Generalized Clustering and Multi-Manifold Learning with Geometric Structure Preservation

Lirong Wu [1,2], Zicheng Liu [2], Jun Xia [2], Zelin Zang [2], Siyuan Li [2], Stan Z. Li [2,†]
[1] Zhejiang University [2] Westlake University
{wulirong, liuzicheng, xiajun, zangzelin, lisiyuan, stan.zq.li}@westlake.edu.cn

Abstract

Though manifold-based clustering has become a popular research topic, we observe that one important factor has been omitted by these works, namely that the defined clustering loss may corrupt the local and global structure of the latent space. In this paper, we propose a novel Generalized Clustering and Multi-manifold Learning (GCML) framework with geometric structure preservation for generalized data, i.e., not limited to 2-D image data and has a wide range of applications in speech, text, and biology domains. In the proposed framework, manifold clustering is done in the latent space guided by a clustering loss. To overcome the problem that the clustering-oriented loss may deteriorate the geometric structure of the latent space, an isometric loss is proposed for preserving intra-manifold structure locally and a ranking loss for inter-manifold structure globally. Extensive experimental results have shown that GCML exhibits superior performance to counterparts in terms of qualitative visualizations and quantitative metrics, which demonstrates the effectiveness of preserving geometric structure. Code has been made available at: https://github.com/LirongWu/GCML.

1. Introduction

Clustering, a fundamental tool for data analysis and visualization, has been an essential research topic in data science. This paper focuses on *generalized clustering*, which takes vector data as input and is applicable to data with various dimensions, not limited to 2-D image data. Conventional clustering algorithms such as K-Means [11], Gaussian Mixture Models [1], and Spectral Clustering [19] perform clustering based on distance or similarity measures. However, handcrafted distance or similarity measures are rarely reliable for large-scale high-dimensional data, making it increasingly challenging to achieve effective clustering. An intuitive solution is to transform the data from the high-dimensional input space to the low-dimensional latent space and then to cluster the data in the latent space. This can be achieved by applying manifold-based dimensionality reduction techniques, such as t-SNE [10], and UMAP [13]. However, since these methods are not specifically designed for clustering tasks, some of their properties may be contrary to our expectations, e.g., two data points from different manifolds that are close in the input space will be closer in the latent space learned by UMAP. Therefore, the first question here is *how to perform multi-manifold learning for dimensionality reduction that favors clustering?*

The two main points for the multi-manifold learning are *Point (1)* preserving the local geometric structure within each manifold and *Point (2)* ensuring the discriminability between different manifolds. Most previous works seem to start with the assumption that data labels are known, and then design the algorithm in a *supervised manner*. However, it is challenging to decouple complex crossover relations and ensure discriminability between different manifolds, especially in *unsupervised* settings. One natural strategy is to achieve *Point (2)* through clustering to get pseudo-labels and then performing single-manifold learning for each manifold. However, the clustering-oriented loss may deteriorate the geometric structure of the latent space[1], and hence clustering is somewhat contrary to *Point (1)* (this will be detailed in Sec. 3.3). Therefore, it is important to alleviate this contradiction so that clustering helps both *Point (1)* and *Point (2)*. Thus, the second question here is *how to cluster data that favors multi-manifold learning?*

In this paper, we propose to jointly perform generalized clustering and multi-manifold learning with geometric structure preservation. Inspired by [24], the clustering centers are defined as a set of ***learnable*** parameters, and we use a clustering loss to simultaneously guide the separation of data points from different manifolds and the learning of the clustering centers. To prevent clustering loss from deteriorating the latent space, an isometric loss and a ranking loss are proposed to preserve the intra-manifold local structure and inter-manifold global structure. Finally, we achieve the

[1]This claim was first made by IDEC [4], but they did not provide any experiment or analysis to support it. In this paper, however, we show that the geometric structure of the latent space is indeed corrupted by extensive visualizations (Fig. 3 and Fig. 4) and statistical analysis (Fig. 6).

following three goals related to clustering, geometric structure preservation, and multi-manifold learning: (1) Clustering helps to ensure inter-manifold discriminability *(Point 2)*; (2) Local structure preservation *(Point 1)* can be compatible with clustering; (3) Geometric structure preservation helps to cluster. The contributions are summarized as:

- Proposing to combine generalized deep clustering and multi-manifold learning into a unified framework with local and global structure preservation.
- Setting the clustering centers as a set of *learnable* parameters and achieve global structure preservation in a faster, more efficient, and easier to optimize manner by applying ranking loss to the clustering centers.
- Pointing out contradictions between two optimization goals of clustering and local structure preservation and proposing an elegant training strategy to alleviate it.

2. Related Work

Clustering Analysis. As a fundamental tool in machine learning, clustering has been widely applied in various domains. One branch of classical clustering is K-Means and Gaussian Mixture Models (GMM), which are fast, easy to understand, and can be applied to a large number of problems. However, limited by Euclidean measures, their performance on high-dimensional data is often unsatisfactory. Spectral clustering and its variants (such as SC-Ncut [1]) extend clustering to high-dimensional data by allowing more flexible similarity measures. However, limited by the computational efficiency of the full Laplace matrix, spectral clustering is challenging to extend to large-scale datasets.

Deep Clustering. The success of deep learning has contributed to the growth of *deep clustering* [2, 9]. One branch of deep learning performs clustering after learning low-dimensional embeddings. For example, [21] uses autoencoder to learn low-dimensional features and then runs K-Means to get clustering results (AE+K-Means). Instead, N2D [12] applies UMAP to find the best clusterable manifold of the learned embeddings, and then runs K-Means to discover higher-quality clusters. The other category of algorithms tries to optimize clustering and dimensionality reduction jointly. The closest work to us is DEC [24], which learns a mapping from the input space to a low-dimensional latent space through iteratively optimizing clustering-oriented objective. As a modified version of DEC, while IDEC claims to preserve the local structure of data, their contribution is nothing more than adding a reconstruction loss. Besides, JULE [26] unifies representation learning with clustering, which can be considered as a neural extension of hierarchical clustering. Instead, SpectralNet [18] directly embeds the input into the Laplacian eigenspace in which clustering is performed. DSC [27] devises a dual autoencoder to embed data into latent space, and then deep spectral clustering is applied to obtain label assignments. Moreover, DDC [16] proposes to use a density-based clustering algorithm to initialize cluster centers and then perform image cluster discovery.

To avoid any possible misunderstanding, we would like to highlight that *generalized deep clustering and visual self-supervised learning (SSL)* are two different research fields. SSL typically uses more powerful CNN architecture (applicable only to 2-D image data) and applies sophisticated techniques such as contrastive learning [6], data augmentation, and clustering [28, 22] for better performance on large-scale datasets, such as ImageNet. For example, ASPC-DA [5] combines data augmentation with self-paced learning to encourage the learned embeddings to be cluster-oriented. Besides, ClusterGAN [14] trains a generative adversarial network jointly with a clustering-specific loss to achieve clustering in the latent space. The generalized deep clustering, on the other hand, uses a generalized MLP architecture (applicable to all kinds of data with various dimensions, not limited to 2-D image data) and has a very wide range of applications in images, text, and biology domains.

Manifold Learning. The manifold assumption hypothesizes that a low-dimensional manifold is embedded in a high-dimensional space, and the manifold learning aims to achieve dimensionality reduction via a nonlinear mapping that preserves the geometric structure. Isomap, as a classical algorithm of *single-manifold* learning, aims to seek a global optimal subspace that best preserves the geodesic distance between data points [20]. In contrast, some algorithms, such as the Locally Linear Embedding (LLE) [17], are more concerned with the preservation of local neighborhood information. Furthermore, *multi-manifold* learning has been proposed to obtain intrinsic properties of different manifolds. [25] proposes a supervised discriminant isomap where data points are partitioned into different manifolds according to label information. Similarly, [29] proposes a semi-supervised learning framework that applies the labeled and unlabeled training samples to perform the joint learning of local-preserving features. In most previous work, it is assumed that the label is known or partially known, which significantly simplifies the problem. However, it is challenging to decouple multiple overlapping manifolds in unsupervised settings, and that is what this paper aims to explore.

3. Proposed Method

Consider a dataset X with N samples, and each sample $x_i \in \mathbb{R}^d$ is sampled from C different manifolds $\{M_c\}_{c=1}^{C}$. Assume that each category in the dataset lies in a compact low-dimensional manifold, and the number of manifolds C is prior knowledge. Define two nonlinear mapping $z_i = f(x_i, \theta_f)$ and $y_i = g(z_i, \theta_g)$, where $z_i \in \mathbb{R}^m$ is the embedding of x_i in the latent space, and y_i is the reconstruction of x_i. The j-th cluster center is denoted as $\mu_j \in \mathbb{R}^m$, where $\{\mu_j\}_{j=1}^{C}$ is defined

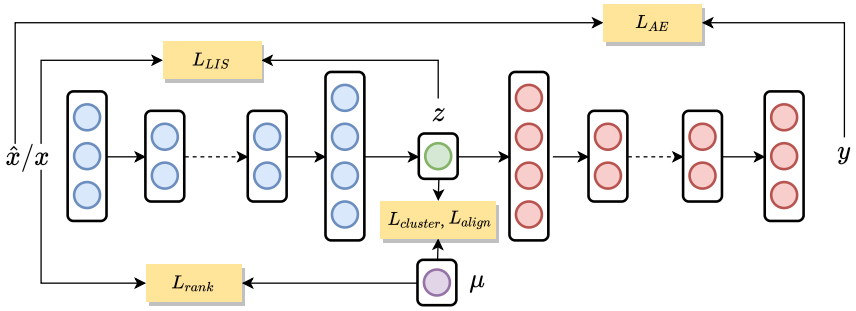


Figure 1. The GCML framework. The encoder, decoder, latent space, and cluster centers are marked as blue, red, green, and purple.

as a set of ***learnable*** parameters. We aim to find optimal parameters θ_f and $\{\mu_j\}_{j=1}^{C}$ so that the embeddings $\{z_i\}_{i=1}^{N}$ can achieve clustering with local and global structure preservation. To this end, a denoising autoencoder shown in Fig. 1 is first pre-trained in an unsupervised manner to learn an initial latent space. Denoising autoencoder aims to optimize the self-reconstruction loss $L_{AE} = MSE(\hat{x}, y)$, where the $\hat{x}$ is a copy of x with Gaussian noise added, that is, $\hat{x} = x + N(0, \sigma^2)$. Then the autoencoder is fine-tuned by optimizing the following clustering-oriented loss $\{L_{cluster}(z, \mu)\}$ and structure-oriented losses $\{L_{rank}(x, \mu), L_{LIS}(x, z), L_{align}(z, \mu)\}$. Since the clustering should be performed on features of *clean* data, instead of noised data $\hat{x}$ that is used in denoising autoencoder, the clean data x is used for fine-tuning.

3.1. Clustering-oriented Loss

First, the cluster centers $\{\mu_j\}_{j=1}^{C}$ in the latent space Z are initialized (the initialization method will be introduced in Sec. 4.1). Then the similarity between the embedded point z_i and cluster centers $\{\mu_j\}_{j=1}^{C}$ is measured by Student's t-distribution, as follows

$$q_{ij} = \frac{\left(1 + \|z_i - \mu_j\|^2\right)^{-1}}{\sum_{j'} \left(1 + \|z_i - \mu_{j'}\|^2\right)^{-1}} \tag{1}$$

The auxiliary target distribution is further designed to help manipulate the latent space, defined as:

$$p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_{j'} q_{ij'}^2 / f_{j'}}, \quad where \quad f_j = \sum_i q_{ij} \tag{2}$$

where f_j is the normalized cluster frequency, used to balance the size of different clusters. Then the encoder is optimized by the following objective:

$$L_{cluster} = \mathrm{KL}(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3}$$

The gradient of $L_{cluster}$ with respect to each learnable cluster center μ_j can be computed as:

$$\frac{\partial L_{cluster}}{\partial \mu_j} = -\sum_i \left(1 + \|z_i - \mu_j\|^2\right)^{-1} \cdot (p_{ij} - q_{ij})(z_i - \mu_j) \tag{4}$$

where $L_{cluster}$ facilitates the aggregation of data points within the same manifold, while data points from different manifolds are kept away from each other. However, we found that the clustering-oriented losses may deteriorate the geometric structure of the latent space. To prevent this deterioration, we introduces three other structure-oriented losses to preserve the local and global manifold structures.

3.2. Structure-oriented Loss

Intra-manifold Isometry Loss. The intra-manifold local structure is preserved by optimizing the objective as:

$$L_{LIS} = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i^Z} |d_X(x_i, x_j) - d_Z(z_i, z_j)| \cdot \pi(l(x_i) = l(x_j)) \tag{5}$$

where $\mathcal{N}_i^Z$ represents the neighborhood of data point z_i in the latent space Z, and the kNN is applied to determine the neighborhood. $\pi(\cdot) \in \{0, 1\}$ is an indicator function, and $l(x_i)$ is a *manifold determination function* that returns the manifold s_i where sample x_i is located, that is, $s_i = l(x_i) = \arg\max_j p_{ij}$. Then we can derive C manifolds $\{M_c\}_{c=1}^{C}$ by $M_c = \{x_i;\ s_i = c, i = 1, 2, ..., N\}$. The loss L_{LIS} constrains the isometry within each manifold.

Inter-manifold Ranking Loss. The inter-manifold global structure is preserved by optimizing the objective as:

$$L_{rank} = \sum_{i=1}^{C} \sum_{j=1}^{C} \left| d_Z(\mu_i, \mu_j) - \kappa \cdot d_X\left(v_i^X, v_j^X\right) \right| \tag{6}$$

where $\{v_j^X\}_{j=1}^{C}$ is defined as the ground-truth centers of different manifolds in the input space X with $v_j^X = \frac{1}{|M_j|} \sum_{i \in M_j} x_i$ $(j = 1, 2, ..., C)$. The parameter κ determines the extent to which different manifolds move away from each other. The larger κ is, the further away the different manifolds are from each other. The derivation for the gradient of L_{rank} with respect to each learnable cluster center μ_j is placed in **Appendix A.2**. Note that L_{rank} is optimized in an *iterative* manner, rather than by initializing $\{\mu_j\}_{j=1}^{C}$ once and then separating different clusters based only on initialization results. Additionally, contrary to us,

the conventional methods for dealing with inter-manifold separation typically impose push-away constraints on all data points from different manifolds [29, 25], defined as:

$$L_{sep} = -\sum_{i=1}^{N}\sum_{j=1}^{N} d_Z\left(z_i, z_j\right) \cdot \pi(l(x_i) \neq l(x_j)) \tag{7}$$

The main differences between L_{rank} and L_{sep} are as follows: (1) L_{sep} imposes constraints on embedding $\{z_i\}_{i=1}^{N}$, which indirectly affects the network parameters θ_f. In contrast, L_{rank} imposes constrains directly on parameters $\{\mu_j\}_{j=1}^{C}$ in the form of *regularization* item. (2) L_{rank} is easier to optimize, faster to process, and more accurate. L_{sep} is imposed on all data points from different manifolds, which involves $N\times N$ *point-to-point* relationships. This means that each point may be subject to the push-away force from other manifolds, but at the same time, each point has to meet the isometry constraint with its neighboring points. Under these two constraints, optimization is difficult and it is easy to fall into a local optimal solution. In contrast, L_{rank} is imposed directly on the clustering centers, involving only $C\times C$ *cluster-to-cluster* relationships, which avoids the above problem and makes it easier to optimize. (3) The parameter κ introduced in L_{rank} allows us to control the extent of separation between manifolds.

Alignment Loss. Global ranking loss L_{rank} is imposed directly on $\{\mu_j\}_{j=1}^{C}$, so optimizing L_{rank} only updates $\{\mu_j\}_{j=1}^{C}$ rather the encoder's parameter θ_f. However, the optimization of $\{\mu_j\}_{j=1}^{C}$ not only relies on L_{rank}, but is also constrained by $L_{cluster}$, which ensures that data points remain roughly distributed around cluster centers and do not deviate significantly from them during the optimization process. Alignment loss L_{align}, as an auxiliary term, aims to help align learnable centers $\{\mu_j\}_{j=1}^{C}$ with the ground-truth centers $\{v_j^Z\}_{j=1}^{C}$ and *make this binding stronger*, defined as

$$L_{align} = \sum_{j=1}^{C} ||\mu_j - v_j^Z|| \tag{8}$$

where $\{v_j^Z\}_{j=1}^{C}$ are defined as $v_j^Z = \frac{1}{|M_j|}\sum_{i\in M_j} z_i$. The derivation for the gradient of L_{align} with respect to center μ_j is placed in **Appendix A.2**. As shown in Fig. 1, three structure-oriented losses $\{L_{rank}(x,\mu), L_{LIS(x,z)}, L_{align}(z,\mu)\}$, form a closed loop between input x, embeddings z, and cluster centers μ.

3.3. Training Strategy

Contradiction. The contradiction between clustering and local structure preservation is analyzed from the *forces analysis* perspective. As shown in Fig. 2 (a), we assume that there exists a data point (red point) and its three nearest neighbors (blue points) around a cluster center (gray point). When clustering and local structure preserving are optimized simultaneously, it's easy to fall into a local optimum, where the data point is in steady-state, and the resultant force from its three nearest neighbors is equal in magnitude and opposite to the gravitational forces of the cluster.

Alternating Training. To solve the above problem and integrate the goals of clustering and local structure preservation into a unified framework, we take an alternating training strategy. Within each epoch, we first optimize $L_{cluster}$ and L_{rank} in a *mini-batch*, with joint loss defined as

$$L_1 = \alpha L_{cluster} + L_{rank} \tag{9}$$

Then at each epoch, we jointly optimize isometry loss L_{LIS} and L_{align} on *the whole dataset*, defined as

$$L_2 = \beta L_{LIS} + L_{align} \tag{10}$$

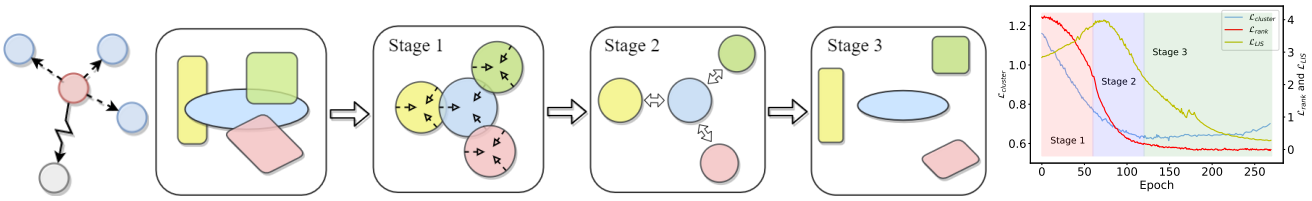
Weight Continuation. At different stages of training, we have different requirements for clustering and structure preservation. At the beginning of training, to successfully decouple the overlapping manifolds, we hope that the $L_{cluster}$ will dominate and L_{LIS} will be auxiliary. When the margin between different manifolds is sufficiently large, the weight α for $L_{cluster}$ can be gradually reduced, while the weight β for L_{LIS} can be gradually increased, focusing on the preservation of the local isometry. The whole algorithm is summarized in Algorithm 1.

Algorithm 1 Algorithm for GCML

Input: Input samples: X; Number of clusters: C; Number of batches: B; Number of iterations: E.
Output: Autoencoder's weights: θ_f and θ_g; Cluster labels $\{s_i\}_{i=1}^{N}$; Trainable cluster centers $\{\mu_j\}_{j=1}^{C}$.

Initialize the weight $\{\mu_j\}_{j=1}^{C}$, θ_f and θ_g, and obtain initialized soft label assignment $\{s_i\}_{i=1}^{N}$.
for *epoch* $\in$ {0,1,$\cdots$,E-1} **do**
  Compute embedded points $\{z_i\}_{i=1}^{N}$ and distribution Q;
  Update target distribution P;
  Compute soft cluster centers $\{v_i^X\}_{i=1}^{C}$ and $\{v_i^Z\}_{i=1}^{C}$.
  for *batch* $\in$ {0,1,$\cdots$,B} **do**
    Pick up one batch of samples X_{batch} from X;
    Compute corresponding distribution Q_{batch} and it's reconstruction Y_{batch};
    Pick up target distribution batch P_{batch} from P;
    Compute loss L_{ae}, $L_{cluster}$ and L_{rank};
    Update the weight θ_f, θ_g and $\{\mu_j\}_{j=1}^{C}$.
  end for
  Compute L_{iso} and L_{align} on all samples;
  Update the weight θ_f and $\{\mu_j\}_{j=1}^{C}$;
  Assign new soft labels $\{s_i\}_{i=1}^{N}$.
end for
return θ_f, θ_g, $\{s_i\}_{i=1}^{N}$, $\{\mu_j\}_{j=1}^{C}$.

Three-stage Explanation. The training process can be roughly divided into three stages, as shown in Fig. 2 (b), to explain the training strategy more vividly. Also, we provide the learning curves of key losses $L_{cluster}$, L_{rank}, L_{LIS} on the MNIST-test dataset in Fig. 2 (c). At first, four different manifolds overlap. At Stage 1, $L_{cluster}$ dominates, thus data points within each manifold converge towards cluster centers to form spheres, but the local structure of manifolds



(a) (b) Schematic of training strategy (c) Learning curve

Figure 2. Fig. 2 (a) is the force analysis of the contradiction between clustering and local structure preservation. Fig. 2 (b) is the schematic of training strategy, where four different colors and shapes represent four intersecting manifolds, and three stages involve the manifold clustering, separation, and geometric recovery. Fig. 2 (c) is the learning curve of losses $L_{cluster}$, L_{rank}, L_{LIS} on the MNIST-test dataset.

is corrupted. At Stage 2, L_{rank} dominates, thus different manifolds in the latent space move away from each other to increase the manifold margin and enhance the discriminability. At stage 3, these manifolds gradually recover their original local structure from the spherical shape with L_{LIS} dominating. Note that the above losses may coexist rather than being completely independent at different stages, but the role played by different losses varies due to the alternating training and weight continuation strategies.

4. Experiments

4.1. Experimental Setups

In this section, the effectiveness of GCML is evaluated in 7 benchmark datasets: MNIST-full, MNIST-test, USPS, Fashion-MNIST, REUTERS-10K, HAR, and Pendigits on which GCML is compared with 12 other methods in 8 evaluation metrics including metrics specifically designed for clustering and multi-manifold learning. The brief descriptions of the datasets are given in **Appendix A.1**.

Evaluation Metrics. Two standard evaluation metrics: Accuracy (ACC) and Normalized Mutual Information (NMI) are used to evaluate clustering performance. Besides, six evaluation metrics are adopted in this paper to evaluate the performance of multi-manifold learning, including Relative Rank Error (RRE), Trustworthiness (Trust), Continuity (Cont), Root Mean Reconstruction Error (RMRE), Locally Geometric Distortion (LGD) and Cluster Rank Accuracy (CRA). Limited by space, their precise definitions are available in **Appendix A.3**.

Parameters Settings. The encoder structure is d-500-500-500-2000-10 where d is the dimension of the input data, and the decoder is its mirror. After pretraining, to initialize the learnable clustering centers, the t-SNE is applied to find the best clustable manifold in the latent space Z, and then the K-Means algorithm is run to obtain the label assignments for each data point. The centers of each category in the latent space Z are set as initial cluster centers $\{\mu_j\}_{j=1}^{C}$. Besides, Adam optimizer with learning rate λ=0.001 is used, the batch size is set to 256, the epoch is set to 300, the parameter k for nearest neighbor is set to 5, and the parameter κ is set to 3 for all datasets. Sensitivity analysis for parameters k and κ is available in **Appendix A.4**. As described in Sec. 3.3, the weight continuation is applied to train the model. The weight parameter α for $L_{cluster}$ decreases linearly from 0.1 to 0 within epoch 0-150. In contrast, the weight parameter β for L_{LIS} increases linearly from 0 to 1.0 within epoch 0-150. In this paper, each set of experiments is run 5 times with different random seeds, and the results are averaged into the final performance metrics.

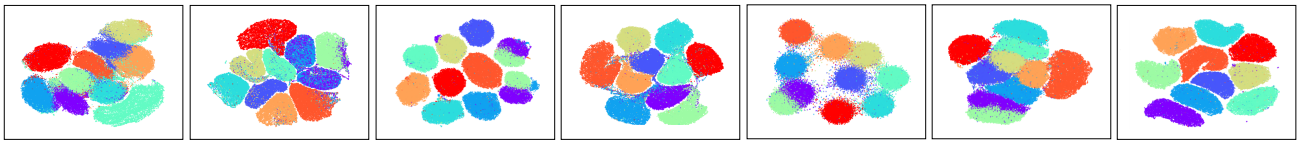
4.2. Evaluation of Clustering

Quantitative Comparison. The metrics ACC/NMI of different methods are reported in Tab. 1. For those methods whose results are not reported or experimental settings are not clear, we run the released code with the same provided hyperparameters and mark them with (*). Moreover, we mark those methods that are only applicable to 2-D image data as (-) on the *vector* dataset. While ASPC-DA achieves the best performance on three datasets (MNIST-test, MNIST-full, and USPS), its performance gains do not come directly from clustering, but from sophisticated modules such as data augmentation and self-paced learning. Once these modules are removed, there is large performance degradation. For example, with Data Augmentation (DA) removed, ASPC achieves less competitive performance, e.g., an accuracy of 0.931 (*vs* 0.988) on MNIST-full, 0.813 (*vs* 0.973) on MNIST-test and 0.768 (*vs* 0.982) on USPS. Since ASPC-DA is based on the MLP architecture, its image-based augmentation cannot be applied directly to vector data, which explains why ASPC has no performance advantage (even compared to DEC and IDEC) on the vector datasets, such as REUTERS-10K and HAR datasets.

In a fair comparison (without considering ASPC-DA and marking its results in Tab. 1 as gray color), we find that GCML outperforms K-Means, GNN, and SC-Ncut by a significant margin and surpasses the other nine compared DNN-based algorithms on all datasets except MNIST-test. Nevertheless, even with the MNIST-test dataset, GCML still ranks second, outperforming the third by 0.9%. In particular, we obtain the best performance on the Fashion-MNIST, REUTERS-10K, and HAR datasets, and more no-

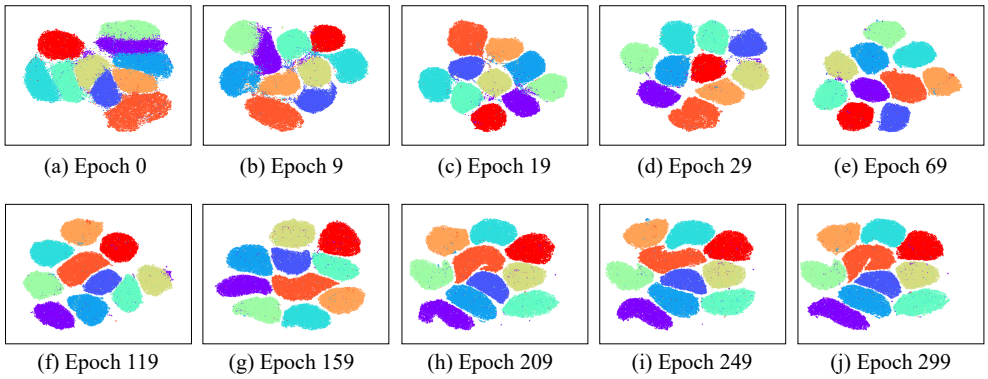
Table 1. Clustering performance (ACC/NMI) of different algorithms on seven datasets. The best metrics are marked in **bold**.

Algorithms	MNIST-full	MNIST-test	USPS	Fashion-MNIST	REUTERS-10K	HAR	pendigits
K-Means [1]	0.532/0.500	0.546/0.501	0.668/0.601	0.474/0.512	0.599/0.375*	0.599/0.588	0.666/0.681
SC-Ncut [1]	0.656/0.731	0.660/0.704	0.649/0.794	0.508/0.575	0.658/0.401*	0.538/0.741	0.724/0.784
GMM [1]	0.389/0.333	0.464/0.465	0.562/0.540	0.463/0.514	0.613/0.388*	0.585/0.648	0.673/0.682
AE+K-Means [21]	0.818/0.747	0.815/0.784*	0.662/0.693	0.566/0.585*	0.721/0.432*	0.674/0.670*	0.713/0.733*
DEC [24]	0.903/0.854*	0.885/0.851*	0.889/0.873*	0.554/0.576*	0.773/0.528*	0.759/0.695*	0.679/0.671*
IDEC [4]	0.918/0.868*	0.876/0.817*	0.893/0.876*	0.572/0.601*	0.785/0.541*	0.786/0.718*	0.739/0.757*
VaDE [7]	0.945/0.876	0.287/0.287	0.566/0.512	0.578/0.630	0.795/0.556*	0.801/0.720*	0.762/0.743*
DEPICT [3]	0.965/0.917	0.963/0.915	0.899/0.906	0.392/0.392	-	-	-
JULE [26]	0.964/0.913	0.961/0.915	0.950/0.913	0.563/0.608	-	-	-
DSC [27]	0.978/0.941	**0.980/0.946**	0.869/0.857	0.662/0.645	-	-	-
ASPC-DA [5]	0.988/0.966	0.973/0.936	0.982/0.951	0.591/0.654	-	-	-
ASPC (w/o DA) [5]	0.931/0.886*	0.813/0.792*	0.768/0.803*	0.576/0.632*	0.692/0.418*	0.769/0.682*	0.769/0.751*
N2D [12]	0.969/0.928*	0.954/0.897*	0.954/0.898*	0.671/0.678*	0.784/0.536*	0.807/0.721*	0.847/0.808*
GCML (ours)	**0.980/0.946**	0.972/0.930	**0.958/0.902**	**0.710/0.685**	**0.836/0.590**	**0.844/0.762**	**0.855/0.814**



(a) AE+K-mean (b) DEC (c) IDEC (d) JULE (e) DSC (f) N2D (g) GCML (Ours)

Figure 3. Visualization of the embeddings learned by different algorithms on the MNIST-full dataset.



(a) Epoch 0 (b) Epoch 9 (c) Epoch 19 (d) Epoch 29 (e) Epoch 69

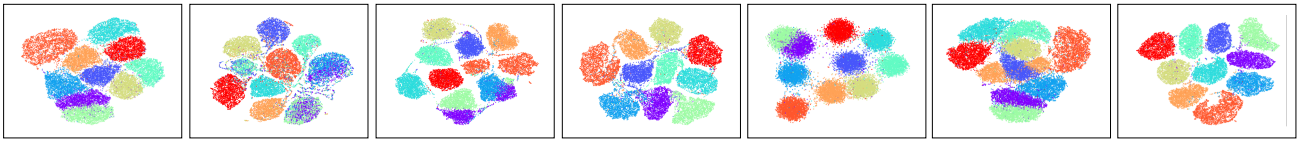
(f) Epoch 119 (g) Epoch 159 (h) Epoch 209 (i) Epoch 249 (j) Epoch 299

Figure 4. Clustering visualization at different stages during the training process on the MNIST-full dateset.

tably, our clustering accuracy exceeds the current SOTA method by 3.9%, 4.1%, and 3.8%, respectively.

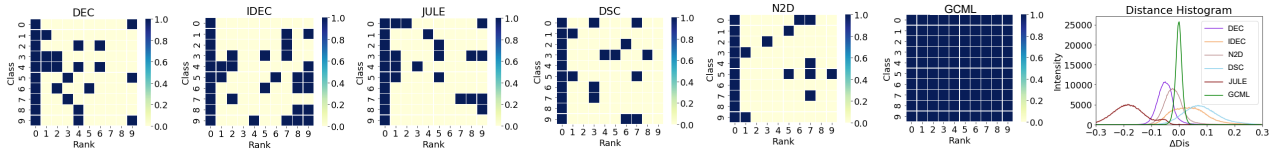
Clustering Visualization. The visualization of GCML with comparison methods is shown in Fig. 3 (visualized using UMAP). Among all methods, only DEC, IDEC, and GCML can hold clear boundaries between different clusters, while the cluster boundaries of the other methods are indistinguishable. Though DEC and IDEC successfully separate different clusters, they group many data points from different classes into the same cluster. Most importantly, due to the use of the clustering-oriented loss, the embedding learned by these algorithms (especially DSC) *tend to form spheres and disrupt the original topological structure*. Instead, GCML overcomes the above problems and achieves almost perfect separation between different clusters while preserving the local and global structure.

The embeddings of the latent space during the training process are visualized in Fig. 4 for explaining how both clustering and structure-preserving are achieved. We can see that the different clusters initialized by the pre-trained autoencoder are closely adjacent. In the early stage of training, with clustering loss $L_{cluster}$ and global ranking loss L_{rank}, different manifolds are separated from each other, but each manifold loses its local structure, and all of them degenerate into spheres. As the training progresses, the weight α for $L_{cluster}$ gradually decreases, while the weight β for L_{LIS} increases and *the optimization is gradually focused from global to local*, with each manifold gradually recovering its original geometric structure from the sphere. These visualizations show that clustering-oriented loss does



(a) AE+K-mean (b) DEC (c) IDEC (d) JULE (e) DSC (f) N2D (g) GCML (Ours)

Figure 5. Visualization of the embeddings learned from testing samples on the MNIST-full dataset.



(a) DEC (b) IDEC (c) JULE (d) DSC (e) DSC (f) GCML(Ours) (g) Local Isometry

Figure 6. Statistical analysis for evaluating the capability of geometric structure preservation from the input space to the latent space.

deteriorate the geometric structure of the latent space, and the designed structure-oriented losses help to recover it.

Generalizability Evaluation. Tab. 2 demonstrates that a learned GCML can generalize well to unseen data with high clustering accuracy. Taking MNIST-full as an example, GCML was trained using 50,000 training samples and then tested on the remaining 20,000 testing samples using the learned model. In terms of the metrics ACC and MNI, GCML is optimal for both training and testing samples. More importantly, there is hardly any degradation in the performance of GCML on the testing samples compared to the training samples, while all other methods show a significant drop in performance, e.g., DEC from 84.1% to 74.8%. This demonstrates the importance of geometric structure preservation for good generalizability. The visualization results of the testing samples are shown in Fig. 5; even for testing samples, GCML still shows distinguishable inter-cluster discriminability, while other methods couple different clusters together, which shows GCML's great generalizability.

Table 2. Generalizability evaluated by ACC/NMI.

Algorithms	**training samples**	**testing samples**
AE+K-Means	0.815/0.736	0.751/0.711
DEC	0.841/0.773	0.748/0.704
IDEC	0.845/0.860	0.826/0.842
JULE	0.958/0.907	0.921/0.895
DSC	0.975/0.939	0.969/0.921
N2D	0.974/0.930	0.965/0.911
GCML (ours)	**0.978/0.941**	**0.978/0.941**

4.3. Evaluation of Multi-Manifold Learning

Quantitative Metrics. Though many previous works have claimed that they brought clustering and dimensionality reduction into a unified framework, unfortunately, they all lacked an analysis of the effectiveness of the learned embeddings. In this paper, we compare GCML with the other five methods in six quantitative metrics on seven datasets. Limited by space, only the results of MNIST-full and Fashion-MNIST are provided on the left side of Tab. 3 and more results are in **Appendix A.5**. The results show that GCML outperforms all other methods, especially in the CRA metric, which is not only the best on all datasets but reaches 1.0, which means that the "rank" between different manifolds in the latent space is completely preserved and proves the effectiveness of the global ranking loss L_{rank}.

Statistical Analysis. The statistical analysis is performed to show the extent to which local and global structure is preserved in the latent space for each algorithm. Taking MNIST-full as an example, the statistical analysis of the global rank-preservation is shown in Fig. 6 (a)-(f). For the i-th cluster, if the rank (in terms of Euclidean distance) between it and the j-th cluster is preserved from the input space to the latent space, then the grid in the i-th row and j-th column is marked as blue, otherwise yellow. As shown in the figure, only GCML can fully preserve the global rank between different clusters, while all other methods fail.

Moreover, we perform statistical analysis for the local isometry property of each algorithm. For each sample x_i, it forms a number of point pairs with its neighborhood samples $\{(x_i, x_j)|i = 1, 2, ..., N; x_j \in \mathcal{N}_i^X\}$. We compute the differences in the distance of these point pairs from the input space to the latent space $\{d_Z(x_i, x_j) - d_X(x_i, x_j)|i = 1, 2, ..., N; x_j \in \mathcal{N}_i\}$, and plot them as a histogram. As shown in Fig. 6 (g), the curve of GCML are distributed on both sides of the 0 value, with *maximum peak height* and *minimum peak-bottom width*, respectively, which indicates that GCML achieves the best local isometry. Though IDEC [4] claims that they can preserve the local structure well, their results are still far from ours.

Downstream Tasks. Numerous deep clustering algo-

Table 3. Performance for multi-manifold learning (left) and downstream tasks (right) on the MNIST-full and Fashion-MNIST datasets.

Datasets	Algorithms	RRE↓	Trust↑	Cont↑	d-RMSE↓	LGD↓	CRA↑	MLP↑	RFC↑	SVM↑	LR↑
MNIST-full	DEC	0.09988	0.84499	0.94805	44.8535	4.37986	0.28	0.8647	0.8706	0.8707	0.8566
	IDEC	0.00984	0.99821	0.97936	24.5803	1.71484	0.33	0.9797	0.9737	0.9852	0.9650
	JULE	0.02657	0.93675	0.98321	28.3412	2.12955	0.27	0.9802	0.9825	0.9787	0.9743
	DSC	0.09785	0.87315	0.92508	6.98098	1.19886	0.23	0.9622	0.9501	0.9837	0.9752
	N2D	0.01002	0.99243	0.98466	5.7162	0.69946	0.21	0.9796	0.9803	0.9799	0.9792
	GCML (ours)	**0.00567**	**0.99978**	**0.98716**	**5.4986**	**0.69168**	**1.00**	**0.9851**	**0.9874**	**0.9869**	**0.9841**
Fashion-MNIST	DEC	0.04787	0.93896	0.95450	39.3274	3.87731	0.37	0.6268	0.9853	0.6377	0.6245
	IDEC	0.01089	0.99683	0.97797	25.4024	1.91385	0.27	0.8367	0.9918	**0.8607**	0.7514
	JULE	0.03013	0.97732	0.97923	15.2213	1.43642	0.43	0.8541	0.9892	0.8566	0.7723
	DSC	0.05168	0.95013	0.96121	17.2201	1.42091	0.36	0.8084	0.9823	0.8618	0.7676
	N2D	0.00894	0.99062	0.98054	14.49079	**1.28180**	0.26	0.8412	0.9493	0.8230	0.7753
	GCML (ours)	**0.00836**	**0.99868**	**0.98203**	**13.3788**	1.33893	**1.00**	**0.8642**	**0.9942**	0.8468	**0.7768**

Table 4. Ablation study of loss items and training strategies used in the proposed GCML framework.

Datasets	Methods	ACC/NMI↑	RRE↓	Trust↑	Cont↑	d-RMSE↓	LGD↓	CRA↑
MNIST-full	w/o SL	0.976/0.939	0.0093	0.9967	0.9816	24.589	1.6747	0.32
	w/o CL	0.814/0.736	0.0004	0.9998	0.9990	7.458	0.0487	1.00
	w/o WC	0.977/0.943	0.0065	0.9987	0.9860	5.576	0.6968	0.98
	w/o AT	0.978/0.944	0.0069	0.9986	0.9851	5.617	0.7037	0.96
	full model	**0.980/0.946**	**0.0056**	**0.9997**	**0.9871**	**5.498**	**0.6916**	**1.00**
Fashion-MNIST	w/o SL	0.706/0.682	0.0108	0.9964	0.9781	25.954	1.8936	0.30
	w/o CL	0.576/0.569	0.0004	0.9994	0.9995	7.654	0.0523	1.00
	w/o WC	0.702/0.695	0.0084	0.9972	0.9814	**13.238**	1.3474	**1.00**
	w/o AT	0.708/0.694	0.0097	0.9975	0.9798	13.354	1.3611	**1.00**
	full model	**0.710/0.685**	**0.0083**	**0.9986**	**0.9820**	13.378	**1.3389**	**1.00**

rithms have recently claimed to obtain meaningful low-dimensional embeddings; however, they have not analyzed and experimented with the "meaningful" ones. Therefore, we are interested in whether these proposed methods can really learn manifold embeddings that are useful for downstream tasks. Four different classifiers, including a linear classifier (Logistic Regression; LR), two nonlinear classifiers (MLP, SVM), and a tree-based classifier (Random Forest Classifier; RFC) are used as downstream tasks, all of which use default parameters and default implementations in sklearn [15] for a fair comparison. The learned embeddings are *frozen* and used as input for training. The classification accuracy evaluated on the test set serves as a metric to evaluate the effectiveness of learned embeddings. Limited by space, only the results of MNIST-full and Fashion-MNIST are provided on the right side of Tab. 3 and more results are in **Appendix A.5**. The results show that GCML outperforms the other methods overall on all seven datasets, with MLP, RFC, and LR as downstream tasks.

4.4. Ablation Study

Tab. 4 evaluates the effectiveness of the proposed loss terms and training strategies with 5 sets of experiments: the model without (A) Structure-oriented Loss (SL); (B) Clustering-oriented Loss (CL); (C) Weight Continuation (WC); (D) Alternating Training (AT), and (E) the full model. Limited by space, only the results MNIST-full and Fashion-MNIST are provided and more results are in **Appendix A.6**. After analyzing the results, we can conclude: (1) CL is the most important factor for obtaining excellent clustering performance, the lack of which leads to unsuccessful clustering, hence the numbers in the table are not meaningful and marked in gray color. (2) SL not only brings subtle improvements in clustering performance but greatly improves the performance of multi-manifold learning. (3) The training strategies (WC and AT) both improve the performance of clustering and multi-manifold learning to some extent, especially on metrics such as RRE, Trust, CRA, etc.

5. Conclusion

The proposed GCML framework imposes clustering-oriented and structure-oriented constraints to optimize the latent space for simultaneously performing clustering and multi-manifold learning with geometric structure preservation. Extensive experiments demonstrate that GCML is not only comparable to the SOTA clustering algorithms but learns effective manifold embeddings, which is beyond the capability of those algorithms that only care about clustering accuracy. Finally, GCML is appliable to generalized data with various dimensions, not limited to 2-D image data.

References

[1] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[2] Zhangyang Gao, Haitao Lin, Stan Li, et al. Clustering based on graph of density topology. *arXiv preprint arXiv:2009.11612*, 2020.

[3] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE international conference on computer vision*, pages 5736–5745, 2017.

[4] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017.

[5] Xifeng Guo, Xinwang Liu, En Zhu, Xinzhong Zhu, Miaomiao Li, Xin Xu, and Jianping Yin. Adaptive self-paced deep clustering with data augmentation. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[6] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[7] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016.

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[9] Stan Z Li, Lirong Wu, and Zelin Zang. Consistent representation learning for high dimensional data analysis. *arXiv preprint arXiv:2012.00481*, 2020.

[10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[11] J MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Math., Stat., and Prob*, page 281, 1965.

[12] Ryan McConville, Raul Santos-Rodriguez, Robert J Piechocki, and Ian Craddock. N2d:(not too) deep clustering via clustering the local manifold of an autoencoded embedding. *arXiv preprint arXiv:1908.05968*, 2019.

[13] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[14] Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4610–4617, 2019.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[16] Yazhou Ren, Ni Wang, Mingxia Li, and Zenglin Xu. Deep density-based image clustering. *Knowledge-Based Systems*, 197:105841, 2020.

[17] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[18] Uri Shaham, Kelly Stanton, Henry Li, Boaz Nadler, Ronen Basri, and Yuval Kluger. Spectralnet: Spectral clustering using deep neural networks. *arXiv preprint arXiv:1801.01587*, 2018.

[19] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[20] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[21] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *Aaai*, volume 14, pages 1293–1299. Citeseer, 2014.

[22] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[23] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[24] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.

[25] Bo Yang, Ming Xiang, and Yupei Zhang. Multi-manifold discriminant isomap for visualization and classification. *Pattern Recognition*, 55:215–230, 2016.

[26] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016.

[27] Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4066–4075, 2019.

[28] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6688–6697, 2020.

[29] Yan Zhang, Zhao Zhang, Jie Qin, Li Zhang, Bing Li, and Fanzhang Li. Semi-supervised local multi-manifold isomap by linear embedding for feature extraction. *Pattern Recognition*, 76:662–678, 2018.