Deep Clustering With Sample-Assignment Invariance Prior

Xi Peng, *Member, IEEE*, Hongyuan Zhu, *Member, IEEE*, Jiashi Feng, Chunhua Shen, Haixian Zhang, *Member, IEEE*, and Joey Tianyi Zhou

***Abstract*— Most popular clustering methods map raw image data into a projection space in which the clustering assignment is obtained with the vanilla k-means approach. In this article, we discovered a novel prior, namely, there exists a common invariance when assigning an image sample to clusters using different metrics. In short, different distance metrics will lead to similar soft clustering assignments on the manifold. Based on such a novel prior, we propose a novel clustering method by minimizing the discrepancy between pairwise sample assignments for each data point. To the best of our knowledge, this could be the first work to reveal the sample-assignment invariance prior based on the idea of treating labels as ideal representations. Furthermore, the proposed method is one of the first end-to-end clustering approaches, which jointly learns clustering assignment and representation. Extensive experimental results show that the proposed method is remarkably superior to 16 state-of-the-art clustering methods on five image data sets in terms of four evaluation metrics.**

***Index Terms*— Label as representation, least square regression, low-rank representation, subspace clustering.**

I. Introduction

DATA clustering aims to group a collection of samples into different clusters by simultaneously minimizing intercluster similarity and maximizing intracluster similarity, which is a popular unsupervised learning technique to analyze unlabeled data [1]. Two challenging problems in clustering analysis are the curse of high dimensionality and linear inseparability of inherent clusters—which have attracted numerous works during past decades [2]–[5]. These two problems are caused by the same factor to a certain extent. To be specific, many real-world data sets, such as documents and images, are with high dimensionality in the input space, thus leading to the curse of high dimensionality. As the high-dimensional data always lie on a low-dimensional manifold, the Euclidean distance cannot accurately measure the dissimilarity between them and thus leads to the linearly inseparable issue. In summary, it is a daunting task to cluster these data using the Euclidean distance-based clustering approaches, such as the vanilla k-means clustering.

To cluster high-dimensional nonlinear data, various methods have been proposed [6]–[8], among which subspace clustering is one of the most effective approaches [9]–[11]. According to the definition given in [12], subspace clustering aims at first implicitly seeking a low-dimensional subspace to fit each group of data points and then separating these data in the projection space with the following steps: 1) learning low-dimensional representations for a given data set and 2) clustering data based on the representations. Through exploiting the low-dimensional subspace structure, subspace clustering could effectively alleviate both the problem of dimensionality curse and linear inseparability.

During past years, most existing subspace clustering methods mainly investigate how to learn a good data representation that is beneficial to discovering inherent clusters [13]–[24], [24]–[31]. Like the standard spectral clustering (SC) [6], those methods achieve data clustering with the following three steps. First, an affinity graph is built to describe the relationship between the data points. Second, low-dimensional data representation is learned by using the graph as an invariance. Third, k-means is conducted on the data representation to obtain clustering assignments. One could find that the first two steps are identical to conducting dimension reduction with manifold learning [32], [33], which learns a low-dimensional representation by first constructing a similarity graph in the input space and then embedding the graph into another space. The earlier observation on the relationship between manifold learning and SC-based subspace clustering was presented in [34]. It should be pointed out that such a unified view is not the unique way to understand the SC-based methods. Here, we adopt this view just for better understanding subspace clustering in the deep learning era.

Although those subspace clustering methods have shown encouraging performance, we observe that they may suffer

Manuscript received August 1, 2018; revised June 17, 2019 and October 3, 2019; accepted December 3, 2019. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant YJ201949, in part by the NFSC under Grant 61806135, Grant 61625204, Grant 61836006, and Grant 61836011, and in part by the Singapore Government's Research, Innovation and Enterprise 2020 Plan (Advanced Manufacturing and Engineering Domain) under Grant A18A1b0045. *(Corresponding author: Joey Tianyi Zhou.)*

X. Peng and H. Zhang are with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: pengx.gm@gmail.com; zhanghaixian@scu.edu.cn).

H. Zhu is with the Institute for Infocomm Research, A*STAR, Singapore 138632 (e-mail: zhuh@i2r.a-star.edu.sg).

J. Feng is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: elefjia@nus.edu.sg).

C. Shen is with the School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia (e-mail: chunhua.shen@adelaide.edu.au).

J. T. Zhou is with the Institute of High Performance Computing, A*STAR, Singapore 138632 (e-mail: joey.tianyi.zhou@gmail.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2958324

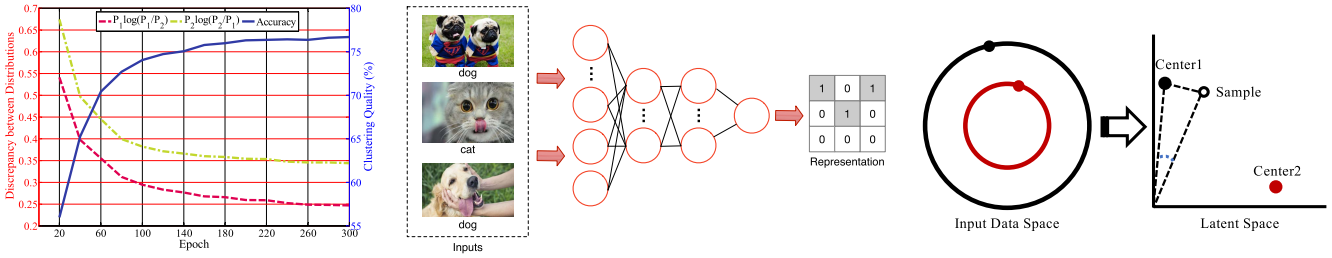

Fig. 1. Illustration of our basic idea. (a) Empirical observation on the prior of sample-assignment invariance. In the figure, the left y-axis denotes the difference between $\mathcal{P}_1$ and $\mathcal{P}_2$ and the right y-axis denotes the clustering performance in terms of Accuracy, where $\mathcal{P}_1$ and $\mathcal{P}_2$ are two clustering assignments in a 10-D space based on the Euclidean and Cosine distances. Specifically, we project the mnist raw data into a latent space using an encoder and then calculate $\mathcal{P}_1$, $\mathcal{P}_2$, and their discrepancy using the Euclidean distance, cosine distance, and the KL divergence loss, respectively. More details could refer to our experimental setting (see Section V). With more training epochs, Fig. 1(a) shows that: 1) the performance remarkably increases from 58% to 76% and 2) the discrepancy between $\mathcal{P}_1$ and $\mathcal{P}_2$ decreases monotonically. Better representation always gives a smaller distance between $\mathcal{P}_1$ and $\mathcal{P}_2$ and better clustering results. Note that the clustering assignment is obtained by directly performing k-means on the representation. Fig. 1(b) shows our insight that the "best" representation for clustering task is the label since the label is "the most ideal" representation. In other words, we propose that the label is the most desirable representation for recognition tasks, including clustering, because all within-cluster data points are represented by the corresponding label. (c) simple example to show that the key to achieve our idea is seeking a latent space in which different metrics will lead to the same cluster assignment. For example, in the figure, the sample will be assigned to Cluster#1 in terms of Euclidean and Cosine distance, i.e., the cluster assignments derived upon these two metrics will be identical. Note that, the red and black (inner and outer) circles denote two different clusters. (a) Empirical observation on the proposed invariance of sample assignment. (b) The insight of treating the label as a representation. Ideally, the "best" representation for recognition task should be the label. (c) The key is learning a latent space wherein the assignments with different metrics will converge to the same result.

from the following limitations. First, most of those methods learn data representation via shallow models, which may be unable to capture the complex latent structure of big data. Second, the methods required to access the whole data set and use it as the dictionary, thus causing difficulty in handling large-scale data sets. To address these challenges, we hold that deep neural networks could offer a promising solution due to its outstanding representative capacity and fast inference speed. In fact, [26] and [35]–[40] have very recently proposed to learn representation for data clustering using deep neural networks. However, most of these methods focus on learning representation and less attention is paid on clustering.

Like other unsupervised tasks, the key of clustering is seeking a suitable prior so that the data could be clustered into different categories without the help of human-labeled data. To achieve this key, we propose a novel trainable deep clustering method that embraces the end-to-end learning manner. The basic ideas of our method are in twofold, i.e., sample-assignment invariance prior and treating the label as a representation. The prior roots into our observation [see Fig. 1(a)]. More specifically, for a given data point $\mathbf{x}$, we obtain its representation using a parametric model, such as the neural network. With the learned representation $\mathbf{h}$, we compute two soft clustering assignments $\mathcal{P}_1(\mathbf{h}|\Omega)$ and $\mathcal{P}_2(\mathbf{h}|\Omega)$ using two different distance metrics, where Ω denotes the collection of cluster centers (could be initialized by k-means). To compute the discrepancy between $\mathcal{P}_1(\mathbf{h}|\Omega)$ and $\mathcal{P}_2(\mathbf{h}|\Omega)$, we propose a KL divergence-based loss function. With the increasing number of training epoch, we observe that the neural network learns a better representation in terms of clustering accuracy and a decreasing discrepancy between $\mathcal{P}_1(\mathbf{h}|\Omega)$ and $\mathcal{P}_2(\mathbf{h}|\Omega)$ in terms of the loss. The observation induces the so-called sample-assignment invariance prior. Namely, different distance metrics will give similar even the same cluster assignment on the manifold. In fact, such an observation/proposal is consistent with common sense. Taking the most ideal situation as an example, the "best" (invariant/distinct) representation for clustering/classification tasks should be a vector that has only one nonzero entry to indicate the index of the assigned cluster [see Fig. 1(b)]. Clearly, such a representation will lead to the same prediction assignment even though different metrics are used since the representation itself could be regarded as the predicted label. In other words, the key to our idea is learning a metric-invariant space, as shown in Fig. 1(c).

Based on the earlier observation, we propose a provable method to achieve the invariance of sample assignment. The proposed method is a deep neural network with a novel objective function, which enjoys an end-to-end pipeline. More specifically, the proposed method consists of two steps. The first step aims to learn representation, which is conducted to map inputs into a latent space in the forward pathway of our neural network. The second step implements data clustering in the backward pathway of our neural network, which simultaneously forward propagates a supervision signal to update the clustering membership and parametric transformations. With such a strategy, even no manual annotation is provided, our neural network can still be trained in an end-to-end manner and such a manner will lead to better representation and clustering results as shown in our experimental studies.

The major contribution of this article is summarized as follows.

1) To the best of our knowledge, this could be the first work to reveal the sample-assignment invariance prior and explicitly treat the label as a representation. Therefore, we assume that this article could provide a novel insight toward the community.
2) The proposed method is among the first end-to-end clustering models. Comparing with most existing subspace clustering methods, the proposed method jointly learns data representation and performs clustering, whereas the popular way is to treat them as two separate steps.

TABLE I
NOTATIONS

Notations	Definition
d	dimension of representation
m	dimension of input
n	data size
k	number of cluster centers
M	the number of hidden layers
$\mathbf{x}_i$	a data point
$\mathbf{X}$	a given data set
$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n]$	representation of $\mathbf{X}$
$\mathbf{\Omega} = [\omega_1, \omega_2, \cdots, \omega_k]$	collection of cluster centers
$\mathcal{P}(\mathbf{h}_i \vert \omega_j)$	probability of $\mathbf{h}_i$ belonging to ω_i
$\mathbf{W}^{(i)}$	weight of the i-th layer
$\mathbf{b}^{(i)}$	bias of the i-th layer
$\mathbf{\Theta} = \{\mathbf{W}^{(\mathbf{i})}, \mathbf{b}^{(\mathbf{i})}\}_{i=1}^{M}$	the parametric neural network

3) We prove that the proposed KL divergence loss could achieve the invariance of sample assignment, which is used as a new prior to learn data representation and perform clustering in an unsupervised way.

Organization and Notations: This article is organized as follows. In Section II, we briefly introduce some related works on subspace clustering and deep learning. In Section III, we elaborate the proposed method by introducing the representation learning module, clustering module, and implementation details. In Section IV, we prove that the proposed objective function could achieve the invariance of sample assignment in theory. In Sections V and VI, the experiments are conducted and the conclusion is given, respectively. In the following, we use lower case bold letters to represent column vectors and upper case bold ones to denote matrices. Table I summarizes some notations used throughout this article.

II. Related Works

A. Subspace Clustering

Recently, subspace clustering has shown significant developments in a variety of applications, such as segmentation, clustering, and so on [12]. Most of the existing approaches could be deemed as the variants of SC [6], which employ manifold learning to learn a data representation and then perform k-means on the representation to obtain the clustering membership. One major difference among these methods is their ways to construct the similarity graph to learn the representation. Mathematically, they build the graph using the reconstruction coefficient via

$$\min_{\mathbf{c}_i} \frac{1}{2}\|\mathbf{x}_i - \mathbf{X}\mathbf{c}_i\|_F^2 + \lambda \mathcal{R}(\mathbf{c}_i) \tag{1}$$

where $\mathbf{x}_i$ indicates the ith data point of $\mathbf{X}$, $\mathbf{c}_i$ is the self-expression coefficients of $\mathbf{x}_i$, $\mathcal{R}(\mathbf{c}_i)$ is the adopted prior on $\mathbf{c}_i$, and $\|\cdot\|_F$ denotes the Frobenius norm. Different methods employ different $\mathcal{R}(\cdot)$ and three of them are most popular, namely, ℓ_1-norm-based sparsity [13], [14], [27], [41]–[44], nuclear-norm-based low rankness [15], [16], [23], [45]–[49], and Frobenius norm-based methods [17], [20], [22], [25], [50].

Different from those algorithms, the proposed method learns the representations using a neural network rather than graph-regularized approaches. Such a difference brings several advantages to our method. First, the proposed method does not access the whole data set as a dictionary and solve a singular value decomposition (SVD) problem to obtain data representation. Second, our model jointly optimizes representation learning and data clustering. More specifically, a neural network maps data point $\mathbf{x}$ into a latent space to get representation $\mathbf{h}$, and clustering assignment is obtained by minimizing the discrepancy among different distributions (i.e., soft sample assignment) of $\mathbf{h}$ and cluster centers $\mathbf{\Omega}$ in terms of different distance metrics. In contrast, most existing subspace clustering methods treat these two steps separately. As our method utilizes clustering membership as a supervisor, a better representation could be learned. Third, the proposed neural network is a deep neural network, which could enjoy the more powerful capacity to capture the complex distribution of real-world data sets.

B. Deep Learning

During past years, deep neural networks have demonstrated promising performance in a variety of applications. However, the huge success of deep neural networks is mainly achieved in the setting of supervised learning [51], [52], and only a few works have been conducted in the unsupervised scenario. As one of the most important unsupervised learning tasks, clustering analysis has only attracted limited interests [26], [35], [37]–[39], [53], [54] to examine how to make it beneficial from neural networks.

Unlike those deep clustering approaches, the proposed method proposes treating the label as a representation, which may be beneficial to provide a novel insight into the community. Besides, some existing works achieve results in an off-the-shelf manner, which is different from the end-to-end manner adopted by our method. Extensive studies have proved that the task-specific end-to-end deep learning is more promising and attractive [55], [56]. Furthermore, this article is also different from [39] in the following aspects. First, the proposed method is generalized to different distance metrics. The generalized heterogeneous models could give greater diversity compared with the homogeneous model (HOMO), thus boosting the clustering performance especially when the data set is relatively large. Second, in this article, we prove that our objective function achieves the global optimality when the sample-centers probability distributions converge to the same point. In other words, this article shows that the proposed objective function is provable to achieve the invariance of sample assignment. Third, we provide the comprehensive experimental evaluations on the proposed method by comparing it with 16 competitive approaches and five data set to demonstrate the superiority of our proposed method.

III. Proposed Method

In this section, we first introduce how the proposed method achieves clustering in an end-to-end manner. To be specific, we will elaborate on the representation learning and clustering modules of the proposed method. After that, we will introduce the implementation details of our model.

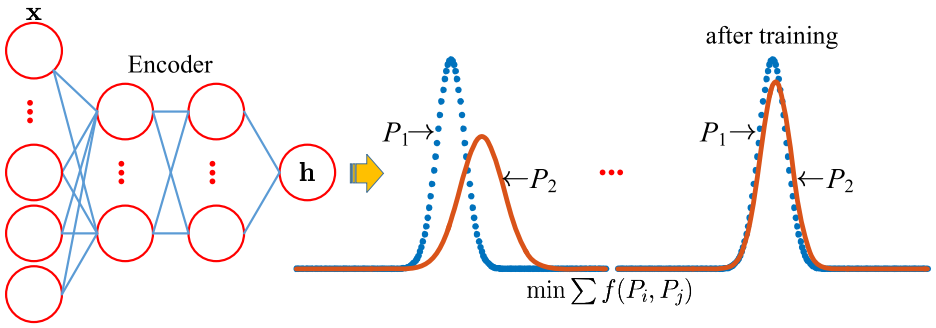

Fig. 2. Structure of our neural network. For the data point $\mathbf{x}$, our model works in an end-to-end manner to learn data representation and perform clustering in two continuing modules. Specifically, the first module (the left-hand side) is an encoder, which maps $\mathbf{x}$ into a latent space to get $\mathbf{h}$. The second module is to perform data clustering by minimizing the distance between different clustering memberships (or called distribution) of $\mathbf{h}$ with respect to cluster centers Ω. In the figure, $\mathcal{P}_1(\mathbf{h}|\Omega), \mathcal{P}_2(\mathbf{h}|\Omega), \ldots$ are two clustering memberships of $\mathbf{h}$ with respect to Ω regarding to two distance measurements, and the objective $f(\mathcal{P}_1, \mathcal{P}_2)$ (i.e., $\mathcal{P}_1(\mathbf{h}|\Omega), \mathcal{P}_2(\mathbf{h}|\Omega)$ for simplicity) indicates the distance between two distributions. $\sum f(\mathcal{P}_i, \mathcal{P}_j)$ is the sum of multiple pairwise discrepancy of distributions regarding to two distance metrics.

A. Representation Learning Model

Let $\mathbf{X} \in \mathbb{R}^{m\times n}$ be a given data set, and we aim at exclusively assigning the data point $\mathbf{x}_i \in \mathbf{X}$ into one of k clusters, where m is the dimension of input data and n denotes the number of data points. To ease of presentation, one could denote the cluster by the centroid $\omega_j \in \Omega$. To achieve the end, our method performs a clustering analysis with two joint modules. One is an encoding neural network that is used to learn data representation, and the other is a clustering layer that clusters data via minimizing the discrepancy between pairwise sample assignments for each data point. Fig. 2 shows the network architecture and basic idea of our method.

To learn the data representation $\mathbf{H} \in \mathbb{R}^{d\times n}$, our method employs a series of nonlinear transformations to progressively project $\mathbf{X}$ into a low-dimensional space, where d denotes the dimensionality of the latent space and the transformations are modeled by a collection of stacked neural components, such as the convolutional neural network (CNN) [57]. In this article, we adopt fully connected layers to construct our network because the experimental studies will show that such a simple network can remarkably outperform the well-established baseline methods.

To ease of presentation, we introduce our model with the simplest case, namely, one hidden-layer neural network as follows:

$$\mathbf{h}_i = g(\mathbf{x}_i|\Theta) = g(\mathbf{W}\mathbf{x}_i + \mathbf{b}) \tag{2}$$

where $\Theta = \{\mathbf{W}, \mathbf{b}\}$ indicates a parametric network with the weight $\mathbf{W} \in \mathcal{R}^{d\times m}$ and the bias $\mathbf{b} \in \mathcal{R}^{d}$. $g(\cdot)$ denotes the adopted nonlinear activation function. To initialize Θ, we take the self-supervised learning method [58]. To be precise, we first train an autoencoder via

$$\min_{\Theta} \|\mathbf{X} - \hat{\mathbf{X}}\|_F \tag{3}$$

where $\hat{\mathbf{X}}$ denotes the output of the autoencoder. After the autoencoder converging, the learned weights of the encoder, i.e., the first half of the hidden layers, are used as the initialization to our network.

B. Clustering Models

To perform clustering, we map the output of the encoder (i.e., $\mathbf{H}$) to the corresponding clusters by using HOMO (see Section III-B1) or heterogeneous model (see Section III-B2) that are based on the following KL divergence-based objective function:

$$\min_{\Theta,\Omega} \sum_{i,j} \mathcal{P}_j(\mathbf{H}|\Omega) \log \frac{\mathcal{P}_j(\mathbf{H}|\Omega)}{\mathcal{P}_i(\mathbf{H}|\Omega)} \tag{4}$$

where $\mathcal{P}_i$ and $\mathcal{P}_j$ are two conditional distributions (probability maps or soft assignments) of $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$ with respect to $\Omega = [\omega_1, \omega_2, \ldots, \omega_k]$ with respect to two distance measurements, where Ω denotes the cluster centroids. Our loss function is designed to obtain invariance of sample assignment by minimizing the distance between $\mathcal{P}_i$ and $\mathcal{P}_j$, which will be further analyzed in Theorem 1.

As the KL divergence loss is asymmetrical, one generally treats $\mathcal{P}_i$ and $\mathcal{P}_j$ as the predicted and the target distribution, respectively. In consequence, we could obtain the predicted label using the index of the maximal entry of $\mathcal{P}_i$. In Section V, we will provide some discussions and experimental analysis on this choice.

Considering the simplest case of (4), i.e., only two assignments are considered, and we have the following objective:

$$\min_{\Theta,\Omega} \ \mathcal{P}_2 \log \frac{\mathcal{P}_2}{\mathcal{P}_1} \tag{5}$$

where $\mathcal{P}_1$ and $\mathcal{P}_2$ correspond to $\mathcal{P}_i$ and $\mathcal{P}_j$ in (4), respectively.

Let $\mathcal{P}(\mathbf{h}_i|\omega_j)$ be an element of $\mathcal{P}(\mathbf{H}|\Omega)$, i.e., the probability of $\mathbf{h}_i$ belonging to ω_j, and then, we have the following definition:

$$\mathcal{P}(\mathbf{h}_i|\omega_j) = \frac{\mathcal{Q}^r(\mathbf{h}_i|\omega_j)/f_j}{\sum_j \mathcal{Q}^r(\mathbf{h}_i|\omega_j)/f_j} \tag{6}$$

where $r = \{1, 2, \ldots\}$ is used to raise $\mathcal{Q}$ to the rth power, $\mathcal{Q}(\mathbf{h}_i|\omega_j)$ denotes the closeness between $\mathbf{h}_i$ and ω_j, and f_j denotes the frequency of each cluster, which is adopted to normalize the loss contribution itself so that distorting the hidden space by larger clusters is prevented. In this article,

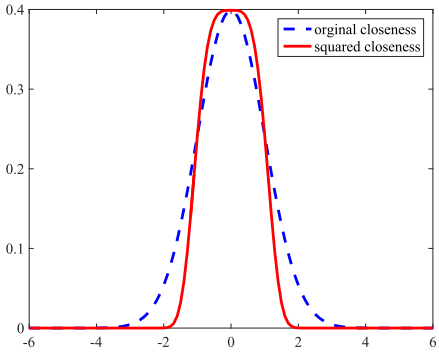

Fig. 3. Illustration of the effectiveness of squared closeness. The dashed curve is the closeness, and the solid curve is the squared closeness.

the squared closeness (i.e., $r = 2$) is adopted because it simultaneously suppresses the responses from dissimilar points and enhances the responses from similar points, which makes the result more discriminative and sparser, as shown in Fig. 3.

Different choices in distance metric will lead to different variants of our method. In general, there are two cases, i.e., homogeneous and heterogeneous models. In short, HOMO defines $\mathcal{P}_1(\mathbf{H}|\Omega)$ and $\mathcal{P}_2(\mathbf{H}|\Omega)$ using the same distance metric. Alternatively, the heterogeneous model employs different metrics to compute the conditional distribution between $\mathbf{H}$ and Ω.

1) HOMO: The idea behind the HOMO is that the same distance metric will also lead to different conditional probability distributions with different formulations. Such a model may be successful in handling the data set that is of small variance since the perturbation derived from homogeneous distributions is large enough to achieve the invariance of sample assignment on the manifold. To implement the HOMO, we define the predicted distribution using the Euclidean distance as follows:

$$\mathcal{Q}_1(\mathbf{h}_i|\omega_j) = \max(0, \mu_i - z_{ij}) \tag{7}$$

where z_{ij} denotes the Euclidean distance between the ith data point and the jth cluster centroid, which is computed by $z_{ij} = \|\mathbf{h}_i - \omega_j\|_2$. μ_i is the mean of $\mathbf{z}_i$. The above-mentioned formulation has two advantages. On the one hand, it transforms the Euclidean distance-based dissimilarity into similarity. On the other hand, it could guarantee the sparsity of the probability distribution. Regarding $\mathcal{Q}_2$, the definition is given by

$$\mathcal{Q}_2(\mathbf{h}_i|\omega_j) = \frac{\left(1 + z_{ij}^2\right)^{-1}}{\sum_{j'}\left(1 + z_{ij'}^2\right)^{-1}} \tag{8}$$

where z_{ij} is defined earlier. The equation plays two roles. On the one hand, it transforms the dissimilarity to similarity. On the other hand, it normalizes the obtained similarity into the range of $[0, 1]$. Moreover, the addition of 1 is used to avoid trivial solutions.

2) Heterogeneous Models: Compared with the HOMO, the heterogeneous models might lead to greater diversity due to a bigger difference between the used two distance metrics. Such diversity may give a better performance when the data set is complex. In our implementations, we adopt the formulation (7) to define $\mathcal{Q}_1(\mathbf{h}_i|\omega_j)$ and employ three popular distance metrics to calculate $\mathcal{Q}_2(\mathbf{h}_i|\omega_j)$.

The first formulation of $\mathcal{Q}_2$ is with the cosine distance. Formally, the corresponding z_{ij} is given by

$$z_{ij} = \frac{\mathbf{h}_i \cdot \omega_j}{\|\mathbf{h}_i\|_2 \|\omega_j\|_2} \tag{9}$$

where $\cdot$ denotes the dot product.

The other two formulations of $\mathcal{Q}_2$ are based on the correlation distance and the cityblock distance, respectively. Mathematically

$$z_{ij} = \frac{(\mathbf{h}_i - \bar{\mathbf{h}}_i) \cdot (\omega_j - \bar{\omega}_j)}{\|(\mathbf{h}_i - \bar{\mathbf{h}}_i)\|_2 \|(\omega_j - \bar{\omega}_j)\|_2} \tag{10}$$

and

$$z_{ij} = \|\mathbf{h}_i - \omega_j\|_1 \tag{11}$$

where $\bar{\mathbf{h}}_i$ and $\bar{\omega}_i$ denote the mean of $\mathbf{h}_i$ and ω_i, respectively. Note that the correlation distance [see (10)] will degrade to the cosine distance if the data are with zero mean and normalized to have the unit two norm.

Algorithm 1 Clustering via Cross-Metrics Verification

Input: A given data set $\mathbf{X}$.

// Initialization:

Initialize $\Theta = \{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^{M}$ by training an autoencoder and Ω with k-means, where Θ is the encoding parameters of the autoencoder.

// Training

while *not converged* **do**

  Randomly select a data point $\mathbf{x}_i$,

  // Forward propagation

  Compute $\{\mathcal{P}_1, \mathcal{P}_2\}$ as in Eqn.(6)–(11).

  // Backward propagation

  Update $\{\Theta, \Omega\}$ by using SGD to minimize the objective function in Eqn.(5)

end

// Inference:

Obtain the cluster assignment of $\mathbf{x}_i$ via

$$\mathcal{I}_i = \arg\max_j(\mathcal{P}_1(\mathbf{h}_i|\omega_j)) \tag{12}$$

Output: $\{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^{M}$ and clustering results.

Algorithm 1 summarizes the proposed method. In our implementation, two convergence conditions are considered. If one of these is satisfied, our model is considered in achieving convergence. The first condition is the widely used maximum training epoch and the second one is the ratio of inconsistence sample assignments between two continuous training epochs. More specifically, we assume that the algorithm converges if the updating model cannot lead to a significant change in prediction. In our experiment, we fix this number to 0.03%.

C. Implementation Details

The proposed neural network jointly optimizes $\Theta = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=1}^{M}$ and the cluster centers $\Omega = [\omega_1, \omega_2, \ldots, \omega_k]$

using the stochastic subgradient descent (SGD) with weights decay and momentum.

To obtain a good initialization, we first train a denoising autoencoder (DAE) [58], where the noise ratio, the momentum, and the weight decay rate are fixed to 0.3, 0.9, and 10^{-6}, respectively. The used autoencoder is a nine-layer neural network of which the number of neurons is m-500-500-2 000-d-2 000-500-500-m from the input layer to the output layer, where m and d denote the dimensionality of input and feature space, respectively. As in [59], we use the rectified linear units (ReLu) as the activation function. After the above-mentioned autoencoder converging, the weights of the first four hidden layers are used to initialize Θ and k-means is performed to initialize cluster centers $\mathbf{\Omega}$.

Note that the proposed method could also be based on other neural networks, including but not limited to restricted Boltzmann machine (RBM) and CNN. We adopt a fully connected network instead of others due to the following two reasons. First, compared with other neural networks, we experimentally found that a fully connected network is more easily tuned due to fewer user-specified parameters, in particular, in the scenario of clustering. In fact, most existing deep clustering methods [26], [35], [36], [38] are based on fully connected networks. Second, fully connected networks could handle different kinds of data, e.g., documents and images. In contrast, the network, such as CNN, mainly achieves success in handling visual data.

IV. Theoretical Results and Computational Complexity Analysis

As the aforementioned, our basic idea is that the conditional probability distribution (i.e., sample assignment) between data points $\mathbf{H}$ and cluster centers $\mathbf{\Omega}$ is invariant to different distance measurements on the manifold, which could be learned by a parametric model. In this section, we theoretically show that the proposed objective [see (5)] is well-established to implement our idea. Namely, our objective could achieve the sample-assignment invariance.

Theorem 1: For any data point $\mathbf{h} \in \mathbf{H}$ and $\omega \in \mathbf{\Omega}$, the global optimality of the objective (5) is achieved with the minimizer of $\mathcal{P}_1(\mathbf{h}|\omega) = \mathcal{P}_2(\mathbf{h}|\omega)$. At that point, the loss achieves the value 0.

Proof: Equation (5) can be rewritten as

$$\sum_{\mathbf{x},\omega} \mathcal{P}_2(\mathbf{x}|\omega)\log\mathcal{P}_2(\mathbf{x}|\omega) - \mathcal{P}_2(\mathbf{x}|\omega)\log\mathcal{P}_1(\mathbf{x}|\omega) = \mathcal{H}(\mathcal{P}_1,\mathcal{P}_2) - \mathcal{H}(\mathcal{P}_2) \quad (13)$$

where $\mathcal{H}(\mathcal{P}_1,\mathcal{P}_2)$ denotes the cross entropy of $\mathcal{P}_1$ and $\mathcal{P}_2$, and $\mathcal{H}(\mathcal{P}_2)$ is the entropy of $\mathcal{P}_2$.

Clearly, $\mathcal{H}(\mathcal{P}_1,\mathcal{P}_2) = \mathcal{H}(\mathcal{P}_2)$ gives the global optimality (5) with 0. According to the definition of cross entropy, one could conclude that the minimizer

$$\mathcal{P}_1(\mathbf{H}|\mathbf{\Omega}) = \mathcal{P}_2(\mathbf{H}|\mathbf{\Omega}) \quad (14)$$

gives

$$\mathcal{H}(\mathcal{P}_1,\mathcal{P}_2) = \mathcal{H}(\mathcal{P}_2) \quad (15)$$

as desired.

Theorem 1 demonstrates that (5) achieves the minimum when the distribution $\mathcal{P}_1$ is the same as $\mathcal{P}_2$. In other words, the proposed objective function could achieve the invariance of sample assignment as claimed. Note that although the above-mentioned result corresponds to the simplified version of our model, the conclusion could be easily extended to the setting of multiple distributions [see (4)] due to the property of the summation operator.

Note that the above-mentioned theoretical analysis does not prove the sample-assignment invariance. Instead, it shows that our objective could achieve the invariance. In fact, it is a daunting task to directly prove the sample-assignment invariance prior in theory. Fortunately, we find that the theoretical guarantee could be obtained by proving that there exists a latent space in which all samples could be partitioned into the corrected clusters regardless of the used distance metric. In other words, we only need to prove that there exists a model which could approximate any complex distribution so that the cluster assignment of learned representations keeps unchanged with respect to the used distance metric. With the universal approximation theorem [60], [61], ones have known that neural networks could be the desirable model as proved in extensive theoretical and experimental studies. In other words, the universal approximation theorem could provide a theoretical guarantee toward the proposed sample-assignment invariance. As the theorem is an open issue in the community, we would like to remain it in future exploration.

A. Computational Complexity

Suppose that our method consists of M layers and n_i denotes the number of neurons at the ith layer, and then, the time complexity for training is $\sum_{i=1}^{M-1} t n_i n_{i+1}$, where t is the iteration number. Note that the complexities for the feed- and backpropagation are the same.

V. Experimental Results

In this section, we carry out experiments by comparing our method with 16 state-of-the-art clustering methods. For comprehensive investigations, we use four different performance metrics to evaluate the clustering quality.

A. Experiment Settings

Regarding the proposed method, we implement it using a modular neural networks library, i.e., Keras [62] based on Theano [63]. For the tested algorithms, we obtain the source codes from authors' websites. The experiments are carried out on a machine with a 24x Intel Xeon CPU, 64Gb memory, and a Titan X GPU.

1) Baseline Algorithms: We compare our neural networks with 16 clustering methods, including k-means, nonnegative matrix factorization with locality preservation (NMF-LP) [64], zeta function-based agglomerative clustering (ZAC) [8], agglomerative clustering with average linkage (ACAL) [1], agglomerative clustering with weighted linkage (ACWL) [1], standard SC [6], LRR [16], low-rank subspace clustering (LRSC) [45], SSC [14], least square regression with/without

TABLE II

TUNED PARAMETERS OF OUR NETWORK. l IS THE LEARNING RATE WHICH IS DIVIDED BY de FOR EVERY ep EPOCHS UNTIL THE NETWORK ACHIEVES CONVERGENCE OR THE OPERATION IS REPEATED BY $re-1$ TIMES. bs DENOTES THE BATCH SIZE

data sets	d	lr	de	ep	re	bs
mnist-full	10	10^{-5}	0.9	300	15	256
mnist-test	10	10^{-5}	0.9	300	10	256
reuters	10	1.0	0.9	500	6	256
cifar10	100	10^{-4}	0.9	300	15	256
cifar100	10	1.0	0.95	-	-	256

diagonal constraint (LSR1/LSR2) [17], smooth representation clustering (SMR) [20], large-scale SC (LSC-R/LSC-K) [65], and deep embedding clustering (DEC) [36]. Moreover, we also investigate the performance of our model without backpropagation, i.e., the DAE with k-means (DAE+k-means). For all tested method, we tune their parameters for different data sets and then report their best performance.[1] For our method, we only tune the parameters of the optimizer to achieve convergence. The used parameters of our model are shown in Table II. Note that the used parameters are slightly different from that used in our conference work. In this article, we use the same rather than different values of ep for different data sets to alleviate the effort for parameters' selection. Moreover, we optimize our method using the Adadelta algorithm [66] instead of SGD on the cifar100 data set for investigating the efficacy of alternative optimizers. Regarding to our method, four variants are proposed according to Section III-B, which are HOMO, Heterogeneous model with the cOsinE (HOE), Heterogeneous model with the cOrrelaTion (HOT), and Heterogeneous model with the cITyBlock distance (HIT).

2) Data Sets: Our experiments are conducted on five data sets, including full mnist [57] (denote by mnist-full), the test partition of mnist (denote by mnist-test), the testing subset of cifar10 [67], a subset of reuters [68], and the first subset of cifar100 [67]. mnist-full and mnist-test consist of 70 000 and 10 000 images that are sampled from ten handwritten digits and each image is with the size of 28×28. The cifar10 testing partition consists of 10 000 images, which distributes over ten objects and the resolution of images is 32×32. The used reuters corpus consists of 10 000 documents, which samples from four root categories. In the experiments, each document is represented as a TF-IDF vector of 2000 most frequent words. The used cifar100 data set consists of 3000 color images from the first superclass (i.e., aquatic mammals). For these testing data sets, no preprocessing steps are conducted except decentralization.

3) Evaluation Criteria: Typical clustering approaches usually formulate the objective function by achieving high intracluster similarity (samples from the same cluster are similar) and low intercluster similarity (samples from different clusters are dissimilar). Such an internal criterion is proposed to improve the clustering quality. In practice, however, good scores on the internal criterion do not necessarily give the desired result. Therefore, an alternative way is directly evaluating the application of interest by utilizing the label information. Based on this observation, some evaluation metrics have been studied [69]. In our experiments, four of most popular evaluation metrics are adopted, namely, Accuracy (ACC), adjusted rand index (ARI), normalized mutual information (NMI) [70], and Precision. Note that, Accuracy and Precision have to use the ground truth to align the prediction and then compute their consist number. Different from Accuracy and Precision, NMI and ARI do not depend on the assigned prediction. More specifically, NMI is designed based on information theory, which trades off the quality of the clustering against the number of clusters to avoid the bias caused by the unbalanced label distribution. ARI views the clustering processing as a series of decisions, which evaluates on a pairwise basis if pathways are incorrectly grouped. These four metrics evaluate the performance of our method from different perspectives.

[1]ZAC (K, a, z), LRR (λ), LRSC (λ), LSR1 and LSR2 (λ), NMF-LP (α), SC (α), SMR (α, ϵ), SSC (λ, ϵ), LSC (K), and DEC (τ).

B. Comparisons With State-of-the-Art Methods

In this section, we compare the proposed algorithm with some recent clustering algorithms on five data sets in Tables III–V, from which one could see that the following holds.

1) The proposed method gives obvious improvements compared with 16 clustering algorithms on the used five data set. In terms of Accuracy, the performance gaps between our method and the best baseline approach are +7.50%, +9.53%, +1.20%, +3.39%, and +1.03% on the used five data sets.
2) In accordance with the other three performance metrics, the proposed method is also the best algorithm in most cases. For example, it is +0.9% in NMI, +4.17% in ARI, and +3.82% in Precision higher than the other algorithms on mnist-full.
3) The comparisons between DAE+k-means and our method show the effectiveness of the proposed sample-assignment invariance prior since our method will degrade to DAE+k-means without enforcing the prior. On the mnist-full data set, for example, the performance gap of our method over DAE+k-means is +7.50%, +4.29%, +7.36%, and +6.56% regarding those four metrics.
4) Comparing with the HOMO, the heterogeneous models (HOE, HOT, and HIT) usually perform slightly better, especially, on small-scale data sets. By using mnist-full and mnist-test as showcases, HOE outperforms HOMO by +0.84% with respect to Accuracy on the former data set, whereas the performance gap is just about +0.04% on the second data set. The potential reason is that the large-scale data set is more diverse than the small-scale one, thus better embracing the diversity derived from different metrics.

Besides the performance with fully connected autoencoder as aforementioned, we also investigate the performance of our method by collaborating with a convolutional autoencoder in Table III [denoted by HOE (CNN)]. More specifically, the used convolutional encoder is a six-layer network that

TABLE III

PERFORMANCE COMPARISONS WITH 16 CLUSTERING APPROACHES. PARS REPORTS THE TUNED PARAMETERS FOR THE EVALUATED ALGORITHMS. RESULTS IN BOLDFACE ARE SIGNIFICANTLY BETTER THAN THE OTHERS, ACCORDING TO THE T-TEST WITH A SIGNIFICANCE LEVEL AT 0.05

Data Set	mnist-full					mnist-test				
Methods	ACC	NMI	ARI	Precision	Pars	ACC	NMI	ARI	Precision	Pars
k-means	55.27	52.74	40.28	44.99	-	56.07	53.58	41.54	46.41	-
NMF-LP	48.85	42.63	29.89	36.39	7.0	66.48	46.40	39.84	59.99	5
ZAC	60.00	65.47	54.07	44.35	20,0.95,0.02	60.16	66.01	54.30	44.42	60,0.95,0.01
ACAL	11.77	0.51	0.02	10.04	-	12.20	0.81	0.00	10.03	-
ACWL	30.83	22.31	11.59	17.03	-	41.58	38.96	26.18	28.46	-
LRR	11.07	0.43	0.03	10.01	0.01	59.27	59.16	46.29	47.07	0.01
LRSC	13.67	0.98	0.26	10.16	0.05	60.21	53.90	43.14	47.12	0.03
LSR1	40.42	31.51	21.35	28.60	0.4	43.81	30.74	21.92	29.14	0.8
LSR2	41.43	30.03	20.00	30.03	0.1	41.70	30.40	20.00	30.03	0.6
SC	71.28	73.18	62.18	62.58	1.0	69.33	**70.97**	59.75	60.43	1.0
SMR	22.89	35.74	9.78	41.81	2^{-14},10^{-3}	67.59	41.33	36.23	56.35	2^{-16},0.01
SSC	67.65	69.37	58.61	60.36	0.01,0.01	60.96	64.65	50.79	50.28	0.01,0.01
LSC-R	59.64	56.68	45.98	50.09	6	59.27	56.84	45.75	49.52	7
LSC-K	72.07	69.88	60.81	62.94	6	70.35	68.98	58.89	60.80	6
DEC	83.65	73.60	70.10	72.81	10	63.12	60.51	50.22	54.29	640
DAE+k-means	79.66	71.21	66.91	69.87	-	72.70	61.43	55.34	58.95	-
HOMO	86.32	74.41	72.42	74.82	-	82.19	67.96	64.95	67.84	-
HOE	**87.16**	**75.50**	**74.27**	**76.43**	-	82.18	67.95	64.93	67.82	-
HOT	86.30	74.38	72.39	74.79	-	**82.23**	69.00	**65.01**	**67.89**	-
HIT	86.24	74.28	72.28	74.69	-	82.20	67.97	64.97	67.85	-
HOE (CNN)	**93.25**	**86.39**	**82.14**	**83.59**	-	**91.03**	**78.38**	**76.05**	**77.13**	-

TABLE IV

PERFORMANCE COMPARISONS WITH 16 CLUSTERING APPROACHES. PARS REPORTS THE TUNED PARAMETERS FOR THE EVALUATED ALGORITHMS. RESULTS IN BOLDFACE ARE SIGNIFICANTLY BETTER THAN THE OTHERS, ACCORDING TO THE T-TEST WITH A SIGNIFICANCE LEVEL AT 0.05

Data Set	cifar10					reuters				
Methods	ACC	NMI	ARI	Precision	Pars	ACC	NMI	ARI	Precision	Pars
k-means	19.88	6.39	3.09	12.67	-	54.01	34.91	27.79	45.51	-
NMF-LP	17.97	5.10	2.58	12.26	10	66.48	34.40	39.84	59.99	5
ZAC	10.14	0.17	0.00	9.99	20,0.95,0.01	43.66	1.11	0.71	31.38	20,0.95,0.01
ACAL	10.90	0.51	0.03	10.00	-	43.98	0.16	0.06	31.48	-
ACWL	15.79	3.62	1.87	11.01	-	42.17	0.75	0.65	30.90	-
LRR	11.07	0.43	0.03	10.01	0.01	53.94	33.42	27.35	45.54	0.1
LRSC	20.79	6.31	3.81	12.23	0.05	64.26	32.33	32.01	51.97	10^{-4}
LSR1	20.29	6.15	3.64	12.99	0.7	67.31	34.65	34.79	57.99	1.0
LSR2	20.08	6.17	3.65	12.01	0.4	67.15	34.43	34.41	57.36	0.7
SC	20.18	6.73	3.33	12.83	10	66.33	33.91	30.40	48.69	50
SMR	20.76	6.29	3.81	12.13	2^{-16},0.1	67.59	34.33	36.23	56.35	2^{-16},0.02
SSC	19.82	6.10	3.25	12.84	0.01,0.01	43.22	0.21	0.07	31.15	0.01,0.01
LSC-R	20.17	6.48	3.20	12.71	5	64.24	32.33	34.64	56.17	6
LSC-K	20.29	6.69	3.27	12.80	8	53.66	35.61	27.73	47.66	8
DEC	20.80	5.73	3.84	12.84	10	66.68	34.73	42.03	60.53	20
DAE+k-means	20.86	6.95	3.67	12.94	-	63.96	35.81	38.63	61.52	-
HOMO	22.00	6.99	3.91	13.33	-	69.45	34.24	42.73	61.78	-
HOE	**22.06**	7.02	3.93	13.34	-	69.81	34.33	42.59	62.02	-
HOT	21.77	7.02	3.90	13.30	-	**70.98**	**36.01**	**44.26**	**62.06**	-
HIT	21.85	**7.21**	**3.98**	**13.36**	-	69.49	35.08	43.80	61.70	-

is with conv(64, 5)-pool(2)-conv(32, 5)-pool(2)-FCL(1024)-FCL(10), where "conv (64, 5)" denotes a convolutional layer with the filter size of 64 and the kernel size of 5, "pool(5)" denotes max-pooling operation with the kernel size of 2, and "FCL(1024)" is a fully connected layer with 1024 neurons. The decoder is symmetric to the encoder. Similar to fully connected autoencoder, ReLu is used as the activation function for all layers except the last one of encoder and decoder that adopts the sigmoid function. The experiments are conducted on mnist. From the result, one could observe that the performance of our method could be further improved with a more powerful network.

C. Influence of Parameters

One major challenge of deep neural networks is to find optimal parameter combination for a good performance. In this

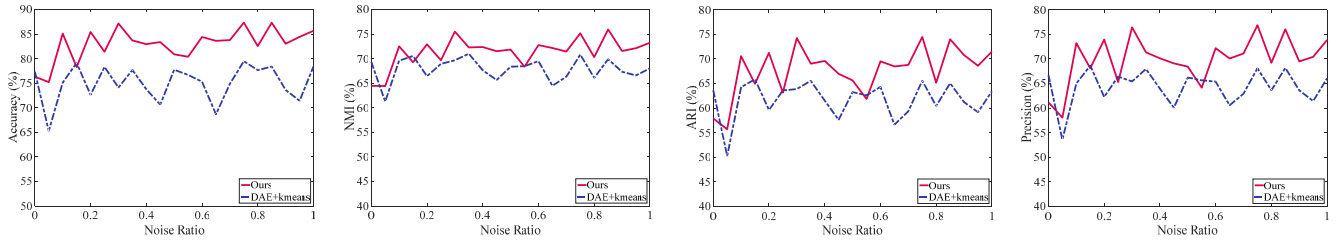

Fig. 4. Influence of noise ratio. (a) Accuracy. (b) NMI. (c) ARI. (d) Precision.

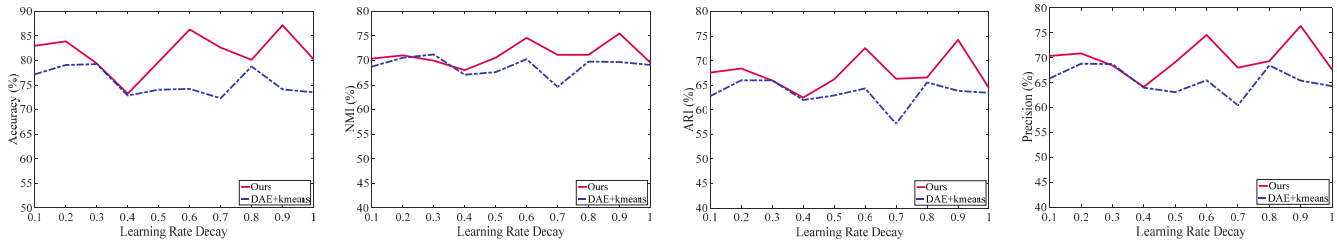

Fig. 5. Clustering performance versus varying learning rate decay of SGD. The learning rate will reduce by the decay ratio for every 300 epochs. (a) Accuracy. (b) NMI. (c) ARI. (d) Precision.

TABLE V
PERFORMANCE COMPARISONS ON THE CIFAR100 DATA SET

Methods	ACC	NMI	ARI	Precision	Pars
k-means	32.75	10.54	7.66	25.97	-
NMF-LP	31.30	9.43	6.76	25.29	8
ZAC	20.13	0.24	0.00	19.97	100,0.01
ACAL	20.97	0.63	0.03	19.99	-
ACWL	27.73	3.52	3.31	21.72	-
LRR	21.77	0.13	0.03	19.95	5
LRSC	28.33	4.18	3.23	22.52	1e-04
LSR1	21.93	0.17	0.01	19.98	0.1
LSR2	22.93	0.65	0.37	20.15	0.5
SC	33.17	11.27	8.32	26.02	50
SMR	26.80	3.63	2.24	21.70	2^{-16},0.1
SSC	24.00	0.49	0.01	19.98	0.01,1
LSC-R	31.97	9.78	7.26	25.54	1
LSC-K	32.36	9.95	7.56	25.76	2
DEC	33.93	11.07	8.51	26.77	20
DAE	34.10	12.55	10.74	23.12	-
HOMO	35.00	13.69	10.74	23.69	-
HOE	**35.93**	13.67	**11.50**	**28.71**	-
HOT	35.87	13.72	11.47	23.72	-
HIT	35.07	**13.85**	10.92	23.85	-

section, we investigate the influence of three parameters (i.e., noise ratio, learning rate decay, and feature dimension), which experimentally demonstrate the heaviest impact on the performance of our method. For the simplification, we carry out experiments using the HOE. For each evaluation, we also report the performance of k-means with the DAE. We carry out experiments on the mnist-full data set. In this test, all parameters except the evaluated one are fixed.

To speed up convergence, we train a DAE by adding some corruptions into the original input [58]. Fig. 4 shows the performance of our neural network with increasing noise ratio. One can see that the proposed method is relatively robust to the change of noise ratio. More specifically, the Accuracy of our method changes in the range of 82%-87% when the ratio increases from 0.2 to 1.0.

The convergence largely depends on the choice of learning rate whose value is based on the initial value and learning rate decay. In the experiment, we investigate the influence of the learning rate decay. The result is shown in Fig. 5, from which one could see that this parameter is quite important to the convergence performance of our deep model. In particular, the Accuracy seems more sensitive to the value of this parameter than the other three metrics.

Besides the above-mentioned two parameters, we also examine the influence of the hyperparameter d, i.e., the dimension of latent space. Fig. 6 shows that both the proposed model and DAE+k-means usually give better clustering results with when $d = 10$. The reason may be that larger d could preserve more information, but a too large d will reduce the discrimination of the learned representation. Moreover, larger d will give a more significant improvement to our method compared with DAE+k-means.

D. Influence of Different Inference Approaches

In the most ideal situation, $\mathcal{P}_1$ and $\mathcal{P}_2$ could converge to the same point, as shown in Theorem 1. However, such a result is hard to achieve since the noise contained in the data set or nonsmooth data distribution will destroy the structure of manifold, as pointed out in [71]. As a consequence, there are two choices to obtain the clustering assignment, i.e., using the index of the maximal entry in $\mathcal{P}_1$ or $\mathcal{P}_2$. In the above-mentioned experimental studies, we perform inference only based on $\mathcal{P}_1$. Such a choice is derived from the definition of our KL divergence-based loss, i.e., $\mathcal{P}_1$ and $\mathcal{P}_2$ are treated as the predicted distribution and the ground truth, respectively.

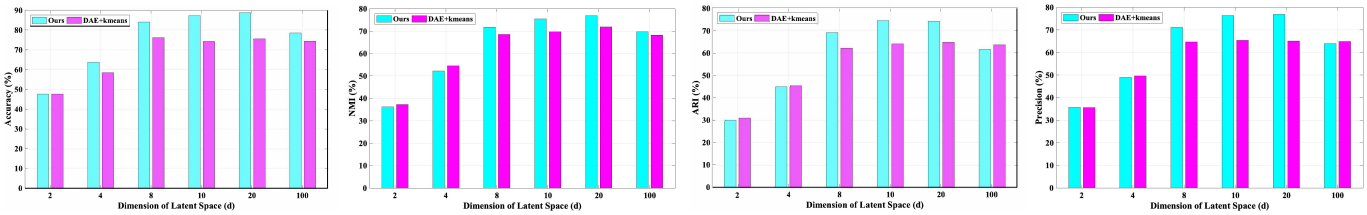

Fig. 6. Performance of our method with different feature dimensions. The baseline is DAE+kmeans. (a) Accuracy. (b) NMI. (c) ARI. (d) Precision.

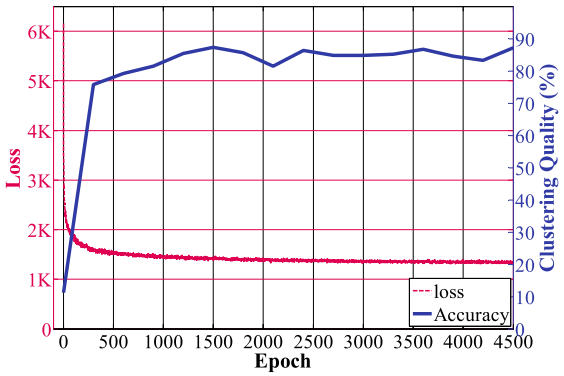

Fig. 7. Convergence curve of our neural network, where the x-axis is the training epoch, the left y-axis indicates the loss, and the right one is the corresponding clustering Accuracy.

TABLE VI

PERFORMANCE OF OUR METHOD WITH THE PREDICTION OF $\mathcal{P}_1$ AND $\mathcal{P}_2$. "PR" IS SHORT FOR PRECISION

Metrics	$\mathcal{P}_1$				$\mathcal{P}_2$			
	ACC	NMI	ARI	Pr	ACC	NMI	ARI	Pr
HOMO	86.32	74.41	72.42	74.82	82.92	72.58	66.85	67.53
HOE	87.16	75.50	74.27	76.43	81.10	70.40	66.15	66.03
HOT	86.30	74.38	72.39	74.79	84.81	72.48	69.64	71.62
HIT	86.24	74.28	72.28	74.69	76.91	61.93	57.43	60.48

For comprehensive studies, in this section, we consider another choice, i.e., computing clustering assignments using $\mathcal{P}_2$ instead of $\mathcal{P}_1$. The results are reported in Table VI, which show that $\mathcal{P}_1$-based prediction is superior to $\mathcal{P}_2$-based results. In general, it outperforms the counterpart by 3% in terms of Accuracy.

E. Convergence Analysis

In this section, we illustrate the convergence curve of our neural network on the full mnist data set. The result is demonstrated in Fig. 7, which shows that our model achieves convergence after 1500 training epochs. After that, its Accuracy ranges between 83% and 88%. Moreover, the time cost investigation shows that our method is quite computationally efficient. To be specific, it takes about 1.1 s to handle 70 000 samples for each training epoch. Note that as some of the tested methods such as SSC are carried out on CPU instead of GPU, we do not report the time cost of these baselines for fair comparisons.

VI. CONCLUSION

In this article, we proposed a novel prior and developed a new deep clustering method by minimizing the discrepancy between sample assignments with respect to multiple distance metrics. The proposed method employs a fully connected neural network to jointly learn a collection of hierarchical representation and cluster assignments in an end-to-end manner.

This article may have the following advantages. From the view of clustering, it provides a novel way to implement clustering by exploiting the prior of invariant sample assignment. We believe that the work may provide novel insights to the community. On the one hand, task-specified representation learning could be further unified by treating the label as a representation. On the other hand, the focus of unsupervised subspace clustering could benefit from such a unified framework.

There are some directions to improve this article. First, theoretical guidance in the selection of distance metrics could be explored in the future. In this article, we employ four popular distance metrics to design our method. Although experimental results demonstrate their effectiveness, the theoretical studies on the choice of metrics are still missing since model selection is an open challenging issue. Second, the KL divergence-based loss is not the only one choice to implement the invariance of sample assignment, new objective functions could be established based on other measurements of the probability distribution.

ACKNOWLEDGMENT

The authors would like to thank the anonymous associate editor and reviewers for their valuable comments and constructive suggestions to improve the quality of this article.

REFERENCES

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.
[2] T. Zhang, C. L. P. Chen, L. Chen, X. Xu, and B. Hu, "Design of highly nonlinear substitution boxes based on I-Ching operators," *IEEE Trans. Cybern*, vol. 48, no. 12, pp. 3349–3358, Dec. 2018.
[3] X. Lu, X. Zheng, and Y. Yuan, "Remote sensing scene classification by unsupervised representation learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5148–5157, Sep. 2017.
[4] T. Zhang, G. Su, C. Qing, X. Xu, B.Cai, and X. Xing, "Hierarchical lifelong learning by sharing representations and integrating hypothesis," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be publication.
[5] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "COMIC: Multi-view clustering without parameter selection," in *Proc. Int. Conf. Mach. Learn.*, Long Beach, CA, USA, vol. 97, Jun. 2019, pp. 5092–5101.
[6] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2002, pp. 849–856.
[7] C. Ding and T. Li, "Adaptive dimension reduction using discriminant analysis and K-means clustering," in *Proc. 24th Int. Conf. Mach. Learn.*, Corvalis, OR, USA, Jun. 2007, pp. 521–528.

[8] D. Zhao and X. Tang, "Cyclizing clusters via Zeta function of a graph," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2009, pp. 1953–1960.

[9] P. S. Bradley and O. L. Mangasarian, "K-plane clustering," *J. Global Optim.*, vol. 16, no. 1, pp. 23–32, 2000.

[10] P. Tseng, "Nearest q-flat to m points," *J. Optim. Theory Appl.*, vol. 105, no. 1, pp. 249–252, 2000.

[11] P. K. Agarwal and N. H. Mustafa, "*k*-means projective clustering," in *Proc. 23rd ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst.* 2004, pp. 155–165.

[12] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[13] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, Granada, Spain, 2011, pp. 55–63.

[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, Nov. 2013.

[15] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 663–670.

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

[17] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, De-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 347–360.

[18] F. Nie, H. Wang, H. Huang, and C. Ding, "Unsupervised and semi-supervised learning via *l*1-norm graph," in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 2268–2273.

[19] J. Feng, Z. Lin, H. Xu, and S. C. Yan, "Robust subspace segmentation with block-diagonal prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 3818–3825.

[20] H. Hu, Z. C. Lin, J. Feng, and J. Zhou, "Smooth representation clustering," in *Proc. 27th IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, Jun. 2014, pp. 3834–3841.

[21] Y.-X. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When LRR meets SSC," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, Dec. 2013, pp. 64–72.

[22] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *Proc. 29th AAAI Conf. Artif. Intell.*, Austin, TX, USA, Jan. 2015, pp. 3827–3833.

[23] S. Xiao, M. Tan, D. Xu, and Z. Y. Dong, "Robust Kernel low-rank representation," *IEEE Trans. Neural. Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2268–2281, Sep. 2015.

[24] S. Li, K. Li, and Y. Fu, "Temporal subspace clustering for human motion segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4453–4461.

[25] P. Ji, M. Salzmann, and H. Li, "Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 4687–4695.

[26] Z. Wang, Y. Yang, S. Chang, J. Li, S. Fong, and T. S. Huang, "A joint optimization framework of sparse coding and discriminative clustering," in *Proc. 24th Int. Conf. Artif. Intell.*, Jul. 2015, pp. 3932–3938.

[27] Y. Yang, J. Feng, N. Jojic, J. Yang, and T. S. Huang, "ℓ^0-sparse subspace clustering," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 731–747.

[28] C. Lu, J. Feng, Z. Lin, and S. Yan, "Correntropy induced L2 graph for robust subspace clustering," in *Proc. 14th IEEE Conf. Comput. Vis.*, Sydney, VIC, Australia, Dec. 2013, pp. 1345–1352.

[29] H. Jia and Y.-M. Cheung, "Subspace clustering of categorical and numerical data with an unknown number of clusters," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3308–3325, Aug. 2018.

[30] P. Zhou, Y. Hou, and J. Feng, "Deep adversarial subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1596–1604.

[31] S. Li and Y. Fu, "Unsupervised transfer learning via low-rank coding for image clustering," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 1795–1802.

[32] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

[33] C. Zhang, H. Fu, Q. Hu, P. Zhu, and X. Cao, "Flexible multi-view dimensionality co-reduction," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 648–659, Feb. 2017.

[34] U. V. Luxburg, O. Bousquet, and M. Belkin, "Limits of spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, Hyatt Regency, ON, Canada, 2005, pp. 857–864.

[35] X. Peng, S. Xiao, J. Feng, W.-Y. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, New York, NY, USA, Jul. 2016, pp. 1925–1931.

[36] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, New York, NY, USA, Jun. 2016, pp. 478–487.

[37] J. Yang, D. Parikh, and D. Batra, "Joint Unsupervised learning of deep representations and image clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5147–5156.

[38] Z. Wang, Q. Ling, and T. S. Huang, "Learning deep $\ell 0$ encoders," in *Proc. 30th AAAI Conf. Artif. Intell.*, Feb. 2016, pp. 2194–2200.

[39] X. Peng, J. Feng, J. Lu, W.-Y. Yau, and Z. Yi, "Cascade subspace clustering," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Feb. 2017, pp. 2478–2484.

[40] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.

[41] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 3928–3937.

[42] C. You, D. P. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 3918–3927.

[43] C. G. Li, C. You, and R. Vidal, "Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2988–3001, Jun. 2017.

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

[45] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognit. Lett.*, vol. 43, pp. 47–61, Jul. 2014.

[46] X. Zhang, F. Sun, G. Liu, and Y. Ma, "Fast low-rank subspace segmentation," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1293–1297, May 2014.

[47] R. Liu, Z. Lin, F. De la Torre, and Z. Su, "Fixed-rank representation for unsupervised visual learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 598–605.

[48] R. He, Y. Zhang, Z. Sun, and Q. Yin, "Robust subspace clustering with complex noise," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4001–4013, Nov. 2015.

[49] S. Li, K. Li, and Y. Fu, "Self-taught low-rank coding for visual learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 645–656, Mar. 2018.

[50] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2016.

[51] A. Krizhevsky, L. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, CA, USA, 2012, pp. 1097–1105.

[52] L. Shao, D. Wu, and X. Li, "Learning deep and wide: A spectral method for learning deep networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2303–2308, Dec. 2014.

[53] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards K-means-friendly spaces: Simultaneous deep learning and clustering," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 3861–3870.

[54] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montral, QC, Canada, Dec. 2017, pp. 24–33.

[55] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Proc. Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[56] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[57] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[58] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[59] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Stat.*, Fort Lauderdale, FL, USA, vol. 15, Aug. 2011, pp. 315–323.
[60] A. Pinkus, "Approximation theory of the MLP model in neural networks," *Acta Numer.*, vol. 8, pp. 143–195, Jan. 1999.
[61] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6231–6239.
[62] F. Chollet. (2015) *Keras*. [Online]. Available: https://github.com/fchollet/keras
[63] T. D. Team, "Theano: A python framework for fast computation of mathematical expressions," May 2016, *arXiv:1605.02688*. [Online]. Available: https://arxiv.org/abs/1605.02688
[64] D. Cai, X. He, X. Wang, H. Bao, and J. Han, "Locality preserving nonnegative matrix factorization," in *Proc. 21st Int. Jont Conf. Artif. Intell.* San Francisco, CA, USA: Morgan Kaufmann, 2009, pp. 1010–1015.
[65] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1669–1680, Aug. 2015.
[66] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *arXiv:1212.5701*, 2012. [Online]. Available: https://arxiv.org/abs/1212.5701
[67] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.
[68] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, Dec. 2004.
[69] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge Univ. Press, 2008.
[70] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.
[71] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

Xi Peng (S'10–M'14) received the Ph.D. degree in computer science from the Sichuan University, Chengdu, China, in 2013.

He currently is a National Distinguished Youth Professor with the College of Computer Science, Sichuan University.

Dr. Peng has served as an Associate Editor/Guest Editor for six journals, such as the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and an Area Chair/Session Chair/Program Chair/Tutorial Organization Chair for over 30 international conferences, such as the AAAI Conference on Artificial Intelligence (AAAI) and the European Conference on Computer Vision (ECCV).

Hongyuan Zhu (S'13–M'14) received the B.S. degree in software engineering from the University of Macau, Macau, China, in 2010, and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2014.

He is currently a Research Scientist with the Institute for Infocomm Research, A*STAR, Singapore. His research interests include multimedia content analysis and segmentation.

Jiashi Feng received the B.E. degree from University of Science and Technology of China, Hefei, China, in 2007, and the Ph.D. degree from the National University of Singapore, Singapore, in 2014.

He was a Post-Doctoral Researcher with the University of California from 2014 to 2015. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, National University of Singapore. His current research interests focus on machine learning and computer vision techniques for large-scale data analysis.

Chunhua Shen is currently a Professor of computer science with The University of Adelaide, Adelaide, SA, Australia.

Haixian Zhang (M'14) received the Ph.D. degree in applied mathematics from the University of Electronic Science and Technology of China, Chengdu, China, in 2010.

She is currently an Associate Professor with the College of Computer Science, Sichuan University, Chengdu. Her current research interests include neural networks and computer games.

Joey Tianyi Zhou received the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2015.

He is currently a Scientist with the Institute of High Performance Computing, A*STAR, Singapore. His current research interests include differentiable programming, transfer learning, and sparse coding.

Dr. Zhou was a recipient of the Best Poster Honorable Mention at the Asian Conference on Machine Learning (ACML) 2012, the Best Paper Nomination at the European Conference on Computer Vision (ECCV) 2016, and the NIPS 2017 Best Reviewer Award.