DLME: Deep Local-flatness Manifold Embedding

Zelin Zang[1,2]*, Siyuan Li[1,2]*, Di Wu[1,2], Ge Wang[1,2], Kai Wang[3], Lei Shang[3], Baigui Sun[3], Hao Li[3], and Stan Z. Li[1,2,✉]

[1] Zhejiang University, Hangzhou, 310000, China
{zangzelin,stan.zq.li}@westlake.edu.cn
[2] Westlake University, AI Lab, School of Engineering, Hangzhou, 310000, China
[3] Alibaba Group, Hangzhou, China
* Equal contribution, ✉ Corresponding author

Abstract. Manifold learning (ML) aims to seek low-dimensional embedding from high-dimensional data. The problem is challenging on real-world datasets, especially with under-sampling data, and we find that previous methods perform poorly in this case. Generally, ML methods first transform input data into a low-dimensional embedding space to maintain the data's geometric structure and subsequently perform downstream tasks therein. The poor local connectivity of under-sampling data in the former step and inappropriate optimization objectives in the latter step leads to two problems: *structural distortion* and *underconstrained embedding*. This paper proposes a novel ML framework named Deep Local-flatness Manifold Embedding (DLME) to solve these problems. The proposed DLME constructs semantic manifolds by data augmentation and overcomes the structural distortion problem using a smoothness constrained based on a *local flatness* assumption about the manifold. To overcome the underconstrained embedding problem, we design a loss and theoretically demonstrate that it leads to a more suitable embedding based on the local flatness. Experiments on three types of datasets (toy, biological, and image) for various downstream tasks (classification, clustering, and visualization) show that our proposed DLME outperforms state-of-the-art ML and contrastive learning methods.

1 Introduction

The intrinsic dimension of high-dimensional data is usually much lower and how to effectively learn a low-dimensional representation is a fundamental problem in traditional machine learning [32], data mining [1], and pattern recognition [6]. Manifold learning (ML), based on solid theoretical foundations and assumptions, discusses manifold representation problems under unsupervised conditions and has a far-reaching impact. However, practical applications of the manifold learning method are limited in real-world scenarios, and we attribute the reasons to the following two reasons. **(D1) Underconstrained manifold embedding.** ML methods focus on local relationships, while it is prone to distorted embeddings that affect the performance of downstream tasks (in Fig. 1 (D2) and Fig. 6). Paper [18,19] suggests even the most advanced ML methods lose performance on downstream tasks due to inadequate constraints on the latent space.

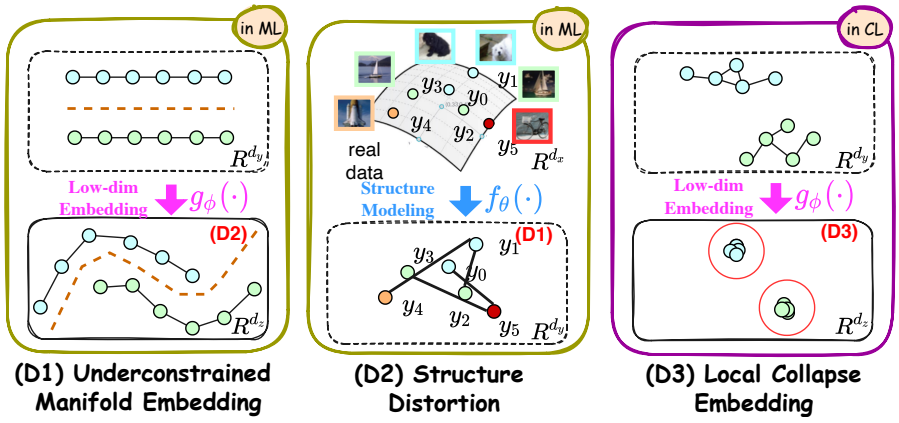

Fig. 1. Problems in ML and CL. **(D1)** Local field of view & first-order (similarity/dissimilarity) constraints → underconstrained manifold embedding. **(D2)** complexity of real-world data (ultra-high dimensionality or non well-sampling) → broke the local connectivity of manifold → structure distortions. **(D3)** unsmoothed losses function → local collapse embedding.

The reason is attributed to the limitations of traditional ML methods based on similarity/dissimilarity loss function design. **(D2) Structural distortion.** ML methods focus on handcraft or easy datasets and are not satisfactory in handling real-world datasets. Most of these approaches use the locally connected graphs constructed in the input space to define structure-preserving unsupervised learning loss functions [28,27]. These methods introduce a stringent assumption (local connectivity assumption (LCA)) which suggests the metric of input data well describes the data's neighbor relationship. However, LCA requires the data to be densely sampled and too ideal in the real world, e.g., pictures of two dogs are not necessarily similar in terms of pixel metric (in Fig. 1 (D2)).

Meanwhile, Contrastive Learning (CL) is enthusiastically discussed in the image and NLP fields [3,14,11]. These methods have shown excellent performance by introducing prior knowledge of the data with the help of data augmentation. However, we have encountered significant difficulties applying such techniques to the ML domain. The above methods require a large amount of data for pretraining [39,42], so it is not easy to achieve good results in areas where data is expensive (e.g., biology, medicine, etc.). We consider that the core issues can be summarized as **(D3) Local collapse embedding.** The unsmoothed loss of the CL leads to the model that is prone to local collapse and requires a large diversity of data to learn valid knowledge (in Fig. 2 (D3)).

We want to propose a novel deep ML model to constrain the latent space better and solve the structural distortion problem with the help of CL. At the same time, we hope the proposed method avoids the local collapse phenomenon in the CL. The process of ML perspective includes *structural modeling sub-processes* and *low-dimensional embedding sub-processes*. The structural modeling

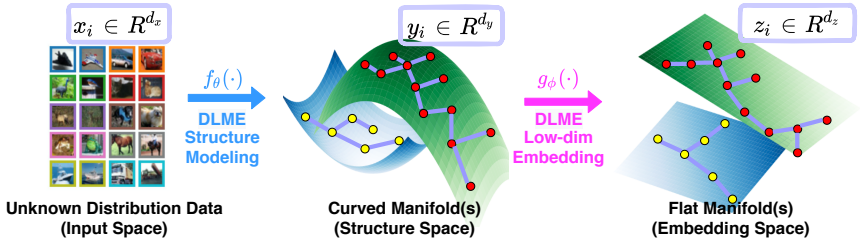

Fig. 2. DLME includes the structure modeling network $f_\theta(\cdot)$ and low-dim embedding network $g_\phi(\cdot)$. The $f_\theta(\cdot)$ maps the input data into structure space to describe the data relationship. The $g_\phi(\cdot)$ maps the curled manifold into the flat embedding space to improve the discriminative performance and friendliness to downstream tasks. $f_\theta(\cdot)$ and $g_\phi(\cdot)$ are compatible with any neural network.

sub-process obtains the graph structures of data manifolds by measuring the relationship of each sample pair which serves as the guidance for low-dimensional embedding. The low-dimensional embedding process maps the provided graph structures into the embedding space.

We propose a novel deep ML framework named deep local-flatness manifold embedding (DLME) to solve both problems by merging the advantages of ML and CL. Firstly, a novel local flatness assumption (LFA) is proposed to obtain a reasonable latent space by adding a second-order manifold curvature constraint (for **D1**), thus improving the performance on downstream tasks. Secondly, a new neural network framework is designed to accommodate data augmentation and emhance the net work trainging (for **D2**). DLME framework accomplishes the two sub-processes with two networks ($f_\theta(\cdot)$ and $g_\phi(\cdot)$) optimized by the proposed DLME loss between the latent space of $f_\theta(\cdot)$ and $g_\phi(\cdot)$ in an end-to-end manner (framework is in Fig. 2). Furthermore, an LFA-based smoother loss function is designed to accommodate data augmentation. It is based on a long-tailed t-distribution and guides the network learning through the two latent spaces (for **D3**). Finally, we further illustrate mathematically: (1) the differences between DLME loss and conventional CL loss and (2) why DLME loss can obtain a locally flat embedding.

In short, DLME makes the following contributions: (1) DLME provides a novel deep ML framework that utilizes neural networks instead of distance metrics in data space to better model structural relationships, thus overcoming structural distortions. (2) DLME put forward the concept of local flatness and theoretically discusses that the DLME loss can enhance the flatness of manifolds and obtain transfer abilities to downstream tasks. (3) The effectiveness of DLME is demonstrated on three downstream tasks with three types of datasets (toy, biological, and image). Experiment results show that DLME outperforms current state-of-the-art ML and CL methods.

2 Related Works

In **manifold learning**, MDS [20], ISOMAP [38], and LLE [33] model the structure of data manifolds based on local or global distances (dissimilarity) and linearly project to the low-dimensional space. SNE [16], t-SNE [27] and UMAP [28] use normal distribution to define local similarities and apply the Gaussian or t-distribution kernel to transform the distance into the pair-wise similarity for *structural modeling*. They perform the manifold embedding by preserving the local geometric structures explored from the input data.

In **deep manifold learning**, Parametric t-SNE (P-TSNE) [26] and Parametric UMAP [35] learn more complex manifolds by non-linear neural networks and can transfer to unseen data. However, they inherit the original *structural modeling* strategies in t-SNE and UMAP. Topological autoencoder (TAE), Geometry Regularized AutoEncoders (GRAE) [7], and ivis [37] abandon the accurate modeling of input data and directly achieve *low-dimensional embedding* using distances or contrast training.

In **self-supervised contrastive learning**, contrastive-based methods [41,3,14,11] which learns instance-level discriminative representations by contrasting positive and negative views have largely reduced the performance gap between supervised models on various downstream tasks. Deep clustering methods is another popular form of self-supervised pertraining.

3 Methods

3.1 Problem Description and Local Flatness Assumptions

According to Nash embedding theorems [31], we mainly discuss manifolds represented in Euclidean coordinates and provide the definitions of manifold learning (ML) and deep manifold learning (DML) in practical scenarios.

Definition 1 (Manifold Learning, ML). Let $\mathcal{M}$ be a d-dimensional embedding in Euclidean space $\mathbb{R}^d$ and $f : \mathcal{M} \to \mathbb{R}^D$ be a diffeomorphic embedding map, for $D > d$, the purpose of manifold learning is to find $\{z_i\}_{i=1}^N, z_i \in \mathcal{M}$ from the sufficient sampled(observed) data $X = \{x_i\}_{i=1}^N, x_i \in \mathbb{R}^D$.

Based on Definition 1, the DML aims finding the embedding $\{z_i\}_{i=1}^N, z_i \in \mathcal{M}$ by mapping $g_\theta : \mathbb{R}^D \to \mathbb{R}^d$ with the neural network parameters θ. Each ML method designs a loss function based on the specific manifold assumption to map the observed data $\{x_i\}$ back to the intrinsic manifold $\{z_i\}$. For example, LLE [34] assumes that the local manifold is linear, and UMAP [28] assumes that the local manifold is uniform. We propose a novel assumption, considering the nature of the manifold is local flatness.

Assumptions 1 (Local Flatness Assumption, LFA). Let $\mathcal{M}$ be a manifold, and $\{x_i\}$ be a set of observations in the manifold. We expect each data point and its neighbors to lie on or close to a local flatness patch. The mean curvature $\overline{K}_{\mathcal{M}}$ is introduced to quantify the flatness of high-dimensional manifolds according

to the Gauss-Bonnet theorem [36],

$$\begin{aligned} \overline{K}_{\mathcal{M}} &= \sum_{x_i \in X} k(x_i) \\ k(x_i) &= 2\pi\chi(\mathcal{M}) - \theta(x_{|\mathrm{H}_1(x_i)|}, x_i, x_0) - \sum_{j \in \{0,1,\cdots,|\mathrm{H}_1(x_i)|-1\}} \theta(x_j, x_i, x_{j+1}), \end{aligned} \tag{1}$$

where $\chi(\mathcal{M})$ is Euler Characteristic [13]. The $\mathrm{H}_1(x_i)$ is the hop-1 neighbor of x_i, and $\theta(x_i, x_j, x_k)$ is the angle of three point x_i x_j, x_k.

From first-order constraint to second-order curvature constraint. ML methods (e.g., LLE and UMAP) design distance-preserving or similarity-preserving objective functions, hoping to guarantee the first-order relationship of the data. However, first-order relation preservation is not a tight enough constraint if the local structure of the manifold is simple, thus leading to underconstrained manifold embedding in ML. We introduce a second-order (curvature) constraint to solve the distortion problem. Due to the expensive complexity of second-order losses, we directly minimize the manifold's curvature by a mundane flatness assumption.

The Empirical benefits of LFA. Similar to most ML and CL methods, LFA is an assumption of latent space, which is beneficial for downstream tasks. In the case of the single-manifold, the assumption of 'Local Flatness' reduces curling in the unsuitable embedding space (see Fig. 6), thus avoiding distortion during embedding. In the case of the multi-manifolds, assuming 'Local Flatness' can simplify the discriminative relations of multi-manifolds. Therefore, the proposed assumption can avoid representation collapse. Meanwhile, it also reduces the possibility of different manifolds overlapping so that the downstream can be accomplished by a simple linear model easily.

3.2 DLME Framework

As shown in in Fig. 3, the DLME framework contains two neural networks (f_θ and g_ϕ) and a DLME loss function L_D. The network f_θ achieves *structural modeling* in its structure space $\mathbb{R}^{d_y}$, and the network g_ϕ learns *low-dimensional embedding* in the embedding space $\mathbb{R}^{d_z}$. The DLME loss is calculated based on the A_{ij} and the pairwise similarity in spaces $\mathbb{R}^{d_y}$ and $\mathbb{R}^{d_z}$ used to train two neural networks from scratch. The A_{ij} indicate the homologous relationships, if x_i and x_j are augmentations of the same original data, then $A_{ij} = 1$ else $A_{ij} = 0$.

Data augmentation for solving structural distortion. ML methods have difficulty efficiently identifying neighboring nodes, causing structural distortions, when dealing with complex and not well-sampled data. DLME solves this problem with a priori knowledge provided by data augmentation. Data augmentation schemes have been widely used in self-supervised contrastive learning (CL) to solve problems in CV and NLP. From the ML perspective, data augmentation is a technique to make new observations in the intrinsic manifold based on prior knowledge. Since data augmentation changes the semantics of the original data as little as possible, it generates specific neighborhood data

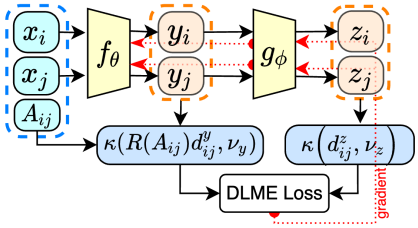

Fig. 3. The framework of DLME. (x_i, x_j) is a pair of input sample, and the neighbor relationship A_{ij} indicates whether x_i and x_j are homologous pairs. The red dashed line marks the direction of the gradient back-propagation.

for each isolated data when the local connectivity of ML data is broken. DLME trains a neural network $f_\theta(\cdot)$. The $f_\theta(\cdot)$ is guided by data augmentation and loss functions to map the data into a latent space that better guarantees local connectivity.

Data augmentation is designed based on domain knowledge. For example, in CV datasets, operations such as color jittering [43], random cropping [4], applying Gaussian blur [9], Mixup [24,21,25] are proven useful. In biology and some easy datasets, linear combinations $\tau_{lc}(\cdot)$ in k-nearest neighbor data is a simple and effective way. The linear combinations is

$$\tau_{lc}(x) = rx + (1-r)x^n, x^n \sim \mathrm{KNN}(x), \tag{2}$$

where x^n is sampled from the neighborhood set of data x, and $r \in [0,1]$ is a combination parameter. For special domain data, the prior knowledge in the domain can be used to establish data augmentation.

The forward propagation of DLME is,

$$\begin{aligned} y_i &= f_\theta(x_i), y_i \in \mathbb{R}^{d_y}, x_i \sim \tau(x), x_j \sim \tau(x), \\ z_i &= g_\phi(y_i), z_i \in \mathbb{R}^{d_z}, d_z < d_y, \end{aligned} \tag{3}$$

where x_i and x_j sampled form different random augmentation of raw data x, the d_y and d_z are the dimension number of $\mathbb{R}^{d_y}$ and $\mathbb{R}^{d_z}$.

The loss function of DLME is,

$$L_{\mathrm{D}} = E_{x_i,x_j}\left[\mathcal{D}\left(\kappa\left(R(A_{ij})d^y_{ij}, \nu_y\right), \kappa\left(d^z_{ij}, \nu_z\right)\right)\right], \tag{4}$$

where $d^y_{ij} = d(y_i, y_j)$, $d^z_{ij} = d(z_i, z_j)$ and d^y_{ij}, d^z_{ij} are the distance metrics of data node i and j in spaces $\mathbb{R}^{d_y}$ and $\mathbb{R}^{d_z}$. The two-way divergence [22] $\mathcal{D}(p,q)$ is introduced to measure the dis-similarity between two latent spaces,

$$\mathcal{D}(p,q) = p\log q + (1-p)\log(1-q), \tag{5}$$

where $p \in [0,1]$. Notice that $\mathcal{D}(p,q)$ is a continuous version of the cross-entropy loss. The two-way divergence is used to guide the pairwise similarity of two latent

spaces to fit each other. The effect of the loss function on the two networks will be discussed in Sec 3.3 and Sec 3.4.

The structure space requires a larger dimensionality to accurately measure data relationships, while the embedding space requires sufficient compression of the output dimension. Thus the t-distribution kernel function is used to calculate the pairwise similarity. The different degrees of freedom ν_y and ν_z in different spaces are essential to enhance the flatness of embedding space (in Sec 3.4).

$$\kappa\left(d,\nu\right)=\frac{\mathrm{Gam}\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\mathrm{Gam}\left(\frac{\nu}{2}\right)}\left(1+\frac{d^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \tag{6}$$

where $\mathrm{Gam}(\cdot)$ is the Gamma function, and the degrees of freedom ν controls the shape of the kernel function.

DLME design $R(A_{ij})$ to integrate the neighborhood information in A_{ij}.

$$R(A_{ij}) \ =1+(\alpha-1)A_{ij}=\begin{cases}\alpha & \text{if } A_{ij}=1\\ 1 & \text{otherwise}\end{cases}, \tag{7}$$

where $\alpha\in[0,1]$ is a hyperparameters. If x_i is the neighbor of x_j, the distance in *structure space* will be reduced by α, and the similarity of x_i and x_j will increase.

3.3 Against Local Collapse by a Smoother CL Framework

The CL loss in self-supervised contrastive learning (CL) frameworks is

$$L_{\mathrm{C}}=-\mathbb{E}_{x_i,x_j}\left[A_{ij}\log\kappa(d_{ij}^z)+(1-A_{ij})\log(1-\kappa(d_{ij}^z))\right], \tag{8}$$

where similarity kernel function $\kappa(d_{ij}^z)$ is defined in Eq.(6). The CL is not smooth and can be analogous to bang-bang control [8] in control systems. Because the learning target of the point pair will switch between $\log\kappa(d_{ij}^z)$ and $\log(1-\kappa(d_{ij}^z))$ with the change of A_{ij}.

The proposed framework is a smoother CL framework because DLME compromises the learning process and avoids sharp conflicts in gradients. To compare the difference between the DLME loss and the CL loss, we assume that $g_\phi(\cdot)$ satisfies K-Lipschitz continuity [10], then

$$d_{ij}^z=k^*d_{ij}^y, k^*\in[1/K,K], \tag{9}$$

where k^* is a Lipschitz constant. The difference of CL loss and DLME loss is

$$|L_{\mathrm{D}}-L_{\mathrm{c}}|=\mathbb{E}_{x_j,x_j}\left[A_{ij}-\kappa\left((1+(\alpha-1)A_{ij})k^*d_{ij}^z\right)\log(\frac{1}{\kappa(d_{ij}^z)}-1)\right], \tag{10}$$

The detailed derivation is provided in Appendix B. If i and j are neighbors, $A_{ij}=1$, when $\alpha\to0$, then $\alpha k^*d_{ij}^z\to0$ and then $1-\kappa(\alpha k^*d_{ij}^z)\to0$, finally we have the $|L_{\mathrm{D}}-L_{\mathrm{c}}|\to0$. When $\alpha\to0$, the two losses have the same effect on

the samples within each neighbor system. When $\alpha > 0$, the optimal solution of L_{D} retain a remainder about the embedding structure d_{ij}^{z} (in Appendix) which indicates that the DLME loss does not maximize the similarity of the neighborhood as the CL loss, but depends on the current embedding structure. Eq.(10) indicates that the DLME loss is smoother and can preserve the data structure in the embedding space. When $\alpha > 0$, the DLME loss is a smooth version of the CL loss, which causes a minor collapse of local structures.

Generally, $f_\theta(\cdot)$ explores the structure of the prior manifolds defined by the given data augmentations smoothly, which can model the manifold structure more accurately than previous ML and DML methods.

3.4 Why DLME Leads to Local Flatness

This section discusses why the DLME loss optimizes the local curvature to be flatter. Network $f_\theta(\cdot)$ maps the data in the input space to the structure space for accurate structural modeling, although it is not guaranteed to obtain locally flat manifolds. Curling can cause overlap and deformation of the manifold, which can cause degradation of downstream task performance. To improve the performance in downstream tasks, we need to obtain an embedding space as flat as possible. The simplest linear methods can perform the discriminative tasks (classification, clustering, visualization).

DLME loss can enforce the flatness of the manifold in the embedded space. Similar to t-SNE, we use the kernel function of the long-tailed t-distribution to transform the distance metric into similarity. Further, we apply different 'degrees of freedom' parameters ν in the two latent spaces. The differences in the degree of freedom ν form two different kernel functions $\kappa(d, \nu_y)$ and $\kappa(d, \nu_z)$, and the difference of kernel functions will make the manifold in the embedding space flatter during the training process.

As described by Eq. (1), we use the local curvature description of the discrete surface to represent the flatness of the manifold. Next, we theoretically discuss why DLME's loss can minimize the local curvature. We use the Push-pull property to describe the action of DLME loss on the embedding space.

Lemma 1 (Push-pull property). let $\nu^y > \nu^z$ and let $d^{z+} = \kappa^{-1}(\kappa(d, \nu^y), \nu^z)$ be the solution of minimizing L_{D}. Then exists d_p so that $(d^y - d_p)(d^{z+} - d^y) > 0$.

The proof of lemma 1 is detailed in Appendix . Lemma 1 describes the push-pull property of the DLME loss between sample pairs in the embedding space. L_{D} decreases the distance between sample pairs within the threshold d_p (as similar pairs) and increases the distance between sample pairs beyond d_p (as dis-similar pairs), which shows pushing and pulling effects between two kinds of sample pairs. Next, we prove that the DLME loss minimizes the average local curvature of the embedding based on the push-pull property.

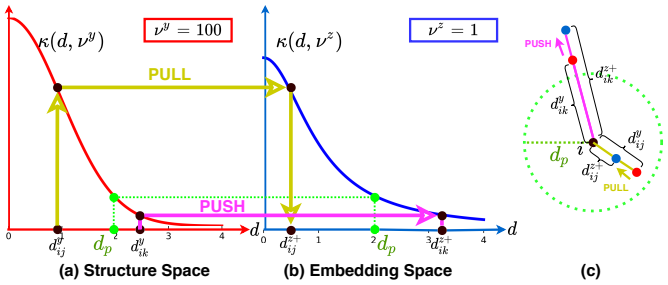

Fig. 4. Push-pull property: if $d^y < d_p$ then $d^{z+} < d^y$ (in yellow), if $d^y > d_p$ then $d^{z+} > d^y$ (in pink).

Lemma 2. Assume $f_\theta(\cdot)$ satisfies HOP1-2 order preserving: $\max(\{d^y_{ij}\}_{j\in \mathrm{H}_1(x_i)}) < \min(\{d^z_{ij}\}_{j\in \mathrm{H}_2(x_i)})$ Then $\overline{K}^{z+}_{\mathcal{M}} < \overline{K}^{y}_{\mathcal{M}}$ where $\overline{K}^{y}_{\mathcal{M}}$ is the mean curvature in the structure space, and $\overline{K}^{z+}_{\mathcal{M}}$ is the mean curvature optimization results of L_{D} in the embedding space.

Lemma 2 indicates that the DLME loss encourages the flatness of the embedding space by decreasing the local average curvature. As Fig. 6, Lemma 2 describes that the optimization result of DLME loss is to flatten the manifold of the embedded space, which means that we can represent the data in a latent space as linear as possible. **DLME's pseudo-code** is shown in Algorithm 1.

Algorithm 1 The DLME algorithm

Input: Data: $\mathcal{X} = \{x_i\}_{i=1}^{|\mathcal{X}|}$, Learning rate: η, Epochs: E, Batch size: B, α, ν^y, ν^z, Network: f_θ, g_ϕ, **Output**: Graph Embedding: $\{e_i\}_{i=1}^{|\mathcal{X}|}$.

while $i = 0; i < E; i{+}{+}$ **do**
  $X^+ \leftarrow X \cup \tau(X)$. # Data augmentation
  while $b = 0; b < [|\mathcal{X}|/B]; b{+}{+}$ **do**
    $\{x_{a,1}, x_{a,2} \sim X^+\}_{a\in\mathcal{B}}$, $\mathcal{B} = \{1, \cdots, B\}$; # Sampling
    $\{y_{a,0}, y_{a,1} \leftarrow f_\theta(x_{a,0}), f_\theta(x_{a,1})\}_{a\in\mathcal{B}}$; # Map to $\mathbb{R}^{d_y}$
    $\{z_{a,0}, z_{a,1} \leftarrow g_\phi(y_{a,0}), g_\phi(y_{a,1})\}_{a\in\mathcal{B}}$; # Map to $\mathbb{R}^{d_z}$
    $\{d^y_{a,ij} \leftarrow d(y_{a,0}, y_{a,1})\}_{a\in\mathcal{B}}$; $\{d^z_{a,ij} \leftarrow d(z_{a,0}, z_{a,1})\}_{a\in\mathcal{B}}$; #Cal. dist in $\mathbb{R}^{d_y}$ & $\mathbb{R}^{d_z}$
    $\{S^y_a \leftarrow \kappa(R(B_{a,ij})d^y_{a,ij}, \nu_y)\}_{a\in\mathcal{B}}$; $\{S^z_a \leftarrow \kappa(d^z_{a,ij}, \nu_z)\}_{a\in\mathcal{B}}$; #Cal. sim in $\mathbb{R}^{d_y}$ & $\mathbb{R}^{d_z}$
    $\mathcal{L}_{\mathrm{D}} \leftarrow E(\{D(S^y_a, S^z_a)\}_{a\in\mathcal{B}})$ by Eq. (4); # Cal. loss function
    $\theta \leftarrow \theta - \eta\frac{\partial \mathcal{L}_{\mathrm{D}}}{\partial \theta}$, $\phi \leftarrow \phi - \eta\frac{\partial \mathcal{L}_{\mathrm{D}}}{\partial \phi}$; # Update parameters
  end while
end while
$\{z_i \leftarrow f_\theta(g_\phi(x_i))\}_{i\in\{1,2,\cdots,\mathcal{X}\}}$; # Cal. the embedding result

4 Experiments

In this section, we evaluate the effectiveness of the proposed DLME on four downstream tasks (classification/linear test, clustering, visualization) and analyze each proposed component with the following questions.
(**Q**1) How to intuitively understand structural distortions? (**Q**2) Does DLME overcome structural distortions? (**Q**3) How to intuitively understand underconstrained manifold embedding? (**Q**4) Does DLME overcome underconstrained manifold embedding and obtain locally flat embeddings? (**Q**5) How much does DLME improve the performance of downstream tasks on ML and CL datasets? (**Q**6) Can smoother losses bring better performance to CL?

4.1 Visualization of Structural Distortions (Q1,Q2)

Experimental setups. This section illustrates structural distortions on image datasets and experimentally demonstrates that DLME can overcome structural distortions by introducing prior knowledge of data augmentation. In this experiment, all the data are mapped to a 2-D latent space to facilitate the visualization. All compared ML methods (t-SNE, PUMAP, ivis, and PHA) will fail in the CIFAR dataset; we only show the results of UMAP.

Structural Distortions. The ML approach uses distance metrics from observations to model the structure. The complexity of the data (data with dimensionality and not-well sampling) leads to a failure of the distance metric, confusing the semantic nearest neighbors and subsequently destroying local connectivity, ultimately creating structural distortions in the ML process. DLME constructs a smoother CL framework with the help of data augmentation. The proposed framework obtains richer prior knowledge with data augmentation. It maps the data into the latent space for structural modeling with neural networks, which can achieve more accurate modeling and thus overcome structural distortion.

4.2 Visualization of Underconstrained Manifold Embedding (Q3,Q4)

This section illustrates underconstrained manifold embedding with toy datasets and experimentally demonstrates the DLME potential to solve this problem by constraining the local curvature. **Experimental setups.** The experiments include two 3D toy datasets, TwainSwissRoll and StarFruit. The TwainSwissRoll dataset has two tangled but disjoint SwissRolls. The StarFruit dataset has a locally curved surface. The input data and outputs of compared methods are shown in Fig. 6. The details of the datasets and outputs are shown in Appendix.

Underconstrained manifold embedding. Fig. 6 shows that the compared ML methods produce bends that should not exist. We attribute these bends to inadequate constraints on the loss function. These bends affect downstream tasks such as classification and clustering. In addition, these bends may cause more significant damage when the data situation is more complex.

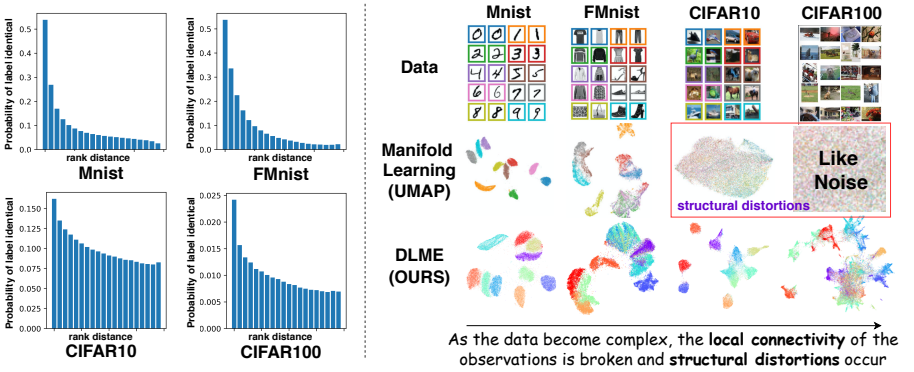

Fig. 5. (left) The Bar plot of probabilities of identical label v.s. rank distance. A higher left end of the bar plot indicates a higher probability of the same label for the nearest neighbor sample, implying that local connectivity is guaranteed. (right) The results of ML methods (UMAP) for four image datasets. For complex data, local connectivity cannot be guaranteed, leading to the embedding failure of the ML method. The proposed DLME method has better embedding results on the more complex CIFAR dataset.

Benefits of local flat embedding. Since a flat localization is assumed, DLME tries to obtain a locally flat embedding. The statistics of the mean curvature show that DLME can receive a flatter local manifold. We consider that a flat embedding possesses practical benefits. A flatter local manifold can achieve better performance in downstream tasks and is more suitable for interpretable analysis of deep models. For example, it is easy to distinguish two sub-manifolds of TwainSwissRoll using a linear classifier, and it is easy to perform regression tasks on StarFruit.

4.3 Comparison on Traditional ML Datasets (Q5, Q6)

Experimental setups: The compared methods include two ML methods (UMAP [28], t-SNE (tSNE) [15]) and three deep ML methods (PHATE (PHA) [29], ivis [37] and parametric UMAP (PUM) [35].) The experiments are on six image datasets (Digits, Coil20, Coil100, Mnist, EMnist, and KMnist) and six biological datasets (Colon, Activity (Acti), MCA, Gast, SAMUSIK (SAMU), and HCL). For a fair comparison, all the compared methods map the input data into a 2D space and then evaluated by 10-fold cross-validation. The MLP architecture of f_θ is [-1,500,300,80], where -1 is the dimension of input data. The MLP architecture of g_ϕ is [80,500,80,2]. The comparison results of linear SVM and k-means are shown in Table 1. Details of datasets, baseline methods, and evaluation metrics are in Appendix.

Analyse: DLME has advantages over all 12 datasets, and DLME is 5% higher than other methods in 14 items (24 items in total). We observe that the proposed DLME has advantages in classification and clustering metrics. We sum-

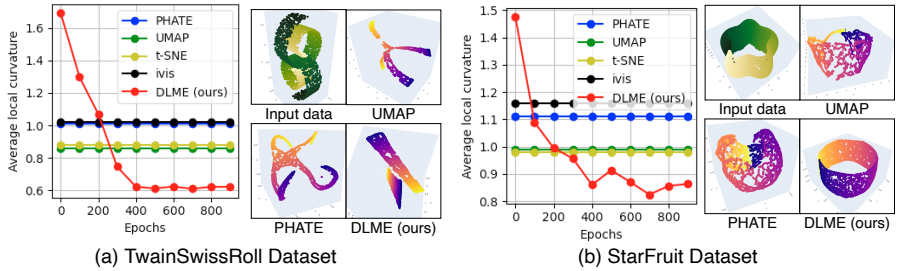

Fig. 6. Average local curvature and scatter plot on TwainSwissRoll and StarFruit dataset. The two examples indicate that traditional ML produces distorted embeddings, which affect the performance of downstream tasks. In contrast, DLME can get as flat embeddings as possible by optimizing the local curvature.

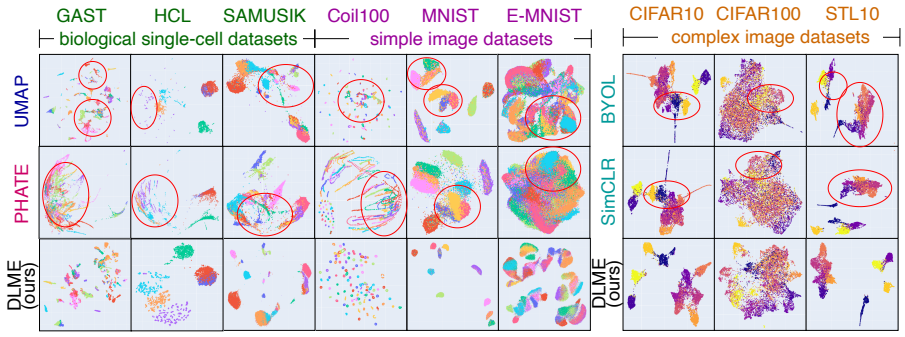

Fig. 7. Visualization results of biological datasets, simple image datasets and complex image datasets. The red circle indicates the clusters confused by the baseline method.

marize the reasons why DLME has performance advantages as follows. (1) Compared with ML methods, DLME overcomes structural distortions to a certain extent to model the structure more accurately. (2) DLME reduces the overlap of different clusters, improving the performance of classification and clustering. (3) The locally flat embeddings learned by DLME are linearly characterized and more suitable for the linear model.

4.4 Comparison on CL Datasets (Q5, Q6)

Experimental setups: Due to structural distortions, ML methods fail in CV Datasets, and our comparison is limited to the CL and deep clustering (DC) domain. The compared methods include CL methods (NPID [41], ODC [44], SimCLR [3], MOCO.v2 [14] and BYOL [11]) and deep clustering methods (DAC [12], DDC [2], DCCM [40], PICA [17], CC [23], and CRLC [5]). The datasets include six image datasets: CIFAR10, CIFAR100, STL10, TinyImageNet, ImageNet-Dog and ImageNet100. The f_θ is ResNet50, and g_ϕ is MLP

Table 1. Performance comparison on 12 datasets. **Bold** denotes the best result and **Underline** denotes 5% higher than others.

	Classification Accuracy (linear SVM)						Clustering Accuracy (K-means)					
	tSNE	UMAP	PUM	ivis	PHA	DLME	tSNE	UMAP	PUM	ivis	PHA	DLME
Digits	0.949	0.960	0.837	0.767	0.928	**0.973**	0.938	0.875	0.763	0.726	0.794	**0.956**
Coil20	0.799	0.834	0.774	0.672	0.828	**0.909**	0.763	0.821	0.722	0.612	0.655	**0.899**
Coil100	0.760	0.756	N/A	0.542	0.653	**0.952**	0.763	0.785	N/A	0.492	0.515	**0.944**
Mnist	0.963	0.966	0.941	0.671	0.796	**0.976**	0.904	0.801	0.772	0.466	0.614	**0.977**
EMnist	0.420	0.588	0.384	0.190	0.416	**0.657**	0.478	0.537	0.363	0.178	0.352	**0.641**
KMnist	0.738	0.656	0.674	0.547	0.607	**0.782**	0.586	0.668	0.706	0.522	0.594	**0.712**
Colon	0.932	0.893	0.918	0.942	0.930	**0.947**	0.862	0.847	0.861	0.922	0.855	**0.924**
Acti	0.861	0.844	0.849	0.831	0.798	**0.921**	0.784	0.639	0.783	0.681	0.679	**0.898**
MCA	0.719	0.675	0.667	0.634	0.552	**0.774**	0.475	0.532	0.464	0.443	0.414	**0.563**
Gast	0.821	0.846	0.706	0.687	0.676	**0.918**	0.534	0.546	0.512	0.427	0.523	**0.598**
SAMU	0.556	0.678	0.599	0.625	0.675	**0.700**	0.335	0.387	0.345	0.328	0.511	**0.572**
HCL	**0.874**	0.863	0.767	0.454	0.393	**0.874**	0.689	0.743	0.619	0.308	0.263	**0.753**

Table 2. The linear-test Performance comparison on image datasets.

Dataset	CIFAR10	CIFAR100	STL10	TinyImageNet	ImageNet100
NPID	0.827	0.571	0.825	0.382	0.721
ODC	0.799	0.521	0.734	0.287	0.645
SimCLR	0.882	0.574	0.869	0.384	0.756
MoCo.v2	0.886	0.614	0.856	0.374	0.780
BYOL	0.881	0.644	0.887	0.388	0.785
DLME	**0.913**	**0.661**	**0.901**	**0.449**	**0.793**
DLME-A1	0.910	0.653	0.881	0.428	0.785
DLME-A2	0.902	0.626	0.879	0.432	0.791
DLME-A3	0.888	0.624	0.873	0.401	0.783

Table 3. The clustering Performance comparison on image datasets.

Dataset	CIFAR10	CIFAR100	STL10	TinyImageNet	ImageNet-Dog
DAC	0.522	0.238	0.470	0.066	0.219
DCCM	0.623	0.327	0.482	0.108	0.321
PICA	0.696	0.337	0.713	0.098	0.352
CC	0.747	0.429	0.850	0.140	0.445
CRLC	0.799	0.425	0.818	0.153	0.461
DLME	**0.822**	**0.441**	**0.883**	**0.182**	**0.483**
DLME-A1	0.792	0.417	0.872	0.145	0.479
DLME-A2	0.783	0.421	0.859	0.133	0.480
DLME-A3	0.779	0.417	0.852	0.134	0.477

with architecture of [2048, 256]. We use the same settings as SimCLR for the linear test and use the same settings as CC[23] for deep clustering. The results are shown in Table 2 and Table 3 and the detailed setup is in Appendix .

Analyse: In all datasets, DLME outperformed the SOTA method by a large margin. And it beat the other techniques by 2% in 6 items (out of 10 items). The reason is the smoother DLME framework avoids problems such as falling into local collapse. Another reason is locally flat embeddings learned by DLME are linearly characterized and more suitable for a linear model.

Ablation Study. We designed ablation experiments to demonstrate the effectiveness of DLME. **Ablation 1 (DLME-A1)**, we detach the L_{D}'s gradient on the model $f_\theta(\cdot)$ and replace it with the CL loss (in Eq.(8)). The model is divided into two separate parts. One obtains embedding with CL, and the other emphasizes the flatness of manifold with similarity loss. **Ablation 2 (DLME-A2)**, based on DLME-A1, We ablate the t-distribution kernel and use a standard distribution kernel in both spaces. **Ablation 3 (DLME-A3)**, finally, we further ablate the structure of the two networks and transform the model into a CL method. The results of ablation experiments are in Table 2 and Table 3. The experimental results show that the three critical operations of DLME can improve the performance in complex manifold embedding tasks.

4.5 Visualization of ML and CV Datasets (Q5, Q6)

DLME is an appropriate method for visualizing high-dimensional data. A typical setup for data visualization using DLME is to embed the data directly into 2D space. As the selected visualization results are shown in Fig. 7, DLME significantly outperforms other methods in terms of visualization results. Because the distortion problem is overcome, the DLME embedding results in a minimum mixture of different clusters with clear boundaries. The detailed results are shown in Appendix.

5 Conclusion

We propose Deep Local-flatness Manifold Embedding (DLME), a novel ML framework to obtain reliable manifold embedding by reducing distortion. In the experiments, by demonstrating the effectiveness of DLME on downstream classification, clustering, and visualization tasks with three types of datasets (toy, biological, and image), our experimental results show that DLME outperforms SOTA ML & CL methods.

Acknowledgement

This work is supported by National Natural Science Foundation of China, named Geometric Deep Learning and Applications in Proteomics-Based Cancer Diagnosis (No. U21A20427). This work is supported by Alibaba Innovative Research (AIR) Programme.

References

1. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the 1998 ACM SIGMOD international conference on Management of data. pp. 94–105 (1998)
2. Chang, J., Guo, Y., Wang, L., Meng, G., Xiang, S., Pan, C.: Deep Discriminative Clustering Analysis. arXiv:1905.01681 [cs, stat] (May 2019), http://arxiv.org/abs/1905.01681, arXiv: 1905.01681
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A Simple Framework for Contrastive Learning of Visual Representations. arXiv:2002.05709 [cs, stat] (Jun 2020), http://arxiv.org/abs/2002.05709, arXiv: 2002.05709
4. Cheng, B., Wu, W., Tao, D., Mei, S., Mao, T., Cheng, J.: Random cropping ensemble neural network for image classification in a robotic arm grasping system. IEEE Transactions on Instrumentation and Measurement **69**(9), 6795–6806 (2020)
5. Do, K., Tran, T., Venkatesh, S.: Clustering by maximizing mutual information across views (2021)
6. Donoho, D.L., et al.: High-dimensional data analysis: The curses and blessings of dimensionality. AMS math challenges lecture **1**(2000), 32 (2000)
7. Duque, A.F., Morin, S., Wolf, G., Moon, K.: Extendable and invertible manifold learning with geometry regularized autoencoders. 2020 IEEE International Conference on Big Data (Big Data) (Dec 2020). https://doi.org/10.1109/bigdata50022.2020.9378049, http://dx.doi.org/10.1109/BigData50022.2020.9378049
8. Flugge-Lotz, I.: Discontinuous automatic control. Princeton University Press (2015)
9. Flusser, J., Farokhi, S., Höschl, C., Suk, T., Zitova, B., Pedone, M.: Recognition of images degraded by gaussian blur. IEEE transactions on Image Processing **25**(2), 790–806 (2015)
10. Gouk, H., Frank, E., Pfahringer, B., Cree, M.: Regularisation of Neural Networks by Enforcing Lipschitz Continuity. arXiv:1804.04368 [cs, stat] (Sep 2018), http://arxiv.org/abs/1804.04368, arXiv: 1804.04368
11. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. arXiv:2006.07733 [cs, stat] (Jun 2020), http://arxiv.org/abs/2006.07733, arXiv: 2006.07733
12. Haeusser, P., Plapp, J., Golkov, V., Aljalbout, E., Cremers, D.: Associative Deep Clustering: Training a Classification Network with No Labels. In: Brox, T., Bruhn, A., Fritz, M. (eds.) Pattern Recognition. pp. 18–32. Springer International Publishing (2019)
13. Harer, J., Zagier, D.: The euler characteristic of the moduli space of curves. Inventiones mathematicae **85**(3), 457–485 (1986)
14. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum Contrast for Unsupervised Visual Representation Learning. arXiv:1911.05722 [cs] (Mar 2020), http://arxiv.org/abs/1911.05722, arXiv: 1911.05722
15. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006). https://doi.org/10.1126/science.1127647

16. Hinton, G.E., Roweis, S.T.: Stochastic neighbor embedding. In: Advances in neural information processing systems. pp. 857–864 (2003)
17. Huang, J., Gong, S., Zhu, X.: Deep Semantic Clustering by Partition Confidence Maximisation. pp. 8849–8858 (2020)
18. Kobak, D., Linderman, G.C.: Umap does not preserve global structure any better than t-sne when using the same initialization. BioRxiv (2019)
19. Kobak, D., Linderman, G.C.: Initialization is critical for preserving global data structure in both t-sne and umap. Nature biotechnology **39**(2), 156–157 (2021)
20. Kruskal, J.B.: Nonmetric multidimensional scaling: a numerical method. Psychometrika **29**(2), 115–129 (1964)
21. Li, S., Liu, Z., Wu, D., Liu, Z., Li, S.Z.: Boosting discriminative visual representation learning with scenario-agnostic mixup. arXiv preprint arXiv:2111.15454 (2021)
22. Li, S.Z., Zang, Z., Wu, L.: Deep manifold transformation for dimension reduction. arXiv preprint arXiv:2010.14831 (2020)
23. Li, Y., Hu, P., Liu, Z., Peng, D., Zhou, J.T., Peng, X.: Contrastive Clustering. AAAI2021 (Sep 2021), http://arxiv.org/abs/2009.09687, arXiv: 2009.09687
24. Liu, Z., Li, S., Di Wu, Z.C., Wu, L., Guo, J., Li, S.Z.: Automix: Unveiling the power of mixup (2021)
25. Liu, Z., Li, S., Wang, G., Tan, C., Wu, L., Li, S.Z.: Decoupled mixup for data-efficient learning. arXiv preprint arXiv:2203.10761 (2022)
26. Maaten, L.v.d.: Learning a Parametric Embedding by Preserving Local Structure. In: Artificial Intelligence and Statistics. pp. 384–391. PMLR (Apr 2009), http://proceedings.mlr.press/v5/maaten09a.html, iSSN: 1938-7228
27. Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
28. McInnes, L., Healy, J., Melville, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 [cs, stat] (Feb 2018), http://arxiv.org/abs/1802.03426, arXiv: 1802.03426 version: 1
29. Moon, K.R., van Dijk: Visualizing structure and transitions in high dimensional biological data. Nature biotechnology **37**(12), 1482–1492 (2019)
30. Moon, K.R., van Dijk, D., Wang, Z., Gigante, S., Burkhardt, D.B., Chen, W.S., Yim, K., van den Elzen, A., Hirn, M.J., Coifman, R.R., et al.: Visualizing structure and transitions in high-dimensional biological data. Nature biotechnology **37**(12), 1482–1492 (2019)
31. Nash, J.: The imbedding problem for riemannian manifolds. Annals of mathematics pp. 20–63 (1956)
32. Pers, T.H., Albrechtsen, A., Holst, C., Sørensen, T.I., Gerds, T.A.: The validation and assessment of machine learning: a game of prediction from high-dimensional data. PLoS One **4**(8), e6287 (2009)
33. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. science **290**(5500), 2323–2326 (2000), publisher: American Association for the Advancement of Science
34. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. science **290**(5500), 2323–2326 (2000)
35. Sainburg, T., McInnes, L., Gentner, T.Q.: Parametric UMAP embeddings for representation and semi-supervised learning. arXiv:2009.12981 [cs, q-bio, stat] (Apr 2021), http://arxiv.org/abs/2009.12981, arXiv: 2009.12981
36. Satake, I.: The gauss-bonnet theorem for v-manifolds. Journal of the Mathematical Society of Japan **9**(4), 464–492 (1957)

37. Szubert, B., Cole, J.E., Monaco, C., Drozdov, I.: Structure-preserving visualisation of high dimensional single-cell datasets. Scientific Reports **9**(1), 8914 (Jun 2019)
38. Tenenbaum, J.B.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science **290**(5500), 2319–2323 (Dec 2000). https://doi.org/10.1126/science.290.5500.2319, https://www.sciencemag.org/lookup/doi/10.1126/science.290.5500.2319
39. Weng, L.: Contrastive representation learning. lilianweng.github.io (2021), https://lilianweng.github.io/posts/2021-05-31-contrastive/
40. Wu, J., Long, K., Wang, F., Qian, C., Li, C., Lin, Z., Zha, H.: Deep Comprehensive Correlation Mining for Image Clustering. pp. 8150–8159 (2019)
41. Wu, Z., Xiong, Y., Yu, S., Lin, D.: Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination. arXiv:1805.01978 [cs] (May 2018), http://arxiv.org/abs/1805.01978, arXiv: 1805.01978
42. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: International Conference on Machine Learning. pp. 12310–12320. PMLR (2021)
43. Zhan, X., Xie, J., Liu, Z., Ong, Y.S., Loy, C.C.: Online deep clustering for unsupervised representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6688–6697 (2020)
44. Zhan, X., Xie, J., Liu, Z., Ong, Y.S., Loy, C.C.: Online Deep Clustering for Unsupervised Representation Learning. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6687–6696. IEEE, Seattle, WA, USA (Jun 2020). https://doi.org/10.1109/CVPR42600.2020.00672, https://ieeexplore.ieee.org/document/9157142/