Personalized Clustering via Targeted Representation Learning

Xiwen Geng[1,2], Suyun Zhao[1,2*], Yixin Yu[3], Borui Peng[3], Pan Du[1,2]
Hong Chen[1,2], Cuiping Li[1,2], Mengdie Wang[1,2]

[1]Key Lab of Data Engineering and Knowledge Engineering of MOE Renmin University of China
[2]School of Information, Renmin University of China
[3]School of Statistics, Remin University of China
{xiwen, zhaosuyun, yuyixin3642, pengborui, chong, licuiping}@ruc.edu.cn, {du_pan, wangmd_ruc}@163.com

Abstract

Clustering traditionally aims to reveal a natural grouping structure within unlabeled data. However, this structure may not always align with users' preferences. In this paper, we propose a personalized clustering method that explicitly performs targeted representation learning by interacting with users via modicum task information (e.g., *must-link* or *cannot-link* pairs) to guide the clustering direction. We query users with the most informative pairs, i.e., those pairs most hard to cluster and those most easy to miscluster, to facilitate the representation learning in terms of the clustering preference. Moreover, by exploiting attention mechanism, the targeted representation is learned and augmented. By leveraging the targeted representation and constrained contrastive loss as well, personalized clustering is obtained. Theoretically, we verify that the risk of personalized clustering is tightly bounded, guaranteeing that active queries to users do mitigate the clustering risk. Experimentally, extensive results show that our method performs well across different clustering tasks and datasets, even when only a limited number of queries are available.

1 Introduction

Deep clustering refers to some unsupervised machine learning techniques that leverage deep learning to reveal the grouping structures hidden in data samples. Unlike traditional clustering methods, deep clustering aims to learn latent representations of the data that facilitate clustering in terms of its underlying patterns (Xie, Girshick, and Farhadi 2016; Caron et al. 2018; Li et al. 2021; Zhong et al. 2021; Liu et al. 2022; Li et al. 2022, 2023).

Usually, most of the existing methods conduct clustering with the unique goal of maximizing clustering performance, while ignoring the personalized demand of clustering tasks. In real scenarios, however, users may tend to cluster unlabeled data according to their preferences, such as distinctive objectives (animals, architectures, and characters etc.), and then some personalized clustering demands are put forward. For example, Fig.1 depicts the diverse criteria of clustering. There are multiple objects in each image, such as animals and architectures. Some users require clustering in terms of animals, so the two images with dogs should be grouped in the same cluster, classified as dogs. In contrast, some others may prefer to group images according to architectures; thus, the two images with the Eiffel Tower should be clustered together. In cases with such personalized demands, many existing clustering techniques may decline or even be unworkable without user guidance.Therefore, it is still a challenging problem to cluster along a desired orientation.

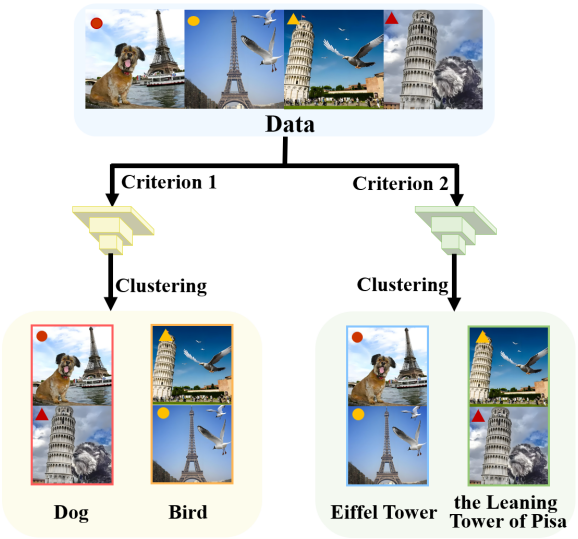

Figure 1: The diversity of cluster orientation. Different tasks have different orientations for feature learning and image clustering.

A candidate solution to this issue is constrained clustering, which incorporates prior knowledge through constraints to guide the neural network toward a desired cluster orientation (Wagstaff et al. 2001). The most commonly used constraint is the pairwise constraint, which provides information by indicating whether a pair of samples belongs to the same (*must-link*) or distinct (*cannot-link*) clusters. While they use constraints to facilitate models to achieve superior clustering performance, rather than towards a desired configuration. Sometimes, randomly selected constraint pairs (Manduchi et al. 2021; Sun et al. 2022) may lead to inferior clustering results in personalized scenarios, just as demonstrated in

*Corresponding authors.

the experiments in Section 4. Accordingly, actively querying informative sample pairs may facilitate personalized clustering. Unfortunately, several proposed active query strategies (Sun et al. 2022; Pei, Liu, and Fern 2015; Biswas and Jacobs 2014) tend to pick those samples beneficial for the default cluster orientation (Sun et al. 2022). Typically, their learned representations do not align with the desired cluster orientation, resulting in poor performance on personalized clustering (Angell et al. 2022).

To narrow this gap, we propose the personalized clustering model via targeted representation learning (PCL), which selects the most valuable sample pairs to learn targeted representations, thereby achieving clustering in the desired orientation. Specifically, we propose an active query strategy which picks those pairs most hard to cluster and those most easy to mis-cluster, to facilitate the representation learning in terms of the clustering task. Simultaneously, a constraint loss is designed to control targeted representation learning in terms of the cluster orientation. By exploiting attention mechanism, the targeted representation is augmented. Extensive experiments demonstrate that our method is effective in performing personalized clustering tasks. Furthermore, by strict mathematical reasoning, we verify the effectiveness of the proposed PCL method.

To summarize, our main contributions are listed as follows:

- We propose a novel model, called PCL, for a personalized clustering task, which leverages active query to control the targeted representation learning.
- A theoretical analysis on clustering is conducted to verify the effectiveness of active query in constraint clustering. Simultaneously, the tight upper bound of generalization risk of PCL is given.
- Extensive experiments show that our model outperforms five unsupervised clustering approaches, four semi-supervised clustering approaches on three image datasets. Our active query strategy also performs well when compared to other methods.

The remainder of this paper is organized as follows. Related works are reviewed in Section 2. We propose our method in Section 3. Furtherly, Section 4 evaluated our proposed approach experimentally. We conclude our work in Section 5.

2 Related Work

2.1 Deep Clustering

Deep neural networks have been explored to enhance clustering performance due to their powerful ability of representation learning (Vincent et al. 2010; Kingma and Welling 2014). A notable method is DEC (Xie, Girshick, and Farhadi 2016), which optimizes cluster centers and features simultaneously by minimizing the KL-divergence in the latent subspace. Similarly, IDFD (Tao, Takagi, and Nakata 2021) improves data similarity learning through sample discrimination and then reduces redundant correlations among features via feature decorrelation.

Recently, more and more methods achieve clustering representations by leveraging contrastive learning techniques, thereby facilitating deep clustering. For instances, DCCM (Wu et al. 2019) mines various kinds of sample relations by contrastive learning techniques and then train its network using the highly-confident information. PICA (Huang, Gong, and Zhu 2020) minimizes the cosine similarity between cluster-wise assignment vectors to achieve the most semantically plausible clustering solution. Moreover, CC (Li et al. 2021) proposes a dual contrastive learning framework aimed at achieving deep clustering representations. GCC (Zhong et al. 2021) selects positive and negative pairs using a KNN graph constructed on sample representations. Meanwhile, SPICE (Niu, Shan, and Wang 2022) divides the clustering network into a feature model and a clustering head, splitting the training process into three stages.

Though these deep clustering methods perform well in default orientation, they work poorly in personalized tasks.

2.2 Constrained Clustering

Constrained clustering refers to a type of clustering that incorporates additional information into the clustering process, e.g., pairwise (*must-link* and *cannot-link*) constraints (Chien, Zhou, and Li 2019).

Cop-Kmeans (Wagstaff et al. 2001) was the first to introduce pairwise constraints to traditional K-means clustering algorithms to enhance clustering performance, and many subsequent works are then proposed (Basu, Banerjee, and Mooney 2002; Bilenko, Basu, and Mooney 2004; Zheng and Li 2011; Yang et al. 2012, 2013). In recent years, constrained clustering approaches have been combined with deep neural networks. For instance, by introducing pairwise constraints, SDEC (Ren et al. 2019) extends the work of DEC (Xie, Girshick, and Farhadi 2016), and DCC (Zhang, Basu, and Davidson 2019) extends the work of IDEC (Guo et al. 2017). Bai et al. explored ways to improve clustering quality in scenarios where constraints are sourced from different origins (Bai, Liang, and Cao 2020). DC-GMM (Manduchi et al. 2021) explicitly integrates domain knowledge in a probabilistic form to guide the clustering algorithm toward a data distribution that aligns with prior clustering preferences. CDAC+ (Lin, Xu, and Zhang 2019) constructs pair constraints to transfer relations among data and discover new intents (clusters) in dialogue systems. Pairwise constraints are widely used to direct clustering models.

Some constrained clustering methods also integrate active learning (Ren et al. 2022), which aims to query the most informative samples to reduce labeling costs while maintaining performance (Ashtiani, Kushagra, and Ben-David 2016). COBRA (van Craenendonck, Dumancic, and Blockeel 2018) merges small clusters resulting from K-means into larger ones based on pairwise constraints by maximally exploiting constraint transitivity and entailment. ADC (Sun et al. 2022) constructs contrastive loss by comparing pairs of constraints and integrates deep representation learning, clustering, and data selection into a unified framework.

Although these methods typically cluster according to prior preferences, they do not consider the diverse criteria of

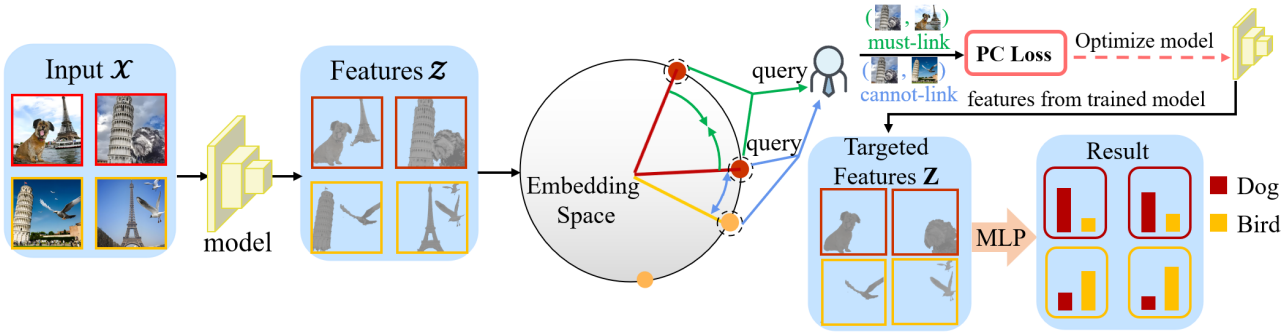

Figure 2: The framework of PCL. A deep neural network creates representations from two random augmentations of the data. By assessing the position of images in the feature space, PCL selects the most informative sample pairs to guide the training of the model. The model is then retrained by querying whether these pairs are *must-link* or *cannot-link*. This approach allows the final model to concentrate on features relevant to the desired cluster orientation, resulting in accurate clustering outcomes.

clustering but focus on improving performance or reducing the query budget. Dasgupta and Ng proposed an active spectral clustering algorithm that specifies the dimension along the sentiment embedded in the data points, which is similar to our problem of diverse image clustering (Dasgupta and Ng 2010). However, their method relies on label constraints and is designed for natural language processing.

3 Methods

3.1 Personlized Clustering

We propose a novel Personalized Clustering model (PCL) that leverages active learning to guide the model's clustering in a specified orientation. Using a designed pairwise scoring function within a query strategy, certain image pairs are selected for annotation. Oracles then judge whether these image pairs belong to the same class, responding with *YES* or *NO*. These responses form *must-link* or *cannot-link* constraints, which are incorporated into the construction of a constrained contrastive loss, adjusting the network parameters through backpropagation. By integrating the cross-instance attention module in this way, the model can capture features that align more closely with the desired cluster orientation (Gan et al. 2023). The specific framework of PCL is illustrated in Figure 2.

Given a mini-batch of images $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ where N is the batch size, we apply two stochastic data transformations, T^a and T^b, from the same family of augmentations $\mathcal{T}$, resulting in augmented data $\{\mathbf{x}_1, ..., \mathbf{x}_N, \mathbf{x}_{N+1}, ..., \mathbf{x}_{2N}\}$. This process constructs the initial positive pairs $(\mathbf{x}_i, \mathbf{x}_{i+N})_{i=1}^N$. The network then extracts preliminary features from the augmented data and refines them using the cross-instance attention module. It projects these representations into the feature space, yielding target features $\{\mathbf{z}_1, ..., \mathbf{z}_{2N}\}$, and into the cluster space, yielding assignment probabilities $\{\mathbf{p}_1, ..., \mathbf{p}_{2N}\}$. These are expressed in matrices $\mathbf{Z} \in \mathbb{R}^{2N\times M}$ and $\mathbf{P} \in \mathbb{R}^{2N\times K}$, where M is the feature dimension and K is the number of clusters.

Next, we select the most informative pair based on the scoring function, submit a query to the oracle, and document the feedback in the indicator matrix $\mathbb{1}^+$ and $\mathbb{1}^-$. These matrices represent the relationship of pairs, where $\mathbb{1}^+_{ij} = 1$ if $y_i = y_j$(*must-link*) and $\mathbb{1}^-_{ij} = 1$ if $y_i \neq y_j$(*cannot-link*). Meanwile, due to the initial positive pairs, we use the indicator matrix $\mathbb{1}$ to represent the positive pair $(\mathbf{x}_i, \mathbf{x}_j)$ is constructed by data augment, where $\mathbb{1}_{ij} = 1$ if $(\mathbf{x}_i, \mathbf{x}_j)$ is constructed by data augment. These constraints guide the clustering process to the desired orientation via the constrained contrastive loss.

In the following two subsections, we first introduce our constrained contrastive loss in Subsection 3.1, followed by the pairwise query strategy in Subsection 3.2.

3.2 Constrained Contrastive Loss

To achieve effective and targeted clustering, we employ a constrained contrastive loss similar to infoNCE (Oord, Li, and Vinyals 2018) to evaluate the loss of each sample. Different weights are set for positive pairs based on their construction methods. We focus on clustering impacts in both feature and clustering spaces, while also aiming to differentiate category representations in the clustering space. In the feature space, the loss function for a positive pair of examples(i, j) is defined as:

$$l(\mathbf{z}_i, \mathbf{z}_j, \mathbf{Z}) = -\log \frac{\exp(s(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\mu_i \sum_{k=1}^{2N} (1 + \mathbb{1}^-_{ik}) \exp(s(\mathbf{z}_i, \mathbf{z}_k)/\tau)}, \tag{1}$$

where $s(\mathbf{a}, \mathbf{b}) = (\mathbf{a}\mathbf{b}^\top)/(\|\mathbf{a}\|\|\mathbf{b}\|)$ measures pair-wise similarity by cosine distance, $\mu_i = N/(N + \sum_{j=1}^{2N}(\mathbb{1}^-_{ij}))$ normalizes the range of the denominator, and τ is the temperature parameter.

Positive pairs from data augmentation and queries are weighted differently in the loss term. A symmetric matrix $\mathbf{W} \in \mathcal{R}^{2N\times 2N}$ is constructed with elements:

$$w_{ij} := \begin{cases} N_+, & \text{if } \mathbb{1}_{ij} = 1 \\ \lambda N, & \text{if } \mathbb{1}^+_{ij} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where λ is the hyperparameter and N_+ is the total number of positive pairs constructed by querying.

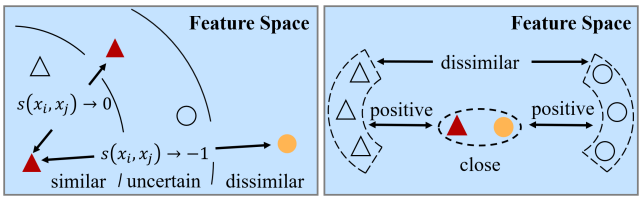

Figure 3: (Left) The illustration shows a sample pair with uncertain similarity, highlighted as red nodes, which are prioritized for querying. (Right) Some samples, due to incorrect feature extraction, are mistakenly grouped together despite belonging to different clusters, as shown by the red and yellow nodes. Identifying these can be achieved by assessing the similarity between their positive samples.

The loss of sample i is defined as:

$$\ell_i = \sum_{j=1}^{2N} \frac{w_{ij}(l(\mathbf{z}_i, \mathbf{z}_j, \mathbf{Z}) + l(\mathbf{p}_i, \mathbf{p}_j, \mathbf{P}))}{\sum_{k=1}^{2N} w_{ik}}. \tag{3}$$

In the clustering space, it is crucial to ensure that each cluster remains distinct. Letting $\mathbf{C} = \mathbf{P}^\top \in \mathcal{R}^{K \times 2N}$, we obtain a set $\{\mathbf{c}_1, ..., \mathbf{c}_K, \mathbf{c}_{K+1}, ..., \mathbf{c}_{2K}\}$, where $\mathbf{c}_i (i \leq K)$ and $\mathbf{c}_{i+K}$ represent the i-th cluster under augmentations $T^a(\mathcal{X})$ and $T^b(\mathcal{X})$. A loss function is defined to differetiate i-th cluster from others:

$$\hat{\ell}_i = l(\mathbf{c}_i, \mathbf{c}_{i+N}, \mathbf{C}) + l(\mathbf{c}_{i+N}, \mathbf{c}_i, \mathbf{C}) \tag{4}$$

Our overall loss is defined as:

$$\mathcal{L}_{PC} = \frac{1}{2N} \sum_{i=1}^{2N} \ell_i + \frac{1}{2K} \sum_{i=1}^{K} \hat{\ell}_i + H(\hat{Y}) \tag{5}$$

where $H(\hat{Y}) = -\sum_{i=1}^{K}[P(\hat{Y}_i) log(P(\hat{Y}_i))]$ serves as a regularization term.

3.3 Query Strategy

Our model learns features conducive to clustering by utilizing a constrained contrastive loss, emphasizing the importance of selecting informative sample pairs to guide cluster orientation. We propose an innovative query strategy that considers both the uncertainty of sample pairs and hard negatives, as illustrated in Fig.3. This strategy identifies critical sample pairs to refine cluster orientation and enhance performance.

Since features extracted by untrained models are initially unreliable, random constraints are generated on the data before training, and the model is pre-trained before applying the query strategy. We believe that the initial aim is to establish a general cluster orientation and it becomes essential to rectify samples misclassified by the model and make minor adjustments to the cluster orientation later. Therefore, at the outset of the task, the uncertainty of sample pairs takes precedence. As the task progresses, focusing on hard negatives becomes crucial.

Uncertainty of pairs. To establish cluster orientation, we prioritize the selection of the most uncertain sample pairs. Cosine similarity of sample pairs ranges from -1 to 1. Pairs with similarity near 1 likely belong to the same cluster, whereas pairs near -1 are likely in different clusters. Pairs with similarity close to 0 are uncertain for the model, offering both similar and distinct features that provide valuable insights into cluster orientation. The uncertainty score for an sample pair $(\mathbf{x}_i, \mathbf{x}_j)$ is defined as:

$$\mathcal{S}_{up}(\mathbf{x}_i, \mathbf{x}_j) = \sigma[-|s(\mathbf{z}_i, \mathbf{z}_j) - \epsilon|], \tag{6}$$

where $\mathbf{z}_i$ and $\mathbf{z}_j$ are the features of samples $\mathbf{x}_i$ and $\mathbf{x}_j$ in feature space. Similarity is averaged over two data augmentations. $\sigma[\cdot]$ denotes the Min-Max normalization operator (Jain, Nandakumar, and Ross 2005), and ϵ is a small positive number as *must-link* constraints are more informative than *cannot-links* (Sun et al. 2022).

Hard negatives. Hard negatives are samples with a significant discrepancy between the actual clustering and the current results of the model. Although actual clustering results cannot be directly observed, they can be inferred using known positive and negative examples (Zhuang and Moulin 2023). For two samples $\mathbf{x}_i$ and $\mathbf{x}_j$, $P(\mathbf{x}_i, \mathbf{x}_j) = \sigma[s(\mathbf{x}_i, \mathbf{x}_j)]$ represents the probability that these samples are in the same cluster. $Q(\mathbf{x}_i, \mathbf{x}_j) = \sigma[\sum_{k=1}^{N}(\mathbb{1}^{+}_{jk} s(\mathbf{x}_i, \mathbf{x}_k) - \mathbb{1}^{-}_{jk} s(\mathbf{x}_i, \mathbf{x}_k))]$ represents the probability calculated from the known pairs. Relative entropy is used to assess the difference between $P(\mathbf{x}_i, \mathbf{x}_j)$ and $Q(\mathbf{x}_i, \mathbf{x}_j)$, identifying challenging samples to classify:

$$\mathcal{S}_{hp}(\mathbf{x}_i, \mathbf{x}_j) = \sigma[P(\mathbf{x}_i, \mathbf{x}_j) \log \frac{P(\mathbf{x}_i, \mathbf{x}_j)}{Q(\mathbf{x}_i, \mathbf{x}_j)}]. \tag{7}$$

Query strategy. The strategy for querying the most informative sample pairs combines the scores defined above. The joint query score is given by:

$$\mathcal{S}(\mathbf{x}_i, \mathbf{x}_j) = r[\mathcal{S}_{up}(\mathbf{x}_i, \mathbf{x}_j)] + (1 - r)\mathcal{S}_{hp}(\mathbf{x}_i, \mathbf{x}_j) \tag{8}$$

where r is the ratio of the remaining query budget to the total budget, indicating that the weight of hard-to-learn pairs increases with more queries. As queries increase, the cluster orientation is refined and the strategy focuses on maximizing performance.

The full querying and clustering algorithm[1] of the model is detailed in the Appendix.[2]

3.4 Theoretical Analysis

We focus on the feature extractor function $f \in \mathcal{F} : \mathcal{X} \to Z$ which determines the final clustering results (Saunshi et al. 2019). The unsupervised loss of sample i is defined as:

$$\ell_i'(f) = \sum_{j=1}^{2N} \frac{w_{ij}(l(f(\mathbf{x}_i), f(\mathbf{x}_j), \mathbf{Z})}{\sum_{k=1}^{2N} w_{ik}}. \tag{9}$$

[1]Code is available at https://github.com/hhdxwen/PCL.

[2]Appendix is available at https://arxiv.org/abs/2412.13690.

Definition 3.1 (Unsupervised Loss) *If we can query q times, the overall unsupervised loss is defined as:*

$$\hat{\mathcal{L}}_q(f) \coloneqq \frac{1}{2N}\sum_{i=1}^{2N}\ell_i'(f), \quad q \coloneqq \sum_{1\le i<j\le 2N}(\mathbb{1}_{ij}^{+}+\mathbb{1}_{ij}^{-}). \tag{10}$$

and $\hat{f}_q$ is defined as $\arg\min_{f\in\mathcal{F}}\hat{\mathcal{L}}_q(f)$.

The unsupervised population loss after q queries is defined as:

$$\mathcal{L}_q(f) \coloneqq \mathbb{E}_{x\sim\mathcal{X}}[\hat{\mathcal{L}}_q(f)], \tag{11}$$

and f_q^* is defined as $\arg\min_{f\in\mathcal{F}}\mathcal{L}_q(f)$.

Firstly we propose Theorem 3.1. All proofs are in Appendix.

Theorem 3.1 *Suppose we can completely query each sample. We use Q as the total number of queries. With probability at least $1-\delta$,*

$$|\hat{\mathcal{L}}_Q(\hat{f}_Q)-\mathcal{L}_Q(f_Q^*)| \le \mathcal{O}(\frac{\eta\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{N}+\sqrt{\frac{\log\frac{1}{\delta}}{N}}). \tag{12}$$

where $\mathcal{R}_{\mathcal{S}}(\mathcal{F})$ is the Rademacher complexity of $\mathcal{F}$ with respect to training set $\mathcal{S}$, and η is the Lipschitz constant.

Theorem 3.1 indicates that $\mathcal{L}_Q(f_Q^*)$ can be well bounded if the trained model $\hat{f}_Q$ is good. However, due to the limited queries (less than Q), we cannot calculate $\hat{\mathcal{L}}_Q(\hat{f}_Q)$. As a compromise, can we get closer to it?

Theorem 3.2 (Queries Reduce the Gap). *We only consider adding one query. If $q \ll N$:*

$$|\hat{\mathcal{L}}_q(\hat{f}_q)-\hat{\mathcal{L}}_Q(\hat{f}_Q)| \ge |\hat{\mathcal{L}}_{q+1}(\hat{f}_{q+1})-\hat{\mathcal{L}}_Q(\hat{f}_Q)|. \tag{13}$$

This just matches the process of algorithm: we initially make a new query request based on the existing model $\hat{f}_q$, followed by updating $\hat{\mathcal{L}}_q$ to $\hat{\mathcal{L}}_{q+1}$ based on the query results, and finally optimizing the loss to get $\hat{f}_{q+1}$. Theorem 3.2 guarantees the gap between the loss of the current model and the complete queries loss will be reduced after each query. Now we show that our strategy is indeed effective.

Theorem 3.3 (Query Strategy Is Effective). *Suppose $q \ll N$, the sample pair $(\mathbf{x}_i,\mathbf{x}_j)$ is queried, and the result is cannot-link, we have:*

$$|\hat{\mathcal{L}}_q(f)-\hat{\mathcal{L}}_{q+1}(f)| \propto \exp\left(\mathcal{S}_{up}(\mathbf{x}_i,\mathbf{x}_j)\right), \tag{14}$$

and if the result is must-link, we have:

$$|\hat{\mathcal{L}}_q(f)-\hat{\mathcal{L}}_{q+1}(f)| \propto \mathcal{S}_{up}(\mathbf{x}_i,\mathbf{x}_j). \tag{15}$$

Since we want to reduce the gap in Theorem 3.2, we should equivalently maximize $|\hat{\mathcal{L}}_q(f)-\hat{\mathcal{L}}_{q+1}(f)|$ and then train the model according to $\hat{\mathcal{L}}_{q+1}$. Theorem 3.3 states that our query strategy is effective in this sense, because when q is small, $\mathcal{S}_{up}$ matters more in the scoring function. Besides, we have added $\mathcal{S}_{hp}$ in the score and weight it more if q is larger, making our strategy more robust and efficient. The following experiments have proved the effectiveness in practice.

Dataset	Instances	Classes	Clusters
CIFAR10-2	60,000	10	2
CIFAR100-4	60,000	100	4
ImageNet10-2	12,818	10	2

Table 1: A summary of the datasets.

4 Experiments

In this section, we present a comprehensive set of experiments to evaluate the effectiveness of the proposed method.

4.1 Experiments Settings

Datasets. We assessed our method using three widely-used image datasets: CIFAR-10, CIFAR-100 (Krizhevsky 2009), and ImageNet-10 (Deng et al. 2009). To demonstrate that constraints can guide cluster orientation, we differentiate between training and test sets, applying constraints only to the training set. For CIFAR-10, CIFAR-100, and ImageNet-10, we added 10k, 10k, and 4k constraints respectively, representing 0.0004%, 0.0004%, and 0.002% of the total training set constraints.

To test different cluster orientations, we reconstructed the original datasets into two artificial versions: default and personalized. We set target clusters to 2 or 4, facilitating division into distinct semantic orientations. We named these datasets CIFAR10-2, CIFAR100-4, and ImageNet10-2 to reflect the differences of labels from the originals. Tab. 1 and Appendix illustrate the details of the adopted datasets. The artificial datasets share the same number of target clusters. The default orientation aligns with deep clustering tendencies, while the personalized orientation is intentionally opposite. By comparing results on these datasets, we evaluate whether our PCL outperforms state-of-the-art deep clustering and semi-supervised methods.

Compared Methods. To verify the effectiveness of the proposed method, we select some representative and frontier methods for comparison. We compare with the known or SOTA clustering methods as follows:

- **K-means** (MacQueen 1967): A classical partition-based clustering algorithm.
- **MiCE** (Tsai, Li, and Zhu 2020): Combines contrastive learning with latent probability models.
- **CC** (Li et al. 2021): A leading method for contrastive clustering at instance and cluster levels.
- **GCC** (Zhong et al. 2021): Extends positive sample pairs using KNN graph neighbors.
- **SPICE** (Niu, Shan, and Wang 2022): A three-stage training method with excellent clustering performance.

Besides, we also compared with constrained clustering methods:

- **DCC** (Zhang, Basu, and Davidson 2019): Extends IDEC (Guo et al. 2017) with constraints.

Default		CIFAR10-2				CIFAR100-4				ImageNet10-2			
		NMI	ARI	F	ACC	NMI	ARI	F	ACC	NMI	ARI	F	ACC
Unsupervised	kmeans	0.054	0.079	0.550	0.641	0.038	0.033	0.281	0.358	0.078	0.103	0.553	0.661
	MiCE	0.664	0.755	0.881	0.934	0.230	0.228	0.423	0.590	0.762	0.849	0.925	0.960
	CC	0.575	0.662	0.835	0.907	0.176	0.166	0.379	0.515	0.508	0.615	0.807	0.892
	GCC	0.653	0.753	0.880	0.934	**0.344**	**0.337**	**0.503**	**0.604**	**0.908**	**0.952**	**0.976**	**0.988**
	SPICE	0.600	0.638	0.823	0.899	0.250	0.197	0.399	0.496	0.642	0.725	0.864	0.925
Constrained	DCC	0.572	0.686	0.848	0.914	0.134	0.120	0.345	0.449	0.432	0.536	0.768	0.866
	ADC	0.544	0.662	0.838	0.907	0.047	0.055	0.319	0.390	0.453	0.550	0.775	0.870
	SDEC	0.355	0.430	0.720	0.828	0.070	0.048	0.317	0.380	0.192	0.254	0.627	0.752
	DC-GMM	0.542	0.657	0.838	0.905	0.107	0.121	0.349	0.431	0.490	0.598	0.799	0.886
	Ours	**0.737**	**0.830**	**0.918**	**0.955**	0.197	0.208	0.407	0.575	0.722	0.817	0.908	0.952

Table 2: The clustering performance under the default orientation, on three object image benchmarks. The default orientation completely follows the tendency of deep clustering. The best results are shown in boldface.

Personlized		CIFAR10-2				CIFAR100-4				ImageNet10-2			
		NMI	ARI	F	ACC	NMI	ARI	F	ACC	NMI	ARI	F	ACC
Unsupervised	kmeans	0.009	0.015	0.519	0.563	0.028	0.022	0.274	0.314	0.027	0.037	0.519	0.597
	MiCE	0.028	0.045	0.535	0.607	0.159	0.135	0.355	0.399	0.024	0.032	0.516	0.591
	CC	0.022	0.035	0.528	0.594	0.132	0.115	0.342	0.405	0.032	0.043	0.521	0.605
	GCC	0.035	0.057	0.542	0.620	0.230	0.203	0.404	0.508	0.025	0.034	0.517	0.592
	SPICE	0.005	0.007	0.513	0.543	0.232	0.215	0.413	0.504	0.068	0.088	0.548	0.648
Constrained	DCC	0.244	0.334	0.679	0.789	0.099	0.095	0.340	0.473	0.204	0.267	0.634	0.758
	ADC	0.206	0.291	0.665	0.771	0.024	0.023	0.374	0.322	0.102	0.138	0.570	0.686
	SDEC	0.013	0.021	0.521	0.573	0.092	0.083	0.354	0.417	0.075	0.102	0.552	0.660
	DC-GMM	0.060	0.074	0.671	0.655	0.072	0.060	0.311	0.384	0.000	0.000	0.675	0.512
	Ours	**0.453**	**0.558**	**0.765**	**0.873**	**0.234**	**0.247**	**0.437**	**0.597**	**0.409**	**0.510**	**0.755**	**0.857**

Table 3: The clustering performance under the personalized orientation, on three object image benchmarks. The personalized orientation is artificially designed opposite to the default one. The best results are shown in boldface.

- **ADC** (Sun et al. 2022): Constructs contrastive losses using only constraint pairs.
- **SDEC** (Ren et al. 2019): Extends DEC (Xie, Girshick, and Farhadi 2016) with constraints.
- **DC-GMM** (Manduchi et al. 2021): A recent semi-supervised method using probabilistic domain knowledge.

Implementation Details. All methods used ResNet34 as the backbone network without modification. Parameters related to deep contrastive clustering followed previous methods (Li et al. 2021; Zhong et al. 2021). The batch size was 128, and Adam with an initial learning rate of 1e-5 was used for optimization. All images were resized to 128×128. The feature dimensionality M was set to 128. Hyperparameters were consistent across datasets with $\lambda = 4$ and $\epsilon = 0.2$. Constraints were extended by labeling similar pairs with high confidence after several iterations. For semi-supervised and active constrained methods, 10k pairwise constraints were set on the three datasets. Our method used a training epoch $E = 500$ to determine final performance.

Evaluation Metrics. We adopted four standard clustering metrics to evaluate our method including Normalized Mutual Information (NMI), Accuracy (ACC), F-measures (F) , and Adjusted Rand Index (ARI). These metrics reflect the performance of clustering from different aspects, and higher values indicate better performance.

4.2 Results

Clustering Performance. The clustering performance of PCL and comparison methods on four datasets with two cluster orientations are presented in Tables 2 and 3. More compared methods can be found at our Appendix.

Key observations include: i) In the default orientation, both unsupervised and constrained methods perform well, with some unsupervised methods even surpassing constrained ones. Our method performs comparably, with slight gaps in some metrics. ii) In the personalized orientation, unsupervised methods perform poorly, and existing constrained methods fail to achieve the desired orientation. However, PCL excels in the personalized orientation, outperforming other methods.

These results confirm our expectations. Unsupervised methods struggle across all orientations, while existing constrained methods cannot adequately control cluster orientation. In contrast, PCL performs well across orientations, ef-

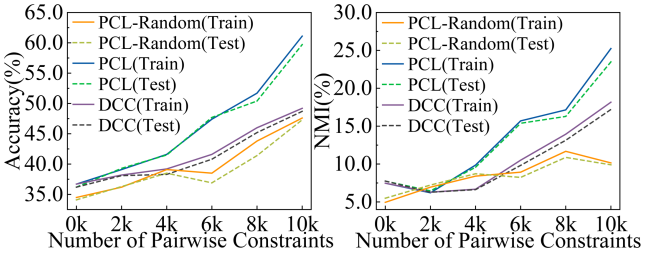

Figure 4: Clustering accuracy and NMI on train sets and test sets for different number of pairwise constraints.

fectively identifying useful samples to learn cluster orientations and addressing personalized clustering challenges. Notably, the DC-GMM method is highly dependent on feature extraction results. For fairness, we used the original trainable VGG16 without a pre-trained network when comparing DC-GMM.

Overall, PCL successfully clusters along the desired orientation, surpassing state-of-the-art methods.

Query Efficiency. To evaluate the effectiveness of our method and query strategy, we compared it against a random strategy and DCC, as depicted in Fig. 4. We conducted experiments with varying numbers of constraints on the personalized cluster orientation of the CIFAR100-4 dataset. To further demonstrate that our method effectively learns task-related features, we differentiated between the training and test sets, applying constraints exclusively to the training set.

Key observations include: i) As the number of constraints increases, clustering performance improves. However, with the same number of constraints, our method significantly outperforms the other two methods. ii) Our method demonstrates comparable performance on both the test and training sets under identical conditions.

These observations indicate that our method and query strategy excel in performing task-based clustering, focusing on features critical to the task. Additionally, the constraints used (10k) represent only 0.0004% of the total, highlighting our ability to achieve superior clustering results with minimal constraint information.

Overall, our approach effectively identifies task-related features with fewer queries, addressing personalized clustering challenges efficiently.

Module	NMI	ARI	F	ACC
CNN only	0.127	0.122	0.344	0.445
CNN+Attention	**0.234**	**0.247**	**0.437**	**0.597**

Table 4: Effect of Cross-Attention Module in personalized clustering on CIFAR100-4.

4.3 Visualization and Ablation Studies.

Visualization. To vividly display the personalized clustering results, we visualize the distribution of samples in CIFAR10-2 after clustering. In Fig. 4, we have the following observations. i) Subfigure (a) depicts that the sample points

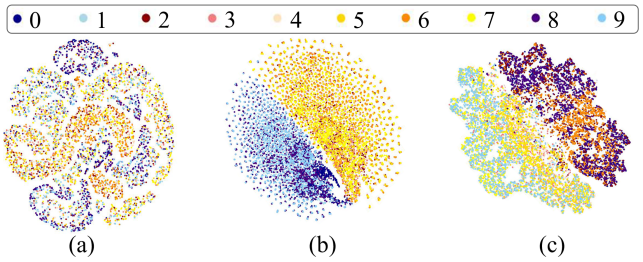

Figure 5: The evolution of features across the training process under two cluster orientations: (a) Initial distribution of samples, (b) Sample distribution after clustering along the default orientation and (c) Sample distribution after clustering along a given personalized orientation. The color of the dots denotes their original class labels of CIFAR-10.

Query strategy	NMI	ARI	F	ACC
random	0.172	0.170	0.380	0.487
S_{up}	0.196	0.211	0.411	0.569
S_{hp}	0.160	0.154	0.369	0.481
$S_{up} + S_{hs}$	**0.234**	**0.247**	**0.437**	**0.597**

Table 5: Effect of query strategy in personalized clustering on CIFAR100-4.

before clustering is chaos. ii) Subfigure (b) depicts that the yellow and orange sample points cluster together, while in subfigure (c) the dark blue and dark orange dots are clustered together. The above facts show that PCL can focus on the relationship between sample representations on the targeted orientation rather than any other orientations.

Effect of Cross-Attention Module. We conducted an ablation analysis by removing the cross-attention module. Results along the personalized cluster orientation of CIFAR100-4 are shown in Table 4. The highest clustering performance was achieved with the cross-attention module, highlighting its necessity for extracting task-specific features.

Effect of joint query strategy. We performed an ablation analysis by removing components of the query strategy. Results along the personalized cluster orientation of CIFAR100-4 are shown in Table 5. Personalized clustering achieved better results using our query strategy.

5 Conclusion

In this study, we attempt to tackle the problem of personalized clustering by interacting with users and then propose a clustering model named PCL. In our work, we designed a kind of active query strategy to identify sample pairs that are informative for targeted representation learning. To further enhance the learning process, we propose a constrained clustering loss and leverage attention mechanism, maximizing the power of constraints in guiding the desired cluster orientation. We have theoretically verified the effectiveness of our query strategy and loss function, and extensive experiments have validated our findings, simultaneously.

Acknowledgements

This work is supported by the National Key Research & Develop Plan (2023YFB4503600), National Natural Science Foundation of China (U23A20299, U24B20144, 62172424, 62276270, 62322214). It is also partially supported by the Opening Fund of Hebei Key Laboratory of Machine Learning and Computational Intelligence. We sincerely appreciate the valuable and insightful feedback provided by all reviewers.

References

Angell, R.; Monath, N.; Yadav, N.; and McCallum, A. 2022. Interactive Correlation Clustering with Existential Cluster Constraints. In *ICML*.

Ashtiani, H.; Kushagra, S.; and Ben-David, S. 2016. Clustering with Same-Cluster Queries. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Bai, L.; Liang, J.; and Cao, F. 2020. Semi-supervised clustering with constraints of different types from multiple information sources. *IEEE Transactions on pattern analysis and machine intelligence*, 43(9): 3247–3258.

Basu, S.; Banerjee, A.; and Mooney, R. J. 2002. Semi-supervised Clustering by Seeding. In *International Conference on Machine Learning*.

Bilenko, M.; Basu, S.; and Mooney, R. J. 2004. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, 11.

Biswas, A.; and Jacobs, D. W. 2014. Active Image Clustering with Pairwise Constraints from Humans. *International Journal of Computer Vision*, 108: 133–147.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep Clustering for Unsupervised Learning of Visual Features. In *ECCV*.

Chien, E.; Zhou, H.; and Li, P. 2019. HS2: Active learning over hypergraphs with pointwise and pairwise queries. In *AISTATS*.

Dasgupta, S.; and Ng, V. 2010. Which clustering do you want? inducing your ideal clustering with minimal feedback. *Journal of Artificial Intelligence Research*, 39: 581–632.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Gan, Z.; Zhao, S.; Kang, J.; Shang, L.; Chen, H.; and Li, C. 2023. Superclass Learning with Representation Enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24060–24069.

Guo, X.; Gao, L.; Liu, X.; and Yin, J. 2017. Improved Deep Embedded Clustering with Local Structure Preservation. In *International Joint Conference on Artificial Intelligence*.

Huang, J.; Gong, S.; and Zhu, X. 2020. Deep Semantic Clustering by Partition Confidence Maximisation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8846–8855.

Jain, A.; Nandakumar, K.; and Ross, A. 2005. Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12): 2270–2285.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. *CoRR*, abs/1312.6114.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, Toronto.

Li, Y.; Hu, P.; Liu, Z.; Peng, D.; Zhou, J. T.; and Peng, X. 2021. Contrastive Clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10): 8547–8555.

Li, Y.; Hu, P.; Peng, D.; Lv, J.; Fan, J.; and Peng, X. 2023. Image clustering with external guidance. *arXiv preprint arXiv:2310.11989*.

Li, Y.; Yang, M.; Peng, D.; Li, T.; Huang, J.; and Peng, X. 2022. Twin contrastive learning for online clustering. *International Journal of Computer Vision*, 130(9): 2205–2221.

Lin, T.-E.; Xu, H.; and Zhang, H. 2019. Discovering New Intents via Constrained Deep Adaptive Clustering with Cluster Refinement. In *AAAI Conference on Artificial Intelligence*.

Liu, Y.; Tu, W.; Zhou, S.; Liu, X.; Song, L.; Yang, X.; and Zhu, E. 2022. Deep Graph Clustering via Dual Correlation Reduction. In *AAAI*.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

Manduchi, L.; Chin-Cheong, K.; Michel, H.; Wellmann, S.; and Vogt, J. E. 2021. Deep Conditional Gaussian Mixture Model for Constrained Clustering. In *NeurIPS*.

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

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Pei, Y.; Liu, L.-P.; and Fern, X. Z. 2015. Bayesian Active Clustering with Pairwise Constraints. In *ECML/PKDD*.

Ren, P.; Xiao, Y.; Chang, X.; Huang, P.; Li, Z.; Chen, X.; and Wang, X. 2022. A Survey of Deep Active Learning. *ACM Computing Surveys (CSUR)*, 54: 1–40.

Ren, Y.; Hu, K.; Dai, X.; Pan, L.; Hoi, S. C. H.; and Xu, Z. 2019. Semi-supervised deep embedded clustering. *Neurocomputing*, 325: 121–130.

Saunshi, N.; Plevrakis, O.; Arora, S.; Khodak, M.; and Khandeparkar, H. 2019. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 5628–5637. PMLR.

Sun, B.; Zhou, P.; Du, L.; and Li, X. 2022. Active deep image clustering. *Knowl. Based Syst.*, 252: 109346.

Tao, Y.; Takagi, K.; and Nakata, K. 2021. Clustering-friendly Representation Learning via Instance Discrimination and Feature Decorrelation. In *9th International Conference on Learning Representations (ICLR)*.

Tsai, T. W.; Li, C.; and Zhu, J. 2020. Mice: Mixture of contrastive experts for unsupervised image clustering. In *International conference on learning representations*.

van Craenendonck, T.; Dumancic, S.; and Blockeel, H. 2018. COBRA: A Fast and Simple Method for Active Clustering with Pairwise Constraints. In *International Joint Conference on Artificial Intelligence*.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P.-A. 2010. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *J. Mach. Learn. Res.*, 11: 3371–3408.

Wagstaff, K. L.; Cardie, C.; Rogers, S.; and Schrödl, S. 2001. Constrained K-means Clustering with Background Knowledge. In *International Conference on Machine Learning*.

Wu, J.; Long, K.; Wang, F.; Qian, C.; Li, C.; Lin, Z.; and Zha, H. 2019. Deep Comprehensive Correlation Mining for Image Clustering. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 8149–8158.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised Deep Embedding for Clustering Analysis. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, 478–487. PMLR.

Yang, Y.; Rutayisire, T.; Lin, C.; Li, T.; and Teng, F. 2013. An Improved Cop-Kmeans Clustering for Solving Constraint Violation Based on MapReduce Framework. *Fundam. Informaticae*, 126: 301–318.

Yang, Y.; Tan, W.; Li, T.; and Ruan, D. 2012. Consensus clustering based on constrained self-organizing map and improved Cop-Kmeans ensemble in intelligent decision support systems. *Knowl. Based Syst.*, 32: 101–115.

Zhang, H.; Basu, S.; and Davidson, I. 2019. A Framework for Deep Constrained Clustering - Algorithms and Advances. In *ECML/PKDD*.

Zheng, L.; and Li, T. 2011. Semi-supervised Hierarchical Clustering. *2011 IEEE 11th International Conference on Data Mining*, 982–991.

Zhong, H.; Wu, J.; Chen, C.; Huang, J.; Deng, M.; Nie, L.; Lin, Z.; and Hua, X. 2021. Graph Contrastive Clustering. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9204–9213.

Zhuang, F.; and Moulin, P. 2023. Deep semi-supervised metric learning with mixed label propagation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3429–3438.