other elements are zeros. Therefore, we can solve (14) row by row. When solving the i-th row, we replace the i-th row by $[1, 0, \ldots, 0], [0, 1, 0, \ldots, 0], \ldots, [0, \ldots, 0, 1]$ respectively, and select the one that has the lowest objective function value as the solution of the i-th row as [64] did.

5) Optimizing α_i*:* When optimizing α_i, we have the following subproblem

$$\min_{\alpha_i} \quad \sum_{i=1}^m \alpha_i^2 \|\mathbf{H} - \mathbf{Y}^{(i)}\mathbf{R}^{(i)}\|_F^2 \qquad \text{s.t.} \quad 0 \le \alpha_i \le 1, \ \sum_{i=1}^m \alpha_i = 1. \tag{15}$$

According to Cauchy-Schwarz Inequality, we obtain the closed-form solution of α_i as:

$$\alpha_i = \frac{\|\mathbf{H} - \mathbf{Y}^{(i)}\mathbf{R}^{(i)}\|_F^{-2}}{\sum_{j=1}^m \|\mathbf{H} - \mathbf{Y}^{(j)}\mathbf{R}^{(j)}\|_F^{-2}}. \tag{16}$$

Algorithm 1 summarizes the process of our FCE. Notice that when solving each subproblem, the objective function value decreases monotonously, and the objective function has a lower bound, which guarantees the convergence of Algorithm 1.

D. Discussions

Now we briefly analyze the time complexity of Algorithm 1. When solving $\mathbf{H}$, $\mathbf{R}^{(i)}$, and $\mathbf{R}$, we need the SVD of n-by-c, c-by-c, and c-by-c matrices, respectively. The time complexity of the SVDs are $O(nc^2)$, $O(c^3)$, and $O(c^3)$, respectively. Notice that c is the number of clusters which is often small in real applications. When solving one row of $\mathbf{Y}$, we need some matrix multiplications, whose time complexity is $O(nc^2)$. Therefore, updating the whole $\mathbf{Y}$ needs $O(n^2c^2)$ time. Updating α_i costs $O(nc^2)$ time. To sum up, the time complexity is $O(n^2c^2)$. Notice that the bottleneck is the matrix multiplication when solving $\mathbf{Y}$ and it is often fast in practice [64]. We can also easily parallelize the matrix multiplication for further speedup.

The main part of the method is the fairness regularized term $\|\mathbf{G}^T\mathbf{Y}\|_F^2$, which is simple yet effective. Notice that this term only needs the pseudo-labels $\mathbf{Y}$ and protected groups matrix $\mathbf{G}$. Therefore, this term can be plugged into many other machine learning methods easily to improve fairness, such as k-means and feature selection. This term is a kind of fairness regularized term with universality.

Although the motivation of the regularized term $\|\mathbf{G}^T\mathbf{Y}\|_F^2$ is to improve fairness, we observe that this term can also be used for clustering with equal capacity. We just need to put all instances into one protected group, which means that we let $\mathbf{G}$ be an n-dimensional vector whose elements are all 1's, and then this term, i.e., $\|\mathbf{1}^T\mathbf{Y}\|_F^2$, degenerates to an equal cluster capacity regularized term. Therefore, the clustering ensemble with equal cluster capacity is a special case of our proposed framework.

IV. Experiments

In this section, we conduct experiments on some benchmark data sets to show the effectiveness of the proposed method.

A. Data Sets

We conduct experiments on six widely-used data sets in fair machine learning works, including D&S [65], HAR [66], MNIST-USPS [18], Reverse MNIST [18], JAFFE [67], Yale [68]. D&S is a human daily and sports activities data set including 8 participants. HAR is a human action recognition data set including 30 participants. In both D&S and HAR data sets, the instances of each participant form a protected group. MNIST-USPS is an image data set containing images of handwritten digits from MNIST[1] and USPS[2] data. Following [18], we randomly sample 2000 images from MNIST to form one protected group and randomly sample 1800 images from USPS to form the other protected group. Reverse MNIST is an image data set generated from MNIST. Also following [18], we randomly sample 2000 images from MNIST to form one protected group and randomly sample 2000 images and reverse them to form the other protected group. JAFFE is a face image data set. Following [18], we put the

[1]http://yann.lecun.com/exdb/mnist
[2]https://www.kaggle.com/bistaumanga/usps-dataset

TABLE I
DESCRIPTION OF THE DATA SETS

Data sets	# of Instances	# of Features	# of Cluster	Protected Groups
D&S	9120	5625	19	Person Identity (8)
HAR	10299	561	6	Person Identity (30)
MNIST-USPS	3800	256	10	Source of images (2)
Reverse MNIST	4000	784	10	Original or reversed (2)
JAFFE	213	676	10	Expression (7)
Yale	165	1024	15	w/o glasses (2)

face images with the same expressions into a protected group. Yale is also a face image data set. Following [18], the people wearing glasses form a protected group and other people form the other protected group. The detailed information of these data sets is summarized in Table I,

B. Experimental Setup

To evaluate the performance of the proposed method, we conduct two groups of experiments. In the first group of experiments, following a similar experimental protocol in [69], we run k-means 100 times with different initializations to obtain 100 base clustering results. We divide these 100 base results into 10 subsets. Then we run clustering ensemble methods on the 10 subsets and report the average results and standard deviation on the 10 subsets. To evaluate the performance on the base results with more diversity, we conduct the second group of experiments, which is a comparison on different base clustering algorithms. Specifically, in each subset, we ensemble 10 base results including 3 k-means results, 3 spectral clustering results, 3 hierarchical clustering results, and 1 kernel k-means result. Other setups are the same as the first group of experiments.

We compare our FCE with the following eleven mainstream clustering ensemble methods:

- **BCE** [69], which is a probability framework for ensemble to generate a stable consensus result.
- **RCE** [70], which minimizes Kullback Leibler (KL) divergence between each base result and learns a robust consensus result.
- **LWGP** [71], which applies a local weighting strategy to a graph partition consensus clustering method.
- **LWEA** [71], which applies a local weighting strategy to an agglomerative consensus clustering method.
- **DREC** [72], which is a clustering ensemble method based on the dense representation.
- **RSEC** [26], which is a spectral-based robust clustering ensemble method.
- **TRCE** [32], which is a tri-level robust clustering ensemble method.
- **ECPCS-MC** [73], which is a clustering ensemble method by propagating cluster-wise similarities with a meta-cluster.
- **ECPCS-HC** [73], which is a clustering ensemble method by propagating cluster-wise similarities with hierarchical consensus function.
- **CESHL** [74], which is a consensus clustering method with structured hypergraph learning.
- **PFREFF** [75], which is a parameter-free robust ensemble framework for fuzzy clustering.

In addition, we also report **KM**, which is the average result of all base clusterings. To show the effectiveness of our designed fairness regularized term, we also conduct an ablation study by comparing it with our degenerated version **FCE-f**, which removes the fairness regularized term (i.e., (8)). In FCE, we use rotation matrices $\mathbf{R}^{(i)}$ to align the clusters in each base result. Another straightforward way to align the clusters is to use the Hungary algorithm. To show the effectiveness of our rotation matrices, we also compare our FCE with a variant that first aligns the clusters with Hungary algorithm and then does ensemble with our designed fairness regularized term. We denote this version as **FCE-a**.

For all methods and all data sets, we set the number of clusters c as the true number of classes. We fix the hyper-parameter λ_1 to 0.001 as introduced before. The hyper-parameter λ_2 controls the fairness and cluster capacity equality, and we tune it in $[10^{-5}, 10^{1}]$. For other methods, we tune the hyper-parameters as their papers suggested. For example, in RSEC, following the authors' suggestion, we tune λ_1 in the set $\{0.01, 0.1, 1\}$ and λ_2 in the set $\{0.1, 1\}$. In DREC, we set λ as 100.

We use ACCuracy (ACC) and Normalized Mutual Information (NMI) to measure the clustering performance. We use Balance (Bal) [13] and Minimal Normalized Conditional Entropy (MNCE) [16] to evaluate the fairness. In more detail, Bal is defined as

$$\mathrm{Bal}\,(\mathcal{C}) = \min_k \left(\frac{N_k^{\min}}{N_k^{\max}} \right) \in [0,1], \tag{17}$$

where N_k^{min} and N_k^{max} denote the number of instances in the smallest and the largest (in size) protected groups in cluster π_k, respectively. MNCE is defined as

$$\mathrm{MNCE} = \frac{\min_k \left(-\sum_i \frac{|\mathcal{G}_i \cap \pi_k|}{|\pi_k|} \log \frac{|\mathcal{G}_i \cap \pi_k|}{|\pi_k|} \right)}{-\sum_i \frac{|\mathcal{G}_i|}{n} \log \frac{|\mathcal{G}_i|}{n}} \in [0,1]. \tag{18}$$

In addition, we also use our Definition 2, denoted as CCE, and Normalized Entropy (NE) [61] to measure the cluster capacity equality. NE is defined as

$$\mathrm{NE} = -\frac{1}{\log(c)} \sum_{k=1}^{c} \frac{|\pi_k|}{n} \log \left(\frac{|\pi_k|}{n} \right) \in [0,1]. \tag{19}$$

In addition to the above indicators, we also report fairness_CCE(f_CCE) in Definition 3.

All metrics are the larger the better.

TABLE II
EXPERIMENTAL RESULTS ON MNIST-USPS AND REVERSE-MNIST DATA SETS

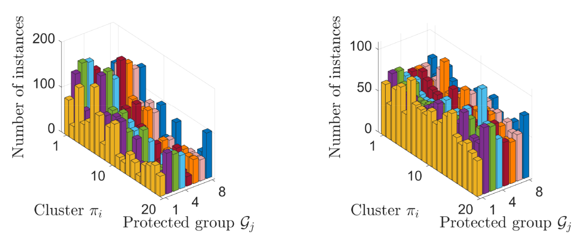
Methods	MNIST-USPS							Reverse MNIST						
	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE
KM	0.371 ±0.012	0.343 ±0.016	0.001 ±0.003	0.006 ±0.012	0.320 ±0.040	0.966 ±0.014	0.000 ±0.000	0.318 ±0.015	0.285 ±0.021	0.003 ±0.009	0.011 ±0.025	0.293 ±0.037	0.961 ±0.016	0.004 ±0.035
BCE [69]	0.385 ±0.017	0.389 ±0.009	0.000 ±0.000	0.000 ±0.000	0.276 ±0.106	0.974 ±0.012	0.000 ±0.000	0.346 ±0.025	0.335 ±0.021	0.000 ±0.000	0.000 ±0.000	0.237 ±0.101	0.962 ±0.015	0.000 ±0.000
RCE [70]	0.384 ±0.015	0.400 ±0.006	0.000 ±0.000	0.000 ±0.000	0.372 ±0.061	0.980 ±0.005	0.000 ±0.000	0.350 ±0.005	0.367 ±0.013	0.000 ±0.000	0.000 ±0.000	0.345 ±0.018	0.976 ±0.002	0.000 ±0.000
LWGP [71]	0.366 ±0.017	0.369 ±0.031	0.000 ±0.000	0.000 ±0.000	0.180 ±0.136	0.955 ±0.021	0.000 ±0.000	0.348 ±0.008	0.362 ±0.016	0.000 ±0.000	0.000 ±0.000	0.314 ±0.086	0.973 ±0.007	0.000 ±0.000
LWEA [71]	0.390 ±0.024	0.361 ±0.017	0.000 ±0.000	0.000 ±0.000	0.004 ±0.004	0.876 ±0.040	0.000 ±0.000	0.344 ±0.012	0.346 ±0.021	0.000 ±0.000	0.000 ±0.000	0.193 ±0.170	0.942 ±0.033	0.000 ±0.000
DREC [72]	0.387 ±0.017	0.399 ±0.013	0.000 ±0.000	0.000 ±0.000	0.341 ±0.110	0.975 ±0.013	0.000 ±0.000	0.342 ±0.006	0.358 ±0.011	0.000 ±0.000	0.000 ±0.000	0.296 ±0.056	0.967 ±0.011	0.000 ±0.000
RSEC [26]	0.383 ±0.017	0.405 ±0.009	0.000 ±0.000	0.000 ±0.000	0.377 ±0.045	0.983 ±0.004	0.000 ±0.000	0.351 ±0.006	0.366 ±0.013	0.000 ±0.000	0.000 ±0.000	0.343 ±0.052	0.979 ±0.004	0.000 ±0.000
TRCE [32]	0.379 ±0.016	0.400 ±0.011	0.000 ±0.000	0.000 ±0.000	0.326 ±0.043	0.976 ±0.005	0.000 ±0.000	0.332 ±0.006	0.342 ±0.011	0.000 ±0.000	0.000 ±0.000	0.319 ±0.035	0.976 ±0.004	0.000 ±0.000
ECPCS-HC [73]	0.306 ±0.025	0.203 ±0.035	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.582 ±0.065	0.000 ±0.000	0.297 ±0.028	0.244 ±0.058	0.000 ±0.000	0.000 ±0.000	0.034 ±0.105	0.736 ±0.104	0.000 ±0.000
ECPCS-MC [73]	0.374 ±0.016	0.394 ±0.022	0.000 ±0.000	0.000 ±0.000	0.350 ±0.057	0.979 ±0.004	0.000 ±0.000	0.349 ±0.007	**0.368** ±0.005	0.000 ±0.000	0.000 ±0.000	0.337 ±0.030	0.976 ±0.003	0.000 ±0.000
CESHL [74]	0.342 ±0.053	0.314 ±0.070	0.000 ±0.000	0.000 ±0.000	0.002 ±0.003	0.793 ±0.155	0.100 ±0.316	0.327 ±0.039	0.311 ±0.073	0.000 ±0.000	0.000 ±0.000	0.132 ±0.170	0.847 ±0.145	0.000 ±0.000
PFREFF [75]	0.371 ±0.016	0.403 ±0.011	0.000 ±0.000	0.000 ±0.000	0.321 ±0.107	0.974 ±0.014	0.000 ±0.000	0.350 ±0.004	0.359 ±0.013	0.000 ±0.000	0.000 ±0.000	0.290 ±0.080	0.970 ±0.015	0.000 ±0.000
FCE-f	0.395 ±0.014	**0.406** ±0.009	0.000 ±0.000	0.000 ±0.000	0.391 ±0.065	0.985 ±0.003	0.000 ±0.000	**0.352** ±0.007	**0.368** ±0.012	0.000 ±0.000	0.000 ±0.000	0.340 ±0.121	0.977 ±0.013	0.000 ±0.000
FCE-a	0.389 ±0.046	0.381 ±0.055	0.098 ±0.054	0.382 ±0.113	0.531 ±0.153	0.982 ±0.017	0.141 ±0.015	0.302 ±0.016	0.243 ±0.017	**0.342** ±0.042	**0.816** ±0.036	0.748 ±0.025	0.998 ±0.000	**0.186** ±0.042
FCE	**0.419** ±0.019	0.385 ±0.013	**0.105** ±0.037	**0.445** ±0.094	**0.703** ±0.084	**0.997** ±0.002	**0.190** ±0.052	0.328 ±0.011	0.266 ±0.014	0.339 ±0.030	0.814 ±0.027	**0.774** ±0.021	**0.999** ±0.000	0.174 ±0.051

FCE-f represents the degenerated version of FCE without the fairness regularized term and FCE-a denotes the version that aligns the base results with the Hungary algorithm. The best and second best results are denoted in bold and underlined, respectively.

C. Experimental Results

We report the results of all clustering ensemble methods on all k-means base results in Tables II, III, and IV. The best results are denoted in bold and the second best results are denoted underlined. From these Tables, we can find that our FCE outperforms other compared methods on all data sets w.r.t. fairness (i.e., Bal and MNCE) and cluster capacity equality (i.e., CCE and NE), demonstrating our method's motivation. When comparing w.r.t. the clustering performance, i.e., ACC and NMI, although our method focuses on fairness and cluster capacity equality, it is still comparable with other compared methods on many data sets and even better than them on some data sets.

When compared with the degenerated version FCE-f, which is without the designed fairness regularized term, FCE performs better w.r.t. the fairness and the cluster capacity equality. It demonstrates the effectiveness of our designed fairness regularized term. In addition, when removing this fairness regularized term, the ACC and NMI of FCE-f can outperform other compared methods, which shows the effectiveness of our ensemble strategy. Moreover, it is interesting to see that the clustering performance of ACC and NMI sometimes is even better than the original FCE. It shows that sometimes we can only achieve a trade-off between accuracy, fairness, and cluster capacity equality. Notice that when computing the ACC and NMI, we need the ground truth of the data sets. However, the ground truth of some data sets may be naturally imbalanced or unfair. On these data sets, when improving the fairness and cluster capacity equality, we should sacrifice the clustering performance to some extent. Notice that in our setting, we assume that the base results have the same number of clusters. However, our method can be extended to handle the base results with different numbers of clusters. The detailed results are shown in the Appendix.



(a) Before learning (b) After learning

Fig. 3. The distribution of different protection groups in each cluster before and after learning on the D&S data set.

To further show the fairness, we show some visualization results in Fig. 3. Fig. 3 shows the number of instances of each protected group $\mathcal{G}_j$ in each cluster π_i in the D&S data set before and after learning. Fig. 3(a) shows the result before learning. The numbers of instances of one protected group in each cluster have a great difference, which means the base results are unfair. Fig. 3(b) shows the number of instances of each protected group in each cluster after learning, which is much more fair than the

TABLE III
EXPERIMENTAL RESULTS ON D&S AND HAR DATA SETS

Methods	D&S							HAR						
	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE
KM	0.347 ±0.014	0.475 ±0.009	0.000 ±0.000	0.021 ±0.023	0.002 ±0.000	0.703 ±0.020	0.000 ±0.000	0.579 ±0.018	0.571 ±0.020	0.000 ±0.000	0.812 ±0.031	0.241 ±0.034	0.940 ±0.014	0.003 ±0.003
BCE [69]	0.388 ±0.036	0.468 ±0.016	0.000 ±0.000	0.683 ±0.253	0.011 ±0.012	0.807 ±0.055	0.000 ±0.000	0.538 ±0.024	0.534 ±0.017	0.000 ±0.000	0.699 ±0.085	0.176 ±0.137	0.908 ±0.072	0.000 ±0.000
RCE [70]	0.386 ±0.019	0.500 ±0.012	0.000 ±0.000	0.020 ±0.007	0.008 ±0.002	0.710 ±0.015	0.000 ±0.000	0.623 ±0.016	**0.604** ±0.016	0.000 ±0.000	0.792 ±0.060	0.281 ±0.107	0.953 ±0.029	0.000 ±0.000
LWGP [71]	0.351 ±0.026	0.457 ±0.016	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.622 ±0.026	0.000 ±0.000	0.616 ±0.012	0.595 ±0.007	0.000 ±0.000	0.799 ±0.033	0.307 ±0.044	0.968 ±0.006	0.000 ±0.000
LWEA [71]	0.300 ±0.033	0.425 ±0.020	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.571 ±0.037	0.000 ±0.000	0.615 ±0.015	0.581 ±0.011	0.000 ±0.000	0.657 ±0.242	0.170 ±0.119	0.919 ±0.043	0.000 ±0.000
DREC [72]	0.339 ±0.027	0.489 ±0.014	0.000 ±0.000	0.114 ±0.122	0.008 ±0.003	0.720 ±0.022	0.000 ±0.000	0.605 ±0.010	0.583 ±0.011	0.000 ±0.000	0.831 ±0.046	0.206 ±0.139	0.924 ±0.046	0.000 ±0.000
RSEC [26]	0.407 ±0.016	0.525 ±0.007	0.000 ±0.000	0.106 ±0.201	0.051 ±0.024	0.874 ±0.036	0.000 ±0.000	0.609 ±0.012	0.581 ±0.010	0.000 ±0.000	0.806 ±0.012	0.342 ±0.099	0.973 ±0.007	0.000 ±0.000
TRCE [32]	0.401 ±0.020	0.519 ±0.011	0.000 ±0.000	0.018 ±0.010	0.008 ±0.003	0.750 ±0.011	0.000 ±0.000	0.613 ±0.013	0.591 ±0.010	0.000 ±0.000	0.789 ±0.030	0.282 ±0.031	0.965 ±0.004	0.000 ±0.000
ECPCS-HC [73]	0.304 ±0.012	0.441 ±0.005	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.578 ±0.015	0.000 ±0.000	0.492 ±0.121	0.506 ±0.108	0.000 ±0.000	0.126 ±0.240	0.001 ±0.003	0.593 ±0.186	0.000 ±0.000
ECPCS-MC [73]	0.369 ±0.028	0.491 ±0.017	0.000 ±0.000	0.005 ±0.010	0.002 ±0.003	0.718 ±0.026	0.000 ±0.000	0.605 ±0.015	0.589 ±0.010	0.000 ±0.000	0.786 ±0.036	0.262 ±0.031	0.960 ±0.008	0.000 ±0.000
CESHL [74]	0.161 ±0.075	0.181 ±0.141	0.000 ±0.000	0.000 ±0.000	0.129 ±0.148	0.864 ±0.109	0.000 ±0.000	0.584 ±0.064	0.575 ±0.056	0.000 ±0.000	0.393 ±0.416	0.129 ±0.148	0.835 ±0.145	0.000 ±0.000
PFREFF [75]	0.397 ±0.032	0.517 ±0.016	0.000 ±0.000	0.039 ±0.069	0.010 ±0.005	0.761 ±0.018	0.000 ±0.000	0.602 ±0.028	0.582 ±0.026	0.000 ±0.000	0.789 ±0.030	0.265 ±0.057	0.956 ±0.022	0.000 ±0.000
FCE-f	0.415 ±0.025	**0.541** ±0.019	0.000 ±0.000	0.030 ±0.019	0.036 ±0.023	0.854 ±0.034	0.000 ±0.000	0.633 ±0.030	0.600 ±0.018	0.000 ±0.000	0.858 ±0.052	0.393 ±0.155	0.972 ±0.020	0.000 ±0.000
FCE-a	0.392 ±0.025	0.463 ±0.019	0.213 ±0.055	0.905 ±0.021	0.227 ±0.011	0.958 ±0.004	0.332 ±0.005	0.600 ±0.021	0.527 ±0.033	**0.194** ±0.100	**0.986** ±0.060	0.569 ±0.096	0.989 ±0.007	**0.403** ±0.190
FCE	**0.425** ±0.019	0.482 ±0.016	**0.222** ±0.038	**0.920** ±0.034	**0.387** ±0.049	**0.985** ±0.002	**0.374** ±0.048	**0.648** ±0.026	0.557 ±0.024	0.149 ±0.148	0.977 ±0.002	**0.731** ±0.045	**0.997** ±0.000	0.331 ±0.050

FCE-f represents the degenerated version of FCE without the fairness regularized term and FCE-a denotes the version that aligns the base results with the Hungary algorithm. The best and second best results are denoted in bold and underlined, respectively.

TABLE IV
EXPERIMENTAL RESULTS ON JAFFE AND YALE DATA SETS

Methods	JAFFE							Yale						
	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE
KM	0.714 ±0.023	0.789 ±0.015	0.000 ±0.000	0.303 ±0.058	0.153 ±0.026	0.935 ±0.006	0.000 ±0.000	0.387 ±0.011	0.446 ±0.013	0.000 ±0.000	0.000 ±0.000	0.064 ±0.016	0.917 ±0.012	0.000 ±0.000
BCE [69]	0.777 ±0.087	0.836 ±0.050	0.000 ±0.000	0.347 ±0.339	0.211 ±0.131	0.953 ±0.030	0.000 ±0.000	0.427 ±0.030	0.475 ±0.030	0.000 ±0.000	0.000 ±0.000	0.140 ±0.095	0.949 ±0.024	0.000 ±0.000
RCE [70]	0.825 ±0.054	0.858 ±0.025	0.000 ±0.000	0.525 ±0.182	0.290 ±0.124	0.974 ±0.013	0.000 ±0.000	0.426 ±0.022	0.483 ±0.019	0.000 ±0.000	0.000 ±0.000	0.142 ±0.061	0.953 ±0.013	0.000 ±0.000
LWGP [71]	0.847 ±0.031	0.870 ±0.026	0.000 ±0.000	0.369 ±0.341	0.247 ±0.145	0.976 ±0.012	0.000 ±0.000	0.408 ±0.029	0.462 ±0.032	0.000 ±0.000	0.000 ±0.000	0.048 ±0.028	0.914 ±0.030	0.000 ±0.000
LWEA [71]	0.780 ±0.059	0.837 ±0.033	0.000 ±0.000	0.275 ±0.317	0.142 ±0.113	0.951 ±0.026	0.000 ±0.000	0.410 ±0.024	0.448 ±0.029	0.000 ±0.000	0.000 ±0.000	0.042 ±0.028	0.893 ±0.020	0.000 ±0.000
DREC [72]	0.792 ±0.036	0.850 ±0.022	0.000 ±0.000	0.208 ±0.244	0.099 ±0.054	0.953 ±0.014	0.000 ±0.000	0.426 ±0.021	0.483 ±0.018	0.000 ±0.000	0.000 ±0.000	0.117 ±0.040	0.945 ±0.013	0.000 ±0.000
RSEC [26]	0.843 ±0.065	0.869 ±0.033	0.000 ±0.000	0.700 ±0.128	0.492 ±0.119	0.988 ±0.014	0.000 ±0.000	0.431 ±0.018	0.493 ±0.009	0.000 ±0.000	0.000 ±0.000	0.281 ±0.070	0.979 ±0.007	0.000 ±0.000
TRCE [32]	0.850 ±0.029	0.863 ±0.017	0.000 ±0.000	0.561 ±0.172	0.308 ±0.109	0.980 ±0.007	0.000 ±0.000	0.424 ±0.026	0.473 ±0.023	0.000 ±0.000	0.000 ±0.000	0.062 ±0.035	0.917 ±0.028	0.000 ±0.000
ECPCS-HC [73]	0.702 ±0.064	0.784 ±0.037	0.000 ±0.000	0.069 ±0.146	0.227 ±0.157	0.901 ±0.028	0.000 ±0.000	0.357 ±0.019	0.395 ±0.024	0.000 ±0.000	0.000 ±0.000	0.022 ±0.004	0.804 ±0.036	0.000 ±0.000
ECPCS-MC [73]	0.792 ±0.054	0.840 ±0.030	0.000 ±0.000	0.348 ±0.378	0.350 ±0.057	0.915 ±0.030	0.000 ±0.000	0.364 ±0.032	0.406 ±0.044	0.000 ±0.000	0.000 ±0.000	0.042 ±0.030	0.909 ±0.023	0.000 ±0.000
CESHL [74]	0.655 ±0.091	0.731 ±0.076	0.000 ±0.000	0.000 ±0.000	0.037 ±0.018	0.863 ±0.058	0.000 ±0.000	0.363 ±0.054	0.405 ±0.060	0.000 ±0.000	0.000 ±0.000	0.023 ±0.012	0.816 ±0.084	0.000 ±0.000
PFREFF [75]	0.816 ±0.046	0.851 ±0.032	0.000 ±0.000	0.565 ±0.165	0.304 ±0.112	0.976 ±0.012	0.000 ±0.000	0.416 ±0.018	0.487 ±0.014	0.000 ±0.000	0.000 ±0.000	0.110 ±0.058	0.942 ±0.017	0.000 ±0.000
FCE-f	0.858 ±0.061	0.872 ±0.042	0.000 ±0.000	0.449 ±0.343	0.318 ±0.180	0.971 ±0.016	0.000 ±0.000	**0.444** ±0.032	**0.509** ±0.027	0.000 ±0.000	0.000 ±0.000	0.241 ±0.091	0.982 ±0.000	0.000 ±0.000
FCE-a	0.809 ±0.089	0.786 ±0.091	0.466 ±0.172	0.979 ±0.029	0.767 ±0.089	0.998 ±0.001	**0.645** ±0.205	0.413 ±0.031	0.459 ±0.035	0.161 ±0.060	0.746 ±0.159	0.626 ±0.103	0.996 ±0.003	0.520 ±0.140
FCE	**0.913** ±0.086	**0.896** ±0.077	**0.483** ±0.032	**0.986** ±0.005	**0.810** ±0.025	**0.999** ±0.001	0.639 ±0.013	0.426 ±0.025	0.476 ±0.017	**0.194** ±0.048	**0.838** ±0.127	**0.806** ±0.070	**0.999** ±0.004	**0.526** ±0.094

FCE-f represents the degenerated version of FCE without the fairness regularized term and FCE-a denotes the version that aligns the base results with the Hungary algorithm. The best and second best results are denoted in bold and underlined, respectively.

TABLE V
EXPERIMENTAL RESULTS WITH DIFFERENT BASE CLUSTERING ALGORITHMS ON MNIST-USPS AND REVERSE-MNIST DATA SETS

Methods	MNIST-USPS							Reverse MNIST						
	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE
KM	0.338 $\pm$0.028	0.355 $\pm$0.070	0.000 $\pm$0.000	0.000 $\pm$0.000	0.149 $\pm$0.078	0.895 $\pm$0.059	0.000 $\pm$0.000	0.369 $\pm$0.023	0.430 $\pm$0.052	0.000 $\pm$0.000	0.000 $\pm$0.000	0.305 $\pm$0.101	0.966 $\pm$0.018	0.000 $\pm$0.000
BCE [69]	0.354 $\pm$0.023	0.386 $\pm$0.020	0.000 $\pm$0.000	0.000 $\pm$0.000	0.132 $\pm$0.053	0.923 $\pm$0.016	0.000 $\pm$0.000	0.382 $\pm$0.019	0.429 $\pm$0.024	0.000 $\pm$0.000	0.000 $\pm$0.000	0.153 $\pm$0.106	0.945 $\pm$0.022	0.000 $\pm$0.000
RCE [70]	0.363 $\pm$0.002	0.398 $\pm$0.006	0.000 $\pm$0.000	0.000 $\pm$0.000	0.119 $\pm$0.004	0.908 $\pm$0.002	0.000 $\pm$0.000	0.393 $\pm$0.003	0.464 $\pm$0.001	0.000 $\pm$0.000	0.000 $\pm$0.000	0.354 $\pm$0.005	0.979 $\pm$0.001	0.000 $\pm$0.000
LWGP [71]	0.350 $\pm$0.003	0.401 $\pm$0.003	0.000 $\pm$0.000	0.000 $\pm$0.000	0.148 $\pm$0.003	0.916 $\pm$0.001	0.000 $\pm$0.000	0.389 $\pm$0.002	0.468 $\pm$0.002	0.000 $\pm$0.000	0.000 $\pm$0.000	0.316 $\pm$0.005	0.973 $\pm$0.002	0.000 $\pm$0.000
LWEA [71]	<u>0.371</u> $\pm$0.011	0.404 $\pm$0.014	0.000 $\pm$0.000	0.000 $\pm$0.000	0.111 $\pm$0.032	0.903 $\pm$0.013	0.000 $\pm$0.000	0.386 $\pm$0.008	0.465 $\pm$0.004	0.000 $\pm$0.000	0.000 $\pm$0.000	0.299 $\pm$0.005	0.968 $\pm$0.003	0.000 $\pm$0.000
DREC [72]	**0.384** $\pm$0.022	**0.421** $\pm$0.020	0.000 $\pm$0.000	0.000 $\pm$0.000	0.170 $\pm$0.028	0.937 $\pm$0.008	0.000 $\pm$0.000	**0.403** $\pm$0.005	**0.470** $\pm$0.011	0.000 $\pm$0.000	0.000 $\pm$0.000	0.322 $\pm$0.039	0.974 $\pm$0.010	0.000 $\pm$0.000
RSEC [26]	0.355 $\pm$0.008	0.393 $\pm$0.015	0.000 $\pm$0.000	0.000 $\pm$0.000	0.174 $\pm$0.067	0.941 $\pm$0.027	0.000 $\pm$0.000	0.387 $\pm$0.006	0.453 $\pm$0.012	0.000 $\pm$0.000	0.000 $\pm$0.000	0.313 $\pm$0.064	0.976 $\pm$0.005	0.000 $\pm$0.000
TRCE [32]	0.339 $\pm$0.026	0.361 $\pm$0.032	0.000 $\pm$0.000	0.000 $\pm$0.000	0.116 $\pm$0.021	0.912 $\pm$0.004	0.000 $\pm$0.000	0.362 $\pm$0.010	0.428 $\pm$0.009	0.000 $\pm$0.000	0.000 $\pm$0.000	0.304 $\pm$0.025	0.962 $\pm$0.005	0.000 $\pm$0.000
ECPCS-HC [73]	0.261 $\pm$0.003	0.189 $\pm$0.003	0.000 $\pm$0.000	0.000 $\pm$0.000	0.000 $\pm$0.000	0.600 $\pm$0.015	0.000 $\pm$0.000	0.315 $\pm$0.013	0.354 $\pm$0.017	0.000 $\pm$0.000	0.000 $\pm$0.000	0.002 $\pm$0.000	0.798 $\pm$0.024	0.000 $\pm$0.000
ECPCS-MC [73]	0.352 $\pm$0.009	0.380 $\pm$0.009	0.000 $\pm$0.000	0.000 $\pm$0.000	0.133 $\pm$0.010	0.913 $\pm$0.007	0.000 $\pm$0.000	0.398 $\pm$0.003	0.463 $\pm$0.001	0.000 $\pm$0.000	0.000 $\pm$0.000	0.350 $\pm$0.005	0.976 $\pm$0.002	0.000 $\pm$0.000
CESHL [74]	0.301 $\pm$0.035	0.281 $\pm$0.069	0.000 $\pm$0.000	0.000 $\pm$0.000	0.046 $\pm$0.043	0.769 $\pm$0.086	0.000 $\pm$0.000	0.398 $\pm$0.002	0.462 $\pm$0.003	0.000 $\pm$0.000	0.000 $\pm$0.000	0.346 $\pm$0.005	0.975 $\pm$0.002	0.000 $\pm$0.000
PFREFF [75]	0.361 $\pm$0.004	0.395 $\pm$0.013	0.000 $\pm$0.000	0.000 $\pm$0.000	0.128 $\pm$0.050	0.912 $\pm$0.019	0.000 $\pm$0.000	0.395 $\pm$0.006	0.465 $\pm$0.002	0.000 $\pm$0.000	0.000 $\pm$0.000	0.348 $\pm$0.002	0.979 $\pm$0.002	0.000 $\pm$0.000
FCE-f	0.368 $\pm$0.010	<u>0.407</u> $\pm$0.015	0.000 $\pm$0.000	0.000 $\pm$0.000	0.185 $\pm$0.045	0.945 $\pm$0.012	0.000 $\pm$0.000	<u>0.400</u> $\pm$0.008	<u>0.469</u> $\pm$0.017	0.000 $\pm$0.000	0.000 $\pm$0.000	0.324 $\pm$0.111	0.976 $\pm$0.013	0.000 $\pm$0.000
FCE-a	0.341 $\pm$0.141	0.277 $\pm$0.011	**0.303** $\pm$0.021	**0.783** $\pm$0.021	<u>0.618</u> $\pm$0.048	<u>0.994</u> $\pm$0.002	**0.552** $\pm$0.021	0.343 $\pm$0.004	0.319 $\pm$0.003	<u>0.296</u> $\pm$0.005	<u>0.775</u> $\pm$0.005	<u>0.672</u> $\pm$0.008	<u>0.996</u> $\pm$0.000	**0.532** $\pm$0.007
FCE	0.360 $\pm$0.022	0.331 $\pm$0.013	<u>0.241</u> $\pm$0.038	<u>0.707</u> $\pm$0.053	**0.749** $\pm$0.045	**0.998** $\pm$0.002	<u>0.403</u> $\pm$0.079	0.359 $\pm$0.012	0.356 $\pm$0.009	**0.311** $\pm$0.023	**0.790** $\pm$0.022	**0.754** $\pm$0.026	**0.998** $\pm$0.001	<u>0.468</u> $\pm$0.032

FCE-f represents the degenerated version of FCE without the fairness regularized term and FCE-a denotes the version that aligns the base results with the Hungary algorithm. The best and second best results are denoted in bold and <u>underlined</u>, respectively.

results before learning. It well shows that our FCE can effectively achieve fairness as a post-processing for the standard clustering method.

The results on different base clustering algorithms are shown in Tables V, VI, and VII. We can see that our method also achieves better fairness and cluster capacity equality in this setting.

When comparing with FCE-a, we observe that FCE often outperforms FCE-a. The main reason is that the Hungary algorithm may be a little inappropriate for our clustering ensemble task. Hungary algorithm is a kind of methods for "hard" alignment or matching, which means there should be a bijection between two objects. However, in the clustering ensemble, the "hard" matching for clusters may be unrealistic and there may even not exist such a bijection, because each base result may be in different semantic spaces. That is why the rotations method, which is a "soft" alignment method, outperforms the Hungary algorithm. Notice that, on some data sets, FCE-a achieves comparable performance on fairness. That is because, in FCE-a, we obtain the final clustering result also with our designed fairness regularized term. This term enforces the results to be fair despite that the aligned base results $\mathbf{Y}^{(i)}$ may be not good enough.

D. Comparison With Fair Clustering Methods

To show the effectiveness of our method on fairness, we also compare it with some state-of-the-art fair clustering methods, including:

- **SpFC** [15], which embeds the fairness constraints into the Laplacian matrix of a graph for clustering.
- **VFC** [46], which is a universal variational fair clustering framework.
- **FFC** [76], which is a three-stage fair clustering method based on k-means algorithm.
- **KFC** [77], which is a flexible fair clustering method based on k-center algorithm.
- **SFD** [19], which is a fast fair decomposition algorithm based on fairlet subsets.
- **CFC** [52], which is a robust fair clustering framework via consensus k-means.

For fair clustering methods, we also set the hyperparameters according to the suggestions in their papers. Specifically, in SpFC, we first compute the similarity matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$, whose (p,q)-th element is $S_{pq} = e^{-\frac{\|\mathbf{x}_{p.} - \mathbf{x}_{q.}\|_2^2}{2\sigma^2}}$, where σ is a bandwidth parameter and is set as 0.5. Then we construct the k-NN graph from $\mathbf{S}$ with the number of neighbors $k = 15$. In VFC, we set the clustering mode as k-means and search λ to find the best trade-off between clustering performance and fairness in a range $[1, 10]$. In FFC, the balance parameter δ is set as 0.2. For KFC, we use the default parameter value of $\delta = 0.1$. For SFD, we set the parameters $\alpha = \{1, 2\}$ and $\beta = 5$. In CFC, it takes the same base clustering results as ours as the inputs to construct the input graph for further learning.

We show the results in Table VIII. The number in the parentheses denotes the rank of the method w.r.t. the evaluation metric. We also report the average rank over all metrics of each method, which is denoted as **avg rank**. Notice that FFC cannot run a

TABLE VI
EXPERIMENTAL RESULTS WITH DIFFERENT BASE CLUSTERING ALGORITHMS ON D&S AND HAR DATA SETS

Methods	D&S							HAR						
	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE
KM	0.537 ±0.058	0.644 ±0.049	0.000 ±0.000	0.118 ±0.188	0.108 ±0.083	0.919 ±0.056	0.000 ±0.000	0.540 ±0.072	0.532 ±0.090	0.000 ±0.000	0.656 ±0.260	0.154 ±0.109	0.887 ±0.089	0.003 ±0.000
BCE [69]	0.548 ±0.032	0.653 ±0.021	0.000 ±0.000	0.096 ±0.167	0.082 ±0.070	0.940 ±0.017	0.000 ±0.000	0.594 ±0.061	**0.616** ±0.011	0.000 ±0.000	0.768 ±0.170	0.183 ±0.119	0.930 ±0.038	0.000 ±0.000
RCE [70]	0.574 ±0.004	0.686 ±0.003	0.000 ±0.000	0.023 ±0.001	0.123 ±0.007	0.948 ±0.001	0.000 ±0.000	0.581 ±0.022	0.582 ±0.011	0.000 ±0.000	0.878 ±0.080	0.463 ±0.109	0.980 ±0.013	0.000 ±0.000
LWGP [71]	0.576 ±0.006	0.681 ±0.004	0.000 ±0.000	0.013 ±0.011	0.131 ±0.005	0.948 ±0.003	0.000 ±0.000	0.531 ±0.009	0.573 ±0.010	0.000 ±0.000	0.933 ±0.004	0.308 ±0.005	0.958 ±0.001	0.000 ±0.000
LWEA [71]	0.578 ±0.013	0.678 ±0.008	0.000 ±0.000	0.006 ±0.011	0.062 ±0.062	0.945 ±0.945	0.000 ±0.000	0.584 ±0.025	0.578 ±0.009	0.000 ±0.000	0.750 ±0.220	0.353 ±0.270	0.955 ±0.046	0.000 ±0.000
DREC [72]	0.573 ±0.023	0.689 ±0.012	0.000 ±0.000	0.049 ±0.100	0.054 ±0.044	0.942 ±0.012	0.000 ±0.000	0.585 ±0.020	0.585 ±0.011	0.000 ±0.000	0.620 ±0.001	0.113 ±0.025	0.919 ±0.038	0.000 ±0.000
RSEC [26]	0.580 ±0.015	0.669 ±0.012	0.000 ±0.000	0.246 ±0.169	0.137 ±0.058	0.964 ±0.006	0.000 ±0.000	**0.621** ±0.040	0.595 ±0.018	0.000 ±0.000	0.909 ±0.001	0.496 ±0.057	0.984 ±0.009	0.000 ±0.000
TRCE [32]	0.579 ±0.004	0.688 ±0.003	0.000 ±0.000	0.020 ±0.007	0.124 ±0.006	0.946 ±0.003	0.000 ±0.000	0.601 ±0.059	0.613 ±0.023	0.000 ±0.000	0.745 ±0.200	0.231 ±0.134	0.949 ±0.015	0.000 ±0.000
ECPCS-HC [73]	0.519 ±0.024	0.628 ±0.021	0.000 ±0.000	0.005 ±0.009	0.001 ±0.000	0.826 ±0.031	0.000 ±0.000	0.544 ±0.001	0.581 ±0.000	0.000 ±0.000	0.159 ±0.019	0.000 ±0.000	0.600 ±0.003	0.000 ±0.000
ECPCS-MC [73]	0.585 ±0.008	**0.690** ±0.005	0.000 ±0.000	0.020 ±0.007	0.120 ±0.008	0.940 ±0.002	0.000 ±0.000	0.591 ±0.012	0.604 ±0.011	0.000 ±0.000	0.308 ±0.103	0.038 ±0.021	0.821 ±0.039	0.000 ±0.000
CESHL [74]	0.334 ±0.105	0.448 ±0.132	0.000 ±0.000	0.002 ±0.007	0.006 ±0.006	0.641 ±0.182	0.000 ±0.000	0.208 ±0.057	0.028 ±0.083	0.068 ±0.217	0.110 ±0.314	0.100 ±0.316	0.915 ±0.041	0.100 ±0.316
PFREFF [75]	0.478 ±0.012	0.574 ±0.024	0.000 ±0.000	0.024 ±0.003	0.028 ±0.007	0.895 ±0.023	0.000 ±0.000	0.531 ±0.188	0.488 ±0.258	0.137 ±0.289	0.868 ±0.115	0.427 ±0.322	0.976 ±0.020	0.200 ±0.421
FCE-f	0.582 ±0.009	**0.690** ±0.004	0.000 ±0.000	0.023 ±0.001	0.134 ±0.006	0.954 ±0.003	0.000 ±0.000	0.604 ±0.047	0.591 ±0.028	0.000 ±0.000	0.858 ±0.110	0.327 ±0.113	0.968 ±0.012	0.000 ±0.000
FCE-a	0.581 ±0.023	0.619 ±0.015	0.232 ±0.089	0.923 ±0.052	0.426 ±0.091	0.987 ±0.008	0.383 ±0.135	0.581 ±0.002	0.532 ±0.006	0.196 ±0.003	0.979 ±0.003	0.660 ±0.005	0.995 ±0.000	0.362 ±0.004
FCE	**0.592** ±0.016	0.643 ±0.011	**0.269** ±0.016	**0.950** ±0.005	**0.600** ±0.027	**0.996** ±0.001	**0.423** ±0.028	0.598 ±0.055	0.542 ±0.034	**0.213** ±0.021	**0.985** ±0.001	**0.760** ±0.071	**0.997** ±0.002	**0.409** ±0.047

FCE-f represents the degenerated version of FCE without the fairness regularized term and FCE-a denotes the version that aligns the base results with the Hungary algorithm. The best and second best results are denoted in bold and underlined, respectively.

TABLE VII
EXPERIMENTAL RESULTS WITH DIFFERENT BASE CLUSTERING ALGORITHMS ON JAFFE AND YALE DATA SETS

Methods	JAFFE							Yale						
	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE
KM	0.951 ±0.034	0.948 ±0.032	0.025 ±0.075	0.856 ±0.126	0.652 ±0.155	0.996 ±0.005	0.033 ±0.100	0.495 ±0.033	0.558 ±0.023	0.006 ±0.022	0.037 ±0.136	0.234 ±0.126	0.969 ±0.020	0.013 ±0.054
BCE [69]	0.859 ±0.068	0.899 ±0.040	0.050 ±0.158	0.810 ±0.293	0.386 ±0.185	0.973 ±0.018	0.053 ±0.169	0.492 ±0.042	0.543 ±0.019	0.000 ±0.000	0.000 ±0.000	0.097 ±0.072	0.945 ±0.021	0.000 ±0.000
RCE [70]	0.978 ±0.005	0.971 ±0.004	0.000 ±0.000	0.909 ±0.029	0.688 ±0.067	0.998 ±0.001	0.000 ±0.000	**0.537** ±0.012	**0.596** ±0.007	0.000 ±0.000	0.000 ±0.000	0.226 ±0.030	0.977 ±0.003	0.000 ±0.000
LWGP [71]	0.978 ±0.005	0.971 ±0.004	0.000 ±0.000	0.909 ±0.029	0.688 ±0.067	0.998 ±0.001	0.000 ±0.000	0.500 ±0.025	0.561 ±0.021	0.000 ±0.000	0.000 ±0.000	0.120 ±0.071	0.956 ±0.015	0.000 ±0.000
LWEA [71]	0.962 ±0.006	0.968 ±0.006	0.000 ±0.000	0.902 ±0.039	0.683 ±0.077	0.997 ±0.001	0.000 ±0.000	0.513 ±0.024	0.558 ±0.019	0.000 ±0.000	0.000 ±0.000	0.105 ±0.067	0.955 ±0.013	0.000 ±0.000
DREC [72]	0.980 ±0.004	0.972 ±0.005	0.000 ±0.000	0.911 ±0.029	0.701 ±0.058	0.998 ±0.001	0.000 ±0.000	0.528 ±0.020	0.590 ±0.011	0.000 ±0.000	0.000 ±0.000	0.207 ±0.062	0.971 ±0.007	0.000 ±0.000
RSEC [26]	0.981 ±0.004	0.974 ±0.005	0.000 ±0.000	0.919 ±0.008	0.707 ±0.040	0.998 ±0.001	0.000 ±0.000	0.523 ±0.021	0.586 ±0.014	0.000 ±0.000	0.000 ±0.000	0.237 ±0.114	0.975 ±0.007	0.000 ±0.000
TRCE [32]	0.965 ±0.011	0.970 ±0.008	0.000 ±0.000	0.893 ±0.021	0.660 ±0.053	0.997 ±0.001	0.000 ±0.000	0.504 ±0.018	0.568 ±0.009	0.000 ±0.000	0.000 ±0.000	0.136 ±0.055	0.964 ±0.005	0.000 ±0.000
ECPCS-HC [73]	0.979 ±0.005	0.970 ±0.004	0.000 ±0.000	0.911 ±0.029	0.701 ±0.058	0.058 ±0.001	0.000 ±0.000	0.487 ±0.034	0.552 ±0.029	0.000 ±0.000	0.000 ±0.000	0.053 ±0.031	0.934 ±0.019	0.000 ±0.000
ECPCS-MC [73]	0.970 ±0.031	0.966 ±0.019	0.000 ±0.000	0.918 ±0.008	0.677 ±0.099	0.996 ±0.005	0.000 ±0.000	0.496 ±0.028	0.567 ±0.019	0.010 ±0.031	0.058 ±0.183	0.205 ±0.064	0.973 ±0.009	0.032 ±0.101
CESHL [74]	0.958 ±0.009	0.950 ±0.008	0.000 ±0.000	0.909 ±0.029	0.688 ±0.067	0.998 ±0.001	0.000 ±0.000	0.529 ±0.021	0.583 ±0.020	0.000 ±0.000	0.000 ±0.000	0.190 ±0.088	0.967 ±0.018	0.000 ±0.000
PFREFF [75]	0.862 ±0.032	0.888 ±0.048	0.000 ±0.000	0.828 ±0.045	0.376 ±0.111	0.978 ±0.006	0.000 ±0.000	0.501 ±0.013	0.559 ±0.007	0.000 ±0.000	0.000 ±0.000	0.162 ±0.049	0.968 ±0.005	0.000 ±0.000
FCE-f	0.970 ±0.034	0.966 ±0.020	0.000 ±0.000	0.828 ±0.291	0.650 ±0.220	0.992 ±0.017	0.000 ±0.000	0.530 ±0.013	0.590 ±0.011	0.000 ±0.000	0.000 ±0.000	0.249 ±0.113	0.979 ±0.008	0.000 ±0.000
FCE-a	0.977 ±0.006	0.968 ±0.007	**0.500** ±0.000	0.989 ±0.000	**0.869** ±0.002	**0.999** ±0.003	0.633 ±0.010	0.506 ±0.011	0.540 ±0.012	0.127 ±0.050	0.654 ±0.142	0.616 ±0.114	0.995 ±0.002	0.465 ±0.093
FCE	**0.990** ±0.007	**0.983** ±0.008	**0.500** ±0.000	**0.999** ±0.000	**0.869** ±0.001	**0.999** ±0.001	**0.645** ±0.001	0.517 ±0.004	0.544 ±0.005	**0.161** ±0.041	**0.757** ±0.107	**0.738** ±0.039	**0.998** ±0.001	**0.508** ±0.097

FCE-f represents the degenerated version of FCE without the fairness regularized term and FCE-a denotes the version that aligns the base results with the Hungary algorithm. The best and second best results are denoted in bold and underlined, respectively.

TABLE VIII
COMPARISON WITH FAIR CLUSTERING METHODS

Methods	D&S								HAR							
	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	avg rank	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	avg rank
SpFC [15]	0.430(4)	0.551(4)	0.000(5)	0.000(8)	0.072(6)	0.891(6)	0.000(5)	5.4	0.538(8)	0.587(4)	0.000(6)	0.000(9)	0.006(9)	0.621(8)	0.000(6)	7.1
VFC [46]	0.539(3)	0.617(3)	0.186(4)	0.923(3)	0.236(4)	0.981(3)	0.133(4)	3.4	0.600(6)	**0.654**(1)	0.200(2)	0.983(3)	0.156(7)	0.913(7)	0.099(4)	4.3
FFC [76]	-	-	-	-	-	-	-	-	0.602(5)	0.490(8)	0.007(5)	0.955(6)	0.468(3)	0.981(3)	0.015(5)	5.0
KFC [77]	0.154(8)	0.180(8)	0.000(5)	0.666(5)	0.007(8)	0.392(8)	0.000(5)	6.7	0.252(9)	0.133(9)	0.000(6)	**0.985**(1)	0.023(8)	0.470(9)	0.000(6)	6.8
SFD [19]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CFC [52]	0.187(7)	0.413(7)	0.251(2)	0.947(2)	0.463(2)	0.961(4)	0.280(3)	3.9	0.643(2)	0.531(7)	0.166(3)	0.975(5)	0.275(6)	0.915(6)	0.162(3)	4.6
FCE-f(ours)	0.415(6)	0.541(5)	0.000(5)	0.030(6)	0.036(7)	0.854(7)	0.000 (5)	5.9	0.633(3)	0.600(2)	0.000(6)	0.858(7)	0.393(4)	0.972(4)	0.000(6)	4.6
FCE_dbase-f(ours)	0.582(2)	**0.690**(1)	0.000(5)	0.023(7)	0.134(5)	0.954(5)	0.000(5)	4.3	0.604(4)	0.591(3)	0.000(6)	0.858(7)	0.327(5)	0.968(5)	0.000(6)	5.1
FCE(ours)	0.425(5)	0.482(6)	0.222(3)	0.920(4)	0.387(3)	0.985(2)	0.374(2)	3.6	**0.648**(1)	0.557(5)	0.149(4)	0.977(4)	0.731(2)	**0.997**(1)	0.331(2)	2.7
FCE_dbase(ours)	**0.592**(1)	0.643(2)	**0.269**(1)	**0.950**(1)	**0.600**(1)	**0.996**(1)	**0.423**(1)	**1.1**	0.598(7)	0.542(6)	**0.213**(1)	**0.985**(1)	**0.760**(1)	**0.997**(1)	**0.409**(1)	**2.6**

Methods	MNIST-USPS								Reverse MNIST							
	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	avg rank	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	avg rank
SpFC [15]	0.371(4)	**0.447**(1)	0.000(8)	0.000(8)	0.135(6)	0.912(6)	0.000(8)	5.9	0.350(5)	0.421(2)	0.000(6)	0.000(6)	0.120(7)	0.985(4)	0.000(6)	5.1
VFC [46]	0.360(6)	0.306(8)	0.142(6)	0.544(6)	0.067(8)	0.846(8)	0.085(4)	6.6	0.329(6)	0.306(6)	0.015(5)	0.114(5)	0.203(6)	0.975(7)	0.039(5)	5.7
FFC [76]	**0.437**(1)	0.412(2)	0.217(5)	0.684(5)	0.605(3)	0.993(3)	0.370(2)	3.0	0.309(8)	0.217(8)	0.114(4)	0.477(4)	0.498(3)	0.989(3)	0.135(4)	4.9
KFC [77]	0.145(9)	0.014(9)	**0.500**(1)	**0.920**(1)	0.002(10)	0.055(10)	0.010(7)	6.7	0.172(9)	0.045(9)	0.000(6)	0.000(6)	0.001(10)	0.449(10)	0.000(6)	8
SFD [19]	0.139(10)	0.012(10)	0.230(4)	0.697(4)	0.008(9)	0.524(9)	0.033(6)	7.4	0.122(10)	0.006(10)	**0.709**(1)	**0.979**(1)	0.029(9)	0.558(9)	0.140(3)	6.1
CFC [52]	0.328(8)	0.386(5)	0.231(3)	0.702(3)	0.115(7)	0.904(7)	0.084(5)	5.4	0.377(2)	0.317(5)	0.000(6)	0.000(6)	0.085(8)	0.758(8)	0.000(6)	5.9
FCE-f(ours)	0.395(3)	0.406(4)	0.000(8)	0.000(8)	0.391(4)	0.985(4)	0.000(8)	5.6	0.352(4)	0.368(3)	0.000(6)	0.000(6)	0.340(4)	0.977(5)	0.000(6)	4.9
FCE_dbase-f(ours)	0.368(5)	0.407(3)	0.000(8)	0.000(8)	0.185(5)	0.945(5)	0.000(8)	6.0	**0.400**(1)	**0.469**(1)	0.000(6)	0.000(6)	0.324(5)	0.976(6)	0.000(6)	4.4
FCE(ours)	0.419(2)	0.385(6)	0.105(7)	0.445(7)	0.703(2)	0.997(2)	0.190(3)	4.1	0.328(7)	0.266(7)	0.339(2)	0.814(2)	**0.774**(1)	**0.999**(1)	0.174(2)	3.1
FCE_dbase(ours)	0.360(6)	0.331(7)	0.241(2)	0.707(2)	**0.749**(1)	**0.998**(1)	**0.403**(1)	**2.9**	0.359(3)	0.356(4)	0.311(3)	0.790(3)	0.754(2)	0.998(2)	**0.468**(1)	**2.6**

Methods	JAFFE								Yale							
	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	avg rank	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	avg rank
SpFC [15]	0.952(5)	0.941(5)	0.000(7)	0.909(6)	0.826(3)	**0.999**(1)	0.000(7)	4.9	0.393(8)	0.471(9)	0.000(6)	0.000(6)	0.121(8)	0.920(8)	0.000(6)	7.3
VFC [46]	0.981(3)	0.969(2)	0.400(4)	0.983(4)	0.852(2)	**0.999**(1)	0.625(3)	2.7	0.406(7)	0.489(7)	0.125(4)	0.664(4)	0.272(3)	0.965(6)	0.400(3)	4.9
FFC [76]	0.901(7)	0.918(6)	0.250(5)	0.924(5)	0.307(8)	0.983(7)	0.322(5)	6.1	0.472(3)	0.527(3)	0.125(4)	0.664(4)	0.153(7)	0.968(5)	0.116(5)	4.4
KFC [77]	0.319(9)	0.375(9)	0.250(5)	0.898(7)	0.075(9)	0.746(9)	0.201(6)	7.7	0.260(10)	0.371(10)	0.000(6)	0.000(6)	0.027(10)	0.795(10)	0.000(6)	8.2
SFD [19]	-	-	-	-	-	-	-	-	0.315(9)	0.512(4)	0.000(6)	0.000(6)	0.061(9)	0.816(9)	0.000(6)	7.0
CFC [52]	0.988(2)	0.951(4)	0.463(3)	0.985(3)	0.792(5)	**0.999**(1)	0.563(4)	3.1	0.466(4)	0.492(6)	0.142(3)	0.718(3)	0.209(6)	0.942(7)	0.250(4)	4.7
FCE-f(ours)	0.858(8)	0.872(8)	0.000(7)	0.449(9)	0.318(7)	0.971(8)	0.000(7)	7.7	0.444(5)	0.509(5)	0.000(6)	0.000(6)	0.241(5)	0.982(3)	0.000(6)	5.1
FCE_dbase-f(ours)	0.970(4)	0.966(3)	0.000(7)	0.828(8)	0.650(6)	0.992(6)	0.000(7)	5.9	**0.530**(1)	**0.590**(1)	0.000(6)	0.000(6)	0.249(4)	0.979(4)	0.000(6)	3.7
FCE(ours)	0.913(6)	0.896(7)	0.483(2)	0.986(2)	0.810(4)	**0.999**(1)	0.639(2)	4.0	0.426(6)	0.476(8)	**0.194**(1)	**0.838**(1)	**0.806**(1)	**0.999**(1)	**0.526**(1)	2.7
FCE_dbase(ours)	**0.990**(1)	**0.983**(1)	**0.500**(1)	**0.999**(1)	**0.869**(1)	**0.999**(1)	**0.645**(1)	**1.0**	0.517(2)	0.544(2)	0.161(2)	0.757(2)	0.738(2)	0.998(2)	0.508(2)	**2.0**

FCE is the original version of our method. FCE_dbase represents the version that uses different base clustering methods as input. FCE-f and FCE_dbase-f denote the versions without the fairness regularized term of FCE and FCE_dbase, respectively. The best and second best results are denoted in bold and underlined, respectively. avg rank denotes the average rank over all metrics of each method.

result in acceptable time on the large data set D&S. SFD can only handle data sets with two protected groups, and thus they only have results on HAR, MNIST-USPS, and Reverse MNIST. From Table V, we find that our method can outperform these fair clustering methods on cluster capacity equality (i.e., CCE, and NE) on all data sets. Regarding fairness, our method can also often achieve comparable or better performance on many data sets. Notice that KFC and SGD achieve a very high performance w.r.t. Bal and MNCE on MNIST-USPS and Reverse MNIST, respectively. However, we observe that they put most data into one cluster. For example, on MNIST-USPS, the numbers of data in each cluster of KFC are 3746, 7, 12, 6, 16, 6, and 7, respectively. On Reverse MNIST, the numbers of data in each cluster of SFD are 2242, 54, 244, 128, 282, 152, 184, 254, 210, and 250, respectively. This also demonstrates our motivation to consider cluster capacity equality. When comparing w.r.t. ACC and NMI, our FCE sometimes performs worse than other methods. Notice that, these fair clustering methods need the original features of data whereas ours only uses the base clustering results from k-means without the original features. Poor base clustering results may limit the performance of our FCE. To see this, we also report the results of FCE_dbase, which is our FCE with different base clustering algorithms as introduced in Section IV-C. We can see that better base results can improve the ACC and NMI of our FCE, and FCE_dbase can even outperform other fair clustering methods on some data sets. We also report the versions without the fairness regularized term of FCE and FCE_dbase, denoted as FCE-f and FCE_dbase-f. The results show that with the fairness regularized term, the overall performance of our method can be further improved.

E. Efficiency Results

Fig. 4 shows the convergence curves of our method on all data sets. It can be seen that our method can often converge very fast (i.e., often converges within 10 iterations).

Fig. 5 shows the running time of our method compared with other clustering ensemble methods on all data sets. Since some methods are very time-consuming, we report the logarithm of the time in seconds for better comparison. From Fig. 5, we can see that our method is comparable with the mainstream clustering ensemble methods. Ours is even faster than some state-of-the-art methods, such as RSEC and CESHL. Despite this, since the time complexity of our method is still square in the number of instances, in the future, we will study how to further speed up this method.

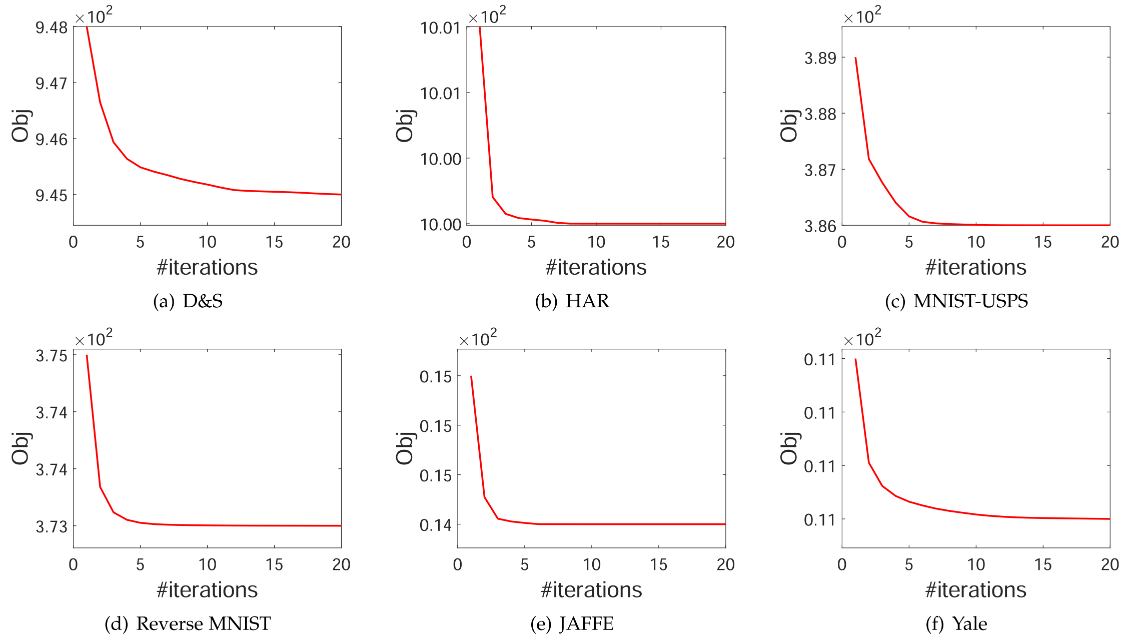


Fig. 4. Convergence curves of FCE on all data sets.

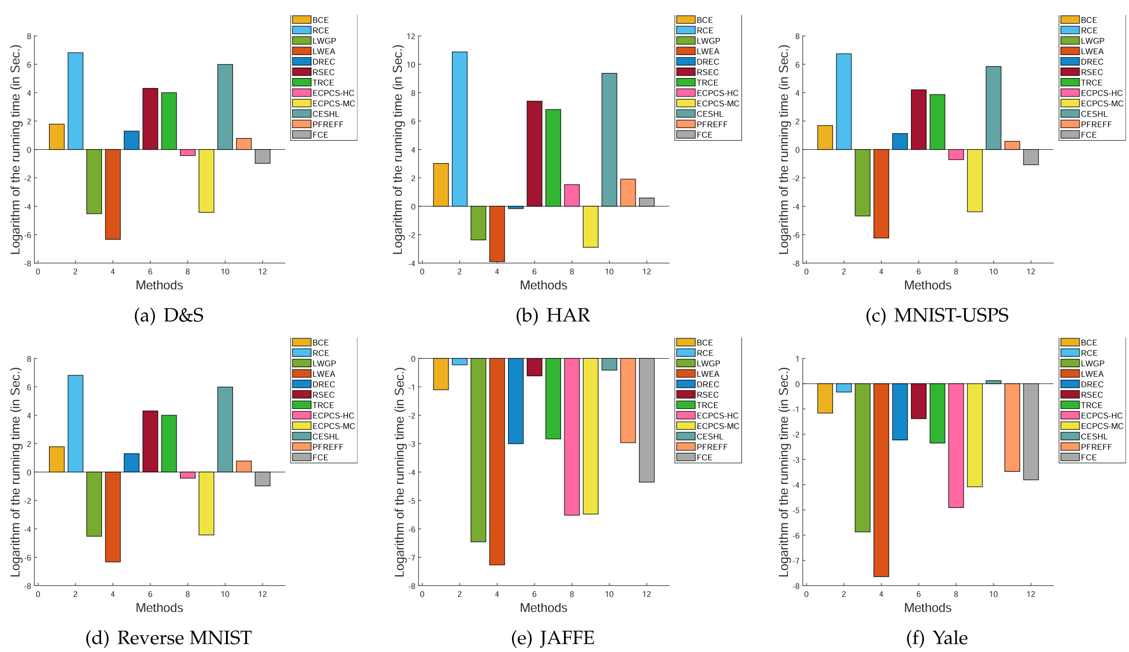
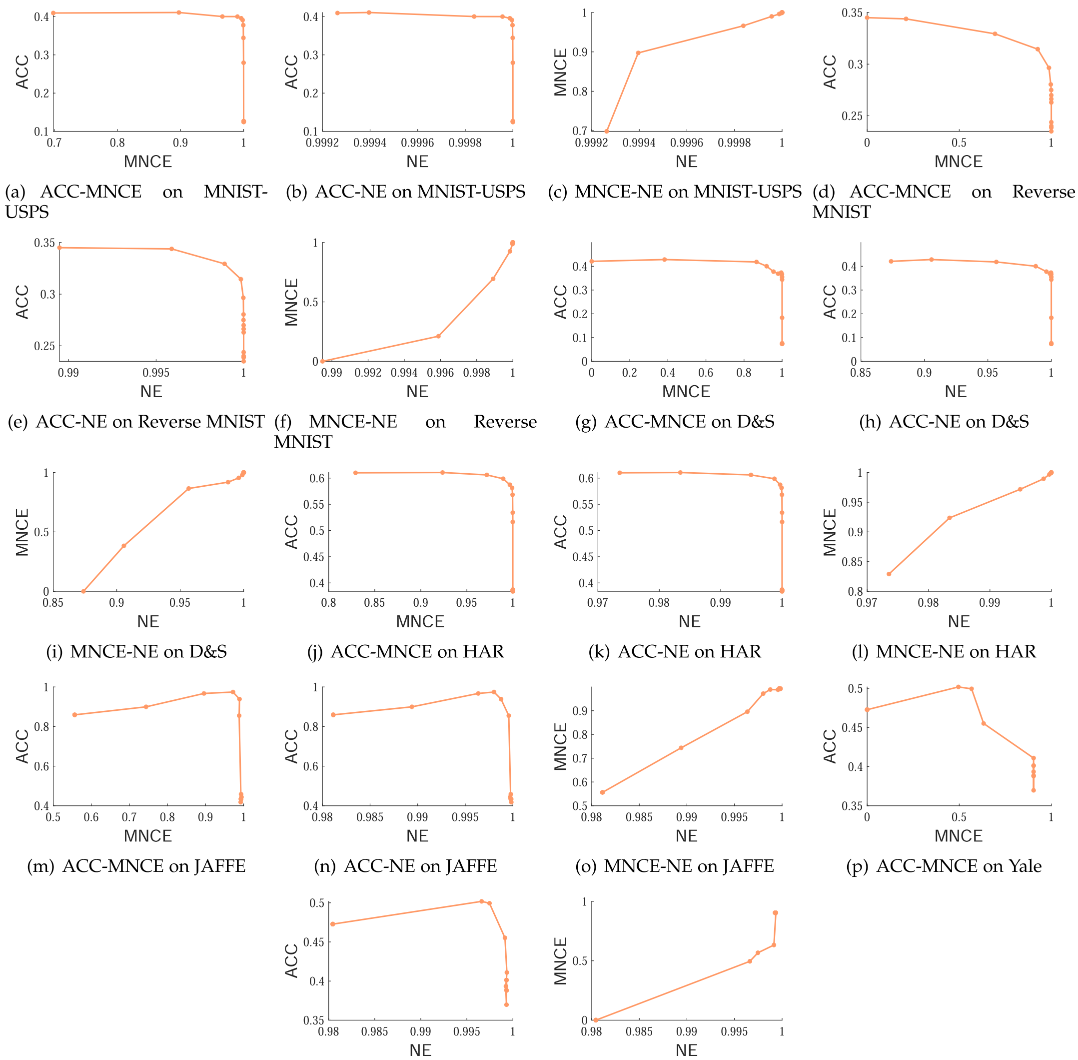


Fig. 5. Running time of all methods on all data sets.

F. Trade-Off Between Accuracy, Fairness, and Cluster Capacity Equality

In this subsection, we show the trade-off curves of clustering accuracy (e.g. ACC), fairness (e.g. MNCE), and cluster capacity equality (e.g. NE) by tuning the hyper-parameter λ_2 . The trade-off curves are shown in Fig. 6. We can see that, there exists a trade-off between accuracy and fairness and cluster capacity equality. Too high fairness or cluster capacity equality may lead to a decrease in accuracy. Despite this, our method can still obtain a good trade-off on most data sets, because the inflection points often appear in the upper right. Moreover, from the trade-off curves of fairness and cluster capacity equality, we find that our regularized term can indeed improve fairness and cluster capacity equality simultaneously, demonstrating the effectiveness of the regularized term.



(a) ACC-MNCE on MNIST-USPS (b) ACC-NE on MNIST-USPS (c) MNCE-NE on MNIST-USPS (d) ACC-MNCE on Reverse MNIST

(e) ACC-NE on Reverse MNIST (f) MNCE-NE on Reverse MNIST (g) ACC-MNCE on D&S (h) ACC-NE on D&S

(i) MNCE-NE on D&S (j) ACC-MNCE on HAR (k) ACC-NE on HAR (l) MNCE-NE on HAR

(m) ACC-MNCE on JAFFE (n) ACC-NE on JAFFE (o) MNCE-NE on JAFFE (p) ACC-MNCE on Yale

(q) ACC-NE on Yale (r) MNCE-NE on Yale

Fig. 6. Tade-off curves between the accuracy, fairness, and the cluster capacity equality.

TABLE IX
EXPERIMENTAL RESULTS OF FCE AND FCE_UNKNOW_K ON ALL DATA SETS, WHERE FCE_UNKNOWN_K DENOTES THE VERSION THAT AUTOMATICALLY DECIDING THE NUMBER OF CLUSTERS

Methods	MNIST-USPS							Reverse MNIST						
	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE
FCE	0.419 ±0.019	0.385 ±0.013	0.105 ±0.037	0.445 ±0.094	0.703 ±0.084	0.997 ±0.002	0.190 ±0.052	0.328 ±0.011	0.266 ±0.014	0.339 ±0.030	0.814 ±0.027	0.774 ±0.021	0.999 ±0.000	0.174 ±0.051
FCE_unknow_k	0.406 ±0.014	0.364 ±0.012	0.220 ±0.058	0.671 ±0.083	0.726 ±0.055	0.998 ±0.001	0.353 ±0.081	0.321 ±0.019	0.290 ±0.018	0.145 ±0.126	0.462 ±0.259	0.746 ±0.035	0.998 ±0.001	0.236 ±0.177

Methods	D&S							HAR						
	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE
FCE	0.425 ±0.019	0.482 ±0.016	0.222 ±0.038	0.920 ±0.034	0.387 ±0.049	0.985 ±0.002	0.374 ±0.048	0.648 ±0.026	0.557 ±0.024	0.149 ±0.148	0.977 ±0.002	0.731 ±0.045	0.997 ±0.000	0.331 ±0.050
FCE_unknow_k	0.406 ±0.031	0.465 ±0.029	0.161 ±0.134	0.796 ±0.163	0.317 ±0.173	0.975 ±0.015	0.254 ±0.156	0.649 ±0.018	0.558 ±0.022	0.150 ±0.021	0.977 ±0.002	0.732 ±0.039	0.997 ±0.001	0.337 ±0.050

Methods	JAFFE							Yale						
	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE	ACC	NMI	Bal	MNCE	CCE	NE	f_CCE
FCE	0.913 ±0.086	0.896 ±0.077	0.483 ±0.032	0.986 ±0.005	0.810 ±0.025	0.999 ±0.001	0.639 ±0.013	0.426 ±0.025	0.476 ±0.017	0.194 ±0.048	0.838 ±0.127	0.806 ±0.070	0.999 ±0.004	0.526 ±0.094
FCE_unknow_k	0.888 ±0.054	0.870 ±0.054	0.453 ±0.122	0.977 ±0.015	0.772 ±0.038	0.998 ±0.001	0.561 ±0.134	0.412 ±0.018	0.476 ±0.019	0.135 ±0.061	0.653 ±0.245	0.742 ±0.080	0.998 ±0.001	0.487 ±0.185

G. Selection of Number of Clusters

Like many other mainstream clustering methods and clustering ensemble methods, our method also assumes that the number of clusters is known. If the number of clusters is unknown, we can use some internal indices to guide the decision of the number of clusters. In our method, we tried to use the Silhouette Coefficient Index as the internal index to search the number of clusters. In more detail, we concatenate the indicator matrices of the base results as a representation of data to compute the Silhouette Coefficient Index. Then, we run our algorithm to automatically search for the optimal number of clusters by selecting the one that maximizes the Silhouette Coefficient Index. The results of this strategy are shown in Table IX, which is denoted as FCE_unknown_k. We can see that, the results are very close to the original version which uses the ground truth of the number of clusters. It well demonstrates the effectiveness of our strategy for deciding the number of clusters for FCE.

V. Conclusion

This paper proposed a new notion of fair clustering ensemble. When observing the limitation of the traditional definition of fairness to handle the cluster capacity, we designed a simple yet effective regularized term to simultaneously achieve fairness and cluster capacity equality. Then, we plugged this carefully designed regularized term into a clustering ensemble framework, leading to our novel Fair Clustering Ensemble method. Extensive experiments on benchmark data sets by comparing with state-of-the-art clustering ensemble methods shew our superiority in fairness and cluster capacity equality.

References

[1] W.-L. Chang and T.-H. Lin, "A cluster-based approach for automatic social network construction," in *Proc. IEEE 2nd Int. Conf. Social Comput.*, 2010, pp. 601–606.

[2] A. T. Murray and T. H. Grubesic, *Exploring Spatial Patterns of Crime Using Non-hierarchical Cluster Analysis*. Dordrecht, Netherlands: Springer, 2013, pp. 105–124, doi: 10.1007/978-94-007-4997-9_5.

[3] F. Wang, X. Wang, and T. Li, "Generalized cluster aggregation," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, Pasadena, CA, USA, 2009, pp. 1279–1284. [Online]. Available: http://ijcai.org/Proceedings/09/Papers/215.pdf

[4] A. Strehl and J. Ghosh, "Cluster ensembles — A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 583–617, 2003.

[5] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "From ensemble clustering to multi-view clustering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2843–2849.

[6] F. Li, Y. Qian, J. Wang, C. Dang, and L. Jing, "Clustering ensemble based on sample's stability," *Artif. Intell.*, vol. 273, pp. 37–55, 2019.

[7] P. Zhou, L. Du, and X. Li, "Self-paced consensus clustering with bipartite graph," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, 2020, pp. 2133–2139.

[8] P. Zhou, B. Sun, X. Liu, L. Du, and X. Li, "Active clustering ensemble with self-paced learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 12186–12200, Sep. 2024.

[9] X. Liu et al., "One pass late fusion multi-view clustering," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 6850–6859.

[10] P. Zhou, X. Liu, L. Du, and X. Li, "Self-paced adaptive bipartite graph learning for consensus clustering," *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 5, pp. 62:1–62:35, 2023.

[11] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–35, 2021.

[12] F. Chierichetti, R. S. KumarLattanzi, and S. Vassilvitskii, "Fair clustering through fairlets," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5029–5037.

[13] H. Zhang and I. Davidson, "Deep fair discriminative clustering," 2021, *arXiv:2105.14146*.

[14] M. Schmidt, C. Schwiegelshohn, and C. Sohler, "Fair coresets and streaming algorithms for fair K-means clustering," Dec. 2018, *arXiv: 1812.10854*.

[15] M. Kleindessner, S. Samadi, P. Awasthi, and J. Morgenstern, "Guarantees for spectral clustering with fairness constraints," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3458–3467.

[16] P. Zeng, Y. Li, P. Hu, D. Peng, J. Lv, and X. Peng, "Deep fair clustering via maximizing and minimizing mutual information," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 23986–23995.

[17] S. Bera, D. Chakrabarty, N. Flores, and M. Negahbani, "Fair algorithms for clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4955–4966.

[18] P. Li, H. Zhao, and H. Liu, "Deep fair clustering for visual learning," in *Proc. 2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9067–9076.

[19] A. Backurs, P. Indyk, K. Onak, B. Schieber, A. Vakilian, and T. Wagner, "Scalable fair clustering," in *Proc. 36th Int. Conf. Mach. Learn.*, Long Beach, CA, USA, 2019, pp. 405–413.

[20] H. Liu, J. Han, F. Nie, and X. Li, "Balanced clustering with least square regression," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2231–2237.

[21] X. Chen, B. Fain, L. Lyu, and K. Munagala, "Proportionally fair clustering," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1032–1041.

[22] N. Iam-On, T. Boongoen, S. Garrett, and C. Price, "A link-based approach to the cluster ensemble problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2396–2409, Dec. 2011, doi: 10.1109/tpami.2011.84.

[23] T. Wang, "CA-tree: A hierarchical structure for efficient and scalable coassociation-based cluster ensembles," *IEEE Trans. Syst. Man, Cybern. B. Cybern.*, vol. 41, no. 3, pp. 686–698, Jun. 2011, doi: 10.1109/tsmcb.2010.2086059.

[24] A. Lourenço, S. RotaBulò, N.A. L. N. Rebagliati, M. A. T. FredFigueiredo, and M. Pelillo, "Probabilistic consensus clustering using evidence accumulation," *Mach. Learn.*, vol. 98, no. 1–2, pp. 331–357, Jan. 2015, doi: 10.1007/s10994-013-5339-6.

[25] X. Liu et al., "Late fusion incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2410–2423, Oct. 2019.

[26] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "Robust spectral ensemble clustering via rank minimization," *ACM Trans. Knowl. Discov. From Data*, vol. 13, no. 1, pp. 1–25, 2019.

[27] Y. Jia, S. Tao, R. Wang, and Y. Wang, "Ensemble clustering via co-association matrix self-enhancement," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 11168–11179, Aug. 2024.

[28] D. Huang, C. Wang, J. Wu, J. Lai, and C. Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1212–1226, Jun. 2020.

[29] Z. Yu, L. Li, J. Liu, J. Zhang, and G. Han, "Adaptive noise immune cluster ensemble using affinity propagation," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 12, pp. 3176–3189, Dec. 2015, doi: 10.1109/tkde.2015.2453162.

[30] D. Huang, J.-H. Lai, and C.-D. Wang, "Combining multiple clusterings via crowd agreement estimation and multi-granularity link analysis," *Neurocomputing*, vol. 170, pp. 240–250, Dec. 2015, doi: 10.1016/j.neucom.2014.05.094.

[31] H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu, "Spectral ensemble clustering," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 715–724.

[32] P. Zhou, L. Du, Y.-D. Shen, and X. Li, "Tri-level robust clustering ensemble with multiple graph learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11125–11133.

[33] M.-S. Chen, J.-Q. Lin, C.-D. Wang, W.-D. Xi, and D. Huang, "On regularizing multiple clusterings for ensemble clustering by graph tensor learning," in *Proc. 31st ACM Int. Conf. Multimedia*, New York, NY, USA: Association for Computing Machinery, 2023, pp. 3069–3077, doi: 10.1145/3581783.3612313.

[34] P. Zhou, L. Du, X. Liu, Y. Shen, M. Fan, and X. Li, "Self-paced clustering ensemble," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1497–1511, Apr. 2021.

[35] P. Zhou, B. Hu, D. Yan, and L. Du, "Clustering ensemble via diffusion on adaptive multiplex," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 4, pp. 1463–1474, Apr. 2024.

[36] D. Cristofor and D. Simovici, "Finding median partitions using information-theoretical-based genetic algorithms," *J. Universal Comput. Sci.*, vol. 8, pp. 153–172, Jan. 2002.

[37] A. Topchy, A. Jain, and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1866–1881, Dec. 2005, doi: 10.1109/tpami.2005.237.

[38] L. Franek and X. Jiang, "Ensemble clustering by means of clustering embedding in vector spaces," *Pattern Recognit.*, vol. 47, pp. 833–842, Feb. 2014, doi: 10.1016/j.patcog.2013.08.019.

[39] L. Bai, J. Liang, H. Du, and Y. Guo, "An information-theoretical framework for cluster ensemble," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 8, pp. 1464–1477, Aug. 2019.

[40] D. Huang, J.-H. Lai, and C.-D. Wang, "Robust ensemble clustering using probability trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 5, pp. 1312–1326, May 2016, doi: 10.1109/tkde.2015.2503753.

[41] P. Zhou et al., "Partial clustering ensemble," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 5, pp. 2096–2109, May 2024.

[42] Z.-H. Zhou and W. Tang, "Clusterer ensemble," *Knowl.-Based Syst.*, vol. 19, no. 1, pp. 77–83, 2006.

[43] X. Zheng, S. Zhu, J. Gao, and H. Mamitsuka, "Instance-wise weighted nonnegative matrix factorization for aggregating partitions with locally reliable clusters," in *Proc. Int. Conf. Artif. Intell.*, 2015, pp. 4091–4097.

[44] A. Chhabra, K. Masalkovaite, and P. Mohapatra, "An overview of fairness in clustering," *IEEE Access*, vol. 9, pp. 130698–130720, 2021.

[45] C. Rösner and M. Schmidt, "Privacy preserving clustering with constraints," in *Proc. Int. Colloq. Automata, Lang. Program.*, 2018, pp. 96:1–96:14.

[46] I. M. Ziko, J. Yuan, E. Granger, and I. Ben Ayed, "Variational fair clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 11202–11209, doi: 10.1609/aaai.v35i12.17336.

[47] M. Abbasi, A. Bhaskara, and S. Venkatasubramanian, "Fair clustering via equitable group representations," in *Proc. ACM Conf. Fairness Accountability, Transparency*, 2021, pp. 504–514, doi: 10.1145/3442188.3445913.

[48] P. Li and H. Liu, "Achieving fairness at no utility cost via data reweighing with influence," in *Proc. Int. Con. Mach. Learn.*, 2022, pp. 12917–12930.

[49] M. Ghadiri, S. Samadi, and S. Vempala, "Socially fair K-means clustering," in *Proc. ACM Conf. Fairness Accountability Transparency*, 2021, pp. 438–448, doi: 10.1145/3442188.3445906.

[50] A. Chhabra, A. Singla, and P. Mohapatra, "Fair clustering using antidote data," in *Algorithmic Fairness Through the Lens of Causality and Robustness Workshop*. Seattle, WA, USA: PMLR, 2021, pp. 19–39. [Online]. Available: https://proceedings.mlr.press/v171/chhabra22a.html

[51] B. Wang and I. Davidson, "Towards fair deep clustering with multi-state protected variables," 2019, *arXiv: 1901.10053*.

[52] A. Chhabra, P. Li, P. Mohapatra, and H. Liu, "Robust fair clustering: A novel fairness attack and defense framework," in *Proc. 11th Int. Conf. Learn. Representations*, Kigali, Rwanda, 2023.

[53] P. S. Bradley, K. P. Bennett, and A. Demiriz, *Constrained K-Means Clustering*. Redmond, WA, USA: Microsoft Research, 2000.

[54] M. I. Malinen and P. Fränti, "Balanced K-means for clustering," in *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop*. Berlin, Germany: Springer, 2014, pp. 32–41.

[55] L. R. Costa, D. Aloise, and N. Mladenović, "Less is more: Basic variable neighborhood search heuristic for balanced minimum sum-of-squares clustering," *Inf. Sci.*, vol. 415, pp. 247–253, 2017.

[56] A. Banerjee and J. Ghosh, "On scaling up balanced clustering algorithms," in *Proc. 2002 SIAM Int. Conf. Data Mining*, 2002, pp. 333–349.

[57] A. Banerjee and J. Ghosh, "Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 702–719, May 2004.

[58] H. Liu, Z. Huang, Q. Chen, M. Li, Y. Fu, and L. Zhang, "Fast clustering with flexible balance constraints," in *Proc. 2018 IEEE Int. Conf. Big Data*, 2018, pp. 743–750.

[59] P. Zhou, J. Chen, M. Fan, L. Du, Y. Shen, and X. Li, "Unsupervised feature selection for balanced clustering," *Knowl. Based Syst.*, vol. 193, 2020, Art. no. 105417.

[60] P. Zhou, J. Chen, L. Du, and X. Li, "Balanced spectral feature selection," *IEEE Trans. Cybern.*, vol. 53, no. 7, pp. 4232–4244, Jul. 2023.

[61] H. Liu, J. Han, F. Nie, and X. Li, "Balanced clustering with least square regression," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, 2017, pp. 2231–2237.

[62] Z. Li, F. Nie, X. Chang, Z. Ma, and Y. Yang, "Balanced clustering via exclusive lasso: A pragmatic approach," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3596–3603.

[63] S. X. Yu and J. Shi, "Multiclass spectral clustering," in*Proc. 9th IEEE Int. Conf. Comput. Vis.*, Nice, France, 2003, pp. 313–319.

[64] F. Nie, J. Xue, W. Yu, and X. Li, "Fast clustering with anchor guidance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 4, pp. 1898–1912, Apr. 2023.

[65] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognit.*, vol. 43, pp. 3605–3620, Oct. 2010, doi: 10.1016/j.patcog.2010.04.019.

[66] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. Eur. Symp. Artif. Neural Netw.*, 2013, pp. 437–442.

[67] M. J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, and J. Budynek, "The Japanese female facial expression (Jaffe) database," in *Proc. 3rd Int. Conf. Autom. Face Gesture Recognit.*, 1998, pp. 14–16.

[68] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–7.

[69] H. Wang, H. Shan, and A. Banerjee, "Bayesian cluster ensembles," *Statist. Anal. Data Mining: ASA Data Sci. J.*, vol. 4, no. 1, pp. 54–70, 2011.

[70] P. Zhou, L. Du, H. Wang, L. Shi, and Y.-D. Shen, "Learning a robust consensus matrix for clustering ensemble via Kullback-Leibler divergence minimization," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2015, pp. 4112–4118.

[71] D. Huang, C.-D. Wang, and J.-H. Lai, "Locally weighted ensemble clustering," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1460–1473, May 2018.

[72] J. Zhou, H. Zheng, and L. Pan, "Ensemble clustering based on dense representation," *Neurocomputing*, vol. 357, pp. 66–76, 2019.

[73] D. Huang, C.-D. Wang, H. Peng, J. Lai, and C.-K. Kwoh, "Enhanced ensemble clustering via fast propagation of cluster-wise similarities," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 1, pp. 508–520, Jan. 2021.

[74] P. Zhou, X. Wang, L. Du, and X. Li, "Clustering ensemble via structured hypergraph learning," *Inf. Fusion*, vol. 78, pp. 171–179, 2022.

[75] Z. Shi, L. Chen, W. Ding, C. Zhang, and Y. Wang, "Parameter-free robust ensemble framework of fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 12, pp. 4205–4219, Dec. 2023, 2023.

[76] R. Pan and C. Zhong, "Fairness first clustering: A multi-stage approach for mitigating bias," *Electronics*, vol. 12, Jul. 2023, Art. no. 2969, doi: 10.3390/electronics12132969.

[77] E. Harb and H. S. Lam, "KFC: A scalable approximation algorithm for K-center fair clustering," in in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/a6d259bfbfa2062843ef543e21d7ec8e-Abstract.html

Peng Zhou received the BE degree in computer science and technology from the University of Science and Technology of China, in 2011, and PhD degree in computer science from the Institute of Software, Chinese Academy of Sciences, in 2017. He is currently an associate professor with the School of Computer Science and Technology, Anhui University. He has published more than 60 papers in highly regarded conferences and journals, including *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Cybernetics*, *ACM Transactions on Knowledge Discovery from Data*, NeurIPS, IJCAI, AAAI, MM, etc. His research interests include machine learning and data mining. More publications and codes can be found in his homepage: https://doctor-nobody.github.io/.

Rongwen Li received the BE degree from Sichuan Agricultural University, in 2022. He is currently working toward the MS degree with the School of Computer Science and Technology, Anhui University. His research interests include machine learning and data mining.

Zhaolong Ling received the PhD degree from the School of Computer and Information, Hefei University of Technology, China, in 2020. He is a lecturer with the School of Computer Science and Technology, Anhui University, China. His research interests include feature selection, casual discovery, and data mining.

Liang Du received the BE degree in software engineering from Wuhan University, in 2007, and the PhD degree in computer science from the Institute of Software, University of Chinese Academy of Sciences, in 2013. From 2013 to 2014, he was a softwares engineer with Alibaba Group. He was also an assistant researcher with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. He is currently an associate professor with Shanxi University. He has published more than 40 papers in top conferences and journals, including KDD, IJCAI, AAAI, ICDM, *IEEE Transactions on Knowledge and Data Engineering*, SDM, and CIKM. His research interests include clustering with noise and heterogeneous data, ranking for feature selection, active learning, and document summarization.

Xinwang Liu received the PhD degree from the National University of Defense Technology (NUDT), China, in 2013. He is now professor with the School of Computer, NUDT. His current research interests include kernel learning, multi-view clustering, and unsupervised feature learning. He has published more than 80 peer-reviewed papers, including those in highly regarded journals and conferences, such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Information Forensics and Security*, ICML, NeurIPS, CVPR, ICCV, AAAI, IJCAI, etc. He is an associate editor of *IEEE Transactions on Neural Networks and Learning Systems* and Information Fusion Journal.