Enhanced Adjacency-constrained Hierarchical Clustering using Fine-grained Pseudo Labels

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

***Abstract*—Hierarchical clustering is able to provide partitions of different granularity levels. However, most existing hierarchical clustering techniques perform clustering in the original feature space of the data, which may suffer from overlap, sparseness, or other undesirable characteristics, resulting in noncompetitive performance. In the field of deep clustering, learning representations using pseudo labels has recently become a research hotspot. Yet most existing approaches employ coarse-grained pseudo labels, which may contain noise or incorrect labels. Hence, the learned feature space does not produce a competitive model. In this paper, we introduce the idea of fine-grained labels of supervised learning into unsupervised clustering, giving rise to the enhanced adjacency-constrained hierarchical clustering (ECHC) model. The full framework comprises four steps. One, adjacency-constrained hierarchical clustering (CHC) is used to produce relatively pure fine-grained pseudo labels. Two, those fine-grained pseudo labels are used to train a shallow multilayer perceptron to generate good representations. Three, the corresponding representation of each sample in the learned space is used to construct a similarity matrix. Four, CHC is used to generate the final partition based on the similarity matrix. The experimental results show that the proposed ECHC framework not only outperforms 14 shallow clustering methods on eight real-world datasets but also surpasses current state-of-the-art deep clustering models on six real-world datasets. In addition, on five real-world datasets, ECHC achieves comparable results to supervised algorithms.**

***Index Terms*—clustering, hierarchical clustering, fine-grained, pseudo labels, representation learning.**

I. INTRODUCTION

CLUSTERING is a well-known machine learning technique for detecting latent patterns in a set of unlabeled data [1], [2]. It is used in almost all science disciplines, such as chemistry, biology, astronomy, image processing, text analysis, and pattern recognition. Clustering is usually divided into the following categories: partition-based, hierarchy-based, density-based, model-based, and grid-based [3].

Unlike other types of clustering algorithms, hierarchical clustering can provide partitions of different granularity levels according to the inherent characteristics of the given data. In this way, hierarchical clustering is able to meet the requirements of users who need to divide data with different resolutions. Hierarchical agglomerative clustering is one of the most typical hierarchical clustering strategies. The most typical and intuitive paradigm is to initially regard each sample as a cluster and then continue merging the two closest clusters until a certain condition is met. In the past few decades, many researchers have devoted their attention to hierarchical clustering. For example, the single-linkage method assumes that the distance between two clusters is the minimum distance between any sample in one cluster and any sample in the other cluster [3] [4], i.e., $d(A,B) = \min_{a\in A, b\in B} d(a,b)$, while the average-linkage method assumes that the distance between two clusters is the average distance between any sample in one cluster and any sample in the other cluster [3] [5], i.e., $d(A,B) = \frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B} d(a,b)$. The complete-linkage method defines the maximum distance between any sample in one cluster and any sample in the other cluster as the distance between the two clusters [3] [6], i.e., $d(A,B) = \max_{a\in A, b\in B} d(a,b)$. In addition, there are many other methods that further optimize hierarchical agglomerative clustering. For example, Joe H. Ward, Jr. proposed the ward-linkage algorithm [7]. The algorithm follows a general agglomerative hierarchical clustering process in which the optimal value of an objective function is used to determine which pair of clusters to merge at each phase. Wei Zhang et al. proposed the graph degree linkage method as a way to help hierarchical clustering better identify the manifolds of a dataset. This method investigates the functions of indegree and outdegree – two important concepts in graph theory [8]. M. Saquib Sarfraz et al. leveraged the first neighbor relations to speed up traditional agglomerative clustering, reducing its time complexity to $O(nlogn)$ [9]. Yang et al. [10] introduced HCDC, an innovative hierarchical clustering approach that uses density-distance cores to effectively process datasets of varying density. The approach demonstrates a significant performance improvement over other techniques. Huang et al. [11] also introduced an innovative algorithm, this time based on ensemble hierarchical clustering. The algorithm selects primary clusters from the partitions based on their merit, which is calculated via a normalized mutual information measure. Decelle and colleagues [12] present a novel and universally applicable method of performing hierarchical clustering by constructing relational data trees using the learning dynamics of restricted Boltzmann machines. From the above, we can see that most previous studies mainly focus on optimizing how the distance between clusters is calculated and how different clusters are merged. However, most methods still perform the clustering in the original feature space of the data. Hence, if the feature space has overlapping data or is sparse, performance will suffer. For more complex

Jie Yang and Chin-Teng Lin are with Computational Intelligence and Brain Computer Interface Lab, GrapheneX-UTS HAI Centre, Australian AI Institute, FEIT, University of Technology Sydney, Australia (e-mail: jie.yang-1@uts.edu.au, chin-teng.lin@uts.edu.au).

datasets, such as those with high-dimensional sparseness, users often employ clustering algorithms combined with metric or subspace learning, such as spectral clustering [13] or subspace clustering [14]. More specifically, Chang et al. recently introduced two techniques – one that relies on unsupervised feature selection based on multiple graph fusion [15] and another that involves unified one-step spectral clustering [16]. Both are adeptly designed for clustering high-dimensional datasets and have showcased commendable performance. However, these algorithms cannot provide partitions with different granularity levels.

In recent years, neural network-based clustering has come to the fore. These algorithms, often called deep clustering algorithms, typically use deep neural network structures to learn low-dimensional representations that help when clustering the more complex datasets [17]. Since clustering is an unsupervised learning paradigm, one of the most common methods of deep clustering is to use the results of clustering algorithms (i.e., pseudo labels) to approximate ground-truth labels and then complete the learning of good feature spaces or representations. However, these pseudo labels are often noisy or incorrect, so they cannot and do not fully express the interclass discrimination power of ground-truth labels. Consequently, many studies have been devoted to optimizing pseudo labels to improve the final clustering performance. For example, to reduce the noise in pseudo labels, Xiao Zhang et al. proposed properly estimating the similarity of pseudo labels between consecutive training generations with clustering consensus [18]. Sungwon Park et al. proposed robust learning for unsupervised clustering, which attempts to remove incorrect labels from a dataset in a label-smoothing manner while retaining clean samples with pseudo labels [19]. In Hu Lu et al.'s approach, samples close to the cluster centers are selected as reliable samples and their pseudo labels are used to train deep neural networks, reducing the impact of pseudo label noise on performance [20]. Shanshan Wang et al. proposed using cluster-soft pseudo labels instead of hard pseudo labels to train deep representations so as to reduce the impact of noise on model performance [21]. Louis Mahon et al. proposed an ensemble clustering strategy to fuse pseudo labels from multiple clustering algorithms to improve the accuracy and reliability of pseudo labels and to enhance the performance of the deep clustering model [22]. So we can see that many past studies have focused on improving the accuracy of pseudo labels and on reducing noise. Yet few studies have explored the potential of using fine-grained pseudo labels in clustering to improve the model's performance. By fine-grained pseudo labels, we mean pseudo labels where the number of clusters is greater than the ground truth. This is as opposed to coarse-grained pseudo labels where the number of clusters is equal to the ground truth.

Our strategy of using fine-grained pseudo labels to enhance clustering performance was inspired by supervised learning. More specifically, we were motivated by the idea that training neural networks with fine-grained labels improves both the network optimization and the model's ability to generalize, which in turn improves the networks' classification accuracy [23]. One of the reasons for this is that fine-grained labels contain richer information than coarse-grained labels, which helps the neural networks to learn more discriminative representations. It therefore makes sense to introduce this same idea into unsupervised clustering – the result being enhanced adjacency-constrained hierarchical clustering (ECHC) with fine-grained pseudo labels. Our ECHC framework comprises four steps. First, adjacency-constrained hierarchical clustering (CHC) from one of our previous works [24] is used to produce relatively pure fine-grained pseudo labels. Second, the raw dataset with its fine-grained pseudo labels is fed into a multilayer perceptron (MLP) [25] with only one hidden layer for training. Third, once the training is complete, the corresponding representation of each sample in the learned new space is used to construct a similarity matrix. Fourth, CHC is used to generate the final partitioning scheme based on the similarity matrix.

We have two objectives with this paper. First, we aspire to open new avenues for pseudo label-based clustering algorithms by hypothesizing that using fine-grained pseudo labels for training representations may yield better clustering results than using coarse-grained pseudo labels. Second, we present the ECHC model, which integrates the representations provided by a shallow MLP with the clustering results of the CHC algorithm. Notably, the ECHC model delivers a new benchmark in state-of-the-art performance on six real-world datasets. Hence, ECHC offers a competitive alternative to other deep neural network-based clustering algorithms, particularly in scenarios where computational resources are limited.

Thus, the key contributions of our paper are:

1) Drawing inspiration from the use of fine-grained labels in supervised learning, we are the first to introduce the concept of fine-grained pseudo labels into unsupervised clustering, pioneering a new direction for pseudo label-based clustering algorithms.
2) We propose the ECHC model, an enhanced adjacency-constrained hierarchical clustering algorithm that relies on fine-grained pseudo labels.
3) We demonstrate the superiority of our proposed ECHC model through a comparative analysis with 14 advanced shallow clustering methods on eight real-world datasets.
4) We pit our ECHC model against the current state-of-the-art deep clustering algorithms and two supervised classification algorithms. The results show that ECHC achieves new benchmark for excellence on six real-world datasets and reaches accuracy comparable to supervised algorithms on five real-world datasets.

By introducing the innovative concept of fine-grained pseudo labels to unsupervised clustering, our ECHC framework charts a novel trajectory for hierarchical clustering in that it can craft representations that more adeptly navigate the challenges of sparsity and overlap in high-dimensional data spaces.

II. Related Work

In our previous work, we proposed a simple CHC method that provides both partitions of different granularities and partitions matching a specified number of clusters [24].

Given a dataset X, each sample is initially its own cluster. The number of samples contained in a cluster is regarded as the size of the cluster; therefore, initially, the size of each cluster equals 1. The following rule is then applied to form connections between clusters:

$$\zeta_j \rightarrow \zeta_j^N, if\ size(\zeta_j) \leq size(\zeta_j^N) \quad (1)$$

where ζ_j denotes the j-th cluster, ζ_j^N denotes the 1-nearest cluster of ζ_j, and $size(\zeta_j)$ represents the size of ζ_j. The symbol " → " denotes a connection (i.e., a merger) C_j between ζ_j and ζ_j^N. This process can also be defined in a graph G,

$$A(\zeta_j, \zeta_j^N) = \begin{cases} 1, & if\ size(\zeta_j) \leq size(\zeta_j^N) \\ 0, & otherwise \end{cases} \quad (2)$$

where A is the adjacency matrix of G. Then, new clusters can be obtained by calculating the connected components of the adjacency matrix A. At this point, one iteration has been completed. By repeating this merger process according to Eq. (2), all clusters will eventually merge into one cluster and form a hierarchical tree. Each layer of the hierarchical tree can then be regarded as a partition given a specific granularity. Obviously, this merging process is different from previous hierarchical clustering algorithms because it not only relies on 1-nearest statistics, it also coordinates the clusters according to size – that is, the larger-sized clusters guide the smaller-sized clusters to complete the merger.

Each connection (i.e., merger) C_j has two intuitive properties.

One is the product of the size of the two clusters it is connecting:

$$M_j = size(\zeta_j) \times size(\zeta_j^N) \quad (3)$$

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

$$D_j = d^2(\zeta_j, \zeta_j^N) \quad (4)$$

A reasonable partition can be obtained through a certain layer (granularity) of the clustering tree. In addition, CHC can be assigned the desired number of clusters K. After simply removing K-1 connections with a relatively large $M_j \times D_j$, we have a partition containing K clusters.

CHC uses constrained adjacency merging to prevent unexpected super-large clusters from forming, which, in turn, prevents some erroneous mergers. However, similar to most previous hierarchical clustering algorithms, CHC still calculates similarity (or dissimilarity) based on the original feature space to determine the 1-nearest clusters, resulting in poor performance on some of the more complex datasets.

III. Proposed Model

A. Fine-Grained Pseudo labels from CHC

In supervised learning, researchers have used fine-grained labels instead of coarse-grained labels to improve the generalization ability and recognition accuracy of neural networks and also to train the parameters. [23]. This is because fine-grained labels encourage the networks to learn more features, thus improving the interclass discrimination power of the learned representations. Fig. 1 demonstrates the granularity level of some labels, showing different levels of granularity in the label information from left to right from coarse-grained to fine-grained.

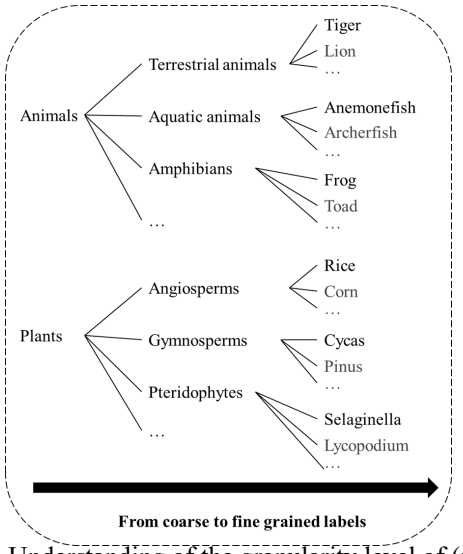

Figure 1. Understanding of the granularity level of (pseudo) labels.

However, unlike supervised learning, in clustering analysis, the labels generated by clustering algorithms are usually referred to as pseudo labels. Further, pseudo labels with more clusters than the ground truth are typically called fine-grained pseudo labels, while the labels with clusters equal to the ground truth are called coarse-grained pseudo labels. In previous studies, most deep clustering algorithms approximate the coarse-grained pseudo labels generated from clustering as ground-truth labels to train good representations [17], whereas very few algorithms use fine-grained pseudo labels to train the representations.

In Fig. 1, we can see that there is a hierarchical affiliation structure between the coarse-grained labels and the fine-grained labels. Therefore, a feasible way to obtain fine-grained pseudo labels is to use hierarchical clustering algorithms. Here, we used CHC to generate fine-grained pseudo labels because CHC's constrained merging process effectively prevents samples of different classes from merging, making purer and cleaner fine-grained partitions. For more clarity, we compared the fine-grained pseudo labels generated by CHC with those generated by other hierarchical clustering algorithms at the same granularity level on a synthetic dataset. As shown in Fig. 2, the other hierarchical clustering algorithms included agglomerative clustering average-linkage (AC-A), ward-linkage (AC-W), and single-linkage (AC-S) [26]. CHC automatically generates different granularity levels according to the natural structure of the dataset. We used a purity score to evaluate how close the pseudo labels provided by each hierarchical clustering algorithm were to the ground-truth labels. A score of 1 indicates the highest degree of closeness to the ground truth at each granularity level. Notably, however, none of the algorithms other than CHC provided good fine-grained pseudo labels.

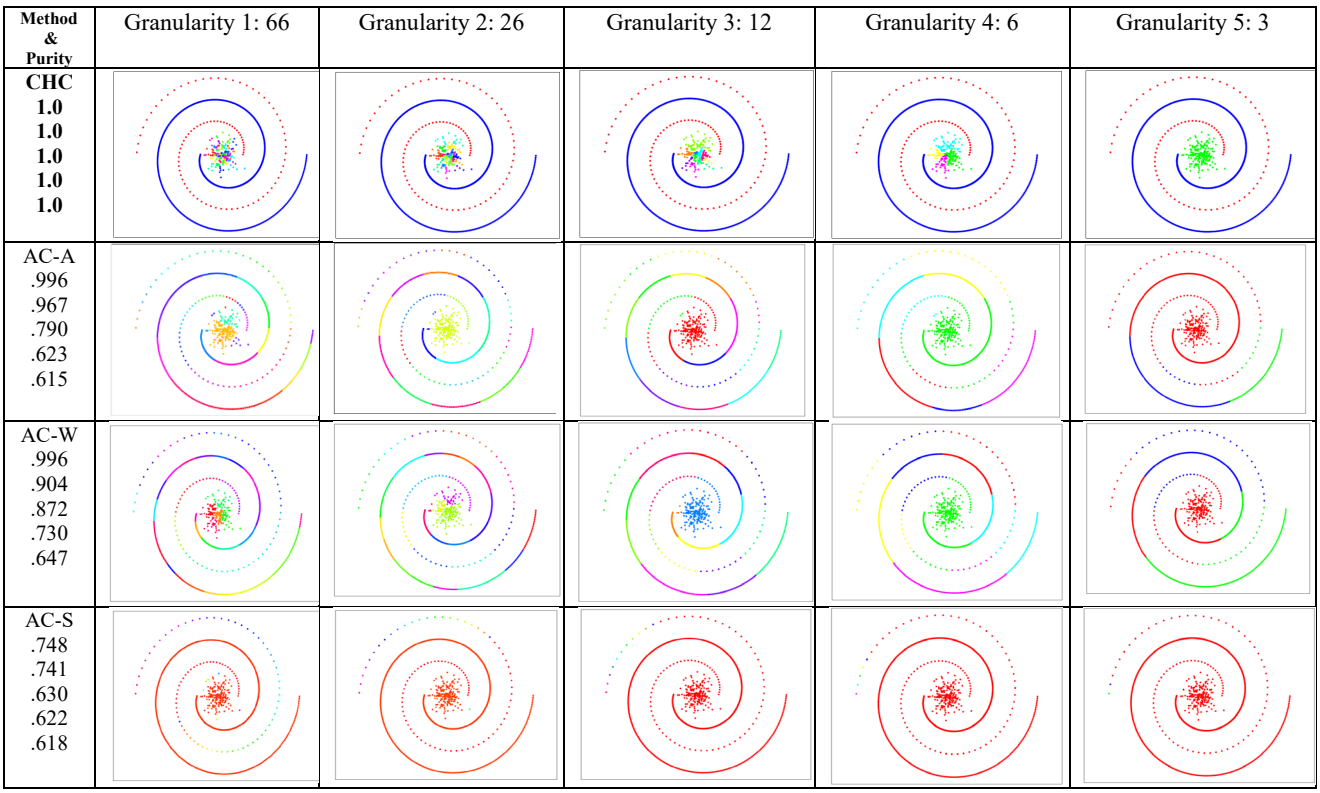

Figure 2. A comparison between the fine-grained pseudo labels generated by CHC and those generated by other hierarchical clustering algorithms on a synthetic dataset and at different granularity levels. The assessment is in terms of purity score. CHC's constrained merging process found fine-grained pseudo labels that were much closer to the ground truth.

B. Training Good Representations using Fine-Grained Pseudo labels and Constructing a Similarity Matrix

Most deep clustering algorithms based on pseudo labels use deep autoencoders, deep convolutional neural networks, or other deep network structures as a backbone to train good representations [17]. However, our ECHC framework adopts a multilayer perceptron (MLP) with only one hidden layer as the primary training structure. There are two main benefits to this approach. First, a shallow MLP converges faster and has better interpretability than other deep network structures. Second, since this paper specifically explores the impact of fine-grained pseudo labels on clustering, using a shallow MLP does not introduce too many uncontrollable factors as would be the case with other complex network structures.

The fine-grained pseudo labels provided by CHC are considered to be ground-truth labels and are used to train an MLP with only one hidden layer. The training process is the same as supervised learning. This basic MLP structure was implemented using the "pattern recognition network tool" provided in MATLAB. We fixed the number of neurons in the hidden layer to 100, but used the default MATLAB settings for all other settings. For example, the loss function is a categorical cross-entropy function. The activation function in the hidden layer is a hyperbolic tangent function, and the activation function in the output layer is a softmax function. A scaled conjugate gradient-based backpropagation algorithm trains the weight parameters of the MLP, and the learning rate is 0.01.

Once this shallow MLP is trained based on fine-grained pseudo labels, the 100-dimensional embeddings of the hidden layer are regarded as newly learned representations. A similarity matrix is then constructed from these learned representations using a *K*-nearest graph as follows [27]:

Consider a dataset with n samples, $X = \{x_1, x_2, \dots, x_n\}$ and its corresponding learned representations $Z = \{z_1, z_2, \dots, z_n\}$. The similarity between any two samples x_i, x_j on the learned space can be defined as

$$s_{ij} = \begin{cases} exp\left(-\frac{dist(z_i,z_j)^2}{\sigma^2}\right), if\ z_j \in \mathcal{N}_i^K \\ 0, \quad otherwise, \end{cases} \tag{5}$$

where $dist(z_i, z_j)$ is the distance between z_i and z_j, and $\mathcal{N}_i^K$ is the set of the K-nearest neighbors of z_i. $\sigma^2 = \frac{1}{nK}\left[\sum_{i=1}^{n}\sum_{z_j \in \mathcal{N}_i^K} dist(z_i, z_j)^2\right]$. K is the hyperparameter to be set.

C. Proposed ECHC Framework

A simple flow chart of the four steps comprising the ECHC method is shown in Fig. 3. In Step (1), CHC finds the fine-grained pseudo labels. In Step (2), the raw dataset with its fine-grained pseudo labels is fed into the MLP with only one hidden layer for training. In Step (3),

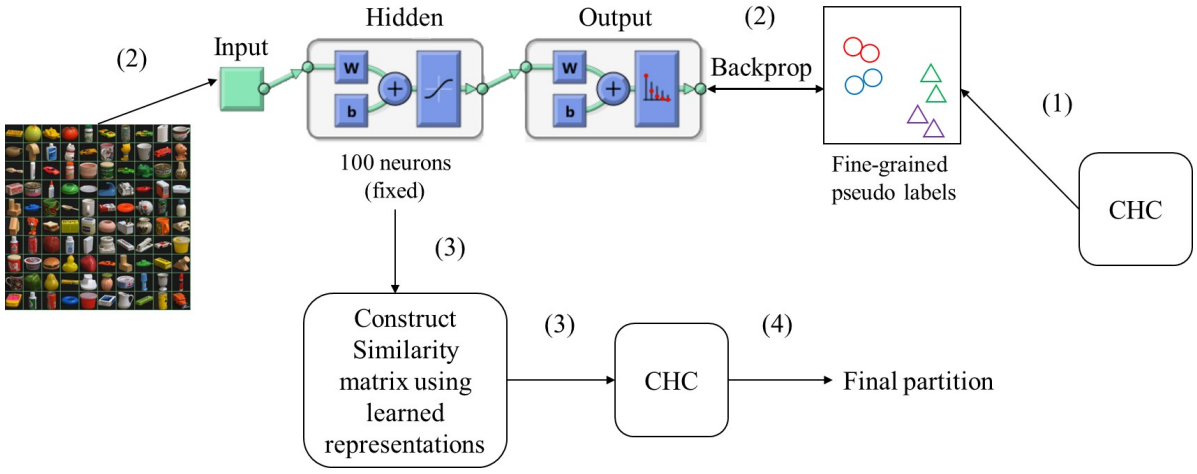

Figure 3. Simple flowchart of the proposed ECHC framework.

the corresponding representation of each sample in the learned space is used to construct a similarity matrix. In Step (4), the CHC algorithm is used again to derive the final partition based on the similarity matrix.

D. Framework Analysis

The proposed ECHC method trains clustering-friendly representations based on the fine-grained pseudo labels provided by CHC and the shallow MLP. After constructing the similarity matrix, CHC is used again to perform post-clustering and determine the final partition.

ECHC's time complexity is as follows:

Suppose a dataset has *n* samples and *d*-dimensional features. In Step (1), CHC is used to generate fine-grained pseudo labels at a complexity of $O(n^2)$. In Step (2), the complexity of training a shallow MLP is approximately $O(nd)$. In Step (3), the complexity of constructing the similarity matrix is $O(n^2)$. And, in Step (4), the complexity of the post-clustering procedure is $O(n^2)$. Therefore, the total time complexity is approximately $O\big(n(n+d)\big)$.

Alternatively, since CHC is a nearest-neighbor-based hierarchical clustering method, nearest neighbor estimation algorithms, such as the k-d tree algorithm, could be used to construct a sparse similarity or dissimilarity matrix [9]. With this approach, the total complexity for Steps (1), (3), and (4) would be reduced to $O(nlogn)$, making the final time complexity $O\big(n(logn+d)\big)$.

The ECHC framework includes two hyperparameters: one being the granularity of the fine-grained pseudo labels generated by CHC, and the other being the *K* value used to construct the *K* nearest neighbor graph when constructing the similarity matrix. The specific value of the granularity of the fine-grained pseudo labels is automatically generated by CHC, but users can choose one of several granularities to suit their needs. For example, in Fig. 2, CHC provides fine-grained pseudo labels with five different granularities. A rule of thumb is that, if the original feature space of the target dataset contains a large overlap, the user should choose fine-grained pseudo labels with finer granularity to train the representations for the post-clustering step.

IV. Experiment

We conducted extensive experiments to assess the efficacy of ECHC. Descriptions of the datasets, comparative baselines, and experimental settings used follow.

A. Datasets

1) **COIL-100** [28]**:** This dataset comprises 72 photos of each of 100 objects, totaling 7200 images in the dataset. The 72 images of each object were captured by placing the object on a 360° rotating turntable and taking a photo every 5°. The resulting photos reflect quite complicated geometry and reflectance features in the objects. Each photo has 49,152 features (128 x 128 pixels in the RGB channels).
2) **COIL-40:** COIL-40 is a subset of the COIL-100 dataset containing a total of 2880 images of the first 40 objects. Many previous studies have used this dataset as a benchmark dataset, so we have followed this common practice.
3) **COIL-20** [29]**:** This dataset contains 72 photographs of each of 20 objects making a total of 1,440 images. Each image has 256 different shades of gray and is 128x128 pixels. A vector of 16,384 dimensions represents each image.
4) **FRGC-v2.0** [30]**:** FRGC-v2.0 is a set of color images of faces. We used a portion of this dataset, as outlined in [49], where 20 subjects were chosen, and each image was cropped and shrunk to a fixed size of 32x32 pixels. As the images are RGB, each image consists of three channels.
5) **UMIST** [31]**:** The UMIST Face dataset consists of 575 photos of 20 people of mixed races, genders, and appearances. Each person is displayed in a variety of positions, from frontal to profile views. The dimensions of each image are 112 × 92 pixels.
6) **Extend-YaleB:** The Extended YaleB database has approximately 64 frontal-face photos per person and 2414 frontal-face images at a size of 192x168 pixels spread among 38 people. Different lighting situations and diverse facial expressions were used during image acquisition.

7) **Mice protein** [32]**:** This dataset contains the expression levels of 77 proteins that were evaluated in the cerebral cortex of eight groups of control and trisomic mice.
8) **Segment** [33]**:** This traditional image segmentation dataset is from the UCI machine learning repository. Seven types of high-level numeric-valued attributes are used to describe the image data. The samples were chosen at random from a database containing seven outdoor pictures. The photos were manually split to assign a category to each pixel.

The descriptive statistics of these eight datasets are shown in Table I.

TABLE I. FULL STATISTICS OF DATASETS.

Datasets	#Samples	#Dimensions	#Clusters
COIL-100	7200	49152	100
COIL-40	2880	49152	40
COIL-20	1440	16384	20
FRGC-v2.0	2462	3072	20
UMIST	575	10304	20
Extend-YaleB	2414	32256	38
Mice protein	1077	77	8
Segment	2310	19	7

B. Compared Algorithms and Configurations

To quantitatively evaluate the performance of ECHC, we compared it to 14 other well-known or recent clustering algorithms. These algorithms include: K-means++ (K-M++) [34]; spectral clustering (SC) [13], [35]; DBSCAN (DB) [36]; density peak clustering (DPC) [37] and one of its recent variants, shared-nearest-neighbor-based density peak clustering (SNNDPC) [38]; conventional hierarchical agglomerative clustering with single-linkage (AC-S), average-linkage (AC-A), and ward-linkage (AC-W) [26]; two recent agglomerative clustering algorithms, FINCH [9] and the adjacency-constrained hierarchical clustering (CHC) method [24]; plus four recent subspace clustering methods, CDMGC [39], GFSC [40], MSGL [41], and MGSF [42]. We tuned the free hyperparameters of each of the compared algorithms according to its best performance over a large number of possible configurations and runs. For example, DPC has a parameter, *dc*, that is used to calculate the density of samples. To maximize the clustering quality, we followed the guidelines in the original paper and tested *dc* at 1.0% through 2.0% in .1% increments. We then chose the best setting for each dataset. Similarly, DB has two parameters: Minpts and Eps. We tested $Minpts = 10, 20, \dots, 50$ and then chose the best setting for each dataset. The best Eps settings for all the datasets were determined by the method outlined in [43]; i.e., $\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d(x_i, \bar{x})$, where $\bar{x} = \sum_{j=1}^{n} \frac{x_j}{n}$, $d(,)$ denotes the distance and x_i or x_j denotes a sample. We tested $Eps = \bar{d}, \frac{\bar{d}}{2}, \frac{\bar{d}}{3}, \dots, \frac{\bar{d}}{10}$ and chose the best setting for each dataset.

We assessed performance using Accuracy (ACC) and normalized mutual information (NMI) [44] – two commonly used external validation indices for clustering. The top outcomes appear in bold and the next best performance is underlined. All experiments were conducted on a workstation with two 14-core Intel Xeon 6132 CPUs clocked at 2.6 GHz and 3.7 GHz with 96 GB memory. Our code is available at https://github.com/brucejak/ECHC-clustering.

C. Experimental Results and Analysis

The clustering results for ECHC and the other 14 compared clustering algorithms on all datasets are shown in Tables II and III. In general, the proposed ECHC method achieved better performance than all other clustering techniques on all datasets. ECHC showed unparalleled performance benefits when compared to both the non-hierarchical and hierarchical clustering techniques. The percentage improvement in ACC of the ECHC method over the next-best performing algorithm were approximately 8.0% (COIL-100), 12.3% (COIL-20), 11.6% (FRGC-v2.0), 4.3% (UMIST), and 9.4% (Extend-YaleB). In terms of NMI, the improvement was approximately 4.7% (COIL20), 13.5% (FRGC-v2.0), 3.3% (Extend-YaleB), and 7.5% (Mice protein).

D. Comparison with Deep Clustering Algorithms

Most deep clustering methods are based on deep neural networks where clustering-friendly representations are learned to improve accuracy. However, ECHC is only based on an MLP with a single hidden layer, where fine-grained pseudo labels are used to learn shallow representations. But, even though ECHC is a shallow framework, we felt it necessary to compare ECHC with some of the latest deep clustering algorithms. On the famous open-source academic website "papers with code" [45], we found the performance rankings of the latest state-of-the-art deep clustering methods on the six datasets mentioned above – COIL-100, COIL-40, COIL-20, FRGC-v2.0, UMIST, and Extend-YaleB. We also examined the latest papers on deep clustering algorithms and the leaderboards of "papers with code" and gathered together the results of the top seven deep algorithms that have performed best on these datasets (as measured by ACC and NMI). These results are shown in Tables IV and V. Surprisingly, ECHC delivered a new benchmark in state-of-the-art performance against the deep algorithms, surpassing all previous deep clustering methods in terms of both ACC and NMI. More specifically, the ACC results for ECHC method were approximately 6.8, 5.8, and 8.8 percent higher than those of the next best deep clustering method on COIL-40, FRGC-v2.0, and UMIST, respectively. In terms of NMI, the improvement was 2.8, 4.2, and 3.7 on COIL-40, FRGC-v2.0, and UMIST, respectively.

E. Comparison with supervised classification algorithms

One of the ultimate goals of clustering is to perform comparably to the supervised methods. Therefore, we also compared ECHC with two popular supervised classification methods: MLP and *K* nearest

TABLE II. COMPARISON OF ECHC AND THE OTHER 14 CLUSTERING METHODS IN TERMS OF ACC.

Datasets/Methods	K-M++	SC	AC-A	AC-W	AC-S	DPC	DB	FINCH	SNNDPC	CHC	CDMGC	GFSC	MSGL	MGSF	ECHC
COIL-100	.5427	.6418	.2794	.6053	.3504	.5482	.4313	.5351	.3089	.8951	.8856	.5640	.5553	.7699	**.9754**
COIL-40	.6031	.7543	.3080	.6799	.5354	.6382	.5674	.6271	.4917	.9587	.8993	.6323	.6524	.8424	**.9903**
COIL-20	.5551	.7990	.4007	.5444	.3993	.6299	.5674	.4910	.3354	.8764	.8771	.5060	.7347	.7611	**1.0000**
FRGC-v2.0	.3055	.3880	.1430	.2697	.1105	.4525	.2754	.2201	.2262	.4167	.3253	.2961	.3493	.2750	**.5684**
UMIST	.3904	.4590	.4000	.4243	.3478	.6400	.3409	.2939	.4226	.8800	.7391	.4256	.5339	.7670	**.9229**
Extend-YaleB	.6237	.8969	.0356	.8923	.0327	.7510	.1019	.1645	.3165	.7196	.8322	.7584	.7150	.7788	**.9905**
Mice protein	.4356	.5060	.3779	.5395	.1551	.5265	.4067	.4828	.4568	.4875	.3900	.4609	.5357	.5311	**.5645**
Segment	.6200	.5502	.5636	.6519	.1459	.6515	.5913	.4364	.6463	.6126	.5333	.7119	.7160	.5420	**.7466**
Rank	*9.6*	*5.5*	*13.0*	*6.4*	*14.0*	*6.0*	*11.0*	*12.0*	*12.0*	*4.4*	*6.0*	*7.6*	*5.8*	*5.9*	***1.0***

TABLE III. COMPARISON OF ECHC AND THE OTHER 14 CLUSTERING METHODS IN TERMS OF NMI.

Datasets/Methods	K-M++	SC	AC-A	AC-W	AC-S	DPC	DB	FINCH	SNNDPC	CHC	CDMGC	GFSC	MSGL	MGSF	ECHC
COIL-100	.8117	.8562	.7256	.8353	.6990	.8657	.7223	.8160	.6919	.9770	.9774	.8169	.8001	.9439	**.9926**
COIL-40	.8177	.9007	.7221	.8450	.8511	.8825	.7881	.8441	.7802	.9890	.9751	.8180	.8447	.9622	**.9951**
COIL-20	.7449	.8686	.7152	.7601	.7415	.7952	.7542	.7702	.5576	.9528	.9463	.7131	.8498	.8948	**1.0000**
FRGC-v2.0	.3770	.5556	.1484	.3888	.0568	.5072	.3663	.2832	.2307	.5738	.4307	.3692	.4548	.3592	**.7089**
UMIST	.5834	.6775	.6109	.6216	.5844	.8109	.4920	.5866	.6146	.9366	.8707	.6294	.7447	.9042	**.9537**
Extend-YaleB	.8485	.9584	.0855	.9568	.0806	.8964	.2045	.3366	.6410	.8794	.9360	.9115	.8467	.8785	**.9917**
Mice protein	.3946	.5353	.3584	.4886	.0853	.4849	.4753	.5164	.4857	.5429	.5134	.4214	.5259	.5282	**.6176**
Segment	.5776	.6599	.6526	.6010	.0342	.7282	.5632	.5376	.6572	.6741	.6637	.6430	.6816	.6948	**.7523**
Rank	*11.0*	*4.8*	*13.0*	*7.8*	*13.0*	*5.6*	*12.0*	*10.0*	*12.0*	*3.1*	*4.4*	*9.6*	*7.0*	*5.1*	***1.0***

TABLE IV. COMPARISON OF ECHC AND THE CURRENT SOTA DEEP CLUSTERING METHODS IN TERMS OF ACC.

Datasets	COIL-100	COIL-40	COIL-20	FRGC-v2.0	UMIST	Extend-YaleB
Rank 1	.947 DCV[46] 2022	.922 A-DSSC [47] 2020	1 JULE [48] 2016	.510 ASC2D[49] 2021	.835 AASSC[50] 2022	.989 DSC-FEDL [51] 2020
Rank 2	.916 JULE [48] 2016	.899 J-DSSC [47] 2020	.983 DSC-FEDL [51] 2020	.504 DDSNnet [52] 2021	.819 DSC-FEDL [51] 2020	.988 SADSC[53] 2021
Rank 3	.863 DGMM [54] 2021	.842 DSC-FEDL [51] 2020	.975 SADSC [53] 2021	.470 DEPICT [55] 2017	.810 RGRL [56] 2020	.986 DSCSC[57] 2021
Rank 4	.824 A-DSSC [47] 2020	.842 DSC-DAG [58] 2020	.973 DSC-DAG [58] 2020	.464 DNB [59] 2021	.809 JULE [48] 2016	.986 DASC [60] 2018
Rank 5	.796 J-DSSC [47] 2020	.840 RGRL [56] 2020	.968 S^2DSCAG [58] 2020	.461 JULE [48] 2016	.781 S^2DSCAG [58] 2020	.973 DSC-2[61] 2017
Rank 6	.775 DBC [62] 2018	.839 S^2DSCAG [58] 2020	.968 PSOC[63] 2022	.431 MI-ADM [64] 2021	.772 DSC-DAG [58] 2020	.924 J-DSSC [47] 2020
Rank 7	.718 DDSNnet[52] 2021	.835 DASC [60] 2018	.946 DGMM [54] 2021	.381 DPSC [65] 2021	.710 DNB [59] 2021	.917 A-DSSC [47] 2020
ECHC	**.975**	**.990**	**1**	**.568**	**.923**	**.991**

neighbor (KNN) [66]. Here, we randomly divided each dataset, using 70% of the samples for training and the remaining 30% for the test set. We then estimated classification accuracy with the test set through cross-validation by parsing ECHC across each full dataset, and then calculating its accuracy on the test set. This process was performed three times, with the average results reported in Table VI. Interestingly, on the five datasets of COIL-100, COIL-40, COIL-20, Extend-YaleB, and UMIST, ECHC delivered performance comparable to the supervised methods.

F. Comparison on larger-scale datasets

To assess the scalability of ECHC, we conducted additional comparative tests against five leading baselines — SC, DPC, CHC, MSGL, and MGSF — on larger- scale datasets. These methods were handpicked based on their ranks in Table II. We selected the renowned Pendigits[1] and CMUPIE[2] datasets for this endeavor, both of which contain around 11,000 samples. As depicted in Table VII, even on these larger-scale datasets, ECHC consistently demonstrates a marked performance superiority over its counterparts.

V. ABLATION STUDY

To further assess the contribution of each component of ECHC to the final clustering results, we conducted an ablation study.

A. Impact of the Fine-grained Pseudo labels from CHC

To evaluate the benefits of training with fine-grained pseudo labels from CHC, we replaced the fine-grained

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#pendigits

[2] https://github.com/jindongwang/transferlearning/blob/master/data/dataset.md#pie

TABLE V. COMPARISON OF ECHC AND THE CURRENT SOTA DEEP CLUSTERING METHODS IN TERMS OF NMI.

Datasets	COIL-100	COIL-40	COIL-20	FRGC-v2.0	UMIST	Extend-YaleB
Rank 1	.985 JULE [48] 2016	.967 A-DSSC [47] 2020	1 JULE [48] 2016	.667 ASC2D[49] 2021	.917 DSC-FEDL [51] 2020	.989 SADSC[53] 2021
Rank 2	.979 DCV[46] 2022	.963 J-DSSC [47] 2020	.981 DSC-FEDL [51] 2020	.651 DNB [59] 2021	.893 S^2DSCAG [58] 2020	.982 DSCSC[57] 2021
Rank 3	.946 A-DSSC [47] 2020	.951 DSC-FEDL [51] 2020	.979 SADSC [53] 2021	.610 DEPICT [55] 2017	.890 DSC-DAG [58] 2020	.980 DASC [60] 2018
Rank 4	.943 J-DSSC [47] 2020	.928 RGRL [56] 2020	.974 S^2DSCAG [58] 2020	.580 MI-ADM [64] 2021	.890 AASSC[50] 2022	.974 DSC-FEDL [51] 2020
Rank 5	.910 DGMM [54] 2021	.920 DASC [60] 2018	.974 PSOC[63] 2022	.574 JULE [48] 2016	.881 RGRL [56] 2020	.970 DSC-2[61] 2017
Rank 6	.905 DBC [62] 2018	.916 DSC-DAG [58] 2020	.958 DSC-DAG [58] 2020	.544 DPSC [65] 2021	.877 JULE [48] 2016	.952 J-DSSC [47] 2020
Rank 7	.886 DDSNnet[52] 2021	.913 S^2DSCAG [58] 2020	.910 DGMM [54] 2021	.522 DDSNnet [52] 2021	.851 DNB [59] 2021	.947 A-DSSC [47] 2020
ECHC	**.993**	**.995**	**1**	**.709**	**.954**	**.992**

TABLE VI. COMPARISON OF ECHC AND TWO SUPERVISED CLASSIFICATION METHODS.

		COIL-100	Extend-YaleB	COIL-20	COIL-40	UMIST	FRGC-v2.0	Mice Protein	Segment
MLP (supervised)	NMI	.9875	**.9970**	.9782	.9946	**.9752**	**.9867**	**.9805**	**.9119**
	ACC	.9838	**.9972**	.9823	**.9946**	**.9748**	**.9919**	**.9907**	**.9533**
KNN (supervised)	NMI	.9905	.9931	.9932	.9933	.9673	.9097	.9705	.8979
	ACC	**.9873**	.9936	.9938	.9927	.9671	.9413	.9845	.9485
ECHC (unsupervised)	NMI	**.9931**	**.9894**	**1**	**.9956**	**.9695**	.6960	.6217	.7823
	ACC	**.9731**	**.9848**	**1**	**.9915**	**.9380**	.5646	.5841	.7759

TABLE VII. COMPARISON OF ECHC AND FIVE LEADING BASELINES ON TWO LARGER-SCALE DATASETS.

	SC		DPC		CHC		MSGL		MGSF		ECHC	
	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC
Pendigits	.7846	.7331	.7709	.7975	.8310	.8193	.7586	.8292	.7223	.6041	**.8522**	**.8873**
CMUPIE	.3783	.1733	.4834	.2330	.5803	.3168	.2514	.1206	.2516	.0893	**.7271**	**.5199**

pseudo labels of CHC in ECHC framework with the coarse-grained pseudo labels of SC and CHC, while keeping the other parts of ECHC unchanged. According to Table II, the performance of these two methods was second only to that of ECHC. Next, we replaced the fine-grained pseudo labels of CHC in ECHC framework with the fine-grained pseudo labels from other hierarchical clustering methods, including AC-A, AC-W, and AC-S. As shown in Table VIII, the ECHC framework with the fine-grained (fg) pseudo labels from CHC yielded the best performance on almost all datasets. It is worth mentioning that, on the COIL-40 dataset, ECHC with the coarse-grained (cg) pseudo labels from CHC yielded the best result. This is because the partition with the coarse granularity generated by CHC on the COIL-40 dataset already has high accuracy (see Tables II-III) and is close to the ground-truth partition. For better clarity, we took the Extend-YaleB dataset and used fine-grained pseudo labels of AC-A, AC-W, AC-S, and CHC and coarse-grained pseudo labels of SC and CHC to train the representations and then used t-SNE [67] to visualize them. As shown in Figure 4, all configurations except for the representations obtained when using fine-grained pseudo labels of CHC had different degrees of overlap, resulting in lower post-clustering performance. More details of the fine-grained pseudo labels from CHC on the eight real-world datasets are shown in Table IX.

B. Impact of CHC

After learning good representations based on the fine-grained pseudo labels of CHC, ECHC uses CHC again to cluster the representations based on the constructed similarity matrix. We chose CHC as the clustering algorithm because it performs better than any other clustering algorithms, as shown in Tables II-III. To further illustrate this issue, we replaced the CHC module used for the post-clustering step with other clustering algorithms and compared performance. The other algorithms were K-means++, SC, AC-A, AC-W, and AC-S. As shown in Table X, compared to the other clustering algorithms, CHC better clusters the learned representations to improve accuracy.

C. Impact of the Proposed ECHC Framework

TABLE VIII. COMPARISON OF THE FINE-GRAINED (FG) PSEUDO LABELS OF CHC AND OTHER FINE-GRAINED (FG) OR COARSE-GRAINED (CG) PSEUDO LABELS.

Measures	NMI						ACC					
Datasets/Methods	AC-A(fg)+CHC	AC-W(fg)+CHC	AC-S(fg)+CHC	SC(cg)+CHC	CHC(cg)+CHC	CHC(fg)+CHC	AC-A(fg)+CHC	AC-W(fg)+CHC	AC-S(fg)+CHC	SC(cg)+CHC	CHC(cg)+CHC	CHC(fg)+CHC
COIL-100	.9833	.9855	.9887	.9872	.9925	**.9926**	.9367	.9417	.9514	.9597	.9751	**.9754**
COIL-40	.9920	.9931	.9931	.9944	**.9961**	.9951	.9729	.9743	.9868	.9875	**.9958**	.9903
COIL-20	.9679	.9690	.9601	.9554	.9722	**1.0000**	.9215	.9264	.9222	.9118	.9167	**1.0000**
FRGC-v2.0	.6993	.6957	.6511	.5769	.5812	**.7089**	.5445	.5616	.4968	.4236	.4204	**.5684**
UMIST	.9289	.8461	.9463	.8708	.9519	**.9537**	.8730	.7426	.8974	.7861	.9107	**.9229**
Extend-YaleB	.9793	.9774	.9217	.9811	.9365	**.9917**	.9689	.9698	.8335	.9664	.8459	**.9905**
Mice protein	.5823	.5734	.5964	.5647	.5988	**.6176**	.5599	.5432	.5292	.5097	.5487	**.5645**
Segment	.7007	.7204	.6708	.6843	.6990	**.7523**	.6885	.7128	.6864	.6885	.6835	**.7466**

TABLE IX. THE NUMBER OF CLUSTERS PROVIDED AT EACH STEP (GRANULARITY) BY CHC ON EIGHT REAL-WORLD DATASETS IS DETAILED. HERE, #C REPRESENTS THE GROUND-TRUTH NUMBER OF CLUSTERS, AND * DENOTES THE NUMBER OF CLUSTERS INCLUDED IN THE FINE-GRAINED PSEUDO LABELS ADOPTED BY ECHC.

Datasets/Methods	#C	Granularity 1	Granularity 2	Granularity 3	Granularity 4	Granularity 5	Granularity 6	Granularity 7	Granularity 8	Granularity 9
COIL-100	100	2211	975	473	241	**119***	53	18	8	2
COIL-40	40	876	391	192	**98***	55	30	9	4	2
COIL-20	20	421	191	**86***	37	19	6	—		
FRGC-v2.0	20	**775***	260	100	32	9	3	2	—	
UMIST	20	190	81	38	**22***	11	5	2	—	
Extend-YaleB	38	**528***	216	98	35	6	—			
Mice protein	8	351	**130***	62	23	5	2	—		
Segment	7	664	254	**123***	56	24	9	3	—	

TABLE X. COMPARISON OF THE POST CLUSTERING PROCEDURES OF CHC AND THE OTHER CLUSTERING METHODS .

Measures	NMI						ACC					
Datasets/Methods	CHC(fg)+K-M++	CHC(fg)+SC	CHC(fg)+AC-A	CHC(fg)+AC-W	CHC(fg)+AC-S	CHC(fg)+CHC	CHC(fg)+K-M++	CHC(fg)+SC	CHC(fg)+AC-A	CHC(fg)+AC-W	CHC(fg)+AC-S	CHC(fg)+CHC
COIL-100	.8948	.9396	.8695	.9206	.9425	**.9926**	.6946	.8060	.5379	.7682	.7400	**.9754**
COIL-40	.8825	.9473	.8647	.9151	.9347	**.9951**	.6998	.8293	.5788	.8000	.7382	**.9903**
COIL-20	.7844	.9069	.7496	.7893	.9202	**1.0000**	.6219	.7872	.4861	.6444	.7528	**1.0000**
FRGC-v2.0	.5360	.6247	.1997	.6196	.0574	**.7089**	.4311	.5153	.1738	.4752	.1113	**.5684**
UMIST	.7969	.8908	.8301	.8549	.8649	**.9537**	.6179	.7243	.6087	.7235	.6522	**.9229**
Extend-YaleB	.9119	.9760	.9329	.9424	.8594	**.9917**	.7919	.9480	.8128	.8927	.5982	**.9905**
Mice protein	.4175	.5227	.3723	.4750	.1157	**.6176**	.4778	.5404	.3965	.4995	.1578	**.5645**
Segment	.5661	.6438	.3661	.5999	.0441	**.7523**	.6277	.5611	.3645	.6325	.1468	**.7466**

TABLE XI. COMPARISON OF ECHC AND OTHER IMPROVED HIERARCHICAL CLUSTERING FRAMEWORKS.

Methods	AC-S(fg)+AC-S		AC-A(fg)+AC-A		AC-W(fg)+AC-W		CHC(fg)+CHC	
Datasets/Measures	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC
COIL-100	.9610	.8173	.2730	.1012	.4247	.1992	**.9926**	**.9754**
COIL-40	.9738	.8848	.2684	.1216	.4227	.2295	**.9951**	**.9903**
COIL-20	.9063	.7505	.2785	.1428	.4304	.2486	**1.0000**	**1.0000**
FRGC-v2.0	.2388	.1894	.1486	.1317	.1596	.1277	**.7089**	**.5684**
UMIST	.8783	.6713	.3956	.2551	.5256	.3681	**.9537**	**.9229**
Extend-YaleB	.5728	.3481	.3264	.1636	.3970	.2113	**.9917**	**.9905**
Mice protein	.3202	.3185	.1444	.1820	.1751	.2037	**.6176**	**.5645**
Segment	.6311	.4460	.0891	.1659	.0926	.1746	**.7523**	**.7466**

Essentially, ECHC employs the hierarchical clustering algorithm, CHC, from which it obtains fine-grained pseudo labels. These labels are then utilized to derive effective representations. Following this, CHC is applied once again to these learned representations in a post-clustering procedure to yield the final partition. In the preceding two sections, we conducted separate ablation studies for each of these CHC-related procedures. In this section, we execute a combined ablation study, simultaneously substituting the CHC used both for representation learning and the post-clustering procedure within ECHC with other hierarchical clustering algorithms, namely AC-A, AC-W, and AC-S. As demonstrated in Table XI, even with these alterations, ECHC consistently demonstrates a performance edge over other frameworks.

D. Parameter Sensitivity Analysis

ECHC contains two hyperparameters: one being the granularity of the pseudo labels used to train the representations; the other being the number of neighbors K used to construct the similarity matrix. In this section, we explored how sensitive ECHC is to these two hyperparameters by running ECHC with different hyperparameter combinations on the COIL-20 dataset. We generated different granularities for the pseudo labels from CHC, $K \in (4{:}\,2{:}\,50)$. As shown in Figure 5, the results obtained by ECHC under different parameter combinations were all close to 100%.

VI. DISCUSSION

ECHC is a clustering algorithm based on representations of a shallow MLP, so, strictly speaking, it is a shallow clustering method that, performance-wise, should not be able to compete with deep clustering methods. However, using fine-grained pseudo labels to train and get good representations makes the performance of ECHC quite competitive. Our experiments

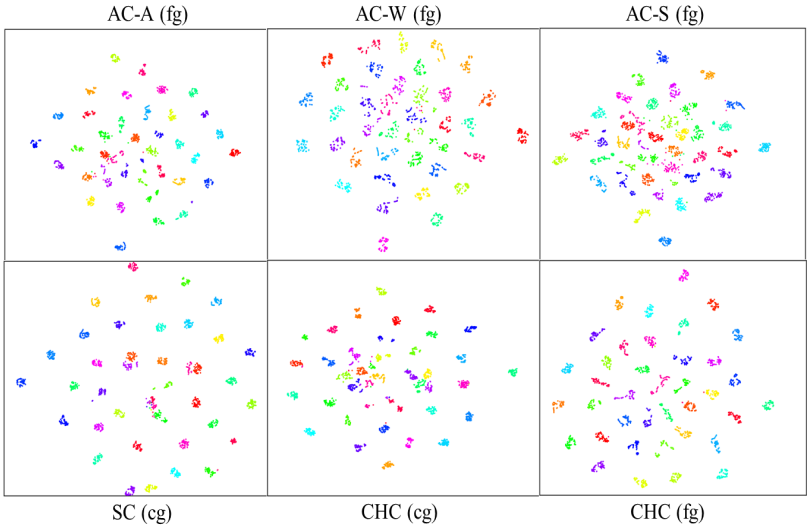

Figure 4. Visualizations of learned representations using the fine-grained (fg) pseudo labels of CHC and other fine-grained (fg) or coarse-grained (cg) pseudo labels on the Extend-YaleB dataset.

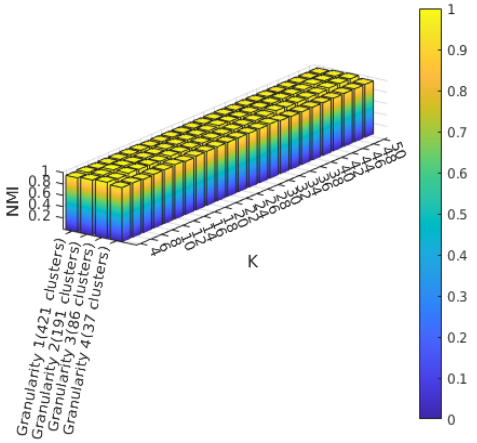

Figure 5. The performance of ECHC with different hyperparameter combinations on the COIL-20 dataset.

with seven state-of-the-art deep clustering algorithms and six real-world datasets show that ECHC does more than compare, it actually sets a new benchmark in state-of-the-art performance. Considering a shallow MLP has better convergence and requires less training time than other deep network structures, we believe that ECHC is a better choice when computing power is limited.

Despite its competitive performance, ECHC does have several potential limitations. First, like the majority of historical clustering algorithms, ECHC requires the target number of clusters to be specified in advance. Yet, in many real-world scenarios, determining the ideal number of clusters without a priori knowledge can be challenging. In future implementations, we intend to introduce an effective internal evaluation index to guide ECHC in finding partitions with a larger inter-cluster distance and smaller intra-cluster distance. This will mean the number of clusters can be determined automatically. Second, the current version of ECHC uses a shallow MLP to learn representations. Yet, for more intricate datasets – where, for example, the original feature maps of the samples bear complex semantic information – a shallow MLP might struggle to produce satisfactory representations even with fine-grained pseudo labels. Moving forward, we are considering more sophisticated network architectures, such as a convolutional neural network (CNN) or variant, to enhance the quality of the representations. We believe this might further improve the accuracy of ECHC. However, it is noteworthy that ECHC may encounter several new challenges when adopting more complex network structures. These challenges include addressing vanishing or exploding gradients and adjusting more hyperparameters. We acknowledge these potential difficulties and will attempt to resolve these issues in our future work.

VII. Conclusion and Future Work

In this paper, we integrated the idea of fine-grained labels from the supervised learning paradigm into unsupervised clustering, proposing enhanced adjacency-constrained hierarchical clustering (ECHC) with fine-grained pseudo labels. The ECHC framework comprises four steps. First, adjacency-constrained hierarchical clustering (CHC) is used to produce relatively pure fine-grained pseudo labels. Second, those fine-grained pseudo labels are used to train a shallow MLP to generate good representations. Third, the corresponding representations of each sample in the learned space are used to construct a similarity matrix. Fourth, CHC is used again to generate the final partition based on the similarity matrix. The experimental results show that this shallow ECHC framework not only outperforms 14 shallow clustering methods on eight real-world datasets but also surpasses seven current state-of-the-art deep clustering models on six real-world datasets. In addition, on five real-world datasets, ECHC achieves results comparable to those of supervised algorithms.

In future work, we will apply the proposed ECHC method to more complex scenarios, such as person reidentification, point cloud segmentation, and image matching. We also intend to improve ECHC by integrating more advanced neural

network architectures, while concurrently addressing any emergent challenges that may arise.

ACKNOWLEDGMENTS

This work was supported in part by the Australian Research Council (ARC) under discovery under Grant DP210101093 and Grant DP220100803, in part by the Australian National Health and Medical Research Council (NHMRC) Ideas under Grant APP2021183, in part by the UTS Human-Centric AI Centre funding sponsored by GrapheneX (2023-2031), in part by the Australia Defence Innovation Hub under Grant P18-650825, and in part by Australian Cooperative Research Centres Projects (CRC-P) Round 11 under Grant CRCPXI000007.

REFERENCES

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

[2] J. Yang, Y.-K. Wang, X. Yao, and C.-T. Lin, "Adaptive initialization method for k-means algorithm," *Front. Artif. Intell.*, vol. 4, 2021, Accessed: May 01, 2022.

[3] A. Saxena *et al.*, "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, Dec. 2017, doi: 10.1016/j.neucom.2017.06.053.

[4] P. H. A. Sneath and R. R. Sokal, *Numerical taxonomy: the principles and practice of numerical classification*. W. H. Freeman and Co., 1973.

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

[6] B. King, "Step-wise clustering procedures," *J. Am. Stat. Assoc.*, vol. 62, no. 317, pp. 86–101, Mar. 1967, doi: 10.1080/01621459.1967.10482890.

[7] J. H. Ward, "Hierarchical grouping to optimize an objective function," *J. Am. Stat. Assoc.*, vol. 58, no. 301, pp. 236–244, Mar. 1963, doi: 10.1080/01621459.1963.10500845.

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

[9] S. Sarfraz, V. Sharma, and R. Stiefelhagen, "Efficient parameter-free clustering using first neighbor relations," in *CVPR*, Jun. 2019, pp. 8926–8935. doi: 10.1109/CVPR.2019.00914.

[10] Q.-F. Yang *et al.*, "HCDC: a novel hierarchical clustering algorithm based on density-distance cores for data sets with varying density," *Inf. Syst.*, vol. 114, p. 102159, Mar. 2023, doi: 10.1016/j.is.2022.102159.

[11] Q. Huang, R. Gao, and H. Akhavan, "An ensemble hierarchical clustering algorithm based on merits at cluster and partition levels," *Pattern Recognit.*, vol. 136, p. 109255, Apr. 2023, doi: 10.1016/j.patcog.2022.109255.

[12] A. Decelle, L. Rosset, and B. Seoane, "Unsupervised hierarchical clustering using the learning dynamics of RBMs," *Phys. Rev. E*, vol. 108, no. 1, p. 014110, Jul. 2023, doi: 10.1103/PhysRevE.108.014110.

[13] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *NeurIPS*, MIT Press, 2002, pp. 849–856. Accessed: Mar. 05, 2020.

[14] E. Elhamifar and R. Vidal, "Sparse subspace clustering: algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, 2013, doi: 10.1109/TPAMI.2013.57.

[15] C. Tang, X. Zheng, W. Zhang, X. Liu, X. Zhu, and E. Zhu, "Unsupervised feature selection via multiple graph fusion and feature weight learning," *Sci. China Inf. Sci.*, vol. 66, no. 5, p. 152101, Apr. 2023, doi: 10.1007/s11432-022-3579-1.

[16] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang, and E. Zhu, "Unified one-step multi-view spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6449–6460, Jun. 2023, doi: 10.1109/TKDE.2022.3172687.

[17] Y. Ren *et al.*, "Deep clustering: a comprehensive survey." arXiv, Oct. 08, 2022. doi: 10.48550/arXiv.2210.04142.

[18] X. Zhang, Y. Ge, Y. Qiao, and H. Li, "Refining pseudo labels with clustering consensus over generations for unsupervised object re-identification," in *CVPR*, 2021, pp. 3436–3445. Accessed: Jan. 08, 2023.

[19] S. Park *et al.*, "Improving unsupervised image clustering with robust learning," in *CVPR*, 2021, pp. 12278–12287. Accessed: Jan. 08, 2023.

[20] H. Lu, C. Chen, H. Wei, Z. Ma, K. Jiang, and Y. Wang, "Improved deep convolutional embedded clustering with re-selectable sample training," *Pattern Recognit.*, vol. 127, p. 108611, Jul. 2022, doi: 10.1016/j.patcog.2022.108611.

[21] S. Wang, L. Zhang, W. Chen, F. Wang, and H. Li, "Refining pseudo labels for unsupervised Domain Adaptive Re-Identification," *Knowl.-Based Syst.*, vol. 242, p. 108336, Apr. 2022, doi: 10.1016/j.knosys.2022.108336.

[22] L. Mahon and T. Lukasiewicz, "Selective pseudo-label clustering," in *GCAI*, 2021, pp. 158–178. doi: 10.1007/978-3-030-87626-5_12.

[23] Z. Chen, R. Ding, T.-W. Chin, and D. Marculescu, "Understanding the impact of label granularity on CNN-based image classification." arXiv, Jan. 21, 2019. doi: 10.48550/arXiv.1901.07012.

[24] J. Yang and C.-T. Lin, "Clustering via torque balance with mass and distance." arXiv, Apr. 27, 2020. doi: 10.48550/arXiv.2004.13160.

[25] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*. in Springer Series in Statistics. New York, NY: Springer, 2009. doi: 10.1007/978-0-387-84858-7.

[26] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, Sep. 1967, doi: 10.1007/BF02289588.

[27] F. Zhu, X. Wang, and N. Yu, "Crowd tracking with dynamic evolution of group structures," in *ECCV*, 2014, pp. 139–154. doi: 10.1007/978-3-319-10599-4_10.

[28] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-100)," *Tech. Rep.*, no. CUCS-006-96, 1996.

[29] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-20)," *Tech. Rep.*, vol. Technical Report CUCS-005-96, 1996.

[30] "Face recognition grand challenge (FRGC v.2.0) data collection." [Online]. Available: https://cvrl.nd.edu/projects/data/#face-recognition-grand-challenge-frgc-v20-data-collection

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

[32] C. Higuera, K. J. Gardiner, and K. J. Cios, "Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome," *PLOS ONE*, vol. 10, no. 6, p. e0129126, Jun. 2015, doi: 10.1371/journal.pone.0129126.

[33] "UCI machine learning repository: data sets." https://archive.ics.uci.edu/ml/datasets.php (accessed Oct. 20, 2019).

[34] D. Arthur and S. Vassilvitskii, "K-means++: the advantages of careful seeding," in *Proc. Eighteenth Annu. ACM-SIAM Symp. Discrete Algorithms Soc. Ind. Appl. Math.*, pp. 1027–1035, 2007.

[35] Jianbo Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000, doi: 10.1109/34.868688.

[36] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "Density-based spatial clustering of applications with noise," in *KDD*, vol. 96, no. 34, pp. 226-231, 1996.

[37] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014, doi: 10.1126/science.1242072.

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

[39] S. Huang, I. W. Tsang, Z. Xu, and J. Lv, "Measuring diversity in graph learning: a unified framework for structured multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5869–5883, Dec. 2022, doi: 10.1109/TKDE.2021.3068461.

[40] Z. Kang *et al.*, "Multi-graph fusion for multi-view spectral clustering," *Knowl.-Based Syst.*, vol. 189, p. 105102, Feb. 2020, doi: 10.1016/j.knosys.2019.105102.
[41] Z. Kang, Z. Lin, X. Zhu, and W. Xu, "Structured graph learning for scalable subspace clustering: from single view to multiview," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8976–8986, Sep. 2022, doi: 10.1109/TCYB.2021.3061660.
[42] K. Zhan, C. Niu, C. Chen, F. Nie, C. Zhang, and Y. Yang, "Graph structure fusion for multiview clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1984–1993, Oct. 2019, doi: 10.1109/TKDE.2018.2872061.
[43] L. Bai, X. Cheng, J. Liang, H. Shen, and Y. Guo, "Fast density clustering strategies based on the k-means algorithm," *Pattern Recognit.*, vol. 71, pp. 375–386, Nov. 2017, doi: 10.1016/j.patcog.2017.06.023.
[44] A. Strehl and J. Ghosh, "Cluster ensembles --- a knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, no. Dec, pp. 583–617, 2002, Accessed: Mar. 20, 2020.
[45] "Papers with code-image clustering." https://paperswithcode.com/task/image-clustering
[46] L. Wu, L. Yuan, G. Zhao, H. Lin, and S. Z. Li, "Deep clustering and visualization for end-to-end high-dimensional data analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–12, 2022, doi: 10.1109/TNNLS.2022.3151498.
[47] D. Lim, R. Vidal, and B. Haeffele, "Doubly stochastic subspace clustering," *arXiv preprint arXiv:2011.14859*, 2020.
[48] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *CVPR*, Jun. 2016, pp. 5147–5156. doi: 10.1109/CVPR.2016.556.
[49] W. Xia, X. Zhang, Q. Gao, and X. Gao, "Adversarial self-supervised clustering with cluster-specificity distribution," *Neurocomputing*, vol. 449, pp. 38–47, Aug. 2021, doi: 10.1016/j.neucom.2021.03.108.
[50] Z. Peng, H. Liu, Y. Jia, and J. Hou, "Adaptive attribute and structure subspace clustering network," *IEEE Trans. Image Process.*, vol. 31, pp. 3430–3439, 2022, doi: 10.1109/TIP.2022.3171421.
[51] Q. Huang, Y. Zhang, H. Peng, T. Dan, W. Weng, and H. Cai, "Deep subspace clustering to achieve jointly latent feature extraction and discriminative learning," *Neurocomputing*, vol. 404, pp. 340–350, Sep. 2020, doi: 10.1016/j.neucom.2020.04.120.
[52] W. Wang, F. Chen, Y. Ge, S. Huang, X. Zhang, and D. Yang, "Discriminative deep semi-nonnegative matrix factorization network with similarity maximization for unsupervised feature learning," *Pattern Recognit. Lett.*, vol. 149, pp. 157–163, Sep. 2021, doi: 10.1016/j.patrec.2021.06.013.
[53] Z. Chen, S. Ding, and H. Hou, "A novel self-attention deep subspace clustering," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 8, pp. 2377–2387, 2021, doi: 10.1007/s13042-021-01318-4.
[54] J. Wang and J. Jiang, "Unsupervised deep clustering via adaptive GMM modeling and optimization," *Neurocomputing*, vol. 433, pp. 199–211, Apr. 2021, doi: 10.1016/j.neucom.2020.12.082.
[55] K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *ICCV*, IEEE Computer Society, Oct. 2017, pp. 5747–5756. doi: 10.1109/ICCV.2017.612.
[56] Z. Kang, X. Lu, J. Liang, K. Bai, and Z. Xu, "Relation-guided representation learning," *Neural Netw.*, vol. 131, pp. 93–102, Nov. 2020, doi: 10.1016/j.neunet.2020.07.014.
[57] B. Peng and W. Zhu, "Deep structural contrastive subspace clustering," in *ACML*, PMLR, Nov. 2021, pp. 1145–1160. Accessed: Jun. 16, 2022.
[58] Z. Yu, Z. Zhang, W. Cao, C. Liu, J. Philip Chen, and H. S. Wong, "GAN-based enhanced deep subspace clustering networks," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2020, doi: 10.1109/TKDE.2020.3025301.
[59] Z. Wang, Y. Ni, B. Jing, D. Wang, H. Zhang, and E. Xing, "DNB: a joint learning framework for deep bayesian nonparametric clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–11, 2021, doi: 10.1109/TNNLS.2021.3085891.
[60] P. Zhou, Y. Hou, and J. Feng, "Deep adversarial subspace clustering," in *CVPR*, Jun. 2018, pp. 1596–1604. doi: 10.1109/CVPR.2018.00172.
[61] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *NeurIPS*, Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp. 23–32.
[62] F. Li, H. Qiao, and B. Zhang, "Discriminatively boosted image clustering with fully convolutional auto-encoders," *Pattern Recognit.*, vol. 83, pp. 161–173, Nov. 2018, doi: 10.1016/j.patcog.2018.05.019.
[63] J. Wang, L. Wang, and J. Jiang, "Preserving similarity order for unsupervised clustering," *Pattern Recognit.*, vol. 128, p. 108670, Aug. 2022, doi: 10.1016/j.patcog.2022.108670.
[64] M. Jabi, M. Pedersoli, A. Mitiche, and I. B. Ayed, "Deep clustering: on the link between discriminative models and k-means," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 06, pp. 1887–1896, Jun. 2021, doi: 10.1109/TPAMI.2019.2962683.
[65] W. Hu, C. Chen, F. Ye, Z. Zheng, and Y. Du, "Learning deep discriminative representations with pseudo supervision for image clustering," *Inf. Sci.*, vol. 568, pp. 199–215, Aug. 2021, doi: 10.1016/j.ins.2021.03.066.
[66] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967, doi: 10.1109/TIT.1967.1053964.
[67] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008, Accessed: Apr. 29, 2022.

Jie Yang (M'23) received the Ph.D. degree from Australian Artificial Intelligence Institute (AAII), University of Technology Sydney (UTS). He currently is a postdoctoral fellow at Computational Intelligence and Brain-Computer Interface Lab, AAII, UTS, Australia. His current research interests include unsupervised learning, clustering, representation learning, and EEG data processing. In addition, he serves as a reviewer for many top-tier journals, such as IEEE Transactions on Fuzzy Systems, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Industrial Informatics, and Information Sciences.

Chin-Teng Lin (S'88–M'91–SM'99–F'05) received a Bachelor's of Science from National Chiao-Tung University (NCTU), Taiwan in 1986, and holds Master's and PhD degrees in Electrical Engineering from Purdue University, USA, received in 1989 and 1992, respectively. He is currently a distinguished professor and Co-Director of the Australian Artificial Intelligence Institute within the Faculty of Engineering and Information Technology at the University of Technology Sydney, Australia. He is also an Honorary Chair Professor of Electrical and Computer Engineering at NCTU. For his contributions to biologically inspired information systems, Prof Lin was awarded Fellowship with the IEEE in 2005, and with the International Fuzzy Systems Association (IFSA) in 2012. He received the IEEE Fuzzy Systems Pioneer Award in 2017. He has held notable positions as editor-in-chief of IEEE Transactions on Fuzzy Systems from 2011 to 2016; seats on Board of Governors for the IEEE Circuits and Systems (CAS) Society (2005-2008), IEEE Systems, Man, Cybernetics (SMC) Society (2003-2005), IEEE Computational Intelligence Society (2008-2010); Chair of the IEEE Taipei Section (2009-2010); Chair of IEEE CIS Awards Committee (2022, 2023); Distinguished Lecturer with the IEEE CAS Society (2003-2005) and the CIS Society (2015-2017); Chair of the IEEE CIS Distinguished Lecturer Program Committee (2018-2019); Deputy Editor-in-Chief of IEEE Transactions on Circuits and Systems-II (2006-2008); Program Chair of the IEEE International Conference on Systems, Man, and Cybernetics (2005); and General Chair of the 2011 IEEE International Conference on Fuzzy Systems. Prof Lin is the co-author of Neural Fuzzy Systems (Prentice-Hall) and the author of Neural Fuzzy Control Systems with Structure and Parameter Learning (World Scientific). He has published more than 425 journal papers including about 200 IEEE journal papers in the areas of neural networks, fuzzy systems, brain-computer

interface, multimedia information processing, cognitive neuro-engineering, and human-machine teaming, that have been cited more than 39,500 times. Currently, his h-index is 95, and his i10-index is 457.