High-dimensional Clustering onto Hamiltonian Cycle

Tianyi Huang [1 2] **Shenghui Cheng** [1 2] **Stan Z. Li** [1 2] **Zhengjun Zhang** [3 4]

Abstract

Clustering aims to group unlabelled samples based on their similarities and is widespread in high-dimensional data analysis. However, most of the clustering methods merely generate pseudo labels and thus are unable to simultaneously present the similarities between different clusters and outliers. This paper proposes a new framework called High-dimensional Clustering onto Hamiltonian Cycle (HCHC) to solve the above problems. First, HCHC combines global structure with local structure in one objective function for deep clustering, improving the labels as relative probabilities, to mine the similarities between different clusters while keeping the local structure in each cluster. Then, the anchors of different clusters are sorted on the optimal Hamiltonian cycle generated by the cluster similarities and mapped on the circumference of a circle. Finally, a sample with a higher probability of a cluster will be mapped closer to the corresponding anchor. In this way, our framework allows us to appreciate three aspects visually and simultaneously - clusters (formed by samples with high probabilities), cluster similarities (represented as circular distances), and outliers (recognized as dots far away from all clusters). The theoretical analysis and experiments illustrate the superiority of HCHC.

1. Introduction

High-dimensional data, i.e., the data described by a large number of features, are widely existing in many research fields, such as image processing, pattern recognition, and bioinformatics (Bühlmann & Van De Geer, 2011; Donoho et al., 2000). Analyzing high-dimensional data is a significant but challenging task (Verleysen et al., 2003). Clustering is widespread in high-dimensional data analysis (Mccarthy et al., 2004; Jain et al., 1999). It can group samples so that the samples within the same cluster are broadly more similar to one another than those in other clusters (Hartigan, 1975; Niu et al., 2022; Huang et al., 2020). Then by simultaneously presenting the clusters, similarities, and outliers, we can well recognize the insight of high-dimensional data. For example, in the clustering of community detection, we can find the topological information between the different clusters of the community nodes by presenting the similarities between these clusters and thus explain the community detection result (Fortunato, 2010). However, most of the traditional clustering methods merely generate pseudo labels. In this case, it is hard to get the knowledge of similarities and outliers in the clustering.

A frequently-used tool to explore the outliers and similarities between the different clusters is the dendrogram in hierarchical clustering (Guha et al., 1998; Karypis et al., 1999). But, some critical knowledge for explaining the recognized similar clusters and outliers cannot be presented in this way. For example, the dendrogram cannot present the in-between samples of the different clusters to explain the recognized similar clusters. It also cannot present the clustering probability distributions of the samples to explain the recognized outliers. Deep clustering methods, such as DEC, have been proposed to cluster high-dimensional data by simultaneously learning clustering probability distributions and the embedded features of the samples (Aljalbout et al., 2018; Xie et al., 2016). Then, we can extract similarities and outliers in the generated clustering probability distributions (Li et al., 2020). Unfortunately, there does not exist an effective way to simultaneously present the mined knowledge, including clusters, similarities, and outliers. Although some embedding methods, such as MDS and t-SNE (Kruskal, 1978; Van der Maaten & Hinton, 2008; McInnes et al., 2018), can visualize the distances between the samples or the local-manifold in each cluster, the visualized result may be inconsistent with the clustering result. One reason is that it is hard to visualize all of the distinguishing information of high-dimensional in 2D space. The other one is that these methods may have their limits in the

[1]School of Engineering, Westlake University, Hangzhou, China [2]Westlake Institute for Advanced Study, Hangzhou, China [3]School of Economics and Management, the University of Chinese Academy of Sciences, Beijing, China [4]School of Computer, Data & Information Sciences, the University of Wisconsin, Madison, USA. Correspondence to: Shenghui Cheng <Chengshenghui@westlake.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

visualization as shown in Appendix D.1 (Zu & Tao, 2022; Li et al., 2020). For example, in the embedding space of deep clustering, MDS cannot well keep the local-strcutre of data and t-SNE cannot capture the global structure of data.

This paper proposes High-dimensional Clustering onto Hamiltonian Cycle (HCHC) to solve the above problems. It comprises two key components: (1) extracting the clustering probability distributions by deep clustering and (2) mapping the clustering probability distributions onto the optimal Hamiltonian cycle. For extracting the clustering probability distributions, we construct a new deep clustering method, GLDC, by combining global structure with local structure in one objective function. GLDC constructs a weighted adjacency matrix associated with a similarity graph of the samples. By learning the structure of the unconnected samples and the connected samples, respectively, the loss function of GLDC can well mine the similarities between different clusters while keeping the local-structure of the samples in the same cluster. Thus, the extracted clustering probability distributions in GLDC can well represent the similarities between different clusters and the samples in the same cluster. Then, enlightened from RadViz Deluxe (Cheng et al., 2017; 2018; Grinstein & Wierse, 2002), an effective visualization method to analyze the relationships between different features in data, we utilize the Hamiltonian cycle to present the mined knowledge in the extracted clustering probability distributions. We find that by mapping the anchors of different clusters on the circumference of a circle with Hamiltonian cycle, the similarities between the different clusters can be well presented. Concretely, to present the similarities between the different clusters, the anchors of different clusters are sorted on the optimal Hamiltonian cycle generated by the similarities between these clusters and mapped on the circumference of a circle by their orders. Based on the polar coordinates of the anchors on the circumference, a sample with a higher probability of a cluster will be mapped closer to the corresponding anchor. In this way, the data can be visualized by the following three aspects: (1) the samples with a high probability in the same cluster can be mapped together; (2) similar clusters can be mapped close to each other; (3) the samples with low probability to any cluster can be regarded as outliers and mapped far away from all clusters.

An example of our HCHC is shown in Appendix A and a head-to-head comparison between HCHC and Radviz Deluxe is shown in Appendix B. Compared with visualizing the deep features by existing embedding methods, such as t-SNE, our HCHC can better match the clustering result while presenting the outliers and similarities between the different clusters. Compared with the dendrogram in hierarchical clustering, our HCHC can better explain not only the recognized similar clusters by showing the in-between samples of different clusters but also the recognized outliers by showing their clustering probability distributions. We perform experiments on six real-world datasets and a COVID-19 dataset to illustrate the effectiveness of our HCHC. The source code can be downloaded from https://github.com/TianyiHuang2022.

2. Related Work

In this section, we review high-dimensional clustering and visualization. They are highly related to our HCHC.

2.1. High-dimensional Clustering

Clustering has been a long-standing problem in machine learning (Hartigan, 1975; Braun et al., 2022; Wang et al., 2022). There are many well-known clustering methods, such as k-means (MacQueen et al., 1967), DBSCAN (Ester et al., 1996), and Gaussian Mixture models (Rasmussen, 1999). Clustering can also be combined with other techniques like category discovery and semantic instance segmentation (Shi & Malik, 2000; Wang et al., 2021). However, clustering high-dimensional data is a hard issue, because of the large time complexity and the complex structure of data resulting from high-dimensional space (Aljalbout et al., 2018). Spectral clustering methods are often used to address this issue (Macgregor & Sun, 2022a; Ng et al., 2001; Von Luxburg, 2007; Bianchi et al., 2020; Macgregor & Sun, 2022b). It explores the manifold structure of high-dimensional data in a low-dimensional space by the eigenvectors of the Laplacian matrix from the corresponding similarity graph. Another important tool for clustering high-dimensional data is multitask clustering. It can handle high-dimensional data by exploiting the knowledge shared by related tasks, such as inter-task clustering correlation and intra-task learning correlation (Yang et al., 2014; Zhang et al., 2016). Unfortunately, most of the above clustering methods just generate the pseudo sample label, a binary choice of belonging to a cluster or not, and thus are unable to represent the other interesting knowledge in high-dimensional clusterings, such as the similarities between clusters and outliers.

With the advances in deep learning, combining neural networks into clustering tasks has drawn significant attention in the literature (Chang et al., 2017; Chen, 2015; Ji et al., 2019; Tian et al., 2014). Deep clustering can mine not only clusters but also similarities and outliers in high-dimensional data by learning the clustering probability distributions. We will give a brief introduction of the promising works in deep clustering including DEC, IDEC, deep spectral clustering, and data augmentation next (Xie et al., 2016; Guo et al., 2017; Shaham et al., 2018; Li et al., 2021).

DEC starts with pretraining a nonlinear mapping by an autoencoder and then removes the decoder. The remaining encoder is finetuned by minimizing the KL divergence. To enhance the local structure preservation, IDEC improves DEC by incorporating the autoencoder into DEC in the whole training process (Guo et al., 2017). By combining with deep learning, spectral clustering can work with large datasets (Shaham et al., 2018; Bianchi et al., 2020). In deep spectral clustering, the loss function is based on a weighted adjacency matrix associated with a similarity graph which includes the pairwise similarities between data points in each mini-batch. In this way, the time complexity of deep spectral clustering is mainly dependent on the size of the mini-batch. Recently, data augmentation is combined into deep image clustering and achieves great success, especially in contrastive learning (Li et al., 2021; Van Gansbeke et al., 2020; Niu et al., 2022). The augmentations of an image sample can be computed by the transformation functions, e.g., random rotation, shifting, and cropping (Guo et al., 2018; Park et al., 2021). Then the loss function for learning the relationship between this sample and its augmentation can be computed by the dissimilarity between their corresponding embedded features.

2.2. High-dimensional Visualization

High-dimensional visualization is the visual representation of the data with a large number of features (Grinstein et al., 2001). In this process, samples are mapped to numerical form and translated into a 2D graphical representation. Many embedding methods can map high-dimensional data in a 2D space for visualization. In this way, the important information from the features or the sample similarities in data can be well presented to us. These methods are also used to visualize the embedded features in deep clustering for presenting the mined knowledge (Xie et al., 2016; Huang et al., 2022).

MDS (Kruskal, 1978), PCA (Abdi & Williams, 2010), Isomap (Balasubramanian, 2002), t-SNE (Van der Maaten & Hinton, 2008) and UMAP (McInnes et al., 2018) are five commonly used embedding methods. MDS is a multivariate statistical method for estimating the scale values along one or more continuous dimensions such that those dimensions account for proximity measures defined over pairs of samples. PCA extracts the important information from the features in a set of new orthogonal variables called principal components to display the similarities between the samples in the data. Isomap extends classical multidimensional scaling by considering approximate geodesic distance instead of Euclidean distance. t-SNE is a variation of stochastic neighbour embedding (SNE) (Hinton & Roweis, 2002). Compared with SNE, t-SNE is much easier to optimize and produces significantly better visualizations by reducing the tendency to crowd points together in the centre of the map. UMAP is based on Riemannian geometry and algebraic topology. This method is competitive with t-SNE for visualization quality and arguably preserves more of the global structure with superior run time performance. However, most of the above embedding methods only map the data samples without considering the relations between the features.

Radviz can solve this problem by assigning the features to points called dimensional anchors placed on the circumference of a circle (Grinstein & Wierse, 2002). Original RadVis computes the polar coordinates of the anchors by a function of the effectiveness for discriminating each class of the samples in data (Mccarthy et al., 2004). The details of RadVis are in Appendix C. In unsupervised learning, the polar coordinates of the anchors can be computed by the similarities between the features (Sharko et al., 2008). Then the locations of the samples are determined by a weighting formula where sample features with higher values will receive a higher attraction to the corresponding anchors. As an improvement of RadViz, RadViz Deluxe sorts the feature anchors by Hamiltonian cycle and then maps the similarities between these features by adjusting the distances between the anchors on the circumference (Cheng et al., 2017). Thus the similarities between the different features can be better presented.

3. GLDC onto Hamiltonian Cycle

In this section, we propose HCHC to cluster high-dimensional data and then visualize the clustering results including clusters, cluster similarities, and outliers. The details of the motivation of HCHC are shown in Appendix D. The architecture of HCHC is shown in Fig. 1. The dataset $\boldsymbol{X}$ and its augmentation $\widetilde{\boldsymbol{X}}$ are the inputs of our GLDC network. In the pretraining of our GLDC, the autoencoder is trained by minimizing the reconstruction loss to encode $\boldsymbol{X}$ in $\boldsymbol{Z}$. After pretraining, GLDC computes the clustering probability distribution of each sample in $\boldsymbol{Z}$ by minimizing our clustering loss which includes the reconstruction loss, the graph learning loss, and the augmentation learning loss. Finally, the clustering probability distributions from GLDC is visualized based on a Hamiltonian cycle.

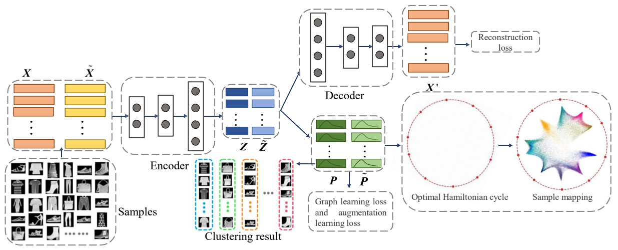


Figure 1: The architecture of our HCHC.

3.1. Combining Global-structure with Local-structure

In this subsection, we propose a new deep clustering method, GLDC, by incorporating global-structure with local-structure in one objective function. At the beginning of GLDC, self-training is used as the pretraining by minimizing the reconstruction loss of data. In this way, we can capture the most salient features (Guo et al., 2017; Lin, 2007).

For a given dataset $\boldsymbol{X} = \{\boldsymbol{x_1}; \boldsymbol{x_2}; \cdots; \boldsymbol{x_n}\}$, an encoder G_θ, and the corresponding decoder $G_{\theta'}$, the reconstruction loss in the pretraining can be written as

$$L_r = \sum_i^n ||\boldsymbol{x_i} - G_{\theta'}(\boldsymbol{z_i})||_2^2 = \sum_i^n ||\boldsymbol{x_i} - G_{\theta'}(G_\theta(\boldsymbol{x_i}))||_2^2 \quad (1)$$

After the pretraining, in the space of $\boldsymbol{Z}$ of the l-th mini-batch, the network is optimized by a weighted adjacency graph which is associated with a weighted adjacency matrix $\boldsymbol{W^l}$. Each entry in $\boldsymbol{W^l}$ is defined as

$$w_{i,j}^l = \begin{cases} e^{-\frac{||\boldsymbol{z_i}-\boldsymbol{z_j}||_2^2}{\sigma^2}} & \boldsymbol{x_j} \in \mathcal{N}_i^k \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where $\mathcal{N}_i^K$ is the set of k-nearest neighbors of $\boldsymbol{z_i}$ in the mini-batch. Different from the existing deep spectral clustering where merely the local-structure of the data is represented in $\boldsymbol{Z}$, our GLDC represents the local-structure and the global-structure of data in the output $\boldsymbol{p_i}$, the learned probability distribution of $\boldsymbol{x_i}$ for different clusters. In our network, we calculate the $\boldsymbol{p_i} = \{p_{i,1}, ..., p_{i,c}\}$ by

$$\boldsymbol{p_i} = \mathcal{P}_\phi(\boldsymbol{z_i}) = \mathcal{P}_\phi(G_\theta(\boldsymbol{x_i})) \quad (3)$$

Then the graph learning loss based on $\boldsymbol{W^l}$ in the mini-batch is defined as (Rebuffi et al., 2021).

$$L_w = -\frac{1}{B^2}(\sum_{i,j}^B w_{i,j}^l \log P(i=j) + (1-w_{i,j}^l)\log P(i \neq j)) \quad (4)$$

where $P(i = j)$ is the probability that $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ belong to the same cluster while $P(i \neq j)$ is the probability that $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ belong to the different clusters. The item $w_{i,j}^l \log P(i = j)$ is to make the connected samples in the adjacency matrix have similar clustering probability distributions and thus can keep the local-structure of data in the same cluster. $(1 - w_{i,j}^l) \log P(i \neq j)$ is to make the unconnected samples have diverse clustering probability distributions, i.e., far points should be in different clusters, and thus can be used to analyze the global-structure of the data. Therefore, in our GLDC clustering, the local structure of data and the similarities between different clusters can be considered together and presented in our visualization. Because

$$\begin{aligned} P(i=j) &= \sum_{h=1}^c P(i=h, j=h) \\ &= \sum_{h=1}^c p_{i,h} \times p_{j,h} \\ &= \boldsymbol{p_i^\intercal p_j}, \end{aligned} \quad (5)$$

where c is the cluster number, L_w can be written as (Rebuffi et al., 2021)

$$L_w = -\frac{1}{B^2}(\sum_{i,j}^B w_{i,j}^l \log \boldsymbol{p_i^\intercal p_j} + (1-w_{i,j}^l)\log(1-\boldsymbol{p_i^\intercal p_j})) \quad (6)$$

We also use a generalized data augmentation method to improve our clustering. In our GLDC, $T(\boldsymbol{x_i})$ is defined as

$$\widetilde{\boldsymbol{x_i}} = T(\boldsymbol{x_i}) = \boldsymbol{x_i} + \epsilon, \epsilon \sim \mathcal{N}(0,\xi) \quad (7)$$

where $\mathcal{N}(0,\xi)$ is a Gaussian distribution. Define $\widetilde{\boldsymbol{p_i}}$ as the output of the probability of $\widetilde{\boldsymbol{x_i}}$, then the loss of the data augmentation learning can be defined as

$$L_a = \sum_i ||\boldsymbol{p_i} - \widetilde{\boldsymbol{p_i}}||_2^2 \quad (8)$$

The overall clustering loss is given by

$$L_{clu} = L_r + \beta_1 L_w + \beta_2 L_a \quad (9)$$

The algorithm of GLDC is summarized in Appendix E. For the input dataset $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_n}\}$, in this algorithm we can get the distributions $\boldsymbol{P} = \{\boldsymbol{p_1}, \boldsymbol{p_2}, \cdots, \boldsymbol{p_n}\}$, where $p_{i,j}$ is the the probability that $\boldsymbol{x_i}$ belongs to cluster j. Based on $\boldsymbol{p_i}$, the label c_i assigned to $\boldsymbol{x_i}$ can be obtained by

$$c_i = \arg\max_j p_{i,j} \quad (10)$$

3.2. Mapping the Distributions by the Optimal Hamiltonian Cycle

After the deep clustering, we can get the distribution matrix $\boldsymbol{P} = \{\boldsymbol{\rho_1}; \boldsymbol{\rho_2}; \cdots; \boldsymbol{\rho_c}\}$, where c is the number of the clusters and $\boldsymbol{\rho_i} = \{p_{1,i}, p_{2,i}, \cdots, p_{n,i}\}$. Then we show the clustering results in $\boldsymbol{P}$ by the optimal Hamiltonian cycle of the different clusters. This is the shortest cycle that passes through every $\boldsymbol{\rho_i}$ exactly once, except the first passed one (Dirac, 1952). The mapping process is shown in Fig. 2 and detailed as follows.

First, we use Pearson correlation coefficient to compute the similarities between $\boldsymbol{\rho_i}$ and $\boldsymbol{\rho_j}$ as

$$s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}) = \frac{\sum_{l=1}^n (p_{l,i} - \overline{\boldsymbol{\rho_i}})(p_{l,j} - \overline{\boldsymbol{\rho_j}})}{\sqrt{\sum_{l=1}^n (p_{l,i} - \overline{\boldsymbol{\rho_i}})^2}\sqrt{\sum_{l=1}^n (p_{l,j} - \overline{\boldsymbol{\rho_j}})^2}} \quad (11)$$

Thus, the dissimilarities between the $\boldsymbol{\rho_i}$ and $\boldsymbol{\rho_j}$ can be defined as

$$dis(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}) = \frac{1 - s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})}{\sum_{i=1}^{c-1}\sum_{j=i+1}^{c}(1 - s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}))} \tag{12}$$

Secondly, based on the above dissimilarities, $\{\boldsymbol{\rho_1}; \boldsymbol{\rho_2}; \cdots; \boldsymbol{\rho_c}\}$ are ordered by an optimal Hamiltonian cycle. We use dynamic programming to get the optimal Hamiltonian cycle by the following definition.

Definition 3.1. Define $dp(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}, \boldsymbol{state})$ as the minimum path between $\boldsymbol{\rho_i}$ and $\boldsymbol{\rho_j}$ with the path state $\boldsymbol{state} = \{pass(\boldsymbol{\rho_c}), \cdots pass(\boldsymbol{\rho_2}), pass(\boldsymbol{\rho_1})\}$, where if $\boldsymbol{\rho_i}$ has been passed, $pass(\boldsymbol{\rho_i}) = 1$, otherwise, $pass(\boldsymbol{\rho_i}) = 0$.

In the dynamic programming, $dp(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}, \boldsymbol{state})$ can be updated by

$$\begin{aligned} dp(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}, \boldsymbol{state}) = \min\{&dp(\boldsymbol{\rho_i}, \boldsymbol{\rho_k}, \boldsymbol{state} \\ &\oplus (1 << (j-1)) + dis(\boldsymbol{\rho_k}, \boldsymbol{\rho_j})\} \end{aligned} \tag{13}$$

where $\oplus$ is XOR operation and $<<$ is shift-arithmetic-left operation. Then we can get $\Pi^* = \{\boldsymbol{\rho^1}; \boldsymbol{\rho^2}; \cdots; \boldsymbol{\rho^c}\}$ as the optimal Hamiltonian cycle of each cluster by the following definition.

Definition 3.2. Let $\boldsymbol{\rho^i} \in \{\boldsymbol{\rho_1}, \cdots, \boldsymbol{\rho_c}\}$ be the i-th passed vertice of a Hamiltonian cycle. Then this Hamiltonian cycle can be defined as $\Pi = \{\boldsymbol{\rho^1}; \boldsymbol{\rho^2}; \cdots; \boldsymbol{\rho^c}\}$.

The angle $\alpha_{\boldsymbol{\rho^i}}$ of $\boldsymbol{\rho^i}$ is used to map the anchors of different clusters on a circle and can be computed by

$$\alpha_{\boldsymbol{\rho^i}} = \begin{cases} 0, & i = 1 \\ \alpha_{\boldsymbol{\rho^{i-1}}} + & \\ 2\pi \frac{dis(\boldsymbol{\rho^i}, \boldsymbol{\rho^{i-1}})}{\sum_{j=2}^{c} dis(\boldsymbol{\rho^j}, \boldsymbol{\rho^{j-1}}) + dis(\boldsymbol{\rho^c}, \boldsymbol{\rho^1})} & \text{otherwise} \end{cases} \tag{14}$$

Thirdly, based on $\alpha_{\boldsymbol{\rho^i}}$ the position of the anchor of $\boldsymbol{\rho^i}$ on a circle can be computed by

$$\boldsymbol{\mu_{p^i}} = [r \times \cos(\alpha_{\boldsymbol{\rho^i}}), r \times \sin(\alpha_{\boldsymbol{\rho^i}})] \tag{15}$$

Finally, the position of $\boldsymbol{x_i}$ in the circle can be computed by

$$\boldsymbol{\mu_{x_i}} = \sum_{j=1}^{c} \frac{p_{i,j}}{\sum_{k=1}^{c} p_{i,k}} \boldsymbol{\mu_{\rho^j}} = \sum_{j=1}^{c} p_{i,j} \boldsymbol{\mu_{\rho^j}} \tag{16}$$

with $\sum_{k=1}^{c} p_{i,k} = 1$. A Hamiltonian cycle can present the cluster similarities $\{s(\boldsymbol{\rho^1}, \boldsymbol{\rho^2}), \cdots, s(\boldsymbol{\rho^{c-1}}, \boldsymbol{\rho^c}), s(\boldsymbol{\rho^1}, \boldsymbol{\rho^c})\} \in \{s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}) | i = 1, \cdots, c, i < j\}$ and based on the following theorem, the optimal Hamiltonian cycle will select high similarities between the clusters.

Theorem 3.3. *For the cluster similarities* $\{s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}) | i = 1, \cdots, c-1, i < j\}$, *the mapping of the optimal Hamiltonian cycle will maximize*

$$S_{sam} = \sum_{i=1}^{c-1} s(\boldsymbol{\rho^i}, \boldsymbol{\rho^{i+1}}) + s(\boldsymbol{\rho^1}, \boldsymbol{\rho^c}) \tag{17}$$

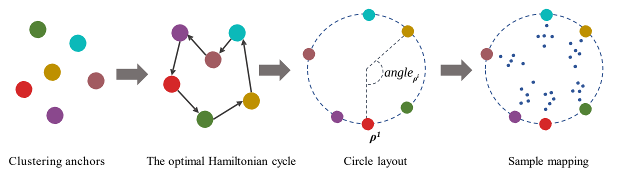


Figure 2: Illustration of our mapping process by optimal Hamiltonian cycle.

Therefore, by the optimal Hamiltonian cycle, we can get the global optimal path to make similar clusters close to each other in the circumference of a circle. In this way, the similarity between $\boldsymbol{\rho^i}$ and $\boldsymbol{\rho^{i+k(k>1)}}$ can be measured by the corresponding geodesic distance on this optimal Hamiltonian cycle. The algorithm of our mapping is presented in Appendix E The theoretical analysis to proof theorem 3.3 is in Appendix F.1. The theoretical analysis in Appendix F.2 illustrates that the optimal Hamiltonian cycle can improve the sample mapping based on the following assumption. The further away a point is from the center, the more informative its position is, being the point closer to the attributes having the highest values (Angelini et al., 2019).

4. Experimental Results

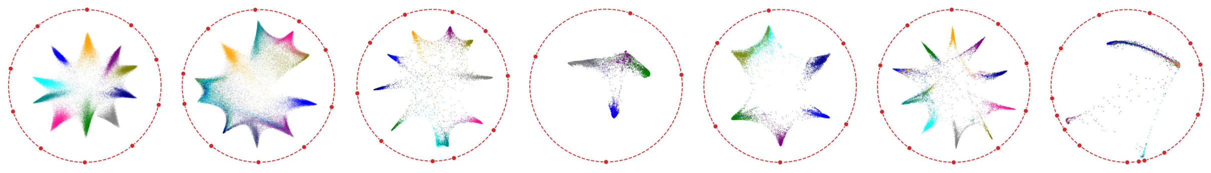
In this section, first, we show the visualized results of our HCHC and other visualization methods. Then, we compare our GLDC with different clustering methods. Finally, we analyze a dataset of COVID-19 by HCHC. The experimental setting is shown in Appendix G. The time cost analysis and case study are shown in Appendix H.2 and H.4, respectively. The parameter analysis is shown in Appendix H.5.

4.1. Visualized Result

The visualized results of HCHC with MNIST (Deng, 2012), Fashion (Xiao et al., 2017), USPS (Hull, 1994), Reuters10k (Lewis et al., 2004), HHAR (Stisen et al., 2015), Pendigits (ASUNCION, 2007), and BH (Abdelaal et al., 2019) are shown in Fig. 3.

From Fig. 3 (a), we can see that the different classes in MNIST are well separated by the clustering in GLDC. The visualized result of the samples is consistent with their labels. By HCHC, we can also see the similarities between the different classes such as that the cyan class and the dark cyan class are closer to each other than other pairs of classes.

From Fig. 3 (b), we can see that the outliers in Fashion are much more than those in MNIST. The orange class and blue class can be well separated from the others. The similarities between the different clusters can also be shown in Fashion by HCHC. Then we can see that the samples in the dark-blue class, green class, and cyan class are easily mixed in

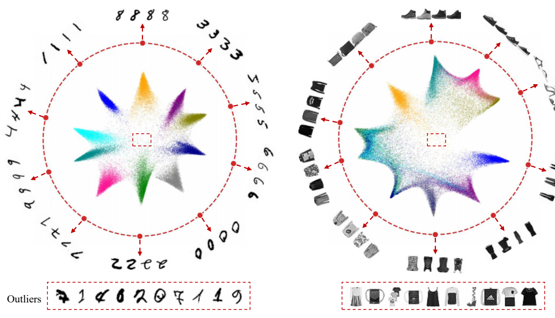


(a) MNIST (b) Fashion (c) USPS (d) Reuters10k (e) HHAR (f) Pendigit (g) BH

Figure 3: Visualized results on different datasets.

Table 1: ACCs and NMIs of different clustering methods.

Method	MNIST		Fashion		USPS		Reuter10K		HHAR		Pendigits		BH	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
k-means	0.532	0.500	0.474	0.512	0.668	0.601	0.559	0.375	0.599	0.588	0.666	0.681	0.492	0.561
GMM	0.439	0.356	0.476	0.532	0.553	0.529	0.665	0.430	0.597	0.593	0.673	0.682	0.593	0.656
SC	0.680	0.759	0.551	0.630	0.712	0.656	0.658	0.401	0.538	0.741	0.724	0.784	0.554	0.601
DEC	0.863	0.834	0.518	0.546	0.762	0.767	0.773	0.528	0.764	0.700	0.776	0.706	0.648	0.618
IDEC	0.881	0.867	0.529	0.557	0.761	0.785	0.785	0.541	0.722	0.785	0.793	0.742	0.406	0.548
DSC	0.938	0.873	0.633	0.647	0.866	0.859	0.725	0.472	0.713	0.764	0.820	0.791	0.607	0.492
JULE	0.964	0.913	0.563	0.608	0.950	0.913	-	-	-	-	-	-	-	-
DSCDAN	0.978	0.941	0.662	0.645	0.869	0.857	-	-	-	-	-	-	-	-
N2D	0.979	0.942	0.672	0.684	0.958	0.901	0.784	0.536	0.801	0.783	0.885	0.863	0.554	0.570
GLDC	0.979	0.941	0.715	0.691	0.910	0.862	0.834	0.629	0.878	0.821	0.867	0.830	0.704	0.655
GLDC(w/o L_a)	0.965	0.914	0.694	0.652	0.879	0.817	0.804	0.587	0.822	0.743	0.803	0.744	0.687	0.614
GLDC(w/o L_r)	0.975	0.932	0.621	0.635	0.825	0.787	0.724	0.506	0.765	0.723	0.781	0.768	0.680	0.591



(a) MNIST (b) Fashion

Figure 4: The clusters, samples, labels, and outliers on MNIST and Fashion, respectively.

clustering. This is a piece of important information for us to improve the clustering for Fashion dataset.

From Fig. 3 (c), we can see that as the visualized result of MNIST, the different clusters in USPS are well separated. However, some samples in the cyan class and the dark cyan class are easily mixed.

From Fig. 3 (d), we can see that for Reuters10k, the blue class, the green class, and the grey class are well separated by GLDC. However, the samples in the purple class are mixed with the samples in the green class and the grey class.

From Fig. 3 (e), we can see that most of the classes in HHAR are well separated by GLDC. However, some samples in the cyan class and olive class are easily mixed.

From Fig. 3 (f), we can see that most of the classes in Pendigits are also well separated by GLDC. However, some samples in the dark cyan class and olive class are easily mixed while some samples in the orange class and blue class are also easily mixed.

From Fig. 3 (g), we can see that BH is a challenging dataset. GLDC can correctly cluster most of the samples in the purple class, cycan class, and orange class.

Overall, the above results illustrate that HCHC can well mine the clusters, similarities, and outliers in real-world datasets. In most cases, the mined result of a dataset can be highly related to the corresponding labels.

Fig. 4 (a) and (b) show the clustering results and the visualizations of Fashion and MNIST by HCHC, respectively. In this Figure, we show the randomly selected 4 images of each cluster and the recognized outliers. As we can see, by HCHC the similar clusters are close to each other and the outliers can be well recognized. For example, as shown in Fig. 4 (a) the shapes of digit numbers "9" and "4" are similar, so their clusters are mapped close to each other. HCHC also can recognize the illegible handwritten digits as the outliers. As shown in Fig. 4 (b) for Fashion, HCHC can put the clusters of different types of shoes together, it also can recognize the outliers in the data.

We compare the different visualization methods on our embedded features of MNIST and Fashion, respectively. The

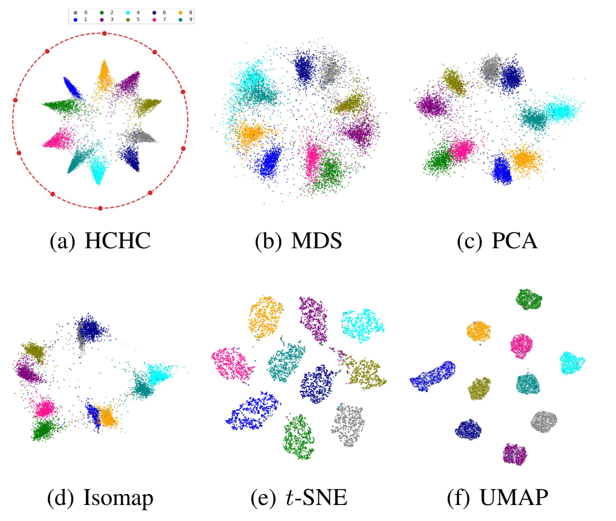


Figure 5: Different visualization methods on MNIST-test.

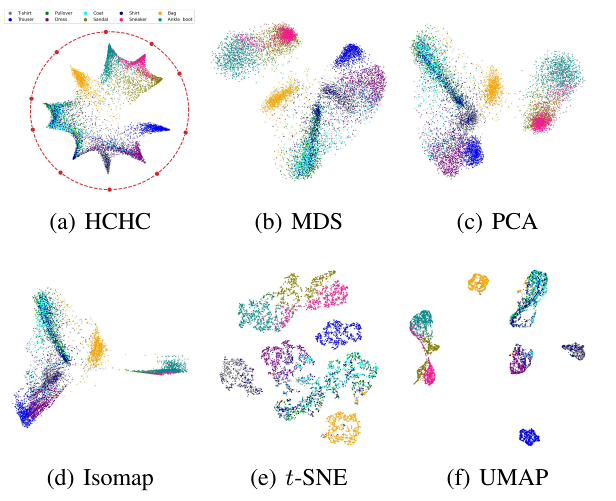


Figure 6: Different visualization methods on Fashion-test.

results are shown in Fig. 5 and 6. For MNIST, our HCHC can better visualize the cluster number and index cluster result than MDS, PCA, and Isomap. Compared with t-SNE and UMAP, HCHC can better present the cluster similarities and outliers. For Fashion, with the help of the anchors, our HCHC can better visualize the number of the clusters and index cluster result than any other visualization methods. HCHC also can better present the cluster similarities and outliers than t-SNE and UMAP for Fashion. The visualized results of the other datasets are shown in Appendix H.3.

4.2. The Comparison of Different Clustering Methods

In this subsection, we compared our GLDC with existing clustering methods in both the numerical and visualization cases. We use ACC and NMI to measure the clustering performance of GLDC with nine existing clustering methods (Kuhn, 1955). These methods are k-means (MacQueen et al., 1967), GMM (Rasmussen, 1999), SC (Shi & Malik, 2000), DEC (Xie et al., 2016), IDEC (Guo et al., 2017), DSC (Shaham et al., 2018), JULE (Yang et al., 2016), DSC-DAN (Yang et al., 2019), and N2D (McConville et al., 2021). The ACCs and NMIs of different clustering methods are shown in Table 1. As we can see, our GLDC has the best ACCs in clustering tasks with MNIST, Fashion, Reuters-10k, HHAR, and BH. Our GLDC also has the best NMIs with Fashion, Reuters-10k, and HHAR. Moreover, we perform the ablation study with L_a and L_r in our overall clustering loss. As we can see in Table 1, both L_a and L_r can well improve the clustering performance of GLDC.

In HCHC, a high clustering performance in ACC or MNI is not enough for satisfactory visualization. The similarities between the clusters and outliers also need to be presented. Compared with the existing deep clustering methods, our GLDC can better analyze the similarities between the different clusters by considering the global-structure in data and thus get a better visualized result. We compare GLDC with three universal deep clustering methods including DEC, IDEC, and DSC in the framework of HCHC. The results are shown in Fig. 7.

As we can see, it is hard to see the similarities between the different clusters by DEC, IDEC, and DSC, However, our GLDC can well show the similarities between different clusters by considering the global-structure of data in the generated distributions. For example, by GLDC in HCHC we can find that (1) in MNIST the blue class is not similar to other classes and thus can be well discovered in the clustering task. (2) in Reuters-10k, the purple class is similar to the grey class and the green class, thus these three classes are easily mixed in the clustering task. To this end, we need to analyze the possibilities of a non-outlier sample belonging to different clusters. Thus even if a non-outlier sample $\boldsymbol{x_i}$ does not belong to cluster j, $p_{i,j}$ may not tend to 0. In this case, the max value in $\boldsymbol{p_i}$ may not tend to 1 with $\sum_{j=1}^{c} p_{i,j} = 1$. Therefore, the mapped samples may not be very close to the anchor of the corresponding cluster on the red circle.

We also can see the advantages of GLDC in visualized comparisons in Fig. 7. Concretely, we have the following observations. (1) For the MNIST dataset, DEC can not well recognize the dark cyan class and orange class, IDEC can not well recognize the dark cyan class and olive class, DSC cannot cluster the green class very well, however, GLDC can well recognize and cluster all of the classes. (2) GLDC can well recognize most classes in Fashion, but DEC, IDEC, and DSC cannot do so. (3) For the USPS dataset, DEC divides the dark cyan class into all the clusters, IDEC mistakenly clusters the purple class and olive class together and cannot

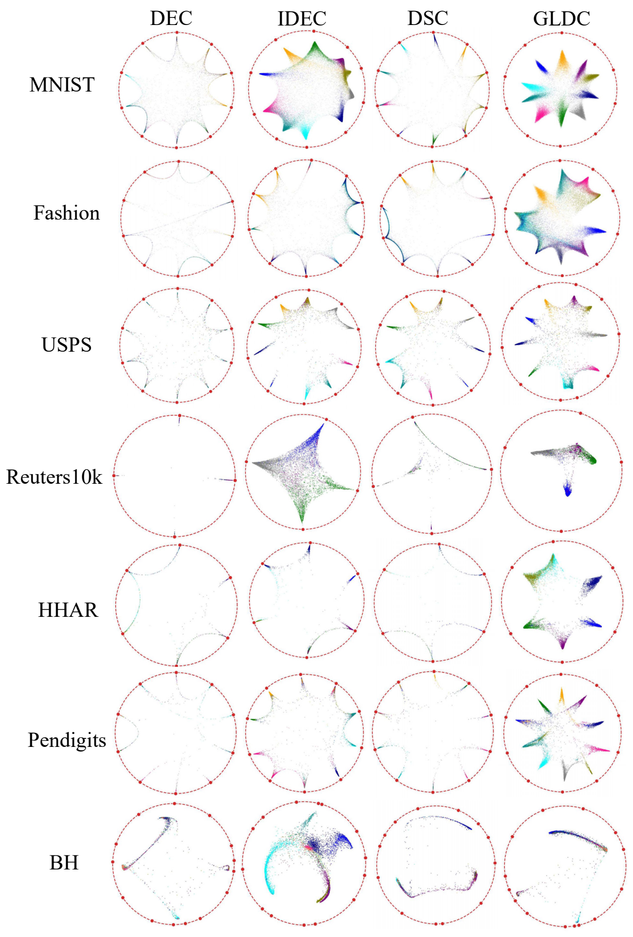
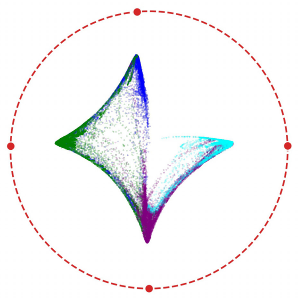
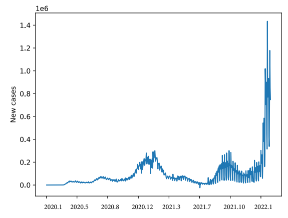


Figure 7: Different deep clustering methods on optimal Hamiltonian cycle.

well recognize the grey class, DSC also mistakenly cluster the purple class and olive class together, however, GLDC can well recognize and cluster all of the classes. (4) For the Reuters10k dataset, DEC cannot well recognize most classes, IDEC mistakenly clusters the purple class and blue class together and divide the samples in the green class into two clusters, DSC mistakenly mixes the samples in the green class and purple class, however, GLDC can well recognize most of the classes, i.e., the green class, blue class, and grey class. (5) For the HHAR dataset, DEC cannot well recognize most classes, IDEC mistakenly clusters the cyan class and olive class together and divides the samples in the dark blue class into two clusters, DSC mistakenly divides the samples in the dark blue class into two clusters, however, GLDC can well recognize most of the classes. (6) For the Pendigits dataset, DEC, IDEC, and DSC cannot recognize the pink class and dark cyan class, however, GLDC can well recognize these two classes. (7) For the BH dataset, DEC and GLDC can better cluster the samples in the cyan and purple class than IDEC and DSC.



(a) Labeled by every 6 months



(b) Daily case

Figure 8: The results of HCHC on the COVID-19 dataset.

4.3. The Result on the COVID-19 Dataset

The above experiment illustrates the effectiveness of HCHC to discover the knowledge, i.e., clusters, similarities, and outliers, in real-world datasets. Then, in this subsection, we use HCHC to learn the temporal features of COVID-19 by a dataset that includes the available COVID-19 daily information of the different states in the USA from 21/1/2020 to 21/1/2022. The daily information includes daily cases, daily deaths, and so on. There are 37525 samples in this dataset and their features are detailed in Appendix G.

In Fig. 8 (a), where the data are labelled by every 6 months, the samples from the same period can be mapped close to the same cluster anchor. The results of the data labelled by every 4 month, every 5 month, and every 8 month, are shown in Appendix H.1. The total daily cases in the USA from 21/1/2020 to 21/1/2022 are shown in Fig. 8 (b). Combining Fig. 8 (a) and (b), we can see that after 2020.8 the period of COVID-19 can be around 6 months, i.e., the number of daily cases increased from autumn to winter and decreased from spring to summer. This phenomenon may illustrate that there are more people infected with COVID-19 in the low-temperature environment than in the high-temperature environment.

5. Conclusion

This paper proposes HCHC to cluster high-dimensional data, and then visualize the clustering result. It combines global structure with local structure in one objective function, improving the labels as relative probabilities, to mine the similarities between different clusters while keeping the local structure in each cluster. Then, the anchors of different clusters are sorted on the optimal Hamiltonian cycle generated by the cluster similarities and mapped on the circumference of a circle. Finally, a sample with a higher probability of a cluster will be mapped closer to the corresponding anchor. In this way, HCHC allows us to appreciate three aspects at the same time - (1) cluster recognition, (2) cluster similarities, and (3) outlier recognition. The experimental result shows the effectiveness of HCHC. As shown in our experiment, HCHC also can serve as a visualized clustering measure by mapping the labelled samples. In this way, we can find the reasons for an unsatisfactory clustering result.

In further research, we will investigate how to use HCHC to solve the multi-cluster problem. Because we rely on an NP-hard optimization step, the optimal Hamiltonian cycle to sort the clusters, it is necessary to give an approximate solution in acceptable time consumption as shown in Appendix F.1. Mapping too many clusters in 2D space may cause high mapping error. Therefore we can investigate how to map these clusters in 3D space.

Acknowledgements

The authors thank partial supports from NSF-DMS-2012298, NNSFC-71991471, 11931015, 71974176.

References

Abdelaal, T., Michielsen, L., Cats, D., Hoogduin, D., Mei, H., Reinders, M. J., and Mahfouz, A. A comparison of automatic cell identification methods for single-cell rna sequencing data. *Genome biology*, 20:1–19, 2019.

Abdi, H. and Williams, L. J. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

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

Angelini, M., Blasilli, G., Lenti, S., Palleschi, A., and Santucci, G. Towards enhancing radviz analysis and interpretation. In *2019 IEEE Visualization Conference (VIS)*, pp. 226–230. IEEE, 2019.

ASUNCION, A. Uci machine learning repository. *http://www. ics. uci. edu/˜ mlearn/MLRepository. html*, 2007.

Balasubramanian, M. The isomap algorithm and topological stability. *Science*, 295(7a), 2002.

Bianchi, F. M., Grattarola, D., and Alippi, C. Spectral clustering with graph neural networks for graph pooling. In *International Conference on Machine Learning*, pp. 874–883. PMLR, 2020.

Braun, G., Tyagi, H., and Biernacki, C. An iterative clustering algorithm for the contextual stochastic block model with optimality guarantees. In *International Conference on Machine Learning*, pp. 2257–2291. PMLR, 2022.

Bühlmann, P. and Van De Geer, S. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

Chang, J., Wang, L., Meng, G., Xiang, S., and Pan, C. Deep adaptive image clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5879–5887, 2017.

Chen, G. Deep learning with nonparametric clustering. *arXiv preprint arXiv:1501.03084*, 2015.

Cheng, S., Xu, W., and Mueller, K. Radviz deluxe: An attribute-aware display for multivariate data. *Processes*, 5(4):75, 2017.

Cheng, S., Xu, W., and Mueller, K. Colormap nd: A data-driven approach and tool for mapping multivariate data to color. *IEEE Transactions on Visualization and Computer Graphics*, 25(2):1361–1377, 2018.

Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Dirac, G. A. Some theorems on abstract graphs. *Proceedings of the London Mathematical Society*, 3(1):69–81, 1952.

Donoho, D. L. et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, 1(2000):32, 2000.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pp. 226–231, 1996.

Fortunato, S. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

Grinstein, G., Trutschl, M., and Cvek, U. High-dimensional visualizations. In *Proceedings of the Visual Data Mining Workshop, KDD*, volume 2, pp. 120, 2001.

Grinstein, U. M. F. G. G. and Wierse, A. *Information visualization in data mining and knowledge discovery*. Morgan Kaufmann, 2002.

Guha, S., Rastogi, R., and Shim, K. Cure: An efficient clustering algorithm for large databases. *ACM Sigmod Record*, 27(2):73–84, 1998.

Guo, X., Gao, L., Liu, X., and Yin, J. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pp. 1753–1759, 2017.

Guo, X., Zhu, E., Liu, X., and Yin, J. Deep embedded clustering with data augmentation. In *Asian Conference on Machine Learning*, pp. 550–565. PMLR, 2018.

Hartigan, J. A. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

Hinton, G. E. and Roweis, S. Stochastic neighbor embedding. *Advances in Neural Information Processing Systems*, 15, 2002.

Hoogeveen, J. Analysis of christofides' heuristic: Some paths are more difficult than cycles. *Operations Research Letters*, 10(5):291–295, 1991.

Huang, T., Wang, S., and Zhu, W. An adaptive kernelized rank-order distance for clustering non-spherical data with high noise. *Int. J. Mach. Learn. Cybern.*, 11(8):1735–1747, 2020.

Huang, T., Cai, Z., Li, R., Wang, S., and Zhu, W. Consolidation of structure of high noise data by a new noise index and reinforcement learning. *Information Sciences*, 614: 206–222, 2022.

Hull, J. J. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323, 1999.

Ji, X., Henriques, J. F., and Vedaldi, A. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9865–9874, 2019.

Karypis, G., Han, E.-H., and Kumar, V. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32 (8):68–75, 1999.

Kruskal, J. B. *Multidimensional scaling*. Sage, 1978.

Kuhn, H. W. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2): 83–97, 1955.

Lewis, D. D., Yang, Y., Russell-Rose, T., and Li, F. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004.

Li, S. Z., Wu, L., and Zang, Z. Consistent representation learning for high dimensional data analysis. *arXiv preprint arXiv:2012.00481*, 2020.

Li, Y., Hu, P., Liu, Z., Peng, D., Zhou, J. T., and Peng, X. Contrastive clustering. In *2021 AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Lin, C.-J. *Large-scale kernel machines*. MIT Press, 2007.

Macgregor, P. and Sun, H. A tighter analysis of spectral clustering, and beyond. In *International Conference on Machine Learning*, pp. 14717–14742. PMLR, 2022a.

Macgregor, P. and Sun, H. A tighter analysis of spectral clustering, and beyond. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 14717–14742. PMLR, 2022b.

MacQueen, J. et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 281–297. Oakland, CA, USA, 1967.

Mccarthy, J. F., Marx, K. A., Hoffman, P. E., Gee, A. G., O'neil, P., Ujwal, M. L., and Hotchkiss, J. Applications of machine learning and high-dimensional visualization in cancer detection, diagnosis, and management. *Annals of the New York Academy of Sciences*, 1020(1):239–262, 2004.

McConville, R., Santos-Rodriguez, R., Piechocki, R. J., and Craddock, I. N2d:(not too) deep clustering via clustering the local manifold of an autoencoded embedding. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 5145–5152. IEEE, 2021.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Ng, A., Jordan, M., and Weiss, Y. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14, 2001.

Nielsen, F. Hierarchical clustering. In *Introduction to HPC with MPI for Data Science*, pp. 195–211. Springer, 2016.

Niu, C., Shan, H., and Wang, G. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31:7264–7278, 2022.

Park, S., Han, S., Kim, S., Kim, D., Park, S., Hong, S., and Cha, M. Improving unsupervised image clustering with robust learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12278–12287, 2021.

Rani^1, Y. and Rohil, H. A study of hierarchical clustering algorithm. *ter S & on Te SIT*, 2:113, 2013.

Rasmussen, C. The infinite gaussian mixture model. *Advances in Neural Information Processing Systems*, 12, 1999.

Rebuffi, S.-A., Ehrhardt, S., Han, K., Vedaldi, A., and Zisserman, A. Lsd-c: Linearly separable deep clusters. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1038–1046, 2021.

Shaham, U., Stanton, K., Li, H., Nadler, B., Basri, R., and Kluger, Y. Spectralnet: Spectral clustering using deep neural networks. In *6th International Conference on Learning Representations, ICLR 2018*. The Weizmann Institute of Science, 2018.

Sharko, J., Grinstein, G., and Marx, K. A. Vectorized radviz and its application to multiple cluster datasets. *IEEE transactions on Visualization and Computer Graphics*, 14(6):1444–1427, 2008.

Shi, J. and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjærgaard, M. B., Dey, A., Sonne, T., and Jensen, M. M. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pp. 127–140, 2015.

Tian, F., Gao, B., Cui, Q., Chen, E., and Liu, T.-Y. Learning deep representations for graph clustering. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1293–1299, 2014.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., and Van Gool, L. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pp. 268–285. Springer, 2020.

Verleysen, M. et al. Learning high-dimensional data. *Nato Science Series Sub Series III Computer And Systems Sciences*, 186:141–162, 2003.

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

Wang, J., Ma, Z., Nie, F., and Li, X. Progressive self-supervised clustering with novel category discovery. *IEEE Transactions on Cybernetics*, 2021.

Wang, P., Liu, H., So, A. M.-C., and Balzano, L. Convergence and recovery guarantees of the k-subspaces method for subspace clustering. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, pp. 22884–22918. PMLR, 2022.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xie, J., Girshick, R., and Farhadi, A. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pp. 478–487. PMLR, 2016.

Yang, J., Parikh, D., and Batra, D. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5147–5156, 2016.

Yang, X., Deng, C., Zheng, F., Yan, J., and Liu, W. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4066–4075, 2019.

Yang, Y., Ma, Z., Yang, Y., Nie, F., and Shen, H. T. Multitask spectral clustering by exploring intertask correlation. *IEEE Transactions on Cybernetics*, 45(5):1083–1094, 2014.

Zang, Z., Cheng, S., Lu, L., Xia, H., Li, L., Sun, Y., Xu, Y., Shang, L., Sun, B., and Li, S. Z. Evnet: An explainable deep network for dimension reduction. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

Zhang, X., Zhang, X., and Liu, H. Self-adapted multi-task clustering. In *IJCAI*, pp. 2357–2363, 2016.

Zhao, Y., Luo, F., Chen, M., Wang, Y., Xia, J., Zhou, F., Wang, Y., Chen, Y., and Chen, W. Evaluating multidimensional visualizations for understanding fuzzy clusters. *IEEE transactions on visualization and computer graphics*, 25(1):12–21, 2018.

Zu, X. and Tao, Q. SpaceMAP: Visualizing high-dimensional data by space expansion. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 27707–27723. PMLR, 2022.

A. An example of HCHC

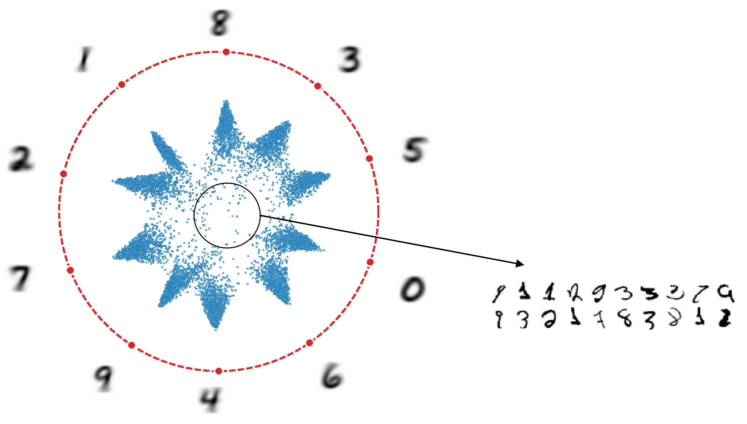


Figure 9: An example on MNIST-test

Fig.1 is an example to show the visualized result on a handwriting digit number dataset, MNIST-test, by our HCHC. In this figure, the layout of different clusters is on the circumference of a red circle by the optimal Hamiltonian cycle. Each red dot on the circumference is the anchor of a cluster, and the blue dots in the circle are mapped samples. If a mapped sample is close to a red dot, this sample will have a high probability of belonging to the corresponding cluster. The pictures around the red circle are the means of the different clusters. As we can see, HCHC well mines and represents the local similarities between the samples and the global similarities between the different clusters. Concretely, the samples of each class in the MNIST-test can be mapped together, and thus the mean of each cluster can be a corresponding digit number. The more similar clusters can be mapped closer to each other, e.g., the shapes of digit numbers “9” and “4” are similar, so their clusters are mapped close to each other. Any sample in the centre of the circle will have a low possibility to belong to any cluster. Such samples are usually not well-written and can be regarded as outliers.

B. The comparison between HCHC and Radviz Deluxe

To clarify the difference between our new HCHC and the existing Radviz Deluxe, we offer a direct comparison in the following table.

Table 2: The Comparison between HCHC and Radviz Deluxe

Method	Hamiltonian cycle	Deep clustering	Anchor	Visualization	Outlier mapping
Radviz Deluxe	Yes	No	Data features	Feature values	Apart from clusters
HCHC	Yes	Yes	Cluster labels	Clustering result	Near the centre

Although both HCHC and Radviz Deluxe map the samples by the Hamiltonian cycle on a circle, there are still four differences between HCHC and Radviz Deluxe. 1. HCHC include a new deep clustering method GLDC to mine the clusters, cluster similarities, and outliers in data, however, Radviz Deluxe is only for data visualization without clustering. 2. The Anchors in HCHC index the cluster labels, however, the anchors in Radviz Deluxe index the data features. 3. HCHC aims to visualize the mined clustering result in GLDC; however, Radviz Deluxe aims to visualize the feature values of the samples. 4. HCHC maps the samples by their clustering probability distributions; therefore, the outliers with low probability to any cluster can be mapped near the circle’s centre. However, Radviz Deluxe maps the samples by their feature values, so the outliers may be anywhere in the circle.

C. The Introduction of Radviz

Radviz defines the polar coordinate of feature $\boldsymbol{f_j}$ as

$$\boldsymbol{\mu_{f_j}} = [r \times \cos(\alpha_{\boldsymbol{f_j}}), r \times \sin(\alpha_{\boldsymbol{f_j}})] \tag{18}$$

where $\alpha_{\boldsymbol{f_j}}$ is the angle corresponding to the position of $\boldsymbol{f_j}$ and r is the radius of the circle. The the position of $\boldsymbol{x_i}$ in the circle can be computed by

$$\boldsymbol{\mu}_{\boldsymbol{x_i}} = \sum_{j=1}^{d} \frac{x_{i,j}}{\sum_{k=1}^{d} x_{i,k}} \boldsymbol{\mu}_{\boldsymbol{f_j}} \tag{19}$$

where d is the number of the features.

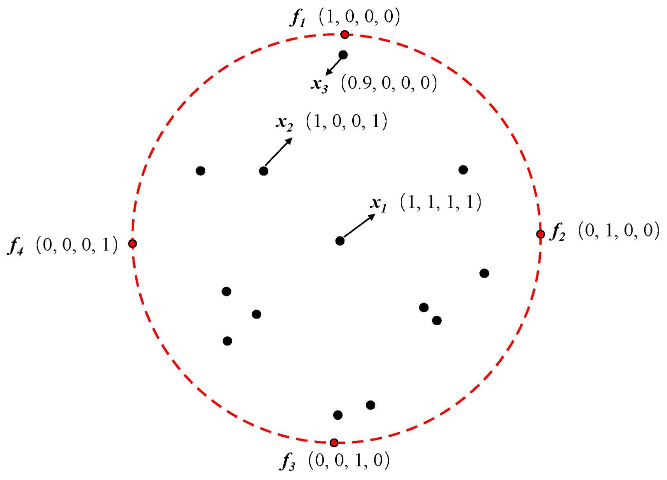


Figure 10: An example of RadVis visualization.

An example of the Radviz is shown in Fig 10. The coordinates represent the values of features $\boldsymbol{f_1}$, $\boldsymbol{f_2}$, $\boldsymbol{f_3}$, and $\boldsymbol{f_4}$ for the indicated samples. $\boldsymbol{x_1}$ is at the center of the circle and has coordinates $(1, 1, 1, 1)$. $\boldsymbol{x_2}$ has coordinates $(1, 0, 0, 1)$. Therefore, the distance between $\boldsymbol{x_2}$ and $\boldsymbol{f_1}$ is the same as the distance between $\boldsymbol{x_2}$ and $\boldsymbol{f_4}$. $\boldsymbol{x_3}$ has coordinates $(0.9, 0, 0, 0)$. It is located near to anchor the of $\boldsymbol{f_1}$.

D. The Motivation of Our Work

D.1. The Limitations in the Presentation of High-dimensional Clustering

For an informative clustering result that is more than pseudo sample labels for high-dimensional data, researchers use some embedding methods, such as MDS and t-SNE, to visualize the sample structures in the embedded space of deep clustering (Zang et al., 2022). However, the visualized result may be inconsistent with the clustering result. One reason is that it is hard to visualize all of the distinguishing information of high-dimensional in 2D space. The other one is that these methods may have biases in the visualization. To give a detailed analysis, in Fig 11 and 12, we show the visualized results of five popular embedding methods including MDS (Kruskal, 1978), PCA (Abdi & Williams, 2010), Isomap (Balasubramanian, 2002), t-SNE (Van der Maaten & Hinton, 2008), and UMAP (McInnes et al., 2018) in the embedded space of our GLDC on MNIST-test. Note that in this deep clustering, the clustering accuracy (ACC) of MNIST-test is 0.97. In other words, most distinguishing information of the different classes in MNIST-test is included in the embedded space.

As we can see from Fig. 11 and 12, although most distinguishing information of the different classes is included in the embedded space, we still hard to recognize the pink, green, cyan, and dark cyan classes by MDS and PCA without the colour labels. This is because MDS and PCA cannot visualize all of the distinguishing information in embedded space with 2 dimensions. Isomap can better visualize the dissimilarities between the different classes than MDS and PCA, but we are hard to recognize the grey and dark blue classes by Iosmap. t-SNE and UMAP can produce a better visualization than other methods, but they cannot well capture the global-structure of the data in deep clustering, i.e., t-SNE and UMAP cannot visualize similar classes, such as green and pink classes. Furthermore, t-SNE also cannot show the sample distribution, such as the density distribution of different classes, to explain the clustering result.

Agglomerative hierarchical clustering combines the samples in data into clusters, those clusters into larger clusters, and so forth, creating a hierarchy of clusters (Rani[1] & Rohil, 2013). In this way, some agglomerative hierarchical clustering

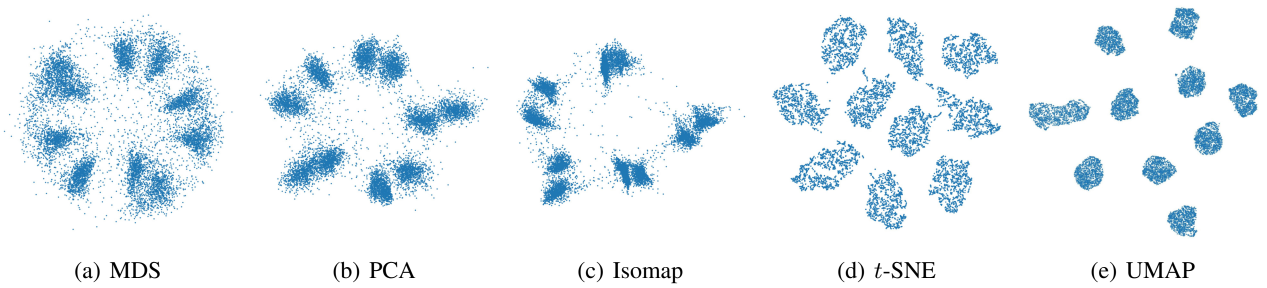


Figure 11: The visualized results of deep features by different visualization methods on MNIST-test.

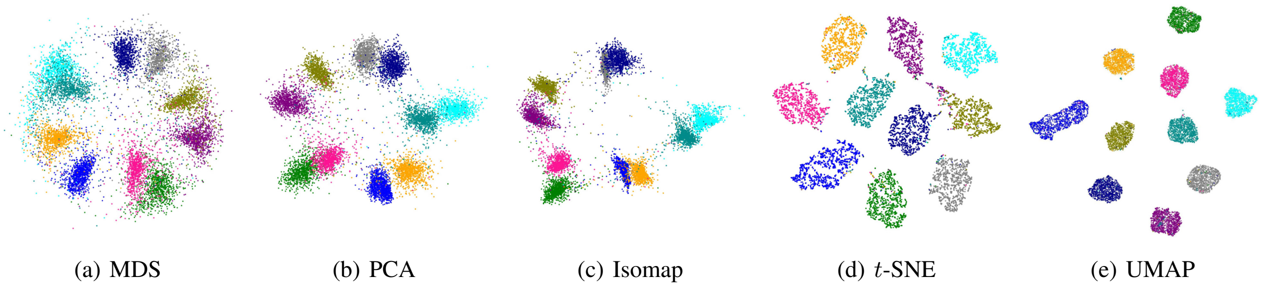


Figure 12: The visualized results of deep features by different visualization methods on MNIST-test. The different colours present the labels of the samples.

methods can mine the similarities between clusters and outliers in high-dimensional data and then present the mined knowledge in a dendrogram from the clustering process (Nielsen, 2016; Ester et al., 1996). Fig. 13 is an example to show the dendrogram with six samples.

As we can see the limitations of the presentations in the dendrogram are as follows. 1) The dendrogram cannot show the in-between samples of the different clusters and the clustering probability distributions, thus the recognized similar clusters and outliers cannot be well explained. 2) It is hard for a dendrogram to show all of the samples by the leaves when the number of samples is large.

D.2. How to Improve the Visualization of High-dimensional Clustering

From the above analysis, we should improve the visualization of high-dimensional clustering in the following three goals. 1) There should be an indicator for the cluster of each non-outlier sample and the visualization should be consistent with the clustering result. 2) Besides the clusters, the similarities between different clusters, and outliers also should be analyzed and then visualized. 3) The visualization should be an explanation for the clustering result. For example, the mapped in-between samples of the different clusters can be used to explain the recognized similar clusters and the clustering probability distributions of different clusters can be used to explain the recognized outliers.

Combining deep clustering with RadViz Deluxe can be a solution for improving the visualization of high-dimensional clustering. By mapping the extracted clustering probability distributions on the deep clustering in RadViz Deluxe, the cluster of each non-outlier sample can be indexed by its nearest cluster anchor and thus the visualization will be consistent with the clustering result (Zhao et al., 2018). When we analyze the similarities between the different clusters in deep clustering, the optimal Hamiltonian cycle generated by these similarities is the global optimal solution for placing the cluster anchors on the circumference of a circle. In this way, the similarities between the different clusters can be well presented in the visualization. Finally, each sample at the centre of the circle can be seen as an outlier, because it has a low probability of any cluster and the mapped in-between samples of different clusters can be used to explain the recognized similar clusters.

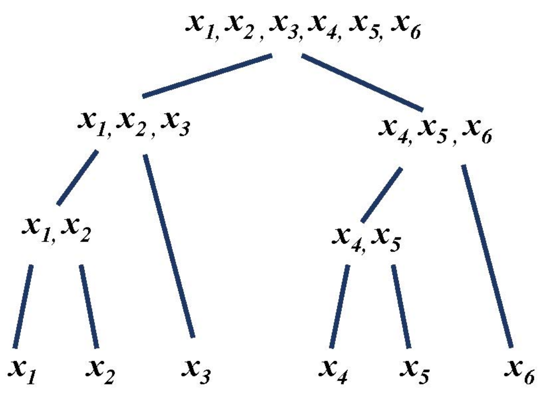


Figure 13: An example of dendrogram in hierarchical clustering.

E. The Algorithms of HCHC

The algorithm of GLDC is summarized in Algorithm 1.

Algorithm 1 GLDC

```
 1: Initialize $\theta$, $\theta'$, $\phi$, $\beta_1$, $\beta_2$, and $\gamma$
 2: Initialize a random process $\mathcal{N}$
 3: Get $\widetilde{\boldsymbol{X}}$ of $\boldsymbol{X}$ by (7)
 4: for $episode = 0$ to $pretraining - episode_{max}$ do
 5:     for mini-batch in $\boldsymbol{X}$ do
 6:         Update $\theta$ and $\theta'$ by minimizing reconstruction loss in (1)
 7:     end for
 8: end for
 9: for $episode = 0$ to $episode_{max}$ do
10:     for mini-batch in $\boldsymbol{X}$ do
11:         Get the embedded samples by $\boldsymbol{z_i} = G_\theta(\boldsymbol{x_i})$
12:         Compute the reconstruction loss by (1)
13:         Construct weighted adjacency matrix $\boldsymbol{W^l}$ by (2)
14:         Get the clustering probability distributions by (3)
15:         Compute the graph learning loss by (6)
16:         Compute the augmentation learning loss by (8)
17:         Update $\theta$ and $\phi$ by minimizing the clustering loss in (9)
18:     end for
19: end for
20: Get $\boldsymbol{P} = \{\boldsymbol{p_1}, \boldsymbol{p_2}, \cdots, \boldsymbol{p_n}\}$.
```

The whole algorithm of our mapping is summarized in Algorithm 2

Algorithm 2 HCHC

```
1: Initialize $\boldsymbol{state} = 1$
2: Initialize $\boldsymbol{cur} = (1 << c) - 1$
3: Initialize $last = 1$
4: Compute the dissimilarities between each pair of $\boldsymbol{\rho_i}$ and $\boldsymbol{\rho_j}$ based on (12)
5: while $\boldsymbol{state} < (1 << c)$ do
6:   for $j = 2$ to $c$ do
7:     if $(\boldsymbol{state}\&(1 << j)! = 0)$ then
8:       for $k = 2$ to $c$ do
9:         if $(\boldsymbol{state}\&(1 << k)! = 0)$ then
10:          Update $dp(\boldsymbol{\rho_1}, \boldsymbol{\rho_j}, \boldsymbol{state})$ by (13)
11:        end if
12:      end for
13:    end if
14:  end for
15:  $\boldsymbol{state}$ = $\boldsymbol{state} + 2$
16: end while
17: for $i = c$ to 1 do
18:   temp = 1
19:   for $j = 1$ to $c$ do
20:     if $((\boldsymbol{cur}\&1 << j)! = 0$ and $dp(\boldsymbol{\rho_1}, \boldsymbol{\rho_j}, \boldsymbol{cur}) + dis(\boldsymbol{\rho_{temp}}, \boldsymbol{\rho_{last}}) > dp(\boldsymbol{\rho_1}, \boldsymbol{\rho_j}, \boldsymbol{cur}) + dis(\boldsymbol{\rho_j}, \boldsymbol{\rho_{last}})$ then
21:       temp = j
22:     end if
23:   end for
24:   $\boldsymbol{\rho^i} = \boldsymbol{\rho_{temp}}$
25:   $\boldsymbol{cur}\oplus = 1 << tem$
26:   $last = temp$
27: end for
28: Compute $\alpha_{\boldsymbol{\rho^i}}$ for each $\boldsymbol{\rho^i}$ by (14)
29: Compute $\boldsymbol{\mu}_{\boldsymbol{\rho^i}}$ for each $\boldsymbol{\rho^i}$ by on (15)
30: Compute $\boldsymbol{\mu_{x_i}}$ in $\boldsymbol{P}$ by on (16)
31: Map each $\boldsymbol{\mu}_{\boldsymbol{\rho^i}}$ on a circle
32: Map each $\boldsymbol{\mu_{x_i}}$ in the circle
```

F. The Theoretical Analysis of Hamiltonian Cycle Mapping

F.1. Analysis on the Mapping of Cluster Similarity

The Hamiltonian cycle problem is formulated by an Irish mathematician, William Rowan Hamilton, to ask whether there is a cycle in a graph $\boldsymbol{G}$ passes through every vertex $\boldsymbol{v_i}$ exactly once, except the first passed one $\boldsymbol{v_1}$ as $\Pi = \{\boldsymbol{v^1}, \boldsymbol{v^2}, \cdots, \boldsymbol{v^m}, \boldsymbol{v^1}\}$, where m is the number of the vertices (Dirac, 1952). If $\boldsymbol{G}$ has a Hamiltonian cycle, $\boldsymbol{G}$ is Hamiltonian. Hamiltonian cycle is highly related to a classical problem, the travelling salesman. This problem is defined as follows (Hoogeveen, 1991).

Definition F.1. Given a complete undirected graph $\boldsymbol{G}$ on m vertices and a distance $dis(\boldsymbol{v_i}, \boldsymbol{v_j})$ for each edge between $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$, find a Hamiltonian cycle of minimum total length. This Hamiltonian cycle is the optimal Hamiltonian cycle.

In HCHC, we use Hamiltonian cycle to present the similarities between different clusters. A Hamiltonian cycle can present the sampled similarities $\{s(\boldsymbol{\rho^1}, \boldsymbol{\rho^2}), \cdots, s(\boldsymbol{\rho^{c-1}}, \boldsymbol{\rho^c}), s(\boldsymbol{\rho^1}, \boldsymbol{\rho^c})\}$ from $\{s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}) | i = 1, \cdots, c, i < j\}$. Define $t(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})$ as the weighted $s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})$ as

$$t(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}) = T[s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})] \tag{20}$$

where $T(\cdot)$ is the weighting function and for any s, $-1 < T(s) < 1$. This weight can be computed by different definitions to select the aspect for sampling the similarities. In this paper, we define $t(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}) = s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})$. This definition means that the similarity and its importance are in the direct ratio. We have the following theorem to get the global optimal mapping of the similarities in the selected aspect.

Theorem F.2. *The mapping of the optimal Hamiltonian cycle will maximize the sum of the weights of the selected similarities as*

$$S_{sam} = \sum_{i=1}^{c-1} t(\boldsymbol{\rho^i}, \boldsymbol{\rho^{i+1}}) + t(\boldsymbol{\rho^1}, \boldsymbol{\rho^c}) \tag{21}$$

The proof of the above theorem is as follows.

Proof. We can define a normalized $dis(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})$ for $\boldsymbol{\rho_i}, \boldsymbol{\rho_j}$ as

$$dis(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}) = \frac{1 - t(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})}{\sum_{i=1}^{c-1} \sum_{j=i+1}^{c} (1 - t(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}))} \tag{22}$$

Then

$$\sum_{i=1}^{c-1} dis(\boldsymbol{\rho^i}, \boldsymbol{\rho^{i+1}}) + dis(\boldsymbol{\rho^1}, \boldsymbol{\rho^c}) = \frac{c - (\sum_{i=1}^{c-1} t(\boldsymbol{\rho^i}, \boldsymbol{\rho^{i+1}}) + t(\boldsymbol{\rho^1}, \boldsymbol{\rho^c}))}{\sum_{i=1}^{c-1} \sum_{j=i+1}^{c} (1 - t(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}))} \tag{23}$$

As we can see, c and $\sum_{i=1}^{c-1} \sum_{j=i+1}^{c} (1 - t(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}))$ are two invariant constants, thus S_{sam} can be maximized by

$$\arg\min_{\Pi} || \sum_{i=1}^{c} dis(\boldsymbol{\rho^i}, \boldsymbol{\rho^{i+1}}) + dis(\boldsymbol{\rho^1}, \boldsymbol{\rho^c}) ||_2 \tag{24}$$

Therefore, to get the global optimal mapping of the similarities, this problem can be solved by the optimal Hamiltonian cycle Π^* of $\boldsymbol{G_P}$ with the vector $\boldsymbol{\rho_i}$ and the distance $dis(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})$. □

We can define $T[s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})]$ as

$$T(s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})) = sgn(s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})) \times |s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})|^{\gamma} \tag{25}$$

where $sgn(\cdot)$ is a sign function.

$$sgn(s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})) = \begin{cases} 1 & s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}) > 0 \\ -1 & \text{otherwise} \end{cases} \tag{26}$$

If $\gamma = 0$, the weights of all similarities in $\{s(\boldsymbol{\rho_i}, \boldsymbol{\rho_j}) | i = 1, \cdots, c, i < j\}$ will be same, in this case S_{sam} can be maximized by a randomly selected Hamiltonian cycle Π. If $\gamma \to \infty$, S_{sam} can be maximized by selecting the shortest unselected edge that cannot form the cycle in each iteration, except the last iteration. In the last iteration, we select the edge that connects the first vertex and the last vertex. The time complexity of this process is $O(c^3)$, in this complexity, we can get the orders of 1000 clusters in acceptable time consumption.

The visualized results with different γ on MNIST-test are shown in Fig. 14. As we can see, the similarities of the different clusters in MNIST-test can be well presented when $\gamma = 1$ but cannot be well presented by the randomly selected Hamiltonian cycle. When γ is very large, these similarities can be presented with acceptable time consumption.

F.2. Analysis on the Mapping Performance

The main idea to improve the mapping of Radviz is that the further away a point is from the center, the more informative its position is, being the point closer to the attributes having the highest values (Angelini et al., 2019). Thus, on the circumference of a circle, the objective function to improve mapping by the layout of the anchors can be defined as

$$\arg\max_{\Pi = \{\boldsymbol{\rho^1}; \cdots; \boldsymbol{\rho^c}\}} \sum_{i=1}^{n} || \sum_{j=1}^{c} p_{i,j} \mu_{\boldsymbol{\rho^j}} ||_2^2 \tag{27}$$

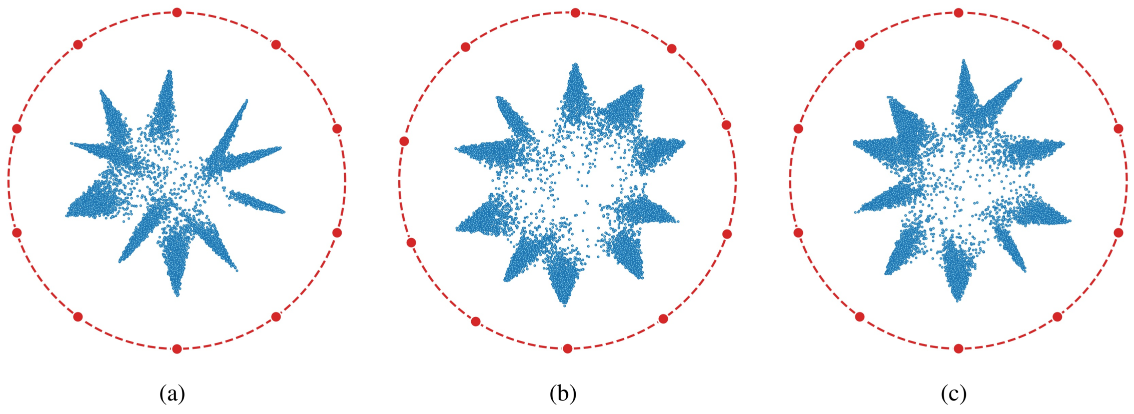


Figure 14: Visualized results with different γ. (a) Random Hamiltonian cycle with the same similarity, e.g., $\gamma = 0$. (b) $\gamma = 1$. (c) $\gamma = 1000$.

This objective function aims to make the points far away from the center. Define $r = 1$, then we have

$$\begin{aligned} & \sum_{i=1}^{n} || \sum_{j=1}^{c} p_{i,j} \mu_{\boldsymbol{\rho^j}} ||_2^2 \qquad (28) \\ = & \sum_{i=1}^{n} || \sum_{j=1}^{c} p_{i,j} cos(\alpha_{\boldsymbol{\rho^j}}), \sum_{j=1}^{c} p_{i,j} sin(\alpha_{\boldsymbol{\rho^j}}) ||_2^2 \\ = & \sum_{i=1}^{n} \sum_{j=1}^{c} p_{i,j}^2 (cos(\alpha_{\boldsymbol{\rho^j}})^2 + sin(\alpha_{\boldsymbol{\rho^j}})^2) \\ + & \ 2 \sum_{i=1}^{n} \sum_{j=1}^{c-1} \sum_{k=j+1}^{c} p_{i,j} p_{i,k} (cos(\alpha_{\boldsymbol{\rho^j}}) cos(\alpha_{\boldsymbol{\rho^k}}) + sin(\alpha_{\boldsymbol{\rho^j}}) sin(\alpha_{\boldsymbol{\rho^k}})) \\ = & \sum_{i=1}^{n} \sum_{j=1}^{c} p_{i,j}^2 (cos(\alpha_{\boldsymbol{\rho^j}})^2 + sin(\alpha_{\boldsymbol{\rho^j}})^2) + 2 \sum_{i=1}^{n} \sum_{j=1}^{c-1} \sum_{k=j+1}^{c} p_{i,j} p_{i,k} (cos(\alpha_{\boldsymbol{\rho^k}} - \alpha_{\boldsymbol{\rho^j}})) \\ = & \sum_{i=1}^{n} \sum_{j=1}^{c} p_{i,j}^2 + 2 \sum_{i=1}^{n} \sum_{j=1}^{c-1} \sum_{k=j+1}^{c} p_{i,j} p_{i,k} cos(\alpha_{\boldsymbol{\rho^k}} - \alpha_{\boldsymbol{\rho^j}}) \\ = & \sum_{i=1}^{n} \sum_{j=1}^{c} p_{i,j}^2 + \sum_{j=1}^{c-1} \sum_{k=j+1}^{c} (\boldsymbol{\rho^j})^\intercal \boldsymbol{\rho^k} cos(\alpha_{\boldsymbol{\rho^k}} - \alpha_{\boldsymbol{\rho^j}}) \end{aligned}$$

Thus the objective function (27) can be maximized by

$$\arg \max_{\Pi = \{\boldsymbol{\rho^1}; \cdots ; \boldsymbol{\rho^c}\}} \sum_{j=1}^{c-1} \sum_{k=j+1}^{c} (\boldsymbol{\rho^j})^\intercal \boldsymbol{\rho^k} cos(\alpha_{\boldsymbol{\rho^k}} - \alpha_{\boldsymbol{\rho^j}}) \qquad (29)$$

Based on Pearson correlation coefficient, the normalized objective function can be defined by as

$$\arg \max_{\Pi = \{\boldsymbol{\rho^1}; \cdots ; \boldsymbol{\rho^c}\}} \sum_{j=1}^{c-1} \sum_{k=j+1}^{c} \frac{(\boldsymbol{\rho^j} - \overline{\boldsymbol{\rho^j}})^\intercal (\boldsymbol{\rho^k} - \overline{\boldsymbol{\rho^k}})}{|\boldsymbol{\rho^j} - \overline{\boldsymbol{\rho^j}}||\boldsymbol{\rho^k} - \overline{\boldsymbol{\rho^k}}|} cos(\alpha_{\boldsymbol{\rho^k}} - \alpha_{\boldsymbol{\rho^j}}) \qquad (30)$$

To solve this problem, we should put the anchors of similar clusters together. From the analysis in Appendix F.1, we can see that the optimal Hamiltonian cycle Π^* is an approximate solution of the objective function (30).

G. Experiment Setting

We use seven datasets including MNIST (Deng, 2012), Fashion (Xiao et al., 2017), USPS (Hull, 1994), Reuters10k (Lewis et al., 2004), HHAR (Stisen et al., 2015), Pendigits (ASUNCION, 2007), and BH (Abdelaal et al., 2019) to illustrates the effectiveness of HCHC. The details of the datasets are shown in Table 3.

There are some parameters that need to be tuned. The initial β_1 is set as 5. As we can see, with the increase of the iteration numbers in training, the magnitudes of L_r and L_a will become smaller and smaller. Thus a discount factor γ is used to tune the magnitude of β_1 in every iteration t as

$$\beta_1 = \gamma^t \beta_1 \tag{31}$$

where γ is set as 0.8. β_2 is set as 10. σ^2 is set from $\{0.05, 0.1, 0.2\}$. ξ is set from $\{0.005, 0.05, 0.1, 0.2\}$. k is set from $\{3, 4, 5, 30\}$. The batch size is set as 128. We use Adam optimizer in our training and the learning rate is set as 0.002. The autoencoder is composed of eight layers with dimensions D-500- 500-2000-5-2000-500-500-D, where D is the dimension of the input samples.

Table 3: Data description.

DID	Dataset	Instances	Features	Classes
1	MNIST	70000	784	10
2	Fashion	70000	784	10
3	USPS	9298	256	10
4	Reuters10k	10000	2000	4
5	HHAR	10299	561	6
6	Pendigits	10992	16	10
7	BH	8569	17499	14

We also compare our GLDC with nine clustering methods including k-means (MacQueen et al., 1967), GMM (Rasmussen, 1999), SC (Shi & Malik, 2000), DEC (Xie et al., 2016), IDEC (Guo et al., 2017), DSC (Shaham et al., 2018), JULE (Yang et al., 2016), DSCDAN (Yang et al., 2019), N2D (McConville et al., 2021). The details of the compared methods are as follows.
1. k-means works by computing centres of the different clusters and cluster assignments iteratively by the Euclidean distance (MacQueen et al., 1967).
2. GMM works by computing the centres of the different clusters and cluster assignments iteratively by the Gaussian model (Rasmussen, 1999).
3. SC learns a map that embeds input data points into the eigenspace of their associated Laplacian matrix and then clusters them by k-means (Shi & Malik, 2000).
4. DEC simultaneously learns feature representations and cluster assignments by deep neural networks (Xie et al., 2016).
5. IDEC improves DEC by integrating the clustering loss and autoencoder reconstruction loss (Guo et al., 2017).
6. DSC learns a map that embeds input data points into the eigenspace of their associated Laplacian matrix and then clusters them, in the deep neural network (Shaham et al., 2018).
7. JULE consists of a multinomial logistic regression function stacked on top of a multi-layer convolutional autoencoder in deep clustering (Yang et al., 2016).
8. DSCDAN discriminatively performs feature embedding and spectral clustering by CNN for image clustering (Yang et al., 2019).
9. N2D replaces the clustering layer with a manifold learning technique on the autoencoder representations (McConville et al., 2021).

We use ACC and NMI to measure the clustering performance of GLDC with seven existing clustering methods (Kuhn, 1955). The definition of ACC is as follows. Denote $\boldsymbol{a} = \{a_1, a_2, \cdots, a_n\}$ as the clustering results and $\boldsymbol{b} = \{b_1, b_2, \cdots, b_n\}$ as the ground truth label of X. ACC is defined as:

$$ACC = \frac{\sum_{i=1}^{n} \delta(a_i, map(b_i))}{n} \tag{32}$$

where $\delta(a, b) = 1$, if $a = b$ and $\delta(a, b) = 0$, otherwise. $map(b_i)$ is the best mapping function that permutes clustering labels to match the given truth labels using the Kuhn-Munkres algorithm. The larger ACC is, the better the clustering result

Table 4: The mapping colours and corresponding time periods (day/month/year).

Sample color	(a)	(b)	(c)	(d)
blue	22/1/20 to 21/9/20	22/1/20 to 21/7/20	22/1/20 to 21/6/20	22/1/20 to 21/5/20
green	22/9/20 to 21/5/21	22/7/20 to 21/1/21	22/6/20 to 21/11/20	22/5/20 to 21/9/20
purple	22/5/21 to 21/1/22	22/1/21 to 21/7/21	22/11/20 to 21/4/21	22/9/20 to 22/1/21
cyan	-	22/7/21 to 21/1/22	22/4/21 to 21/9/21	22/1/21 to 21/5/21
olive	-	-	22/9/21 to 21/1/22	22/5/21 to 21/9/21
dark-blue	-	-	-	22/9/21 to 21/1/22

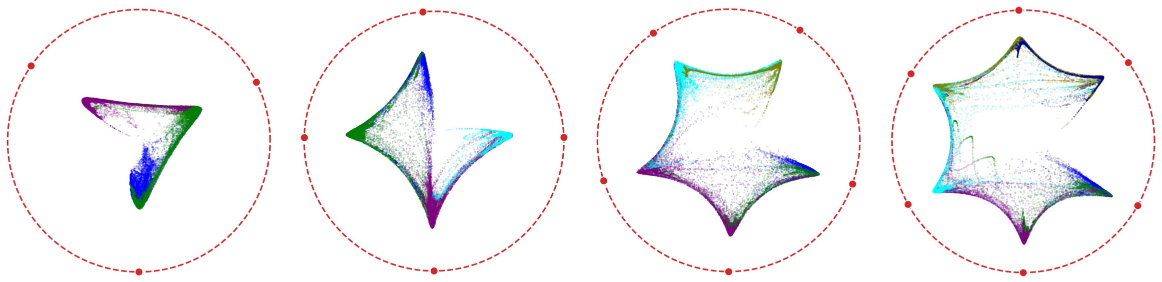

(a) Labeled by every 8 months (b) Labeled by every 6 months (c) Labeled by every 5 months (d) Labeled by every 4 months

Figure 15: The results of HCHC on the COVID-19 dataset.

is. The NMI is defined as

$$NMI(\boldsymbol{b}, \boldsymbol{a}) = \frac{MI(\boldsymbol{b}, \boldsymbol{a})}{\sqrt{(H(\boldsymbol{b})H(\boldsymbol{a}))}} \tag{33}$$

where $H(\boldsymbol{b})$ and $H(\boldsymbol{a})$ are the entropies of $\boldsymbol{b}$ and $\boldsymbol{a}$. $MI(\boldsymbol{b}, \boldsymbol{a})$ is the mutual information metric of $\boldsymbol{b}$ and $\boldsymbol{a}$. The larger NMI is, the better the clustering result is.

We use HCHC to learn the temporal features of COVID-19 by a dataset that includes the available COVID-19 daily information of the different states in the USA from 21/1/2020 to 21/1/2022. There are 37525 samples in this dataset and their features are detailed in Table 5.

Table 5: The features of the COVID-19 dataset.

Feature	Brief explanation
#1	daily cases
#2	daily cases per $100K$ capita
#3	daily deaths
#4	daily deaths per $100K$ capita
#5	the percentage of total cases / total deaths
#6	the percentage of total cases / population
#7	weekly cases
#8	the change of weekly cases
#9	the cases in 28 days
#10	the change of the cases in 28 days

H. Supplementary Experiment

H.1. Supplementary Experiment for Covid-19 Dataset

In Fig. 15 (a) where the data are labeled by every 8 months, the samples from 22/9/20 to 21/5/21 are divided into two clusters; In Fig. 15 (c) where the data are labeled by every 5 months, the samples from 22/1/20 to 21/6/20 and 22/6/20 to

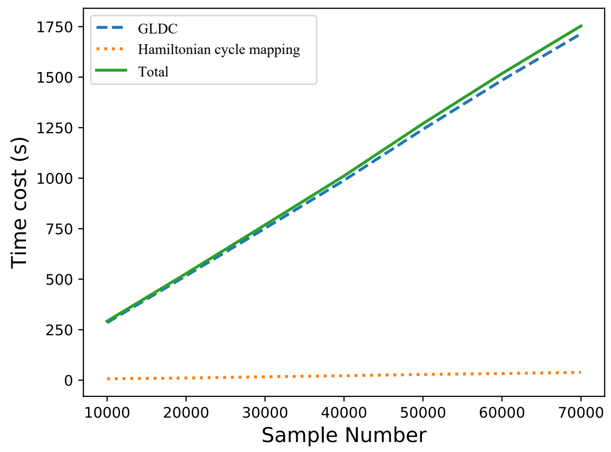


Figure 16: The time cost of HCHC.

21/11/20 are clustered in one cluster, the samples from 22/11/20 to 21/4/21 are divided into two clusters, and the samples from 22/4/21 to 21/9/21 are also divided into two clusters; In Fig. 15 (d) where the data are labeled by every 4 months, except the samples which from 22/1/20 to 21/5/20 and 22/5/20 to 21/9/20 are clustered into one cluster, the samples from any other period are divided into two clusters. Whereas, in Fig. 15 (b) where the data are labeled by every 6 months, the samples from the same period can be mapped close to the same cluster anchor.

H.2. Time Complexity Analysis

The time cost of our HCHC is linear to the number of the samples in data, thus it can effectively visualize the clustering results in big data. Fig. 16 shows the time cost of HCHC on MNIST with different numbers of the samples. The epoch number is set as 200. We can find that the time cost of GLDC is more than the time cost of Hamiltonian cycle mapping on MNIST dataset.

H.3. Supplementary Experiment for Different Visualization Methods

Here we show the visualized results of HCHC, MDS, PCA, Isomap, t-SNE, and UMAP on the datasets of USPS, Reuters10k, HHAR, Pendigits, and BH. The results are shown in Fig. 17.

From Fig. 17, we can see that for USPS, HCHC can better better present the cluster number and index the clutering results than MDS, PCA, and Isomap. It also can better present the cluster similarities and outliers than t-SNE and UMAP for USPS and BH. For HHAR, Pendigit, and BH, with the help of anchors, HCHC can better present the cluster number and index the clustering results than any other visualization method. HCHC also can better present the cluster similarities than t-SNE for HHAR and better present outliers than t-SNE and UMAP for Pendigit. For these two datasets, MDS, PCA, and Isomap can better present the cluster similarities and outliers than t-SNE and UMAP, but t-SNE and UMAP can better present the cluster structure than MDS, PCA, and Isomap. Although HCHC, PCA, and Isomap can well present the clustering result of Reuters10k, only HCHC can well present the similarities between the purple class, grey class, and green class.

H.4. Case Study

This subsection summarizes the feedback from the ten experts on different visualization methods for clustering. Five of the experts major in clustering and five of the experts major in visualization. We design the following three tasks to evaluate the quality of different visualization methods. Task (a) is to verify the ability of each visualization method to present the number of clusters and correctly identify which cluster a sample belongs to. Task (b) is to verify the ability of each visualization method to present the similarities between the samples from different classes. Task (c) is to verify the ability of each visualization method to present the outliers. Each task is scored on a scale from 0 to 5. The results of six visualization methods including HCHC, MDS, PCA, Isomap, t-SNE, and UMAP are used to visualize the clustering results

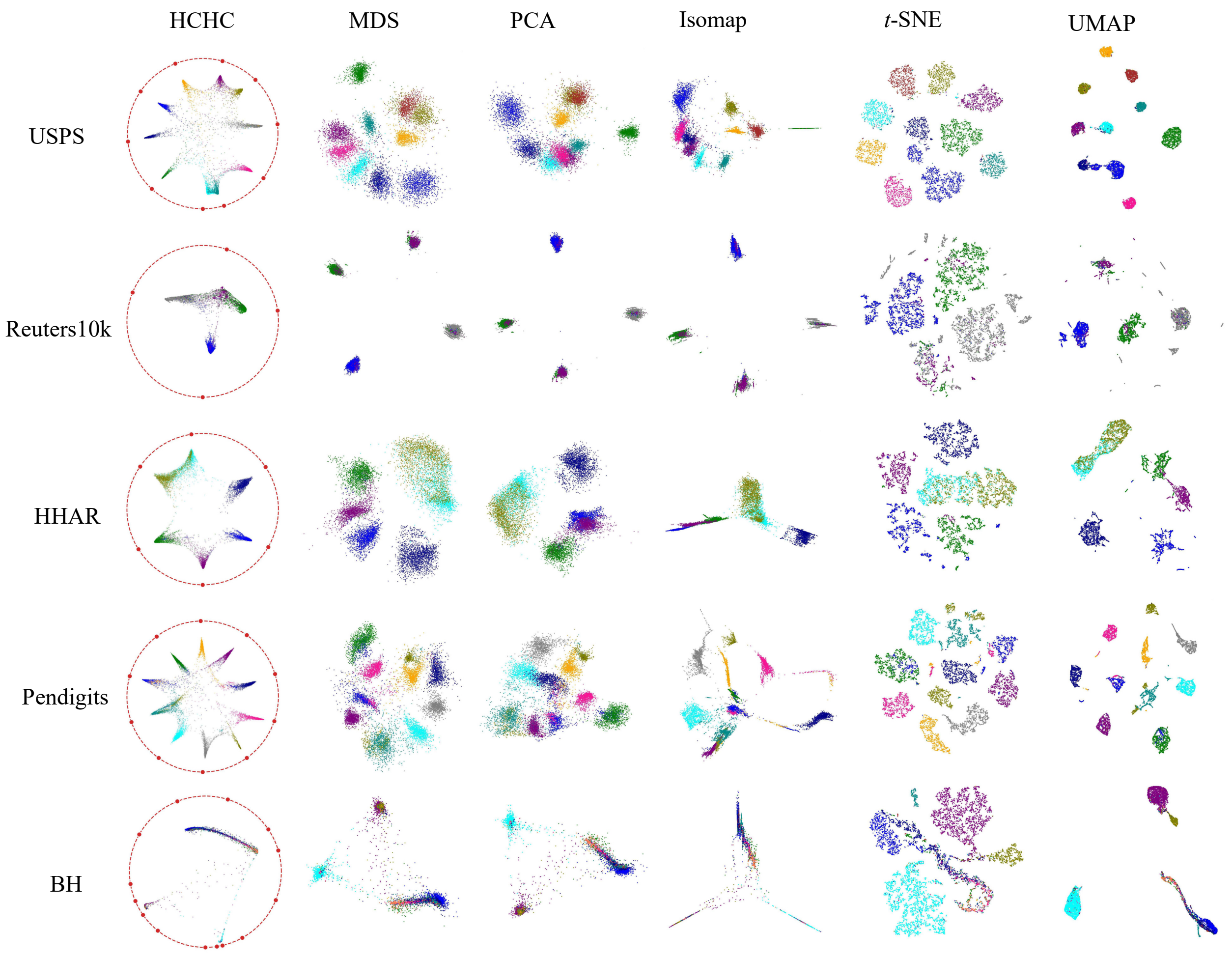


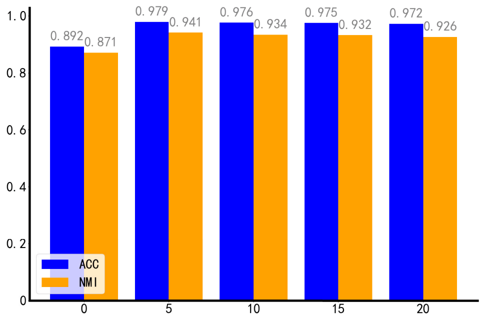
Figure 17: The visualized results of the different visualization methods. Samples are coloured by their labels.

Table 6: Expert comparison of different visualization methods on MNIST.

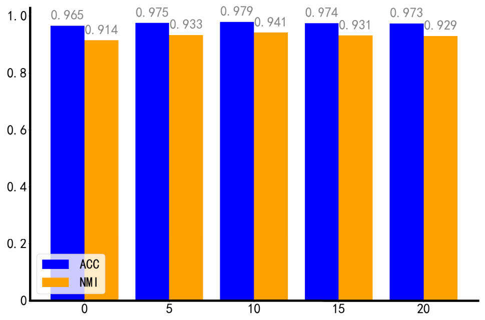
Method	Task (a)	Task (b)	Task (c)	Average
MDS	3.3	4.2	4.3	3.9
PCA	3.6	3.9	4.5	4
Isomap	3.9	4.4	4.1	4.1
t-SNE	4.5	2.1	2.5	3.0
UMAP	4.9	3.2	1.5	3.2
HCHC	4.5	4.6	4.5	4.5

Table 7: Expert comparison of different visualization methods on Fashion.

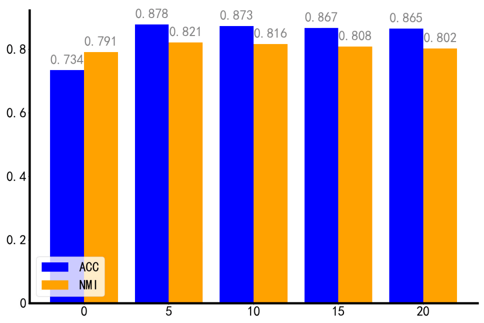
Method	Task (a)	Task (b)	Task (c)	Average
MDS	2.1	4.6	4.3	3.7
PCA	2.4	4.6	4.3	3.8
Isomap	1.8	4.3	4.1	3.4
t-SNE	3.0	4.0	3.5	3.5
UMAP	3.3	3.5	2.1	3.0
HCHC	4.0	4.4	4.4	4.3



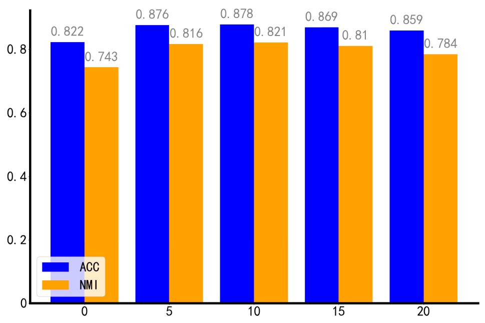
(a) MNIST-β_1



(b) MNIST-β_2



(c) HHAR-β_1



(d) HHAR-β_2

Figure 18: Parameter analysis.

of MNIST-test and Fashion-test in Fig. 5 and 6, respectively. Then based on the visualization results, the ten experts are asked to score these visualization methods on our three designed tasks.

The average scores of the ten experts in different tasks for MNIST-test are shown in Table 6. For the results of MNIST-test, as we can see, HCHC can well perform all of our three designed tasks and thus get the highest average score. However, MDS, PCA, and Isomap cannot well perform task (a). For example, by MDS and PCA, we are hard to correctly identify the pink, green, cyan, and dark cyan clusters. By Isomap, we are hard to correctly identify grey and dark blue clusters. t-SNE and UMAP cannot well perform tasks (b) and (c). For example, the shapes of handwriting digit numbers "9" and "4" are similar, but t-SNE and UMAP cannot map the clusters of these two digit numbers close to each other.

The average scores of in different tasks for Fashion-test are shown in Table 7. For the results of Fashion-test, as we can see, HCHC also can well perform all of our three designed tasks and thus get the highest average score. For task (a) With the help of the anchors, our HCHC can better visualized the cluster number and index cluster result than any other visualization methods. For task (b) HCHC, MDS, PCA, and Isomap can well present between the samples of shirt and the samples of dress, but t-SNE and UMAP cannot do so. We are also hard to see the outliers by t-SNE and UMAP.

H.5. Parameter Analysis

In this subsection, we analyze the parameter sensitivities of β_1 and β_2 in our objective function on MNIST and HHAR. The results are shown in Fig 18. In subfigure (a) and (c), β_2 is fixed as 10. In subfigure (b) and (d), the initial β_1 is fixed as 5. It can be seen that our method is not sensitive to these two parameters.