Interactive Visual Cluster Analysis by Contrastive Dimensionality Reduction

Jiazhi Xia, Linquan Huang, Weixing Lin, Xin Zhao, Jing Wu, Yang Chen, Ying Zhao, and Wei Chen

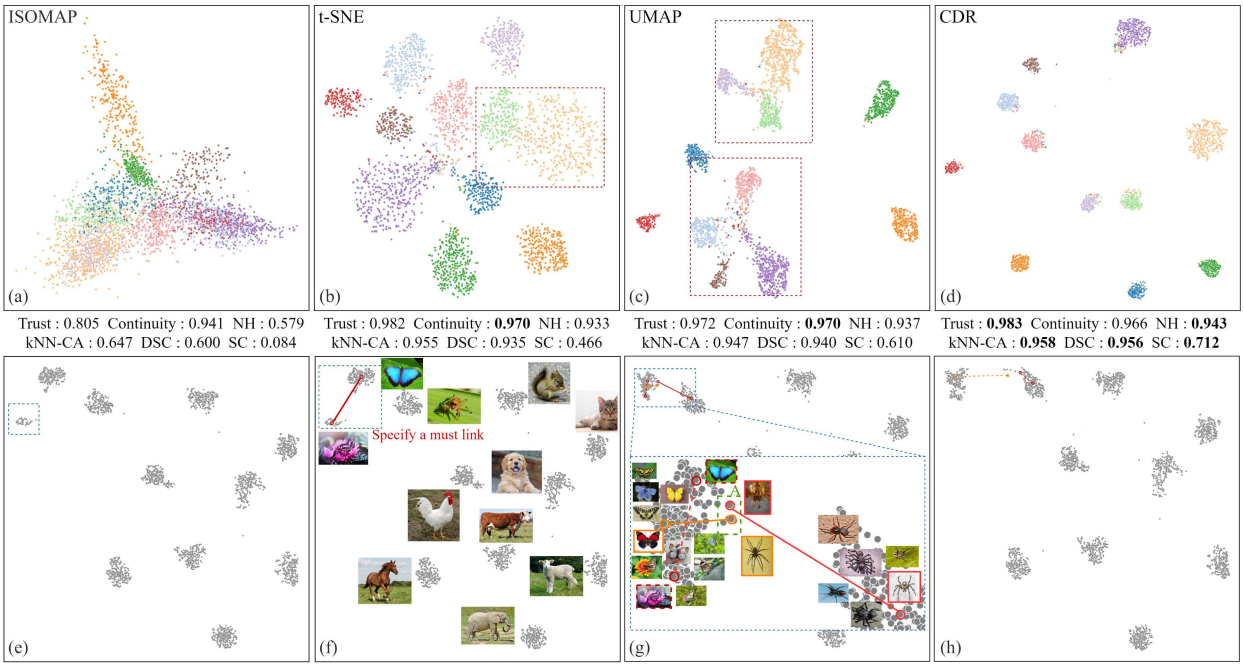


Fig. 1. The embedding results by dimensionality reduction techniques. Top: the embedding results of the Indian Food dataset [4] by (a) ISOMAP, (b) t-SNE, (c) UMAP, and (d) CDR (the Contrastive Dimensionality Reduction), respectively. The data points are color-encoded by class labels. Bottom: the interactive analysis of the Animals dataset [26] by CDR. (e) The initial embedding result. (f) A must link is added to merge the butterfly clusters. (g) An additional *cannot link* (orange), and an additional *must link* (red) are added to separate the spider and the butterfly clusters. (h) The updated embedding result after the *cannot link* and *must link* interactions.

Abstract—We propose a contrastive dimensionality reduction approach (CDR) for interactive visual cluster analysis. Although dimensionality reduction of high-dimensional data is widely used in visual cluster analysis in conjunction with scatterplots, there are several limitations on effective visual cluster analysis. First, it is non-trivial for an embedding to present clear visual cluster separation when keeping neighborhood structures. Second, as cluster analysis is a subjective task, user steering is required. However, it is also non-trivial to enable interactions in dimensionality reduction. To tackle these problems, we introduce contrastive learning into dimensionality reduction for high-quality embedding. We then redefine the gradient of the loss function to the negative pairs to enhance the visual cluster separation of embedding results. Based on the contrastive learning scheme, we employ link-based interactions to steer embeddings. After that, we implement a prototype visual interface that integrates the proposed algorithms and a set of visualizations. Quantitative experiments demonstrate that CDR outperforms existing techniques in terms of preserving correct neighborhood structures and improving visual cluster separation. The ablation experiment demonstrates the effectiveness of gradient redefinition. The user study verifies that CDR outperforms t-SNE and UMAP in the task of cluster identification. We also showcase two use cases on real-world datasets to present the effectiveness of link-based interactions.

Index Terms—Dimensionality reduction, visual cluster analysis, contrastive learning

1 Introduction

As visual cluster analysis is an inherent human-in-the-loop task that lacks a universal ground truth, it usually employs dimensionality reduction (DR) techniques to support the visualization and exploration of cluster patterns [9, 84, 87]. Many DR techniques, such as t-SNE and UMAP, employ a "proximity ≈ similarity" metaphor, in which the similarity between a pair of points is retained by the distance between their 2D embeddings [82]. Therefore, points gathered together in the embedding space can visually form implicit clusters. In this paper, we

- *Jiazhi Xia, Linquan Huang, Weixing Lin, Xin Zhao, Yang Chen and Ying Zhao are with School of Computer Science and Engineering, Central South University, China. E-mail: xiajiazhi@csu.edu.cn.*
- *Jing Wu is with Cardiff University. E-mail: wuj11@cardiff.ac.uk.*
- *Wei Chen is with State Key Lab of CAD&CG, Zhejiang University, China and Laboratory of Art and Archaeology Image (Zhejiang University), Ministry of Education, China. E-mail: chenvis@zju.edu.cn.*
- *Yang Chen and Ying Zhao are the corresponding authors. E-mail: chen1984yang@gmail.com, zhaoying@csu.edu.cn.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

propose an interactive DR technique for visual cluster analysis.

Although existing DR techniques are widely used in visual cluster analysis, their results are often not satisfactory. Fig. 1 shows the embedding results on the Indian Food dataset [4] from three widely used DR methods. We can see, when using ISOMAP [73], the data points are heavily confused, making it extremely difficult to identify clusters. For t-SNE [76] and UMAP [55], which are considered more state-of-the-art DR methods [28, 62, 78], their results have better separations between clusters. However, it would still be confusing to identify all the clusters without the color encoding of class labels, which are usually not provided in clustering tasks. Moreover, it is unclear how users can steer these embeddings. From these limitations, we identify three major requirements for a DR technique to support interactive visual cluster analysis.

The first is to correctly construct the neighborhood structures and faithfully preserve them in the embedding space. This ensures a trustworthy presentation of cluster patterns, which is the prerequisite for visual cluster analysis. The state-of-the-art t-SNE and UMAP techniques explicitly or implicitly use the k-nearest neighbors (kNN) to represent the neighborhood structures. However, kNN unavoidably has mistakes due to the limitations of Euclidean similarity measures, as data dissimilarities might be inherently non-Euclidean [57]. Under Euclidean similarity, data points in different manifolds might be considered as k-nearest neighbors of each other. Furthermore, preserving correct neighbors in the embedding space is also non-trivial. Missing neighbors and false neighbors often occur in existing DR techniques [52, 57]. Several parametric techniques, such as parametric UMAP [67], parametric t-SNE [75], and Deep recursive embedding [92], are proposed. However, their performances in terms of neighborhood preservation are worse than the state-of-the-art non-parametric techniques. It is still an open problem to present a high-quality parametric DR technique.

The second is to achieve clear visual cluster separation in the low-dimensional space, which can enhance users' confidence and speed up the process of identifying clusters. Neither t-SNE nor UMAP is favorable in this matter. As shown in Fig. 1, both methods achieved relatively low Silhouette Coefficient (SC) scores (t-SNE: 0.466; UMAP: 0.610) and have visual confusion between clusters. Existing work has been devoted to linear DR techniques that visually discriminate labeled classes [80]. However, it is unclear how to improve visual cluster separation for datasets without class labels. Moreover, although DR techniques aim to preserve neighborhood structures, separating clusters in the embedding space would destroy the structure if two clusters are not well-separated in the high-dimensional space. It is non-trivial to both separate the clusters and preserve the neighborhood. Often there is a trade-off between the two.

The third is to support effective interactions for human-in-the-loop analysis. Users would like to steer the low-dimensional embeddings using their domain knowledge, e.g., two points should be in the same cluster or belong to different clusters. Although several interactive embedding techniques have been developed [31, 45, 47, 49], there are few research efforts on steering DR for visual cluster analysis. Perez et al. [60] allow users to manipulate a single parameter to generating separated clusters. However, the interaction on parameters is not intuitive and requires model knowledge. An intuitive interactive non-linear DR technique is still an open problem.

To fulfill these requirements, we propose a contrastive dimensionality reduction technique (CDR) for interactive visual cluster analysis. Contrastive learning, which is a self-supervised representation learning method, has achieved great success in distinguishing similar/dissimilar data points [23, 37]. Therefore, we firstly introduce contrastive learning into dimensionality reduction to better preserve the neighborhood structures and present trustworthy cluster patterns. We thus carefully design the positive/negative pairs and adapt the contrastive loss in the context of dimensionality reduction. Secondly, with the aim of facilitating visual cluster analysis, we further improve the proposed CDR method to achieve better visual cluster separation while keeping the neighborhood structures. To achieve that, we redefine the gradient of the loss function to negative pairs, so that the model can distinguish similarity/dissimilarity not only between data points but also between clusters. Thirdly, we introduce link-based constraints into CDR. This enables user interactions through *must link* and *cannot link* operations, supporting users' steering of embeddings with domain knowledge. An iterative interaction-and-refine analysis loop is supported to further reduce the interactive burden. With the proposed interactive contrastive dimensionality reduction method, we also develop a prototype interface that integrates the proposed method and a set of visualizations for interactive visual cluster analysis. Through quantitative comparisons on real-world datasets, we show that the proposed CDR outperforms t-SNE and UMAP in presenting trustworthy and presenting well-separated cluster patterns. The ablation experiments show the effectiveness of the proposed method. We also demonstrate the effectiveness of CDR in interactive visual cluster analysis through a user study and two use cases. The code for the proposed CDR is available at https://github.com/DRLib/CDR.

In summary, our major contributions are:

- We propose a contrastive dimensionality reduction (CDR) method, introducing contrastive learning into dimensionality reduction and improving contrastive loss to achieve better neighborhood preservation and visual cluster separation.
- We introduce link-based interactions into CDR to capture user intentions for interactive visual cluster analysis.
- Extensive quantitative comparisons, a user study, and two case studies demonstrate the effectiveness of the proposed CDR, compared with popular DR methods (e.g., t-SNE and UMAP).

2 Related Work

2.1 Visual cluster analysis

Visual cluster analysis leverages visualization techniques to understand and explore data features, clustering models, and cluster patterns. **Data features**, such as statistical distribution and aggregated statistics of dimensions, characterize how feature distributions in dimensions affect the clustering results, enabling an understanding of the relations between clusters and relevant dimensions. They can be visualized with statistical charts such as glyph [64], histogram [18], and parallel coordinates [2, 42]. In addition to directly visualizing the statistical features, reordering them [53, 63], either in parallel coordinates [13] or in similarity matrix [91], is an effective way of clustering since instances with similar features can be closely placed through various reordering strategies. Combining visualization with **clustering models** allows users to understand, steer, and adjust clustering models. For example, hierarchical clustering structures can be explored using heatmap [69] or tree [86] to guide the refinement of clustering results. Moreover, visually encoding models like classification labels aid in examining the correlation of instance attributes within clusters [29]. Recent efforts have been made toward adopting DR techniques to explore **cluster patterns** in the embedding space [82]. In particular, the DR techniques can be used jointly with clustering algorithms, either prior to the clustering [56, 88] or after the clustering [21, 30, 48], to explicitly form clusters in a 2D embedding space. DR techniques that adopt a "proximity $\approx$ similarity" metaphor can be directly visualized to retain implicit clusters [19, 66], establishing the most intuitive relations with the original clustering structures in the high-dimensional space. Since the transformation might result in distortions that make the relative positions of the cluster instances unreliable [39], the challenge is to retain—as much as possible—the neighborhood structures while obtaining high-quality, well-separated clusters. Our approach leverages contrastive learning with interactive clustering to tackle this challenge.

2.2 Interactive clustering

Interactive clustering leverages end-users' domain knowledge to steer the automatic clustering processes. Many different interactive operations have been provided to support the steering processes, such as adjusting the features or weights of cluster instances [11, 58], parameterizing the underlying models [3, 6, 14, 22], or directly manipulating clustering results (e.g., by splitting or merging clusters) [10, 20]. However, these techniques either require a strong knowledge of underlying models or rely on the injection of domain-specific knowledge from external sources. To alleviate the cognitive costs, initial efforts have

been made toward micro-level interactions that focus on relation-based variables among samples in supervised settings. They can be visually manipulable and are independent of specific domains and underlying models. For example, users can adjust embedding distances among particular cluster instances [15, 25], specifying their relations [85], or directly establish *must link* and *cannot link* constraints for pairwise instances to refine various dimensionality reduction models [50]. The link-based interactions for clustering algorithms inspire our application of steering embeddings for interactive visual cluster analysis.

2.3 Interactive dimensionality reduction

A few dimensionality reduction techniques support an interactive DR process where users interactively steer the DR models in semi-automatic settings. Most existing techniques allow users to tweak underlying parameterizations, either for feature weights [19, 40, 81] or directly switching among different DR models [24, 48]. To alleviate the cognitive costs required by parameterizing the underlying models, the observation-level interactions are proposed to directly interact with data points within the embeddings. For example, techniques described in Brown et al. [15] and Endert et al. [31] allow users to directly adjust the positions of data points from which the hidden parameters like weighted distance functions are updated accordingly. Besides, individual points can be assigned as control points whose position adjustment would dynamically modify the global layout of embeddings [41, 54]. Techniques like interAxis [45] and AxiSketcher [49] allow users to directly manipulate data points, e.g., sketching them in lines, for specifying their dimensional relations upon which the embeddings are dynamically updated. Instead of using the above interactions, however, it is desired to leverage link constraints to provide explicit effects on the similarity matrix of embeddings [17]. Similar to our method, link-based interactions have been introduced to embedding techniques like Multidimensional Scaling [47] and kernel PCA [16] where the distance matrix of the embedding can be refined according to the link-based constraints. Unfortunately, they are not optimized for visual cluster separation, leading to relatively poor performance [5]. Few initial efforts have been made toward utilizing interactive DR for facilitating clustering analysis. For example, the technique proposed by Perez et al. [60] allows users to manipulate a single parameter for generating well-separated clusters while preserving the original structures in the high-dimensional space. Compared with these methods, the proposed approach enables direct constraints on individual data points.

3 Method

In this section, we describe our approach to interactive visual cluster analysis. Firstly, we introduce contrastive learning into dimensionality reduction to better preserve neighborhood structures. Secondly, we redefine the gradient of the loss function to negative pairs for improving the visual cluster separation. Thirdly, we introduce link-based interactions, including *must link* and *cannot link*, into CDR to enable users' steering of embeddings. We also implement a prototype interface that integrates the proposed techniques and a set of visualizations for interactive visual cluster analysis.

3.1 Contrastive Learning Based Embedding

Contrastive learning, which is a self-supervised representation learning approach, has achieved great success in instance discrimination in computer vision tasks [23, 37]. Generally, it pulls similar instances together and pushes apart dissimilar instances to learn an effective embedding [74]. The similar and dissimilar relationship are represented by positive pairs and negative pairs, respectively. The embedding network is trained with a contrastive loss function that measures the loss on point pairs. To leverage the ability of contrastive learning, we introduce it in dimensionality reduction for visual cluster analysis. In our approach, the embedding function is $f(\cdot): X \rightarrow Z$, where $X \in \mathbb{R}^{n \times d}$ represents n high-dimensional data points with the dimensionality d. $Z \in \mathbb{R}^{n \times 2}$ is the embedded data points.

Positive Pairs. The positive pairs specify the similar relationship between points. In computer vision tasks, the positive pairs are often specified by data augmentation techniques [23, 37, 74], such as rotating an image, to provide instance discrimination ability. Different from computer vision tasks, we look for cluster-level discrimination rather than instance discrimination. Therefore, we need to specify the pair belonging to the same cluster as a positive pair. We note that the k-nearest neighbors (kNN) [61] are widely considered as belonging to the same classes/clusters, e.g., the kNN-based classification algorithms [89, 90]. Therefore, for each $x_i \in X$, we sample one point x_j in its kNN to construct a positive pair $P(x_i, x_j)$. The sampling probability is defined based on UMAP [55]. Specifically, the probability that point x_i and point x_j are a positive pair is computed as

$$p_{ij} = (p_{j|i} + p_{i|j}) - p_{j|i} p_{i|j} \tag{1}$$

where $p_{j|i}$ is the conditional probability, which is defined as

$$p_{j|i} = \exp\left(-\frac{dist(x_i, x_j) - \rho_i}{\sigma_i}\right) \tag{2}$$

where $dist(x_i, x_j)$ denotes the Euclidean distance between two points. The ρ_i is the distance from x_i to its nearest neighbor. The σ_i is the normalization term calculated as the sum of the probability that point x_i is paired to its k-nearest neighbors.

Negative Pairs. With only positive pairs, the model would collapse to embed all points into the same position [35]. Therefore, we need to specify negative pairs to pull dissimilar points apart. Following the common strategy used in contrastive learning [23], for each point x_i, its negative counterparts are considered as all other points, excluding the positive counterparts. Since dissimilar neighbors have a higher probability of being sampled as negative counterparts than similar neighbors, this strategy ensures balanced positions for dissimilar neighbors in the embedding and thus alleviates the model collapse issue. Specifically, in each iteration, the training data is randomly grouped into equal-sized batches. In our implementation, we set the number of batches as 10. Each batch contains B points and their counterparts of positive pairs. Therefore, each batch finally has $2B$ points. For each point in a batch, the other $2(B-1)$ points, excluding itself and its counterpart of positive pairs, are specified as its counterparts of negative pairs $N(x_i, x_j)$.

The Contrastive Loss Function. We adopt the NT-Xent (the normalized temperature-scaled cross entropy loss) [72] as the contrastive loss function and adapt it for dimensionality reduction. In a training batch, the NT-Xent of point z_i is defined as

$$L_{NT}(z_i) = -\log \frac{\exp\left(q_{ij}/\tau\right)}{\sum_{k=1}^{2B} \mathbb{1}_{[k \neq i]} \exp\left(q_{ik}/\tau\right)} \tag{3}$$

where $\mathbb{1}_{[k \neq i]} \in \{0, 1\}$ is an indicator function evaluating to 1 if $k \neq i$ and τ denotes a temperature parameter. q_{ij} represents the similarity between z_i and z_j, which are the embeddings of x_i and x_j, respectively. Ideally, we define the similarity between z_i and z_j [55] as

$$S(z_i, z_j) = \begin{cases} 1 & \text{if } \|z_i - z_j\|_2 \leq \xi \\ \exp\left(-\left(\|z_i - z_j\|_2 - \xi\right)\right) & \text{otherwise} \end{cases} \tag{4}$$

where ξ is the minimum embedding distance between two points, and its default value is 0.1. However, this definition is non-differentiable. Instead, we define q_{ij} using a Student t-distribution with one degree of freedom to approximate $S(z_i, z_j)$ as

$$q_{ij} = \frac{1}{1 + a(||z_i - z_j||_2^2)^b} \tag{5}$$

where a and b are chosen by non-linear least squares fitting against the curve of S.

Compared to the cross entropy loss, the NT-Xent adds a temperature parameter τ. As mentioned above, the kNN of a point in the high-dimensional space may contain points in different clusters. Simply preserving all k-nearest neighbors in the low-dimensional space would retain these errors in the embedding results. Therefore, we would like to strengthen the ability to discriminate against negative pairs, especially

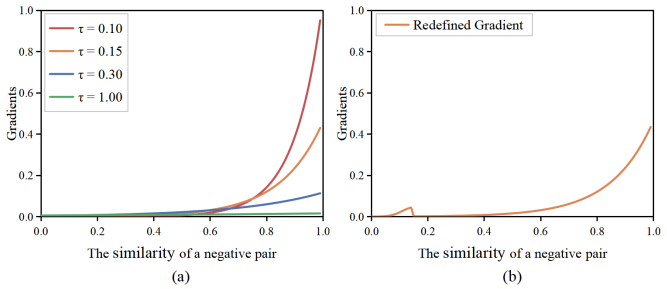


Fig. 2. The gradients with respect to the negative similarity q_{ij}.(a) Gradients of NT-Xent loss under different τ values.(b) Gradients with respect to the negative similarity after gradient redefinition. On the basis of $\tau = 0.15$ and the skewed distribution of SN(-40, 0.11, 0.13).

the negative pairs with high similarity. Considering the gradient of NT-Xent with respect to the similarity q_{ij} of a negative pair $N(z_i, z_j)$,

$$\frac{\partial L_{NT}(z_i)}{\partial q_{ij}} = \frac{1}{\tau} \frac{exp(q_{ij}/\tau)}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} \exp(q_{ik}/\tau)} \tag{6}$$

Fig. 2 (a) presents the gradients corresponding to different values of τ. When τ equals 1, the NT-Xent degenerates to the cross entropy loss. For the gradient to negative pairs with high similarity (the right part of the curve), when the τ becomes lower, its value is also increased. However, a high value of τ also depresses the gradient to negative pairs with medium similarity. Therefore, we choose $\tau = 0.15$ in default.

3.2 Improving Visual Cluster Separation by Gradient Redefinition

Preserving the neighborhood structures does not ensure sufficient separations between clusters for visual cluster analysis. Two discriminated clusters may still be close to each other in the embedding space and require effort to visually identify them. Therefore, we need to improve the visual cluster separation of embedding results.

As shown in Fig. 2(a), if negative pairs have low similarity, the gradients would be low as well. As a consequence, the contribution of these negative pairs to the optimization of embedding would be small. Given the fact that the points of a negative pair often belong to different clusters, we can increase their contribution to the embedding by increasing the gradients to these negative pairs. The increased contribution would result in a larger distance between the two clusters. Therefore, we redefine the gradients to negative pairs to enhance the visual cluster separation.

The challenge, however, lies in finding the appropriate part of negative pairs to redefine. Simply increasing the gradients to all negative pairs would break the balance between positive and negative pairs. The negative pairs with high similarity would also introduce errors into the training because they have the probability of belonging to the same cluster. Therefore, we decide to increase the gradient to the negative pairs with a high probability of being placed across two different clusters. We select 25 real-world datasets from UCI Repository [7], train 25 corresponding contrastive learning models, and count the negative pairs to obtain their statistics. Here we use class labels as the ground truth. Because the embedding has corrected part of false neighbors, we count the similarity q_{ij} in the embedding space rather than the probability p_{ij} in the high-dimensional space. Fig. 3 shows that most negative pairs have a similarity that is lower than 0.1. In such a situation, the number of pairs placed across two clusters is 10 times the number of pairs belonging to the same cluster.

Based on the statistics, we want to only increase the gradients to the negative pairs whose similarity is low and, meanwhile, keep the loss differentiable. The NT-Xent shows that an exponential gradient with respect to the negative similarity works well in separating negative pairs. We thus would like to add a similar but smaller distribution to the gradient in the low band of the similarity to separate clusters. However, directly adding a cut curve will lead to discontinuity in the gradient. Therefore, we choose a positively skewed distribution, which is similar to the exponential distribution on its left and continues with zero on its right. Specifically, the skewed distribution $SN(\eta, \mu, \sigma)$ is defined as

$$sn(q_{ij}) = \frac{2}{\sigma} \Theta\left(\frac{q_{ij}-\mu}{\sigma}\right) \Phi\left(\eta \frac{q_{ij}-\mu}{\sigma}\right) \tag{7}$$

where μ denotes the location, η is the skewness, and σ is the strictly positive scale. $\Theta(\cdot)$ is the standard normal probability density function, and $\Phi(\cdot)$ is the standard normal cumulative distribution function. In our implementation, we set μ as 0.11, σ as 0.13, and η as -40.

Based on the positively skewed distribution, we redefine the gradient to the negative pair $N(z_i, z_j)$ as

$$\frac{\partial \bar{L}(z_i)}{\partial q_{ij}} := \frac{1}{\tau} \frac{exp(q_{ij}/\tau)}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} \exp(q_{ik}/\tau)} + \alpha_i \cdot \frac{sn(q_{ij})}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} sn(q_{ik})} \tag{8}$$

where α_i is the weight parameter that balances the scales of two terms,

$$\alpha_i = 5 \cdot \frac{exp(q_{ij}^{max}/\tau)}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} \exp(q_{ik}/\tau)} \cdot \frac{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} sn(q_{ik})}{sn(q_{ij}^{max})} \tag{9}$$

where q_{ij}^{max} is the negative pair with the maximum gradient under positively skewed distribution. Finally, the curve of redefined gradient is shown in Fig. 2 (b).

3.3 Link-based Steering of Embedding

We introduce link-based interactions into the proposed CDR to steer the embedding results. *Must link* and *cannot link* constraints are widely used in interactive clustering algorithms because it is convenient to make use of the user's instance-level knowledge [79, 85]. *Must link* refers to specifying two points to be clustered into the same cluster. On the contrary, *cannot link* specifies two points that cannot be grouped in a single cluster in the clustering.

Given a pre-trained embedding, the steering process starts by specifying link-based constraints from users. Then, the probability p_{ij} between the points of the specified links is modified in the high-dimensional space. Specifically, we set the probability of *must link* pairs as 1 and that of cannot link pairs as 0. Next, the *must link* pairs, if specified, are added to the positive pairs. As the number of negative pairs is usually large, adding a few negative pairs would have little impact on the model. Therefore, we add the *cannot link* pairs into the positive pairs rather than negative pairs. During the training, we set the similarity of a *cannot link* pair (z_i, z_j) in the low-dimensional space as $1 - q_{ij}$ to push them apart. After the embedding is updated, users can specify more link constraints to refine the model iteratively.

3.4 Network Structure and Training Process

Network Structure. The input of our model are high-dimensional data points. For image data, we emply SimCLR [23] to transfer it into a 512-dimensional vector. Similar to SimCLR [23], our training model has an encoder that learns the representation of data and a projection

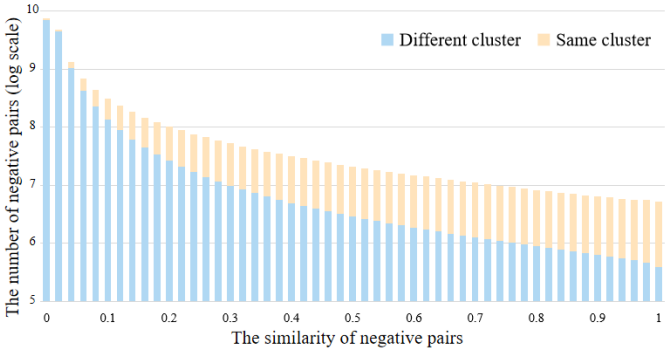


Fig. 3. The distribution of negative pairs in similarity. The stacked bars represent the ratio of two kinds of bars.

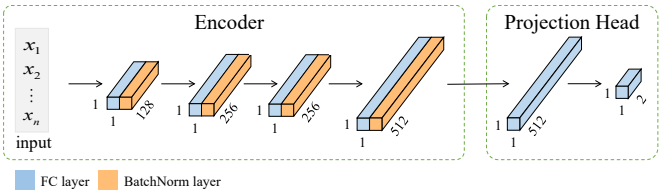


Fig. 4. The network structure of CDR.

head that embeds the representation into a 2-dimensional embedding space (Fig. 4). The encoder is a Dense Neural Network that consists of four densely-connected layers. The four layers contain 128, 256, 256, and 512 units, respectively. We add a batch normalization layer after each layer to avoid the gradient vanishing. A ReLU activation function is applied to each layer. The projection head contains a densely-connected layer with 512 units, a ReLU activation function, and a densely-connected layer with 2 units.

Training Process. We employ the Adam optimizer for network training, with an initial learning rate of 0.001. The learning rate is successively decayed by 10% when the number of epochs reaches $0.8E$ and $0.9E$. E denotes the total number of epochs, which is set as 1000 in our implementation. The batch size is set as $B = n/10$, where n is the size of the dataset. Because the trained model is used to embed the training dataset only, all data points are used for training. We adopt a multi-stage training strategy: 1) we use the NT-Xent to warm up the model for 150 epochs; 2) we train the model with gradient redefinition for 700 epochs; and 3) we use the NT-Xent again to obtain a stable embedding with 150 epochs. When users specify link-based constraints, we sample 30% data points for fine-tuning the model. We use fewer epochs than the initial training to update the embeddings. The default is 30 epochs. The first 85% epochs use the proposed gradient redefinition, and the last 15% epochs use the adapted NT-Xent.

3.5 The Prototype Interface

To demonstrate the effectiveness of the proposed interactive visual cluster analysis, we develop a visual interface that integrates CDR with a set of interactive visualizations (Fig. 5). The interface provides a control panel (Fig. 5(a)) for steering parameters. Embedding results are represented in a scatterplot view (Fig. 5(b)). If the dataset contains images, thumbnails of randomly sampled images can be displayed simultaneously in the picture view. In addition, a parallel coordinates plot is utilized to explore the high-dimensional features of data points (Fig. 5(c)). If the dimensionality of a dataset is too high, users can apply Principal Component Analysis [1] to reduce the dimensionality so that only the major information is represented. The link board records the link-based constraints specified by users (Fig. 5(d)). The context of each constrained link is displayed in a thumbnail, where the link is highlighted in a heatmap of the scatterplot view. The four views are coordinated to support the CDR while allowing users to interactively examine the dimensional features and steering the embedding results through link-based interactions.

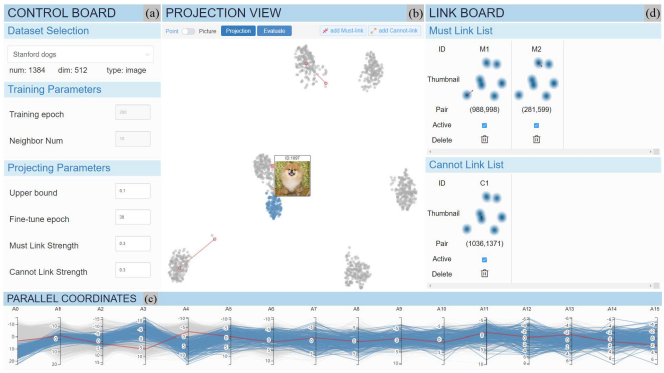


Fig. 5. Interactive visual cluster analysis with CDR.

4 Evaluation and Results

4.1 Quantitative Evaluation

In this section, we use quantitative experiments to evaluate CDR without user steering. The experimental environment uses a desktop PC with Intel Core i9-7900X (3.30 GHz), NVIDIA GeForce GTX 1080 Ti, 128 GB memory, and Windows 10 installed.

Datasets. We use 10 real-world datasets including Animals [26], Cifar10 [46], Indian Food [4], Isolet [33], MNIST [51], Stanford Dogs [44], Texture [7], USPS [38], Weathers [77] and WiFi [12]. The types of data contain image data, tabular data, and text data. All of them have at least two clusters and have class labels as the ground truth.

DR techniques. For comparison, we select five non-parametric DR techniques , i.e., ISOMAP [73], Landmark ISOMAP (LISOMAP) [70], t-SNE [76], AtSNE [34], and UMAP [55]; and four parametric DR techniqcues, i.e., Parametric t-SNE (PtSNE) [75], Parametric UMAP (PUMAP) [67], Deep Recursive Embedding (DRE) [92], and the Dimensionality Reduction by Learning an Invariant Mapping (DrLIM) [36]. Although there are many contrastive learning applications in computer vision, to the best of our knowledge, DrLIM is the only contrastive learning technique for general DR tasks. We employ the implementation of ISOMAP, t-SNE, and UMAP in the sklearn library. For other techniques, we use the code published by their authors. Because we cannot find the code of LISOMAP, we implement it according to the paper. In the ablation experiments, we test the contrastive learning framework with Cross Entropy (CE), NT-Xent (NX-CDR), and gradient redefinition (CDR). All results are without user steering.

Parameters. We set the *perplexity* of t-SNE , AtSNE and PtSNE as 30, and the size of kNN neighbor in UMAP, PUMAP, ISOMAP and LISOMAP as $k = 15$, following the settings in previous studies [55, 67, 76, 84]. The other parameters of these techniques are set as their default values. In consistency with UMAP, we set $k = 15$ in CE, NX-CDR, and CDR. UMAP, t-SNE and AtSNE are optimized with 1000 iterations to ensure convergence on all tested datasets. For all the parametric techniques, we use the same network architecture as described in Section 3.4 and also trained them for 1000 epochs.

4.1.1 Measures

From literature review [32, 68], we employ six measures to evaluate the DR techniques. Specifically, we select *Trustworthiness* & *Continuity* [43] for preserving neighborhood structures, *kNN classifier accuracy* [27] & *Neighbor Hit* [59] for correcting false neighbors, and *Distance Consistency* [71] & *Silhouette Coefficient* [65] for visual cluster separation. For the first four measures that need to specify the kNN, we set $k = 15$, which is consistent with the tested techniques.

Trustworthiness measures how much the kNN neighborhood of a point in the embedding space reflects the true neighborhood in the high-dimensional space.

Continuity measures how much the kNN neighborhood of a point in the high-dimensional space is preserved in the embedding space.

***k-NN classifier accuracy* (kNN-CA)** measures the accuracy of the kNN-based classification in the embedding space. For each point, we predict its class label by majority voting of its kNN in the embedding space. A high kNN-CA close to 1 indicates a high consistency between points and their kNN in terms of classification.

***Neighbor Hit* (NH)** measures the proportion of points in the kNN that belong to the same cluster as their center point in the embedding space. A high NH that is close to 1 indicates good quality. On the contrary, a low NH that is close to 0 indicates that many false neighbors are preserved in the embedding space.

***Distance Consistency* (DSC)** is the proportion of points which have the same class as their nearest class centroids. DSC is the best separation measure according to Sedlmair and Aupetit [68].

***Silhouette Coefficient* (SC)** measures the difference of between-class and within-class average distances normalized by the maximum of them. Compared to DSC which measures the point-class relationship, SC evaluates separation from a class-class perspective. It is also widely used in measuring visual class separation.

Except for SC, which is ranged from -1 to 1, the other five measures are normalized to $[0, 1]$. Higher value refers to higher quality for all six

Table 1. Comparisons of Trustworthiness

Dataset	ISOMAP	LISOMAP	t-SNE	AtSNE	UMAP	PtSNE	PUMAP	DRE	DrLIM	CE	NX-CDR	CDR
Animals	0.755	0.762	**0.969**	0.958	0.932	0.956	0.927	0.872	0.829	0.927	0.966	0.968
Cifar10	0.760	0.736	**0.974**	0.964	0.944	0.966	0.938	0.845	0.868	0.935	0.968	0.972
Indian Food	0.805	0.783	0.982	0.980	0.972	0.974	0.966	0.867	0.934	0.960	0.982	**0.983**
Isolet	0.926	0.907	0.980	0.972	0.975	0.973	0.970	0.705	0.932	0.974	0.977	**0.981**
MNIST	0.755	0.760	**0.978**	0.968	0.958	0.968	0.947	0.872	0.786	0.953	0.974	0.977
Stanford Dogs	0.858	0.843	**0.972**	0.967	0.963	0.965	0.958	0.677	0.937	0.954	0.971	0.971
Texture	0.964	0.955	**0.997**	0.992	0.994	0.993	0.993	0.944	0.974	0.991	0.996	0.996
USPS	0.833	0.831	**0.985**	0.980	0.970	0.981	0.965	0.911	0.875	0.969	0.980	0.982
Weathers	0.795	0.797	0.942	0.931	0.934	0.926	0.912	0.707	0.851	0.923	0.950	**0.952**
WiFi	0.937	0.925	**0.982**	0.980	0.976	0.965	0.964	0.884	0.956	0.969	0.980	**0.982**
Average	0.839	0.830	0.976	0.969	0.962	0.967	0.954	0.828	0.894	0.956	0.974	**0.977**

Table 2. Comparisons of Continuity

Dataset	ISOMAP	LISOMAP	t-SNE	AtSNE	UMAP	PtSNE	PUMAP	DRE	DrLIM	CE	NX-CDR	CDR
Animals	0.935	0.929	0.955	0.951	0.957	0.957	**0.960**	0.930	0.957	0.947	0.947	0.948
Cifar10	0.946	0.923	0.976	0.968	0.973	0.976	0.976	0.953	**0.977**	0.966	0.970	0.966
Indian Food	0.941	0.920	0.970	0.970	0.970	**0.974**	0.972	0.919	0.971	0.961	0.967	0.965
Isolet	0.944	0.932	0.976	0.975	0.979	**0.982**	0.980	0.819	0.978	0.981	0.981	0.980
MNIST	0.949	0.941	**0.967**	0.966	**0.967**	0.966	0.966	0.937	0.953	0.957	0.962	0.962
Stanford Dogs	0.940	0.926	0.965	**0.967**	0.963	0.965	0.964	0.812	0.963	0.938	0.960	0.959
Texture	0.985	0.981	0.992	0.989	0.990	**0.994**	0.993	0.969	0.993	0.989	0.991	0.990
USPS	0.967	0.954	0.978	0.979	0.980	**0.983**	0.981	0.963	0.973	0.978	0.980	0.979
Weathers	0.896	0.874	0.951	0.943	0.952	0.954	0.954	0.809	0.929	0.943	**0.955**	0.951
WiFi	0.963	0.948	0.976	0.968	0.978	0.978	**0.979**	0.916	**0.979**	0.974	0.979	0.978
Average	0.947	0.933	0.971	0.967	0.971	**0.973**	0.972	0.903	0.967	0.964	0.969	0.968

Table 3. Comparisons of kNN-CA

Dataset	ISOMAP	LISOMAP	t-SNE	AtSNE	UMAP	PtSNE	PUMAP	DRE	DrLIM	CE	NX-CDR	CDR
Animals	0.586	0.600	0.883	0.875	0.886	0.906	0.888	0.879	0.754	0.908	0.904	**0.905**
Cifar10	0.636	0.660	0.939	0.912	0.933	0.940	0.943	0.920	0.856	0.939	0.943	**0.945**
Indian Food	0.647	0.618	0.955	0.957	0.947	0.955	0.956	0.877	0.900	0.954	0.958	**0.958**
Isolet	0.901	0.906	0.990	0.985	0.991	0.988	0.983	0.545	0.640	0.991	0.994	**0.995**
MNIST	0.524	0.560	0.943	0.935	0.935	0.931	0.897	0.838	0.501	0.953	0.958	**0.964**
Stanford Dogs	0.795	0.728	0.966	0.972	0.972	0.974	0.961	0.506	0.938	0.803	0.971	**0.978**
Texture	0.842	0.796	0.980	0.940	0.972	0.945	0.955	0.942	0.757	0.974	0.983	**0.986**
USPS	0.704	0.699	0.992	0.986	0.987	0.981	0.964	0.931	0.740	0.991	0.994	**0.996**
Weathers	0.822	0.880	0.980	0.981	0.981	**0.983**	0.982	0.721	0.944	0.982	**0.983**	**0.983**
WiFi	0.984	0.978	**0.986**	0.984	0.984	0.984	0.985	0.965	0.975	0.983	0.984	0.985
Average	0.744	0.743	0.961	0.953	0.959	0.959	0.951	0.812	0.800	0.948	0.967	**0.970**

Table 4. Comparisons of Neighbor Hit

Dataset	ISOMAP	LISOMAP	t-SNE	AtSNE	UMAP	PtSNE	PUMAP	DRE	DrLIM	CE	NX-CDR	CDR
Animals	0.491	0.514	0.841	0.834	0.842	0.863	0.851	0.827	0.679	0.861	0.863	**0.869**
Cifar10	0.568	0.583	0.914	0.883	0.913	0.915	0.922	0.891	0.815	0.920	0.924	**0.926**
Indian Food	0.579	0.539	0.933	0.934	0.937	0.932	0.937	0.815	0.857	0.934	0.941	**0.943**
Isolet	0.878	0.869	0.983	0.967	0.985	0.977	0.976	0.393	0.569	0.982	0.985	**0.988**
MNIST	0.434	0.474	0.911	0.904	0.892	0.895	0.843	0.751	0.427	0.928	0.932	**0.939**
Stanford Dogs	0.720	0.665	0.943	0.942	0.957	0.956	0.950	0.342	0.903	0.741	0.956	**0.965**
Texture	0.795	0.731	0.963	0.916	0.957	0.913	0.933	0.920	0.691	0.964	0.971	**0.979**
USPS	0.610	0.617	0.986	0.979	0.977	0.968	0.941	0.896	0.671	0.990	0.991	**0.993**
Weathers	0.771	0.839	0.964	0.951	0.968	**0.972**	**0.972**	0.594	0.909	0.968	0.971	**0.972**
WiFi	0.975	0.969	0.976	0.975	0.977	0.976	0.973	0.949	0.965	0.972	0.974	**0.978**
Average	0.682	0.680	0.941	0.928	0.940	0.937	0.930	0.738	0.749	0.926	0.951	**0.955**

Table 5. Comparisons of DSC

Dataset	ISOMAP	LISOMAP	t-SNE	AtSNE	UMAP	PtSNE	PUMAP	DRE	DrLIM	CE	NX-CDR	CDR
Animals	0.557	0.580	0.846	0.733	0.862	0.869	0.875	0.828	0.739	0.885	0.878	**0.895**
Cifar10	0.505	0.565	0.904	0.714	0.930	0.870	0.938	0.880	0.830	0.923	0.903	**0.941**
Indian Food	0.600	0.585	0.935	0.901	0.940	0.926	0.943	0.793	0.848	0.925	0.938	**0.956**
Isolet	0.859	0.887	0.985	0.948	0.987	0.976	0.981	0.277	0.573	0.987	0.984	**0.990**
MNIST	0.488	0.512	0.835	0.821	0.899	0.816	0.843	0.754	0.470	0.929	0.940	**0.961**
Stanford Dogs	0.792	0.748	0.962	0.965	0.968	0.967	0.960	0.387	0.926	0.598	0.970	**0.978**
Texture	0.691	0.642	0.895	0.853	0.935	0.808	0.888	0.892	0.700	0.941	0.953	**0.982**
USPS	0.641	0.610	0.975	0.905	0.979	0.941	0.957	0.890	0.705	0.990	0.990	**0.995**
Weathers	0.769	0.859	0.976	0.961	0.979	**0.983**	**0.983**	0.689	0.980	0.979	0.979	**0.983**
WiFi	0.968	0.974	**0.984**	0.974	0.983	0.980	0.982	0.961	0.969	0.982	0.975	**0.984**
Average	0.687	0.696	0.930	0.878	0.946	0.914	0.935	0.735	0.769	0.914	0.951	**0.966**

Table 6. Comparisons of SC

Dataset	ISOMAP	LISOMAP	t-SNE	AtSNE	UMAP	PtSNE	PUMAP	DRE	DrLIM	CE	NX-CDR	CDR
Animals	0.027	0.062	0.284	0.203	0.372	0.336	0.373	0.437	0.171	0.515	0.376	**0.566**
Cifar10	0.013	0.022	0.340	0.172	0.453	0.354	0.467	0.593	0.213	0.624	0.442	**0.674**
Indian Food	0.084	0.072	0.466	0.321	0.610	0.489	0.575	0.390	0.248	0.652	0.502	**0.712**
Isolet	0.489	0.487	0.596	0.403	0.777	0.575	0.702	-0.102	0.089	0.695	0.558	**0.850**
MNIST	-0.005	0.037	0.313	0.241	0.417	0.322	0.336	0.401	-0.001	0.588	0.487	**0.677**
Stanford Dogs	0.237	0.205	0.580	0.424	0.692	0.562	0.641	-0.081	0.361	0.156	0.599	**0.816**
Texture	0.250	0.255	0.471	0.390	0.564	0.423	0.515	0.664	0.192	0.697	0.564	**0.788**
USPS	0.146	0.195	0.431	0.339	0.650	0.468	0.549	0.636	0.156	0.699	0.633	**0.787**
Weathers	0.350	0.415	0.549	0.404	0.674	0.509	0.653	0.130	0.404	0.806	0.616	**0.831**
WiFi	0.634	0.733	0.640	0.445	0.735	0.715	0.706	0.744	0.510	0.755	0.596	**0.778**
Average	0.222	0.248	0.467	0.334	0.594	0.475	0.552	0.381	0.234	0.619	0.537	**0.748**

Table 7. Comparisons of running time(s)

Dataset	ISOMAP	LISOMAP	t-SNE	AtSNE	UMAP	PtSNE	PUMAP	DRE	DrLIM	CE	NX-CDR	CDR
Animals	140.65	119.55	2385.71	**4.96**	9.58	560.75	693.00	444.58	215.56	596.90	661.31	679.23
Cifar10	135.21	110.19	2377.23	**4.96**	9.74	551.64	703.54	467.11	215.82	631.82	608.24	640.47
Indian Food	15.42	12.82	306.13	**4.01**	11.82	482.40	587.94	460.05	132.76	264.05	286.68	305.59
Isolet	2.62	**2.31**	589.23	3.80	4.61	275.58	385.18	350.80	137.08	139.90	153.70	147.60
Mnist	173.41	147.71	2475.91	**5.16**	8.86	683.59	930.89	583.75	305.94	806.33	852.70	895.81
Stanford dogs	2.25	**1.96**	44.85	3.57	4.86	249.72	295.91	309.15	82.10	97.48	102.83	103.50
Texture	5.99	**4.09**	468.51	4.14	5.23	376.78	441.98	303.07	110.20	258.62	272.12	263.68
USPS	42.74	29.17	1295.85	**4.22**	6.80	410.11	726.78	287.83	140.29	406.73	416.77	423.90
Weathers	0.87	**0.75**	20.16	3.47	3.34	323.61	544.82	390.39	96.90	145.35	142.93	154.98
Wifi	0.83	**0.37**	61.98	3.70	4.61	278.31	403.53	341.48	91.92	154.56	153.84	162.90
Average	52.00	42.89	1002.56	**4.26**	6.94	419.25	571.35	393.82	152.86	348.72	363.68	376.22

measures. It is worth noting that we use class labels as the ground truth clusters in kNN-CA, NH, DSC, and SC. Although this is a common practice in evaluating DR techniques [60, 92], the possible class-cluster mismatching [8] should be aware. In our experiments, we choose datasets with well-known cluster structures to address this issue.

4.1.2 Comparison Results

Table 1 to Table 6 present the quantitative comparison of well-known existing techniques with CDR on 10 datasets with respect to each of the six measures. From Table 1 and Table 2, it is observed that CDR achieves the best average *Trustworthiness* than other techniques, but its average *Continuity* is slightly lower than t-SNE, UMAP, PtSNE, and PUMAP. The main reason is that the calculation of *Continuity* includes false neighbors. Removing false neighbors in the embedding space, although improves the embedding quality, results in a decrease of *Continuity*. This is verified by the results on correcting false neighbors. According to Table 3 and Table 4, CDR achieves the best performance in most datasets in terms of *kNN-CA* and *Neighbor Hit*, demonstrating its improved neighborhood accuracy and ability to correct false neighbors. In terms of visual cluster separation, Table 5 and Table 6 show that CDR outperforms all other techniques in DSC and SC. Fig. 6 shows the embedding results of the selected techniques on three typical datasets. CDR presents a much clearer visual cluster separation in the embedding results. In conclusion, CDR achieves comparable quality to t-SNE and UMAP in terms of preserving neighborhood structures. Considering the false neighbors, CDR outperforms all other techniques in terms of preserving correct neighborhood structures. CDR also achieves the best performance in terms of visual cluster separation.

4.1.3 Results of Ablation Experiments

We compare the performance of CE, NX-CDR, and CDR to evaluate the effectiveness of the adapted NT-Xent and the proposed gradient redefinition. In terms of preserving neighborhood structures, NX-CDR performs better than CE. CDR achieves higher average *Trustworthiness* but lower average *Continuity* than NX-CDR (see Table 1 and Table 2). In terms of correcting false neighbors, NX-CDR also performs better than CE. CDR performs better than NX-CDR (see Table 3 and Table 4). Again, the results suggest that *Continuity* is distorted by false neighbors. In terms of visual cluster separation, CDR outperforms CE and NX-CDR in DSC and SC for all datasets (see Table 5 and Table 6). In conclusion, the adapted NT-Xent has positive effects on preserving correct neighborhood structures. The proposed gradient redefinition is effective in improving visual cluster separation.

4.1.4 Runtime Performance

Table 7 shows the running time of the evaluated techniques. Among them AtSNE and UMAP are the fastest with the cost of slightly decreased quality than t-SNE. CDR runs faster than original t-SNE, but much slower than other non-parametric techniques. Among parametric techniques, CDR runs faster than PtSNE and PUMAP, which are the parametric version of t-SNE and UMAP. In our experiments, we use the same backbone network in DRE and CDR. Therefore, they have similar time performances. DrLIM achieves better time performance than CDR because it has a simpler loss function. Generally, parametric techniques run slower than non-parametric ones due to their training process. However, it is worthwhile to develop parametric techniques not only because of the better embedding qualities, but also parametric techniques offer several opportunities to DR applications. We will discuss these opportunities in Section 5.

4.2 User Study

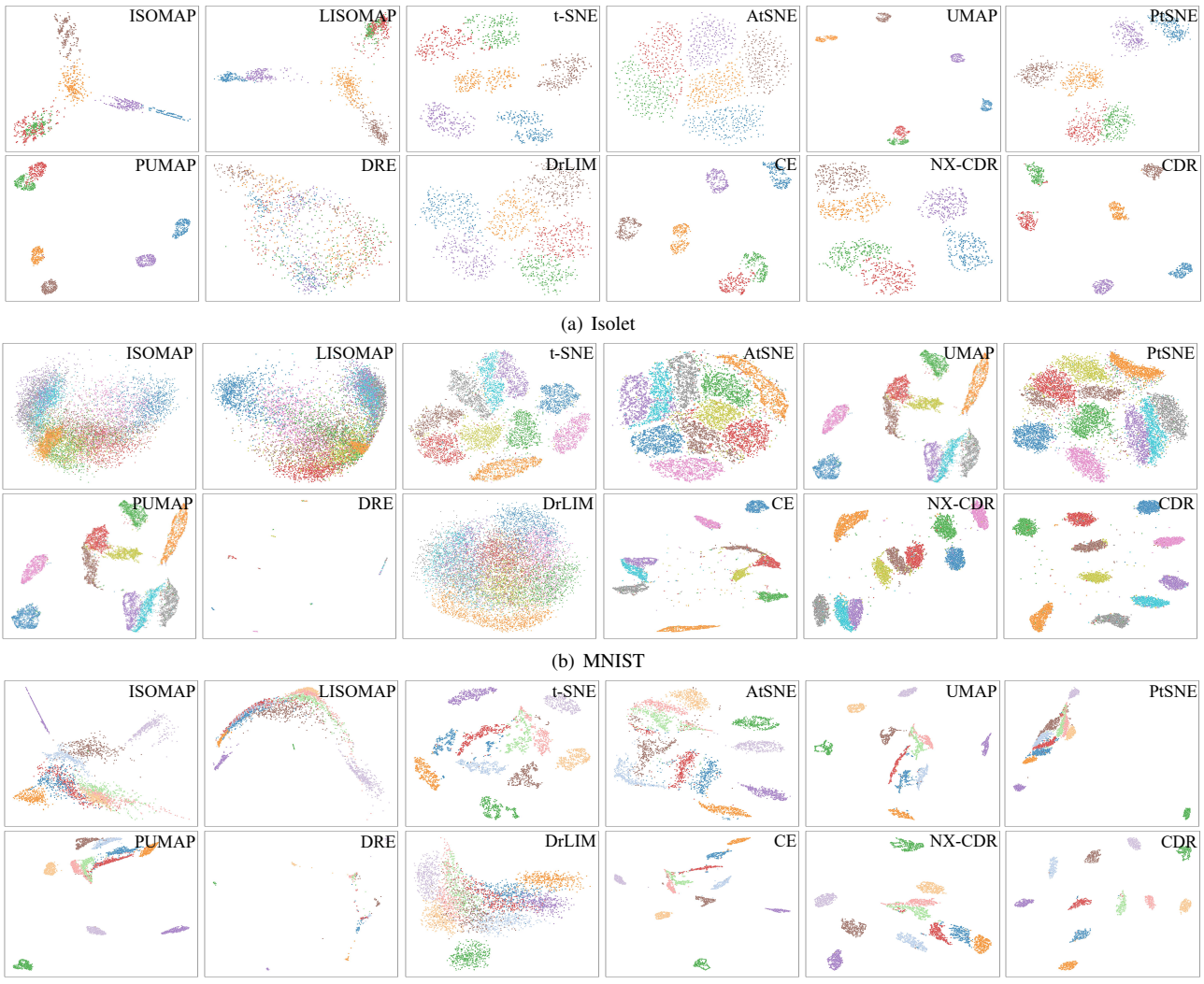
We have conducted a user study to evaluate the ability of CDR without user steering to present well-separated clusters in terms of visual perception. In particular, we focus on the cluster identification task to test whether projected clusters are well separated and easily recognized by users. We consider two state-of-the-art DR techniques, t-SNE and UMAP, as baseline techniques and compare them with CDR. We put forward two hypotheses to verify:

- **H1:** CDR performs better in identifying clusters;
- **H2:** CDR consumes less time for identifying clusters;

Tasks. We designed a controlled experiment for evaluating the performance of cluster identification. They were tested using the three DR techniques on 20 datasets. 10 of them are the same as described Section 4.1. The remainings are described in the appendix. The controlled experiment was based on the cluster identification task (**E1**) described in Xia et al. [84]. We followed the same task design and asked each participant to complete $3(techniques) \times 20(datasets) = 60$ trials for identifying clusters.

Participants, Apparatus, and Procedure. We recruited 20 participants (16 males and 4 females) for the user study, aged 23 to 27. None of them reported color blindness or color weakness. All the participants are graduate students with research experience in data visualization. To remove any bias from the analysis processes, we adopted the same testing application described in Xia et al. [84] and integrated CDR and the baseline techniques. All the participants took the experiments one by one on standard desktop computers in our research lab. Each computer is equipped with a standard keyboard, a mouse, and a DELL 27-inch U2720QM monitor. The computer has a 2,560×1,440 screen resolution and chrome browser. The two experiments were conducted based on the same procedure described in Xia et al. [84]. In the controlled experiment, instead of randomly selecting datasets and DR techniques for each trial, we adopted a Latin square design where the datasets were sorted based on a Latin square upon which one of the three DR techniques was randomly selected for a trial. Compared with Xia et al. [84], this design minimizes the potential learning effects resulting from visualizing the datasets in certain orders. To evaluate the results of the two experiments, we adopted similar analysis conditions, statistical measures, and analysis approaches described in Xia et al. [84]. Here, the Shapiro-Wilk test and Paired Wilcoxon test were added to validate the normality in precision/recall and completion time, respectively.

Results. The results of precision and recall values for the controlled experiment are displayed in Fig. 7(a) and Fig. 7(b). Among the three techniques, CDR has the best precision (80.68%) and recall (69.29%). There is a relatively small difference between t-SNE and UMAP on precision (65.64% and 66.25%, respectively) and recall (55.49% and 55.50%, respectively). The Friedman tests show statistical significance of precision ($\mathscr{X}^2(2) = 14.8, p < 0.05$) and recall($\mathscr{X}^2(2) = 17.5, p < 0.05$). Therefore, hypothesis **H1** is confirmed. In terms of the average completion time (Fig. 7(c)), both CDR and UMAP show better performance (14.481s and 14.603s, respectively) than t-SNE (16.592s). The Friedman tests show statistical



(a) Isolet

(b) MNIST

(c) Texture

Fig. 6. Comparison among different techniques on datasets Isolet, MNIST and Texture.

significance of average completion time ($\mathcal{X}^2(2) = 10.8, p < 0.05$). However, although CDR has a slightly shorter completion time than UMAP, there is no statistical significance between them. As a consequence, the hypothesis **H2** was partially confirmed.

We conducted a post-interview to subjectively compare CDR with t-SNE and UMAP by reviewing their results side by side for each dataset. All participants agree that CDR performs better than t-SNE and UMAP in terms of presenting visually separated cluster patterns. They point out that the clearer visual separation makes them more confident in identifying clusters. However, they also feel somewhat uncertain about the results, as the ground truth is unknown. Some participants point out that providing the quality measures, such as *Trustworthiness* and *Continuity*, can strengthen their confidence.

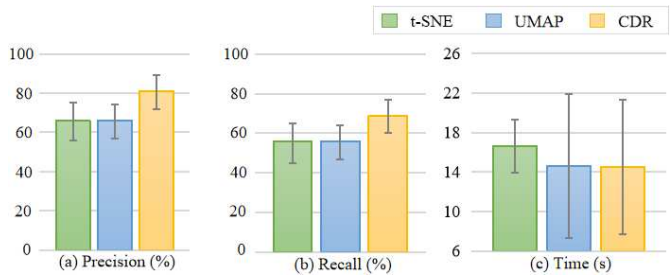


Fig. 7. The results of the user study. (a)–(c) The statistics of precision, recall, and completion time for the objective task.

4.3 Case Studies

In this section, we demonstrate how users can use the visual interface and the link-based interactions to steer the embeddings for visual cluster analysis. Using the visual interface, we carried out analyses on two datasets: an image dataset Animals [26] and a tabular dataset WiFi [12].

Animals. The Animals dataset contains 10000 animal images. Each image is encoded into a 512-dimensional feature vector as mentioned in Section 3.4. Using the default parameters, the data were embedded into the scatterplot view, as shown in Fig. 1(e). 11 clusters were observed. However, there was a much smaller cluster at the top left corner. Hoovering over points in this cluster and its neighboring two clusters, we found both the small cluster and the cluster at the top left corner contained butterfly images. Switched to the image mode, we found that the larger cluster contained images where butterflies dominate the foreground, while the smaller cluster contained images where the backgrounds dominate. That is why they formed two clusters. However, considering the interest is in animals, we would like to merge the two clusters into one. Therefore, we added a *must link* between the two clusters. As merging two clusters is essential to align the two clusters' centers, we chose the two ends of the *must link* to be the butterfly images near the two cluster centers. After re-embedding, the two clusters were successfully merged (Fig. 1(g)). Some spider images were observed at the boundary of the merged cluster (e.g., Fig. 1(g) A). We thus further improved the results by adding more links. We first added a *cannot link* between one of the spider images at the boundary and a nearby butterfly image to push away the mis-clustered spider

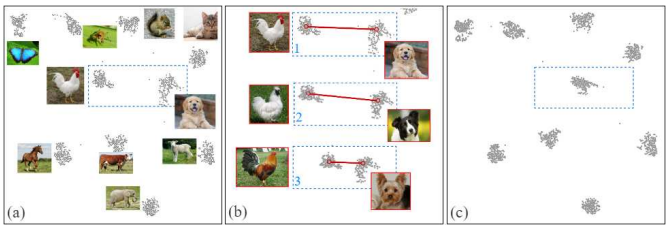


Fig. 8. The case of the Animals dataset. (a) The species of animals in each cluster. (b) Three *must links* are imposed between chicken and dog cluster in three interactions. (c) Updated embedding result.

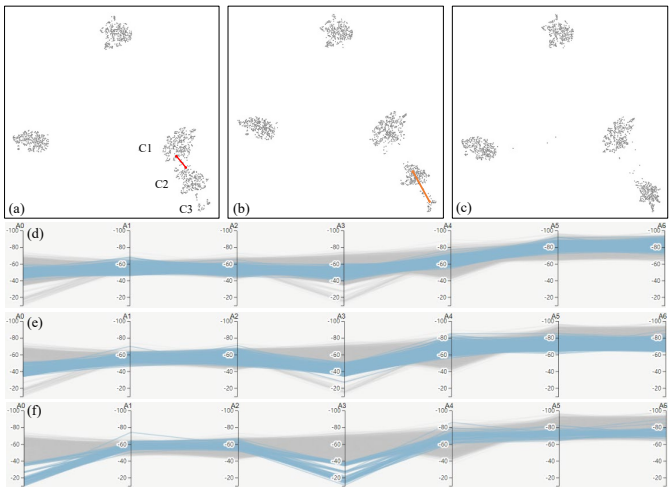


Fig. 9. The case of the WiFi dataset. (a)–(c) The initial embedding, embedding after one *cannot link*, and embedding after one *must link*, respectively. (d)–(f) The parallel coordinates plots of C1, C2, and C3, respectively.

images. We then added a *must link* between one of the mis-clustered spider images and a spider image near the center of the spider cluster to attract them together. Fig. 1(h) shows the results. With only the three links added, our method successfully merged the two butterfly clusters and separated the spider images. This demonstrated the effectiveness of adding links to improve clustering results. However, added links are expected to assist the similarity measure to improve the clustering results, rather than being the dominant factor to determine the clustering results. We thus expect some resistance of our method to "bad" links added. To verify this, we added a *must link* between two significantly different clusters: chicken and dog. As shown at the top of Fig. 8(b), the overall clustering pattern remained unchanged. It was not until three *must links* were added (Fig. 8(b)) then the chicken and the dog clusters were merged (Fig. 8(c)), in comparison to the only one *must link* needed to merge the two butterfly clusters. This demonstrated some degree of resistance of our method to "bad" links. Meanwhile, the *Trustworthiness* and *Continuity* measures were also significantly decreased (from 0.910 to 0.757 and 0.910 to 0.853, respectively).

WiFi. We then moved on to analyze the clustering results on the WiFi dataset. This dataset contains 2000 vectors recording the signal strengths at 2000 locations in four rooms (500 locations in each room). There are 7 routers in the space, from which the received signal strengths constitute the 7 dimensions of each vector. A good clustering of these vectors is expected to correspond to the four different rooms. Fig. 9 (a) shows the embedding results using the default parameters. Five clusters were observed rather than four as expected. The problem seemed to lie in the three clusters (C1, C2, C3) that were closely positioned at the bottom right. We would like to find the reasons and thus had a closer examination of their parallel coordinates. It is observed that the data points in C1 and C2 did exhibit similar patterns in parallel coordinates (Fig. 9 (d), (e)). The distributions along the dimensions had large overlaps, especially along A0, A1, A2, A3, and A4. This means a large number of data points (locations) in C1 and C2 had similar distances to these routers. We considered that is why C1 and C2 were closely positioned in the embedding space. To refine the CDR to better distinguish locations in the two clusters, we added a *cannot link*. The re-embedding results showed the successful separation of the two clusters (Fig. 9 (b)). We then examined C3. C3 had much smaller number of data points, which indicated it was separated from its main cluster. Comparing the parallel coordinates between C3 (Fig. 9 (f)) and C1/C2, it can be observed that C3 had very different distributions along A0 and A3 compared with C1 and C2. However, along the other 5 dimensions, the distributions of C3 were fully enclosed by the distributions of C2. Together with the closest positions of C3 and C2 in the embedding space, these were strong indications that the locations in C3 should be in the same room as those in C2. Their differences along A0 and A3 explained their separation, but these were more likely due to obstacles in the room. Based on the reasoning, we would like to refine the CDR to merge C2 and C3. We thus added a *must link* between C2 and C3. After re-embedding, the results had four well -separated clusters as expected (Fig. 9 (c)), demonstrating the improved dimensionality reduction for visual cluster analysis.

Lesson Learned. Our case studies show that the link-based interaction is very efficient. To merge similar clusters or separate dissimilar clusters, only one or two links are needed. Regarding "bad" links, such as must links between dissimilar clusters, our technique shows some degree of resistance. Three or more links are required for counterfactual interactions. Users should be aware of the power of the interactions when exploring cluster patterns. A suggestion is using the embedding quality measures as the indication of whether the interaction is correct.

5 Discussions

Opportunities of Parametric Techniques. Although parametric techniques are generally slower than non-parametric ones, they provide an explicit representation of the learned manifold mapping, which can be reused in new datasets and shared among multiple users. This brings several application opportunities for parametric techniques, such as fast embedding of new out-of-sample data [67,75], federated learning-based joint projection [83], and reusing edited embedding on similar datasets. These applications are non-trivial for non-parametric techniques.

Comparison between parametric and non-parametric DR techniques. On the one hand, the parametric DR techniques run much slower than non-parametric ones, such as AtSNE and UMAP. The main reason is that the training process for parametric techniques is time-consuming. On the other hand, the proposed CDR achieves better embedding quality than all evaluated techniques in terms of both accuracy and visual cluster separation. Critical issues of non-parametric techniques, including the false neighbors and missing neighbors. With well-designed contrastive settings, we alleviated this issue and thus improved the embedding quality. We also would like to point out that only a parametric framework cannot address the issues of false neighbors and missing neighbors. For example, PtSNE [75] and PUMAP [67] performs worse than their non-parametric counterparts, t-SNE, and UMAP, respectively. The proposed CDR is the first parametric DR technique that achieves comparable embedding quality with state-of-the-art non-parametric techniques.

6 Conclusion

In this paper, we propose an interactive visual cluster analysis approach. We introduce contrastive learning into dimensionality reduction, propose a gradient redefinition technique, and enable link-based interactions on the embedding. The experiments and user study show that the proposed CDR outperforms existing DR techniques in preserving neighborhood structures, correcting false neighbors, and visually separating clusters. Case studies demonstrate the effectiveness of the link-based interactions.

Acknowledgments

The authors would like to thank the helpful comments from the anonymous reviewers. This work is supported by the National Natural Science Foundation of China (No. 61872389, 62132017, 61872388), and the High Performance Computing Center of Central South University.

REFERENCES

[1] H. Abdi and L. J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] G. Andrienko and N. Andrienko. Constructing parallel coordinates plot for problem solving. In *1st International Symposium on Smart Graphics*, pp. 9–14, 2001.

[3] G. Andrienko and N. Andrienko. Interactive cluster analysis of diverse types of spatiotemporal data. *Acm Sigkdd Explorations Newsletter*, 11(2):19–28, 2010.

[4] M. Anshul. Indian food. `https://www.kaggle.com/datasets/anshulmehtakaggl/themassiveindianfooddataset`, 2020. Accessed November, 2020.

[5] D. Araújo, A. Dória Neto, A. Martins, and J. Melo. Comparative study on dimension reduction techniques for cluster analysis of microarray data. In *The 2011 International Joint Conference on Neural Networks*, pp. 1835–1842, 2011.

[6] İ. Arın, M. K. Erpam, and Y. Saygın. I-twec: Interactive clustering tool for twitter. *Expert Systems with Applications*, 96:1–13, 2018.

[7] A. Asuncion and D. Newman. Uci machine learning repository, 2007. `https://archive.ics.uci.edu/ml/index.php`.

[8] M. Aupetit. Sanity check for class-coloring-based evaluation of dimension reduction techniques. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pp. 134–141, 2014.

[9] M. Aupetit, M. Sedlmair, M. M. Abbas, A. Baggag, and H. Bensmail. Toward perception-based evaluation of clustering techniques for visual analytics. In *Proceedings of IEEE Visualization Conference*, pp. 141–145, 2019.

[10] S. Basu, D. Fisher, S. Drucker, and H. Lu. Assisting users with clustering tasks by combining metric learning and classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 24, pp. 394–400, 2010.

[11] R. Bekkerman, H. Raghavan, J. Allan, and K. Eguchi. Interactive clustering of text collections according to a user-specified criterion. In *IJCAI*, vol. 7, pp. 684–689, 2007.

[12] R. Bhatt. Fuzzy-rough approaches for pattern classification: Hybrid measures, mathematical analysis, feature selection algorithms, decision tree algorithms, neural learning, and applications. *Decision tree algorithms, neural learning, and applications. Amazon books*, 2005.

[13] M. Blumenschein, X. Zhang, D. Pomerenke, D. A. Keim, and J. Fuchs. Evaluating reordering strategies for cluster identification in parallel coordinates. *Computer Graphics Forum*, 39(3):537–549, 2020.

[14] L. Boudjeloud-Assala, P. Pinheiro, A. Blansché, T. Tamisier, and B. Otjacques. Interactive and iterative visual clustering. *Information Visualization*, 15(3):181–197, 2016.

[15] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *2012 IEEE conference on visual analytics science and technology (VAST)*, pp. 83–92, 2012.

[16] P. Bruneau and B. Otjacques. Including semi-supervision in a kernel matrix, with a view to interactive visual clustering. working paper or preprint, Nov. 2012.

[17] P. Bruneau and B. Otjacques. A Proposition of Interactive Visual Clustering System. In *EuroVis Workshop on Visual Analytics using Multidimensional Projections*, 2013. doi: 10.2312/PE.VAMP.VAMP2013.001-004

[18] M. Cavallo and Ç. Demiralp. Clustrophile 2: Guided visual clustering analysis. *IEEE transactions on visualization and computer graphics*, 25(1):267–276, 2018.

[19] A. Chatzimparmpas, R. M. Martins, and A. Kerren. t-visne: Interactive assessment and interpretation of t-sne projections. *IEEE transactions on visualization and computer graphics*, 26(8):2696–2714, 2020.

[20] K. Chen and L. Liu. Validating and refining clusters via visual rendering. In *Third IEEE International Conference on Data Mining*, pp. 501–504, 2003.

[21] K. Chen and L. Liu. Vista: Validating and refining clusters via visualization. *Information Visualization*, 3(4):257–270, 2004.

[22] K. Chen and L. Liu. ivibrate: Interactive visualization-based framework for clustering large datasets. *ACM Transactions on Information Systems (TOIS)*, 24(2):245–294, 2006.

[23] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607, 2020.

[24] J. Choo, H. Lee, Z. Liu, J. Stasko, and H. Park. An interactive visual testbed system for dimension reduction and clustering of large-scale high-dimensional data. *Visualization and Data Analysis*, 8654:1–15, 2013.

[25] A. Coden, M. Danilevsky, D. Gruhl, L. Kato, and M. Nagarajan. A method to accelerate human in the loop clustering. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 237–245, 2017.

[26] A. Corrado. Animals-10 dataset. `https://www.kaggle.com/datasets/alessiocorrado99/animals10/`, 2019. Accessed March 14, 2020.

[27] P. Cunningham and S. J. Delany. k-nearest neighbour classifiers-a tutorial. *ACM Computing Surveys (CSUR)*, 54(6):1–25, 2021.

[28] S. Damrich and F. A. Hamprecht. On umap's true loss function. *Advances in Neural Information Processing Systems*, 34:5798–5809, 2021.

[29] S. Das, B. Saket, B. C. Kwon, and A. Endert. Geono-cluster: Interactive visual cluster analysis for biologists. *IEEE Transactions on Visualization and Computer Graphics*, 27(12):4401–4412, 2020.

[30] C. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *Proceedings of the 24th international conference on Machine learning*, pp. 521–528, 2007.

[31] A. Endert, C. Han, D. Maiti, L. House, and C. North. Observation-level interaction with statistical models for visual analytics. In *IEEE conference on visual analytics science and technology*, pp. 121–130, 2011.

[32] M. Espadoto, R. M. Martins, A. Kerren, N. S. Hirata, and A. C. Telea. Toward a quantitative survey of dimension reduction techniques. *IEEE transactions on visualization and computer graphics*, 27(3):2153–2173, 2019.

[33] M. Fanty and R. Cole. Spoken letter recognition. *Advances in neural information processing systems*, 3:220–226, 1991.

[34] C. Fu, Y. Zhang, D. Cai, and X. Ren. Atsne: Efficient and robust visualization on gpu through hierarchical optimization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 176–186, 2019.

[35] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[36] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1735–1742, 2006.

[37] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

[38] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994. doi: 10.1109/34.291440

[39] H. Jeon, M. Aupetit, S. Lee, H.-K. Ko, Y. Kim, and J. Seo. Distortion-aware brushing for interactive cluster analysis in multidimensional projections. *arXiv preprint arXiv:2201.06379*, 2022.

[40] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. ipca: An interactive system for pca-based visual analytics. *Computer Graphics Forum*, 28(3):767–774, 2009.

[41] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2563–2571, 2011.

[42] R. Kanjanabose, A. Abdul-Rahman, and M. Chen. A multi-task comparative study on scatter plots and parallel coordinates plots. *Computer Graphics Forum*, 34(3):261–270, 2015.

[43] S. Kaski, J. Nikkilä, M. Oja, J. Venna, P. Törönen, and E. Castrén. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC bioinformatics*, 4(1):1–13, 2003.

[44] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, vol. 2, 2011.

[45] H. Kim, J. Choo, H. Park, and A. Endert. Interaxis: Steering scatterplot axes via observation-level interaction. *IEEE transactions on visualization and computer graphics*, 22(1):131–140, 2015.

[46] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

[47] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[48] B. C. Kwon, B. Eysenbach, J. Verma, K. Ng, C. De Filippi, W. F. Stewart, and A. Perer. Clustervision: Visual supervision of unsupervised clustering.

IEEE transactions on visualization and computer graphics, 24(1):142–151, 2017.

[49] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. Axisketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE transactions on visualization and computer graphics*, 23(1):221–230, 2016.

[50] H. P. Lai, M. Visani, A. Boucher, and J.-M. Ogier. A new interactive semi-supervised clustering model for large image database indexing. *Pattern Recognition Letters*, 37:94–106, 2014.

[51] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[52] S. Lespinats and M. Aupetit. Checkviz: Sanity check and topological clues for linear and non-linear mappings. *Computer Graphics Forum*, 30(1):113–125, 2011.

[53] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg. Comparative analysis of multidimensional, quantitative data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1027–1035, 2010.

[54] G. M. Mamani, F. M. Fatore, L. G. Nonato, and F. V. Paulovich. User-driven feature space transformation. *Computer Graphics Forum*, 32(3pt3):291–299, 2013.

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

[56] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[57] L. G. Nonato and M. Aupetit. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2650–2673, 2018.

[58] S. Nourashrafeddin, E. Milios, and D. Arnold. Interactive text document clustering using feature labeling. In *Proceedings of the 2013 ACM symposium on Document Engineering*, pp. 61–70, 2013.

[59] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):564–575, 2008.

[60] D. Pérez, L. Zhang, M. Schaefer, T. Schreck, D. Keim, and I. Díaz. Interactive feature space extension for multidimensional data projection. *Neurocomputing*, 150:611–626, 2015.

[61] L. E. Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[62] N. Pezzotti, B. P. Lelieveldt, L. Van Der Maaten, T. Höllt, E. Eisemann, and A. Vilanova. Approximated and user steerable tsne for progressive visual analytics. *IEEE transactions on visualization and computer graphics*, 23(7):1739–1752, 2016.

[63] A. Pilhöfer, A. Gribov, and A. Unwin. Comparing clusterings using bertin's idea. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2506–2515, 2012.

[64] B. Rieck and H. Leitte. Exploring and comparing clusterings of multivariate data sets using persistent homology. *Computer Graphics Forum*, 35(3):81–90, 2016.

[65] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[66] D. Sacha, M. Kraus, J. Bernard, M. Behrisch, T. Schreck, Y. Asano, and D. A. Keim. Somflow: Guided exploratory cluster analysis with self-organizing maps and analytic provenance. *IEEE transactions on visualization and computer graphics*, 24(1):120–130, 2017.

[67] T. Sainburg, L. McInnes, and T. Q. Gentner. Parametric umap embeddings for representation and semisupervised learning. *Neural Computation*, 33(11):2881–2907, 2021.

[68] M. Sedlmair and M. Aupetit. Data-driven evaluation of visual quality measures. *Computer Graphics Forum*, 34(3):201–210, 2015.

[69] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results [gene identification]. *Computer*, 35(7):80–86, 2002.

[70] V. Silva and J. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Proceeding of Advances in neural information processing systems*, vol. 15, pp. 705–712, 2003.

[71] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum*, 28(3):831–838, 2009.

[72] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29:1857–1865, 2016.

[73] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[74] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020.

[75] L. Van Der Maaten. Learning a parametric embedding by preserving local structure. In *Artificial intelligence and statistics*, pp. 384–391, 2009.

[76] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11):2579–2605, 2008.

[77] G. Vijay. Weathers. `https://www.kaggle.com/datasets/vijaygiitk/multiclass-weather-dataset`, 2020. Accessed 2020.

[78] F. Wang and H. Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2495–2504, 2021.

[79] X. Wang and I. Davidson. Active spectral clustering. In *2010 IEEE International Conference on Data Mining*, pp. 561–568, 2010.

[80] Y. Wang, K. Feng, X. Chu, J. Zhang, C.-W. Fu, M. Sedlmair, X. Yu, and B. Chen. A perception-driven approach to supervised dimensionality reduction for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 24(5):1828–1840, 2018.

[81] Y. Wang, J. Li, F. Nie, H. Theisel, M. Gong, and D. J. Lehmann. Linear discriminative star coordinates for exploring class and cluster separation of high dimensional data. *Computer Graphics Forum*, 36(3):401–410, 2017.

[82] J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, and C. North. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE transactions on visualization and computer graphics*, 24(1):131–141, 2017.

[83] J. Xia, T. Chen, L. Zhang, W. Chen, Y. Chen, X. Zhang, C. Xie, and T. Schreck. SMAP: A joint dimensionality reduction scheme for secure multi-party visualization. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 107–118, 2020.

[84] J. Xia, Y. Zhang, J. Song, Y. Chen, Y. Wang, and S. Liu. Revisiting dimensionality reduction techniques for visual cluster analysis: An empirical study. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):529–539, 2021.

[85] C. Xiong, D. M. Johnson, and J. J. Corso. Active clustering with model-based uncertainty reduction. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):5–17, 2016.

[86] W. Yang, X. Wang, J. Lu, W. Dou, and S. Liu. Interactive steering of hierarchical clustering. *IEEE Transactions on Visualization and Computer Graphics*, 27(10):3953–3967, 2020.

[87] J. Yuan, C. Chen, W. Yang, M. Liu, J. Xia, and S. Liu. A survey of visual analytics techniques for machine learning. *Computational Visual Media*, 7(1):3–36, 2021.

[88] H. Zha, X. He, C. Ding, M. Gu, and H. Simon. Spectral relaxation for k-means clustering. *Advances in neural information processing systems*, 14:1057–1064, 2001.

[89] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng. Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(3):1–19, 2017.

[90] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang. Efficient knn classification with different numbers of nearest neighbors. *IEEE transactions on neural networks and learning systems*, 29(5):1774–1785, 2017.

[91] Z. Zhou, L. Zhong, and L. Wang. Locally incremental visual cluster analysis using markov random field. *Neurocomputing*, 136:49–55, 2014.

[92] Z. Zhou, X. Zu, Y. Wang, B. P. Lelieveldt, and Q. Tao. Deep recursive embedding for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 28(2):1237–1248, 2021.