

ClusMatch: Improving Deep Clustering by Unified Positive and Negative Pseudo-label Learning

Jianlong Wu, *Member, IEEE*, Zihan Li, Wei Sun, Jianhua Yin, *Member, IEEE*, Liqiang Nie, *Senior Member, IEEE*, Zhouchen Lin, *Fellow, IEEE*

Abstract—Recently, deep clustering methods have achieved remarkable results compared to traditional clustering approaches. However, its performance remains constrained by the absence of annotations. A thought-provoking observation is that there is still a significant gap between deep clustering and semi-supervised classification methods. Even with only a few labeled samples, the accuracy of semi-supervised learning is much higher than that of clustering. Given that we can annotate a small number of samples in a certain unsupervised way, the clustering task can be naturally transformed into a semi-supervised setting, thereby achieving comparable performance. Based on this intuition, we propose ClusMatch, a unified positive and negative pseudo-label learning based semi-supervised learning framework, which is pluggable and can be applied to existing deep clustering methods. Specifically, we first leverage the pre-trained deep clustering network to compute predictions for all samples, and then design specialized selection strategies to pick out a few high-quality samples as labeled samples for supervised learning. For the unselected samples, the novel unified positive and negative pseudo-label learning is introduced to provide additional supervised signals for semi-supervised fine-tuning. We also propose an adaptive positive-negative threshold learning strategy to further enhance the confidence of generated pseudo-labels. Extensive experiments on six widely-used datasets and one large-scale dataset demonstrate the superiority of our proposed ClusMatch. For example, ClusMatch achieves a significant accuracy improvement of **5.4%** over the state-of-the-art method ProPos on an average of these six datasets. Source code can be found at <https://github.com/XY-ATOE/ClusMatch>.

Index Terms—Deep Clustering, Semi-supervised Learning, Positive and Negative Learning

1 INTRODUCTION

C LUSTERING is an unsupervised learning task that aims to group samples into different clusters, ensuring that samples within the same cluster are similar while those from different clusters are dissimilar. Traditional clustering methods, such as K-means [1], hierarchical clustering [2], spectral clustering [3], subspace clustering [4]–[8], concept factorization [9], [10], etc., usually rely on manually selected features and distance measures, which limits the performance and application of clustering. Deep clustering, on the other hand, provides a new perspective for clustering by automatically learning the representation without the need

for manually defined features [11]–[19], which can mine the intrinsic relationships within the data, leading to more informative clustering results. Compared with traditional methods, deep clustering often shows better robustness and resilience to noise, and is capable of efficiently handling large-scale data [15], [19]–[21].

Currently, there are two main paradigms for deep clustering, including the clustering head based methods and feature learning based methods. The first one tends to add a clustering head after the feature encoder, such as the joint unsupervised learning of deep representations and image clusters (JULE) [22], deep adaptive clustering [11], [13] and contrastive clustering [16], [17], [23], which learns the feature encoder and clustering head simultaneously through end-to-end training. The clustering head can map the feature representation to the clustering space and constrain the samples at a semantic level. Owing to the development of self-supervised learning, the feature learning paradigm has attracted much attention recently, which seamlessly integrates deep learning with traditional clustering techniques. This paradigm utilizes self-supervised learning, e.g., contrastive learning [21], [24]–[27], to train the encoder for discriminative feature representations, which are then fed into a traditional clustering algorithm, such as K-means [19]. Compared to clustering head based methods, this kind of method directly imposes constraints at the feature level on the shoulders of self-supervised learning, showing greater flexibility. Nevertheless, it is imperative to acknowledge that this approach involves a two-stage process, inevitably

Manuscript received 05 December 2023; revised 10 September 2024 and 19 March 2025; accepted 06 July 2025. This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62376069, 62276004, 62172261, and 62176137), in part by the Beijing Natural Science Foundation (Grant No. L257007), in part by the Young Elite Scientists Sponsorship Program by CAST (Grant No. 2023QNRC001), in part by Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515012027), in part by the Shenzhen Science and Technology Program (Grant Nos. KQTD20240729102207002 and ZDSYS20230626091203008), and in part by Jiangsu Science and Technology Major Program (Grant No. BG2024041). (Corresponding authors: Jianhua Yin and Liqiang Nie.)

Jianlong Wu and Liqiang Nie are with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: wujianlong@hit.edu.cn; nieliqiang@gmail.com).

Zihan Li, Wei Sun, and Jianhua Yin are with the School of Computer Science and Technology, Shandong University, Qingdao 266237, China (e-mail: zihanli@mail.sdu.edu.cn; sunweim@gmail.com; jhyin@sdu.edu.cn).

Zhouchen Lin is with the State Key Lab of General AI, School of Intelligence Science and Technology, Peking University, Beijing 100871, China, with the Institute for Artificial Intelligence, Peking University, Beijing 100871, China, and also with the Pazhou Laboratory (Huangpu), Guangzhou 510335, China (e-mail: zlin@pku.edu.cn).

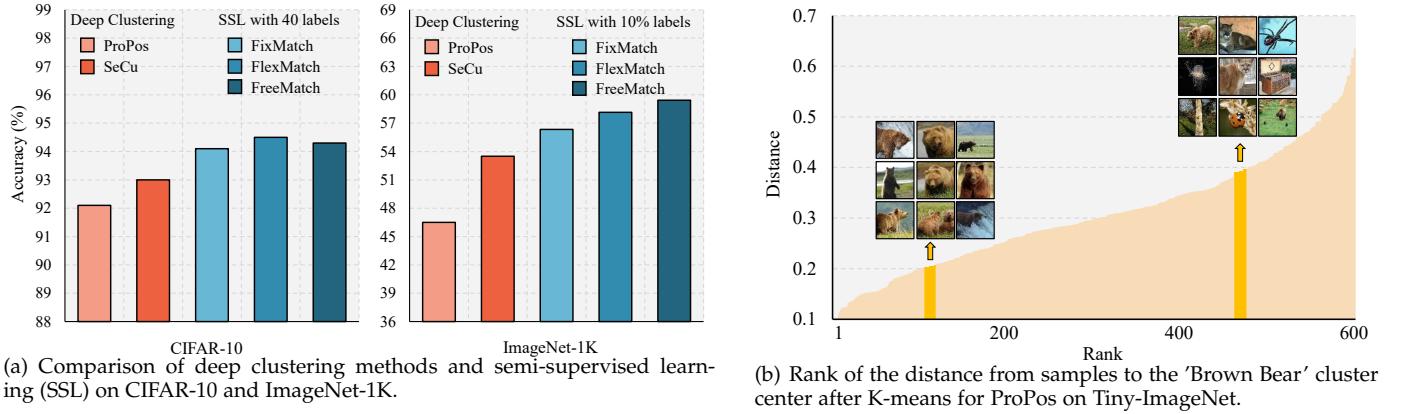


Fig. 1: Motivation of the proposed method. (a) Semi-supervised learning demonstrates better accuracy with only a small amount of labeled data, prompting us to consider revisiting the clustering task in a semi-supervised way. (b) The top-ranked samples all belonged to the correct ‘Brown Bear’ class, while the lower-ranked samples exhibit label confusion, indicating that the top-ranked samples are of high quality and can be treated as labeled data in semi-supervised learning.

requiring manual parameter selection for the traditional clustering algorithm [15], [28].

Both paradigms have achieved great success in recent years, but most of the existing methods view the clustering from a traditional unsupervised learning perspective. In this paper, we revisit the clustering task and formulate it as a pseudo-label based semi-supervised learning task. As depicted in Fig. 1 (a), it is evident that semi-supervised learning methods consistently outperform clustering methods, both on the small dataset CIFAR-10 and the large-scale dataset ImageNet-1K. For example, with only 4 labeled samples per category, the accuracy of the semi-supervised method FlexMatch [29] is 2.4% higher than that of the state-of-the-art clustering method ProPos [19] on CIFAR-10 [30]. Moreover, the accuracy of FreeMatch [31] with only 10% labeled samples for supervised learning surprisingly surpasses ProPos by 12.9% on ImageNet-1K [32]. This insight illuminates the potential that, if we can transform the clustering task into a semi-supervised learning paradigm in some way, the clustering performance could closely approximate that of semi-supervised learning. The key to this transformation lies in how to construct samples with high-quality labels. Note that the term ‘labels’ here refers to notations that distinguish different categories, rather than conventional labels in the true sense of the word. Luckily, we have another observation that the closer a sample is to the cluster center, the higher the credibility of its assigned label. Taking the feature learning based method ProPos as an example, after getting the clustering result by K-means, we sort the samples according to the distance to the cluster center. As shown in Fig. 1 (b), samples near the cluster center often share the same true class, while samples far from the cluster center tend to be of different wrong classes. Consequently, by selecting samples close to the clustering center as labeled samples, we can naturally transform the clustering task into semi-supervised learning. In addition, current clustering methods and semi-supervised learning mainly explore the consistency between samples through positive labels while ignoring the complementary information provided by negative labels, which also leads to limited supervised signals and poor performance.

In view of the above motivation, we propose ClusMatch, which further enhances the performance of deep clustering by semi-supervised fine-tuning with unified positive and negative pseudo-label learning. In order to satisfy the setting of semi-supervised learning, we first use the pre-trained clustering network to annotate positive labels for the samples and design two selection strategies for the clustering head based and feature learning based methods, respectively, to select high-quality samples for supervised training. For the unselected samples, we explore the consistency between samples of different views and utilize the positive pseudo-labels, negative pseudo-labels, and K-means pseudo-labels generated by the weakly-augmented branch as additional supervised signals to guide the learning of the strongly-augmented branch. To control the quality of the generated pseudo-labels, we also design an adaptive positive-negative threshold strategy, which can dynamically adjust the value according to the training status, thus avoiding the interference of low-confidence pseudo-labels. ClusMatch is pluggable and applicable to existing deep clustering methods. Extensive experiments on several commonly-used datasets verify the effectiveness of the proposed method.

Overall, the pivotal contributions of our work can be summarized as follows:

- We reconsider the clustering problem and propose ClusMatch based on a semi-supervised fine-tuning framework, which is pluggable and can further improve the performance of existing clustering methods.
- For the lack of supervised signals, we design novel selection strategies for high-quality samples and propose unified positive and negative pseudo-label learning. To the best of our knowledge, this is the first study that combines negative learning with the clustering task.
- In order to obtain high-confident pseudo-labels, we design an adaptive positive-negative threshold learning strategy, which can dynamically filter uninformative signals according to the learning status of

training.

- We conduct extensive experiments on six benchmark datasets and one large-scale dataset, and the proposed ClusMatch outperforms existing deep clustering methods by a large margin.

2 RELATED WORK

In this section, we provide a brief overview of the advancements and current research status in deep clustering and semi-supervised learning.

2.1 Deep Clustering

Recently, deep clustering [12], [17], [19], [33]–[36] has demonstrated remarkable advancements compared to conventional clustering methods. Image clustering can be broadly categorized into two groups according to evaluation methods: clustering head based methods, such as DeepCluster [37], CC [16], GCC [17], and TCL [23], and feature learning based methods, represented by DEC [38], BYOL [27], PCL [21], and ProPos [19].

Clustering head based methods use a clustering head to directly output predictions. DeepCluster [37] is among the earliest head-based methods designed for end-to-end clustering. It is compatible with many standard clustering algorithms, such as k-means, and requires minimal additional steps. At the time, it achieves state-of-the-art performance. CC [16], GCC [17], and TCL [23] all employ contrastive learning at the instance and cluster level. Especially, GCC contrasts images with their K-nearest neighbors while CC and TCL construct contrastive pairs through data augmentation.

Feature learning based methods, built upon the foundation of self-supervised learning, have demonstrated notable results [21], [25], [27]. They directly leverage features obtained from the pre-trained model and apply the K-means algorithm to yield favorable results. DEC [38] comes up with the joint optimization of deep embedding and clustering, along with a novel iterative refinement through soft assignment, achieving excellent results. Some methods utilize contrastive learning for clustering. BYOL [27] introduces a novel approach for contrastive learning by adding a prediction head instead of using negative samples to prevent the model from collapsing. PCL [21] proposes a prototypical contrastive loss based on MoCo-V2 [39] to enhance feature compactness. ProPos [19] further improves the prototypical contrastive loss based on BYOL [27] and achieves state-of-the-art results.

Owing to the powerful representation learning capability of self-supervised learning, various deep image clustering methods based on labeling technology have also emerged, such as SCAN [15] and SPICE [28]. SCAN introduces self-labeling, which enhances model performance by leveraging high-confidence pseudo-labels generated by a pre-trained self-supervised model. Self-labeling stands out as a versatile approach, demonstrating notable results in other models such as GCC [17] and SeCu [20]. Similarly, SPICE [28] leverages the pre-trained model to assign pseudo-labels by prototype and reliable labeling, which subsequently are utilized to refine the whole model.

Regardless of the clustering head based or feature learning based methods, they have not departed from the traditional unsupervised learning framework. In this paper, we revisit the clustering task from a novel perspective of semi-supervised learning.

2.2 Semi-supervised Learning

Semi-supervised learning is an approach that can produce significant results only with a small amount of labeled data. A common practice of semi-supervised learning is to utilize the confident predictions to generate pseudo-labels [40], [41] and train the model through entropy minimization [42]. MixMatch [43] determines highly-confident pseudo-labels through multiple augmentations. FixMatch [44] employs predefined hyperparameters as a positive threshold to determine which samples are used for training. However, FixMatch does not take into account variations in the learning difficulty of samples across different categories. FlexMatch [29] refines the FixMatch method by allowing dynamic threshold adjustments for different categories during training. It is worth noting that both of these methods unavoidably introduce hyperparameters. In contrast, FreeMatch [31] introduces an adaptive threshold approach that determines pseudo-labels without the need for any additional hyperparameters. SoftMatch [45] assigns different weights to pseudo-labels based on their quality. Furthermore, self-supervised learning methods such as MoCo [25], SimCLR [24], BYOL [27], etc., can also be easily transformed into semi-supervised learning by directly adding a supervised learning loss to the original loss or applying supervised learning with a few labeled samples based on the pre-trained model.

To further enhance the performance, several semi-supervised learning methods also combine negative learning which utilizes negative labels to construct loss functions. Negative labels are often easier to obtain compared to positive labels and can provide additional supervisory signals for model training. Typical negative learning includes complementary labels methods [46], [47] and NLNL [48]. NS³L [49] and UPS [50] introduce negative learning into semi-supervised learning. Both NS³L and UPS employ negative thresholds to filter out noisy samples with negative labels. FullMatch [51] employs a method that ranks negative labels based on the probability of different categories, achieving favorable results. These methods are limited to a fixed threshold or rank strategy, which is unreasonable and not flexible enough.

In our method, we transform the clustering task into semi-supervised learning, leveraging negative learning with a dynamic and flexible threshold strategy to achieve improved clustering performance.

3 METHOD

In this section, we introduce ClusMatch in detail. We first give a brief introduction to the deep clustering task, and then describe the whole framework of ClusMatch. Next, we elaborate on the proposed components of ClusMatch, including label selection, positive pseudo-label learning, and negative pseudo-label learning.

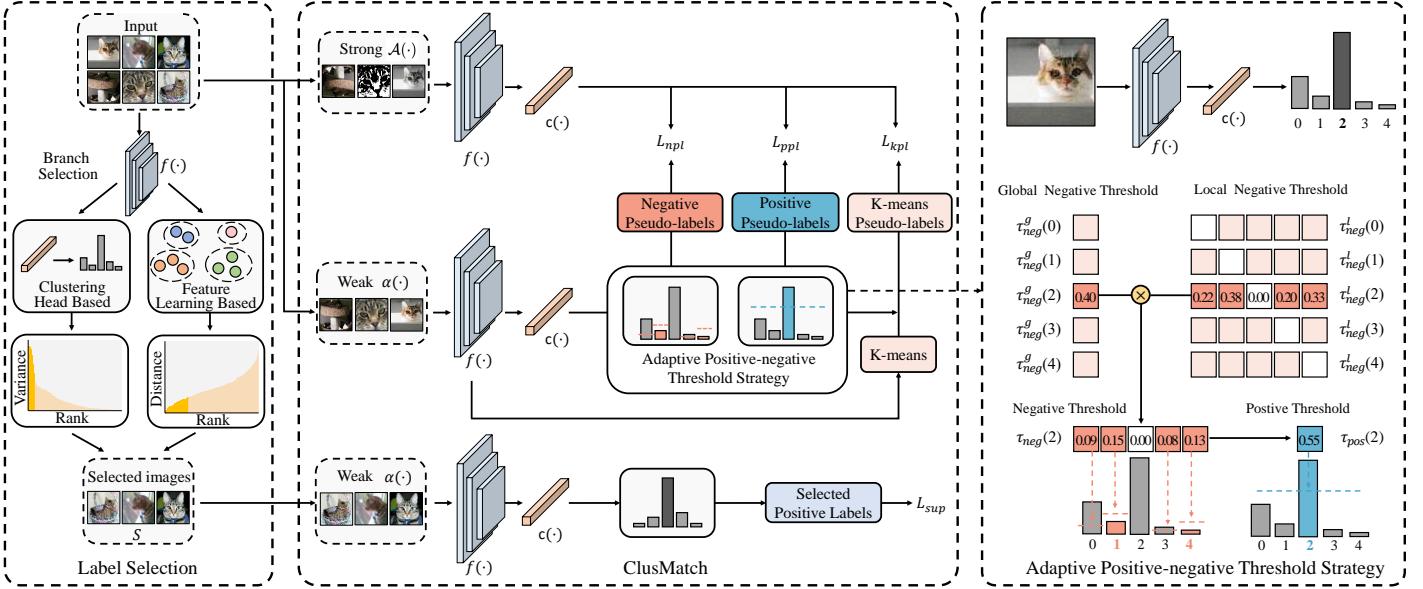


Fig. 2: The overall framework of ClusMatch. We first select high-quality samples using the pre-trained model using two designed strategies for supervised learning. Branch selection means choosing one of the two branches for label selection, depending on the type of pre-trained model. For unselected samples, the weakly-augmented branch generates high-confident positive, negative, and K-means pseudo-labels as the supervised signals to instruct the learning of the strongly-augmented branch. A novel adaptive positive-negative threshold strategy is also proposed to further boost the confidence of generated pseudo-labels.

3.1 Problem Formulation

Given N images $X = \{x_i\}_{i=1}^N$, the deep clustering task aims to group them into different K classes without using any labeled data, ensuring that similar images are clustered together as much as possible. Considering clustering head based methods, the typical framework consists of two main components, a feature encoder $f(\cdot)$ that transforms the image x_i into the corresponding feature, and a clustering head $c(\cdot)$ that maps the features to a K -dimensional probability vector p_i , which satisfies $\sum_{j=1}^K p_{ij} = 1$. Note that p_{ij} refers to the probability that the image x_i is clustered into the j category, and the category with the largest probability usually acts as the clustering result of the image. Finally, by designing a suitable loss function, we can jointly optimize encoder $f(\cdot)$ and clustering head $c(\cdot)$ to obtain a high-performance clusterer.

Different from the traditional approach of together training the encoder $f(\cdot)$ and clustering head $c(\cdot)$ end to end, in this work, we propose a novel approach to optimizing clustering networks that leverages semi-supervised learning to fine-tune the pre-trained encoder $f(\cdot)$ for more discriminative features.

3.2 Overview of ClusMatch

To facilitate the subsequent formulation, we first give the concepts of positive labels, negative labels, positive pseudo-labels, and negative pseudo-labels in this paper. The positive label indicates the category to which the sample is most likely to belong, and it is generally of the one-hot form. As an example, for a probability distribution p_i , the positive label should be the category corresponding to the maximum probability, i.e., $y_i^{pos} = \arg \max(p_i)$. The negative label, on

the other hand, is generally in the form of multi-hot, indicating those categories with non-maximum probabilities, to which the sample is unlikely to belong. In contrast, the definition of positive and negative pseudo-labels is based on the positive-negative threshold strategy. The category corresponding to the maximum probability above the positive threshold will serve as a positive pseudo-label, while the categories with the non-maximum probabilities below the negative thresholds collectively constitute the negative pseudo-label.

In this paper, we propose a new training approach, ClusMatch, which unifies positive and negative pseudo-label learning to fine-tune the pre-trained clustering network for more discriminative feature representation and better clustering performance. The overall framework is shown in Fig. 2. The training process of ClusMatch can be summarized into two stages:

1) Label Selection. Assign positive labels using the pre-trained clustering network for all samples, and select a few samples with high-quality positive labels, ensuring that the number of selected samples of each category is equal. The specific strategies of selection for clustering head based and feature learning based pre-trained networks will be introduced in detail in Section 3.3.

2) Semi-supervised Fine-tuning. Treat the selected samples as the labeled data and generate highly-confident pseudo-labels for unselected samples based on the proposed adaptive positive-negative threshold strategy to perform semi-supervised fine-tuning.

For samples $S = \{s_i, y_i\}_{i=1}^{N_s}$, which are selected and annotated with a positive label y_i in the label selection stage, we follow supervised learning for training. Specifically, we perform weak augmentation $\alpha(\cdot)$ on the sample s_i , after

which they are passed through the encoder $f(\cdot)$ and the clustering head $c(\cdot)$ to obtain the assignment probability. Finally, the cross-entropy between the probability and positive label y_i is calculated as follows:

$$L_{sup} = \frac{1}{B} \sum_{i=1}^B H(y_i, c(f(\alpha(s_i)))), \quad (1)$$

where B denotes the number of labeled samples in a mini-batch and $H(\cdot)$ is the cross-entropy function.

For unselected samples $U = \{u_i\}_{i=1}^{N_u}$, we adopt pseudo-label learning, i.e., we leverage the weakly-augmented branch to generate highly-confident pseudo-labels to guide the learning of the strongly-augmented branch. ClusMatch mainly considers three forms of pseudo-labels, namely, positive pseudo-labels, negative pseudo-labels, and K-means pseudo-labels. In order to improve the confidence of pseudo-labels, we design an adaptive positive-negative threshold strategy, which can automatically adjust the value of the positive and negative thresholds according to the training status. Based on this, unified positive and negative pseudo-label learning is proposed to optimize the entire clustering network, which will be presented in Sections 3.4 and 3.5.

Considering that the clustering head $c(\cdot)$ is randomly initialized at the early stage of training, its output is not convincing. In contrast, K-means results of features are more reliable since the encoder $f(\cdot)$ is a pre-trained deep clustering network. In order to familiarize the whole network with the samples as soon as possible, we also propose K-means pseudo-label learning to warm up the training, where K-means results of the weakly-augmented branch acting as K-means pseudo-labels instruct the strongly-augmented branch:

$$p_i = c(f(\alpha(u_i))), \quad (2)$$

$$L_{kpl} = \frac{1}{\mu B} \sum_{i=1}^{\mu B} \mathbb{1}(\max(p_i) > \tau_{pos}) H(y'_i, c(f(\mathcal{A}(u_i)))), \quad (3)$$

where $\mathbb{1}(\cdot)$ is the indicator function, μ is the ratio of the number of samples for pseudo-label learning and samples for supervised learning in a mini-batch, $\mathcal{A}(\cdot)$ denotes strong augmentation, and y'_i is the K-means pseudo-label after Hungarian algorithm, following ProPos [19]. Note that only samples whose maximum probability exceeds the positive threshold τ_{pos} are used in K-means pseudo-label learning because they tend to be simpler and show more convincing cluster results than those not exceeding.

Finally, we can get the following overall training objective in the semi-supervised fine-tuning stage:

$$L_{total} = L_{sup} + \delta \cdot L_{ppl} + \beta \cdot L_{npl} + \gamma \cdot L_{kpl}, \quad (4)$$

where L_{ppl} is the positive pseudo-label learning loss, and L_{npl} is the negative pseudo-label learning loss. δ , β and γ are constants for balancing the contribution of each term.

3.3 Label Selection

Given a pre-trained deep clustering network, the goal of label selection is to annotate all samples with positive labels and pick out high-quality ones. In other words, the aim is

to select $M = \lfloor \frac{\sigma \cdot N}{K} \rfloor$ samples, whose positive labels are as accurate as possible, for each category. Here σ is a ratio controlling the number of selected samples and $0 \leq \sigma \leq 1$.

For clustering head based methods, we can get an assignment probability matrix $P = [p'_1, p'_2, \dots, p'_N]^T \in R^{N \times K}$ from the clustering head $c(\cdot)$ of the pre-trained model. p'_i denotes the probabilities over K clusters and the category corresponding to the maximum probability of p'_i is treated as the positive label, i.e., $y_i = \arg \max(p'_i)$. In order to filter out the low-quality samples, we only retain the samples with large variances of p'_i . In general, the larger the variance of the probability distribution, the greater the distinction between different categories, and thus the more credible the positive labels [50]. Taking the category a as an example, we first compute the variance v_i of the probability distribution p'_i for all samples with $\arg \max(p'_i) = a$ and then select the top- M samples with the largest v_i .

As to feature learning based methods, we take the result of the K-means algorithm on the features as the positive label. By K-means, we can get a distance matrix $D = [d_1, d_2, \dots, d_N]^T \in R^{N \times K}$, each element d_{ij} denotes the distance from the i -th sample to the j -th clustering center. The category corresponding to the clustering center closest to the sample acts as the positive label, i.e., $y_i = \arg \min(d_i)$. From Fig. 1 (b), we see that the closer the samples are to the clustering center, the higher the confidence of the positive label. Therefore, for samples with positive label a , we select the top- M samples with the smallest $\min(d_i)$ as the high-quality samples.

By choosing one of the two branches for label selection depending on the type of deep clustering model, we can construct set $S = \{s_i, y_i\}_{i=1}^{N_s}$ for supervised learning. Here $N_s = K \times M$.

3.4 Negative Pseudo-label Learning

Let us first consider conventional cross-entropy loss. The optimization objective of cross-entropy is as follows:

$$L = H(y_i^{pos}, p_i) = - \sum_{k=1}^K y_{ik}^{pos} \log p_{ik}, \quad (5)$$

where y_i^{pos} denotes the one-hot positive label. Cross-entropy expects the probability corresponding to the true label to be close to 1. In this paper, we call it positive learning. In contrast, the objective function for negative learning is:

$$L = H(y_i^{neg}, 1 - p_i) = - \sum_{k=1}^K y_{ik}^{neg} \log(1 - p_{ik}), \quad (6)$$

where y_i^{neg} is the negative label of the one-hot form, indicating the category to which the sample is least likely to belong. The objective aims to make the probability corresponding to this category close to zero. Generally, the negative label is in the form of multi-hot, to constrain the sum of probabilities for all negative classes approaching zero, we modify the optimization objective to:

$$L = - \log(1 - \sum_{k \in \{j | y_{ij}^{neg} = 1\}} p_{ik}). \quad (7)$$

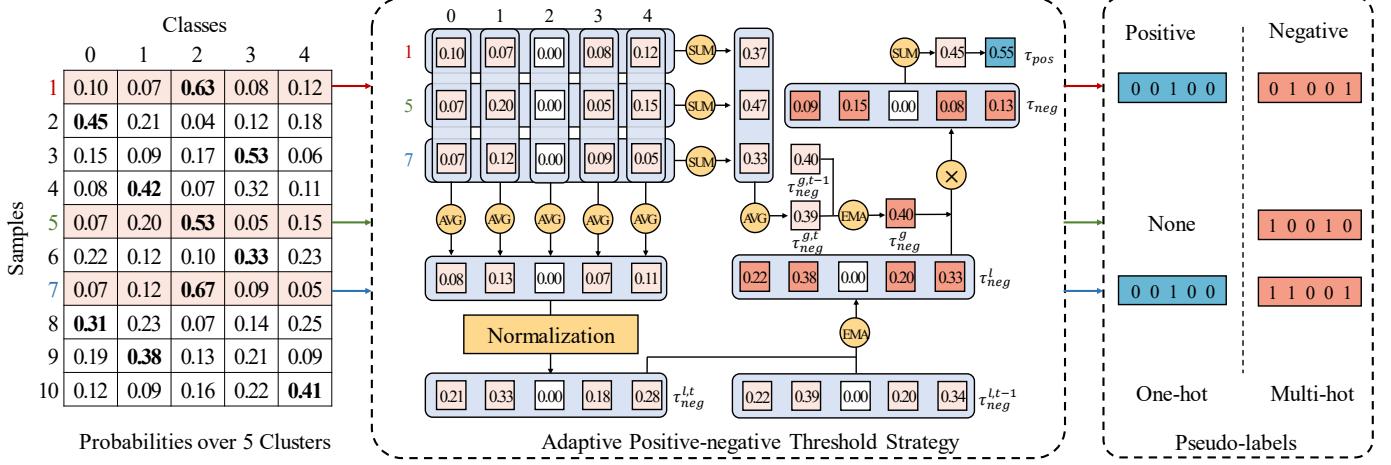


Fig. 3: A toy example for adaptive positive-negative threshold learning strategy. We provide a detailed presentation of the dynamic generation process for positive and negative thresholds of class 2. Initially, 10 samples are given with probability distributions over five classes. Subsequently, samples 1, 5, and 7, whose maximum probabilities correspond to class 2, participate in the threshold construction. The current positive and negative thresholds are then obtained after applying our proposed dynamic strategy. Finally, the involved samples whose probability of class 2 exceeds the positive threshold will generate the one-hot positive pseudo-label, such as samples 1 and 7. Other classes whose probabilities fall below the corresponding negative threshold will constitute the multi-hot negative pseudo-label.

Most of the existing clustering head based methods are based on positive learning and ignore the additional information brought by negative labels. Through negative learning, the clustering network can avoid allocating more effort to impossible categories and fix positive labels faster and more accurately, thus improving the clustering performance.

According to the above theory, unified positive and negative pseudo-label learning based on the adaptive positive-negative threshold strategy is proposed, which combines pseudo-label learning with positive and negative learning for more highly-confident supervised signals. Firstly, we elaborate on the design of the adaptive negative threshold. According to FlexMatch [29] and FreeMatch [31], samples from different categories have varying levels of learning difficulty. So each category should have a corresponding threshold value, which can be adjusted according to the training status.

We define the negative threshold for category a as follows:

$$\tau_{neg}(a) = \tau_{neg}^g(a) \cdot Norm(\tau_{neg}^l(a)), \quad (8)$$

where $\tau_{neg}^g(a)$ is a scalar reflecting the global learning status of all categories except a , while $\tau_{neg}^l(a)$ is a K-dimensional vector indicating the local learning status of each category except a . Note that $Norm(\cdot)$ is the L1-normalization.

For samples whose maximum probability corresponds to the category a , their negative pseudo-labels should be constructed from non- a categories. To assess the overall learning status of them, we define the global threshold $\tau_{neg}^g(a)$ as follows:

$$\tau_{neg}^g(a) = \frac{\sum_{i=1}^{\mu B} \mathbb{1}(\arg \max(p_i) = a)(1 - \max(p_i))}{\sum_{i=1}^{\mu B} \mathbb{1}(\arg \max(p_i) = a)}, \quad (9)$$

which measures the global learning status by the expectation of the sum of the predicted probabilities of non- a categories.

The local learning status of non- a categories is equally significant, which specifically reflects the difficulty of learning different categories. In this regard, we treat the expectation of the predicted probability distributions over non- a categories as the measure:

$$\tau_{neg}^l(a) = \frac{\sum_{i=1}^{\mu B} \mathbb{1}(\arg \max(p_i) = a)\hat{p}_i}{\sum_{i=1}^{\mu B} \mathbb{1}(\arg \max(p_i) = a)}, \quad (10)$$

where \hat{p}_i denotes the result after setting $\max(p_i)$ to 0 in p_i . In accordance with FreeMatch [31], we adopt the exponential moving average (EMA) to update both $\tau_{neg}^g(a)$ and $\tau_{neg}^l(a)$ as follows:

$$\tau_{neg}^{g,t}(a) = m \cdot \tau_{neg}^{g,t-1}(a) + (1 - m)\tau_{neg}^{g,t}(a), \quad (11)$$

$$\tau_{neg}^{l,t}(a) = m \cdot \tau_{neg}^{l,t-1}(a) + (1 - m)\tau_{neg}^{l,t}(a), \quad (12)$$

where $m \in [0, 1]$ is a smoothing factor and t is the current training epoch. From Eq. (8), it can be deduced that $\tau_{neg}(a)$ is also K-dimensional. Intuitively, all categories in p_i with the probabilities lower than the corresponding thresholds in $\tau_{neg}(a)$ will constitute the negative pseudo-label together. According to Eq. (7), we optimize the following negative pseudo-label learning loss to make the probabilities below the corresponding thresholds converge to 0:

$$q_i = c(f(\mathcal{A}(u_i))), \quad (13)$$

$$L_{npl} = -\frac{1}{\mu B} \sum_{i=1}^{\mu B} \log \left(1 - \sum_{j=1}^K \mathbb{1}(p_{ij} < [\tau_{neg}(y_i'')]_j) q_{ij} \right), \quad (14)$$

where y_i'' is the positive label of sample u_i , i.e., $y_i'' = \arg \max(p_i)$.

Training only on negative classes without positive ones can harm classification, as the model cannot learn what

positive samples look like, making it hard to infer them from negative class analysis alone. On datasets like ImageNet, small batches may lack positive samples for some classes, risking training collapse. Unlike SeCu, ClusMatch solves this by using a pre-trained model to select a small set of high-confidence labeled samples, ensuring balanced class representation. Each batch mixes labeled and unlabeled samples. This ensures all labeled samples are used within a few batches, allowing the model to quickly see positive samples from every class. Without labeled samples, the model might miss positive samples for specific classes over many batches. Thus, labeled samples stabilize learning, improve class balance, and enhance negative class learning.

3.5 Positive Pseudo-label Learning

Naturally, the category corresponding to the maximum probability tends to be the true label of the sample. Therefore, we propose positive pseudo-label learning, i.e., the category whose maximum probability exceeds the positive threshold will act as the positive pseudo-label to instruct the strongly-augmented branch. Based on the negative threshold, we construct the adaptive positive threshold as follows:

$$\tau_{pos}(a) = 1 - \sum_{j=1}^K [\tau_{neg}(a)]_j. \quad (15)$$

The learning status of the negative threshold determines the value of $\tau_{pos}(a)$. As the network becomes more and more certain of the negative labels, the negative threshold decreases, thus making the positive threshold larger and larger, which indicates that the network is more capable of recognizing the positive labels. Note that unlike $\tau_{neg}(a)$, $\tau_{pos}(a)$ is a scalar since we only need to consider whether the maximum probability reaches its corresponding positive threshold. A detailed toy example of constructing positive and negative thresholds is shown in Fig. 3. Ultimately, the positive pseudo-labeling learning loss can be formulated as:

$$L_{ppl} = \frac{1}{\mu B} \sum_{i=1}^{\mu B} \mathbb{1}(\max(p_i) > \tau_{pos}(y''_i)) H(y''_i, q_i). \quad (16)$$

The overall training process is summarized in Algorithm 1.

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 Datasets

We evaluated the performance of ClusMatch on seven mainstream datasets, which cover a large variety of instances and categories, including CIFAR-10 [30], CIFAR-100 [30], STL-10 [52], and four ImageNet-related datasets, i.e., ImageNet-10 [11], ImageNet-Dogs [11], Tiny-ImageNet [53], and ImageNet-1K [32].

CIFAR-10 and **CIFAR-100** are the most commonly-used datasets for image tasks, both consisting of 60,000 color images with a size of 32x32. CIFAR-10 contains a total of 10 classes, while CIFAR-100 has a total of 20 superclasses, each of which consists of 5 subclasses. Note that we only consider the superclass for CIFAR-100 in this work.

Algorithm 1 Training Algorithm of ClusMatch

Input: Images $X = \{x_i\}_{i=1}^N$, number of clusters K , a pre-trained clustering network f with parameters θ_f , balancing weights δ, β and γ , label selection ratio σ , and momentum parameter m .

Output: Clustering Results C .

Initializing the clustering head c with parameters θ_c ;

Selecting high-quality samples using the pre-trained f to construct set S ;

for all each epoch **do**

- Generating K-means pseudo-labels;
- for** all each batch **do**

 - Calculating negative and positive thresholds according to Eqs. (8) and (15);
 - Calculating supervised learning loss for the selected samples in S by Eq. (1);
 - Calculating positive pseudo-label learning loss for unselected samples by Eq. (16);
 - Calculating negative pseudo-label learning loss for unselected samples by Eq. (14);
 - Calculating K-means pseudo-label learning loss for unselected samples by Eq. (3);
 - Update parameters θ_f and θ_c with SGD optimizer by minimizing the total loss according to Eq. (4).

- end for**

end for

Generating clustering results C using K-means on the features of f .

STL-10 contains both labeled and unlabeled samples. The labeled samples consist of 10 classes of object images, where each class contains 1,300 images of size 96x96. There are also 100,000 unlabeled samples which we do not use in this work.

ImageNet-10 and **ImageNet-Dogs** are subsets of the ImageNet dataset with image size 96x96. ImageNet-10 contains 10 random categories with a total of 13,000 training images and ImageNet-Dogs has 19,500 images across 15 random dog breeds.

Tiny-ImageNet is also a subset of ImageNet, comprising 200 categories, with each category consisting of 500 training images sized at 64x64 pixels.

ImageNet-1K is a challenging large-scale image dataset, which contains 1,281,167 images for training with size of 224x224 across 1,000 categories.

4.1.2 Compared Methods

We compared ClusMatch with traditional clustering methods and the latest deep clustering methods, including K-means [1], SC [54], AC [55], NMF [10], AE [56], DAE [57], DCGAN [58], DeCNN [59], VAE [60], JULE [22], DEC [38], DAC [11], DCCM [13], IIC [61], PICA [62], SCAN [15], NNM [63], CC [16], MiCE [64], TCL [23], TCC [18], MoCo [25], SimSiam [26], BYOL [27], IDFD [65], PCL [21], SeCu [20], GCC [17] and ProPos [19].

4.1.3 Evaluation Metrics

Following the conventional practice of clustering, we adopted Accuracy (ACC), Normalized Mutual Information

TABLE 1: Clustering results of various methods on six widely-used datasets. The best and second-best results are shown in bold and underline, respectively.

Dataset	CIFAR-10			CIFAR-100			STL-10			ImageNet-10			Imagenet-Dogs			Tiny-ImageNet		
	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
K-means	0.087	0.229	0.049	0.084	0.130	0.028	0.125	0.192	0.061	0.119	0.241	0.057	0.055	0.105	0.020	0.065	0.025	0.005
AC (PR 1978)	0.105	0.228	0.065	0.098	0.138	0.034	0.239	0.332	0.140	0.138	0.242	0.067	0.037	0.139	0.021	0.069	0.027	0.005
SC (NeurIPS 2004)	0.103	0.247	0.085	0.090	0.136	0.022	0.098	0.159	0.048	0.151	0.274	0.076	0.038	0.111	0.013	0.063	0.022	0.004
AE (NeurIPS 2006)	0.239	0.314	0.169	0.100	0.165	0.048	0.250	0.303	0.161	0.210	0.317	0.152	0.104	0.185	0.073	0.131	0.041	0.007
NMF (IJCAI 2009)	0.081	0.190	0.034	0.079	0.118	0.026	0.096	0.180	0.046	0.132	0.230	0.065	0.044	0.118	0.016	0.072	0.029	0.005
DAE (JMLR 2010)	0.251	0.297	0.163	0.111	0.151	0.046	0.224	0.302	0.152	0.206	0.304	0.138	0.104	0.190	0.078	0.127	0.039	0.007
DeCNN (CVPR 2010)	0.240	0.282	0.174	0.092	0.133	0.038	0.227	0.299	0.162	0.186	0.313	0.142	0.098	0.175	0.073	0.111	0.035	0.006
VAE (ICLR 2014)	0.245	0.291	0.167	0.108	0.152	0.040	0.200	0.282	0.146	0.193	0.334	0.168	0.107	0.179	0.079	0.113	0.036	0.006
DCGAN (ICLR 2016)	0.265	0.315	0.176	0.120	0.151	0.045	0.210	0.298	0.139	0.225	0.346	0.157	0.121	0.174	0.078	0.135	0.041	0.007
JULE (CVPR 2016)	0.192	0.272	0.138	0.103	0.137	0.033	0.182	0.277	0.164	0.175	0.300	0.138	0.054	0.138	0.028	0.102	0.033	0.006
DEC (ICML 2016)	0.257	0.301	0.161	0.136	0.185	0.050	0.276	0.359	0.186	0.282	0.381	0.203	0.122	0.195	0.079	0.115	0.037	0.007
DAC (ICCV 2017)	0.396	0.522	0.306	0.185	0.238	0.088	0.366	0.470	0.257	0.394	0.527	0.302	0.219	0.275	0.111	0.190	0.066	0.017
DCCM (ICCV 2019)	0.496	0.623	0.408	0.285	0.327	0.173	0.376	0.482	0.262	0.608	0.710	0.555	0.321	0.383	0.182	0.224	0.108	0.038
IIC (ICCV 2019)	0.513	0.617	0.411	-	0.257	-	0.431	0.499	0.295	-	-	-	-	-	-	-	-	-
PICA (CVPR 2020)	0.591	0.696	0.512	0.310	0.337	0.171	0.611	0.713	0.531	0.802	0.870	0.761	0.352	0.352	0.201	0.277	0.098	0.040
SCAN (ECCV 2020)	0.797	0.883	0.772	0.486	0.507	0.333	0.698	0.809	0.646	-	-	-	-	-	-	-	-	-
MoCo (CVPR 2020)	0.669	0.776	0.608	0.390	0.397	0.242	0.615	0.728	0.524	0.01	0.010	0.010	0.347	0.338	0.197	0.342	0.160	0.080
SimSiam (ICML 2020)	0.786	0.856	0.736	0.522	0.485	0.327	0.659	0.716	0.572	0.831	0.921	0.833	0.583	0.674	0.501	0.351	0.203	0.094
BYOL (NeurIPS 2020)	0.817	0.894	0.790	0.559	0.569	0.393	0.713	0.825	0.657	0.866	0.939	0.872	0.635	0.694	0.548	0.365	0.199	0.100
MiCE (ICLR 2020)	0.737	0.835	0.698	0.436	0.440	0.280	0.635	0.752	0.575	-	-	-	0.423	0.439	0.286	-	-	-
CC (AAAI 2021)	0.705	0.790	0.637	0.431	0.429	0.266	0.764	0.85	0.726	0.859	0.893	0.822	0.445	0.429	0.274	0.340	0.140	0.071
TCC (NeurIPS 2021)	0.790	0.906	0.733	0.479	0.491	0.312	0.732	0.814	0.689	0.848	0.897	0.825	0.554	0.595	0.417	-	-	-
IDFD (ICLR 2021)	0.711	0.815	0.663	0.426	0.425	0.264	0.643	0.756	0.575	0.898	0.954	0.901	0.546	0.591	0.413	-	-	-
PCL (ICLR 2021)	0.802	0.874	0.766	0.528	0.526	0.363	0.718	0.810	0.670	0.841	0.907	0.822	0.440	0.412	0.299	0.350	0.159	0.087
NNM (CVPR 2021)	0.748	0.843	0.709	0.484	0.477	0.316	0.694	0.808	0.65	-	-	-	-	-	-	-	-	-
TCL (IJCV 2022)	0.819	0.887	0.780	0.529	0.531	0.357	0.799	0.868	0.757	0.875	0.895	0.837	0.623	0.644	0.516	-	-	-
SeCu (ICCV 2023)	0.861	0.930	0.857	0.551	0.552	0.397	0.733	0.836	0.693	-	-	-	-	-	-	-	-	-
GCC (ICCV 2021)	0.787	0.870	0.752	0.498	0.494	0.320	0.684	0.788	0.631	0.842	0.901	0.822	0.583	0.630	0.479	0.350	0.165	0.074
GCC + ClusMatch	0.851	0.914	0.829	0.552	0.536	0.367	0.739	0.842	0.702	0.897	0.956	0.906	0.678	0.730	0.596	0.403	0.225	0.109
ProPos (TPAMI 2022)	0.853	0.921	0.844	0.576	0.576	0.409	0.819	0.905	0.808	0.895	0.941	0.884	0.650	0.668	0.546	0.471	0.318	0.188
ProPos + ClusMatch	0.890	0.949	0.893	0.601	0.592	0.426	0.863	0.929	0.854	0.939	0.977	0.949	0.745	0.798	0.685	0.482	0.408	0.232
ProPos [†]	0.871	0.943	0.863	0.591	0.602	0.437	0.854	0.923	0.844	0.902	0.959	0.912	0.725	0.768	0.661	0.402*	0.241*	0.121*
ProPos [†] + ClusMatch	0.909	0.960	0.915	0.627	0.612	0.457	0.886	0.940	0.873	0.933	0.974	0.944	0.785	0.844	0.737	0.454*	0.388*	0.213*

[†] indicates that ProPos uses ResNet-34 except on the Tiny-ImageNet.

* indicates that ProPos uses ResNet-18 with image size 224 × 224 on the Tiny-ImageNet.

(NMI), and Adjusted Rand Index (ARI) as the evaluation metrics, where larger values indicate better clustering performance.

4.1.4 Implementation Details

Model. Considering that ClusMatch is a pluggable scheme, we relied on the state-of-the-art clustering head based method GCC [17] and the feature learning based method ProPos [19] to build our model. Following GCC and ProPos, we used ResNet-18 [66] as the encoder $f(\cdot)$ unless otherwise stated and the clustering head $c(\cdot)$ is a linear layer.

Training Settings. For a fair comparison, we strictly followed the parameter settings of GCC [17] and ProPos [19] for pre-training. In the stage of semi-supervised fine-tuning, we trained a total of 2^{16} steps using SGD with a weight decay of 0.0005 and a momentum coefficient of 0.9. Different learning rates are adopted for the pre-trained encoder $f(\cdot)$ and head $c(\cdot)$. Unless specified otherwise, the learning rate for $f(\cdot)$ is set to 0.0002, while for $c(\cdot)$ it is set to 0.02. The loss weights δ , β , and γ are all set to 0.5. The ratio σ for label selection is 0.1 and m for EMA is set to 0.999. Especially, for CIFAR-10, both $f(\cdot)$ and $c(\cdot)$ are trained with a learning rate of 0.01, $\delta = 0.001$, $\beta = 1.0$, $\gamma = 0.5$, and $\sigma = 0.01$, since it is a relatively simple dataset. For the large-scale dataset ImageNet-1K, we set $\sigma = 0.2$.

Augmentation. Following FixMatch [44], we applied both weak and strong augmentation [67]. For weak augmentation, random crop and random horizontal flip are utilized. As for strong augmentation, we adopted the RandAugment

[68] strategy, which selects augmentations from a predefined list and combines them with different magnitudes. In the test phase, we used the center crop and kept the image size consistent with that during training.

4.2 Main Results

To verify the effectiveness and superiority of ClusMatch, we compared it with the state-of-the-art clustering methods on six commonly-used datasets. Table 1 presents the results for the three metrics NMI, ACC, and ARI. We can see that all metrics are improved significantly after fine-tuning GCC and ProPos with ClusMatch on all datasets. Note that here we reported the K-means clustering results for the features from the encoder $f(\cdot)$. The results for DAE, VAE, MoCo, SimSiam, and BYOL are cited from ProPos, where the cluster labels are directly obtained through feature clustering.

CIFAR-10/100. With ResNet-18 [66], ClusMatch improves 6.4%, 4.4%, and 7.7% on NMI, ACC, and ARI metrics, respectively, over GCC, and 3.7%, 2.8%, and 4.9% over ProPos, on the CIFAR-10 dataset. Even when compared with the most competitive method SeCu, our results are overwhelmingly superior, especially on the NMI and ARI metrics. For CIFAR-100, ClusMatch can get a steady boost of 4%~5% on all metrics based on GCC and 1%~3% based on ProPos.

STL-10. For the challenging STL-10, ClusMatch surpasses GCC by 5.5%, 5.4%, and 7.1% on three metrics. The performance of ProPos is also substantially improved after

TABLE 2: Clustering results on ImageNet-1K based on ResNet-50. * denotes that the results of SeCu are based on our reproduction.

Method	NMI	ACC	ARI
MoCo	0.619	0.305	0.143
TCL	0.671	0.379	0.266
SCAN	0.720	0.399	0.275
SeCu*	0.739	0.521	0.404
SeCu*+ClusMatch	0.763	0.542	0.425
ProPos	0.686	0.465	0.339
ProPos + ClusMatch	0.780	0.634	0.509

TABLE 3: Semi-supervised learning results comparison on CIFAR-10 and CIFAR-100 under ResNet-18. 40 and 400 labeled samples are randomly selected for CIFAR-10 and CIFAR-100, respectively. Metric: ACC.

Dataset	CIFAR-10	CIFAR-100
FixMatch	0.9410	0.6228
FlexMatch	0.9448	0.6799
FreeMatch	0.9438	0.6859
ClusMatch	0.9452	0.6933

adopting ClusMatch, both with ResNet-18 [66] and ResNet-34 [66]. In particular, ClusMatch gained 4.4% and 4.6% improvements on NMI and ARI when using ResNet-18. Even when ProPos uses the larger ResNet-34, its clustering performance is still inferior to that of ClusMatch using ResNet-18.

ImageNet-10/Dogs. On ImageNet-10, the performance of ClusMatch is on par with that of GCC but generally obtains around 3% to 4% improvements compared to ProPos. Notably, ClusMatch performs well on the fine-grained dataset ImageNet-Dogs. Under the ResNet-18, ClusMatch gains 10% or more improvements on all metrics compared to GCC and ProPos, which indicates that ClusMatch does well in fine-grained clustering.

Tiny-ImageNet. ClusMatch shows an obvious superiority for ACC metric on Tiny-ImageNet. With 64×64 image size, ClusMatch gains 6.0% compared to GCC and outperforms ProPos by a large margin of 9.0%. We also found that when the image size is set to 224×224 , all the metrics of ProPos drop drastically, close to 7%, but the performance of ClusMatch drops slightly, only about 2%, which also demonstrates the robustness of our ClusMatch.

4.3 Results on the ImageNet-1K Dataset

To assess the performance of ClusMatch in scenarios with large-scale data, we also conducted experiments on ImageNet-1K. For the fairness of the experiments, all methods used ResNet-50 [66] as the backbone. The results are shown in Table 2. We copied the results of several compared methods from TCL [23] and SeCu [20]. The results of SeCu [20] and SeCu+ClusMatch come from our reproduction based on its released code, since the checkpoint is not given. For ProPos [19] and ProPos+ClusMatch, we measured the three metrics using the released checkpoint.

As can be seen from Table 2, ClusMatch achieves the best results on all three metrics. Specifically, the ACC of ClusMatch reaches 63.4%, which is more than twice that

TABLE 4: Effectiveness of positive, negative, and K-means pseudo-label learning loss. Method: ProPos + ClusMatch. Metric: ACC.

L_{kpl}	L_{ppl}	L_{npl}	CIFAR-10	CIFAR-100
			0.561	0.521
		✓	0.806	0.560
	✓		0.895	0.584
✓			0.563	0.561
	✓	✓	0.898	0.585
✓		✓	0.881	0.567
✓	✓		0.940	0.585
✓	✓	✓	0.949	0.592

TABLE 5: Effectiveness of our adaptive threshold strategy. Method: ProPos + ClusMatch. Metric: NMI.

Method	CIFAR-10	CIFAR-100	Imagenet-Dogs	Tiny-ImageNet
Fixed threshold	0.887	0.572	0.728	0.467
ClusMatch	0.890	0.601	0.745	0.482

of the classical unsupervised method MoCo. ClusMatch exhibits significant gains over ProPos, with 9.4%, 16.9%, and 17.0% improvements on the three metrics, respectively. By incorporating ClusMatch with SeCu, the results can also be further improved. On the whole, the overwhelming results indicate that ClusMatch can maintain excellent performance even with large-scale datasets.

4.4 Results under Semi-Supervised Learning Settings

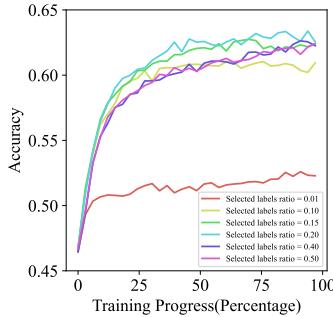
We conducted additional experiments on CIFAR-10 and CIFAR-100 to verify the effectiveness of our approach in semi-supervised learning. Following the settings in image clustering, we adopted ResNet-18 as the backbone network for all three compared methods and primarily focused on the 20 superclasses of CIFAR-100. In this section, no pre-training was performed, and the experiments were conducted under semi-supervised learning with randomly initialized parameters. We randomly selected 40 and 400 labeled samples for CIFAR-10 and CIFAR-100, respectively. We randomly ran the experiments two times and reported the average accuracy. As shown in Table 3, our method achieves better results than these three semi-supervised learning methods, demonstrating the effectiveness of ClusMatch.

4.5 Ablation Study and Overall Analysis

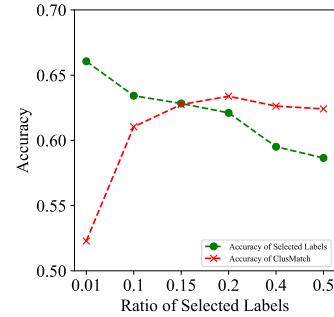
We conducted sufficient ablation experiments to explore the effectiveness of our proposed components, including positive and negative pseudo-label learning, K-means pseudo-label learning, adaptive positive and negative threshold strategy, label selection strategy, and data augmentation. To demonstrate the robustness of ClusMatch, we also analyzed the sensitivity of hyperparameters, including the label selection ratio and the weight of each loss. Note that for experiments involving ImageNet-1K, we utilized ResNet-50 as the backbone, while ResNet-18 is used for all other datasets

4.5.1 Effectiveness of Each Component

Positive, Negative, and K-means Pseudo-label Learning. Pseudo-label learning is essential in our approach, which



(a) Accuracy with different label selection ratios as the training processes.



(b) The relationship among accuracy, the number and the quality of the selected samples.

Fig. 4: Effect of label selection ratio σ on ImageNet-1K. Method: ProPos + ClusMatch. Metric: ACC.TABLE 6: Effectiveness of augmentation. $\alpha(\cdot)$ means weak augmentation, and $\mathcal{A}(\cdot)$ means strong augmentation. Metric: ACC.

Upper Path	Middle Path	CIFAR-10	CIFAR-100
$\alpha(\cdot)$	$\alpha(\cdot)$	0.613	0.452
$\mathcal{A}(\cdot)$	$\mathcal{A}(\cdot)$	0.494	0.558
$\alpha(\cdot)$	$\mathcal{A}(\cdot)$	0.917	0.532
$\mathcal{A}(\cdot)$	$\alpha(\cdot)$	0.949	0.592

can provide more supervised signals for model training. We also proposed K-means pseudo-label learning to warm up the entire clustering network to quickly familiarize the network with the samples during the early stages of training. As shown in Table 4, when we remove all these three pseudo-label learning strategies, the ACC drops sharply, especially on CIFAR-10, with a decrease of 38.8%. By adding each pseudo-label learning module, there is a significant improvement in the results. Moreover, the best result is achieved when all methods are combined.

Adaptive Positive-negative Threshold Strategy. The adaptive threshold strategy can filter out the noisy signals, thereby enhancing the confidence of the generated positive and negative pseudo-labels. To prove the effectiveness of our dynamic threshold strategy, we fixed the positive and negative thresholds to 0.99 and 0.01, respectively, serving as the baseline. Comparison results are shown in Table 5, we can find that our proposed adaptive threshold strategy has a stable NMI improvement of about 2% over the baseline on three different datasets.

Augmentation. In ClusMatch, we combined strong and weak augmentations to train the entire network, which plays a crucial role in the experimental results. To verify its validity, we studied different combinations of weak and strong augmentation. As shown in Fig. 2, the upper path of ClusMatch is used for logits generation, and the middle path can generate labels. We presented the results in Table 6, when only weak or strong augmentation is applied to both branches, there is a significant accuracy drop, especially for CIFAR-10, where the drop reaches 33.6% and 45.5%, respectively. When we applied strong augmentation on the upper path to generate logits and weak augmentation on the middle path to generate pseudo-labels, the results are

TABLE 7: Effectiveness of label selection strategy. Method: ProPos + ClusMatch. Metric: ACC.

Method	GCC + ClusMatch		ProPos + ClusMatch	
	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
Random	0.939	0.579	0.870	0.521
Ours	0.949	0.592	0.914	0.536

TABLE 8: Comparison between ClusMatch and SCAN. Metric: ACC.

Dataset	CIFAR-10	CIFAR-100
GCC	0.870	0.494
GCC+SCAN	0.901	0.523
GCC+ClusMatch	0.914	0.536

the best, which is in line with that in semi-supervised learning [44].

Label Selection Strategy. In the label selection phase, we design two simple but effective strategies for clustering head based and feature learning based methods, respectively, to select high-quality samples for supervised learning. In order to show the superiority of our strategy, we compare it with the scheme that randomly selects samples. As presented in Table 7, it is obvious that the accuracy decreases in varying degrees when the samples are randomly selected. In particular, ClusMatch based on ProPos drops significantly by 4.4% on ACC for CIFAR-10.

Comparison with Self-labeling Strategy of SCAN. SCAN is also a post-processing method to improve the clustering performance, and we also compared our ClusMatch with it. Note that the self-labeling phase of SCAN relies on a head-based clustering method. Since ProPos is a feature-based method, we mainly compared GCC+ClusMatch with GCC+SCAN. As shown in Table 8, ClusMatch improves by 1.3% over SCAN’s postprocessing strategy on both CIFAR-10 and CIFAR-100.

4.5.2 Parameter Sensitivity Analysis

Label Selection Ratio σ . σ is a key parameter in ClusMatch that determines the number of samples for supervised learning. From Table 9, we can see that for CIFAR-10, the best performance is achieved when σ is set to 0.01. For other datasets except ImageNet-1K, the results are relatively stable when

TABLE 9: Effect of label selection ratio. Method: ProPos + ClusMatch.

σ	CIFAR-10			CIFAR-100			ImageNet-Dogs			Tiny-ImageNet		
	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
0.01	0.890	0.949	0.894	0.592	0.603	0.441	0.690	0.729	0.614	0.474	0.319	0.190
0.1	0.866	0.928	0.856	0.601	0.592	0.426	0.745	0.798	0.685	0.482	0.408	0.232
0.15	0.875	0.936	0.871	0.596	0.594	0.436	0.759	0.793	0.696	0.480	0.421	0.241

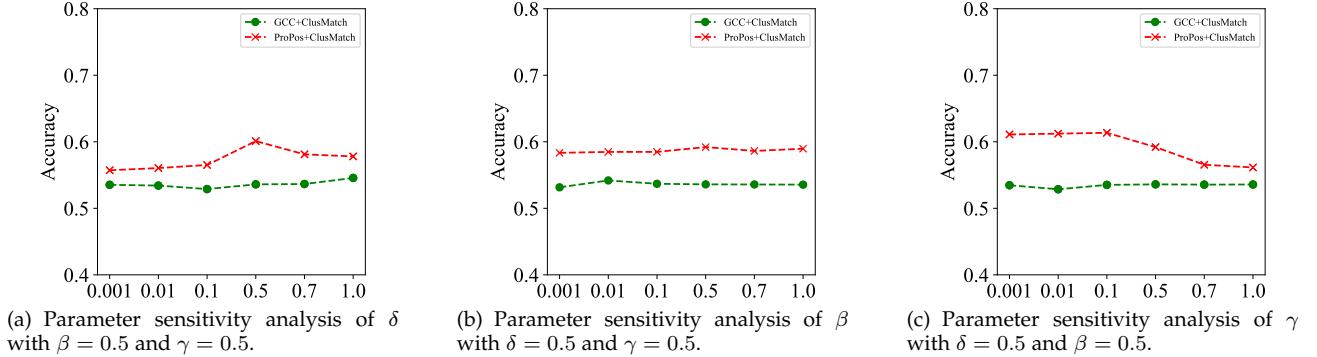


Fig. 5: Effects of loss weights on CIFAR-100. Metric: ACC.

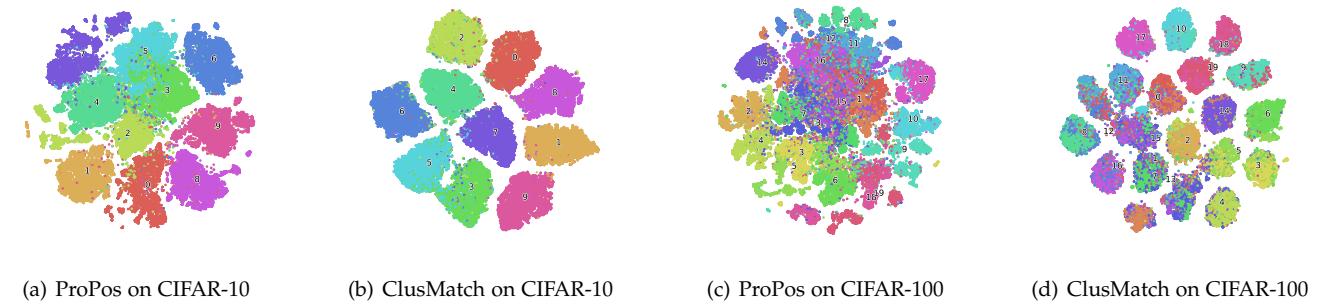


Fig. 6: Visualization of feature representations with t-SNE.

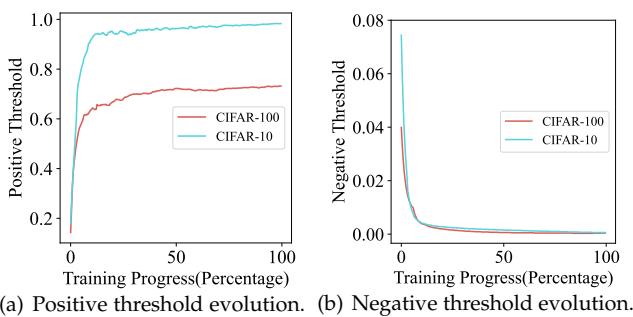


Fig. 7: Evolution of positive and negative thresholds on CIFAR-10 and CIFAR-100. Method: ProPos + ClusMatch. Metric: ACC.

σ varies in $[0.1, 0.15]$. To simplify the parameter settings, we uniformly take $\sigma = 0.1$. Considering the large scale of ImageNet-1K, we performed a separate ablation experiment for σ . As shown in Fig. 4 (a), when only a small number of samples are selected, i.e., $\sigma = 0.01$, ClusMatch suffers a large performance degradation for the lack of highly-

confident supervised signals, resulting in underfitting of the model. As σ varies in the range of $[0.1, 0.15, 0.2, 0.4, 0.5]$, the performance improves sharply and reaches its best when σ is set to 0.2. We also explored the relationship between clustering performance, the number and the quality of selected samples. As shown in Fig. 4 (b), the ACC of ClusMatch does not strictly monotonically increase as more and more samples are selected; it is also affected by the accuracy of the selected samples, i.e., the quality of the samples. When σ varies between $[0.01, 0.1, 0.15, 0.2]$, the number of selected samples has a dominant effect on the results. The more samples are selected, the more high-quality supervised signals and, consequently, the higher the accuracy. However, when σ is greater than 0.2, the quality of the samples plays a decisive role. Therefore, as the accuracy of the selected samples drops sharply, the ACC begins to decrease.

Loss Weights. We also performed sensitivity analysis on the loss weights, δ , β , and γ . As shown in Fig. 5, we can observe that even with varying weights, the accuracy shows great stability, which demonstrates our method is insensitive to the proposed K-means, positive, and negative pseudo-label learning losses, further proving the robustness

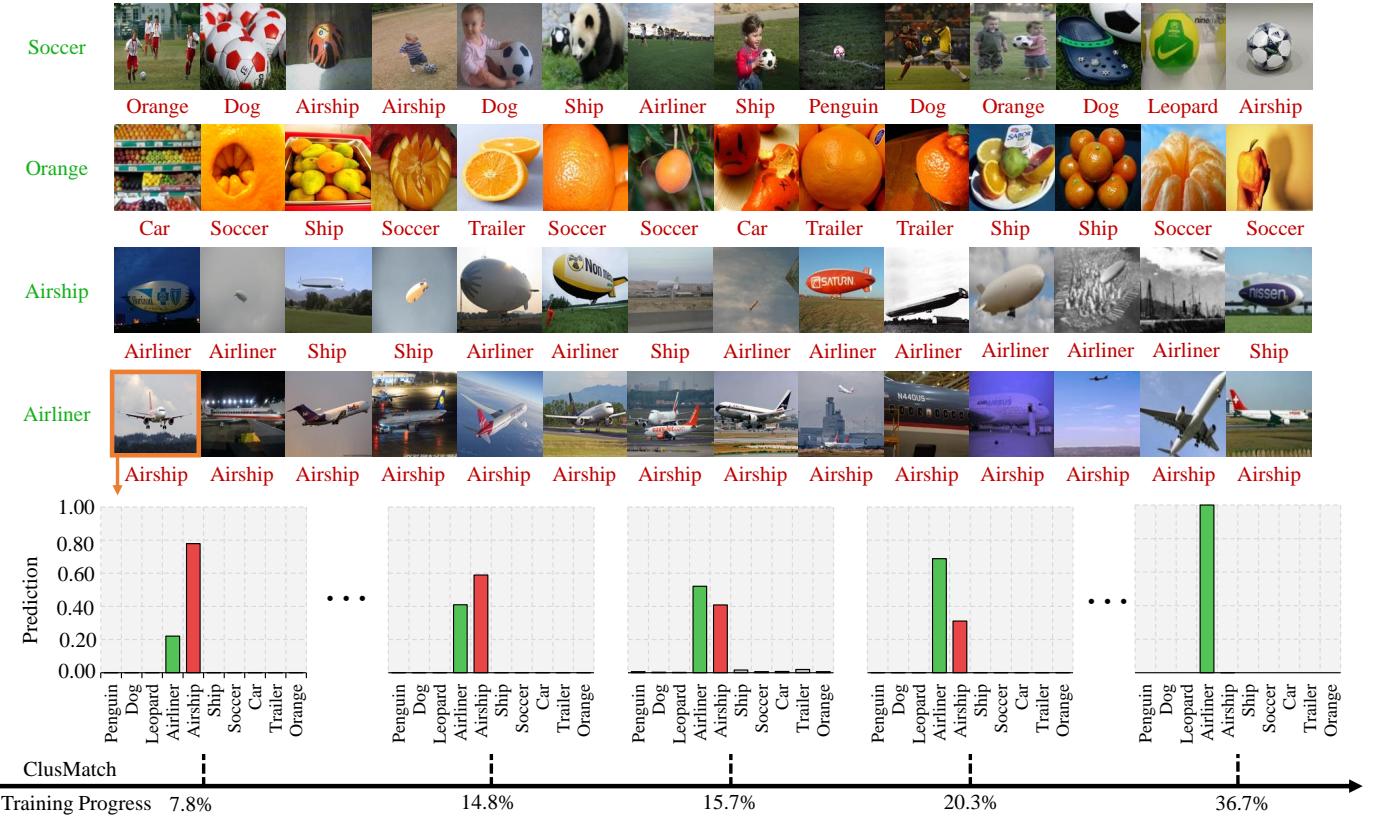


Fig. 8: Case study on Tiny-ImageNet. We present the clustering results of four classes, with the green text representing the right results of ClusMatch and the red text for the wrong results of ProPos. An example (boxed in orange) of error correction is also shown to visualize the effectiveness of ClusMatch.

of our ClusMatch. To unify the parameters and simplify the settings, we set all loss weights to 0.5.

4.6 Qualitative Study

We also visualized the features of ClusMatch by t-SNE [69] to demonstrate the property of discriminability. The evolution of positive and negative thresholds is also presented for better illustration of adaptive threshold. Finally, a case study is conducted to illustrate the error-correction capability of our ClusMatch.

4.6.1 Visualization of Features

To verify that ClusMatch can help the network learn more discriminative representations, we visualized the features from encoder $f(\cdot)$ by t-SNE [69]. The visualization results of ClusMatch and ProPos on CIFAR-10 and CIFAR-100 are presented in Fig. 6. It is apparent that, for ClusMatch, samples within the same category are more tightly grouped, while those from different categories are distinctly separated, demonstrating a more clear demarcation. For instance, on CIFAR-10, the samples of category 2 exhibit a scattered distribution in the feature space of ProPos, whereas in ClusMatch, the corresponding samples are clustered together, which can also be observed for samples of categories 9, 18, and 19 on CIFAR-100.

4.6.2 Evolution of Positive and Negative Thresholds

As shown in Fig. 7, we also presented the evolution of positive and negative thresholds for the ProPos+ClusMatch

method on CIFAR-10 and CIFAR-100. As these thresholds differ from different classes, we showed the mean values across all categories to give a depiction of the overall changes. We can observe that the positive threshold increases while the negative threshold decreases, which aligns with expectations.

4.6.3 Case Study

In order to visually demonstrate the benefits that our approach brings to existing clustering methods, we also conduct a case study. From Fig. 8, we can see that ClusMatch exhibits stronger resistance to interference compared to ProPos, mitigating confusion between target items and foreground or surrounding items. For example, in the sixth image of the first row, ClusMatch accurately identifies the soccer without being influenced by the foreground panda. Similar occurrences can be observed in the twelfth image of the first row and the first image of the second row.

Additionally, ProPos tends to confuse certain classes, such as airliners and airships, whereas ClusMatch can rectify such errors through fine-tuning. Taking the image boxed in orange as an example, we show how its prediction to different classes changes as training progresses. It is apparent that, despite it wrongly assigning this image to the airship class initially, ClusMatch is able to gradually correct it to the true airliner class.

5 CONCLUSION AND FUTURE WORK

In this paper, we introduce ClusMatch, a deep clustering method based on the semi-supervised framework. ClusMatch is pluggable for existing deep clustering methods and can further improve their performance. To satisfy the semi-supervised learning setting, we first propose two selection strategies to choose a few high-quality samples for supervised learning. For the unselected samples, we come up with novel unified positive and negative pseudo-label learning to obtain richer supervised signals. In order to further improve the confidence of generated pseudo-labels, an adaptive positive and negative threshold strategy is designed, which can dynamically filter out low-confidence pseudo-labels according to the learning status of the model. Finally, we validate the effectiveness and superiority of ClusMatch on six widely-used datasets and one large-scale dataset. In future work, we plan to extend ClusMatch to unbalanced datasets, particularly long-tailed datasets, characterized by a large number of samples in minority classes (head classes) and few samples in majority classes (tail classes).

REFERENCES

- [1] J. Neyman and E. L. SCOTT, "Berkeley symposium on mathematical statistics and probability," in *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability*, 1967.
- [2] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.
- [3] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 14, 2001.
- [4] K. Kailing, H.-P. Kriegel, and P. Kröger, "Density-connected subspace clustering for high-dimensional data," in *Proceedings of the International Conference on Data Mining*, 2004, pp. 246–256.
- [5] G. Liu, Z. Zhang, Q. Liu, and H. Xiong, "Robust subspace clustering with compressed data," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 5161–5170, 2019.
- [6] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1582–1590.
- [7] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, and D. Xu, "Generalized latent multi-view subspace clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 1, pp. 86–99, 2018.
- [8] X. Xie, J. Wu, G. Liu, and Z. Lin, "Sscnet: learning-based subspace clustering," *Visual Intelligence*, vol. 2, no. 1, p. 11, 2024.
- [9] Z. Zhang, Y. Zhang, S. Li, G. Liu, D. Zeng, S. Yan, and M. Wang, "Flexible auto-weighted local-coordinate concept factorization: A robust framework for unsupervised clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1523–1539, 2019.
- [10] D. Cai, X. He, X. Wang, H. Bao, and J. Han, "Locality preserving nonnegative matrix factorization," in *Proceedings of International Joint Conference on Artificial Intelligence*, 2009, pp. 1010–1015.
- [11] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5879–5887.
- [12] J. Chang, G. Meng, L. Wang, S. Xiang, and C. Pan, "Deep self-evolution clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 809–823, 2018.
- [13] J. Wu, K. Long, F. Wang, C. Qian, C. Li, Z. Lin, and H. Zha, "Deep comprehensive correlation mining for image clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8150–8159.
- [14] H. Zhong, C. Chen, Z. Jin, and X.-S. Hua, "Deep robust clustering by contrastive learning," *arXiv preprint arXiv:2008.03030*, 2020.
- [15] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, "Scan: Learning to classify images without labels," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 268–285.
- [16] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 8547–8555.
- [17] H. Zhong, J. Wu, C. Chen, J. Huang, M. Deng, L. Nie, Z. Lin, and X.-S. Hua, "Graph contrastive clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 9224–9233.
- [18] Y. Shen, Z. Shen, M. Wang, J. Qin, P. Torr, and L. Shao, "You never cluster alone," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 27734–27746.
- [19] Z. Huang, J. Chen, J. Zhang, and H. Shan, "Learning representation for clustering via prototype scattering and positive sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [20] Q. Qian, "Stable cluster discrimination for deep clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2023, pp. 16645–16654.
- [21] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, "Prototypical contrastive learning of unsupervised representations," in *Proceedings of the International Conference on Learning Representations*, 2021.
- [22] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5147–5156.
- [23] Y. Li, M. Yang, D. Peng, T. Li, J. Huang, and X. Peng, "Twin contrastive learning for online clustering," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2205–2221, 2022.
- [24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the International Conference on Machine Learning*, 2020, pp. 1597–1607.
- [25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.
- [26] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15750–15758.
- [27] 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," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 21271–21284.
- [28] C. Niu, H. Shan, and G. Wang, "Spice: Semantic pseudo-labeling for image clustering," *IEEE Transactions on Image Processing*, vol. 31, pp. 7264–7278, 2022.
- [29] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 18408–18419.
- [30] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [31] Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj *et al.*, "Freematch: Self-adaptive thresholding for semi-supervised learning," in *Proceedings of the International Conference on Learning Representations*, 2022.
- [32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- [33] S. Wang, Z. Chen, S. Du, and Z. Lin, "Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5042–5055, 2021.
- [34] E. Pan and Z. Kang, "Multi-view contrastive graph clustering," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 2148–2159.
- [35] S. Huang, Z. Kang, Z. Xu, and Q. Liu, "Robust deep k-means: An effective and simple method for data clustering," *Pattern Recognition*, vol. 117, p. 107996, 2021.
- [36] C. Zhang, Y. Cui, Z. Han, J. T. Zhou, H. Fu, and Q. Hu, "Deep partial multi-view learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2402–2415, 2020.

- [37] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 132–149.
- [38] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proceedings of the International Conference on Machine Learning*, 2016, pp. 478–487.
- [39] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.
- [40] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proceedings of the International Conference on Machine Learning Workshop*, vol. 3, no. 2, 2013, p. 896.
- [41] J. Wu, H. Yang, T. Gan, N. Ding, F. Jiang, and L. Nie, "Chmatch: contrastive hierarchical matching and robust adaptive threshold boosted semi-supervised learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15762–15772.
- [42] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 17, 2004.
- [43] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [44] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 596–608.
- [45] H. Chen, R. Tao, Y. Fan, Y. Wang, J. Wang, B. Schiele, X. Xie, B. Raj, and M. Savvides, "Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning," in *Proceedings of the International Conference on Learning Representations*, 2023.
- [46] T. Ishida, G. Niu, W. Hu, and M. Sugiyama, "Learning from complementary labels," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [47] X. Yu, T. Liu, M. Gong, and D. Tao, "Learning with biased complementary labels," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 68–83.
- [48] Y. Kim, J. Yim, J. Yun, and J. Kim, "Nlnl: Negative learning for noisy labels," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 101–110.
- [49] J. Chen, V. Shah, and A. Kyriolidis, "Negative sampling in semi-supervised learning," in *Proceedings of the International Conference on Machine Learning*, 2020, pp. 1704–1714.
- [50] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah, "In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning," in *Proceedings of the International Conference on Machine Learning*, 2021.
- [51] Y. Chen, X. Tan, B. Zhao, Z. Chen, R. Song, J. Liang, and X. Lu, "Boosting semi-supervised learning by exploiting all unlabeled data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7548–7557.
- [52] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 215–223.
- [53] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.
- [54] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 17, 2004.
- [55] K. C. Gowda and G. Krishna, "Agglomerative clustering using the concept of mutual nearest neighbourhood," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 10, no. 2, 1978, pp. 105–112.
- [56] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 19, 2006.
- [57] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. 12, 2010.
- [58] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proceedings of the International Conference on Learning Representations*, 2016.
- [59] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2528–2535.
- [60] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of the International Conference on Learning Representations*, 2014.
- [61] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9865–9874.
- [62] J. Huang, S. Gong, and X. Zhu, "Deep semantic clustering by partition confidence maximisation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8849–8858.
- [63] Z. Dang, C. Deng, X. Yang, K. Wei, and H. Huang, "Nearest neighbor matching for deep clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13693–13702.
- [64] T. W. Tsai, C. Li, and J. Zhu, "Mice: Mixture of contrastive experts for unsupervised image clustering," in *Proceedings of the International Conference on Learning Representations*, 2020.
- [65] Y. Tao, K. Takagi, and K. Nakata, "Clustering-friendly representation learning via instance discrimination and feature decorrelation," in *Proceedings of the International Conference on Learning Representations*, 2021.
- [66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [67] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6256–6268.
- [68] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 702–703.
- [69] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.



Jianlong Wu (Member, IEEE) received the B.Eng. degree from the Huazhong University of Science and Technology, China, in 2014, and the Ph.D. degree from Peking University, China, in 2019. He is currently a Professor with the Harbin Institute of Technology, Shenzhen, China. He was an Assistant Professor at the Shandong University from 2019 to 2022. He has published more than 40 papers in top journals and conferences, such as IEEE TPAMI, TIP, ICML, NeurIPS, and ICCV. His research interests include computer vision and machine learning. He received many awards, such as the Outstanding Reviewer of ICML 2020 and the Best Student Paper of SIGIR 2021. He serves as the Area Chair for ICML, NeurIPS, CVPR, and ACM MM.



Zihan Li received the bachelor's degree in computer science from Shandong University in 2022. He is currently a master student in the School of Computer Science and Technology, Shandong University. His research interests include computer vision and machine learning.



Wei Sun received the bachelor's degree in computer science from Shandong University in 2021. He is currently a master student in the School of Computer Science and Technology, Shandong University. His research is primarily focused on action recognition and semi-supervised learning.



Jianhua Yin (Member, IEEE) received the Ph.D. degree in computer science and technology from Tsinghua University, China, in 2017. He is currently an Associate Professor with the School of Computer Science and Technology, Shandong University, China. He has published several papers in the top venues, such as ACM Transactions on Information Systems, IEEE Transactions on Multimedia, ACM MM, ACM SIGKDD, and ACM SIGIR. His research interests include data mining and machine learning applications.



Liqiang Nie (Senior Member, IEEE) received the B.Eng. degree from Xi'an Jiaotong University and the Ph.D. degree from the National University of Singapore (NUS). After the Ph.D. degree, he continued his research with NUS, as a Research Fellow for three years. He is currently a Professor with the Harbin Institute of Technology (Shenzhen). He has coauthored more than 200 articles and four books. He received more than 28,000 Google Scholar citations as of September 2024. His research interests include multimedia computing and information retrieval. He received many awards, such as ACM MM and SIGIR Best Paper Honorable Mention in 2019, the SIGMM Rising Star in 2020, the TR35 China 2020, the DAMO Academy Young Fellow in 2020, the SIGIR Best Student Paper in 2021, and ACM MM Best Paper in 2022. He is the Area Chair of ACM MM from 2018 to 2023. He is an Associate Editor of IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Multimedia, ACM Transactions on Multimedia Computing, Communications, and Applications, and Information Sciences.



Zhouchen Lin (Fellow, IEEE) received the PhD degree in applied mathematics from Peking University, in 2000. He is currently a professor with the Key Laboratory of Machine Perception (MOE), School of Artificial Intelligence, Peking University. His research interests include machine learning and numerical optimization. He was area chairs of ACML, ACCV, CVPR, ICCV, NIPS/NeurIPS, AAAI, IJCAI, ICLR, and ICML many times, a program co-chair of ICPR 2022, and senior area chairs of ICML 2022 and NeurIPS 2022. He was an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and currently is an associate editor of the International Journal of Computer Vision. He is a fellow of the IAPR, and the CSIG.