Wasserstein Embedding Learning for Deep Clustering: A Generative Approach

Jinyu Cai, Yunhe Zhang, Shiping Wang, Jicong Fan, *Senior Member, IEEE*, Wenzhong Guo*

***Abstract*—Deep learning-based clustering methods, especially those incorporating deep generative models, have recently shown noticeable improvement on many multimedia benchmark datasets. However, existing generative models still suffer from unstable training, and the gradient vanishes, which results in the inability to learn desirable embedded features for clustering. In this paper, we aim to tackle this problem by exploring the capability of Wasserstein embedding in learning representative embedded features and introducing a new clustering module for jointly optimizing embedding learning and clustering. To this end, we propose Wasserstein embedding clustering (WEC), which integrates robust generative models with clustering. By directly minimizing the discrepancy between the prior and marginal distribution, we transform the optimization problem of Wasserstein distance from the original data space into embedding space, which differs from other generative approaches that optimize in the original data space. Consequently, it naturally allows us to construct a joint optimization framework with the designed clustering module in the embedding layer. Due to the substitutability of the penalty term in Wasserstein embedding, we further propose two types of deep clustering models by selecting different penalty terms. Comparative experiments conducted on nine publicly available multimedia datasets with several state-of-the-art methods demonstrate the effectiveness of our method.**

***Index Terms*—Unsupervised learning, clustering analysis, Wasserstein embedding, generative models, auto-encoder**

I. Introduction

Clustering [1]–[4] is an essential research problem in machine learning and data mining. With the development of deep learning technology, the combination of deep learning and clustering, also known as deep clustering, has become a promising research issue. Deep clustering takes advantage of deep learning to extract representative features for raw data to facilitate clustering. The classical deep clustering methods [5]–[9] are typically two-stage models, which first utilize various embedding learning methods to learn the representation of original data and then perform clustering.

However, the embedded features obtained in this way are not guaranteed to be suitable for clustering tasks, which leads to unsatisfactory clustering performance. To address this issue, recent studies [10]–[12] have focused on incorporating a clustering component in the network architecture. As a result, the network can be jointly optimized for embedding learning and clustering during the training process, resulting in the embedded representation that is beneficial for clustering tasks. Auto-encoder [13] is the most commonly used embedding learning method in deep clustering, and some studies [14], [15] have further improved clustering performance by exploiting its improved variants.

Generative models like variational auto-encoder (VAE) [16], [17] and generative adversarial networks (GAN) [18], [19] are more effective in representation learning and generating meaningful new samples. Various generative models have received extensive attention in recent years, especially with the proposal of Wasserstein GAN (WGAN) [20], [21] based on optimal transport (OT) theory [22]–[24]. Several works [25]–[28] have utilized generative models to improve clustering performance. We know that, in general, the data in the original space can be well represented in a low-dimensional manifold, and accordingly, the embedding space then supports the data in the original data space. Many distance measurements, such as f-divergences used in classical GAN [29] that reflect the density ratio among two distributions, often indicate a strong distance concept. However, they may fail to provide usable gradients for training, i.e., the gradient vanishes when there is hardly any non-negligible overlap between the distributions (commonly seen in GANs). In contrast, the Wasserstein distance induced from OT has been proven to offer a much weaker topology [20] compared to them, which could provide smoother gradients for training, and consequently, facilitate the embedding learning process in deep clustering.

In this paper, we propose a novel Wasserstein embedding clustering (WEC) model through introducing the Wasserstein embedding [30], [31] and a designed clustering module. Specifically, we transform the optimization problem of Wasserstein distance between data distribution and model distribution into an encoding-decoding-like process (see Theorem 1). Then, we propose to optimize the Wasserstein embedding and clustering simultaneously in the embedding space rather than in complex data space like GANs. Figure 1 summarizes the workflow of our method. The whole framework can be divided into two modules. The first one is the Wasserstein embedding module that consists of an encoder, a decoder, and a penalty term. The encoder aims to learn embedded features from the input data while the decoder conversely reconstructs them from the embedded features. The penalty term works in the expectation that the embedded distribution matches the prior distribution by penalizing the discrepancy between them. Second, a clustering module is developed in the em-

This work is in part supported by the National Natural Science Foundation of China (Grant No. U21A20472 and No. 62276065), and Shenzhen Research Institute of Big Data (Grant No.T00120210002). Corresponding author: Wenzhong Guo (e-mail:guowenzhong@fzu.edu.cn).

Jinyu Cai, Yunhe Zhang, Shiping Wang, Wenzhong Guo are with the College of Computer and Data Science, Fuzhou University, Fujian 350108, China.

Jicong Fan is with The Chinese University of Hong Kong, Shenzhen, and Shenzhen Research Institute of Big Data, 518172, China.

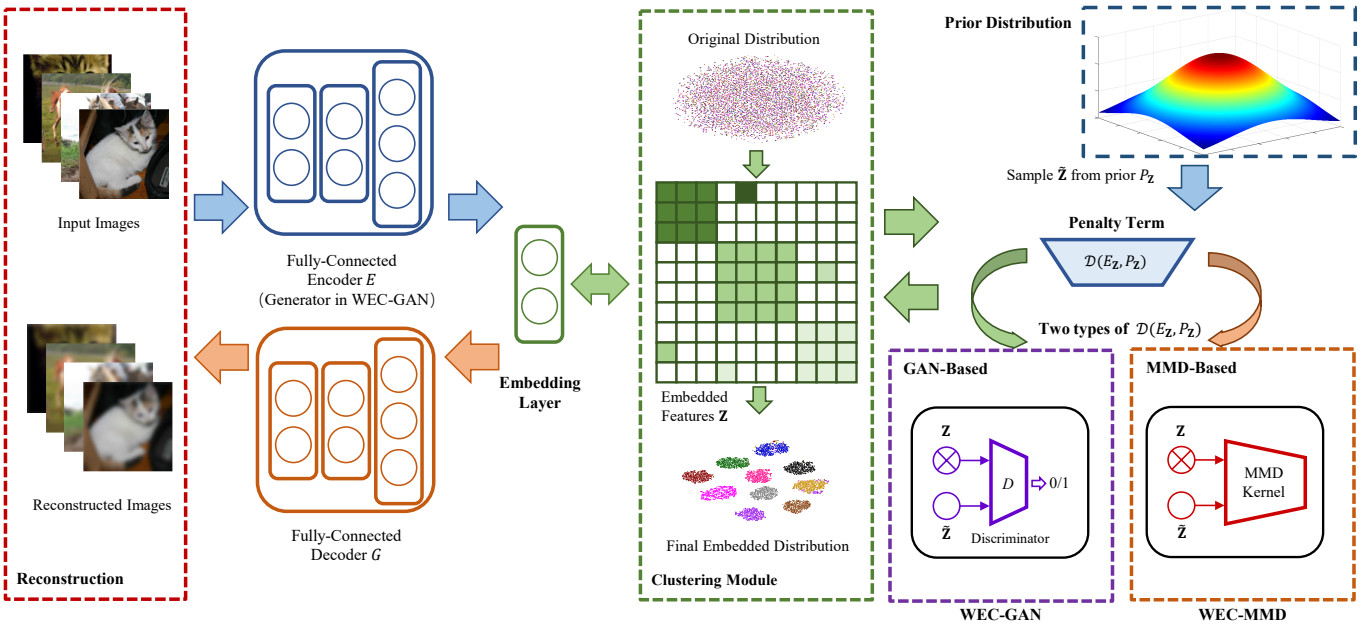


Fig. 1. An illustration of the proposed Wasserstein embedding clustering method, which can be partitioned into two components: the Wasserstein embedding learning module and the clustering module. Specifically, The embedding learning layer aims to obtain the embedded features for the clustering task, while the clustering module provides a clustering objective to guide the embedded features toward clustering, which is a joint optimization process. By introducing a substitutable penalty term $\mathcal{D}(E_{\mathbf{Z}}, P_{\mathbf{Z}})$, two generative deep clustering methods are proposed.

bedding layer to define an explicit clustering objective based on embedded features. This allows us to jointly optimize the Wasserstein embedding and clustering, leading to an end-to-end generative clustering model. Notably, the penalty term is substitutable, implying that we could construct different clustering models by choosing different divergences. Specifically, we choose two ways in practice. One is to employ Jensen-Shannon divergence (JS-divergence) with adversarial training, i.e., WEC-GAN, in which case the encoder is considered the generator, and a discriminator is inserted into the embedding layer to achieve the min-max optimization. The other way is to employ maximum mean discrepancy (MMD) as the penalty term, i.e., WEC-MMD, because of its advantage in matching high dimensional standard normal distribution. In this case, we only need to optimize a non-adversarial problem. Experimental results comparing with some popular clustering methods on nine databases demonstrate the superiority of our proposed method. The main contributions of our work are summed up as follows:

- We propose a novel Wasserstein embedding clustering model, which takes advantage of generative models in learning representative embedded features and facilitates clustering by jointly optimizing the Wasserstein embedding and clustering objective.
- We provide two realization approaches to the Wasserstein embedding clustering, one is based on Jensen-Shannon divergence and adversarial training, and the other is based on the maximum mean discrepancy.
- Extensive experimental results demonstrate that our method significantly outperforms the baselines and has state-of-the-art clustering performance.

The remainder of this paper is organized as follows. Some related works on deep clustering and the concept of Wasserstein distance are briefly reviewed in Section II. The architecture of the two modules of the proposed method and the joint optimization strategy are described in detail in Section III. Then, comprehensive experiments are presented in Section IV to demonstrate the effectiveness of our method. Finally, relevant conclusions are presented in Section V.

II. Related Works

In this section, we briefly present a review of some related works on deep clustering and generative model-based clustering, as well as the concept of Wasserstein distance.

A. Deep Clustering

Deep clustering, which takes advantage of deep learning in capturing representation to facilitate clustering, has been extensively studied in the past decade. It can be broadly divided into two categories, i.e., two-stage based and joint optimization-based approaches. Early works [32]–[36] have focused on the two-stage approach of applying various embedding learning methods to obtain low-dimensional representation and then perform clustering. For example, Patel et al. [37] applied sparse coding to learn a projection of data and discover the sparse coefficients in the latent space, then adopted spectral clustering to realize the cluster labels assignment. Since the approaches based on shallow linear models could fail when dealing with non-linear data structure, Peng et al. [38] improved the deep subspace clustering by introducing a sparsity prior in the embedding learning to capture more meaningful features. Nevertheless, these methods are limited by the difficulty of guaranteeing that the learned features are suitable for clustering during embedding learning.

To address the above-mentioned issue, some recent works have focused on jointly learning the clustering-oriented representation. Xie et al. [10] proposed deep embedded clustering (DEC) method to introduce a clustering objective into optimization, thus enabling the network to be optimized for the clustering task. Since the structure of DEC lacks a decoder in training, which may lead to distortion of feature space. Guo et al. [39] further improved DEC by considering the reconstruction loss of the auto-encoder in training to maintain the local structure of data. Moreover, Yang et al. [11] proposed the joint unsupervised learning (JULE) model, which integrated convolutional neural network with clustering to jointly optimize the embedding learning and clustering based on the idea that a good representation facilitates clustering tasks. Ghasedi et al. [14] proposed the deep embedded regularized clustering (DEPICT) method to simultaneously implement the cluster allocation and learn discriminative representation. Based on the invariance of sample assignments in clustering when different measures are applied, Peng et al. [40] found a new prior for sample-assignment invariance, and proposed an end-to-end deep clustering model (DCSAIP) with this prior through the minimization of the discrepancy among sample assignments of different measures.

B. Generative Model-Based Clustering Approaches

The adoption of embedding learning methods with better data revealing capability, especially like generative models [41]–[43], provides another way to improve clustering. VAE is a widely used generative model, and some studies also applied it to facilitate clustering. By introducing the probabilistic clustering issue to the VAE architecture, Jiang et al. [25] proposed the variational deep embedding approach (VaDE), which combines the VAE network and Gaussian mixture model for clustering. Yang et al. [44] applied Gaussian mixture VAE (GMVAE) [45] to improve the performance of game level clustering, which also allowed the model to generate corresponding game levels as required.

Additionally, GAN is another approach that has been often employed to promote clustering. For example, by introducing the adversarial learning strategy to guide the process of feature learning and clustering, Zhou et al. [46] proposed a novel generative model-based clustering method to learn features that are more beneficial for subspace clustering. Ghasedi et al. [47] proposed a deep clustering model based on GAN by employing an adversarial game between a generator, discriminator, and clusterer.

C. Wasserstein Distance

The optimal transport (OT) problem is based on measuring the distance between two probability distributions, i.e., the optimal strategy for transforming one distribution into the other through transportation. Given the input data distribution $P_{\mathbf{X}}$ and the model distribution $P_{\mathbf{Y}}$, the Kantorovich's formulation of OT cost is defined as follows:

$$W_c(P_{\mathbf{X}}, P_{\mathbf{Y}}) = \inf_{\theta \in \Theta(P_{\mathbf{X}}, P_{\mathbf{Y}})} \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\theta}[c(\mathbf{X},\mathbf{Y})], \tag{1}$$

where $\Theta(P_{\mathbf{X}}, P_{\mathbf{Y}})$ denotes the collection of all the joint distributions of $(\mathbf{X}, \mathbf{Y})$ under margins $P_{\mathbf{X}}$ and $P_{\mathbf{Y}}$ respectively, and $c(\mathbf{X}, \mathbf{Y})$ represents the cost function. Particularly, when a metric $m(\cdot,\cdot)$ satisfies the conditions that $c(\mathbf{X},\mathbf{Y}) = m(\mathbf{X},\mathbf{Y})^d$ and $d \geq 1$, then $W_p := \sqrt[p]{W_c}$, the p-th root of the OT cost W_c is called p-Wassersttein distance.

Consequently, the Wasserstein distance is derived from the OT problem as a way of measuring the discrepancy between two distributions. Compared with other distance measurements such as KL-divergence (in VAEs) and JS-divergence (in GANs), Wasserstein distance provides a more weakly convergent probability measure that benefits the stability of model training. It has been widely studied in machine learning over recent years. The generative models based on Wasserstein distance, especially WGAN and its variants [48], [49], have been a promising solution to improve clustering. Yang et al. [50] proposed to combine WGAN-GP [41] and VAE to construct a new clustering framework, and further improved the stability of the model in clustering when facing outliers. With the discussion of the relation of k-means and OT, Mi et al. [51] solved the OT problem via the variational theory and proposed to simultaneously optimize the Wasserstein distance between the centers and the target domain with clustering error. However, to the best of our knowledge, the combination of Wasserstein embedding and clustering in the embedding space is still an open issue in deep clustering.

III. Proposed Method

In this section, the proposed Wasserstein embedding clustering model is introduced, and Figure 1 summarizes the framework of our method. First, the Wasserstein embedding module is introduced to capture the low-dimensional representation of the original data. Then, the self-optimized clustering module is developed to simultaneously achieve the learning of embedded representation and the assignment of clustering labels in the embedding space. To facilitate the reading of the paper, some major notations and their explanations are summarized in Table I.

A. Wasserstein Embedding Module

The goal of embedding learning is to capture low-dimensional features for input data, and auto-encoder is a common embedding learning framework. It can be regarded as solving the following problem:

Problem 1: Given a input data $\mathbf{X} \in \mathbb{R}^{n\times d}$, find a mapping $E : \mathcal{X} \to \mathcal{Z}$ and an inverse mapping $G : \mathcal{Z} \to \mathcal{X}$ that minimizes the following objective:

$$c(\mathbf{X}, G(E(\mathbf{X}))), \tag{2}$$

where c denotes the cost function, E maps $\mathbf{X}$ to the embedded representation $\mathbf{Z}$ and then G attempts to reconstruct $\mathbf{X}$ from $\mathbf{Z}$. This problem can be easily optimized with deep neural networks, and adopting the commonly used MSE loss as the cost function may be sufficient for handling the reconstruction task. However, our aim is to cluster, which requires the data to be discriminative enough. Thus, we explore the capability

TABLE I
DESCRIPTION OF SOME MAJOR NOTATIONS.

Notation	Explanation
n	Sample size of the input data
d	Dimension of the input data
d'	Dimension of the embedded representation
β, γ	Hyper-parameters of our model
μ	Cluster centroid
$\mathcal{X}$	Original data space
$\mathcal{Z}$	Embedded space
$\mathbf{X}$	Input data matrix
$\mathbf{Y}$	Output reconstructed data matrix
$\mathbf{Z}$	Embedded representation
$\tilde{\mathbf{Z}}$	Samples from prior distribution
$P_{\mathbf{X}}$	Input data distribution
$P_{\mathbf{Y}}$	Model distribution
$E_{\mathbf{Z}}$	Embedded distribution
$P_{\mathbf{Z}}$	Prior distribution
S	soft label distribution in the clustering module
T	Target distribution in the clustering module
$\mathcal{D}$	Penalty term of Wasserstein embedding
E, G	Encoder and Decoder
V	Predicted cluster labels of the model

of the generative model in learning representation to capture more discriminative features for clustering tasks.

Now, we define a latent variable model $P_{\mathbf{Y}}$ as follows:

$$p_{\mathbf{y}}(\mathbf{x}) := \int_{\mathcal{Z}} p_{\mathbf{y}}(\mathbf{x}|\mathbf{z}) p_{\mathbf{z}}(\mathbf{z}) d\mathbf{z}, \tag{3}$$

where $P_{\mathbf{Y}}(\mathbf{X}|\mathbf{Z})$ denotes a non-random decoder distribution, i.e., generative model. And $P_{\mathbf{Y}}$ maps the latent representation $\mathbf{Z}$ to the original data $\mathbf{X} \in \mathcal{X}$ via a mapping $G\colon \mathcal{Z} \to \mathcal{X}$. The generative model aims to minimize the discrepancy among data distribution $P_{\mathbf{X}}$ and model distribution $P_{\mathbf{Y}}$, and typically, the f-divergences are the commonly used strategies. However, these solutions may suffer from gradient vanishing and unstable training, resulting in undesirable embedding learning. Therefore, we introduce the Wasserstein embedding learning to address this issue, as Wasserstein distance can provide more weakly convergent probability measure and smoother gradients, leading to more stable training and learning of more representative representation.

In the OT problem described in Eq. (1), the coupling $\theta(\mathbf{X}, \mathbf{Y}) = \theta(\mathbf{Y}|\mathbf{X}) P_{\mathbf{X}}(\mathbf{X})$ is considered, where $\theta(\mathbf{Y}|\mathbf{X})$ is regarded as the mapping from $\mathbf{X}$ to $\mathbf{Y}$. While in this case, we need to solve the optimization problem in the complex data space $\mathcal{X}$, and it is also hard for us to construct an end-to-end clustering framework in the embedding layer. To this end, we introduce the following Theorem 1 proved in [30] to reparametrize this mapping into an encoding-decoding like process, i.e., to transfer the optimization problem from the original data space $\mathcal{X}$ to the embedding space $\mathcal{Z}$ via the transport $\mathbf{X} \to \mathbf{Z} \to \mathbf{Y}$:

Theorem 1: For a generative model $P_{\mathbf{Y}}$ with deterministic $P_{\mathbf{Y}}(\mathbf{X}|\mathbf{Z})$ and arbitrary mapping function $G\colon \mathcal{Z} \to \mathcal{X}$, it holds:

$$\begin{aligned} W_c(P_{\mathbf{X}}, P_{\mathbf{Y}}) &= \inf_{\theta \in \Theta(P_{\mathbf{X}}, P_{\mathbf{Y}})} \mathbb{E}_{(\mathbf{X},\mathbf{Y}) \sim \theta}[c(\mathbf{X}, \mathbf{Y})] \\ &= \inf_{E\colon E_{\mathbf{Z}} = P_{\mathbf{Z}}} \mathbb{E}_{P_{\mathbf{X}}} \mathbb{E}_{E(\mathbf{Z}|\mathbf{X})}[c(\mathbf{X}, G(\mathbf{Z}))], \end{aligned} \tag{4}$$

where $E_{\mathbf{Z}}$ denotes the marginal of $\mathbf{Z}$ when $\mathbf{X} \sim P_{\mathbf{X}}$ and $\mathbf{Z} \sim E(\mathbf{Z}|\mathbf{X})$.

With Theorem 1, Eq. (1) can be explicitly connected to Problem 1. More specifically, given a deterministic mapping from the prior latent distribution $P_{\mathbf{Z}}$ to $P_{\mathbf{Y}}$, the Wasserstein distance between $P_{\mathbf{X}}$ and $P_{\mathbf{Y}}$ can be transformed from searching for the coupling θ of random variables from two distributions respectively to search for the conditional distribution $E(\mathbf{Z}|\mathbf{X})$, such that its $\mathbf{Z}$-margin is the same as $P_{\mathbf{Z}}$, i.e., $E_{\mathbf{Z}}(\mathbf{Z}) = \int E(\mathbf{Z}|\mathbf{X}) P_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} = P_{\mathbf{Z}}$. $E_{\mathbf{Z}}(\mathbf{Z})$ also denotes the aggregated posterior.

Subsequently, the objective turns out to be the optimization over a probabilistic encoder $E(\mathbf{Z}|\mathbf{X})$. By imposing a penalty term $\mathcal{D}(E_{\mathbf{Z}}, P_{\mathbf{Z}})$ to relax the constraint on $E_{\mathbf{Z}}$, we finally obtain the objective of the Wasserstein embedding module:

$$\begin{aligned} L_{\text{WE}} = &\inf_{E(\mathbf{Z}|\mathbf{X}) \in \mathbf{E}} \mathbb{E}_{P_{\mathbf{X}}} \mathbb{E}_{E(\mathbf{Z}|\mathbf{X})}[c(\mathbf{X}, G(\mathbf{Z}))] \\ &+ \beta \cdot \mathcal{D}(E_{\mathbf{Z}}, P_{\mathbf{Z}}), \end{aligned} \tag{5}$$

where $\mathbf{E}$ denotes a collection of non-parametric probabilistic encoders, and β is a hyper-parameter. $\mathcal{D}$ indicates an arbitrary measure of the discrepancy between $E_{\mathbf{Z}}$ and $P_{\mathbf{Z}}$, and the different choices of $\mathcal{D}$ allow us to construct various types of clustering networks. Note that the encoder E and decoder G are constructed by fully connected networks.

B. Self-Optimized Clustering Module

Since the embedded features obtained by Wasserstein embedding learning are not guaranteed to be suitable for clustering tasks, we develop the self-optimized clustering layer to include a clustering-oriented objective in the network optimization. The embedded representation $\mathbf{Z}$ learned from the Wasserstein embedding learning module is used as the input of the clustering layer.

To be specific, the clustering loss L_{C} can be formalized as the KL-divergence between a target distribution T and a soft label distribution S as follows:

$$L_{\text{C}} = \text{KL}(T \,||\, S) = \sum_{i=1}^{n} \sum_{j=1}^{K} t_{ij} \log \frac{t_{ij}}{s_{ij}}, \tag{6}$$

where s_{ij} denotes the similarity between the learned embedded feature $\mathbf{z}_i$ and the cluster centroid $\boldsymbol{\mu}_j$. Through employing Student's t-distribution to measure the similarity, s_{ij} can be defined as follows:

$$s_{ij} = \frac{(1 + \|\mathbf{z}_i - \boldsymbol{\mu}_j\|^2)^{-1}}{\sum_{j=1}^{K} (1 + \|\mathbf{z}_i - \boldsymbol{\mu}_j\|^2)^{-1}}. \tag{7}$$

It can also be considered a soft label allocation indicating the probability of assigning a sample i for the cluster j. In addition, the target distribution T is computed from soft label distribution S and is formalized as follows:

$$t_{ij} = \frac{s_{ij}^2 / \sum_i s_{ij}}{\sum_j s_{ij}^2 / \sum_i s_{ij}}, \tag{8}$$

where t_{ij} emphasizes the soft label allocations with higher probability by raising s_{ij} to the second power. The clustering module aims to match the soft label distribution S to the target distribution T, thus producing soft labels with high confidence to guide the clustering. Therefore, we can regard the clustering module as a self-optimized network.

C. Joint Optimization Strategy

To obtain a clustering-oriented representation, we jointly optimize the Wasserstein embedding and clustering with the following loss function:

$$\begin{aligned} L =& L_{\text{WE}} + \gamma \cdot L_{\text{C}} \\ =& \inf_{E(\mathbf{Z}|\mathbf{X}) \in \mathbf{E}} \mathbb{E}_{P_{\mathbf{X}}} \mathbb{E}_{E(\mathbf{Z}|\mathbf{X})}[c(\mathbf{X}, G(\mathbf{Z}))] \\ &+ \beta \cdot \mathcal{D}(E_{\mathbf{Z}}, P_{\mathbf{Z}}) + \gamma \cdot \text{KL}(T \,||\, S), \end{aligned} \tag{9}$$

where β and γ control the contributions of regularization penalty term and clustering loss, respectively. $c(\mathbf{X}, G(\mathbf{Z}))$ represents the reconstruction loss, which is implemented by

$$c(\mathbf{X}, G(\mathbf{Z})) = \|\mathbf{X} - G(\mathbf{Z})\|_F^2 = L_{\text{r}}. \tag{10}$$

It is worth mentioning that the multiple choices of $\mathcal{D}$ allow us to construct various clustering algorithms. By applying two different $\mathcal{D}$, we propose GAN-based and MMD-based Wasserstein embedding clustering, namely WEC-GAN and WEC-MMD.

1) WEC-GAN: By applying the JS-divergence, we utilize $\mathcal{D}(E_{\mathbf{Z}}, P_{\mathbf{Z}}) = \mathcal{D}_{\text{JS}}(E_{\mathbf{Z}}, P_{\mathbf{Z}})$ to measure the discrepancy between $E_{\mathbf{Z}}$ and $P_{\mathbf{Z}}$, which can be defined as follows:

$$\mathcal{D}_{\text{JS}}(E_{\mathbf{Z}}, P_{\mathbf{Z}}) = \tfrac{1}{2}\text{KL}(E_{\mathbf{Z}} \parallel \tfrac{E_{\mathbf{Z}}+P_{\mathbf{Z}}}{2}) + \tfrac{1}{2}\text{KL}(P_{\mathbf{Z}} \parallel \tfrac{P_{\mathbf{Z}}+E_{\mathbf{Z}}}{2}). \tag{11}$$

Meanwhile, adversarial training is used in the optimization. To be specific, the encoder $E(\mathbf{Z}|\mathbf{X})$ serves as the generator in this case, and a discriminator D is introduced in the embedding space to distinguish between true samples from prior $P_{\mathbf{Z}}$ and fake samples from generator $E_{\mathbf{Z}}$. Compared with other GAN-based methods, the game of generator and discriminator in WEC-GAN plays in the embedding space rather than the original data space. The training strategy of WEC-GAN is described in Algorithm 1.

2) WEC-MMD: Due to the advantage in matching high dimensional standard normal distribution, the maximum mean discrepancy is another optional way of measuring the discrepancy between $E_{\mathbf{Z}}$ and $P_{\mathbf{Z}}$. Therefore, we propose to employ $\mathcal{D}(E_{\mathbf{Z}}, P_{\mathbf{Z}}) = \mathcal{D}_{\text{MMD}}(E_{\mathbf{Z}}, P_{\mathbf{Z}})$, which is defined as follows:

$$\mathcal{D}_{\text{MMD}}(E_{\mathbf{Z}}, P_{\mathbf{Z}}) = \left\| \int_{\mathcal{Z}} k(\mathbf{z}, \cdot) dE_{\mathbf{Z}}(\mathbf{z}) - \int_{\mathcal{Z}} k(\mathbf{z}, \cdot) dP_{\mathbf{Z}}(\mathbf{z}) \right\|_{\mathcal{H}_k}, \tag{12}$$

where $k : \mathcal{Z} \times \mathcal{Z} \to \mathcal{R}$ denotes the positive definite reproducing kernel and $\mathcal{H}_k$ is the reproducing kernel Hilbert space of the mapping function from $\mathcal{Z}$ to $\mathcal{R}$. Note that we construct an adversary-free model in this case, i.e., the model attempts to solve a min-min optimization issue. The training strategy of WEC-MMD is described in Algorithm 2.

Since the reconstruction loss is too weak to be effective at the beginning, which prevents us from obtaining meaningful representation, we first pre-train an auto-encoder to initialize the network parameters. Then, we perform the joint optimization to improve the embedding learning. Stochastic gradient descent (SGD) is utilized to update the embedded representation $\mathbf{Z}$ and cluster centroid μ according to the gradient of L w.r.t. $\mathbf{Z}$ and μ. In the training process, the target distribution T serves as the ground truth, and it is determined by soft label distribution S. However, updating T and S at each iteration is risky, because it will cause a constant change of the objective, which hinders the embedding learning and convergence. To prevent instability in training, T is updated every five iterations in practice. Finally, with the updated S, we can derive the final allocated label v_i for sample $\mathbf{x}_i$ as follows:

$$v_i = \arg\max_j s_{ij}, \tag{13}$$

where v_i denotes the allocation of the category j with the highest confidence to sample $\mathbf{x}_i$.

Algorithm 1 Wasserstein embedding clustering based on GAN (WEC-GAN)

Input: Original data $\mathbf{X}$, number of clusters K, parameters β and γ, number of epochs MaxEpoch, learning rate α, interval for update O.
Output: The predicted cluster labels V.

Initialize encoder E, decoder G, discriminator D and cluster centroid $\boldsymbol{\mu}$;
for $epoch = 1$ to MaxEpoch **do**
  ———Forward Propagation———
  Obtain embedded representation $\mathbf{Z}$;
  Calculate soft label distribution S using Eq. (7);
  if $epoch\%O == 0$ **then**
    Compute target distribution T according to Eq. (8) ;
  end if
  Sample $\tilde{\mathbf{Z}}$ from the prior $P_{\mathbf{Z}}$;
  Calculate the the reconstruction loss L_{r} via Eq. (10);
  Calculate the penalty term $\mathcal{D}_{\text{JS}}(E_{\mathbf{Z}}, P_{\mathbf{Z}})$ via Eq. (11);
  Calculate the the clustering loss L_{C} via Eq. (6);
  ———Backward Propagation———
  Sample a batch with n_b samples from $\mathbf{X}$;
  Update the discriminator D via back-propagation and ascending $D = D + \frac{\alpha}{n_b}\sum_{i=1}^{n_b}(\log D(\tilde{\mathbf{z}}_i) + \log(1 - D(\mathbf{z}_i))$;
  Update the cluster centroid $\boldsymbol{\mu}_j$ via $\boldsymbol{\mu}_j = \boldsymbol{\mu}_j - \frac{\alpha}{n_b}\sum_{i=1}^{n_b}\frac{\partial L_{\text{C}}}{\partial \boldsymbol{\mu}_j}$;
  Update the weight parameter $\boldsymbol{\Theta}'$ of the decoder G via $\boldsymbol{\Theta}' = \boldsymbol{\Theta}' - \frac{\alpha}{n_b}\sum_{i=1}^{n_b}(\frac{\partial L_{\text{r}}}{\partial \boldsymbol{\Theta}'} + \beta \cdot \frac{\partial \mathcal{D}_{\text{JS}}}{\partial \boldsymbol{\Theta}'})$;
  Update the weight parameter $\boldsymbol{\Theta}$ of the generator (encoder) E via $\boldsymbol{\Theta} = \boldsymbol{\Theta} - \frac{\alpha}{n_b}\sum_{i=1}^{n_b}(\frac{\partial L_{\text{r}}}{\partial \boldsymbol{\Theta}} + \beta \cdot \frac{\partial \mathcal{D}_{\text{JS}}}{\partial \boldsymbol{\Theta}} + \gamma \cdot \frac{\partial L_{\text{C}}}{\partial \boldsymbol{\Theta}})$;
end for
Obtain clustering result from V according to Eq. (13);
return The predicted cluster labels V.

D. Discussion with GAN-based and VAE-based approaches

Here, we briefly discuss the similarities and differences between our method and the GAN-based and VAE-based approaches. We see that the proposed WEC-GAN constructs a

Algorithm 2 Wasserstein embedding clustering based on MMD (WEC-MMD)

Input: Original data $\mathbf{X}$, number of clusters K, parameters β and γ, number of epochs MaxEpoch, learning rate α, interval for update O.

Output: The predicted cluster labels V.

Initialize encoder E, decoder G and cluster centroid $\boldsymbol{\mu}$;
for $epoch = 1$ to MaxEpoch **do**
  ———Forward Propagation———
  Obtain embedded representation $\mathbf{Z}$;
  Calculate soft label distribution S using Eq. (7);
  if $epoch\%O == 0$ **then**
    Update target distribution T according to Eq. (8) ;
  end if
  Sample $\tilde{\mathbf{Z}}$ from the prior $P_{\mathbf{Z}}$;
  Calculate the the reconstruction loss L_{r} via Eq. (10);
  Calculate the penalty term $\mathcal{D}_{\mathrm{MMD}}(E_{\mathbf{Z}}, P_{\mathbf{Z}})$ via Eq. (12);
  Calculate the the clustering loss L_{C} via Eq. (6);
  ———Backward Propagation———
  Sample a batch with n_b samples from $\mathbf{X}$;
  Update the cluster centroid $\boldsymbol{\mu}_j$ via $\boldsymbol{\mu}_j = \boldsymbol{\mu}_j - \frac{\alpha}{n_b}\sum_{i=1}^{n_b}\frac{\partial L_{\mathrm{C}}}{\partial \boldsymbol{\mu}_j}$;
  Update the weight parameter $\boldsymbol{\Theta}'$ of the decoder G via $\boldsymbol{\Theta}' = \boldsymbol{\Theta}' - \frac{\alpha}{n_b}\sum_{i=1}^{n_b}(\frac{\partial L_{\mathrm{r}}}{\partial \boldsymbol{\Theta}'} + \beta \cdot \frac{\partial \mathcal{D}_{\mathrm{MMD}}}{\partial \boldsymbol{\Theta}'})$;
  Update the weight parameter $\boldsymbol{\Theta}$ of the generator (encoder) E via $\boldsymbol{\Theta} = \boldsymbol{\Theta} - \frac{\alpha}{n_b}\sum_{i=1}^{n_b}(\frac{\partial L_{\mathrm{r}}}{\partial \boldsymbol{\Theta}} + \beta \cdot \frac{\partial \mathcal{D}_{\mathrm{MMD}}}{\partial \boldsymbol{\Theta}} + \gamma \cdot \frac{\partial L_{\mathrm{C}}}{\partial \boldsymbol{\Theta}})$;
end for
Obtain clustering result from V according to Eq. (13);
return The predicted cluster labels V.

min-max adversarial training, which is similar to GAN-based methods [47], [50]. Nevertheless, compared with them, WEC-GAN conducts the adversarial game in the embedding space $\mathcal{Z}$ rather than in the data space $\mathcal{X}$. In this case, we only need to match an easier prior distribution $P_{\mathbf{Z}}$ (Gaussian distribution) instead of a more complex data distribution $P_{\mathbf{X}}$.

Furthermore, similar to VAE [16], the Wasserstein embedding module in our method also contains both the reconstruction loss and penalty term. The reconstruction loss aims to learn an encoding-decoding mapping that accurately encodes the input data as low-dimensional embedded features and then reconstructs them inversely. While the regularizer in Wasserstein embedding is different from VAE. Specifically, VAE forces the posterior $E(\mathbf{Z}|\mathbf{X} = \mathbf{x})$ to match $P_{\mathbf{Z}}$ for all the data $\mathbf{X}$ sampled from $P_{\mathbf{X}}$. In contrast, the Wasserstein embedding forces the aggregated posterior $E_{\mathbf{Z}}$ to match $P_{\mathbf{Z}}$, so that the features learned from different samples may be far from each other, thereby obtaining more discriminative embedded features for clustering. The clustering visualization in Section IV-E also demonstrates that our method can reveal better cluster structure compared to VAE. Besides, another difference to most GAN-based and VAE-based approaches is that our method explicitly defines a clustering objective to jointly optimize the embedding learning and clustering, which enables the model to learn clustering-oriented representation.

E. Computational Complexity

Assuming that d_{max} represents the maximal dimensions of the hidden layer, the time complexity of WEC model can be calculated as $\mathcal{O}(nd_{max}^2 + nd_zK + nd_z^2)$, where n and K indicate the number of samples and clusters, d_z denotes the size of embedding layer. Since $K \leq d_z \leq d_{max}$, the time complexity can be further simplify to $\mathcal{O}(nd_{max}^2)$. Consequently, our method is efficient as it can be regarded as a linear clustering method. The time complexity of our method is of the same order of magnitude as some well-known deep clustering methods such as IDEC's $\mathcal{O}(nd_{max}^2 + nd_zK)$ and DCSAIP's $\mathcal{O}(nd_{max}^2)$.

IV. Experiment

In this section, the detailed experimental settings are first illustrated, which contains the description of datasets, comparative methods, and implementation details. Then, we introduce the evaluation metrics used in the experiments. Finally, we conduct a comprehensive experiment to verify the effectiveness of WEC from different perspectives.

A. Experimental Settings

1) Datasets: For validating the effectiveness and competitiveness of WEC compared to other approaches, nine publicly available databases are employed: MNIST[1], Fashion-MNIST[2], STL-10[3], CIFAR-10[4], Reuters-10K [52], ImageNet-10[5], ImageNet-Dog-15, and Tiny-ImageNet[6], and we only briefly describe each database here.

- **MNIST** incorporates 70,000 hand-written digit samples, and 60,000 of them constitute the training set with the rest for the test. They are grouped into 10 different classes, and each sample comes with a 28×28 size.
- **Fashion-MNIST** consists of 10 different kinds of fashion items, including trousers, dresses, sneakers, etc. This dataset comes with an identical number of samples and image sizes to MNIST.
- **STL-10** comprises 13,000 real-world images in 10 different categories such as car, dog, truck, etc. Each sample is a 96×96 RGB image.
- **CIFAR-10** is a database comprising 32×32 RGB images from 10 categories, including 50,000 training examples and 10,000 test examples.
- **REUTERS-10K** is the sub-set of the REUTERS database that covers 10,000 documents in 4 categories. Each document is expressed as a vector of 2,000 dimensions.
- **COIL-20** consists of 1,440 samples with 20 objects in total, each sample is captured from different viewpoints under varying lighting conditions.
- **ImageNet-10** is a widely used subset of ImageNet, which is composed of 13,000 samples in 10 classes.

[1]http://yann.lecun.com/exdb/mnist/
[2]https://github.com/zalandoresearch/fashion-mnist
[3]https://cs.stanford.edu/ acoates/stl10/
[4]http://www.cs.toronto.edu/ kriz/cifar.html
[5]https://image-net.org/download.php
[6]https://www.kaggle.com/c/tiny-imagenet

- **ImageNet-Dog-15** is a subset of the ImageNet dataset focused specifically on dog breeds. It comprises 19,500 samples with a wide variety of dog images belonging to 15 different breeds.
- **Tiny-ImageNet** is a subset of ImageNet, which contains 100,000 training samples and 10,000 testing samples from 20 different superclasses.

The number of instances and classes of the nine datasets, as well as their sizes, are summarized in Table II.

TABLE II
THE DETAILED DESCRIPTION OF THE NINE EXPERIMENTAL DATASETS.

Database	# Samples	# Size	# Classes
MNIST	70,000	28×28	10
Fashion-MNIST	70,000	28×28	10
STL-10	13,000	96×96×3	10
CIFAR-10	60,000	32×32×3	10
REUTERS-10K	10,000	2,000	4
COIL-20	1,440	128×128	20
ImageNet-10	13,000	96×96×3	10
ImageNet-Dog-15	19,500	96×96×3	15
Tiny-ImageNet	110,000	64×64×3	20

2) Comparative Methods: To guarantee the persuasiveness of comparison, the proposed WEC method is compared with several state-of-the-art clustering approaches, including k-means [1], Auto-encoder (AE) [13], VAE [16], CatGAN [53], Gaussian mixture variational auto-encoder (GMVAE) [45], Deep embedding clustering (DEC) [10], Improved deep embedding clustering (IDEC) [39], Variational deep embedding (VaDE) [25], JULE [11], Deep embedded regularized clustering (DEPICT) [14], ClusterGAN [47], Deep clustering with sample-assignment invariance prior (DCSAIP) [40], VaGAN-GMM [50], Deep clustering with contractive representation learning and focal loss (DCCF) [54], Progressive affinity diffusion (PAD) [55] Deep self-evolution clustering (DSEC) [56] GATCluster [57], Partition confidence maximisation (PICA) [58], Mixture of contrastive experts (MiCE) [59], and Nearest neighbor contrastive clustering (NNCC) [60].

3) Implementation Details: To be fair for comparison, all deep learning-based clustering models employ the same network structure. The encoder is constructed with d-500-500-1,000-d' fully connected network, and the decoder is constructed symmetrically with it accordingly, which is taken inspiration from some excellent deep learning works [61], [62]. Besides, d' denotes the dimensions of the embedding layer and is set with the number of categories per dataset. Note that we applied the ResNet50 model to extract 2,048-dimensional features for STL-10 in our experiment. For our method, we use the isotropic Gaussian distribution $P_{\mathbf{Z}}(\mathbf{Z}) = \mathcal{N}(\mathbf{Z}; \mathbf{0}, \sigma_{\mathbf{z}}^2 \cdot \boldsymbol{I}_d)$ as the prior distribution and Adam as the optimizer. The learning rate α is set as $\alpha = 0.001$, and the training epochs are fixed at 200. Regarding the settings of the two parameters β and γ in our objective function, we discuss their effects on the clustering performance at different values in Section IV-F, and give their recommended ranges. In addition, for WEC-GAN, we construct the discriminator by a full-connected network with size d_z-128-1. For WEC-MMD, we employ

$$k(\mathbf{x}, \mathbf{y}) = 2d_{\mathbf{z}}\sigma_{\mathbf{z}}^2 / (2d_{\mathbf{z}}\sigma_{\mathbf{z}}^2 + \|\mathbf{x} - \mathbf{y}\|_2^2), \tag{14}$$

which is called the inverse multi-quadratics kernel.

B. Evaluation Metrics

Two popular metrics for clustering analysis, including Clustering Accuracy (ACC) and Normalized Mutual Information (NMI), are employed to assess the clustering performance in our experiment.

1) Clustering Accuracy: ACC measures the clustering performance by comparing the resulting predicted labels with the ground truth labels. Let $\mathbf{y}$ be the ground truth labels, and $\mathbf{p}$ be the predicted labels, ACC is defined as follows:

$$\text{ACC} = \frac{\sum_{i=1}^{n} \delta(\mathbf{y}_i, map(\mathbf{p}_i))}{n}, \tag{15}$$

where n denotes the sample size, and $map(\cdot)$ indicates a mapping that makes the predicted labels from the clustering algorithm match the true labels best. Generally, the optimal mapping is available through the Hungarian algorithm, which allows us to address the label allocation problem in polynomial time. Moreover, it is worth mentioning that $\delta(x, y) = 1$ only when $x = y$, while $\delta(x, y) = 1$ in other cases.

2) Normalized Mutual Information: Based on a common measure within information theory, i.e., mutual information (MI), NMI is defined. Specifically, let random variables A and B be discrete, then MI is formalized as follows:

$$MI(A, B) = H(A) + H(B) - H(A, B), \tag{16}$$

where the information entropy is applied to calculate $H(A)$ and $H(B)$, and $H(A, B)$ is defined by the joint information entropy of A and B. Consequently, the definition of NMI is given below:

$$\text{NMI}(A, B) = 2\frac{MI(A, B)}{H(A) + H(B)}, \tag{17}$$

it can be regarded as the normalization of MI, and the value of NMI represents the correlation between two variables. Note that the values of ACC and NMI both range from $[0, 1]$, and a higher score means better clustering performance.

C. Experimental Results

The clustering results of the WEC method and other comparative approaches on the six experimental benchmarks are presented in Table III. As can be seen, our method outperforms the other comparative approaches for most cases, which illustrates the superiority of WEC in the clustering task.

It can be observed that the approaches with joint optimization of embedding learning and clustering perform better than those only optimized for embedding learning. For instance, the ACC and NMI scores of DEPICT, JULE, DCSAIP, WEC-MMD, and WEC-GAN are significantly better than k-means, AE, and VAE on the MNIST dataset. This observation illustrates that the clustering objective incorporated in the model

TABLE III
CLUSTERING PERFORMANCE ON SIX EXPERIMENTAL DATASETS. NOTE THAT THE TOP TWO CLUSTERING RESULTS ARE MARKED IN **BOLD**.

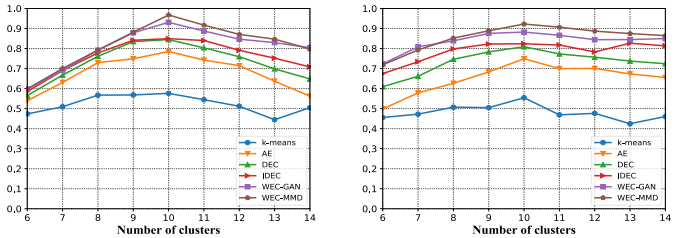
Methods/Datasets	MNIST		Fashion-MNIST		CIFAR-10		STL-10		REUTERS-10K		COIL-20	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
k-means [1]	57.62	55.43	56.34	52.57	22.19	8.24	28.35	23.48	54.08	35.24	47.33	48.00
AE [13]	78.53	74.90	56.72	55.35	21.63	6.71	34.83	30.08	59.76	32.36	68.74	69.52
VAE [16]	72.48	68.13	60.78	57.58	25.01	10.53	57.69	55.50	62.52	32.96	68.59	70.48
CatGAN [53]	82.79	76.37	55.00	60.00	31.52	26.46	29.84	21.00	59.32	32.38	–	–
GMVAE [45]	88.54	79.64	59.60	57.00	33.64	28.47	31.77	23.65	60.08	37.96	71.32	73.80
DEC [10]	84.46	80.91	58.69	59.09	22.37	9.62	36.12	31.82	61.85	31.46	68.90	70.30
IDEC [39]	84.92	82.37	59.23	60.42	23.49	10.38	37.80	32.46	68.43	35.15	71.01	72.80
VaDE [25]	94.50	87.60	55.20	57.30	15.60	3.60	—	—	72.30	41.60	75.48	77.90
JULE [11]	96.40	91.30	56.30	60.80	27.15	19.23	27.69	18.15	—	—	—	—
DEPICT [14]	96.50	91.70	39.20	39.20	32.60	27.40	37.10	30.30	—	—	—	—
ClusterGAN [47]	96.40	92.10	—	—	41.20	32.30	42.30	33.50	—	—	77.00	81.10
DCSAIP [40]	87.16	75.50	—	—	22.06	7.02	—	—	69.81	34.33	—	—
DCCF [54]	**97.41**	**93.32**	62.12	**64.58**	45.81	**36.19**	**72.78**	**66.84**	**83.36**	**55.52**	78.51	82.62
VaGAN-GMM [50]	95.48	91.70	**63.84**	**63.30**	28.79	15.80	—	—	80.12	53.60	79.20	85.10
WEC-GAN	93.07	88.20	**62.34**	62.74	**47.41**	33.38	65.42	66.01	**81.77**	**55.59**	**84.03**	**88.15**
WEC-MMD	**96.74**	**92.23**	62.20	62.96	**49.24**	**37.03**	**70.44**	**67.27**	80.30	51.29	**82.71**	**87.80**

is conducive to guiding the network to obtain the embedded representation appropriate for clustering.

The clustering methods based on generative models also exhibit remarkable clustering results. For example, ClusterGAN and VaGAN-GMM outperform deterministic mapping-based methods such as DEC, IDEC, and DCSAIP on MNIST, CIFAR-10, and STL-10. Yet, these two methods promote clustering in different ways. ClusterGAN improves clustering through a clustering-oriented idea, for which a clusterer is introduced to provide the clustering objective, thus enabling the network to learn more discriminative latent representation. While VaGAN-GMM improves clustering by combining VAE and WGAN to enhance the representation capability of the generator. In addition, VaDE also shows competitive performance on MNIST and REUTERS-10K.

It is noteworthy that as a generative deep clustering algorithm, our approach considers both enhancing the representation capability of the embedding learning module and introducing a clustering objective. The clustering results of our proposed WEC-MMD method are significantly better than ClusterGAN and VaGAN-GMM. Taking the CIFAR-10 database as an instance, WEC-MMD improves 8.04% and 20.45% over ClusterGAN and VaGAN-GMM in terms of ACC. Moreover, our approach also shows competitive performance on other databases, which is mainly attributed to two aspects. First, our method holds the encoder-decoder structure, which can better preserve the data structure during training. Second, the introduced Wasserstein embedding learning improves the representation learning capability and training stability of the network to capture more discriminative embedded features. Third, incorporating the clustering-oriented objective in the Wasserstein embedding allows us to obtain a more suitable representation for clustering in the embedding learning.

Furthermore, Figure 2 shows the clustering performance of six clustering methods with different numbers of clusters on MNIST. As can be seen, WEC-GAN and WEC-MMD exhibit remarkable advantages when the number of clusters is no less than that of true categories. In general, compared with other methods, WEC-GAN and WEC-MMD show smaller fluctuations in the clustering performance when the number of clusters is varied, which fully demonstrates the robustness of the proposed approach.



(a) ACC (b) NMI

Fig. 2. Clustering results on MNIST with different cluster numbers.

D. Feasibility in large-scale clustering

To evaluate the feasibility of the proposed methods on large-scale clustering, we conducted additional experiments on the ImageNet-10 and ImageNet-Dog-15 datasets [57], [58], [60]. These datasets are known for their extensive scale and complexity, making them popular image benchmarks for assessing clustering algorithms [58], [60]. Note that we adopted the MOCO [63] model pre-trained with self-supervised learning as the feature extractor to preprocess the datasets in the experiment, and compare the proposed WEC-GAN and WEC-MMD with eleven strong baselines.

Table IV summarizes the experimental results on the three datasets. We can observe that both WEC-GAN and WEC-MMD exhibit outstanding clustering performance, surpassing other competing methods across various evaluation metrics. For instance, on the ImageNet-10 dataset, WEC-MMD

TABLE IV
CLUSTERING PERFORMANCE ON THREE LARGE-SCALE IMAGE BENCHMARKS. NOTE THAT THE TOP TWO CLUSTERING RESULTS ARE MARKED IN **BOLD**.

Methods	ImageNet-10		ImageNet-Dog-15		Tiny-ImageNet	
	ACC	NMI	ACC	NMI	ACC	NMI
k-means [1]	24.10	11.90	10.50	5.50	2.50	6.50
AE [13]	31.70	21.00	18.50	10.40	4.10	13.10
VAE [16]	33.40	19.30	17.90	10.70	3.60	11.30
DEC [10]	38.10	28.20	17.50	12.20	3.70	11.50
JULE [11]	30.00	17.50	13.80	5.40	3.30	10.20
PAD [55]	65.40	57.80	33.60	32.70	9.80	25.60
DSEC [56]	67.40	58.30	26.40	23.60	6.60	19.00
GATCluster [57]	76.20	60.90	33.30	32.20	—	—
PICA [58]	87.00	80.20	35.20	35.20	9.80	27.70
MiCE [59]	—	—	**43.90**	42.30	—	—
NNCC [60]	75.10	68.30	40.10	37.20	14.10	33.30
WEC-GAN	**92.20**	**85.45**	43.72	**44.09**	**17.68**	**38.29**
WEC-MMD	**93.93**	**88.09**	**45.99**	**46.24**	**21.46**	**40.75**

achieves improvements of 6.93% and 7.89% in terms of ACC and NMI, compared to the runner-up MiCE method. Furthermore, on the challenging Tiny-ImageNet dataset with 200 classes, WEC-GAN and WEC-MMD demonstrate remarkable competitiveness, significantly outperforming other comparative methods. These experimental results strongly establish the feasibility and effectiveness of our proposed methods for large-scale clustering scenarios.

E. Experimental Visualization

To intuitively compare the clustering results, we also employ the t-SNE method to visualize the embedded representation learned on the MNIST dataset by seven algorithms, including AE, VAE, DEC, IDEC, VaDE, WEC-GAN, and WEC-MMD. Figure 3 shows the visualization of clustering results, and the original distribution is used as the baseline. We can see that in the t-SNE visualization of the original distribution, the data points are mostly obfuscated, and it is difficult to see the explicit clustering structure. As we look at the visualization of AE and VAE, we can find a clearer clustering structure, although some of their categories are still confusing. It is worth noting that since there is no embedding learning module in the DEC training, we can see that its distribution is elongated, which implies larger intra-class distances. On the contrary, the visualizations obtained by the approaches with joint optimization of embedding learning and clustering, such as VaDE, IDEC, and our methods, exhibit better clustering structures. More specifically, their visualizations show a circular structure with more compact data distribution, i.e., smaller intra-class distances, which suggests that the embedding learning module helps the network to better capture the data structure of the embedding space. In addition, the clusters in WEC-MMD and WEC-GAN can be more clearly distinguished, and the different classes are more separated from each other than in VaDE and IDEC, which demonstrates the power of the generative models in revealing the data structure.

In addition, we further provide a visualization of clustering results in Figure 4 to understand more intuitively how the algorithm divides the different clusters. Specifically, we run WEC-MMD on STL-10, then randomly select ten samples from the first four clusters and artificially label and color them. From this figure, we have some interesting observations. First, although a monkey is mistakenly clustered with the birds in the second row, they have a very common characteristic: they are climbing in the trees, and the posture of the monkey is slightly similar to that of the birds. Second, we also observe that things with more commonalities are often confused in clustering, such as cars and trucks, cats and dogs in the third and fourth rows. This is consistent with human perception since cars and trucks, cats, and dogs have many similarities in appearance. Third, this visualization also supports the validity of our algorithm, as the samples within each cluster exhibit high similarities.

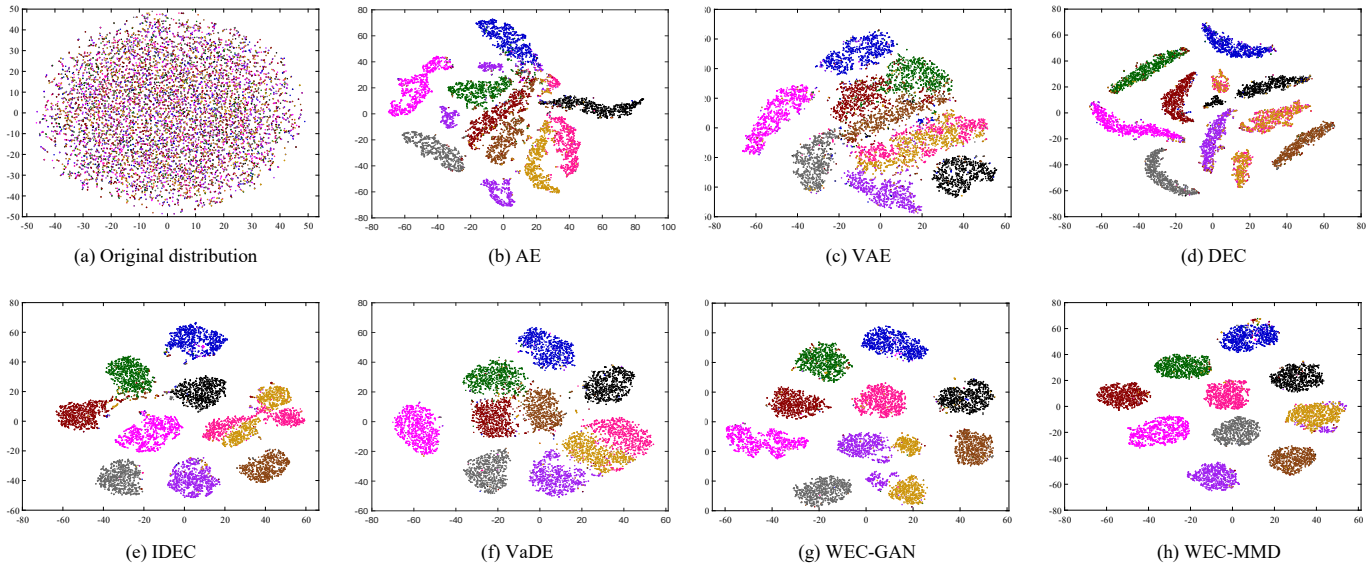
F. Parameter Sensitivity

The proposed WEC model contains two hyper-parameters γ and β, in which γ controls the contribution of clustering loss L_C and β controls the contribution of Wasserstein embedding. By varying the values of these two parameters, we evaluate their influence on clustering performance. Figure 5 and 6 present the clustering results of WEC-GAN and WEC-MMD on five datasets (MNIST, Fashion-MNIST, CIFAR-10, STL-10, and REUTERS-10), where the value of γ ranges from 0.01 to 10 and β takes values from 0.0001 to 0.1. The impacts of the two parameters are analyzed as follows.

First, for WEC-GAN, we can observe that the clustering performance decreases when γ is less than 0.05, especially on the STL-10 dataset. The reason is that too small γ causes the clustering loss to be overlooked, and the clustering module cannot effectively guide the soft label allocation. For WEC-MMD, we can notice from the clustering result on STL-10 and REUTERS-10K that the ACC and NMI maintain a good clustering performance when γ ranging in $[0.05, 10]$ and β ranging in $[0.0001, 0.005]$. While fluctuations occur when β is greater than 0.005. Nevertheless, the performance of WEC-GAN and WEC-MMD with different values of γ and β remains relatively stable over most of the datasets, demonstrating the robustness of the proposed two methods.

Second, taking the MNIST database as an example, we can see the clustering performance improve when the value of γ increases from 0.01 to 1 and the value of β increases from 0.0001 to 0.005. However, when the values of these two parameters are further increased, both the ACC and NMI of two variants fluctuate although they remain generally favorable. This is good evidence that the two loss terms controlled by γ and β are meaningful for the clustering tasks.

Third, it can be seen that compared with NMI, ACC is more sensitive to these two parameters from the experimental results of the STL-10 in Figures 5 and 6. Moreover, we can also observe different ranges of optimal values of γ and β for different datasets. While in general, the clustering performance is improved with the values of γ and β in a certain range. Finally, we provide a general recommended range of values regarding these two parameters for two WEC variants, as they can achieve promising clustering performance under the



(a) Original distribution (b) AE (c) VAE (d) DEC

(e) IDEC (f) VaDE (g) WEC-GAN (h) WEC-MMD

Fig. 3. 2D visualization provided by employing t-SNE on the learned embedded representation of the MNIST dataset. Note that we select the test set to provide the visualization, and the original data is used as the baseline.

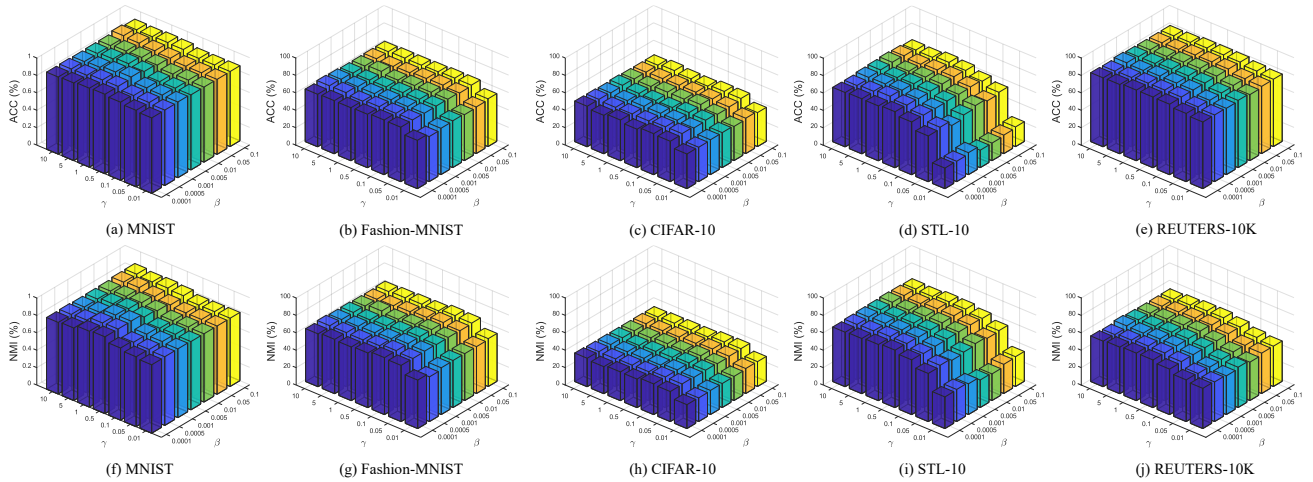
Fig. 4. Visualization of the clustering results on the STL-10 dataset. Note that we run WEC-MMD on STL-10, then randomly select 10 samples from the first 4 clusters and artificially label and color them. (Green for correct identification and red for incorrect identification).

specified range. The recommended parameter values range from $\gamma \in [0.5, 1]$ and $\beta \in [0.0001, 0.005]$ for WEC-GAN, while $\gamma \in [0.05, 0.1]$ and $\beta \in [0.0001, 0.01]$ for WEC-MMD.

G. Ablation Study

The proposed WEC method is composed of a Wasserstein embedding module and a clustering module, and the joint optimization of these two modules allows us to obtain an encouraging clustering performance. Towards demonstrating the validity of the Wasserstein embedding and clustering layer, we conduct the ablation study on three datasets, including MNIST, Fashion MNIST, and REUTERS-10K. Table V shows the comparison between WEC and its degraded models that remove the specified module, from which we have the following observations.

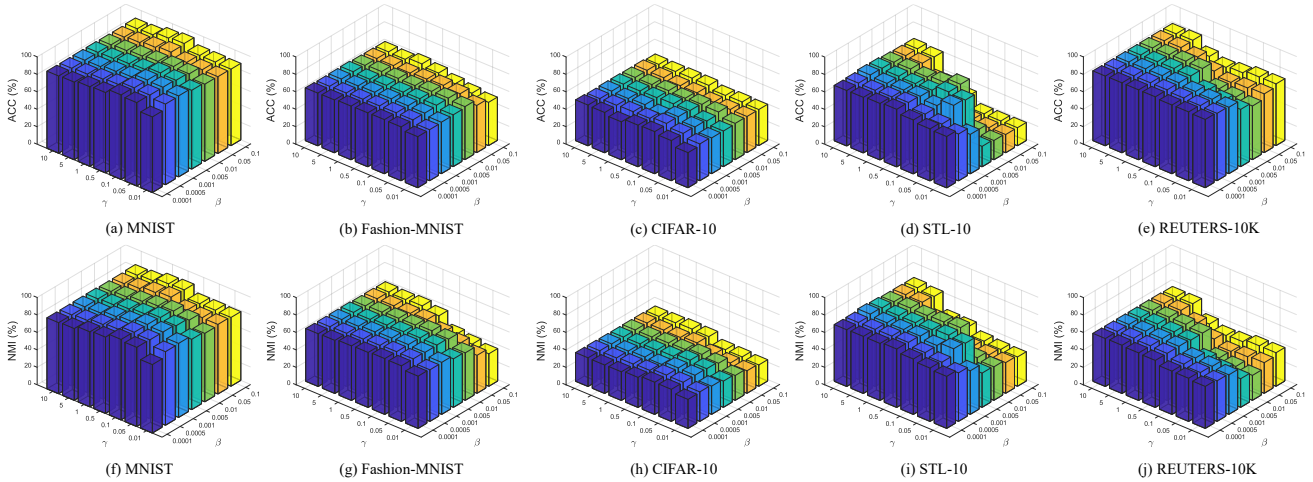
First, WEC-MMD and WEC-GAN perform significantly better than the degraded model $\text{WEC}_{\text{w/o}\mathcal{D}(E_{\mathbf{Z}},P_{\mathbf{Z}})}$ that removes the penalty term $\mathcal{D}(E_{\mathbf{Z}}, P_{\mathbf{Z}})$. This observation indicates that the Wasserstein embedding can better reveal the data structure and thus allow the model to acquire more represen-



(a) MNIST (b) Fashion-MNIST (c) CIFAR-10 (d) STL-10 (e) REUTERS-10K

(f) MNIST (g) Fashion-MNIST (h) CIFAR-10 (i) STL-10 (j) REUTERS-10K

Fig. 5. Clustering performance variations of WEC-GAN with two hyper-parameters γ and β on the five datasets. The first and second rows show the ACC and NMI respectively. Note that the values of γ vary from 0.01 to 10, and β takes values from 0.0001 to 0.1.



(a) MNIST (b) Fashion-MNIST (c) CIFAR-10 (d) STL-10 (e) REUTERS-10K

(f) MNIST (g) Fashion-MNIST (h) CIFAR-10 (i) STL-10 (j) REUTERS-10K

Fig. 6. Clustering performance variations of WEC-MMD with two hyper-parameters γ and β on the five datasets. The first and second rows show the ACC and NMI respectively. Note that the values of γ vary from 0.01 to 10, and β takes values from 0.0001 to 0.1.

TABLE V
CLUSTERING RESULTS OF WEC AND ITS DEGRADED MODELS. THE BEST RESULTS ARE MARKED IN **BOLD**.

Methods	MNIST		Fashion-MNIST		REUTERS-10K	
	ACC	NMI	ACC	NMI	ACC	NMI
$\text{WEC}_{\text{w/o}\mathcal{D}(E_{\mathbf{z}},P_{\mathbf{z}})}$	85.65	83.02	59.98	60.21	69.29	36.70
$\text{WEC-GAN}_{\text{w/o}L_{\text{C}}}$	84.39	77.66	58.63	58.26	79.71	50.68
$\text{WEC-MMD}_{\text{w/o}L_{\text{C}}}$	86.15	78.43	57.87	57.17	76.18	45.34
WEC-GAN	93.07	88.20	**62.34**	62.74	**81.77**	**55.59**
WEC-MMD	**96.74**	**92.23**	62.20	**62.96**	80.30	51.29

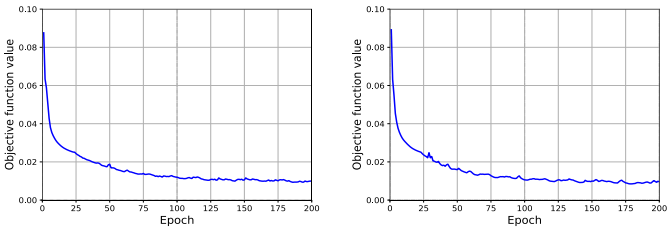
tative features. Second, we find that in the two degraded models $\text{WEC-GAN}_{\text{w/o}L_{\text{C}}}$ and $\text{WEC-MMD}_{\text{w/o}L_{\text{C}}}$, the clustering performance also decreases as the clustering layer removed from WEC-MMD and WEC-GAN. This demonstrates that introducing the clustering module is beneficial for clustering, and it is reasonable to train the model with the clustering objective. Overall, the ablation experiments illustrate that the two modules of WEC are complementary to each other. Through the joint optimization of the Wasserstein embedding and the clustering objective, the model is able to obtain a better clustering assignment.

H. Convergence Analysis

To verify the convergence of the proposed WEC model, we run WEC-GAN and WEC-MMD on MNIST for 200 epochs. Figure 7 presents their convergence curves. We can see from this figure that the objective function values of both two algorithms generally show a decreasing trend and basically reach convergence after 100 epochs, which proves the convergence property of our method. Meanwhile, we can observe that the convergence process of WEC-GAN is smoother than WEC-MMD.

V. CONCLUSION

In this paper, we propose a Wasserstein embedding clustering model to incorporate Wasserstein embedding and clus-



(a) WEC-GAN (b) WEC-MMD

Fig. 7. Convergence curves of WEC-GAN and WEC-MMD on MNIST database.

tering in the embedding space. By solving the optimization problem of optimal transport and introducing the clustering objective in the embedding space, WEC jointly optimizes the Wasserstein embedding and clustering to capture the clustering-oriented representation. Based on the substitutability of penalty term in Wasserstein embedding learning, we choose two different divergences and propose WEC-GAN and WEC-MMD. Experimental results on nine publicly available datasets compared to several state-of-the-art clustering methods show the superiority of our method.

Note that as shown by the visualization part of this paper, our method tends to be less effective when dealing with samples that have significant similarities such as cars and trucks, cats and dogs. In future work, we will focus on the issue of identifying those indistinguishable samples. It is also a promising attempt to extend our approach to multi-modal clustering. Additionally, it is also worth investigating an end-to-end framework that integrates the generative clustering approach with advanced representation learning modules, such as MOCO [63], BYOL [64], and MAE [65], etc. Incorporating these modules into the clustering framework rather than solely adopting them as feature extractors may be able to exploit the synergistic effects from their joint optimization to improve the accuracy and robustness of clustering.

REFERENCES

[1] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[3] T. Zhang, X. Liu, L. Gong, S. Wang, X. Niu, and L. Shen, "Late fusion multiple kernel clustering with local kernel alignment maximization," *IEEE Transactions on Multimedia*, 2021.

[4] X. Liu, L. Liu, Q. Liao, S. Wang, Y. Zhang, W. Tu, C. Tang, J. Liu, and E. Zhu, "One pass late fusion multi-view clustering," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2021, pp. 6850–6859.

[5] K. K. Bharti and P. K. Singh, "Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3105–3114, 2015.

[6] X. Li, G. Cui, and Y. Dong, "Graph regularized non-negative low-rank matrix factorization for image clustering," *IEEE Transactions on Cybernetics*, vol. 47, no. 11, pp. 3840–3853, 2016.

[7] Q. Wang, J. Cheng, Q. Gao, G. Zhao, and L. Jiao, "Deep multi-view subspace clustering with unified and discriminative learning," *IEEE Transactions on Multimedia*, vol. 23, pp. 3483–3493, 2020.

[8] X. Jiang, S. Wei, T. Liu, R. Zhao, Y. Zhao, and H. Huang, "Blind image clustering for camera source identification via row-sparsity optimization," *IEEE Transactions on Multimedia*, vol. 23, pp. 2602–2613, 2020.

[9] J. Fan, C. Yang, and M. Udell, "Robust non-linear matrix factorization for dictionary learning, denoising, and clustering," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1755–1770, 2021.

[10] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proceedings of the International Conference on Machine Learning*, 2016, pp. 478–487.

[11] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5147–5156.

[12] C.-C. Hsu and C.-W. Lin, "Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data," *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 421–429, 2017.

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

[14] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5736–5745.

[15] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang, "Attributed graph clustering: a deep attentional embedding approach," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2019, pp. 3670–3676.

[16] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proceedings of the International Conference on Learning Representations*, 2014, pp. 1–14.

[17] A. Sarkar, N. Mehta, and P. Rai, "Graph representation learning via ladder gamma variational autoencoders." in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 5604–5611.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2014, pp. 2672–2680.

[19] J. Lin, Y. Li, and G. Yang, "FPGAN: Face de-identification method with generative adversarial networks for social robots," *Neural Networks*, vol. 133, pp. 132–147, 2021.

[20] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the International Conference on Machine Learning*, 2017, pp. 214–223.

[21] J. Ren, Y. Liu, and J. Liu, "EWGAN: Entropy-based Wasserstein GAN for imbalanced learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 10 011–10 012.

[22] G. Peyré, M. Cuturi *et al.*, "Computational optimal transport: With applications to data science," *Foundations and Trends in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[23] Z. Zhang, M. Wang, and A. Nehorai, "Optimal transport in reproducing kernel hilbert spaces: Theory and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 7, pp. 1741–1754, 2020.

[24] Z. Ma, X. Wei, X. Hong, H. Lin, Y. Qiu, and Y. Gong, "Learning to count via unbalanced optimal transport," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2319–2327.

[25] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: an unsupervised and generative approach to clustering," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2017, pp. 1965–1972.

[26] M. Yin, W. Huang, and J. Gao, "Shared generative latent representation learning for multi-view clustering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6688–6695.

[27] F. Taherkhani, A. Dabouei, S. Soleymani, J. Dawson, and N. M. Nasrabadi, "Self-supervised Wasserstein pseudo-labeling for semi-supervised image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 267–12 277.

[28] X. Yang, J. Yan, Y. Cheng, and Y. Zhang, "Learning deep generative clustering via mutual information maximization," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[29] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2016, pp. 271–279.

[30] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," in *Proceedings of the International Conference on Learning Representations*, 2018, pp. 1–13.

[31] V. Borghuis, L. Angioloni, L. Brusci, and P. Frasconi, "Pattern-based music generation with Wasserstein autoencoders and prcdescriptions," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2020, pp. 5225–5227.

[32] W. Fan, N. Bouguila, and D. Ziou, "Unsupervised hybrid feature extraction selection for high-dimensional non-Gaussian data clustering with variational inference," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1670–1685, 2012.

[33] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the l2-graph for robust subspace learning and subspace clustering," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 1053–1066, 2016.

[34] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao, "Late fusion incomplete multi-view clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 10, pp. 2410–2423, 2018.

[35] D. Huang, C.-D. Wang, and J.-H. Lai, "Locally weighted ensemble clustering," *IEEE Transactions on Cybernetics*, vol. 48, no. 5, pp. 1460–1473, 2017.

[36] Y. Chen, S. Wang, X. Xiao, Y. Liu, Z. Hua, and Y. Zhou, "Self-paced enhanced low-rank tensor kernelized multi-view subspace clustering," *IEEE Transactions on Multimedia*, 2021.

[37] V. M. Patel, H. Van Nguyen, and R. Vidal, "Latent space sparse subspace clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 225–232.

[38] X. Peng, S. Xiao, J. Feng, W.-Y. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior." in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2016, pp. 1925–1931.

[39] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2017, pp. 1753–1759.

[40] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, and J. T. Zhou, "Deep clustering with sample-assignment invariance prior," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4857 – 4868, 2020.

[41] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2017, pp. 5769–5779.

[42] Z. Zhong and J. Li, "Generative adversarial networks and probabilistic graph models for hyperspectral image classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018, pp. 8191–8193.

[43] S. Suh, H. Lee, P. Lukowicz, and Y. O. Lee, "CEGAN: Classification enhancement generative adversarial networks for unraveling data imbalance problems," *Neural Networks*, vol. 133, pp. 69–86, 2021.

[44] Z. Yang, A. Sarkar, and S. Cooper, "Game level clustering and generation using Gaussian mixture VAEs," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 16, no. 1, 2020, pp. 137–143.

[45] N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with Gaussian mixture variational autoencoders," in *Proceedings of the International Conference on Learning Representations*, 2017, pp. 1–12.

[46] P. Zhou, Y. Hou, and J. Feng, "Deep adversarial subspace clustering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1596–1604.

[47] K. Ghasedi, X. Wang, C. Deng, and H. Huang, "Balanced self-paced learning for generative adversarial clustering network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4391–4400.

[48] J. Wu, Z. Huang, J. Thoma, D. Acharya, and L. Van Gool, "Wasserstein divergence for GANs," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 653–668.

[49] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proceedings of the International Conference on Learning Representations*, 2018.

[50] L. Yang, W. Fan, and N. Bouguila, "Clustering analysis via deep generative models with mixture models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 1, pp. 340 – 350, 2022.

[51] L. Mi, W. Zhang, X. Gu, and Y. Wang, "Variational Wasserstein clustering," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 322–337.

[52] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *Journal of Machine Learning Research*, vol. 5, no. Apr, pp. 361–397, 2004.

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

[54] J. Cai, S. Wang, C. Xu, and W. Guo, "Unsupervised deep clustering via contractive feature representation and focal loss," *Pattern Recognition*, vol. 123, p. 108386, 2022.

[55] J. Huang, Q. Dong, S. Gong, and X. Zhu, "Unsupervised deep learning via affinity diffusion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 029–11 036.

[56] J. Chang, G. Meng, L. Wang, S. Xiang, and C. Pan, "Deep self-evolution clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 809–823, 2020.

[57] C. Niu, J. Zhang, G. Wang, and J. Liang, "Gatcluster: Self-supervised gaussian-attention network for image clustering," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 735–751.

[58] J. Huang, S. Gong, and X. Zhu, "Deep semantic clustering by partition confidence maximisation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8849–8858.

[59] T. W. Tsai, C. Li, and J. Zhu, "Mice: Mixture of contrastive experts for unsupervised image clustering," in *Proceedings of the International Conference on Learning Representations*, 2021.

[60] C. Xu, R. Lin, J. Cai, and S. Wang, "Deep image clustering by fusing contrastive learning and neighbor relation mining," *Knowledge-Based Systems*, vol. 238, p. 107967, 2022.

[61] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[62] L. Van Der Maaten, "Learning a parametric embedding by preserving local structure," in *Artificial Intelligence and Statistics*, 2009, pp. 384–391.

[63] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

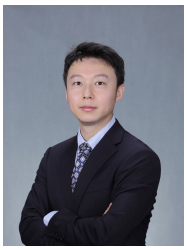
[64] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 271–21 284, 2020.

[65] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

Jinyu Cai received his B.S. degree in Computer Science and Technology from Fuzhou University, Fuzhou, China, in 2018. From 2021-2022, he was a visiting student at the School of Data Science, The Chinese University of Hong Kong, Shenzhen, and Shenzhen Research Institute of Big Data, Shenzhen, China. He received his Ph.D. degree in Computer Science and Technology, from the College of Computer and Data Science, Fuzhou University, in 2023. Currently, he is a Postdoc Research Fellow at the Institute of Data Science, National University of Singapore. His research interests include machine learning, computer vision, and pattern recognition. He published more than 10 papers in international conferences and journals, e.g., NeurIPS, CVPR, and PR. He also served as the reviewer for several top journals/conferences, e.g., IEEE TPAMI/TKDE/TNNLS, NeurIPS, ICLR, ICML, IJCAI, CVPR, ICCV, and ECCV.

Yunhe Zhang received her B.S. degree and M.S. degree in Computer Science and Technology from Fuzhou University, Fuzhou, China, in 2019 and 2022, respectively. Currently, she is a research assistant at the School of Data Science, The Chinese University of Hong Kong, Shenzhen, and Shenzhen Research Institute of Big Data, Shenzhen, China. Her main research interests include machine learning, pattern recognition, and deep learning.

Shiping Wang received his Ph.D. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China in 2014. He worked as a research fellow in Nanyang Technological University from August 2015 to August 2016. He is currently a Professor with the College of Computer and Data Science, Fuzhou University, Fuzhou, China. His research interests include machine learning, computer vision and granular computing.



Jicong Fan (Senior Member, IEEE) received the Ph.D. degree in Electronic Engineering from City University of Hong Kong, Hong Kong, SAR, in 2018. From 2018 to 2020, he was a Post-Doctoral Associate at the School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA. He is currently an Assistant Professor at the School of Data Science, The Chinese University of Hong Kong, Shenzhen, China, and the Shenzhen Research Institute of Big Data, Shenzhen. His research interests include Artificial Intelligence and Machine Learning, particularly focusing on matrix and tensor methods, data clustering, graph learning, and anomaly detection.

Wenzhong Guo received the Ph.D. degree in communication and information system from Fuzhou University in 2010. He is currently a Full Professor with the College of Mathematics and Computer Science, Fuzhou University. His research interests include cloud computing, mobile computing, and evolutionary computation. Currently, he leads the Network Computing and Intelligent Information Processing Laboratory, which is a Key Laboratory of Fujian Province, China.