This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Toward Balance Deep Semisupervised Clustering

Yu Duan, Zhoumin Lu, Rong Wang, Xuelong Li, *Fellow, IEEE*, and Feiping Nie, *Senior Member, IEEE*

***Abstract*— The goal of balanced clustering is partitioning data into distinct groups of equal size. Previous studies have attempted to address this problem by designing balanced regularizers or utilizing conventional clustering methods. However, these methods often rely solely on classic methods, which limits their performance and primarily focuses on low-dimensional data. Although neural networks exhibit effective performance on high-dimensional datasets, they struggle to effectively leverage prior knowledge for clustering with a balanced tendency. To overcome the above limitations, we propose deep semisupervised balanced clustering, which simultaneously learns clustering and generates balance-favorable representations. Our model is based on the autoencoder paradigm incorporating a semisupervised module. Specifically, we introduce a balance-oriented clustering loss and incorporate pairwise constraints into the penalty term as a pluggable module using the Lagrangian multiplier method. Theoretically, we ensure that the proposed model maintains a balanced orientation and provides a comprehensive optimization process. Empirically, we conducted extensive experiments on four datasets to demonstrate significant improvements in clustering performance and balanced measurements. Our code is available at https://github.com/DuannYu/BalancedSemi-TNNLS.**

***Index Terms*— Balanced clustering, deep clustering, Lagrangian multipliers, pairwise information.**

I. Introduction

CLUSTERING, a classic and widely employed technique in unsupervised learning, is employed to uncover the underlying structure of data. For instance, in the realm of e-commerce, clustering enables marketers to extract valuable information from customers, thereby offering essential support to enterprise managers for decision-making [1]. Clustering the categories, characteristics, and genes of diverse plants and animals can yield valuable insights into the identification of distinct species [2]. Additionally, it assumes a crucial role in image retrieval, computer vision, recommendation systems, and other related domains.

In recent years, numerous classical clustering methods have been proposed, including k-means, spectral clustering, affinity propagation [3], nonnegative matrix factorization [4], Gaussian mixture [5], subspace clustering [6], and so on [7], [8].Additionally, various extended clustering tasks, such as semisupervised clustering (SSC) and multiview clustering, have garnered significant attention from researchers. For multiview clustering, most of the existing state-of-the-art concentrate on effectively integrating energy or information derived from multiple modal spaces to achieve superior performance compared to single-modal counterparts. Wang [9] presents a comprehensive overview of existing advancements in multimodal data analytics, spanning from shallow to deep spaces. For example, iterative views agreement (IVA) [10], a multiview clustering method, can well encode the local data manifold structure from each view-dependent feature space, and achieving the multiview agreement via an iterative fashion, while better preserving the flexible nonlinear manifold structure from all views. Extensive experiments conducted on real-world multiview datasets have validated its superiority. However, due to the exponential growth in data scale and dimensions, the aforementioned methods struggle to demonstrate satisfactory performance. Consequently, deep clustering has garnered researchers' attention according to its strong feature extraction capability and batchwise progress. Generally, deep clustering comprises two main components: representation learning and clustering alignment. For the former, researchers have proposed plenty of network structures to capture semantic information, such as autoencoders [11], [12], [13], variational autoencoders [14], generative adversarial networks [15], [16], graph neural networks [17], and many others. Based on the above excellent feature extraction ways, a large number of deep clustering methods have been proposed by considering the clustering alignment strategy. Zhang et al. [18] proposed a robust embedded deep k-means clustering (RED-KC) method. Peng et al. [19] proposed a new subspace clustering method, sparse-first deep subspace clustering. Ji et al. [20] proposed a new network structure for subspace clustering. Shaham et al. [21] proposed that SpectralNet map the input into the feature space of their association graph Laplacian and then obtain clustering results.

In numerous real-world scenarios, there is an increasing demand for balanced clustering. For instance, in the domain of wireless sensor networks (WSNs), maintaining an equal number of sensors in each cluster is crucial for reducing energy consumption [22]. Similarly, in image retrieval, it is important to achieve balanced groups of retrieved images. Additionally, in classroom grouping and personnel assignment, the goal is to maintain an equal number of members in each group. In the field of distributed data management, storing related data in the same cluster and ensuring an equal data scale in each node is crucial for minimizing data transmission overhead during queries.

Manuscript received 31 October 2022; revised 6 October 2023; accepted 1 December 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101902; in part by the Natural Science Basic Research Program of Shaanxi under Grant 2021JM-071; in part by the National Natural Science Foundation of China under Grant 62176212, Grant 61936014, and Grant 61772427; and in part by the Fundamental Research Funds for the Central Universities under Grant G2019KY0501. *(Corresponding author: Feiping Nie.)*

The authors are with the School of Computer Science, the School of Artificial Intelligence, Optics and Electronics (iOPEN), and the Key Laboratory of Intelligent Interaction and Applications (Ministry of Industry and Information Technology), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: duanyuee@gmail.com; walker.zhoumin.lu@gmail.com; wangrong07@tsinghua.org.cn; li@nwpu.edu.cn; feipingnie@gmail.com).

Digital Object Identifier 10.1109/TNNLS.2023.3339680

2162-237X © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

The purpose of balanced clustering is to achieve approximately equal cluster sizes [23]. Broadly speaking, balanced clustering methods can be categorized into two main types: hard-balanced clustering and soft-balanced clustering. Hard-balanced clustering aims to impose strict limitations on the cluster sizes. In contrast, soft-balanced clustering does not enforce such strict limitations but tends toward balance. We will discuss both of them in Section II-A in detail.

In many clustering tasks, it is often feasible to acquire preexisting knowledge about the data, rather than relying solely on unsupervised information as mentioned earlier. Integrating even a small amount of this prior knowledge into the clustering task can significantly enhance performance. Among various methods, SSC [24] is considered a promising approach that has found applications in diverse fields In general, this prior information can exist in various forms, such as having access to certain data labels or having pairwise constraints (must-link and cannot-link). When labeled data is available, researchers commonly employ label propagation to extend multiple labels to unlabeled instances [25]. In the case of pairwise constraints, a common way is utilizing this constraint information to guide the loss function, bringing similar data closer together in the embedding space and pushing dissimilar data further apart [26]. Recently, researchers have focused on generalized SSC, encompassing weakly supervised learning, few-shot learning [27], [28], [29], and learning with noisy or partial labels [30], [31], [32]. Moreover, novel category discovery (NCD) [33] and generalized category discovery (GCD) [34] have further extended SSC to make it more applicable to challenging open-world scenarios.

In summary, the majority of current balanced clustering methods are graph-based. However, when dealing with large-scale data, these methods often suffer from high computational time and space complexity. Additionally, due to the limited extracted representation, they fail to effectively capture the meaningful features of high-dimensional data, resulting in suboptimal performance. Moreover, semisupervised information often serves as a regularizer that cannot directly impose constraints on specific samples. Also, directly imposing constraints on samples often results in nondifferentiable, making it challenging to train them jointly with neural networks. Therefore, jointly training networks with semisupervised constraints is a challenging task.

To address the aforementioned issues, we propose a deep semisupervised balanced clustering model. It comprises three main components: a reconstruction module, a balanced clustering module, and a semisupervised module. Initially, the reconstruction module is employed to acquire the low-dimensional representation of the original data. Subsequently, by integrating the balanced clustering module and the semisupervised module, we obtain a cluster indicator matrix that achieves both clarity and balance. The model is trained jointly until convergence is reached.

Specifically, the reconstruction module is primarily implemented using an autoencoder, and the encoder is responsible for extracting the low-dimensional representations. To achieve clustering with a balanced tendency, we design a novel loss function to meet this objective. Finally, to incorporate semisupervised information into neural network training, we introduce a differentiable semisupervised module that employs the Lagrangian multipliers method to convert pairwise constraints into penalty terms for each violation. These penalties are then integrated into the objective function, and the networks are jointly trained to satisfy the imposed constraints. In sum, the main contributions of this article are summarized as follows.

1) We investigate the problem of balanced clustering with partially known pairwise constraints in the context of representation learning for large-scale and high-dimensional problems. Consequently, we propose a deep semisupervised balanced clustering model that aims to achieve balanced partitions by leveraging both labeled and unlabeled features concurrently. To the best of our knowledge, this particular setting has not yet been investigated.
2) We introduce a novel loss function for balanced clustering that guides the model to achieve a high-quality clustering performance while preserving a balanced tendency.
3) We develop a differentiable module that effectively leverages semisupervised information. This module can be seamlessly integrated into any neural network and jointly optimized using backpropagation.
4) Extensive experiments conducted on commonly used benchmark datasets demonstrate that our proposed method consistently outperforms other models. To better understand the proposed model, we conducted comprehensive investigations encompassing convergence analysis, parameter analysis, ablation experiments, and more.

The rest of this article is organized as follows. Section II briefly reviews the related work. We discuss the proposed model and its optimization in detail in Section III. Experimental results are reported in Section IV. Finally, we conclude this article in Section V.

II. Related Works

A. Balanced Clustering

Given a dataset with balanced distribution, balanced clustering aims to group the data into different clusters, whose scales are almost the same. Balanced clustering can be broadly categorized into two types: hard-balanced clustering and soft-balanced clustering. In the case of hard-balanced clustering, each cluster is compelled to have an equal size. For example, Bradley et al. [35] imposed a lower bound on the size of each cluster to prevent the generation of small clusters. Malinen and Fränti [36] proposed an improvement to the aforementioned approach by strictly assigning data points to each cluster. As observed, hard-balanced clustering strives to achieve strict balance by enforcing clusters of equal sizes for each class. However, achieving perfect balance in practice is nearly impossible. Conversely, researchers often favor a partitioning approach that demonstrates a tendency toward balance.

In contrast to hard-balanced clustering, soft-balanced clustering treats the size of clusters as a flexible constraint and does

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

not require strict balance. Soft-balanced clustering permits samples to be assigned to multiple clusters, while hard-balanced clustering forces each sample to belong to only one cluster. This flexibility empowers soft-balanced clustering to effectively handle ambiguity and uncertainty in the data. In real-world datasets, certain samples may exhibit relationships with multiple clusters due to noise, overlap, or the presence of boundary samples. Moreover, soft-balanced clustering offers information on the membership degree or weight of each sample with respect to each cluster, enabling us to quantify the relationship between individual samples and clusters. For example, Chen et al. [37] first proposed a self-balanced min-cut (SBMC), in which a scalar s is learned to balance the partition across all clusters. Based on SBMC, Chen et al. [38] further proposed enhanced balanced min-cut (EBMC) to solve the problem that the scalar s in SBMC cannot capture the differences among different clusters. Recently, Wu et al. [39] proposed an exponential regularization to achieve a balanced tendency, called Exp-Cut. In this article, we mainly focus on soft-balanced clustering, because it is more suitable for realistic scenarios.

B. Semisupervised Clustering

Instead of relying solely on completely unknown data, it may be possible to obtain the label information for some samples. In contrast to the supervised clustering scenario, the key challenge in SSC is how to effectively utilize this limited amount of available information. Previous studies have primarily employed the mechanism of label propagation. For example, Deng and Yu [40] proposed a method that used estimation of the class proportion of data to enhance the discriminative power of the learned smooth classification function on the graph. Zhang et al. [41] proposed a new dual-constrained deep semisupervised coupled factorization network (DS2CF-Net) for learning hierarchical representations. Wang et al. [42] proposed a method that can propagate labeled information and learn a structured graph simultaneously. Nie et al. [43] directly represent the cannot-link information into the graph by using a well-designed graph regularization. For nonlinear data, some deep SSC has also been proposed, they use some brand-new methods to map the data to a common space [44], [45], [46].

Moreover, some partially aligned multiview clustering can also be regarded as variants of SSC [47], [48]. Although these above methods have made great progress in clustering performance, they often need to construct a pairwise affinity matrix to exploit and propagate semisupervised information, leading to high computational complexity and space storage.

C. Deep Clustering

The core idea of deep clustering is using neural networks to obtain low-dimensional representations from original data, and then clustering. Compared with conventional dimensional reduction algorithms, such as PCA [49], [50], kernel method [51], [52], and spectral clustering [53], deep clustering can obtain more meaningful representations and perform better.

Autoencoder [54], as one of the most classic models of the neural network, is widely used in clustering tasks. In 2016, Xie et al. [55] proposed deep embedding clustering (DEC) based on an autoencoder, which greatly improved performance. Due to that DEC could destroy local features, in the following year, Guo et al. [56] proposed improved DEC (IDEC) based on DEC to handle this defect. Subsequently, a large number of researchers successively proposed plenty of variant clustering models. For example, Guo et al. [57] proposed a clustering model based on convolutional autoencoders, called deep convolutional embedded clustering (DCEC). Jiang et al. [58] proposed variational deep embedding (VaDE) which is based on variational autoencoders. Zhang et al. [59] used a mixture of autoencoders clustering, fusing multiple autoencoders to better extract representations to clustering. Cai et al. [60] added sparse constraints to the autoencoder and proposed deep stack sparse embedded clustering (DSSEC).

III. Deep Semisupervised Balanced Clustering

In this section, we present a novel deep semisupervised balanced clustering approach that leverages a unified framework to obtain meaningful and clustering-favorable representations with a balanced tendency. First, we briefly introduce a deep semisupervised balanced clustering model. Then we present how to tackle this problem with a detailed description.

A. Framework Overview

The overall framework of our proposed model for learning balanced and clustering-favorable representations is illustrated in Fig. 1. It mainly consists of three parts: a reconstruction module, a balanced clustering module, and a semisupervised module. The first two modules aim to learn a representation with balanced clustering assignments, and the semisupervised module is to guide assignments to satisfy the pairwise constraint. Specifically, our method employs a novel balanced loss as a replacement for the Kullback-Leibler divergence (KL) utilized in the wild-spread DEC [55]. This replacement enables the direct acquisition of representations that possess both clustering features and balance. Simultaneously, we incorporate a differentiable semisupervised module to relax the pairwise constraints of must-link and cannot-link. The neural network and semisupervised modules are alternately optimized to obtain the final clustering assignments.

B. Proposed Method

Formally, let $\mathcal{E}$, $\mathcal{D}$, and $\mathcal{A}$ denote the encoder, decoder, and cluster assignment, respectively. Given a dataset with n samples, we obtain a hidden embedding representation using the encoder $\boldsymbol{Z} = \mathcal{E}(\boldsymbol{X})$, and the decoder $\mathcal{D}$ generates a reconstructed output $\hat{\boldsymbol{X}} = \mathcal{D}(\boldsymbol{Z})$. The cluster assignment function $\mathcal{A}$ performs a soft assignment of the data, resulting in $\boldsymbol{Y} = \mathcal{A}(\boldsymbol{Z})$. The overall objective function of the method is

$$\begin{aligned} &\mathcal{L} = \mathcal{L}_r + \alpha \mathcal{L}_{\text{bal}} \\ &\text{s.t. } \boldsymbol{Y} \in \mathcal{M}, \mathcal{C} \end{aligned} \tag{1}$$

where α is a tradeoff parameter, $\mathcal{M}$ to denote the must-link constraint, and $\mathcal{C}$ to denote the cannot-link constraint. The proposed model comprises three main components: reconstruction

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

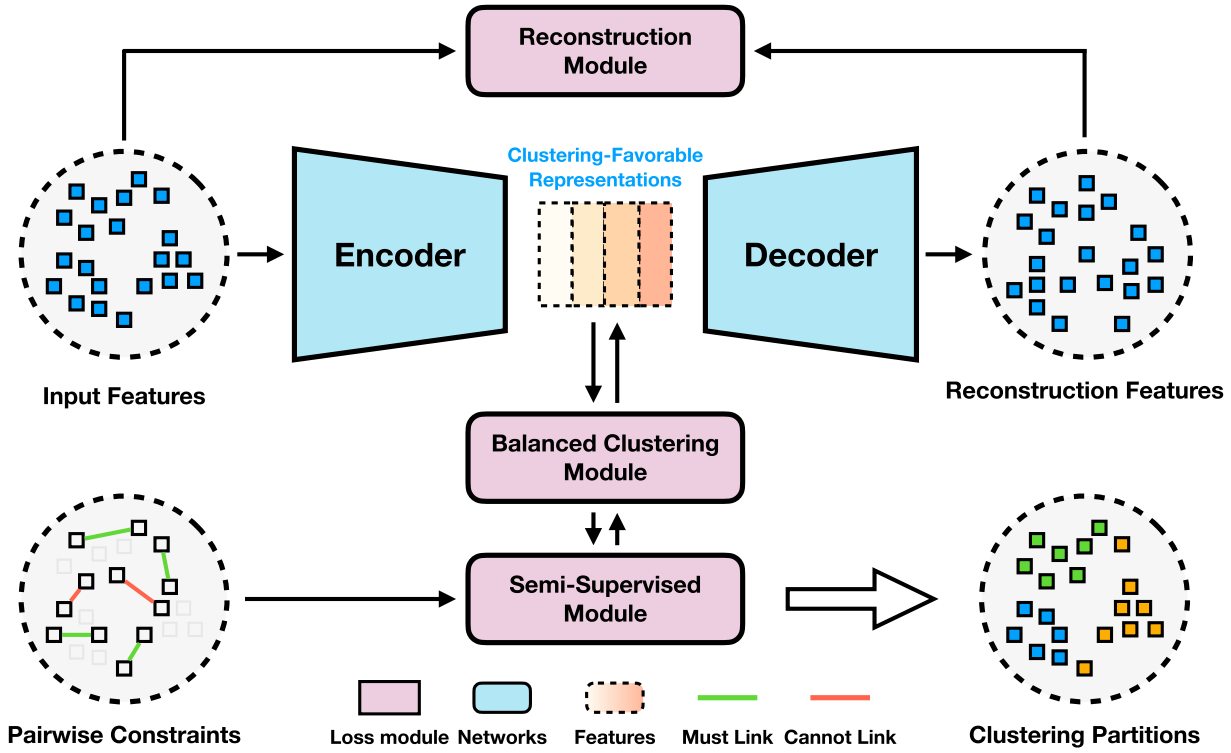

Fig. 1. Pipeline of the proposed deep semisupervised balanced clustering framework. The inputs are original training data $\boldsymbol{X}$ and pairwise constraints, the outputs are clustering partitions with balanced tendency. The whole training process contains two parts: 1) train encoder $\mathcal{E}$ and decoder $\mathcal{D}$ according to the loss $\mathcal{L}$ from three well-designed modules and 2) update Lagrangian multipliers in the semisupervised module. Steps (1) and (2) are trained alternatively until convergence.

loss $\mathcal{L}_r$, balance loss $\mathcal{L}_{\text{bal}}$, and semisupervised constraint. Sections III-B1–III-B3 are dedicated to discussing these components separately.

1) Reconstruction Loss and Clustering Assignment: Let $\boldsymbol{X} \in R^{n\times d}$ denote a dataset, where d represents the data dimension, our objective is to identify an effective encoder $\mathcal{E}$ that produces an embeddings $\boldsymbol{Z} = \mathcal{E}(\boldsymbol{X})$ that is well-suited for the clustering task. Simultaneously, we utilize the decoder $\mathcal{D}$ to reconstruct the original samples, resulting in $\hat{\boldsymbol{X}} = \mathcal{D}(\boldsymbol{Z})$. Then, the reconstruction loss $\mathcal{L}_r$ is measured by mean squared error (mse)

$$\mathcal{L}_r = ||\boldsymbol{X} - \hat{\boldsymbol{X}}||_F^2. \tag{2}$$

Denote $\boldsymbol{Z}_i$ is ith sample of hidden representation $\boldsymbol{Z}$, $\mathcal{C}_j, \forall j = \{1, 2, \ldots, k\}$ is the jth cluster center, where k is the cluster number. Then we follow [55] to use student t-distribution to measure the confidence between representation $\boldsymbol{Z}_i$ and cluster center $\boldsymbol{C}_i$, which can be calculated as

$$p_{ij} = \frac{\left(1 + ||\boldsymbol{Z}_i - \boldsymbol{C}_j||_2^2/\sigma\right)^{-\frac{\sigma+1}{2}}}{\sum_{j'}\left(1 + ||\boldsymbol{Z}_i - \boldsymbol{C}_{j'}||_2^2/\sigma\right)^{-\frac{\sigma+1}{2}}}. \tag{3}$$

Finally, we use the following formula to measure the cluster assignment between the ith sample and the jth center:

$$y_{ij} = \frac{p_{ij}^2/\sum_i p_{ij}}{\sum_j\left(p_{ij}^2/\sum_i p_{ij}\right)}. \tag{4}$$

2) Balanced Loss: The reconstruction and assignment term mentioned above allows for obtaining soft assignments of the data. Previous works, such as DEC [55] and IDEC [56], commonly utilize the KL divergence between p_{ij} and y_{ij} as the loss function for training the autoencoder. However, this self-training strategy often fails to achieve satisfactory performance. It is even impossible to get a balanced tendency (we will show the results in Section IV). Specifically, in this article, we expect that the clustering results exhibit distinct indicators with balanced trends within each category. From this perspective, we first introduce a theorem as follows.

Theorem 1: Suppose $\boldsymbol{Y} \in \{0, 1\}^{n\times c}$ is a cluster indicator matrix, define $||\boldsymbol{Y}||_h = \sum_{j=1}^{c}(\sum_{i=1}^{n} y_{ij}^2)^{-1}$, $||\boldsymbol{Y}||_h$ arrives its minimum when $\sum_{i=1}^{n} y_{ij}$ equals to (n/c) if (n/c) is an integer, or $\{\lfloor(n/c)\rfloor, \lceil(n/c)\rceil\}$ otherwise.

Proof: Let $\boldsymbol{u} \in \mathcal{R}^{c\times 1}$ be a column vector where $u_j = \sum_{i=1}^{n} y_{ij}^2$ and $\sum_{j=1}^{c} u_j = n$. Without loss of generality, we assume that $u_1 \geq u_2 \cdots \geq u_c$ then $(1/u_1) \leq (1/u_2) \cdots \leq (1/u_c)$. So, by Cauchy–Schwarz inequality, we have

$$\begin{aligned} c^2 &= c\left(u_1\frac{1}{u_1} + u_2\frac{1}{u_2} + \ldots + u_c\frac{1}{u_c}\right) \\ &\leq \left(\sum_{j=1}^{c} u_j\right)\left(\sum_{j=1}^{c}\frac{1}{u_j}\right) \end{aligned} \tag{5}$$

Then, we can obtain

$$\frac{c^2}{n} \leq \sum_{j=1}^{c}\frac{1}{u_j}. \tag{6}$$

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Therefore, $||\boldsymbol{Y}||_h$ arrives its minimum when $\sum_{i=1}^{n} y_{ij}^2 = (n/c), \forall j = \{1, \ldots, c\}$. If (n/c) is not an integer, we can verify that $||\boldsymbol{Y}||_h$ arrives its minimum when $\sum_{i=1}^{n} y_{ij}^2 = \{\lfloor (n/c) \rfloor, \lceil (n/c) \rceil\}, \forall j = \{1, \ldots, c\}$ otherwise. ■

Here, we only use $||\boldsymbol{Y}||_h$ as a constraint[1] to get balanced tendency and avoid trivial solution with one isolated object as the cluster.

Inspired by Theorem 1, we propose the balance loss as follows:

$$\mathcal{L}_{\text{bal}} = ||\mathcal{A} \circ \mathcal{E}(\boldsymbol{X})||_h \tag{7}$$

where $\circ$ denotes the function composition. When $||\boldsymbol{Y}||_h$ achieves its minimum, it must be a clear indicator matrix with balanced properties.

3) Semisupervised Loss: Before discussing semisupervised loss in detail, we first give a simple but intuitive observation. Define $\boldsymbol{Y}_i, \forall i = \{1, 2, \ldots, n\}$ as the ith row of $\boldsymbol{Y}$, for any pair of $\boldsymbol{Y}_i$ and $\boldsymbol{Y}_j$, they all satisfy the following constraints:

$$\begin{cases} \langle \boldsymbol{Y}_i, \boldsymbol{Y}_j \rangle = 1, & \forall (i, j) \in \mathcal{M} \\ \langle \boldsymbol{Y}_i, \boldsymbol{Y}_j \rangle = 0, & \forall (i, j) \in \mathcal{C} \end{cases} \tag{8}$$

where $\langle \cdot, \cdot \rangle$ denotes the matrix inner product. Here, we rewrite the original objective function with constraints as follows:

$$\begin{aligned} &\mathcal{L} = \mathcal{L}_r + \alpha \mathcal{L}_{\text{bal}} \\ &\text{s.t. } \boldsymbol{Y} \in \mathcal{M}, \mathcal{C}. \end{aligned} \tag{9}$$

Since the pairwise constraints can be formally expressed, we can use the Lagrangian multiplier method to write the constraints as penalty terms and add them into (9) to train the whole network.

According to (8), for any pairs $\boldsymbol{Y}_i$ and $\boldsymbol{Y}_j$ violating the must-link, their penalty loss can be defined as

$$\text{cost}_{\text{ML}} = 1 - \langle \boldsymbol{Y}_i, \boldsymbol{Y}_j \rangle. \tag{10}$$

Similarly, for any point pair $\boldsymbol{Y}_i$ and $\boldsymbol{Y}_j$ that violate the cannot-link, we can define their penalty loss as

$$\text{cost}_{\text{CL}} = \langle \boldsymbol{Y}_i, \boldsymbol{Y}_j \rangle. \tag{11}$$

So, the Lagrangian objective function can be written by adding penalty terms corresponding to must-link and cannot-link violations to the original objective function. Formally, it can be written as

$$\mathcal{L}(\theta, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathcal{L}_r + \alpha \mathcal{L}_{\text{bal}} + \beta \mathcal{L}_{\text{semi}} \tag{12}$$

where θ is networks' parameter, β is a tradeoff hyperparameter and $\mathcal{L}_{\text{semi}}$ can be defined as

$$\mathcal{L}_{\text{semi}} = \frac{1}{2} \left[\sum_{(i,j) \in \mathcal{M}} \lambda_{(i,j)} \left(1 - \langle \boldsymbol{Y}_i, \boldsymbol{Y}_j \rangle\right) + \sum_{(i,j) \in \mathcal{C}} \mu_{(i,j)} \langle \boldsymbol{Y}_i, \boldsymbol{Y}_j \rangle \right] \tag{13}$$

where $\lambda, \boldsymbol{\mu} \geq 0$ are Lagrangian multiplier vectors (the lengths are equal to the number of their constraints, respectively) whose subscripts (i, j) are only used to index the corresponding elements. And $(1/2)$ mainly refers to normalization, because we calculated all the pairwise constraints twice.

[1] $||\boldsymbol{Y}||_h$ is just a marker for simplicity, not a conventional norm.

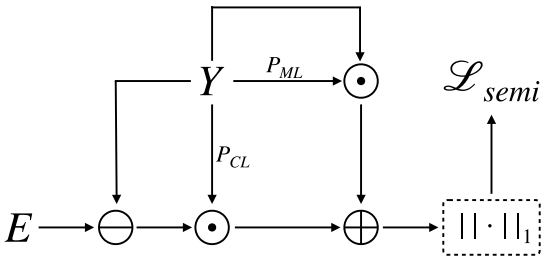

Fig. 2. Differentiable semisupervised module. $\oplus$, $\ominus$, and $\odot$ denote element-wise addition, subtraction and multiplication, respectively. Arrow with matrix means dot product.

For better readability and implementation at the code level, (13) can be converted into the matrix form

$$\mathcal{L}_{\text{semi}} = \frac{1}{2} \left\| \boldsymbol{P}_{\text{ML}} \odot \left(\boldsymbol{E} - \boldsymbol{Y}\boldsymbol{Y}^T\right) + \boldsymbol{P}_{\text{CL}} \odot \left(\boldsymbol{Y}\boldsymbol{Y}^T\right) \right\|_1 \tag{14}$$

where $\odot$ is the Hadamard product, $\boldsymbol{E} \in R^{n \times n}$ with all elements equals to 1, $\boldsymbol{P}_{\text{ML}}$, and $\boldsymbol{P}_{\text{CL}}$ represent the coefficients matrix for each pairwise constraint. Fig. 2 shows the whole pipeline of the semisupervised module. Specifically, if any pair (i, j) is involved in a must-link constraint, then the (i, j)th and the (j, i)th element of $\boldsymbol{P}_{\text{ML}}$ are equal to the corresponding Lagrangian multiplier for that constraint, namely, $\boldsymbol{P}_{\text{ML}}(i, j) = \boldsymbol{P}_{\text{ML}}(j, i) = \lambda_{(i,j)}$. Similarly, when there exists a cannot-link constraint between the ith and the jth samples, $\boldsymbol{P}_{\text{CL}}(i, j) = \boldsymbol{P}_{\text{CL}}(j, i) = \mu_{(i,j)}$.

C. Optimization Method

In this section, we provide a detailed explanation of the optimization procedure for minimizing the Lagrangian function. It can be observed that the network parameters and the semisupervised module are independent of one another. In other words, the parameters θ, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ are independent of each other. Therefore, we use an iterative optimization algorithm to update the parameters alternatively.

First, we employ gradient descent to directly train the network parameters. Second, by treating $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ as Lagrangian multipliers, we update them using the Lagrange dual ascent method. Formally, the dual function of (12) is defined as follows:

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\theta} \mathcal{L}(\theta, \boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{15}$$

Then, we can get the optimal $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ by maximizing the dual-function $g(\boldsymbol{\lambda}, \boldsymbol{\mu})$ which can be written as

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\mu}} \left\| \boldsymbol{P}_{\text{ML}} \odot \left(\boldsymbol{E} - \boldsymbol{Y}\boldsymbol{Y}^T\right) + \boldsymbol{P}_{\text{CL}} \odot \left(\boldsymbol{Y}\boldsymbol{Y}^T\right) \right\|_1. \tag{16}$$

Finally, $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ can be updated by using gradient ascent which can be written as

$$\begin{cases} \lambda_{(i,j)} \leftarrow \lambda_{(i,j)} + \gamma \cdot \left(1 - \langle \boldsymbol{Y}_i, \boldsymbol{Y}_j \rangle\right), & \forall (i, j) \in \mathcal{M} \\ \mu_{(i,j)} \leftarrow \mu_{(i,j)} + \gamma \cdot \langle \boldsymbol{Y}_i, \boldsymbol{Y}_j \rangle, & \forall (i, j) \in \mathcal{C} \end{cases} \tag{17}$$

where γ is the learning rate. Similarly, (17) can also be converted into the matrix form for better readability and implementation at the code level as follows:

$$\begin{cases} \boldsymbol{P}_{\text{ML}} \leftarrow \boldsymbol{P}_{\text{ML}} + \gamma \cdot \boldsymbol{P}_{\text{ML}} \odot (\boldsymbol{E} - \boldsymbol{Y}\boldsymbol{Y}^T), & \forall (i, j) \in \mathcal{M} \\ \boldsymbol{P}_{\text{CL}} \leftarrow \boldsymbol{P}_{\text{CL}} + \gamma \cdot \boldsymbol{P}_{\text{CL}} \odot \boldsymbol{Y}\boldsymbol{Y}^T, & \forall (i, j) \in \mathcal{C}. \end{cases} \tag{18}$$

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

In other words, as the alternating optimization process continues, the penalty coefficient gradually increases, promoting the relaxation term to approach zero and thereby satisfying the requirements of $\mathcal{M}$ and $\mathcal{C}$. Finally, we perform the reconstruction of balanced clustering and semisupervised learning using an alternative approach as described below.

1) *Step 1 (Warm Up):* Pass the data through the network, that is, $\mathcal{E}$ and $\mathcal{D}$ to pretrain the network only using reconstruction loss.
2) *Step 2 (Reconstruction and Balanced Clustering Learning):* Pass the data through the network, computing reconstruction data $\hat{\boldsymbol{X}}$ and clustering assignment $\boldsymbol{Y}$. Then, we calculate the reconstruction loss and balanced loss, updating network parameters via backpropagation.
3) *Step 3 (Semisupervised Learning):* Update all Langrange multiplier by (18).
4) *Step 4:* Repeat Steps 2 and 3 until convergence.

Once the network converges, we feed the entire dataset into the network and directly obtain the clustering assignments via a balanced clustering module without any extra k-means operations like the traditional fashion [61], [62], [63].

IV. Experimental Analysis

In this section, we conduct extensive experiments on four datasets. We compare our proposed algorithm with both balanced clustering and SSC methods that are considered state-of-the-art, aiming to verify its performance superiority and balance tendency. Finally, we perform a detailed analysis of our algorithm by exploring its factors.

A. Setup for Experiments

1) Implementation Details: Following the settings in IDEC [56], we employ a fully connected autoencoder for all datasets. The encoder $\mathcal{E}$ has a structure of d-500-500-2000-10, where d represents the dimension of the input data (features). The decoder $\mathcal{D}$ has the same mirrored structure as the encoder, with dimensions of 10-2000-500-500-d. ReLU [64] is used as the activation function between each layer in the entire network, except for the input, output, and embedding layers.

The clustering loss coefficient α is set to 0.1 for all datasets. The semisupervised loss coefficient β is set to 0.01 for MNIST and 0.1 for the remaining datasets. The hyperparameters mentioned above are determined through a grid search using the values $1e^{-4}$, $1e^{-3}$, $1e^{-2}$, $1e^{-1}$, $1e^{+0}$. We simply initial all Lagrangian multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ to 1 and batch size to 256 for all datasets. We set the initial learning rate $\text{lr} = 1e^{-3}$ for the Adam optimizer to update networks and $\gamma = 1$ in (18) to update Lagrangian multipliers. σ in Student's t-distribution as shown in (3) is set to 1. For all datasets, the number of iterations is set to 50. The number of constraints for all semisupervised models is set to 20% of the dataset size. Before training, we first generate pairwise constraints ($\mathcal{M}$ and $\mathcal{C}$) according to labels in each batch.

We experimentally find that excessively large Lagrangian multipliers degrade the performance. Consequently, we set an upper bound of 10 for the Lagrangian multipliers for all datasets. We consider the algorithm to have converged and stopped training when the change in the objective function becomes smaller than $1e^{-4}$.

TABLE I
Characteristics of the Whole Datasets

	# Instance	# Classes	# Dimensions
MNIST	70000	10	784
USPS	9298	10	256
REUTERS-10K	10000	4	2000
STL-10	13000	10	1024

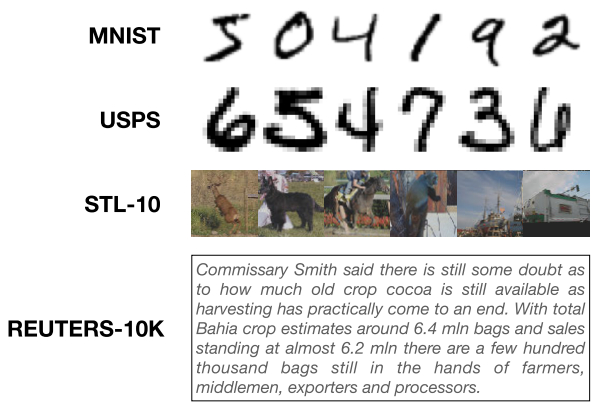

Fig. 3. Visual examples of all datasets.

2) Datasets: In the evaluation, we used four benchmark datasets. Fig. 3 shows the visual examples of datasets and their detailed characteristics which are summarized in Table I.

1) **MNIST**[2] [65] which consists of 70 000 hand-written digits of 28 × 28 pixel size. We expand each sample into a 784-D vector and then use max-normalization on it.
2) **USPS**[3] which is a 16 × 16 color gray-scale handwritten digit images with 9298 samples from ten classes. We also expand each sample into a 256-D vector and then use max-normalization on it.
3) **REUTERS-10K**[4] [66] which contains about 810 000 English news stories labeled with a category tree. Then, we follow [55] and [56] to sample a subset of 10 000 examples, called REUTERS-10K from four root categories, for comparison purposes.
4) **STL-10**[5] [67] is dataset of 96 × 96 color images consisting 13 000 samples from ten categories. We use Res-Net50 [68] to obtain 1024-D hidden embeddings as features to feed into our model. Our implementation is based on Python and PyTorch [69].

3) Metrics: We conduct four measurements in experiments, which are accuracy (ACC), normalized mutual information (NMI), adjusted rand index (ARI), and normalized entropy

[2]http://yann.lecun.com/exdb/mnist/
[3]https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/multiclass.html
[4]https://github.com/slim1017/VaDE/blob/master/dataset/reuters10k
[5]https://cs.stanford.edu/ acoates/stl10/

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

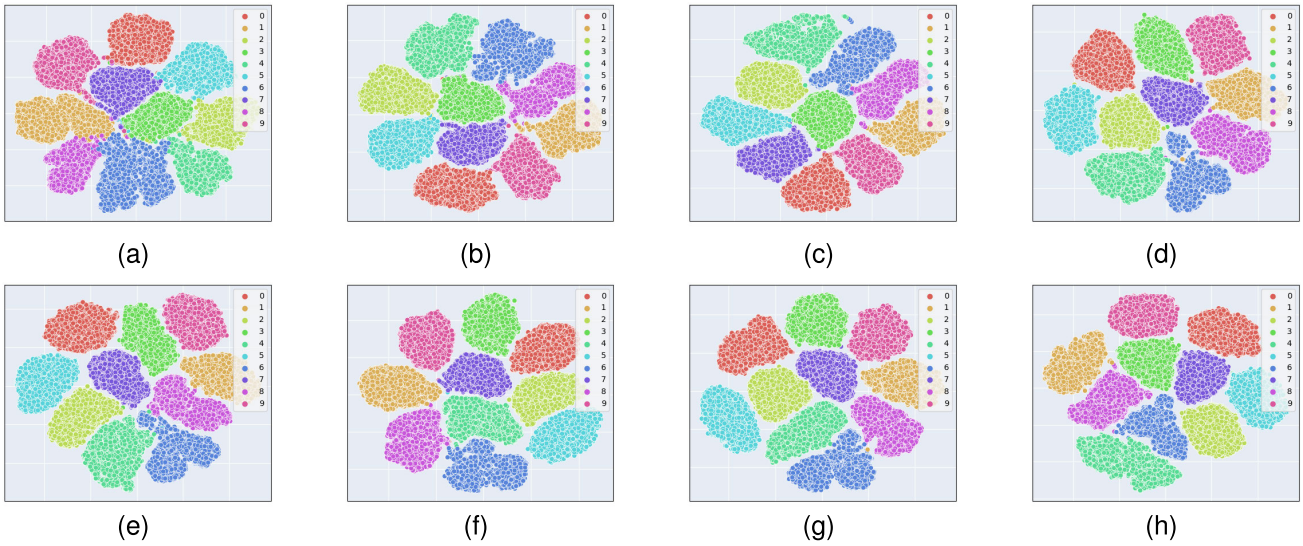

Fig. 4. Feature space visualization on MNIST. (a) Origin features. (b)–(h) Epochs from 5 to 35 with interval 5.

(NE). Specifically, ACC denotes the proportion of clustering samples taking over the whole samples which is computed as

$$\mathrm{ACC} = \frac{1}{n}\sum_{i=1}^{n}\delta\big(y_i, \mathrm{map}\big(\hat{y}_i\big)\big) \tag{19}$$

where y_i and $\hat{y}_i$ are ground truth and predicted labels evenly, map$(\cdot)$ is a function [70] that maps the cluster labels to their best real labels.

NMI is used to measure coherence between two distributions whose definition is as follows:

$$\mathrm{NMI}(Y, \hat{Y}) = \mathrm{MI}(Y, \hat{Y})/\sqrt{H(Y)H(\hat{Y})} \tag{20}$$

where $H(Y)$, $H(Y)H(\hat{Y})$ is entropies of distribution Y and $\hat{Y}$, and MI measures the coherence between them.

ARI is an improved version of the rand index (RI). It can be computed by

$$\mathrm{ARI} = \frac{\sum_i\sum_j C_{n_{ij}}^2 - \big(\sum_i C_{n_i^\tau}\cdot\sum_i C_{n_i^r}\big)/C_n^2}{\frac{1}{2}\big(\sum_i C_{n_i^\tau} + \sum_i C_{n_i^r}\big) - \big(\sum_i C_{n_i^\tau}\cdot\sum_i C_{n_i^r}\big)/C_n^2} \tag{21}$$

where n_i^τ denotes the number of ground truth in the ith cluster and n_i^r is corresponding number of learned labels. C_n^m is a combination operation.

NE measures the clusters' balancing degree. Let $\{n_i\}_{i=1}^c$ be the size of the ith cluster, these two measurements are calculated as follows:

$$\mathrm{NE} = -\frac{1}{\log c}\sum_{i=1}^{c}\frac{n_i}{n}\log\frac{n_i}{n}. \tag{22}$$

4) Competitive Methods: We compare our method with 11 clustering approaches including **unsupervised clustering methods** (K-means [71], DEC [55], IDEC [56]), **balanced clustering methods** [directed normalized cut (DNC) [72], EBMC [38], min-max cut (MMC) [73], Exp-Cut [39], BC [74], and BKNC [75]], and **semisupervised methods** [constrained k-means (COP) [76] and semi-DEC (SDEC) [77]].

For all comparison methods, we adopt the recommended network structure and parameters in the original papers. We run all methods ten times and record their means and standard variances.

B. Quantitative Evaluation

1) Feature Visualization: The performance of the algorithm is visualized using t-SNE [78] on the MNIST dataset. Fig. 4(a) and (b) illustrates that the data belonging to different classes are initially mixed, and the distance between clusters is minimal during the first few epochs. As the training progresses, the inner clusters gradually merge, and the inter-clusters start to separate. As shown in Fig. 4(h), each cluster becomes well separated eventually.

2) Evolution of Cluster Assignment: In this section, we investigate the effectiveness of our model compared with ten clustering algorithms. To comprehensively evaluate the performance of our method, we compute the mean and standard deviation scores for ACC, NMI, ARI, and NE across multiple datasets. The performance results and mean running time are presented in Tables II and III, respectively, from which we can draw the following observations.

1) Our model outperforms state-of-the-art balance clustering models in terms of clustering performance. Our model consistently outperforms the most relevant DEC-based models, including DEC, IDEC, and SDEC, by at least 3%, with a particularly significant improvement of 15% on REUTERS-10K.
2) Our model obtains a remarkable improvement in balance performance on most datasets. While our model's balance performance may not be as strong as IDEC on REUTERS-10K, it outperforms IDEC on other metrics. Due to there being some semisupervised constraints in our model, they may sacrifice some balance in pursuit of clustering performance. In Sections IV-C, we will further analyze the relationship between balance and clustering.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
PERFORMANCE COMPARISON ON FOUR BENCHMARK DATASETS. THE BEST RESULT IS SHOWN AS **BOLDFACE** (MEAN±STD%)

		MNIST				USPS			
		ACC	NMI	ARI	NE	ACC	NMI	ARI	NE
Unsup.	K-means [71]	56.64±03.36	50.32±02.11	39.36±02.78	97.75±00.92	64.96±04.85	61.64±01.85	53.22±04.72	98.98±00.43
	DEC [55]	76.93±00.15	74.67±00.15	68.67±00.37	99.20±00.01	74.16±00.13	73.21±00.05	66.17±00.12	98.68±00.04
	IDEC [56]	83.51±00.08	85.09±00.26	79.05±00.20	97.75±00.02	75.95±00.11	77.28±00.09	69.39±00.14	98.37±00.04
Bal.	DNC [72]	60.93±03.63	59.32±02.81	46.48±03.61	98.34±01.71	68.60±03.47	69.36±01.04	58.92±01.74	98.63±00.46
	EBMC [38]	61.00±03.70	59.25±03.11	46.47±03.86	97.83±00.94	68.63±03.47	69.20±00.97	59.07±02.00	98.30±00.61
	MMC [73]	56.64±03.36	50.32±02.11	39.36±02.78	97.75±00.92	64.96±03.48	61.64±01.18	53.22±01.95	98.52±00.43
	Exp-Cut [39]	62.50±04.22	59.98±04.04	48.45±04.67	99.98±00.00	67.84±00.87	66.31±00.84	55.50±01.05	99.99±00.01
	BC [74]	55.68±00.01	48.60±00.01	38.69±00.01	99.99±00.00	60.51±00.02	57.04±00.01	46.12±00.01	**99.99±00.00**
	BKNC [75]	53.36±00.02	44.47±00.01	33.58±00.01	89.28±00.01	63.82±00.02	58.04±00.01	50.23±00.01	90.05±00.01
Semi.	COP [76]	54.86±02.35	49.66±01.20	38.09±01.86	98.44±00.63	64.41±03.45	62.86±01.43	54.22±01.95	98.31±00.61
	SDEC [77]	88.71±00.11	89.15±00.27	86.80±00.17	97.58±00.01	85.80±00.27	83.14±00.48	82.03±00.57	94.69±00.06
	OURS	**98.08±00.36**	**94.52±00.29**	**95.80±00.47**	**99.99±00.01**	**87.59±00.29**	**85.93±00.55**	**85.76±00.47**	97.79±00.22
		REUTERS-10K				STL-10			
		ACC	NMI	ARI	NE	ACC	NMI	ARI	NE
Unsup.	K-means [71]	65.91±10.96	46.39±09.39	41.01±12.18	81.15±13.32	80.88±06.08	81.37±02.42	74.53±04.71	97.58±01.22
	DEC [55]	68.83±00.31	44.16±00.40	46.69±00.63	98.85±00.22	90.65±00.44	84.58±00.52	81.09±00.89	99.42±00.50
	IDEC [56]	70.31±00.09	46.81±00.12	50.07±00.19	99.29±00.12	90.44±00.19	83.74±00.32	80.97±00.41	99.79±00.70
Bal.	DNC [72]	66.65±10.67	46.56±09.23	41.64±12.14	90.85±00.18	82.85±06.13	83.86±02.56	77.84±05.08	97.57±01.25
	EBMC [38]	66.51±10.72	46.05±09.12	41.35±11.85	80.92±12.50	82.45±06.23	83.24±02.65	77.01±05.17	94.80±02.64
	MMC [73]	65.91±10.96	46.39±09.39	41.01±12.18	81.87±12.31	80.88±06.08	81.37±02.42	74.53±04.71	95.04±02.41
	Exp-Cut [39]	53.63±06.42	23.25±08.41	22.44±10.45	94.76±07.17	86.57±06.20	81.62±05.38	78.19±07.93	99.95±00.01
	BC [74]	66.42±00.02	45.22±00.06	42.37±00.06	**100.00±00.00**	94.20±00.00	87.93±00.00	87.77±00.00	90.00±00.00
	BKNC [75]	61.92±00.00	38.11±00.01	27.51±00.00	64.61±00.07	85.54±00.02	76.98±00.02	72.76±00.02	99.73±00.00
Semi.	COP [76]	69.82±06.75	48.09±09.16	44.09±11.86	89.22±06.32	77.68±08.64	79.92±03.67	72.40±06.59	96.43±02.14
	SDEC [77]	75.33±00.04	60.87±00.06	61.03±00.08	97.53±00.14	91.83±00.07	85.30±00.14	83.49±00.15	99.83±00.69
	OURS	**90.16±00.14**	**69.21±00.18**	**79.73±00.43**	85.13±00.09	**96.65±00.32**	**92.05±00.52**	**92.74±00.67**	**99.99±00.02**

TABLE III
TIME COST COMPARISON ON FOUR BENCHMARK DATASETS (SECONDS)

	MNIST	USPS	REUTERS-10K	STL-10
IDEC [56]	110.10	20.65	24.04	29.38
SDEC [77]	128.71	20.91	25.82	29.69
Ours	138.38	30.68	35.28	38.88

3) It is both intuitive and theoretically supported that semisupervised algorithms outperform unsupervised ones. Furthermore, our model outperforms all other semisupervised competitors. This demonstrates that our model effectively utilizes semisupervised information.
4) The calculated variance shown in Table II demonstrates that our model exhibits high stability across different datasets. This shows that our model is robust to factors such as initialization, distribution of pairwise constraints, and so on.
5) The running time summarized in Table III told us that our proposed method is slightly slower than others. This disparity can be attributed to the utilization of an alternative optimization strategy during the training process. While this choice may result in a minor sacrifice in computational efficiency, it leads to significant improvements in cluster performances.

C. In-Depth Analysis

We conduct ablation studies to investigate the effect of different design choices in our model.

1) Ablation Study: To comprehensively understand the contributions of the different components in our model, we conducted an ablation study with seven variants. As the model comprises three modules, we obtained six variants by using different combinations of the loss function. It is important to note that the only difference between these variants is the loss function, while all other factors such as network structures and hyperparameters remain the same. Table IV shows the experimental results on four datasets with six variants. Overall, it is evident that the best clustering results are achieved when all modules are utilized. The best balance performances are typically achieved when only the balance clustering loss is present. But in this case, the clustering performances are poor. An explanation is that if there is only clustering loss (such as USPS in Table IV), it achieves the data strictly balance without preserving any data distribution. Going a step further, the other three datasets still cannot achieve the best results with reconstruction and balance clustering losses. On the other hand, when there is a semisupervised constraint, the clustering performance is significantly higher than the others. It directly illustrates the superiority of the semisupervised module.

2) Parameter Sensitivity: Now, we investigate the impact of the parameters α and β. In each test, we change their

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE IV

ABLATION STUDY ON FOUR DIFFERENT DATASETS. ✓ INDICATES THE USE OF THIS LOSS, AND BLANK INDICATES NO USE. THE BEST RESULT IS SHOWN AS **BOLDFACE** (MEAN±STD%)

			MNIST				USPS			
$\mathcal{L}_r$	$\mathcal{L}_{clu}$	$\mathcal{L}_{semi}$	ACC	NMI	ARI	NE	ACC	NMI	ARI	NE
✓			76.63±0.11	73.51±0.09	66.26±0.14	98.33±0.03	70.12±0.20	66.68±0.35	56.44±0.59	96.54±0.12
	✓		86.69±0.64	80.73±0.67	76.71±1.08	99.94±0.06	73.29±0.18	73.71±0.18	65.50±0.42	**99.64±0.01**
		✓	88.20±0.46	90.50±0.54	85.76±0.95	94.11±0.22	74.11±0.07	74.93±0.13	72.16±0.13	96.98±0.04
✓	✓		84.00±0.35	79.91±0.36	74.56±0.64	**99.95±0.03**	73.96±0.04	74.19±0.05	66.83±0.10	99.59±0.00
✓		✓	88.34±0.40	90.86±0.31	86.03±0.66	94.11±0.02	65.61±0.06	68.26±0.10	63.04±0.12	95.82±0.06
	✓	✓	97.53±0.07	93.25±0.16	94.62±0.13	99.40±0.09	86.14±0.26	84.99±0.36	83.89±0.43	95.66±0.13
✓	✓	✓	**98.08±0.36**	**94.52±0.29**	**95.80±0.47**	99.94±0.01	**87.59±0.29**	**85.93±0.55**	**85.76±0.47**	97.79±0.22
			REUTERS-10K				STL-10			
$\mathcal{L}_r$	$\mathcal{L}_{clu}$	$\mathcal{L}_{semi}$	ACC	NMI	ARI	NE	ACC	NMI	ARI	NE
✓			68.29±0.96	41.50±1.51	43.51±2.07	97.22±0.31	64.53±1.23	59.23±0.67	42.46±1.12	93.88±0.49
	✓		70.39±0.12	51.97±0.45	52.39±0.35	99.77±0.04	95.46±0.08	90.13±0.11	90.23±0.18	99.96±0.01
		✓	88.25±0.19	66.59±0.41	78.12±0.49	77.41±0.29	96.04±0.03	90.74±0.04	91.44±0.06	99.99±0.00
✓	✓		67.42±0.08	47.74±0.15	47.31±0.15	**99.93±0.01**	95.51±0.07	90.17±0.11	90.34±0.15	99.97±0.01
✓		✓	88.06±0.05	65.20±0.18	77.44±0.16	77.46±0.09	76.83±0.03	79.42±0.06	74.21±0.05	92.37±0.05
	✓	✓	89.29±0.10	68.49±0.20	78.68±0.31	86.21±0.21	95.99±0.29	90.92±0.44	91.34±0.61	99.97±0.03
✓	✓	✓	**90.25±0.02**	**69.90±0.14**	**80.92±0.04**	87.59±0.24	**96.65±0.32**	**92.05±0.52**	**92.74±0.67**	**99.99±0.02**

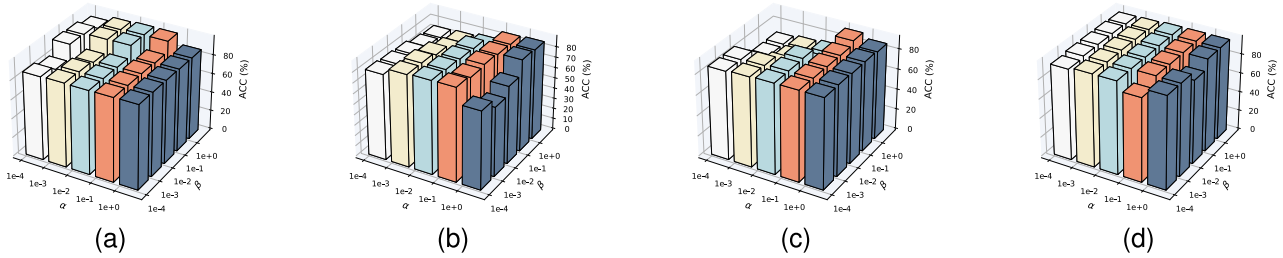

Fig. 5. Clustering performance under different values α and β on four benchmark datasets. (a) MNIST. (b) USPS. (c) REUTERS-10K. (d) STL-10.

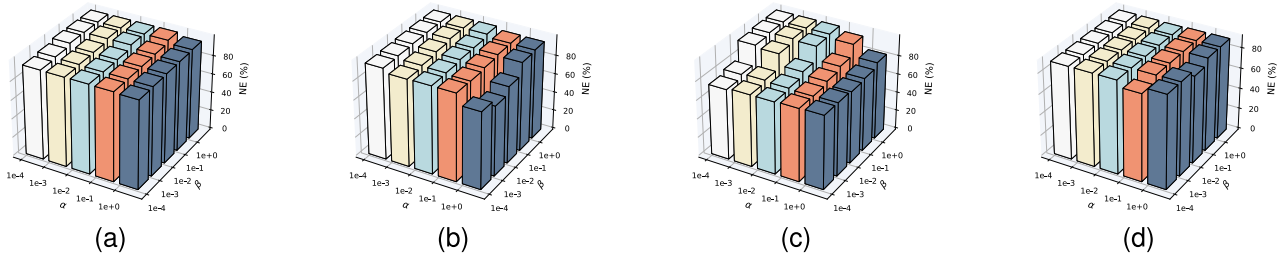

Fig. 6. Clustering performance NE under different values α and β on four benchmark datasets. (a) MNIST. (b) USPS. (c) REUTERS-10K. (d) STL-10.

values and report the mean classification ACC and NE in Figs. 5 and 6, respectively. From Fig. 5, we can see that on the MNIST, when α is small with large β, it gets better results, which is different from other datasets. One reason is that the calculated balance loss value is relatively large due to a large number of samples, so a small α is needed to trade-off the loss on the same scale. For other datasets, a larger β can make the clustering performance better with a suitable α. On the other hand, Fig. 6 shows the effect of parameters on balancing performance. Our model is very stable on all datasets except REUTERS-10K. This is because the data distribution is complex and the boundaries between clusters are tight. It needs more semisupervised information to guide the training to get the best final results. In sum, it can be seen that our model is robust for hyperparameters with a reasonable range.

3) Must-Link Versus Cannot Link: More deeply, we investigate the impact of two different constraints on model performance. We limit the total number of constraints to 20% of the length of the dataset and then adjust different proportions of constraints, such as {ML = 0%, CL = 20%}, {ML = 4%, CL = 16%}, . . . , {ML = 20%, CL = 0%}. As shown in Fig. 7, when there only exists a kind of constraint, the model often performs not well. Specifically, on MNIST and USPS, the performance deteriorates when no cannot-link constraints exist. The model's balanced performance will decrease with the number of cannot-link constraints shrinking on REUTERS-10K and STL-10.

4) Ratio of Pairwise Constraints: In this section, we will discuss how the number of constraints affects model performance. We plot the clustering performances in Fig. 8 corresponding to ACC and NE using a varying number of

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

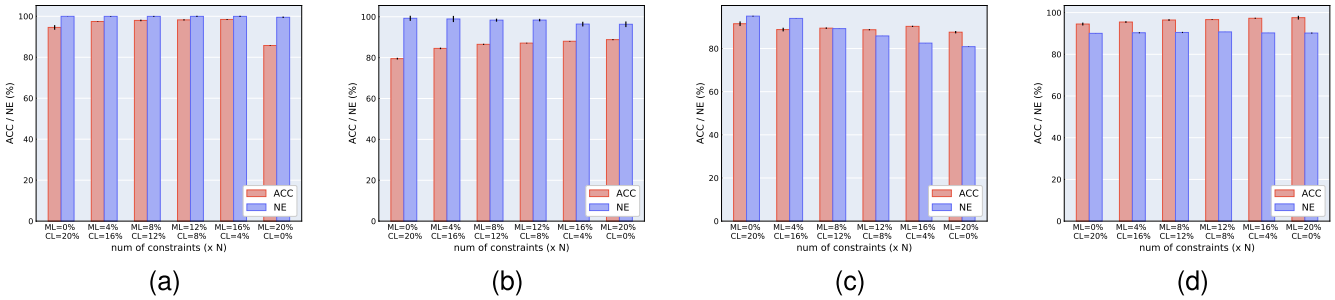

Fig. 7. Clustering performance of our model on four different datasets. When the total number of constraints is limited, the proportion of two different constraints is different. (a) MNIST. (b) USPS. (c) REUTERS-10K. (d) STL-10.

TABLE V
MODEL PERFORMANCE SENSITIVITY TO DIFFERENT NETWORK INITIALIZATION (MEAN±STD%)

	MNIST				USPS			
	ACC	NMI	ARI	NE	ACC	NMI	ARI	NE
Xaiver [79]	98.11±0.05	94.56±0.10	95.88±0.10	99.94±0.01	86.54±0.24	85.39±0.42	84.91±0.34	95.48±0.26
Gaussian	98.07±0.07	94.48±0.15	95.79±0.15	99.94±0.01	86.52±0.20	86.14±0.23	85.35±0.26	95.49±0.07
Kaiming [80]	98.08±0.36	94.52±0.29	95.80±0.47	99.94±0.01	87.59±0.29	85.93±0.55	85.76±0.47	97.79±0.22
	REUTERS-10K				STL-10			
	ACC	NMI	ARI	NE	ACC	NMI	ARI	NE
Xaiver [79]	89.69±0.06	69.06±0.14	80.27±0.16	83.24±0.08	96.25±0.29	91.34±0.46	91.92±0.61	99.98±0.02
Gaussian	88.17±0.10	67.41±0.16	78.39±0.24	84.39±0.27	95.58±0.18	90.34±0.28	90.50±0.38	99.94±0.01
Kaiming [80]	90.25±0.14	69.90±0.18	80.92±0.43	87.59±0.09	96.65±0.32	92.05±0.52	92.74±0.67	99.99±0.02

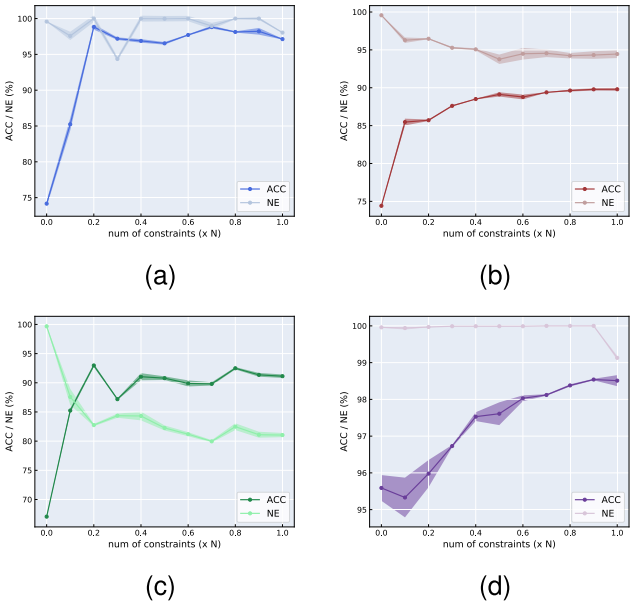

Fig. 8. Clustering performance of our model on four different datasets. The number of constraints varies between $0.1 \times N$ and N, where N is the length of the dataset. (a) MNIST. (b) USPS. (c) REUTERS-10K. (d) STL-10.

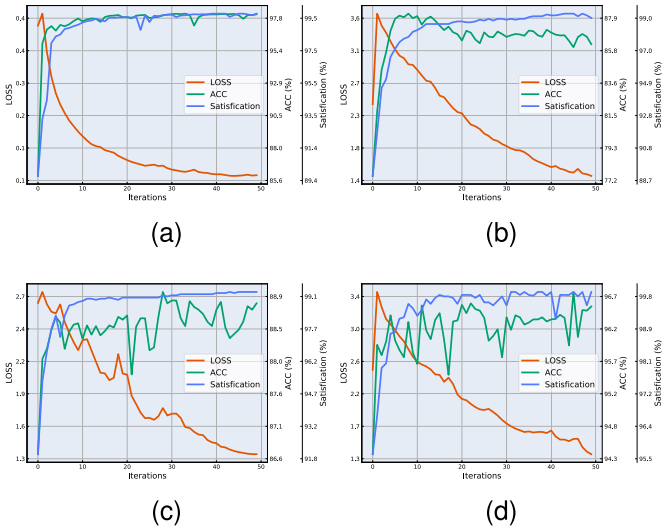

Fig. 9. Convergence analysis on four datasets with ACC and satisfaction of pairwise constraints. (a) MNIST. (b) USPS. (c) REUTERS-10K. (d) STL-10.

constraints from 0 to N. When the number of constraints gets close to N, ACC tends to saturate on MNIST and STL-10. On USPS and REUTERS-10K, NE tends to decrease as the number of constraints increases. One reason maybe that the semisupervised information is only imposed on the labeled data, which destroys the locality of distribution.

5) Convergence Analysis: We analyze the convergence of our model on four datasets. Additionally, we investigate the ACC and satisfaction of the semisupervised constraints during training. As shown in Fig. 9, our model demonstrates convergence. It should be noted that the loss initially increases and then decreases in the first few epochs for each dataset. The main reason is that during the warm-up stage, we only utilize the reconstruction loss. However, in the training stage, the loss temporarily increases when the clustering and semisupervised loss are introduced. As training progresses, these two losses gradually decrease and converge. Regarding the satisfaction of the semisupervised constraints, it is observed that the final results satisfy almost all of the

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

pairwise constraints (all above 99%). However, on MNIST, USPS, REUTERS-10K, and STL-10 datasets, there are a small number of unsatisfied constraints: 153/1400/70 000, 61/1860/9298, 66/2000/10 000, and 10/2600/13 000 (# unsatisfied/constraints/data), respectively.

6) Sensitivity to Model Initialization: Model initialization is an important part of feature learning and clustering performance. Here, we test its effects on our model performance on four datasets. Except for Kaiming [80], which is the default initialization in PyTorch [69], we still evaluate two more initialization ways: Gaussian and Xavier [79]. Table V shows the results that our model has stable performance on different initializations. This shows that our algorithm is insensitive to initialization.

V. Conclusion

In this article, we proposed a novel deep semisupervised balanced clustering with pairwise constraints, which can learn a clustering-favorable representation for clustering assignments. Additionally, we conducted a theoretical analysis to guarantee the clustering quality based on balanced tendency and pairwise constraints. The experimental results on MNIST, USPS, REUTERS-10K, and STL-10 demonstrate that our method achieves substantial performance improvements compared to recent approaches in balanced clustering, deep clustering, and SSC, in terms of cluster validity and balance measurement.

SSC assumes that we can obtain labeled samples from all categories in a close-set manner, however, it is often impractical to collect a large number of human annotations. Therefore, there exists a large number of unseen labeled instances. In the future, we will focus on addressing these aforementioned challenging problems, which are referred to as NCD and GCD.

References

[1] H.-J. Li, Z. Bu, Z. Wang, and J. Cao, "Dynamical clustering in electronic commerce systems via optimization and leadership expansion," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5327–5334, Aug. 2020.

[2] R. Vijayarajeswari, M. Nagabhushan, and P. Parthasarathy, "An enhanced symptom clustering with profile based prescription suggestion in biomedical application," *J. Med. Syst.*, vol. 43, no. 6, pp. 172:1–172:6, Jun. 2019.

[3] K. Wang, J. Zhang, D. Li, X. Zhang, and T. Guo, "Adaptive affinity propagation clustering," 2008, *arXiv:0805.1096*.

[4] Y. Jia, S. Kwong, J. Hou, and W. Wu, "Semi-supervised non-negative matrix factorization with dissimilarity and similarity regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2510–2521, Jul. 2020.

[5] Z. Xu, D. Shen, Y. Kou, and T. Nie, "A synthetic minority oversampling technique based on Gaussian mixture model filtering for imbalanced data classification," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 19, 2022, doi: 10.1109/TNNLS.2022.3197156.

[6] 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.

[7] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 977–986.

[8] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. AAAI*, 2016, pp. 1969–1976.

[9] Y. Wang, "Survey on deep multi-modal data analytics: Collaboration, rivalry, and fusion," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 1s, pp. 1–25, Jan. 2021.

[10] Y. Wang, W. Zhang, L. Wu, X. Lin, M. Fang, and S. Pan, "Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering," 2016, *arXiv:1608.05560*.

[11] Y. Zhang, Z. Lu, and S. Wang, "Unsupervised feature selection via transformed auto-encoder," *Knowl.-Based Syst.*, vol. 215, Mar. 2021, Art. no. 106748.

[12] Y. Zhu, L. Li, and X. Wu, "Stacked convolutional sparse auto-encoders for representation learning," *ACM Trans. Knowl. Discovery Data*, vol. 15, no. 2, pp. 1–21, Apr. 2021.

[13] C. Zhang, Y. Geng, Z. Han, Y. Liu, H. Fu, and Q. Hu, "Autoencoder in autoencoder networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 15, 2022, doi: 10.1109/TNNLS.2022.3189239.

[14] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *Proc. ECCV*, vol. 9911, 2016, pp. 835–851.

[15] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[16] K. Ghasedi, X. Wang, C. Deng, and H. Huang, "Balanced self-paced learning for generative adversarial clustering network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4386–4395.

[17] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," 2015, *arXiv:1511.05493*.

[18] R. Zhang, H. Tong, Y. Xia, and Y. Zhu, "Robust embedded deep K-means clustering," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1181–1190.

[19] X. Peng, S. Xiao, J. Feng, W. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior," in *Proc. IJCAI*, 2016, pp. 1925–1931.

[20] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," 2017, *arXiv:1709.02508*.

[21] U. Shaham, K. Stanton, H. Li, B. Nadler, R. Basri, and Y. Kluger, "SpectralNet: Spectral clustering using deep neural networks," 2018, *arXiv:1801.01587*.

[22] S. Randhawa and S. Jain, "MLBC: Multi-objective load balancing clustering technique in wireless sensor networks," *Appl. Soft Comput.*, vol. 74, pp. 66–89, Jan. 2019.

[23] L. Bai and J. Liang, "K-relations-based consensus clustering with entropy-norm regularizers," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 6, 2023, doi: 10.1109/TNNLS.2023.3307158.

[24] Z. Song, X. Yang, Z. Xu, and I. King, "Graph-based semi-supervised learning: A comprehensive review," 2021, *arXiv:2102.13303*.

[25] T. Xie, B. Wang, and C.-C. Jay Kuo, "GraphHop: An enhanced label propagation method for node classification," 2021, *arXiv:2101.02326*.

[26] X. Yang, X. Hu, S. Zhou, X. Liu, and E. Zhu, "Interpolation-based contrastive learning for few-label semi-supervised learning," 2022, *arXiv:2202.11915*.

[27] I. M. Ziko, J. Dolz, E. Granger, and I. B. Ayed, "Laplacian regularized few-shot learning," in *Proc. ICML*, in Proceedings of Machine Learning Research, vol. 119, 2020, pp. 11660–11670.

[28] Y. Zhang, W. Li, M. Zhang, S. Wang, R. Tao, and Q. Du, "Graph information aggregation cross-domain few-shot learning for hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 30, 2022, doi: 10.1109/TNNLS.2022.3185795.

[29] Y. Cui, W. Deng, H. Chen, and L. Liu, "Uncertainty-aware distillation for semi-supervised few-shot class-incremental learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 31, 2023, doi: 10.1109/TNNLS.2023.3277018.

[30] S. Rajeswar, P. Rodríguez, S. Singhal, D. Vazquez, and A. Courville, "Multi-label iterated learning for image classification with label ambiguity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4773–4783.

[31] H. Wang et al., "PICO+: Contrastive label disambiguation for partial label learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Dec. 13, 2023, doi: 10.1109/TPAMI.2023.3342650.

[32] J. Wen et al., "Deep double incomplete multi-view multi-label learning with incomplete labels and missing views," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 29, 2023, doi: 10.1109/TNNLS.2023.3260349.

[33] K. Han, A. Vedaldi, and A. Zisserman, "Learning to discover novel visual categories via deep transfer clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8400–8408.

[34] S. Vaze, K. Hant, A. Vedaldi, and A. Zisserman, "Generalized category discovery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7482–7491.

[35] P. S. Bradley, K. P. Bennett, and A. Demiriz, "Constrained k-means clustering," *Microsoft Research, Redmond*, vol. 20, no. 2, p. 10, 2000.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[36] M. I. Malinen and P. Fränti, "Balanced k-means for clustering," in *Proc. S+SSPR*, in Lecture Notes in Computer Science, vol. 8621. Cham, Switzerland: Springer, 2014, pp. 32–41.

[37] X. Chen, J. Z. Haung, F. Nie, R. Chen, and Q. Wu, "A self-balanced min-cut algorithm for image clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2080–2088.

[38] X. Chen, W. Hong, F. Nie, J. Z. Huang, and L. Shen, "Enhanced balanced min cut," *Int. J. Comput. Vis.*, vol. 128, no. 7, pp. 1982–1995, Jul. 2020.

[39] D. Wu, F. Nie, J. Lu, R. Wang, and X. Li, "Balanced graph cut with exponential inter-cluster compactness," *IEEE Trans. Artif. Intell.*, vol. 3, no. 4, pp. 498–505, Aug. 2022.

[40] J. Deng and J.-G. Yu, "A simple graph-based semi-supervised learning approach for imbalanced classification," *Pattern Recognit.*, vol. 118, Oct. 2021, Art. no. 108026.

[41] Y. Zhang et al., "Dual-constrained deep semi-supervised coupled factorization network with enriched prior," *Int. J. Comput. Vis.*, vol. 129, no. 12, pp. 3233–3254, Dec. 2021.

[42] Z. Wang, L. Zhang, R. Wang, F. Nie, and X. Li, "Semi-supervised learning via bipartite graph construction with adaptive neighbors," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 5257–5268, May 2023.

[43] F. Nie, H. Zhang, R. Wang, and X. Li, "Semi-supervised clustering via pairwise constrained optimal graph," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 3160–3166.

[44] P. Hu, H. Zhu, X. Peng, and J. Lin, "Semi-supervised multi-modal learning with balanced spectral decomposition," in *Proc. AAAI*, 2020, pp. 99–106.

[45] E. Yu, J. Sun, J. Li, X. Chang, X.-H. Han, and A. G. Hauptmann, "Adaptive semi-supervised feature selection for cross-modal retrieval," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1276–1288, May 2019.

[46] Y. Wen et al., "Unpaired multi-view graph clustering with cross-view structure matching," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 2, 2023, doi: 10.1109/TNNLS.2023.3291696.

[47] Z. Huang, P. Hu, J. T. Zhou, J. Lv, and X. Peng, "Partially view-aligned clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 2892–2902.

[48] H. Yu, J. Tang, G. Wang, and X. Gao, "A novel multi-view clustering method for unknown mapping relationships between cross-view samples," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 2075–2083.

[49] F. Nie, D. Wu, R. Wang, and X. Li, "Truncated robust principle component analysis with a general optimization framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 1081–1097, Feb. 2022.

[50] S. Wang, F. Nie, Z. Wang, R. Wang, and X. Li, "Robust principal component analysis via joint reconstruction and projection," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 11, 2023, doi: 10.1109/TNNLS.2022.3214307.

[51] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[52] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognit. Lett.*, vol. 20, nos. 11–13, pp. 1191–1199, Nov. 1999.

[53] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[54] M. Tschannen, O. Bachem, and M. Lucic, "Recent advances in autoencoder-based representation learning," 2018, *arXiv:1812.05069*.

[55] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, vol. 48, 2015, pp. 478–487.

[56] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1753–1759.

[57] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep clustering with convolutional autoencoders," in *Proc. ICONIP*, in Lecture Notes in Computer Science, vol. 10635. Cham, Switzerland: Springer, 2017, pp. 373–382.

[58] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," 2016, *arXiv:1611.05148*.

[59] D. Zhang, Y. Sun, B. Eriksson, and L. Balzano, "Deep unsupervised clustering using mixture of autoencoders," 2017, *arXiv:1712.07788*.

[60] J. Cai, S. Wang, and W. Guo, "Unsupervised embedded feature learning for deep clustering with stacked sparse auto-encoder," *Expert Syst. Appl.*, vol. 186, Dec. 2021, Art. no. 115729.

[61] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning: Objectives and optimization," 2016, *arXiv:1602.01024*.

[62] C. Zhang, Y. Liu, and H. Fu, "AE^2-Nets: Autoencoder in autoencoder networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2572–2580.

[63] 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.*, vol. 97, Jun. 2019, pp. 5092–5101.

[64] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010, pp. 807–814.

[65] 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, Feb. 1998.

[66] 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.

[67] A. Coates, H. Lee, and A. Y. Ng, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist. (AISTATS)*, vol. 15, 2011, pp. 215–223.

[68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[69] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NeurIPS*, 2019, pp. 8024–8035.

[70] H. W. Kuhn, "The Hungarian method for the assignment problem," Tech. Rep., 2010, pp. 29–47.

[71] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *J. Roy. Stat. Soc., C, Appl. Statist.)*, vol. 28, no. 1, pp. 100–108, 1979.

[72] X. Chen, W. Hong, F. Nie, D. He, M. Yang, and J. Z. Huang, "Spectral clustering of large-scale data by directly solving normalized cut," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1206–1215.

[73] F. Nie, C. H. Q. Ding, D. Luo, and H. Huang, "Improved minmax cut graph clustering with nonnegative relaxation," in *Proc. ECML/PKDD*, in Lecture Notes in Computer Science, vol. 6322. Cham, Switzerland: Springer, 2010, pp. 451–466.

[74] W. Lin, Z. He, and M. Xiao, "Balanced clustering: A uniform model and fast algorithm," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2987–2993.

[75] H. Chen, Q. Zhang, R. Wang, F. Nie, and X. Li, "A general soft-balanced clustering framework based on a novel balance regularizer," *Signal Process.*, vol. 198, Sep. 2022, Art. no. 108572.

[76] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *Proc. ICML*. Burlington, MA, USA: Morgan Kaufmann, 2001, pp. 577–584.

[77] Y. Ren, K. Hu, X. Dai, L. Pan, S. C. H. Hoi, and Z. Xu, "Semi-supervised deep embedded clustering," *Neurocomputing*, vol. 325, pp. 121–130, Jan. 2019.

[78] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–12, 2008.

[79] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, May 2010.

[80] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

Yu Duan received the M.S. degree from Northwestern Polytechnical University, Xi'an, China, in 2021, where he is currently pursuing the Ph.D. degree with the School of Computer Science and the School of Artificial Intelligence, Optics and Electronics (iOPEN).

His research interests include graph learning.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Zhoumin Lu received the M.S. degree in computer technology from Fuzhou University, Fuzhou, China, in 2021. He is currently pursuing the Ph.D. degree with the School of Computer Science, Northwestern Polytechnical University, Xi'an, China.

His research interests include machine learning, deep learning, and their applications, such as pattern recognition and data mining.

Rong Wang received the B.S. degree in information engineering, the M.S. degree in signal and information processing, and the Ph.D. degree in computer science from the Xi'an Research Institute of Hi-Tech, Xi'an, China, in 2004, 2007, and 2013, respectively.

From 2007 to 2013, he was also studied with the Department of Automation, Tsinghua University, Beijing, China, for his Ph.D. degree. He is currently an Associate Professor with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an. His research interests include machine learning and its applications.

Xuelong Li (Fellow, IEEE) is currently a Full Professor with the School of Computer Science and the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China.

Feiping Nie (Senior Member, IEEE) received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2009.

He is currently a Full Professor with Northwestern Polytechnical University, Xi'an, China. He has authored more than 100 papers in the following journals and conferences: IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), *International Journal of Computer Vision* (IJCV), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, and ACM MM. His papers have been cited more than 20 000 times and the H-index is 84. His research interests include machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing, and information retrieval.

Dr. Nie is currently serving as an associate editor or a PC member for several prestigious journals and conferences in related fields.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on October 10,2024 at 03:58:06 UTC from IEEE Xplore. Restrictions apply.