Deep Clustering: A Comprehensive Survey

Yazhou Ren, *Member, IEEE*, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, Philip S. Yu, *Fellow, IEEE*, and Lifang He, *Member, IEEE*

***Abstract*— Cluster analysis plays an indispensable role in machine learning and data mining. Learning a good data representation is crucial for clustering algorithms. Recently, deep clustering (DC), which can learn clustering-friendly representations using deep neural networks (DNNs), has been broadly applied in a wide range of clustering tasks. Existing surveys for DC mainly focus on the single-view fields and the network architectures, ignoring the complex application scenarios of clustering. To address this issue, in this article, we provide a comprehensive survey for DC in views of data sources. With different data sources, we systematically distinguish the clustering methods in terms of methodology, prior knowledge, and architecture. Concretely, DC methods are introduced according to four categories, i.e., traditional single-view DC, semi-supervised DC, deep multiview clustering (MVC), and deep transfer clustering. Finally, we discuss the open challenges and potential future opportunities in different fields of DC.**

***Index Terms*— Deep clustering (DC), multiview clustering (MVC), semi-supervised clustering, transfer learning.**

Nomenclature

i	Counter variable.
j	Counter variable.
$\vert\cdot\vert$	Length of a set.
$\Vert\cdot\Vert$	2-norm of a vector.
X	Data for clustering.
X^s	Data in source domain (UDA methods).
Y^s	Labels of source domain instances (UDA methods).
X^t	Data in target domain (UDA methods).
$\mathcal{D}_s$	Source domain of UDA methods.
$\mathcal{D}_t$	Target domain of UDA methods.
x_i	Vector of an original data sample.
X^i	ith view of X in multiview learning.
$\hat{Y}$	Predicted labels of X.
S	Soft data assignments of X.
R	Adjusted assignments of S.
A	Pairwise constraint matrix.
a_{ij}	Constraint of samples i and j.
z_i	Vector of the embedded representation of x_i.
ε	Noise used in generative model.
$\mathbb{E}$	Expectation.
L_n	Network loss.
L_c	Clustering loss.
L_{ext}	Extra task loss.
L_{rec}	Reconstruction loss of autoencoder network.
L_{gan}	Loss of GAN.
L_{ELBO}	Loss of evidence lower bound.
k	Number of clusters.
n	Number of data samples.
μ	Mean of the Gaussian distribution.
θ	Variance of the Gaussian distribution.
$\text{KL}(\cdot\Vert\cdot)$	Kullback–Leibler divergence.
$p(\cdot)$	Probability distribution.
$p(\cdot\vert\cdot)$	Conditional probability distribution.
$p(\cdot,\cdot)$	Joint probability distribution.
$q(\cdot)$	Approximate probability distribution of $p(\cdot)$.
$q(\cdot\vert\cdot)$	Approximate probability distribution of $p(\cdot\vert\cdot)$.
$q(\cdot,\cdot)$	Approximate probability distribution of $p(\cdot,\cdot)$.
$f(\cdot)$	Feature extractor.
$\phi_e(\cdot)$	Encoder network of AE or VAE.
$\phi_r(\cdot)$	Decoder network of AE or VAE.
$\phi_g(\cdot)$	Generative network of GAN.
$\phi_d(\cdot)$	Discriminative network of GAN.
M	Graph adjacency matrix.
D	Degree matrix of Q.
C	Feature matrix of a graph.
H	Node hidden feature matrix.
W	Learnable model parameters.

I. Introduction

With the development of online media, abundant data with high complexity can be gathered easily. Through pinpoint analysis of these data, we can dig the value out and use these conclusions in many fields, such as face recognition [1], [2], sentiment analysis [3], [4], and intelligent manufacturing [5], [6].

A model that can be used to classify the data with different labels is the base of many applications. For labeled data, it is taken granted to use the labels as the most important information as a guide. For unlabeled data, finding a quantifiable objective as the guide of the model-building process is the key question of clustering. Over the past decades, a large number of clustering methods with shallow models have been

Manuscript received 1 March 2023; revised 15 December 2023; accepted 23 April 2024. This work was supported in part by Shenzhen Science and Technology Program under Grant JCYJ20230807120010021 and Grant JCYJ20230807115959041, and in part by NSF under Grant III-2106758. The work of Lifang He was supported in part by NSF under Grant MRI-2215789, Grant IIS-1909879, and Grant IIS-2319451; in part by NIH under Grant R21EY034179; and in part by the Lehigh's Grants through Accelerator and CORE. *(Corresponding author: Yazhou Ren.)*

Yazhou Ren and Xiaorong Pu are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518000, China (e-mail: yazhou.ren@uestc.edu.cn).

Jingyu Pu, Zhimeng Yang, Jie Xu, and Guofeng Li are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China.

Philip S. Yu is with the Department of Computer Science, University of Illinois Chicago, Chicago, IL 60607 USA.

Lifang He is with the Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 18015 USA.

Digital Object Identifier 10.1109/TNNLS.2024.3403155

proposed, including centroid-based clustering [7], [8], density-based clustering [9], [10], [11], [12], [13], distribution-based clustering [14], hierarchical clustering [15], ensemble clustering [16], [17], and multiview clustering (MVC) [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. These shallow models are effective only when the features are representative, while their performance on the complex data is usually limited due to the poor power of feature learning.

In order to map the original complex data to a feature space that is easy to cluster, many clustering methods focus on feature extraction or feature transformation, such as principal component analysis (PCA) [31], kernel method [32], spectral method [33], and deep neural network (DNN) [34]. Among these methods, the DNN is a promising approach because of its excellent nonlinear mapping capability and its flexibility in different scenarios. A well-designed deep learning-based clustering approach [referred to deep clustering (DC)] aims at effectively extracting more clustering-friendly features from data and performing clustering with learned features simultaneously. Different from the traditional clustering method, the fundamental concept behind DC involves incorporating the clustering objective into the robust representation capabilities offered by deep learning. Thus, acquiring a vital data representation becomes an essential requirement for DC.

Much research has been done in the field of DC and there are also some surveys about DC methods [35], [36], [37], [38]. Specifically, existing systematic reviews for DC mainly focus on the single-view clustering tasks and the architectures of neural networks. For example, Aljalbout et al. [35] focus only on deep single-view clustering methods that are based on deep autoencoder (DAE). Min et al. [36] classify DC methods from the perspective of different deep networks. Nutakki et al. [37] divide deep single-view clustering methods into three categories according to their training strategies: multistep sequential DC, joint DC, and closed-loop multistep DC. Zhou et al. [38] categorize deep single-view clustering methods by the interaction way between feature learning and clustering modules. However, in the real world, the datasets for clustering are always associated, e.g., the taste for reading is correlated with the taste for a movie, and the side face and full face from the same person should be labeled the same. For these data, DC methods based on semi-supervised learning, multiview learning, and transfer learning have also made significant progress. Unfortunately, existing reviews do not discuss them too much.

Therefore, it is important to classify DC from the perspective of data sources. In this survey, we summarize the DC from the perspective of initial settings of data combined with deep learning methodology. We introduce the newest progress of DC from the perspective of network and data structure, as shown in Fig. 1. Specifically, we organize the DC methods into the following four categories. Since we divide methods from the perspective of data source, the implementation of specific methods is not completely separated from each other. There is a part of the method that is interrelated with each other internally.

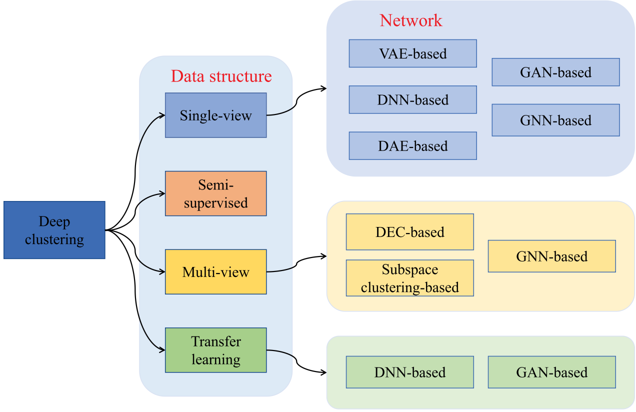


Fig. 1. Directory tree of this survey.

1) *Deep Single-View Clustering:* For conventional clustering tasks, it is often assumed that the data are of the same form and structure, as known as single-view or single-modal data. The extraction of representations for these data by DNNs is a significant characteristic of DC. However, what is more noteworthy is the different applied deep learning techniques, which are highly correlated with the structure of DNNs. To compare the technical route of specific DNNs, we divide those algorithms into five categories: DAE-based DC, DNN-based DC, variational autoencoder (VAE)-based DC, generative adversarial network (GAN)-based DC, and graph neural network (GNN)-based DC.
2) *DC Based on Semi-Supervised Learning:* When the data to be processed contain a small part of prior constraints, traditional clustering methods cannot effectively utilize this prior information and semi-supervised clustering is an effective way to solve this question. In presence, the research of deep semi-supervised clustering has not been well explored. However, semi-supervised clustering is inevitable because it is feasible to let a clustering method become a semi-supervised one by adding the additional information as a constraint loss to the model.
3) *DC Based on Multiview Learning:* In the real world, data are often obtained from different feature collectors or have different structures. We call those data “multiview data” or “multimodal data,” where each sample has multiple representations. The purpose of DC based on multiview learning is to utilize the consistent and complementary information contained in multiview data to improve clustering performance. In addition, the idea of multiview learning may have guiding significance for deep single-view clustering. In this survey, we summarize deep MVC into three categories: deep-embedded clustering (DEC)-based, subspace clustering-based, and GNN-based.
4) *DC Based on Transfer Learning:* For a task that has a limited amount of instances and high dimensions, sometimes we can find an assistant to offer additional information. For example, if task A is similar to another task B, and task B has more information for clustering than A (B is labeled or B is easier to clustering than A), it is useful to transfer the information from B to A. Transfer learning for unsupervised domain adaption (UDA) has been boosted in recent years, which contains two domains: source domain with labels and target domain that is unlabeled. The goal of transfer learning is to apply the knowledge or patterns learned from the source task to a different but related target task. DC methods based on transfer learning aim to improve the performance of current clustering tasks by utilizing information from relevant tasks.

It is necessary to pay attention to the different characteristics and conditions of the clustering data before studying the corresponding clustering methods. In this survey, existing DC methods are systematically classified from data sources. The advantages, disadvantages, and applicable conditions of different clustering methods are analyzed. Finally, we present some interesting research directions in the field of DC.

II. Definitions and Preliminaries

We introduce the notations in this section. Throughout this article, we use uppercase letters to denote matrices and lowercase letters to denote vectors. Unless otherwise stated, the notations used in this article are summarized in the Nomenclature.

This survey will introduce four kinds of DC problems based on different background conditions. Here, we define these problems formally. Given a set of data samples X, we aim at finding a map function F, which can map X into k clusters. The map result is represented with $\hat{Y}$. Therefore, the tasks we cope with are given as follows.

1) *Deep Single-View Clustering:*

$$F(X) \rightarrow \hat{Y}. \tag{1}$$

2) *Semi-Supervised Deep Clustering:*

$$F(X, A) \rightarrow \hat{Y} \tag{2}$$

where A is a constrained matrix.

3) *Deep MVC:*

$$F\left(X^1, \ldots, X^v\right) \rightarrow \hat{Y} \tag{3}$$

where X^i is the ith view of X.

4) *Deep Clustering With Domain Adaptation:*

$$F\left(X^s, Y^s, X^t\right) \rightarrow \hat{Y} \tag{4}$$

where (X^s, Y^s) is the labeled source domain and X^t is the unlabeled target domain.

III. Deep Single-View Clustering

The theory of representation learning [39] shows the importance of feature learning (or representation learning) in machine learning tasks. However, deep representation learning is mostly supervised learning that requires many labeled data. As we mentioned before, the obstacle of the DC problem is what can be used to guide the training process such as labels in supervised problem. The most "supervised" information in DC is the data itself. Thus, how can we train an effective feature extractor to get good representation? According to the way the feature extractor is trained, we divide deep single-view clustering algorithms into five categories: *DAE-based*, *DNN-based*, *VAE-based*, *GAN-based*, and *GNN-based*. The difference of these methods is mainly about the loss components, where the loss terms are defined in Table I and explained as follows.

1) *DAE-Based/GNN-Based:* $L = L_{\text{rec}} + L_c$.
2) *DNN-Based:* $L = L_{\text{ext}} + L_c$.
3) *VAE-Based:* $L = L_{\text{ELBO}} + L_c$.
4) *GAN-Based:* $L = L_{\text{gan}} + L_c$.

In unsupervised learning, the issue we cope with is to train a reliable feature extractor without labels. There are mainly two ways in existing works: 1) a loss function that optimizes the pseudo-labels according to the principle: narrowing the inner cluster distance and widening the intercluster distance and 2) an extra task that can help train the feature extractor. For the clustering methods with specialized feature extractors, such as autoencoder, the reconstruction loss L_{rec} can be interpreted as the extra task. In this article, the clustering-oriented loss L_c indicates the loss of the clustering objective. *DAE-based/GNN-based* methods use an autoencoder/graph autoencoder as the feature extractor, so the loss functions are always composed of a reconstruction loss L_{rec} and another clustering-oriented loss L_c. By contrast, *DNN-based* methods optimize the feature extractor with extra tasks or other strategies L_{ext}. *VAE-based* methods optimize the loss of evidence lower bound (ELBO) L_{ELBO}. *GAN-based* methods are based on the generative adversarial loss L_{gan}. Based on these five dimensions, existing deep single-view clustering methods are summarized in Tables I and II.

A. DAE-Based

The autoencoder network [39] is originally designed for unsupervised representation learning of data and can learn a highly nonlinear mapping function. Using DAE [97] is a common way to develop DC methods. DAE aims to learn a low-dimensional embedding feature space by minimizing the reconstruction loss of the network, which is defined as

$$L_{\text{rec}} = \min \frac{1}{n} \sum_{i=1}^{n} \|x_i - \phi_r(\phi_e(x_i))\|^2 \tag{5}$$

where $\phi_e(\cdot)$ and $\phi_r(\cdot)$ represent the encoder network and decoder network of autoencoder, respectively. Using the encoder as a feature extractor, various clustering objective functions have been proposed. We summarize these DAE-based clustering methods as *DAE-based* DC. In *DAE-based* DC methods, there are two main ways to get the labels. The first way embeds the data into low-dimensional features and then clusters the embedded features with traditional clustering methods such as the k-means algorithm [7]. The second way jointly optimizes the feature extractor and the clustering results. We refer to these two approaches as "separate analysis" and "joint analysis" and elaborate on them in the following.

"Separate analysis" means that learning features and clustering data are performed separately. In order to solve the problem that representations learned by "separately analysis" are not cluster-oriented due to its innate characteristics, Huang et al. [41] propose a deep embedding network (DEN) for clustering, which imposes two constraints based on DAE objective: locality-preserving constraint and group sparsity constraint. Locality-preserving constraint urges the embedded features in the same cluster to be similar. Group sparsity constraint aims to diagonalize the affinity of representations. These two constraints improve the clustering performance while reducing the inner cluster distance and expanding intercluster distance. The objective of most clustering methods based on DAE is working on these two kinds of distance. Thus, in Table I, we summarize these methods from the perspective of "characteristics," which shows the way to optimize the inner cluster distance and intercluster distance.

Peng et al. [42] propose a novel deep learning-based framework in the field of subspace clustering, namely, deep subspace clustering with sparsity prior (PARTY). PARTY enhances the autoencoder by considering the relationship between different samples (i.e., structure prior) and solves the limitation of

TABLE I

SUMMARIES OF *DAE*- AND *DNN-Based* METHODS IN DEEP SINGLE-VIEW CLUSTERING. WE SUMMARIZE THE *DAE-Based* METHODS BASED ON "JOINTLY OR SEPARATELY" AND "CHARACTERISTICS"

Net	Methods	Jointly or Separately	Characteristics
DAE	AEC (2013) [40]	Separately	Optimize the distance between z_i and its closest cluster centroid.
	DEN (2014) [41]	Separately	Locality-preserving constraint, group sparsity constraint.
	PARTY (2016) [42]	Separately	Subspace clustering.
	DEC (2016) [43]	Jointly	Optimize the distribution of assignments.
	IDEC (2017) [44]	Jointly	Improve DEC [43] with local structure preservation.
	DSC-Nets (2017) [45]	Separately	Subspace clustering.
	DEPICT (2017) [46]	Jointly	Convolutional autoencoder and relative entropy minimization.
	DCN (2017) [47]	Jointly	Take the objective of k-means as the clustering loss.
	DMC (2017) [48]	Jointly	Multi-manifold clustering.
	DEC-DA (2018) [49]	Jointly	Improve DEC [43] with data augmentation.
	DBC (2018) [50]	Jointly	Self-paced learning.
	DCC (2018) [51]	Separately	Extend robust continuous clustering [52] with autoencoder. Not given k.
	DDLSC (2018) [53]	Jointly	Pairwise loss function.
	DDC (2019) [54]	Separately	Global and local constraints of relationships.
	DSCDAE (2019) [55]	Jointly	Subspace Clustering.
	NCSC (2019) [56]	Jointly	Dual autoencoder network.
	DDIC (2020) [57]	Separately	Density-based clustering. Not given k.
	SC-EDAE (2020) [58]	Jointly	Spectral clustering.
	ASPC-DA (2020) [59]	Jointly	Self-paced learning and data augmentation.
	ALRDC (2020) [60]	Jointly	Adversarial learning.
	N2D (2021) [61]	Separately	Manifold learning.
	AGMDC (2021) [62]	Jointly	Gaussian Mixture Model. Improve the inter-cluster distance.
Net	Methods	Clustering-oriented loss	Characteristics
DNN	JULE (2016) [63]	Yes	Agglomerative clustering.
	dkAE (2017) [64]	Yes	Information theoretic measures.
	DAC (2017) [65]	No	Self-adaptation learning. Binary pairwise-classification.
	DeepCluster (2018) [66]	No	Use traditional clustering methods to assign labels.
	CCNN (2018) [67]	No	Mini-batch k-means. Feature drift compensation for large-scale image data
	ADC (2018) [68]	Yes	Centroid embeddings.
	ST-DAC (2019) [69]	No	Spatial transformer layers. Binary pairwise-classification.
	RTM (2019) [70]	No	Random triplet mining.
	IIC (2019) [71]	No	Mutual information. Generated image pairs.
	DCCM (2019) [72]	No	Triplet mutual information. Generated image pairs.
	MMDC (2019) [73]	No	Multi-modal. Generated image pairs.
	SCAN (2020) [74]	Yes	Decouple feature learning and clustering. Nearest neighbors mining.
	DRC (2020) [75]	Yes	Contrastive learning.
	PICA (2020) [76]	Yes	Maximize the "global" partition confidence.

TABLE II

SUMMARIES OF *VAE*-, *GAN*-, AND *GNN-Based* METHODS IN DEEP SINGLE-VIEW CLUSTERING

<table>
<tr><th>Net</th><th>Methods</th><th colspan="2">Characteristics</th></tr>
<tr><td rowspan="9">VAE</td><td>VaDE (2016) [77]</td><td colspan="2">Gaussian mixture variational autoencoder.</td></tr>
<tr><td>GMVAE (2016) [78]</td><td colspan="2">Gaussian mixture variational autoencoder. Unbalanced clustering.</td></tr>
<tr><td>DGMC (2017) [79]</td><td colspan="2">Continuous Gumbel-Softmax distribution.</td></tr>
<tr><td>LTVAE (2018) [80]</td><td colspan="2">Latent tree model.</td></tr>
<tr><td>VLAC (2019) [81]</td><td colspan="2">Variational ladder autoencoders.</td></tr>
<tr><td>VAEIC (2020) [82]</td><td colspan="2">No pre-training process.</td></tr>
<tr><td>S3VDC (2020) [83]</td><td colspan="2">Improvement on four generic algorithmic.</td></tr>
<tr><td>DSVAE (2021) [84]</td><td colspan="2">Spherical latent embeddings.</td></tr>
<tr><td>DVAE (2022) [85]</td><td colspan="2">Additional classifier to distinguish clusters.</td></tr>
<tr><th>Net</th><th>Methods</th><th>With DAE</th><th>Characteristics</th></tr>
<tr><td rowspan="7">GAN</td><td>CatGAN (2015) [86]</td><td>No</td><td>Can be applied to both unsupervised and semi-supervised tasks.</td></tr>
<tr><td>DAGC (2017) [87]</td><td>Yes</td><td>Build an encoder to make the data representations easier to cluster.</td></tr>
<tr><td>DASC (2018) [88]</td><td>Yes</td><td>Subspace clustering.</td></tr>
<tr><td>ClusterGAN-SPL (2019) [89]</td><td>No</td><td>No discrete latent variables and applies self-paced learning based on [90].</td></tr>
<tr><td>ClusterGAN (2019) [90]</td><td>No</td><td>Train a GAN with a clustering-specific loss.</td></tr>
<tr><td>ADEC (2020) [91]</td><td>Yes</td><td>Reconstruction loss and adversarial loss are optimized in turn.</td></tr>
<tr><td>IMDGC (2022) [92]</td><td>No</td><td>Integrates a hierarchical generative adversarial network and mutual information maximization.</td></tr>
<tr><th>Net</th><th>Methods</th><th colspan="2">Characteristics</th></tr>
<tr><td rowspan="4">GNN</td><td>AGC (2019) [93]</td><td colspan="2">Attributed graph clustering.</td></tr>
<tr><td>AGAE (2019) [94]</td><td colspan="2">Ensemble clustering.</td></tr>
<tr><td>AGCHK (2020) [95]</td><td colspan="2">Utilize heat kernel in attributed graphs.</td></tr>
<tr><td>SDCN (2020) [96]</td><td colspan="2">Integrate the structural information into deep clustering.</td></tr>
</table>

traditional subspace clustering methods. As far as we know, PARTY is the first deep learning-based subspace clustering method, and it is the first work to introduce the global structure prior to the neural network for unsupervised learning. Different from PARTY, Ji et al. [45] propose another deep subspace clustering network (DSC-Net) architecture to learn nonlinear mapping and introduce a self-expressive layer to directly learn the affinity matrix.

Density-based clustering [9], [98] is another kind of popular clustering method. Ren et al. [57] propose deep density-based image clustering (DDIC) that uses DAE to learn the low-dimensional feature representations and then performs density-based clustering on the learned features. In particular, DDIC does not need to know the number of clusters in advance.

"Joint analysis" aims at learning a representation that is more suitable for clustering, which is different from separate analysis approaches that deep learning and clustering are carried out separately, and the neural network does not have a clustering-oriented objective when learning the features of data. Most subsequent DC studies combine clustering objectives with feature learning, which enables the neural network to learn features conducive to clustering from the potential distribution of data. In this survey, those methods are summarized as "joint analysis."

Inspired by the idea of nonparametric algorithm t-distributed stochastic neighbor embedding (t-SNE) [99], Xie et al. [43] propose a joint framework to optimize feature learning and clustering objective, which is named DEC. DEC first learns a mapping from the data space to a lower dimensional feature space via L_{rec} and then iteratively optimizes the clustering loss $\text{KL}(S\|R)$ (i.e., Kullback–Leibler (KL) divergence). Here, S denotes the soft assignments of data that describe the similarity between the embedded data and each cluster centroid (centroids are initialized with k-means), and R is the adjusted target distribution, which has purer cluster assignments compared to S.

DEC is a representative method in DC due to its joint learning framework and low computing complexity. Based on DEC, a number of variants have been proposed. For example, to guarantee local structure in the fine-tuning phase, improved DEC (IDEC) with local structure preservation [44] is proposed to optimize the weighted clustering loss and the reconstruction loss of autoencoder jointly. DEC with data augmentation (DEC-DA) [49] applies the data augmentation strategy in DEC. Li et al. [50] propose discriminatively boosted image clustering (DBC) to deal with image representation learning and image clustering. DBC has a similar pipeline as DEC, but the learning procedure is self-paced [100], where the easiest instances are first selected and more complex samples are expanded progressively.

In DEC, the predicted clustering assignments are calculated by the Student's t-distribution. Differently, Dizaji et al. [46] propose a deep-embedded regularized clustering (DEPICT) with a novel clustering loss by stacking a softmax layer on the embedded layer of the convolutional autoencoder (CAE). What is more, the clustering loss of DEPICT is regularized by a prior for the frequency of cluster assignments and layer-wise features reconstruction loss function. Yang et al. [47] directly take the objective of k-means as the clustering loss. The proposed model, named DC network (DCN), is a joint dimensionality reduction and k-means clustering approach, in which dimensionality reduction is accomplished via learning a DAE. Shah and Koltun [51] propose deep continuous clustering (DCC), an extension of robust continuous clustering [52] by integrating autoencoder into the paradigm. DCC performs clustering learning by jointly optimizing the defined data loss, pairwise loss, and reconstruction loss. In particular, it does not need prior knowledge of the number of clusters. Tzoreff et al. [53] propose deep discriminative latent space for clustering (DDLSC) to optimize the DAE with respect to a discriminative pairwise loss function.

Deep manifold clustering (DMC) [48] is the first method to apply deep learning in multimanifold clustering [101], [102]. In DMC, an autoencoder consisting of stacked RBMs [103] is trained to obtain the transformed representations. Both the reconstruction loss and the clustering loss of DMC are different from previous methods, that is, the reconstruction of one sample and its local neighborhood are used to define the locality-preserving objective. The penalty coefficient and the distance, measured by the Gaussian kernel between samples and cluster centers, are used to define the clustering-oriented objective.

The recently proposed *DAE-based* clustering algorithms also use the variants of DAE to learn better low-dimensional features and focus on improving the clustering performance by combining the ideas of traditional machine learning methods. For example, deep spectral clustering using dual autoencoder network (DSCDAE) [55] and spectral clustering via ensemble DAE learning (SC-EDAE) [58] aim to integrate spectral clustering into the carefully designed autoencoders for DC. Zhang et al. [56] propose neural collaborative subspace clustering (NCSC) using two confidence maps, which are established on the features learned by autoencoder, as supervision information for subspace clustering. In adaptive self-paced DC with data augmentation (ASPC-DA) [59], self-paced learning idea [100] and data augmentation technique are simultaneously incorporated. Its learning process is the same as DEC and consists of two stages, i.e., pre-training the autoencoder and fine-tuning the encoder.

In general, we notice that the network structure adopted is related to the type of data to be processed. For example, fully connected networks are generally used to extract 1-D data features, while convolutional neural networks (CNNs) are used to extract image features. Most of the above *DAE-based* DC methods can be implemented by both fully connected autoencoder and CAE, and thus, they apply to various types of data to some extent. However, in the field of computer vision, there is a class of DC methods that focus on image clustering. Those methods can date back to [104] and are summarized as *DNN-based* DC because they generally use CNNs to perform image feature learning and semantic clustering.

B. *DNN-Based*

This section introduces the *DNN-based* clustering methods. Unlike *DAE-based* clustering methods, *DNN-based* methods have to design extra tasks to train the feature extractor. In this survey, we summarize *DNN-based* DC methods in Table I from two perspectives: "clustering-oriented loss" and "characteristics." "Clustering-oriented loss" shows whether there is a loss function that explicitly narrows the inner cluster distance or widens the intercluster distance. Fig. 2 shows the framework of deep unsupervised learning based on a CNN.

When the DNN training process begins, the randomly initialized feature extractor is unreliable. Thus, DC methods based on randomly initialized neural networks generally employ traditional clustering tricks such as hierarchical clustering [105] or focus on extra tasks such as instance generation. For instance, Yang et al. [63] propose a joint unsupervised learning method named JULE, which applies agglomerative clustering magic to train the feature extractor. Specifically,

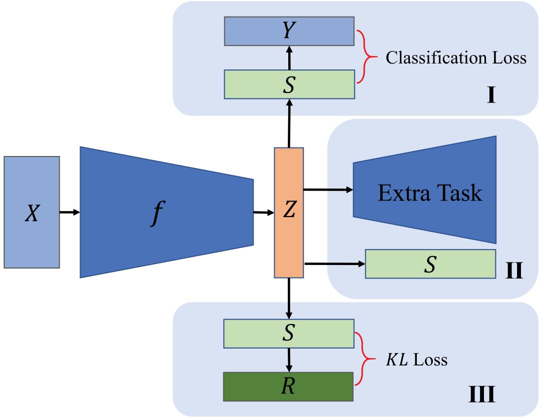


Fig. 2. Framework of *DNN-based* learning (single-view clustering). X is the data for clustering and f is the feature extractor for X. Part I describes the framework of supervised learning. Y means the real labels and S denotes the predicted results. With Y and S, we can compute the classification loss for backpropagation. Part II is the framework of methods with extra tasks. The extra tasks are used to train the nets for good embedding Z. Part III describes the process of the methods that need to fine-tune the cluster assignments. S denotes the predicted results and R is an adjustment of S.

JULE formulates the joint learning in a recurrent framework, where merging operations of agglomerative clustering are considered as a forward pass, and representation learning of DNN is considered as a backward pass. Based on this assumption, JULE also applies a loss that shrinks the inner cluster distance and expands the intracluster distance at the same time. In each epoch, JULE merges two clusters into one and computes the loss for the backward pass.

Chang et al. [65] propose deep adaptive image clustering (DAC) to tackle the combination of feature learning and clustering. In DAC, the clustering problem is reconstructed into binary pairwise classification problems that judge whether the pairwise images with estimated cosine similarities belong to the same cluster. Then, it adaptively selects similar samples to train DNN in a supervised manner. DAC provides a novel perspective for DC, but it only focuses on relationships between pairwise patterns. Deep discriminative clustering (DDC) analysis [54] is a more robust and generalized version of DAC by introducing global and local constraints of relationships. Spatial transformer-deep adaptive clustering (ST-DAC) [69] applies a visual attention mechanism [106] to modify the structure of DAC. Haeusser et al. [68] propose associative DC (ADC), which contains a group of centroid variables with the same shape as image embeddings. With the intuition that centroid variables can carry over high-level information about the data structure in the iteration process, they introduce an objective function with multiple loss terms to simultaneously train those centroid variables and the DNN's parameters along with a clustering mapping layer.

The abovementioned clustering methods estimate the cluster of an instance by passing it through the entire deep network, which tends to extract the global features of the instance [107]. Some clustering methods use a mature classification network to initialize the feature extractor. For instance, DeepCluster [66] applies k-means on the output features of the deep model (such as AlexNet [108] and VGG-16 [109]) and uses the cluster assignments as "pseudo-labels" to optimize the parameters of the CNNs. Hsu and Lin [67] propose clustering CNN (CCNN) that integrates mini-batch k-means with the model pretrained from the ImageNet dataset [110].

To improve the robustness of the model, more and more approaches make use of data augmentation for DC [49], [59], [76]. For example, Huang et al. [76] extend the idea of classical maximal margin clustering [111], [112] to establish a novel deep semantic clustering method [named PartItion Confidence mAximization (PICA)]. In PICA, three operations, including color jitters, random rescale, and horizontal flip, are adopted for data augmentation and perturbations.

Mutual information is also taken as a criterion to learn representations [113], [114] and has become popular in recent clustering methods, especially for image data. Various data augmentation techniques have been applied to generate transformed images that are used to mine their mutual information. For example, Ji et al. [71] propose invariant information clustering (IIC) for semantic clustering and image segmentation. In IIC, every image and its random transformation are treated as a sample pair. By maximizing mutual information between the clustering assignments of each pair, the proposed model can find semantically meaningful clusters and avoid degenerate solutions naturally. Instead of only using pairwise information, deep comprehensive correlation mining (DCCM) [72] is a novel image clustering framework, which uses pseudo-label loss as supervision information. Besides, the authors extend the instance-level mutual information and present triplet mutual information loss to learn more discriminative features. Based on the currently fashionable contrastive learning [115], Zhong et al. [75] propose deep robust clustering (DRC), where two contrastive loss terms are introduced to decrease intraclass variance and increase interclass variance. Mutual information and contrastive learning are related. In DRC, the authors summarize a framework that can turn maximize mutual information into minimizing contrastive loss.

In the field of image clustering on the semantic level, people think that the prediction of the original image should be consistent with that of the transformed image by data augmentation. Therefore, in the unsupervised learning context, data augmentation techniques not only are used to expand the training data but also can easily obtain supervised information. This is why data augmentation can be widely applied in many recently proposed image clustering methods. For example, Nina et al. [70] propose a decoder-free approach with data augmentation [called random triplet mining (RTM)] for clustering and manifold learning. To learn a more robust encoder, the model consists of three encoders with shared weights and is a triplet network architecture conceptually. The first and the second encoders take similar images generated by data augmentation as positive pair, and the second and the third encoders take a negative pair selected by RTM. Usually, the objective of triplet networks [116] is defined to make the features of the positive pair more similar and that of the negative pair more dissimilar.

Although many existing DC methods jointly learn the representations and clusters, such as JULE and DAC, there are specially designed representation learning methods [117], [118], [119], [120], [121] to learn the visual representations of images in a self-supervised manner. Those methods learn semantical representations by training deep networks to solve extra tasks. Such tasks can be predicting the patch context [117], inpainting patches [118], colorizing images [119], solving jigsaw puzzles [120], and predicting rotations [121], and so on. Recently, these self-supervised representation learning methods have been adopted in image clustering. For example, multimodal DC (MMDC) [73] leverages an auxiliary task of predicting rotations to enhance clustering performance. Semantic clustering by adopting nearest neighbors (SCAN) [74] first employs a self-supervised representation learning method to obtain semantically meaningful and high-level features. Then, it integrates the semantically meaningful nearest neighbors as prior information into a learnable clustering approach.

Since DEC [43] and JULE [63] are proposed to jointly learn feature representations and cluster assignments by DNNs, many *DAE-* and *DNN-based* DC methods have been proposed and have made great progress in clustering tasks. However, the feature representations extracted in clustering methods are difficult to extend to other tasks, such as generating samples. The deep generative models have recently attracted a lot of attention because they can use neural networks to obtain data distributions so that samples can be generated (VAE [122], GAN [123], Pixel-RNN [124], InfoGAN [125], and PPGN [126]). Specifically, GAN and VAE are the two most typical deep generative models. In recent years, researchers have applied them to various tasks, such as semi-supervised classification [127], [128], [129], [130], clustering [131], and image generation [132], [133]. In Sections III-C and III-D, we introduce the DC algorithms based on the generated models: *VAE-based* DC and *GAN-based* DC, respectively.

C. VAE-Based

Deep learning with nonparametric clustering (DNC) [134] is a pioneer work in applying deep belief networks to DC. However, in DC based on the probabilistic graphical model, more research comes from the application of VAE, which combines variational inference and DAE together.

Most *VAE-based* DC algorithms aim at solving an optimization problem about ELBO (see the deduction details in [122] and [135]), p is the joint probability distribution, q is the approximate probability distribution of $p(z|x)$, x is the input data for clustering, and z is the latent variable generated corresponding to x

$$L_{\text{ELBO}} = \mathbb{E}_{q(z|x)}\left[\log \frac{p(x,z)}{q(z|x)}\right]. \tag{6}$$

The difference is that different algorithms have different generative models of latent variables or different regularizers. We list several *VAE-based* DC methods that have attracted much attention in recent years as follows. For convenience, we omit the parameterized form of the probability distribution.

Traditional VAE generates a continuous latent vector z, and x is the vector of an original data sample. For the clustering task, the *VAE-based* methods generate latent vector (z, y), where z is the latent vector representing the embedding and y is the label. Thus, the ELBO for optimization becomes

$$L_{\text{ELBO}} = \mathbb{E}_{q(z,y|x)}\left[\log \frac{p(x,z,y)}{q(z,y|x)}\right]. \tag{7}$$

The first proposed unsupervised deep generative clustering framework is variational deep embedding (VaDE) [77]. VaDE models the data generative procedure with a Gaussian mixture model (GMM) [136] combining a VAE. In this method, the cluster assignments and the latent variables are jointly considered in a Gaussian mixture prior rather than a single Gaussian prior.

Similar to VaDE, Gaussian mixture VAE (GMVAE) [78] is another DC method that combines VAE with GMM. Specifically, GMVAE considers the generative model $p(x, z, n, c) = p(x|z)p(z|n, c)p(n)p(c)$, where c is uniformly distributed k categories and n is normally distributed. z is a continuous latent variable, whose distribution is a Gaussian mixture with means and variances of c and n. Based on the mean-field theory [137], GMVAE factors $q(z, n, c|x) = q(z|x)q(n|x)p(c|z, n)$ as posterior proxy. In the same way, those variational factors are parameterized with neural networks and the ELBO loss is optimized.

Based on GMM and VAE, latent tree VAE (LTVAE) [80] applies *latent tree model* [138] to perform representation learning and structure learning for clustering. Differently, LTVAE has a variant of VAE with a superstructure of latent variables. The superstructure is a tree structure of discrete latent variables on top of the latent features. The connectivity structure among all variables is defined as a latent structure of the *latent tree model* that is optimized via message passing [139].

The success of some deep generative clustering methods depends on good initial pre-training. For example, in VaDE [77], pre-training is needed to initialize cluster centroids. In DC via GMVAE with graph embedding (DGG) [140], pre-training is needed to initialize the graph embeddings. Although GMVAE [78] learns the prior and posterior parameters jointly, the prior for each class is dependent on a random variable rather than the class itself, which seems counter-intuitive. Based on the ideas of GMVAE and VaDE, to solve their fallacies, Prasad et al. [82] propose a new model leveraging VAE for image clustering (VAEIC). Different from the methods mentioned above, the prior of VAEIC is deterministic, and the prior and posterior parameters are learned jointly without the need for a pre-training process. Instead of performing Bayesian classification as done in GMVAE and VaDE, VAEIC adopts more straightforward inference and more principled latent space priors, leading to a simpler inference model $p(x, z, c) = p(x|z)p(z|c)p(c)$ and a simpler approximate posterior $q(z, c|x) = q(c|x)q(z|x, c)$. The cluster assignment is directly predicted by $q(c|z)$. What is more, the authors adopt data augmentation and design an image augmentation loss to make the model robust.

In addition to the *VAE-based* DC methods mentioned above, Figueroa and Rivera [79] use the continuous Gumbel-Softmax distribution [141], [142] to approximate the categorical distribution for clustering. Willetts et al. [81] extend variational ladder autoencoders [143] and propose a disentangled clustering algorithm. Cao et al. [83] propose a simple, scalable, and stable variational DC algorithm, which introduces generic improvements for variational DC.

D. GAN-Based

In adversarial learning, standard GANs [123] are defined as an adversarial game between two networks: generator ϕ_g and discriminator ϕ_d. Specifically, the generator is optimized to generate fake data that "fool" the discriminator, and the discriminator is optimized to tell apart real from fake input data, as shown in Fig. 3.

GAN has already been widely applied in various fields of deep learning. Many DC methods also adopt the idea of adversarial learning due to their strength in learning the latent distribution of data. We summarize the important *GAN-based* DC methods as follows. Probabilistic clustering algorithms

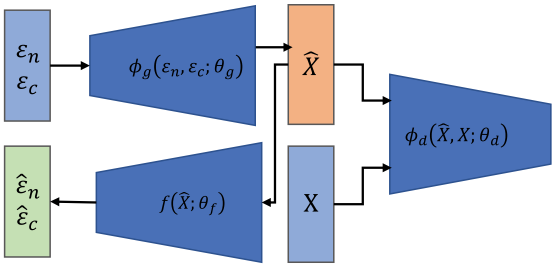

ε_n
ε_c
$\phi_g(\varepsilon_n, \varepsilon_c; \theta_g)$
$\widehat{X}$
$\phi_d(\widehat{X}, X; \theta_d)$
$\hat{\varepsilon}_n$
$\hat{\varepsilon}_c$
$f(\widehat{X}; \theta_f)$
X

Fig. 3. Framework of *GAN-based* learning. ϕ_g is the generator, ϕ_d is the discriminator, both ε_n and ε_c are inputs to the generator, ε_n is the noise, and ε_n is the class information. X is the data for clustering, $\widehat{X}$ is the fake data that "fools' the discriminator, and the function $f(\cdot)$ operates on $\widehat{X}$ to generate $\widehat{\varepsilon_n}$ and $\widehat{\varepsilon_c}$.

address many unlabeled data problems, such as regularized information maximization (RIM) [144] or the related entropy minimization [145]. The main idea of RIM is to train a discriminative classifier with unlabeled data. Unfortunately, these methods are prone to overfitting spurious correlations. Springenberg [86] propose categorical GANs (CatGAN) to address this weakness. To make the model more general, GAN is introduced to enhance the robustness of the classifier. In CatGAN, all real samples are assigned to one of the k categories using the discriminator while staying uncertain of clustering assignments for samples from the generative model rather than simply judging the false and true samples. In this way, the GAN framework is improved so that the discriminator can be used for multiclass classification. In particular, CatGAN can be applied to both unsupervised and semi-supervised tasks.

Interpretable representation learning in the latent space has been investigated in the seminal work of InfoGAN [125]. Although InfoGAN does use discrete latent variables, it is not specifically designed for clustering. VAE [122] can jointly train the inference network and autoencoder, which enables mapping from initial sample X to latent space Z that could potentially preserve cluster structure. Unfortunately, there is no such inference mechanism in GAN. To make use of their advantages, Mukherjee et al. [90] propose ClusterGAN as a new mechanism for clustering. ClusterGAN samples latent variables from a mixture of one-hot variables and continuous variables and establishes a reverse-mapping network to project data into a latent space. It jointly trains a GAN along with the inverse-mapping network with a clustering-specific loss to achieve clustering.

There is another *GAN-based* DC method [89] (we denote it as ClusterGAN-SPL) that has a similar network module with ClusterGAN. The main difference is that ClusterGAN-SPL does not set discrete latent variables but applies self-paced learning [100] to improve the robustness of the algorithm.

In some *GAN-based* DC methods (e.g., deep adversarial Gaussian mixture autoencoder for clustering (DAGC) [87], deep adversarial subspace clustering (DASC) [88], adversarial graph autoencoder (AGAE) [94], and adversarial DEC (ADEC) [91]), GAN and DAE are both applied. For example, inspired by the adversarial autoencoders [131] and GAN [123], Harchaoui et al. [87] propose DAGC. To make the data representations easier to cluster than in the initial space, it builds an autoencoder [146] consisting of an encoder and a decoder. In addition, an adversarial discriminator is added to continuously force the latent space to follow the Gaussian mixture prior [136]. This framework improves the performance of clustering due to the introduction of adversarial learning.

Most existing subspace clustering approaches ignore the inherent errors of clustering and rely on the self-expression of handcrafted representations. Therefore, their performance on real data with complex underlying subspaces is not satisfactory. Zhou et al. [88] propose DASC to alleviate this problem and apply adversarial learning to deep subspace clustering. DASC consists of a generator and a discriminator that learn from each other. The generator outputs subspace clustering results and consists of an autoencoder, a self-expression layer, and a sampling layer. The DAE and self-expression layer are used to convert the original input samples into better representations. In the pipeline, a new "fake" sample is generated by sampling from the estimated clusters and sent to the discriminator to evaluate the quality of the subspace cluster.

Many autoencoder-based clustering methods use reconstruction for pretraining and let reconstruction loss be a regularizer in the clustering phase. Mrabah et al. [91] point out that such a tradeoff between clustering and reconstruction would lead to feature drift phenomena. Hence, the authors adopt adversarial training to address the problem and propose ADEC. It first pretrains the autoencoder, where reconstruction loss is regularized by an adversarially constrained interpolation [147]. Then, the cluster loss (similar to DEC [43]), reconstruction loss, and adversarial loss are optimized in turn. ADEC can be viewed as a combination of DEC and adversarial learning.

Besides the abovementioned methods, there are a small number of DC methods whose used networks are difficult to categorize. For example, information maximizing self-augmented training (IMSAT) [113] uses very simple networks to perform unsupervised discrete representation learning. SpectralNet [148] is a deep learning method to approximate spectral clustering, where unsupervised Siamese networks [149], [150] are used to compute distances. In clustering tasks, it is a common phenomenon to adopt the appropriate neural network for different data formats. In this survey, we focus more on deep learning techniques that are reflected in the used systematic neural network structures.

E. GNN-Based

GNNs [151], [152] allow end-to-end differentiable losses over data with arbitrary graph structure and have been applied to a wide range of applications. Many tasks in the real world can be described as a graph, such as social networks, protein structures, and traffic networks. With the suggestion of Banach's fixed point theorem [153], GNN uses the following classic iterative scheme to compute the state. F is a global transition function, and the value of H is the fixed point of $H = F(H, X)$ and is uniquely defined with the assumption that F is a contraction map [154]

$$H^{t+1} = F\left(H^t, X\right). \tag{8}$$

In the training process of GNN, many methods try to introduce attention and gating mechanism into a graph structure. Among these methods, graph convolutional network (GCN) [155], which utilizes the convolution for information aggregation, has gained remarkable achievement. H is the node hidden feature matrix, W is the learnable model parameters, and C is the feature matrix of a graph. The compact form of GCN is defined as

$$H = \widetilde{D}^{-\frac{1}{2}}\widetilde{Q}\widetilde{D}^{-\frac{1}{2}}CW. \tag{9}$$

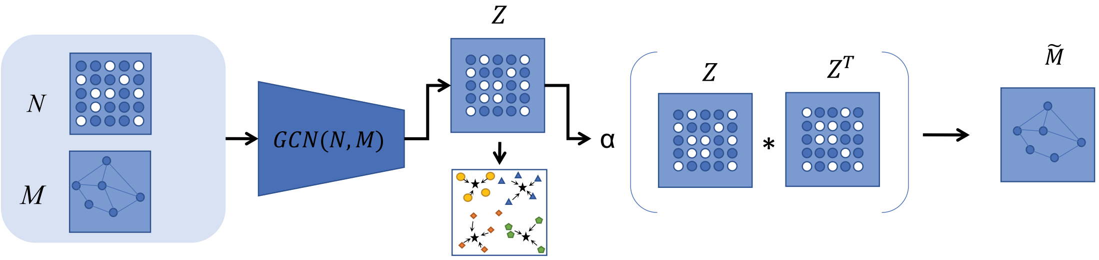


Fig. 4. Data stream framework of graph autoencoder applied in clustering. GCN(N, M) is a graph autoencoder, GCN($\cdot$) is used to represent a graph CNN, and graph autoencoder consists of two layers of graph CNNs. Both node attributes N and graph structure M are utilized as inputs to this encoder. Z is a matrix of node embedding vectors. α is an activation function and $\widetilde{M}$ is the prediction of graph adjacency matrix M.

In the domain of unsupervised learning, there are also a variety of methods trying to use the powerful structure capturing capabilities of GNNs to improve the performance of clustering algorithms. We summarize the *GNN-based* DC methods as follows.

Tian et al. [156] propose learning deep representations for graph clustering (DRGC) to replace traditional spectral clustering with sparse autoencoder and k-means algorithm. In DRGC, sparse autoencoder is adopted to learn nonlinear graph representations that can approximate the input matrix through reconstruction and achieve the desired sparse properties. The last layer of the deep model outputs a sparse encoding and k-means serves as the final step on it to obtain the clustering results. To accelerate graph clustering, Shao et al. [157] propose deep linear coding (DLC) for fast graph clustering. Unlike DRGC, DLC does not require eigendecomposition and greatly saves running time on large-scale datasets while still maintaining a low-rank approximation of the affinity graph.

The research on GNNs is closely related to graph embedding or network embedding [158], [159], [160], as GNNs can address the network embedding problem through a graph autoencoder framework [161]. The purpose of graph embedding [162] is to find low-dimensional features that maintain similarity between the vertex pairs in a sample similarity graph. If two samples are connected in the graph, their latent features will be close. Thus, they should also have similar cluster assignments. Based on this motivation, Yang et al. [140] propose DGG. Like VaDE [77], the generative model of DGG is $p(x, z, c) = p(x|z)p(z|c)p(c)$. The prior distributions of z and c are set as a Gaussian mixture distribution and a categorical distribution, respectively. The learning problem of GMM-based VAE is usually solved by maximizing the ELBO of the log-likelihood function with *reparameterization trick*. To achieve graph embedding, the authors add a graph embedding constraint to the original optimization problem, which exists not only on the features but also on the clustering assignments. Specifically, the similarity between data points is measured with a trained Siamese network [149].

Autoencoder also works on graphs as an effective embedding method. In AGAEs, Tao et al. [94] apply ensemble clustering [16], [163] in the deep graph embedding process and develop an adversarial regularizer to guide the training of the autoencoder and discriminator. Recent studies have mostly focused on the methods that are two-step approaches. The drawback is that the learned embedding may not be the best fit for the clustering task. To address this, Wang et al. [164] propose a unified approach named deep attentional embedded graph clustering (DAEGC). DAEGC develops a graph attention-based autoencoder to effectively integrate both structure and content information, thereby achieving better clustering performance. The data stream framework of graph autoencoder is applied in clustering in Fig. 4.

TABLE III
SEMI-SUPERVISED DEEP CLUSTERING METHODS

Methods	Characteristics
SDEC (2019) [165]	Based on DEC [43].
SSLDEC (2019) [166]	Based on DEC [43].
DECC (2019) [167]	Based on DEC [43].
SSCNN (2020) [168]	Combine k-means loss and pairwise divergence.
GDAN (2022) [169]	Utilize the high-level semantic features.

As one of the most successful feature extractors for deep learning, CNNs are mainly limited by Euclidean data. GCNs have proved that graph convolution is effective in DC, e.g., Zhang et al. [93] propose an adaptive graph convolution (AGC) method for attributed graph clustering. AGC exploits high-order graph convolution to capture global cluster structure and adaptively selects the appropriate order for different graphs. Nevertheless, AGC might not determine the appropriate neighborhood that reflects the relevant information of connected nodes represented in graph structures. Based on AGC, Zhu et al. [95] exploit heat kernel to enhance the performance of graph convolution and propose AGC using heat kernel (AGCHK), which could make the low-pass performance of the graph filter better.

In summary, we can realize the importance of the structure of data. Motivated by the great success of GNNs in encoding the graph structure, Bo et al. [96] propose a structural DCN (SDCN). By stacking multiple layers of GNN, SDCN is able to capture the high-order structural information. At the same time, benefiting from the self-supervision of AE and GNN, the multilayer GNN does not exhibit the so-called oversmooth phenomenon. SDCN is the first work to apply structural information to DC explicitly.

IV. SEMI-SUPERVISED DEEP CLUSTERING

Traditional semi-supervised learning can be divided into three categories, i.e., semi-supervised classification [170], [171], semi-supervised dimension reduction [172], [173], and semi-supervised clustering [13], [174], [175]. Commonly, the constraint of unsupervised data is marked as "must-link" and "cannot-link." Samples with the "must-link" constraint belong to the same cluster, while samples with the "cannot-link" constraint belong to different clusters. Most semi-supervised clustering objectives are the combination of unsupervised clustering loss and constraint loss.

Semi-supervised DC has not been explored well. Here, we introduce several representative works. These works use different ways to combine the relationship constraints and the neural networks to obtain better clustering performance. We summarize these methods in Table III.

Semi-supervised DEC (SDEC) [165] is based on DEC [43] and incorporates pairwise constraints in the feature learning process. Its loss function is defined as

$$\text{Loss} = \text{KL}(S\|R) + \lambda\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{n} a_{ij}\left\|z_i - z_j\right\|^2 \tag{10}$$

where λ is a tradeoff parameter. $a_{ij} = 1$ if x_i and x_j are assigned to the same cluster, and $a_{ij} = -1$ if x_i and x_j satisfy cannot-link constraints, $a_{ij} = 0$ otherwise. As the loss function shows, it is formed by two parts. The first part is KL divergence loss, which has been explained in Section III-A. The second part is semi-supervised loss that denotes the consistency between the embedded feature $\{z_i\}_{i=1}^{n}$ and parameter a_{ij}. Intuitively, if $a_{ij} = 1$, to minimize the loss function, $\|z_i - z_j\|^2$ should be small. In contrast, if $a_{ij} = -1$, to minimize the loss, $\|z_i - z_j\|^2$ should be large, which means that z_i is apart from z_j in the latent space Z.

Like SDEC, most semi-supervised DC methods are based on unsupervised DC methods. It is straightforward to expand an unsupervised DC method to a semi-supervised DC one by adding the semi-supervised loss. Compared with unsupervised DC methods, the extra semi-supervised information of data can help the neural network to extract features more suitable for clustering. There are also some works focusing on extending the existing semi-supervised clustering method to a deep learning version. For example, the feature extraction process of both semi-supervised learning with DEC (SSLDEC) for image classification and segmentation [166] and deep constrained clustering (DECC) [167] is based on DEC. Their training process is similar to semi-supervised k-means [174], which learns feature representations by alternatively using labeled and unlabeled data samples. During the training process, the algorithms use labeled samples to keep the model consistent and choose a high degree of confidence unlabeled samples as newly labeled samples to tune the network. Semi-supervised clustering with neural networks [168] combines a k-means loss and pairwise divergence to simultaneously learn the cluster centers as well as semantically meaningful feature representations. GDAN [169] acquires domain-invariant features via a pretext task, employing instance discrimination criteria. Subsequently, GDAN aligns the two domains by exclusively focusing on high-level semantic features through the clustering of semantic neighbors.

V. Deep MVC

The abovementioned DC methods can only deal with single-view data. In practical clustering tasks, the input data usually have multiple views. For example, the report of the same topic can be expressed in different languages, the same dog can be captured from different angles by cameras, and the same word can be written by people with different writing styles. MVC methods [18], [176], [177], [178], [179], [180], [181], [182], [183], [184], [185] are proposed to make use of the complementary information among multiple views to improve clustering performance.

In recent years, the application of deep learning in MVC has been a hot topic [186], [187], [188], [189], [190]. Those deep MVC algorithms focus on solving clustering problems with different forms of input data. Since the network structures used in most of these methods are autoencoders, we divided them into three categories based on the adopted clustering theoretical basis: *DEC-based*, *subspace clustering-based*, and *GNN-based*. They are summarized in Table IV.

A. DEC-Based

As mentioned previously, DEC [43] uses autoencoder to learn the low-dimensional embedded feature representation and then minimizes the KL divergence of Student's t-distribution and auxiliary target distribution of feature representations to achieve clustering. IDEC [44] emphasizes data structure preservation and adds the term of the reconstruction loss for the lower dimensional feature representation when processing fine-tuning tasks. Some deep MVC methods also adopt this deep learning pipeline.

Traditional MVC methods mostly use linear and shallow embedding to learn the latent structure of multiview data. These methods cannot fully utilize the nonlinear property of data, which is vital to reveal a complex clustering structure. Based on adversarial learning and DAE, Li et al. [191] propose deep adversarial MVC (DAMC) to learn the intrinsic structure embedded in multiview data. Specifically, DAMC consists of a multiview encoder E, a multiview generator (decoder) ϕ_g, V discriminators $D_1, \ldots, D_V$ (V denotes the number of views), and a DEC layer. The multiview encoder outputs low-dimensional embedded features for each view. For each embedded feature, the multiview generator generates the corresponding reconstruction sample. The discriminator is used to identify the generated sample from the real sample and output feedback. The total loss function of DAMC is defined as

$$\text{Loss} = \min_{E,G}\max_{D_1,\ldots,D_V} L_r + \alpha L_c + \beta L_{\text{GAN}} \tag{11}$$

where L_c comes from DEC [43] and represents the clustering loss; L_r and L_{GAN} represent the reconstruction loss and GAN loss, respectively; and α and β are hyperparameters. Compared with traditional MVC algorithms, DAMC can reveal the nonlinear property of multiview data and achieve better clustering performance.

Xu et al. [186] propose a novel collaborative training framework for deep-embedded MVC (DEMVC). Specifically, DEMVC defines a switched shared auxiliary target distribution and fuses it into the overall clustering loss. Its main idea is that by sharing optimization objectives, each view, in turn, guides all views to learn the low-dimensional embedded features that are conducive to clustering. At the same time, optimizing reconstruction loss makes the model retain discrepancies among multiple views. Experiments show that DEMVC can mine the correct information contained in multiple views to correct other views, which helps improve the clustering accuracy. Existing methods tend to fuse multiple views' representations, and Xu et al. [193] present a novel *VAE-based* MVC framework (Multi-VAE) by learning disentangled visual representations.

Lin et al. [194] propose a contrastive multiview hyperbolic hierarchical clustering (CMHHC). It consists of three components, multiview alignment learning, aligned feature similarity learning, and continuous hyperbolic hierarchical clustering. Through capturing the invariance information across views and learning the meaningful metric property for similarity-based continuous hierarchical clustering, CMHHC is capable of clustering multiview data at diverse levels of granularity. Xu et al. [195] propose a framework of multilevel feature learning for contrastive MVC (MFLVC), which combines

TABLE IV
SUMMARIES OF DEEP MVC METHODS

Networks	Methods	Characteristics
DAE + GAN	DAMC (2019) [191]	Capture the data distribution ulteriorly by adversarial training.
VAE	DMVCVAE (2020) [192]	Learn a shared latent representation under the VAE framework.
DAE	DEMVC (2021) [186]	Through collaborative training, each view can guide all views.
DAE	DMVSSC (2018) [188]	Extract multi-view deep features by CCA-guided convolutional auto-encoders.
DAE	RMSL (2019) [189]	Recover the underlying low-dimensional subspaces in which the high dimensional data lie.
DAE	MVDSCN (2019) [190]	Combine convolutional auto-encoder and self-representation together.
VAE	Multi-VAE (2021) [193]	Learn disentangle and explainable representations.
DAE	CMHHC (2022) [194]	Employ multiple autoencoders and hyperbolic hierarchical clustering.
DAE	MFLVC (2022) [195]	Utilize contrastive clustering to learn the common semantics across all views.
DAE	DIMVC (2022) [196]	Imputation-free and fusion-free incomplete multi-view clustering.
GCN	Multi-GCN (2019) [197]	Incorporates nonredundant information from multiple views.
GCN	MAGCN (2020) [198]	Dual encoders for reconstructing and integrating.
GAE	O2MAC (2020) [187]	Partition the graph into several nonoverlapping clusters.
GAE	CMGEC (2021) [199]	Multiple graph autoencoder.
GAE	DMVCJ (2022) [200]	Weighting strategy to alleviate the noisy issue.

MVC with contrastive learning to improve clustering effectiveness. MFLVC can learn different levels of features and reduce the adverse influence of view-private information.

For the incomplete multiview data, the absence of some views will increase the difficulties of information excavation and lead to the failure of most conventional MVC methods based on the assumption of view completion [201], [202]. Lin et al. [203] design to recover the missing data with contrastive learning. Xu et al. [196] also explore incomplete MVC, through mining the complementary information in the high-dimensional feature space via a nonlinear mapping of multiple views. The proposed method is an imputation-free and fusion-free deep IMVC (DIMVC) framework that can handle the incomplete data primely.

B. Subspace Clustering-Based

Subspace clustering [204] is another popular clustering method, which holds the assumption that data points of different clusters are drawn from multiple subspaces. Subspace clustering typically first estimates the affinity of each pair of data points to form an affinity matrix and then applies spectral clustering [205] or a normalized cut [206] on the affinity matrix to obtain clustering results. Some subspace clustering methods based on self-expression [207] have been proposed. The main idea of self-expression is that each point can be expressed with a linear combination C of the data points X themselves. The general objective is

$$\text{Loss} = L_r + \alpha R(C) = \|X - XC\| + \alpha R(C) \tag{12}$$

where $\|X - XC\|$ is the reconstruction loss and $R(C)$ is the regularization term for subspace representation C. In recent years, a lot of works [208], [209], [210], [211], [212], [213], [214] generate a good affinity matrix and achieve better results by using the self-expression methodology.

There are also MVC methods [178], [180], [183], which are based on subspace learning. They construct the affinity matrix with shallow features and lack of interaction across different views, thus resulting in insufficient use of complementary information included in multiview datasets. To address this, researchers focus more on multiview subspace clustering methods based on deep learning recently.

Exploring the consistency and complementarity of multiple views is a long-standing important research topic of MVC [215]. Tang et al. [188] propose the deep multiview sparse subspace clustering (DMVSSC), which consists of a canonical correlation analysis (CCA)-based [216], [217], [218] self-expressive module and CAEs. The CCA-based self-expressive module is designed to extract and integrate deep common latent features to explore the complementary information of multiview data. A two-stage optimization strategy is used in DMVSSC. First, it only trains CAEs of each view to obtain suitable initial values for parameters. Second, it fine-tunes all the CAEs and CCA-based self-expressive modules to perform MVC.

Unlike CCA-based deep MVC methods (e.g., DMVSSC [188]) which project multiple views into a common low-dimensional space, Li et al. [189] present a novel algorithm named reciprocal multilayer subspace learning (RMSL). RMSL contains two main parts: hierarchical self-representative layers (HSRLs) and backward encoding networks (BENs). The self-representative layers (SRLs) contain the view-specific SRL, which maps view-specific features into view-specific subspace representations, and the common SRL, which further reveals the subspace structure between the common latent representation and view-specific representations. BEN implicitly optimizes the subspaces of all views to explore consistent and complementary structural information to get a common latent representation.

Many multiview subspace clustering methods first extract handcrafted features from multiple views and then learn the affinity matrix jointly for clustering. This independent feature extraction stage may lead to the multiview relations in data being ignored. To alleviate this problem, Zhu et al. [190] propose a novel multiview deep subspace clustering network (MVDSCN), which consists of diversity net (Dnet) and universality net (Unet). Dnet is used to learn view-specific self-representation matrices and Unet is used to learn a common self-representation matrix for multiple views. The loss function is made up of the reconstruction loss of autoencoders, the self-representation loss of subspace clustering, and multiple well-designed regularization items.

C. GNN-Based

In the real world, graph data are far more complex. For example, we can use text, images, and links to describe the same web page, or we can ask people with different styles to write the same number. Obviously, traditional single-view clustering methods are unable to meet the needs of such application scenarios, that is, one usually needs to employ a multiview graph [219], rather than a single-view graph, to better represent the real graph data. Since GCN has made considerable achievements in processing graph-structured data, Khan and Blumenstock [197] develop a graph-based convolutional network (Multi-GCN) for multiview data. Multi-GCN focuses attention on integrating subspace learning approaches with recent innovations in GCNs and proposes an efficient method for adapting graph-based semi-supervised learning (GSSL) to multiview contexts.

Most GNNs can effectively process single-view graph data, but they cannot be directly applied to multiview graph data. Cheng et al. [198] propose multiview attribute graph convolution networks for clustering (MAGCN) to handle graph-structured data with multiview attributes. The main innovative method of MAGCN is designed with two-pathway encoders. The first pathway develops multiview attribute graph attention networks to capture the graph embedding features of multiview graph data. Another pathway develops consistent embedding encoders to capture the geometric relationship and the consistency of probability distribution among different views.

Fan et al. [187] attempt to employ deep-embedded learning for multiview graph clustering. The proposed model is named One2Multi graph autoencoder for multiview graph clustering (O2MAC), which utilizes graph convolutional encoder of one view and decoders of multiple views to encode the multiview attributed graphs to a low-dimensional feature space. Both the clustering loss and reconstruction loss of O2MAC are similar to other DEC methods in form. What is special is that GCN [155] is designed to deal with graph clustering tasks [220]. Huang et al. [200] propose deep-embedded MVC via jointly learning latent representations and graphs (DMVCJ). By introducing a self-supervised GCN module, DMVCJ jointly learns both latent graph structures and feature representations.

The graph in most existing GCN-based MVC methods is fixed, which makes the clustering performance heavily dependent on the predefined graph. A noisy graph with unreliable connections can result in ineffective convolution with wrong neighbors on the graph [221], which may worsen the performance. To alleviate this issue, Wang et al. [199] propose a consistent multiple graph embedding clustering (CMGEC) framework, which is mainly composed of multiple graph autoencoder (M-GAE), multiview mutual information maximization module (MMIM), and graph fusion network (GFN). CMGEC develops a multigraph attention fusion encoder to adaptively learn a common representation from multiple views, and thereby, CMGEC can deal with three types of multiview data, including multiview data without a graph, multiview data with a common graph, and single-view data with multiple graphs.

According to our research, deep MVC algorithms have not been explored well. Other than the abovementioned three categories, Yin et al. [192] propose a *VAE-based* deep MVC method [deep MVC via VAEs (DMVCVAE)]. DMVCVAE learns a shared generative latent representation that obeys a mixture of Gaussian distributions and thus can be regarded as the promotion of VaDE [77] in MVC. There are also some application researches based on deep MVC. For example, Perkins and Yang [222] introduce the dialog intent induction task and present a novel deep MVC approach to tackle the problem. Abavisani and Patel [223] and Hu et al. [224] study multimodal clustering, which is also related to MVC. Taking advantage of both DC and multiview learning will be an interesting future research direction of deep MVC.

VI. Deep Clustering With Transfer Learning

Transfer learning has emerged as a new learning framework to address the problem that the training and testing data are drawn from different feature spaces or distributions [225]. For complex data such as high-resolution real pictures of noisy videos, traditional clustering methods even DC methods cannot work very well because of the high dimensionality of the feature space and no uniform criterion to guarantee the clustering process. Transfer learning provides new solutions to these problems through transferring the information from source domain that has additional information to guide the clustering process of the target domain. In the early phase, the ideas of deep domain adaption are simple and clear, such as deep reconstruction-classification networks (DRCNs) [226] that use classification loss for the source domain and reconstruction loss for target domain. The two domains share the same feature extractor. With the development of DNN, we now have more advanced ways to transfer knowledge.

In this section, we introduce some transfer learning work about clustering, which is separated into two parts. The first part is “*DNN-based*,” and the second part is “*GAN-based*.” They are summarized in Table V.

A. DNN-Based

DNN-based UDA methods generally aim at projecting the source and target domains into the same feature space, in which the classifier trained with source embedding and labels can be applied to the target domain.

Through a summary of the network training processes, Yosinski et al. [227] find that many DNNs trained on natural images exhibit a phenomenon in common: the features learned in the first several layers appear not to be specific to a particular dataset or task and applicable to many other datasets or tasks. Features must eventually transition from general to specific by the last layers of the network. Thus, we can use a mature network (e.g., AlexNet [108] and GoogleNet [228]), which can provide credible parameters as the initialization for a specific task. This trick has been frequently used in feature extracted networks.

Domain adaptive neural network (DaNN) [229] first used maximum mean discrepancy (MMD) [230] with DNN.

Many domain-discrepancy-based methods adopt similar techniques with DaNN. Deep adaption networks (DANs) [231] use multiple kernel variants of MMD (MK-MMD) as its domain adaption function. As shown in Fig. 5, the net of DAN minimizes the distance at the last feature-specific layers, and then, the features from source-net and target-net would be projected into the same space. After DAN, more and more methods based on MMD are proposed. The main optimization way is to choose different versions of MMD, such as joint adaption network (JAN) [232] and weighted DAN (WDAN) [233]. JAN maximizes joint MMD to make the distributions of both source and target domains more distinguishable. WDAN is proposed to solve the question about imbalanced data distribution by introducing an auxiliary weight for each class in the source domain. RTN (unsupervised domain adaptation with residual transfer networks) [234] uses residual networks and MMD for UDA task.

Some discrepancy-based methods do not use MMD. Domain adaptive hash (DAH) [235] uses supervised hash loss and unsupervised entropy loss to align the target hash values

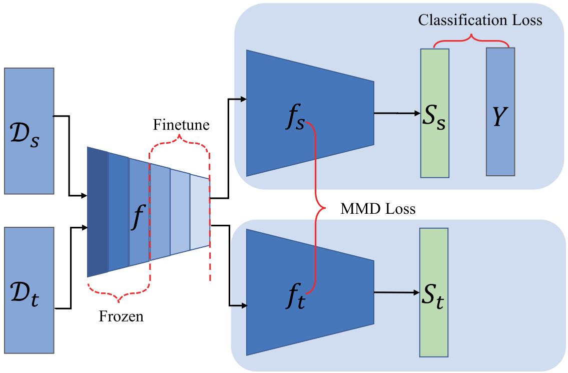


Fig. 5. Data stream framework of DAN. $\mathcal{D}_s$ is the source domain. $\mathcal{D}_t$ is the target domain. f is the shared encoder of both domains, which can be initialized with the existing network. The first layers of f are frozen, and the last layers of f can be fine-tuned in the training process. f_s is the encoder of $\mathcal{D}_s$. f_t is the encoder of $\mathcal{D}_t$. S_s is the predicted label vector of $\mathcal{D}_s$. Y is the real label of $\mathcal{D}_s$. S_t is the predicted result of $\mathcal{D}_t$.

to their corresponding source categories. Sliced Wasserstein discrepancy (SWD) [236] adopts the novel SWD to capture the dissimilarity of probability. Correlation alignment (CORAL) [237] minimizes domain shift by aligning the second-order statistics of source and target distributions. Higher order moment matching (HoMM) [238] shows that the first-order HoMM is equivalent to MMD and the second-order HoMM is equivalent to CORAL. Contrastive adaptation network (CAN) [239] proposes contrastive domain discrepancy (CDD) to minimize the intraclass discrepancy and maximize the interclass margin. Besides, several new measurements are proposed for the source and target domain [240], [241], [242]. Analysis of representations for domain adaptation [243] contributes a lot in the domain adaption distance field. Some works try to improve the performance of UDA in other directions, such as unsupervised domain adaptation via structured prediction-based selective pseudo-labeling that tries to learn a domain-invariant subspace by supervised locality-preserving projection (SLPP) using both labeled source data and pseudo-labeled target data.

The tricks used in DC have also been used in UDA methods. For example, structurally regularized DC (SRDC) [244] implements the structural source regularization via a simple strategy of joint network training. It first minimizes the KL divergence between the auxiliary distribution (that is the same as the auxiliary distribution of DEC [43]) and the predictive label distribution. Then, it replaces the auxiliary distribution with that of ground-truth labels of source data. Wang and Breckon [245] propose a UDA method that uses a novel selective pseudo-labeling strategy and learns domain-invariant subspace by SLPP [246] using both labeled source data and pseudo-labeled target data. Zhou et al. [247] apply ensemble learning in the training process. Prabhu et al. [248] apply entropy optimization in the target domain.

B. GAN-Based

DNN-based UDA methods mainly focus on an appropriate measurement for the source and target domains. By contrast, *GAN-based* UDA methods use the discriminator to fit this measurement function. Usually, in *GAN-based* UDA methods, the generator ϕ_g is used to produce data followed by one distribution from another distribution, and the discriminator ϕ_d is used to judge whether the data generated follow the distribution of the target domain. Traditional GAN cannot satisfy the demands to project two domains into the same space, so different frameworks based on GAN are proposed to cope with this challenge.

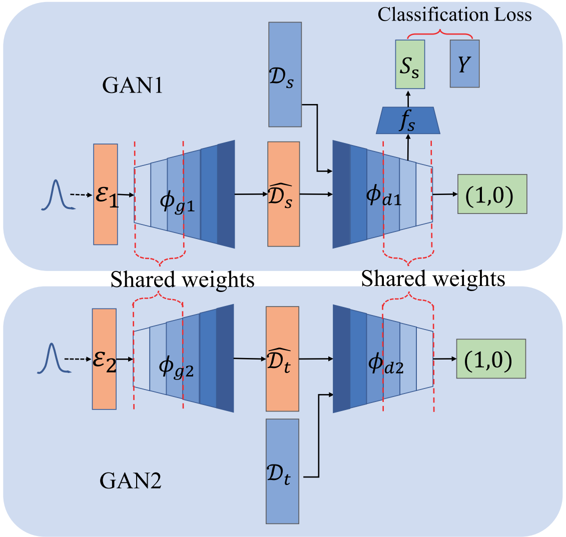


Fig. 6. Data stream framework of Co-GAN applied in UDA. It consists of a pair of GANs: GAN1 and GAN2. GAN1 and GAN2 share the weight in the first layers of ϕ_g and last layers of ϕ_d. $\mathcal{D}_s$ is the source domain. $\mathcal{D}_t$ is the target domain. ϕ_d. $\widehat{\mathcal{D}_s}$ and ϕ_d. $\widehat{\mathcal{D}_t}$ are generated by the noise. The first layer of ϕ_g is responsible for decoding high-level semantics and the last layer of ϕ_d is responsible for encoding high-level semantics. Adding weight-sharing constraint in these layers can guarantee similar high-level semantic representations of both domains with different low-level feature representations.

In 2016, domain-adversarial neural network (DANN) [255] and coupled GANs (Co-GANs) [254] are proposed to introduce adversarial learning into transfer learning. DANN uses a discriminator to ensure that the feature distributions over the two domains are made similar. Co-GAN applies generator and discriminator all in UDA methods. It consists of a group of GANs, each corresponding to a domain. In UDA, there are two domains. The framework of Co-GAN is shown in Fig. 6.

In deep transfer learning, we need to find the proper layers for MMD or weight sharing. In general, we could see that the networks that want to transfer knowledge through domain adaption must pay more attention to the layers that are responsible for high-level semantic layers. In DAN, the first layers are for basic features and the high layers for semantic information are zoomed in where the last layers are chosen to be projected with MMD. In Co-GAN, also the semantic layers are chosen as the transferring layers (take notice, the first layers of DAN are not transferring layers between two domains, as it is transferring the feature extracting power of a mutual network to our domains' feature extracting part). The weight-sharing constraint in the first layers of the generator urges two instances from a different domain to extract the same semantics and are destructed into different low-level details in the last layers of ϕ_g. On opposite, the discriminator learns the features from low level to high level, so if we add a weight-sharing constraint in the last layers, this can stimulate it to learn a joint distribution of multidomain images from different low-level representations.

Co-GAN contributed significant thought to UDA. Adversarial methods in domain adaptation have sprung up. For the job that relies on the synthesized instances to assist the domain

TABLE V

SUMMARIES OF *DNN*- AND *GAN-Based* METHODS IN DEEP CLUSTERING WITH TRANSFER LEARNING

Net	Methods	Characteristics
DNN	DaNN (2014) [229]	MMD and the same feature extracter.
	DAN (2015) [231]	Multi-kernel MMD. Different feature extracters.
	DRCN (2016) [226]	Classification of source and reconstruction of target.
	RTN (2016) [234]	Residual networks and MMD.
	DAH (2017) [235]	Supervised hash loss and unsupervised entropy loss.
	WDAN (2017) [233]	Imbalanced data distribution.
	JAN (2017) [232]	Joint MMD.
	CORAL (2017) [237]	Minimize domain shift by aligning the second-order statistics of source and target distributions.
	SWD (2019) [236]	Sliced Wasserstein discrepancy.
	CAN (2019) [239]	Contrastive Domain Discrepancy.
	SRDC (2020) [244]	KL divergence and auxiliary distribution (the same with DEC [43].).
	SPL (2020) [245]	Supervised locality preserving projection and selective pseudo-labeling strategy
	MDD (2020) [249]	Within-domain class imbalance and between-domain class distribution shift.
	HoMM (2020) [238]	Higher-order moment matching for UDA.
	GSDA (2020) [240]	Model the relationship among the local distribution pieces and global distribution synchronously.
	ETD(2020) [241]	Attention mecanism for samples similarity andattention scores for the transport distances.
	BAIT (2020) [250]	Source-free unsupervised domain adaptation.
	DAEL (2021) [247]	Ensemble Learning.
	SHOT (2021) [251]	Source-free unsupervised domain adaptation.
	SHOT-plus (2021) [252]	Source-free unsupervised domain adaptation.
	SENTRY (2021) [248]	Entropy Optimization.
	RWOT (2021) [242]	Shrinking Subspace Reliability and weighted optimal transport strategy.
	N2DC-EX (2021) [253]	Source-free unsupervised domain adaptation.
GAN	Co-GAN (2016) [254]	A group of GANs with partly weight sharing, discriminator and label predictor are unified.
	DANN (2016) [255]	Domain classifier and label predictor.
	UNIT (2017) [256]	Use variational autoencoder as feature extractor.
	ADDA(2017) [257]	Generalization of Co-GAN [254].
	PixelDA (2017) [258]	Generate instances follow target distribution with source samples.
	GenToAdapt (2018) [259]	Two classifiers and one encoder to embed the instances into vectors.
	SimNet (2018) [260]	Similarity-based classifier.
	MADA (2018) [261]	Multi-domains.
	DIFA (2018) [262]	Extended ADDA [257] uses a pair of feature extractors.
	CyCADA (2018) [263]	Semantic consistency at both the pixel-level and feature-level.
	SymNet (2019) [264]	Category-level and domain-level confusion losses.
	M-ADDA (2020) [265]	Triplet loss function and ADDA [257].
	IIMT (2020) [266]	Mixup formulation and a feature-level consistency regularizer.
	MA-UDASD (2020) [267]	Source-free unsupervised domain adaptation.
	DM-ADA (2020) [268]	Domain mixup is jointly conducted on pixel and feature level.

adaptation process, they always perform not very well on real images such as the *OFFICE* dataset. GenToAdapt-GAN [259] is proposed in cases where data generation is hard, even though the generator network they use performs a mere style transfer, yet this is sufficient for providing good gradient information for successfully aligning the domains. Unlike Co-GAN, there is just one generator and one discriminator. In addition, there are two classifiers and one encoder to embed the instances into vectors.

Co-GAN and GenToAdapt adopt different strategies to train a classifier for an unlabeled domain. The biggest difference between Co-GAN and GenToAdapt-GAN is whether the feature extractor is the same. The feature extractor of Co-GAN is the GAN itself, but the feature extractor of GenToAdapt-GAN is a specialized encoder. In Co-GAN, GAN must do the jobs of adversarial process and encoding at the same time, but in GenToAdapt-GAN, these two jobs are separated, which means that GenToAdapt-GAN will be stabler and perform better when the data are complex. Most of the methods proposed in recent years are based on these two ways. Liu et al. [256] adopted different GAN for different domains and weight sharing. The main change is that the generator is replaced by VAE. Adversarial discriminative domain adaptation (ADDA) [257] adopted the discriminative model as the feature extractor is based on Co-GAN. ADDA can be viewed as a generalization of Co-GAN framework. Volpi et al. [262] extended ADDA using a pair of feature extractors. Laradji and Babanezhad [265] use a metric learning approach to train the source model on the source dataset by optimizing the triplet loss function as an optimized method and then using ADDA to complete its transferring process. SymNet [264] proposed a two-level domain confusion scheme that includes category- and domain-level confusion losses. With the same feature extractor of the source and target domains, multiadversarial domain adaptation (MADA) [261] sets the generator as its feature extractor expanding the UDA problem to multidomains. Similarity-based domain adaption network (SimNet) [260] uses discriminator as a feature extractor and a similarity-based classifier, which compares the embedding of an unlabeled image with a set of labeled prototypes to classify an image. Yan et al. [266] use mixup formulation and a feature-level consistency regularizer to address the generalization performance for target data. Xu et al. [268] use domain mixup on both pixel and feature levels to improve the robustness of models.

There is also a very straightforward way to transfer the knowledge between domains: Generate new instances for the target domain. If we transfer the instance from the source domain into a new instance that follows a joint distribution of both domain and is labeled the same as its mother source instance, then we get a batch of “labeled fake instances in target domain.” Training a classifier with these fake instances should be applicative to the real target data. In this way, we can easily use all the unsupervised adversarial domain adaptation methods in UDA as an effective data augmentation method. This accessible method also performs well in the DC problem and is called pixel-level transfer learning.

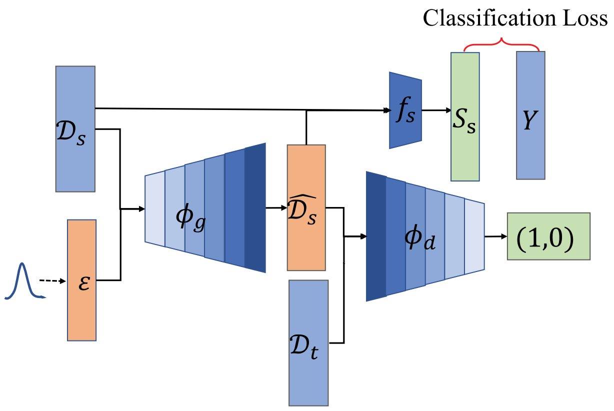


Fig. 7. Overview of the model architecture. The generator ϕ_g generates an image conditioned on a synthetic image, which is fed into the discriminator as fake data and a noise vector ε. The discriminator ϕ_d discriminates between real and fake images. $\mathcal{D}_s$ is the source domain. $\mathcal{D}_t$ is the target domain. $\widehat{\mathcal{D}_s}$ is the fake image and f_s is trained with generated data and source data. Y means the real labels and S_s denotes the predicted results.

Unsupervised pixel-level domain adaptation with GANs (Pixel-GAN) [258] aims at changing the images from the source domain to appear as if they were sampled from the target domain while maintaining their original content (label). The authors proposed a novel GAN-based architecture that can learn such a transformation in an unsupervised manner. The training process of Pixel-GAN is shown in Fig. 7. It uses a generator ϕ_g to propose a fake image with the input composed of a labeled source image and a noise vector. The fake images will be discriminated against with target data by a discriminator ϕ_d. At the same time, fake images $\widehat{\mathcal{D}_s}$ and source images are put into a classifier f_s; when the model is convergent, the classifier can be used on the target domain.

On the whole, Pixel-GAN is a very explicit model, but this net relies on the quality of the generated images too much. Although the classifier can guarantee the invariant information of classes, it is also hard to perform on complex images. Pixel-level transferring and feature-level transferring are not going against each other, as pixel-level can transfer visual features and feature-level transferring can transfer the nature information of the instances. Cycle-consistent adversarial domain adaptation (CyCADA) [263] adapts representations at both pixel level and feature level while enforcing semantic consistency. The authors enforce both structural and semantic consistency during adaptation using a cycle-consistency loss and semantics losses based on a particular visual recognition task. The semantics losses both guide the overall representation to be discriminative and enforce semantic consistency before and after mapping between domains. Except for GAN, adopting data augmentation to transfer learning can also be used in traditional ways. Sun et al. [269] provide the efficiency to make data augmentation in the target domain even if it is unlabeled. It adds self-supervised tasks to target data and shows good performance. More important is that this skill can be combined with other domain adaptation methods such as CyCADA and DAN.

VII. Future Directions of Deep Clustering

Based on the aforementioned literature review and analysis, DC has been applied to several domains, and we attach importance to several aspects worth studying further.

1) *Theoretical Exploration:* Although remarkable clustering performance has been achieved by designing even more sophisticated DC pipelines for specific problem-solving needs, there is still no reliable theoretical analysis on how to qualitatively analyze the influence of feature extraction and clustering loss on final clustering. Thus, exploring the theoretical basis of DC optimization is of great significance for guiding further research in this field.
2) *Massive Complex Data Processing:* Due to the complexity brought by massive data, most of the existing DC models are designed for specific datasets. Complex data from different sources and forms bring more uncertainties and challenges to clustering. At present, deep learning and graph learning are needed to solve complex data processing problems.
3) *Model Efficiency:* Deep clustering algorithm requires a large number of samples for training. Therefore, in small sample datasets, DC is prone to overfitting, which leads to the decrease of clustering effect and the reduction of the generalization performance of the model. On the other hand, the DC algorithm with large-scale data has high computational complexity, so the model structure optimization and model compression technology can be adopted to reduce the computational load of the model and improve the efficiency in practical application conditions.
4) *Fusion of Multiview Data:* In practical application scenarios, clustering is often not just with single image information but also available text and voice information. However, most of the current DC algorithms can only use one kind of information and cannot make good use of the existing information. The subsequent research can consider to fully integrate the information of two or more views and make full use of the consistency and complementarity of data of different views to improve the clustering effect. Furthermore, how to combine features of different views while filtering noise to ensure better view quality needs to be solved.
5) *Deep Clustering Based on Graph Learning:* In reality, a large number of datasets are stored in the form of graph structures. Graph structure can represent the structural association information between sample points. How to effectively use the structural information is particularly important to improve the clustering performance. Whether it is a single-view DC or a relatively wide application of multiview DC, existing clustering methods based on graph learning still have some problems, such as the graph structure information that is not fully utilized, and the differences and importance of different views are not fully considered. Therefore, the effective analysis of complex graph structure information, especially the rational use of graph structure information to complete the clustering task, needs further exploration.

VIII. Summary of Deep Clustering Methods

In this paper, we introduce recent advances in the field of deep clustering. This is mainly kind of data structures: single-view, semi-supervised, multi-view, and transfer learning. Single-view methods are the most important part of our survey, which inherits the problem settings of traditional clustering methods. We systematically distinguish the clustering methods with data source, and further introduce them in terms of the network they are based on. Among these networks, *DAE-based* methods and *DNN-based* methods are proposed earlier but may be limited with their poor performance on real datasets. Compared to *DAE-based* and *DNN-based* methods, *VAE-based* and *GAN-based* methods attract attention in recent years for their strong feature extraction and sample generation power. Graph neural network is one of the most popular networks recently, especially in community discovery problems. So we also summarize the *GNN-based* clustering methods. With the development of the Internet, the data for clustering have different application scenarios, so we summarize some clustering methods which have different problem settings. Semi-supervised clustering methods cluster the data with constraints that can be developed from single-view clustering methods by adding a constraints loss. Multiview clustering methods use the information of different views as a supplement. It has been used widely in both traditional neural networks and graph neural networks. Transfer learning can transfer the knowledge of a labeled domain to an unlabeled domain. We introduce clustering methods based on transfer learning with two types of networks: DNN and GAN. *DNN-based* methods focus on the measurement strategy of two domains, while *GAN-based* methods use discriminators to fit the measurement strategy. The complexity of most deep clustering methods can scale linearly with the data size n, making them suitable for large-scale real-life applications such as social networks, bioinformatics, and electronic commerce.

References

[1] Z. Wang et al., "Masked face recognition dataset and application," 2020, *arXiv:2003.09093*.

[2] J. Guo, X. Zhu, C. Zhao, D. Cao, Z. Lei, and S. Z. Li, "Learning meta face recognition in unseen domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6163–6172.

[3] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: A review," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4335–4385, Aug. 2020.

[4] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522–51532, 2019.

[5] J. Zhou, P. Li, Y. Zhou, B. Wang, J. Zang, and L. Meng, "Toward new-generation intelligent manufacturing," *Engineering*, vol. 4, no. 1, pp. 11–20, 2018.

[6] J. Zhou, Y. Zhou, B. Wang, and J. Zang, "Human–cyber–physical systems (HCPSs) in the context of new-generation intelligent manufacturing," *Engineering*, vol. 5, no. 4, pp. 624–636, 2019.

[7] J. MacQueen et al., "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Oakland, CA, USA, vol. 1, 1967, pp. 281–297.

[8] Y. Ren, U. Kamath, C. Domeniconi, and Z. Xu, "Parallel boosted clustering," *Neurocomputing*, vol. 351, pp. 87–100, Jul. 2019.

[9] M. Ester, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96, 1996, pp. 226–231.

[10] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[11] Y. Ren, U. Kamath, C. Domeniconi, and G. Zhang, "Boosted mean shift clustering," in *Proc. ECML-PKDD*, 2014, pp. 646–661.

[12] Y. Ren, C. Domeniconi, G. Zhang, and G. Yu, "A weighted adaptive mean shift clustering algorithm," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2014, pp. 794–802.

[13] Y. Ren, X. Hu, K. Shi, G. Yu, D. Yao, and Z. Xu, "Semi-supervised DenPeak clustering with pairwise constraints," in *Proc. 15th Pacific Rim Int. Conf. Artif. Intell.*, 2018, pp. 837–850.

[14] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cham, Switzerland: Springer, 2006, pp. 430–439.

[15] 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.

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

[17] Y. Ren, C. Domeniconi, G. Zhang, and G. Yu, "Weighted-object ensemble clustering: Methods and analysis," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 661–689, May 2017.

[18] A. Kumar and H. Daumé, "A co-training approach for multi-view spectral clustering," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 393–400.

[19] A. Kumar, P. Rai, and H. Daumé, III, "Co-regularized multi-view spectral clustering," in *Proc. 25th Annu. Conf. Neural Inf. Process. Syst.*, Granada, Spain, Dec. 2011, pp. 1413–1421.

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

[21] Z. Huang, Y. Ren, X. Pu, and L. He, "Non-linear fusion for self-paced multi-view clustering," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 3211–3219.

[22] Z. Huang, Y. Ren, X. Pu, L. Pan, D. Yao, and G. Yu, "Dual self-paced multi-view clustering," *Neural Netw.*, vol. 140, pp. 184–192, Aug. 2021.

[23] S. Huang, Y. Ren, and Z. Xu, "Robust multi-view data clustering with multi-view capped-norm K-means," *Neurocomputing*, vol. 311, pp. 197–208, Oct. 2018.

[24] S. Park, J. K. Park, S. J. Shin, and I. C. Moon, "Adversarial dropout for supervised and semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2017, pp. 3917–3924.

[25] W. Xia, Q. Gao, Q. Wang, X. Gao, C. Ding, and D. Tao, "Tensorized bipartite graph learning for multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5187–5202, Apr. 2023.

[26] W. Xia, T. Wang, Q. Gao, M. Yang, and X. Gao, "Graph embedding contrastive multi-modal representation learning for clustering," *IEEE Trans. Image Process.*, vol. 32, pp. 1170–1183, 2023.

[27] Q. Wang, Z. Tao, W. Xia, Q. Gao, X. Cao, and L. Jiao, "Adversarial multiview clustering networks with adaptive fusion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7635–7647, Oct. 2023.

[28] S. Shi, F. Nie, R. Wang, and X. Li, "Multi-view clustering via nonnegative and orthogonal graph reconstruction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 1, pp. 201–214, Jan. 2023.

[29] Z. Tao, J. Li, H. Fu, Y. Kong, and Y. Fu, "From ensemble clustering to subspace clustering: Cluster structure encoding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 5, pp. 2670–2681, May 2023.

[30] Z. Uykan, "Fusion of centroid-based clustering with graph clustering: An expectation-maximization-based hybrid clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 4068–4082, Aug. 2023.

[31] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, Aug. 1987.

[32] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Aug. 1998.

[33] M. D. Feit, J. A. Fleck, and A Steiger, "Solution of the Schrodinger equation by a spectral method," *J. Comput. Phys.*, vol. 47, no. 3, pp. 412–433, 1982.

[34] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[35] E. Aljalbout, V. Golkov, Y. Siddiqui, M. Strobel, and D. Cremers, "Clustering with deep learning: Taxonomy and new methods," 2018, *arXiv:1801.07648*.

[36] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: From the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39501–39514, 2018.

[37] G. C. Nutakki, B. Abdollahi, W. Sun, and O. Nasraoui, "An introduction to deep clustering," in *Clustering Methods for Big Data Analytics*. Cham, Switzerland: Springer, 2019, pp. 73–89.

[38] S. Zhou et al., "A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions," 2022, *arXiv:2206.07579*.

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

[40] C. Song, F. Liu, Y. Huang, L. Wang, and T. Tan, "Auto-encoder based data clustering," in *Proc. CIARP*, 2013, pp. 117–124.

[41] P. Huang, Y. Huang, W. Wang, and L. Wang, "Deep embedding network for clustering," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 1532–1537.

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

[43] J. Xie, R. B. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, Jun. 2016, pp. 478–487.

[44] 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.

[45] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 30, 2017, pp. 24–33.

[46] K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5736–5745.

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

[48] D. Chen, J. Lv, and Y. Zhang, "Unsupervised multi-manifold clustering by learning deep representation," in *Proc. AAAI*, 2017, pp. 1–7.

[49] X. Guo, E. Zhu, X. Liu, and J. Yin, "AAAI with data augmentation," in *Proc. ACML*, 2018, pp. 550–565.

[50] F. Li, H. Qiao, and B. Zhang, "Discriminatively boosted image clustering with fully convolutional auto-encoders," *Pattern Recognit.*, vol. 83, pp. 161–173, Nov. 2018.

[51] S. A. Shah and V. Koltun, "Deep continuous clustering," 2018, *arXiv:1803.01449*.

[52] S. A. Shah and V. Koltun, "Robust continuous clustering," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 37, pp. 9814–9819, Sep. 2017.

[53] E. Tzoreff, O. Kogan, and Y. Choukroun, "Deep discriminative latent space for clustering," 2018, *arXiv:1805.10795*.

[54] J. Chang, Y. Guo, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep discriminative clustering analysis," 2019, *arXiv:1905.01681*.

[55] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu, "Deep spectral clustering using dual autoencoder network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4066–4075.

[56] T. Zhang, P. Ji, M. Harandi, W. Huang, and H. Li, "Neural collaborative subspace clustering," 2019, *arXiv:1904.10596*.

[57] Y. Ren, N. Wang, M. Li, and Z. Xu, "Deep density-based image clustering," *Knowl.-Based Syst.*, vol. 197, Jun. 2020, Art. no. 105841.

[58] S. Affeldt, L. Labiod, and M. Nadif, "Spectral clustering via ensemble deep autoencoder learning (SC-EDAE)," *Pattern Recognit.*, vol. 108, Dec. 2020, Art. no. 107522.

[59] X. Guo et al., "Adaptive self-paced deep clustering with data augmentation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 9, pp. 1680–1693, Sep. 2020.

[60] X. Yang, C. Deng, K. Wei, J. Yan, and W. Liu, "Adversarial learning for robust deep clustering," in *Proc. NeurIPS*, vol. 33, 2020, pp. 9098–9108.

[61] R. McConville, R. Santos-Rodríguez, R. J. Piechocki, and I. Craddock, "N2D: (Not Too) deep clustering via clustering the local manifold of an autoencoded embedding," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 5145–5152.

[62] J. Wang and J. Jiang, "Unsupervised deep clustering via adaptive GMM modeling and optimization," *Neurocomputing*, vol. 433, pp. 199–211, Apr. 2021.

[63] 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.

[64] M. Kampffmeyer, S. Løkse, F. M. Bianchi, R. Jenssen, and L. Livi, "Deep kernelized autoencoders," in *Proc. SCIA*, 2017, pp. 419–430.

[65] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5879–5887.

[66] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 132–149.

[67] C. Hsu and C. Lin, "CNN-based joint clustering and representation learning with feature drift compensation for large-scale image data," *IEEE Trans. Multimedia*, vol. 20, no. 2, pp. 421–429, Feb. 2018.

[68] P. Haeusser, J. Plapp, V. Golkov, E. Aljalbout, and D. Cremers, "Associative deep clustering: Training a classification network with no labels," in *Proc. GCPR*, 2018, pp. 18–32.

[69] T. V. M. Souza and C. Zanchettin, "Improving deep image clustering with spatial transformer layers," 2019, *arXiv:1902.05401*.

[70] O. Nina, J. Moody, and C. Milligan, "A decoder-free approach for unsupervised clustering and manifold learning with random triplet mining," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3987–3994.

[71] X. Ji, A. Vedaldi, and J. Henriques, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9865–9874.

[72] J. Wu et al., "Deep comprehensive correlation mining for image clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8150–8159.

[73] G. Shiran and D. Weinshall, "Multi-modal deep clustering: Unsupervised partitioning of images," 2019, *arXiv:1912.02678*.

[74] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, "SCAN: Learning to classify images without labels," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020. pp. 268–285.

[75] H. Zhong, C. Chen, Z. Jin, and X.-S. Hua, "Deep robust clustering by contrastive learning," 2020, *arXiv:2008.03030*.

[76] J. Huang, S. Gong, and X. Zhu, "Deep semantic clustering by partition confidence maximisation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*, Sep. 2020, pp. 8849–8858.

[77] 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*.

[78] N. Dilokthanakul et al., "Deep unsupervised clustering with Gaussian mixture variational autoencoders," 2016, *arXiv:1611.02648*.

[79] J. A. Figueroa and A. R. Rivera, "Is simple better: Revisiting simple generative models for unsupervised clustering," in *Proc. 2nd Workshop Bayesian Deep Learn. (NeurIPS)*, 2017, pp. 1–6.

[80] X. Li, Z. Chen, L. K. M. Poon, and N. L. Zhang, "Learning latent superstructures in variational autoencoders for deep multidimensional clustering," 2018, *arXiv:1803.05206*.

[81] M. Willetts, S. Roberts, and C. Holmes, "Disentangling to cluster: Gaussian mixture variational ladder autoencoders," 2019, *arXiv:1909.11501*.

[82] V. Prasad, D. Das, and B. Bhowmick, "Variational clustering: Leveraging variational autoencoders for image clustering," 2020, *arXiv:2005.04613*.

[83] L. Cao, S. Asadi, W. Zhu, C. Schmidli, and M. Sjöberg, "Simple, scalable, and stable variational deep clustering," 2020, *arXiv:2005.08047*.

[84] L. Yang, W. Fan, and N. Bouguila, "Deep clustering analysis via dual variational autoencoder with spherical latent embeddings," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6303–6312, Sep. 2023.

[85] H. Ma, "Achieving deep clustering through the use of variational autoencoders and similarity-based loss," *Math. Biosci. Eng.*, vol. 19, no. 10, pp. 10344–10360, 2022.

[86] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," 2015, *arXiv:1511.06390*.

[87] W. Harchaoui, P.-A. Mattei, and C. Bouveyron, "Deep adversarial Gaussian mixture auto-encoder for clustering," in *Proc. ICLR*, 2017, pp. 1–5.

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

[89] 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.

[90] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, "ClusterGAN: Latent space clustering in generative adversarial networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4610–4617.

[91] N. Mrabah, M. Bouguessa, and R. Ksantini, "Adversarial deep embedded clustering: On a better trade-off between feature randomness and feature drift," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1603–1617, Apr. 2022.

[92] X. Yang, J. Yan, Y. Cheng, and Y. Zhang, "Learning deep generative clustering via mutual information maximization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6263–6275, Sep. 2023.

[93] X. Zhang, H. Liu, Q. Li, and X.-M. Wu, "Attributed graph clustering via adaptive graph convolution," 2019, *arXiv:1906.01210*.

[94] Z. Tao, H. Liu, J. Li, Z. Wang, and Y. Fu, "Adversarial graph embedding for ensemble clustering," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3562–3568.

[95] D. Zhu, S. Chen, X. Ma, and R. Du, "Adaptive graph convolution using heat kernel for attributed graph clustering," *Appl. Sci.*, vol. 10, no. 4, p. 1473, Feb. 2020.

[96] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, "Structural deep clustering network," in *Proc. Web Conf.*, Apr. 2020, pp. 1400–1410.

[97] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[98] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.

[99] L. Van Der Maaten, "Learning a parametric embedding by preserving local structure," *Artif. Intell. Statist.*, vol. 5, pp. 384–391, Apr. 2009.

[100] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1189–1197.

[101] R. Souvenir and R. Pless, "Manifold clustering," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Sep. 2005, pp. 648–653.

[102] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," in *Proc. NeurIPS*, 2011, pp. 55–63.

[103] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[104] A. Dundar, J. Jin, and E. Culurciello, "Convolutional clustering for unsupervised learning," 2015, *arXiv:1511.06241*.

[105] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, Sep. 1967.

[106] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst.*, vol. 28, Dec. 2015, pp. 2017–2025.

[107] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[108] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[109] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[110] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Aug. 2009, pp. 248–255.

[111] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *Proc. NeurIPS*, vol. 17, 2004, pp. 1537–1544.

[112] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[113] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama, "Learning discrete representations via information maximizing self-augmented training," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1558–1567.

[114] R. D. Hjelm et al., "Learning deep representations by mutual information estimation and maximization," 2018, *arXiv:1808.06670*.

[115] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 539–546.

[116] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[117] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1422–1430.

[118] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2536–2544.

[119] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 649–666.

[120] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 69–84.

[121] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," 2018, *arXiv:1803.07728*.

[122] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[123] I. J. Goodfellow et al., "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680.

[124] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," 2016, *arXiv:1601.06759*.

[125] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Barcelona, Spain, Dec. 2016, pp. 2172–2180.

[126] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug & play generative networks: Conditional iterative generation of images in latent space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3510–3520.

[127] M. E. Abbasnejad, A. Dick, and A. van den Hengel, "Infinite variational autoencoder for semi-supervised learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 781–790.

[128] D. P. Kingma and S. Mohamed, "Semi-supervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 3581–3589.

[129] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther, "Auxiliary deep generative models," 2016, *arXiv:1602.05473*.

[130] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst* Red Hook, NY, USA: Curran Associates 2016, pp. 2234–2242.

[131] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, *arXiv:1511.05644*.

[132] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 658–666.

[133] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[134] G. Chen, "Deep learning with nonparametric clustering," 2015, *arXiv:1501.03084*.

[135] M. D. Hoffman and M. J. Johnson, "ELBO surgery: Yet another way to carve up the variational evidence lower bound," in *Proc. NeurIPS*, 2016, pp. 1–4.

[136] G. J. McLachlan, S. X. Lee, and S. I. Rathnayake, "Finite mixture models," *Annu. Rev. Statist. Appl.*, vol. 6, pp. 355–378, Jan. 2000.

[137] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. thesis, Dept. Gatsby Comput. Neurosci. Unit, UCL Univ. College London, London, U.K., 2003.

[138] N. L. Zhang, "Hierarchical latent class models for cluster analysis," *J. Mach. Learn. Res.*, vol. 5, no. 6, pp. 697–723, 2004.

[139] Daphne Koller and Nir Friedman," *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[140] L. Yang, N.-M. Cheung, J. Li, and J. Fang, "Deep clustering by Gaussian mixture variational autoencoders with graph embedding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6440–6449.

[141] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," 2016, *arXiv:1611.01144*.

[142] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," 2016, *arXiv:1611.00712*.

[143] S. Zhao, J. Song, and S. Ermon, "Learning hierarchical features from generative models," 2017, *arXiv:1702.08396*.

[144] A. Krause, P. Perona, and R. Gomes, "Discriminative clustering by regularized information maximization," in *Proc. NIPS*, vol. 23, 2010, pp. 775–783.

[145] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. NeurIPS*, 2005, pp. 529–536.

[146] 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, Mar. 2010.

[147] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow, "Understanding and improving interpolation in autoencoders via an adversarial regularizer," 2018, *arXiv:1807.07543*.

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

[149] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 1735–1742.

[150] U. Shaham and R. R. Lederman, "Learning by coincidence: Siamese networks and common variable learning," *Pattern Recognit.*, vol. 74, pp. 52–63, Feb. 2018.

[151] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2008.

[152] D. Duvenaud et al., "Convolutional networks on graphs for learning molecular fingerprints," 2015, *arXiv:1509.09292*.

[153] M. A. Khamsi and W. A. Kirk, *An Introduction to Metric Spaces and Fixed Point Theory*, vol. 53. Hoboken, NJ, USA: Wiley, 2011.

[154] J. Zhou et al., "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020.

[155] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[156] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning deep representations for graph clustering," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1293–1299.

[157] M. Shao, S. Li, Z. Ding, and Y. Fu, "Deep linear coding for fast graph clustering," in *Proc. IJCAI*, 2015, pp. 3798–3804.

[158] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 833–852, May 2018.

[159] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Network representation learning: A survey," *IEEE Trans. Big Data*, vol. 6, no. 1, pp. 3–28, Mar. 2018.

[160] H. Cai, V. W. Zheng, and K. C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sep. 2018.

[161] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Mar. 2020.

[162] S. Yan et al., "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2006.

[163] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 835–850, Jun. 2005.

[164] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang, "Attributed graph clustering: A deep attentional embedding approach," 2019, *arXiv:1906.06532*.

[165] 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.

[166] J. Enguehard, P. O'Halloran, and A. Gholipour, "Semi-supervised learning with deep embedded clustering for image classification and segmentation," *IEEE Access*, vol. 7, pp. 11093–11104, 2019.

[167] H. Zhang, S. Basu, and I. Davidson, "A framework for deep constrained clustering-algorithms and advances," in *Proc. ECML-PKDD*, 2019, pp. 57–72.

[168] A. Shukla, G. S. Cheema, and S. Anand, "Semi-supervised clustering with neural networks," in *Proc. IEEE 6th Int. Conf. Multimedia Big Data (BigMM)*, Sep. 2020, pp. 152–161.

[169] A. A. Baffour, Z. Qin, J. Geng, Y. Ding, F. Deng, and Z. Qin, "Generic network for domain adaptation based on self-supervised learning and deep clustering," *Neurocomputing*, vol. 476, pp. 126–136, Mar. 2022.

[170] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proc. Int. Workshop Artif. Intell. Statist.*, 2005, pp. 57–64.

[171] K. Huang, Z. Xu, I. King, and M. R. Lyu, "Semi-supervised learning from general unlabeled data," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 273–282.

[172] Z. Xu, I. King, M. R. Lyu, and R. Jin, "Discriminative semi-supervised feature selection via manifold regularization," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1033–1047, Jul. 2010.

[173] Y. Huang, D. Xu, and F. Nie, "Semi-supervised dimension reduction using trace ratio criterion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 519–526, Mar. 2012.

[174] E. Bair, "Semi-supervised clustering methods," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 5, no. 5, pp. 349–361, 2013.

[175] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: A brief survey," *Rev. Mach. Learn. Techn. Process. Multimedia Content*, vol. 1, pp. 9–16, Jul. 2004.

[176] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proc. ICML*, 2009, pp. 129–136.

[177] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2750–2756.

[178] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, May 2015, pp. 586–594.

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

[180] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4279–4287.

[181] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, Jul. 2018.

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

[183] M. Brbić and I. Kopriva, "Multi-view low-rank sparse subspace clustering," *Pattern Recognit.*, vol. 73, pp. 247–258, Jan. 2018.

[184] Y. Ren, S. Huang, P. Zhao, M. Han, and Z. Xu, "Self-paced and auto-weighted multi-view clustering," *Neurocomputing*, vol. 383, pp. 248–256, Mar. 2020.

[185] C. Xu, D. Tao, and C. Xu, "Multi-view self-paced learning for clustering," in *Proc. 24th Int. Conf. Artif. Intell.*, Jul. 2015, pp. 3974–3980.

[186] J. Xu, Y. Ren, G. Li, L. Pan, C. Zhu, and Z. Xu, "Deep embedded multi-view clustering with collaborative training," *Inf. Sci.*, vol. 573, pp. 279–290, Sep. 2021.

[187] S. Fan, X. Wang, C. Shi, E. Lu, K. Lin, and B. Wang, "One2Multi graph autoencoder for multi-view graph clustering," in *Proc. Web Conf.*, Apr. 2020, pp. 3070–3076.

[188] X. Tang, X. Tang, W. Wang, L. Fang, and X. Wei, "Deep multi-view sparse subspace clustering," in *Proc. 8th Int. Conf. Netw., Commun. Comput.*, Dec. 2018, pp. 115–119.

[189] R. Li, C. Zhang, H. Fu, X. Peng, T. Zhou, and Q. Hu, "Reciprocal multi-layer subspace learning for multi-view clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 8172–8180.

[190] P. Zhu, X. Yao, Y. Wang, B. Hui, D. Du, and Q. Hu, "Multi-view deep subspace clustering networks," 2019, *arXiv:1908.01978*.

[191] Z. Li, Q. Wang, Z. Tao, Q. Gao, and Z. Yang, "Deep adversarial multi-view clustering network," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2952–2958.

[192] M. Yin, W. Huang, and J. Gao, "Shared generative latent representation learning for multi-view clustering," in *Proc. AAAI Conf. Artif. Intell.*, New York, NY, USA, 2020, pp. 6688–6695.

[193] J. Xu et al., "Multi-VAE: Learning disentangled view-common and view-peculiar visual representations for multi-view clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9234–9243.

[194] F. Lin, B. Bai, K. Bai, Y. Ren, P. Zhao, and Z. Xu, "Contrastive multi-view hyperbolic hierarchical clustering," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 3250–3256.

[195] J. Xu, H. Tang, Y. Ren, L. Peng, X. Zhu, and L. He, "Multi-level feature learning for contrastive multi-view clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 16051–16060.

[196] J. Xu et al., "Deep incomplete multi-view clustering via mining cluster complementarity," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 8, pp. 8761–8769.

[197] M. R. Khan and J. E. Blumenstock, "Multi-GCN: Graph convolutional networks for multi-view networks, with applications to global poverty," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, Jul. 2019, pp. 606–613.

[198] J. Cheng, Q. Wang, Z. Tao, D.-Y. Xie, and Q. Gao, "Multi-view attribute graph convolution networks for clustering," in *Proc. IJCAI*, 2020, pp. 2973–2979.

[199] Y. Wang, D. Chang, Z. Fu, and Y. Zhao, "Consistent multiple graph embedding for multi-view clustering," 2021, *arXiv:2105.04880*.

[200] Z. Huang, Y. Ren, X. Pu, and L. He, "Deep embedded multi-view clustering via jointly learning latent representations and graphs," 2022, *arXiv:2205.03803*.

[201] X.-L. Li, M.-S. Chen, C.-D. Wang, and J.-H. Lai, "Refining graph structure for incomplete multi-view clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 2300–2313, Feb. 2024.

[202] S. Liu, X. Liu, S. Wang, X. Niu, and E. Zhu, "Fast incomplete multi-view clustering with view-independent anchors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 7740–7751, Jun. 2024.

[203] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, "COMPLETER: Incomplete multi-view clustering via contrastive prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11169–11178.

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

[205] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Neural Inf. Process. Syst. Nat. Synthetic (NIPS)*, vol. 2, 2002, pp. 849–856.

[206] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[207] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2790–2797.

[208] F. Nie, H. Wang, H. Huang, and C. Ding, "Unsupervised and semi-supervised learning via l1-norm graph," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2268–2273.

[209] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2012, pp. 347–360.

[210] 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.

[211] 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.

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

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

[214] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1582–1590.

[215] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," 2013, *arXiv:1304.5634*.

[216] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, 1st ed., New York, NY, USA: Wiley, 1958.

[217] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1247–1255.

[218] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 1083–1092.

[219] M. Qu, J. Tang, J. Shang, X. Ren, M. Zhang, and J. Han, "An attention-based collaboration framework for multi-view network representation learning," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 1767–1776.

[220] S. E. Schaeffer, "Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, 2007.

[221] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," in *Proc. NeurIPS*, 2019, pp. 11983–11993.

[222] H. Perkins and Y. Yang, "Dialog intent induction with deep multi-view clustering," 2019, *arXiv:1908.11487*.

[223] M. Abavisani and V. M. Patel, "Deep multimodal subspace clustering networks," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1601–1614, Dec. 2018.

[224] D. Hu, F. Nie, and X. Li, "Deep multimodal clustering for unsupervised audiovisual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2019, pp. 9248–9257.

[225] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[226] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.

[227] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, vol. 2, Dec. 2014, pp. 3320–3328.

[228] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2015, pp. 1–9.

[229] M. Ghifary, W. B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in *Proc. Pacific Rim Int. Conf. Artif. Intell.* Cham, Switzerland: Springer, 2014, pp. 898–904.

[230] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A iciarst," *JMLR*, vol. 13, no. 1, pp. 723–773, 2012.

[231] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 97–105.

[232] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

[233] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Sep. 2017, pp. 2272–2281.

[234] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 136–144.

[235] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5018–5027.

[236] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced Wasserstein discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10285–10295.

[237] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," in *Domain Adaptation in Computer Vision Applications*. Cham, Switzerland: Springer, 2017, pp. 153–171.

[238] C. Chen et al., "HoMM: Higher-order moment matching for unsupervised domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 3422–3429.

[239] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Sep. 2019, pp. 4893–4902.

[240] L. Hu, M. Kan, S. Shan, and X. Chen, "Unsupervised domain adaptation with hierarchical gradient synchronization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4043–4052.

[241] M. Li, Y.-M. Zhai, Y.-W. Luo, P.-F. Ge, and C.-X. Ren, "Enhanced transport distance for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13936–13944.

[242] R. Xu, P. Liu, L. Wang, C. Chen, and J. Wang, "Reliable weighted optimal transport for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4394–4403.

[243] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 19, 2006, pp. 137–144.

[244] H. Tang, K. Chen, and K. Jia, "Unsupervised domain adaptation via structurally regularized deep clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8725–8735.

[245] Q. Wang and T. P. Breckon, "Unsupervised domain adaptation via structured prediction based selective pseudo-labeling," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 6243–6250.

[246] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16. Cambridge, MA, USA: MIT Press, 2003, pp. 153–160.

[247] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *IEEE Trans. Image Process.*, vol. 30, pp. 8008–8018, 2021.

[248] V. Prabhu, S. Khare, D. Kartik, and J. Hoffman, "SENTRY: Selective entropy optimization via committee consistency for unsupervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8538–8547.

[249] X. Jiang, Q. Lao, S. Matwin, and M. Havaei, "Implicit class-conditioned domain alignment for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 4816–4827.

[250] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui, "Casting a BAIT for offline and online source-free domain adaptation," 2020, *arXiv:2010.12427*.

[251] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6028–6039.

[252] J. Liang, D. Hu, Y. Wang, R. He, and J. Feng, "Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8602–8617, Nov. 2022.

[253] S. Tang et al., "Nearest neighborhood-based deep clustering for source data-absent unsupervised domain adaptation," 2021, *arXiv:2107.12585*.

[254] M. Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. Adv. Neural Inf. Proc. Syst. (NIPS)*, 2016, pp. 469–477.

[255] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, May 2016.

[256] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 30, 2017, pp. 700–708.

[257] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7167–7176.

[258] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3722–3731.

[259] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8503–8512.

[260] P. O. Pinheiro, "Unsupervised domain adaptation with similarity learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8004–8013.

[261] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," 2018, *arXiv:1809.02176*.

[262] R. Volpi, P. Morerio, S. Savarese, and V. Murino, "Adversarial feature augmentation for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5495–5504.

[263] J. Hoffman et al., "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1989–1998.

[264] Y. Zhang, H. Tang, K. Jia, and M. Tan, "Domain-symmetric networks for adversarial domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5031–5040.

[265] I. H. Laradji and R. Babanezhad, "M-ADDA: Unsupervised domain adaptation with deep metric learning," in *Domain Adaptation for Visual Understanding*. Cham, Switzerland: Springer, 2020, pp. 17–31.

[266] S. Yan, H. Song, N. Li, L. Zou, and L. Ren, "Improve unsupervised domain adaptation with mixup training," 2020, *arXiv:2001.00677*.

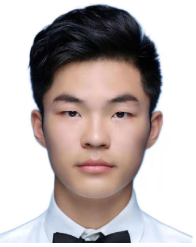
[267] R. Li, Q. Jiao, W. Cao, H. Wong, and S. Wu, "Model adaptation: Unsupervised domain adaptation without source data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9641–9650.

[268] M. Xu et al., "Adversarial domain adaptation with domain mixup," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 6502–6509.

[269] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros, "Unsupervised domain adaptation through self-supervision," 2019, *arXiv:1909.11825*.

Yazhou Ren (Member, IEEE) received the B.Sc. degree in information and computation science and the Ph.D. degree in computer science from South China University of Technology, Guangzhou, China, in 2009 and 2014, respectively.

He visited the Data Mining Laboratory, George Mason University, Fairfax, VA, USA, from 2012 to 2014. He is currently an Associate Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. He has published more than 100 peer-reviewed research articles. His current research interests include multiview clustering, unsupervised learning, and medical data analysis.



Jingyu Pu is currently pursuing the M.Sc. degree in computer science and technology with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China.

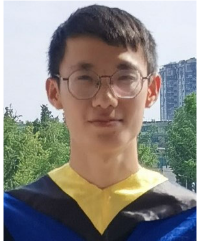
His research interests include deep learning and multiview clustering.

Zhimeng Yang received the M.Sc. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2022.

Her research interests include deep clustering and domain adaptation.

Jie Xu received the B.Eng. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2020, where he is currently pursuing the Ph.D. degree in computer science and technology.

His research interests include deep learning, multiview clustering, and incomplete multiview clustering.



Guofeng Li received the M.Sc. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2022.

His research interests include deep clustering and domain adaptation.

Xiaorong Pu received the Ph.D. degree in computer application from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2007.

She is currently a Professor with the School of Computer Science and Engineering, UESTC. Her current research interests include neural networks, computer vision, computer-aided diagnosis (CAD), and e-health.

Philip S. Yu (Fellow, IEEE) was at the IBM Watson Research Center, Yorktown Heights, NY, USA, where he built a world-renowned Data Mining and Database Department. He is currently a Distinguished Professor and the Wexler Chair in Information Technology at the Department of Computer Science, University of Illinois Chicago (UIC), Chicago, IL, USA. He has published more than 1600 referred conference papers and journal articles cited more than 194 000 times with an H-index of 197. He has applied for more than 300 patents.

Dr. Yu is a fellow of Association for Computing Machinery (ACM). He was the Editor-in-Chief of *ACM Transactions on Knowledge Discovery from Data* from 2011 to 2017 and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 2001 to 2004.

Lifang He (Member, IEEE) worked as a Post-Doctoral Researcher at the Department of Biostatistics and Epidemiology, University of Pennsylvania, Philadelphia, PA, USA. She is currently an Assistant Professor with the Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA. She has published more than 140 papers in refereed journals and conferences, such as Neural Information Processing Systems (NIPS), International Conference on Machine Learning (ICML), Association for Computing Machinery (ACM) SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), International World Wide Web Conference (WWW), International Joint Conference on Artificial Intelligence (IJCAI), AAAI Conference on Artificial Intelligence (AAAI), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), and American Medical Informatics Association (AMIA). Her current research interests include machine learning, data mining, and tensor analysis, with major applications in biomedical data and neuroscience.