A3S: A General Active Clustering Method with Pairwise Constraints

Xun Deng [1 2] **Junlong Liu** [2] **Han Zhong** [3] **Fuli Feng** [1] **Chen Shen** [2] **Xiangnan He** [1] **Jieping Ye** [2] **Zheng Wang** [2]

Abstract

Active clustering aims to boost the clustering performance by integrating human-annotated pairwise constraints through strategic querying. Conventional approaches with semi-supervised clustering schemes encounter high query costs when applied to large datasets with numerous classes. To address these limitations, we propose a novel Adaptive Active Aggregation and Splitting (A3S) framework, falling within the cluster-adjustment scheme in active clustering. A3S features strategic active clustering adjustment on the initial cluster result, which is obtained by an adaptive clustering algorithm. In particular, our cluster adjustment is inspired by the quantitative analysis of Normalized mutual information gain under the information theory framework and can provably improve the clustering quality. The proposed A3S framework significantly elevates the performance and scalability of active clustering. In extensive experiments across diverse real-world datasets, A3S achieves desired results with significantly fewer human queries compared with existing methods.

1. Introduction

In the realm of data science, clustering algorithms have emerged as a cornerstone technology within the domain of unsupervised learning (Khanum et al., 2015; Celebi & Aydin, 2016). By automatically grouping similar data objects based on inherent structures and patterns within datasets, clustering provides an efficacious means to condense and structure complex information and is widely applied in image classification (Caron et al., 2020), social network analysis (He et al., 2022), etc. However, conventional clustering techniques often rely on static parameter settings and one-off computations, rendering them less adaptable to strange or expanding data environments (See Section 2.3.3). This context highlights the growing importance of human-computer collaborative active clustering approaches.

Active clustering refers to a paradigm that actively selects side information (Anand et al., 2014), in the form of pairwise constraints, to maximally improve the clustering performance. Extensive work (González-Almagro et al., 2023) has explored the combination of the strategic selection of pairwise constraints and semi-supervised clustering (SSC) (Basu et al., 2002), and attains much lower query complexity compared to its semi-supervised counterpart (Bilenko et al., 2004). Despite the practicality, current active clustering methods often suffer from high computational and query costs when the number of classes is large.

SSC-based active clustering methods primarily assess the uncertainty of all pairwise constraints, and iteratively choose the most uncertain ones for expert queries. This manner, while systematic, faces notable challenges: it potentially relies on the assumption that a small set of initial pairwise constraints will rapidly cover most real classes. This assumption becomes increasingly unreliable in scenarios with large sample numbers N and class numbers K. Therefore, some active clustering methods (Van Craenendonck et al., 2017; Shi et al., 2020) shift from SSC to a cluster-adjustment scheme. This scheme involves over-clustering data into k clusters via a specific clustering method (where k is greater than K), and subsequently aggregates the resulting small clusters into larger ones based on pairwise constraints. However, it requires a proper cluster number k as an input parameter, which is hard to determine in real applications. Moreover, the human query may be misleading when the selected samples are outliers, as they do not represent the majority sample of a cluster.

This work aims to overcome the drawbacks of existing cluster adjustment schemes. We first present a theoretical result that identifies conditions where aggregating two clusters does not reduce the normalized mutual information (NMI) between the resulting clustering and the real clustering. Here, NMI measures the overlap between two clustering results, with larger values indicating better performance (see Definition 2.3). In addition, to guide active human queries, we quantify the impact of merging

[1]University of Science and Technology of China [2]Alibaba Group [3]Peking University. Correspondence to: Zheng Wang <wz388779@alibaba-inc.com>, Fuli Feng <fulifeng93@gmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

two clusters on the expected NMI value difference from the theoretical side. This characterization enables us to actively select cluster pairs that maximize the NMI gain.

Building upon these theoretical results, we propose Adaptive Active Aggregation and Splitting (A3S), a generic cluster adjustment framework for active clustering. A3S operates in two stages: the *adaptive initialization stage* and the *active aggregation and splitting stage*. The first stage autonomously identifies an appropriate cluster number and produces initial clustering results using a designated clustering algorithm (e.g., K-means and hierarchical clustering). In the latter stage, A3S proactively identifies the cluster pair that is expected to enhance the NMI value mostly. Afterward, it assesses whether the majority of samples in a cluster are of the same class (i.e., cluster purity in Definition 2.4), then merges pure clusters queried to be in the same class by oracles, and divides impure clusters into pure subclusters and outliers. This stage will be repeated a few times to ensure convergence, where no more aggregation and splitting will happen.

Our contributions are summarized as follows:

- We introduce A3S, a general active clustering algorithm that improves the quality of clustering by optimizing the NMI value. A3S strategically selects cluster pairs for aggregation, aiming to maximize the expected NMI improvement within a limited query cost. The quantification of NMI gain is guided by our information-theoretical analysis (Theorem 2.4). Additionally, A3S implements precise splitting on clusters that fail purity tests, ensuring the correction of outlier samples.

- Regarding implementation superiority, A3S can reveal the underlying ground truth clustering structure with substantially fewer human queries compared to traditional methods, especially when prior dataset knowledge is unknown. In addition, A3S locally adjusts the cluster labels, offering a more efficient alternative to rerunning semi-supervised clustering algorithms, and remains feasible for large datasets.

- In terms of practical performance, by conducting comprehensive evaluations on various real-world datasets, we demonstrate that A3S consistently surpasses all baseline models. Notably, baseline methods typically require more than 4000 queries to achieve high-quality clustering results for datasets containing thousands of samples. In contrast, A3S achieves this performance with only a few hundred queries.

2. Methodology

2.1. Notation and Definition

Definition 2.1 (Active Clustering)**.** We denote the true classes of N samples $X = \{x_1, \cdots, x_N\}$ by $Y = \{y_1, \cdots, y_N\}$, where $y_i \in \{1, \cdots, K\}$ and K is the number of classes. Let the ground truth clustering of X as $C = \{c_1, \cdots, c_K\}$, and the initial clustering result as $\Omega = \{w_1, \cdots, w_k\}$, where $X = \cup_{i=1}^{K} c_i = \cup_{i=1}^{k} w_i$ and $c_i \cap c_j = \emptyset, w_i \cap w_j = \emptyset, \forall i, j$. Active Clustering strategically selects sample pairs (x_i, x_j), and requires the oracle to judge if $y_i = y_j$ (two samples are must-linked) or $y_i \neq y_j$ (two samples are cannot-linked). It then updates the clustering result Ω with the queried pairwise constraints accordingly. Active Clustering aims to utilize the queried constraints to maximally reduce the difference between C and Ω, which is measured by the normalized mutual information (NMI) (Kvalseth, 1987; Vinh et al., 2009).

Definition 2.2 (Cluster Adjustment Scheme)**.** We define a cluster adjustment scheme as a label update strategy employed by active clustering algorithms. Specifically, it entails the algorithm's process of either locally aggregating small clusters into larger ones or splitting impure clusters into several subclusters, guided by pairwise constraints.

Definition 2.3 (NMI)**.** NMI is a measure of how much common information two clustering results share. Given N samples and their two clusterings $\Omega = \{w_1, \cdots, w_k\}$ and $\Omega' = \{w'_1, \cdots, w'_k\}$, we define the NMI value as

$$n = \frac{2\mathbb{I}(\Omega; \Omega')}{\mathbb{H}(\Omega) + \mathbb{H}(\Omega')} = \frac{2\mathbb{I}(\zeta; \zeta')}{\mathbb{H}(\zeta) + \mathbb{H}(\zeta')},$$

where $\zeta = (|w_1|/N, \cdots, |w_k|/N)$ and $\zeta' = (|w'_1|/N, \cdots, |w'_{k'}|/N)$ are two distributions induced by Ω and Ω', respectively. Here, $\mathbb{I}(\cdot;\cdot)$ denotes mutual information, and $\mathbb{H}(\cdot)$ denotes entropy.

We also introduce the notion of clustering purity (González-Almagro et al., 2023).

Definition 2.4 (Purity)**.** We define the *dominant class* of a cluster w_i as $\arg\max_j |w_i \cap c_j|$. We label the sample in a cluster that does not belong to its dominant class or the sample that is a single cluster itself as an outlier. Then, the purity of w_i is $\max_j \frac{|w_i \cap c_j|}{|w_i|}$, and the purity of the clustering result Ω is quantified by $\frac{\sum_i \max_j |w_i \cap c_j|}{N}$.

2.2. Theoretical Analysis

Active clustering that adopts the cluster-adjustment scheme does not rely on semi-supervised clustering for updating clustering results. Instead, it emphasizes the use of must-link and cannot-link constraints as indicators to merge or split clusters (Van Craenendonck et al., 2018). This greedy strategy may, however, jeopardize the quality of clustering outcomes for several reasons. For example, the queried samples could be outliers, or the purity of a cluster might not be sufficiently high. In light of this, we introduce a pivotal theorem that offers clear guidance for aggregation actions. This is achieved through an evaluation of the NMI.

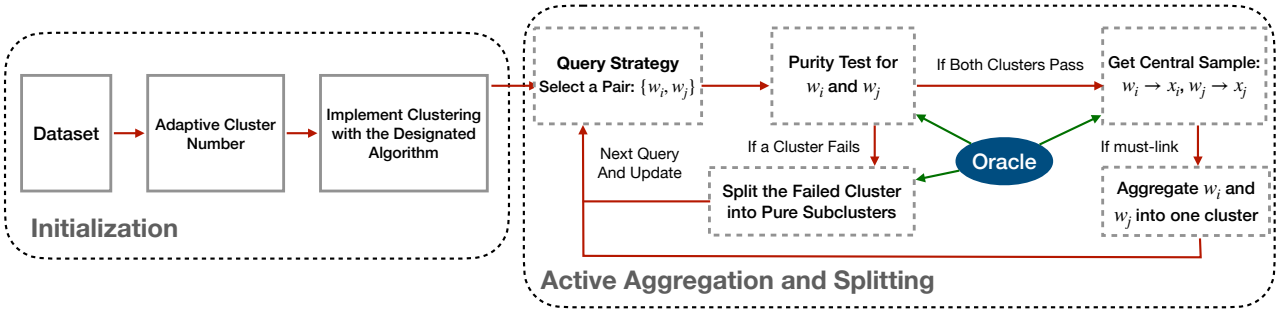

Figure 1. The workflow of A3S which consists of the adaptive clustering stage and active aggregation and splitting stage.

Theorem 2.5 (Guarantee for Cluster Aggregation)**.** *Denote the clustering of N samples as Ω, the ground truth clustering as C, and the NMI value of Ω with respect to C as n_1. For any two clusters in Ω, say w_1 and w_2, suppose they have a common dominant class c_1 with purities t_1 and t_2 respectively, where $t_1, t_2 \in [p, 1]$. By aggregating w_1 and w_2 into a new cluster $w_{1,2} = w_1 \cup w_2$, we arrive at a new clustering $\Omega^\star = \{w_{1,2}, w_3, \cdots, w_k\}$ with NMI value of n_2. This aggregation positively impacts clustering performance (i.e., $n_2 \geq n_1$) if $p \geq 0.7$ and $n_1 \geq 2 \cdot (1.0586 - \min\{t_1, t_2\})$.*

Theorem 2.5 delineates the conditions for non-deteriorating cluster aggregation, and the detailed proof is in Appendix A.1. Specifically, merging two clusters can achieve provable benefits when their purity is at least 0.7 and preceding NMI exceeds $2 \cdot (1.0586 - \min\{t_1, t_2\})$. This finding is important as it suggests that the purity requirement for cluster aggregation is relatively mild, allowing for the inclusion of a small number of outliers within each cluster without compromising the overall clustering performance.

We take a step further by formulating the expected gain in NMI value when we decide to query a sample pair from a cluster pair. This involves aggregating clusters if the query result is "must-link" and keeping them separated if the result is "cannot-link".

Definition 2.6 (Expected NMI Gain)**.** Suppose the dominant class of clusters w_i and w_j is c_m and c_n respectively, which remain unknown before querying. Let n_1 and n_2 represent the NMI values of the clustering result before and after aggregating the two clusters. We denote $\mathbb{P}(c_m = c_n)$ as the probability that the oracle observes a 'must-link' result. Then, we define the expected NMI gain from this query as follows:

$$\mathbb{E}[\Delta\texttt{NMI} \mid w_i, w_j] = \mathbb{P}(c_m = c_n) \cdot (n_2 - n_1), \quad (1)$$

In what follows, we present how to estimate $\mathbb{P}(c_m = c_n)$ and $(n_2 - n_1)$.

We use $e_{st} = 1/0$ to signify if y_s equals y_t or not, and denote the posterior pairwise probability as $\mathbb{P}(e_{st} = 1)$. We estimate the pairwise probability following previous probability clustering methods (e.g., Liu et al., 2022), and discuss the estimation details in Appendix B.1. Then we can express the aggregation probability $\mathbb{P}(c_m = c_n)$ as follows:

$$\mathbb{P}(c_m = c_n) = \frac{\prod_{s \in w_i, t \in w_j} \mathbb{P}(e_{st} = 1)}{\prod_{s \in w_i, t \in w_j} \mathbb{P}(e_{st} = 1) + \prod_{s \in w_i, t \in w_j} \mathbb{P}(e_{st} = 0)}, \quad (2)$$

The detailed derivation of Eq. (2) is in Appendix A.2.

Moving forward, we focus on formulating $n_2 - n_1$. In line with the notations used in Theorem 2.5, we define $\Delta h = \mathbb{H}(\Omega) - \mathbb{H}(\Omega^*)$ and proceed with the following result:

$$n_2 = \frac{2\mathbb{I}(\Omega^\star; C)}{\mathbb{H}(\Omega^\star) + \mathbb{H}(C)} \approx \frac{2\mathbb{I}(\Omega; C)}{\mathbb{H}(\Omega) + \mathbb{H}(C) - \Delta h},$$

where we use the fact that $\mathbb{I}(\Omega^*; C) \approx \mathbb{I}(\Omega, C)$ when the purity of w_i and w_j is sufficiently large. Moreover, when the sizes of clusters w_i and w_j are significantly smaller than the sample size N, the direct calculation gives that $\Delta h \ll \mathbb{H}(\Omega) + \mathbb{H}(C)$ (refer to Appendix A.3 for verification). Hence, we have

$$n_2 - n_1 \approx \frac{2\mathbb{I}(\Omega; C)\Delta h}{(\mathbb{H}(\Omega) + \mathbb{H}(C))^2} \propto \Delta h, \quad (3)$$

Combing Eq. (1), Eq. (2), and Eq. (3), we obtain our estimators of $\mathbb{E}[\Delta\texttt{NMI} \mid w_i, w_j]$ as follows:

$$\mathbb{E}[\Delta\texttt{NMI} \mid w_i, w_j] \propto \text{RHS of Eq. (2)} \cdot \Delta h. \quad (4)$$

where RHS denotes the right-hand side.

2.3. Adaptive Active Aggregation and Splitting

Building on the analysis of cluster aggregation and expected query impact, we detail our Adaptive Active Aggregation and Splitting framework, which comprises two distinct stages. First, the Adaptive Clustering stage is introduced in Section 2.3.1, where we describe the generation of initial clustering results. Second, in Section 2.3.2, we discuss selecting pairwise constraints and updating clustering during the Active Aggregation and Splitting stage. The A3S workflow is illustrated in Figure 1, and the steps are summarized in Algorithm 1.

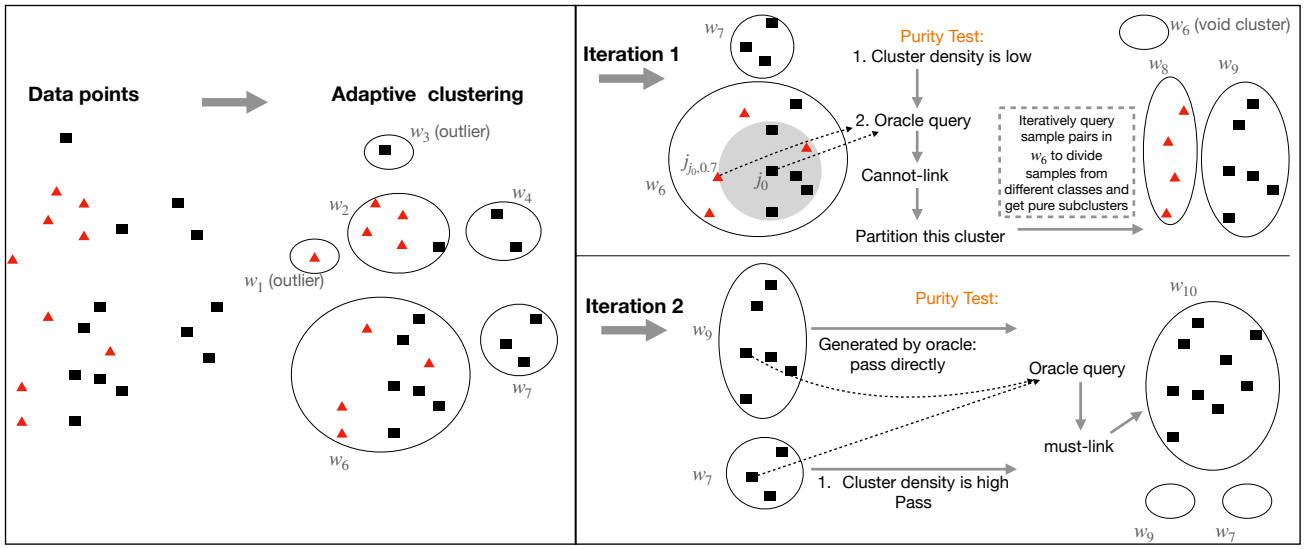

Figure 2. Case study for A3S. In iteration 1, the cluster w_6 does not pass the purity test and the oracle invests 12 queries to split it into two pure subclusters. In iteration 2, the query result for the two central samples is must-link and they are merged into one cluster.

2.3.1. Initialization via Adaptive Clustering

Current active clustering methods often necessitate manually setting initial cluster numbers, a challenging task when the number of classes is unknown. In response, we propose using adaptive clustering methods to determine an appropriate cluster number for initialization. An adaptive clustering method organizes data based on local density, revealing the dataset's inherent structure. It handles noise by isolating each noisy sample into a separate cluster (i.e., outlier), avoiding unsuitable data grouping. This approach ensures a more natural and purer clustering outcome. Classic adaptive clustering methods include Probabilistic Clustering (Liu et al., 2022), density-based clustering (Zhang et al., 2021; Khan et al., 2018), among others. Additionally, the quality of adaptive clustering can be significantly enhanced in multi-view clustering scenarios (Liu et al., 2023). In this process, we aim to obtain a suitable reference for the number of clusters (typically larger than real class numbers), rather than the optimal clustering, hence do not require a precise hyper-parameter search. Once the adaptive cluster number is established, we can employ the desired clustering algorithm to produce the initial clustering result.

2.3.2. Active Aggregation and Splitting

Query Strategy. To ensure a high success rate in establishing 'must-link' connections among selected cluster pairs, We employ a two-step query strategy. The first step involves filtering out low-quality cluster pairs, focusing on those with aggregation probabilities at the top. In the second step, we utilize Eq. (4) to calculate the expected NMI gain for these cluster pairs. Then we choose the pair with the highest NMI gain. By Theorem 2.5, it is necessary to evaluate the purity of the two chosen clusters and select one representative sample (i.e., it belongs to the dominant class of this cluster) from each cluster to form a sample pair. This pair will then be subjected to queries by oracles. To facilitate this process, we specially designed a purity test.

Purity Test. For convenience, we employ the sphere structure to depict a cluster, where the centroid sample of a cluster is denoted as j_0. The other samples in the cluster are marked as $j_{i,\rho}$, indicating that a sphere centered on sample i with a radius of $d(i, j_{i,\rho})$ includes ρ percent of the samples in the cluster. Considering that outlier samples typically reside in the outer regions of a cluster, and samples within impure clusters tend to be more sparsely distributed, we bifurcate the task of purity testing into two consecutive judgments.

First, we evaluate how densely the samples are concentrated within a cluster. This assessment is formalized as the density test for a cluster w, expressed in Eq. (5), where τ represents a pre-set threshold that determines the level of strictness in this density test, and $\mathbf{1}(\cdot)$ denotes the indicator function.

$$\mathcal{DT}(w) = \mathbf{1}\Big(\frac{\sum_{i\in w, j\in w(i)} \mathbb{P}(e_{ij}=1)}{\sum_{i\in w} |w(i)|} > \tau\Big), \quad (5)$$
$$w(i) = \{j \mid j \in w, \mathbb{P}(e_{ij}=1) < \mathbb{P}(e_{ij_{i,0.5}}=1)\}$$

If a cluster fails in the density test, we select a sample pair as $(j_0, j_{j_0,0.7})$, and require oracles to judge $\mathcal{P}(w) = \mathbf{1}(y_{j_0} = y_{j_{j_0,0.7}})$. It estimates whether the purity of this cluster is higher than 0.7 (i.e., satisfying the conditions in Theorem 2.5). Overall, the purity test is as follows:

$$\mathcal{PT}(w) = \mathcal{DT}(w) \quad \text{if} \quad \mathcal{DT}(w) \quad \text{else} \quad \mathcal{P}(w). \quad (6)$$

Clustering Update. In response to different outcomes in the purity test, we adopt the following strategies to update

the clustering: (i) The purity test yields a result of 1 for both clusters, and we require the oracle to query their central samples. Should the query yield a must-link result, we will merge the clusters; if not, we will retain them as separate clusters. (2) The purity test yields 0 for at least one cluster, indicating a need for refinement, we proceed to split it into multiple subclusters with enhanced purity. Given the proven effectiveness of sample-based active clustering in managing small-scale datasets, as noted by Basu et al. (2004), we apply this approach for the splitting task, and the detail is described in Algorithm 2.

Transitive Inference. The must-link and cannot-link constraints possess the transitivity property (e.g., $(x_1, x_2), (x_2, x_3)$ are must-linked, then (x_1, x_3) is must-linked). To store the constraints, we define a state matrix as $S = \{s_{ij}\}_{N\times N}, s_{ij} \in \{-1, 0, 1\}$. Here, 1/-1 denotes must-link/cannot-link, and 0 indicates an unqueried state. To avoid unnecessary queries, we need to augment the constraints set each time a new constraint is added. We assert that this expansion is only relevant to the preceding constraints that share a common sample with the new constraint, and propose a Fast Transitive Inference (Algorithm 3) method to update the constraints. The correctness of this assertion and algorithm is proved in Appendix A.4.

2.3.3. Discussion.

Complexity Analysis. The computational complexity of the A3S algorithm comprises three distinct components. During the pairwise probability estimation, the complexity is $O(NM)$, where M denotes the number of neighbors considered for each sample. This is due to the need to only account for the neighboring clusters in each query. In the adaptive clustering process, the complexity depends on the specific algorithm, which we mark as $O(A)$. The complexity of the query strategy is less than $O(k^2 + kL_2)$, where k is the initial cluster number and L_2 denotes the iteration number for A3S. Because there are k^2 cluster pairs at the beginning, and we only need to re-calculate k cluster pairs in each iteration. Consequently, the computational complexity of A3S is at most $O(NM + A + k^2 + kL_2)$.

Application of A3S. A3S excels in two key scenarios: First, in clustering tasks lacking prior data information (strange environment), like the number of classes or sample distribution, where traditional methods falter or require extensive human queries. Second, in ongoing real-world applications needing regular data aggregation (expanding environment), such as weekly updates. Here, A3S adeptly merges new with existing clusters, efficiently managing redundancy. It's particularly suitable when each period's data is already high-quality, naturally meeting purity constraints.

3. Experiments

We organize the experiments as follows: we explain the experimental setup in Section 3.1; we compare A3S with state-of-the-art active clustering methods and present the detailed results in Section 3.2; then we compare the performance of A3S when applied to different clustering algorithms in Section 3.3; lastly, we explore the influence of components in A3S in Section 3.4 to 3.5.

3.1. Experimental Setup

Datasets. We sampled six datasets from four real-world image sources for the experiments: `Market-1501` (Zheng et al., 2015), which comprises human body images from 1501 individuals. We use two subsets: `MK20` (351 images from 20 people) and `MK100` (1650 images from 100 people); `Humbi` (Yu et al., 2020), a large multiview image dataset focused on human expressions like faces, and we extracted a subset `Humbi-Face` containing 5600 face images from 100 different people; `Handwritten` (Dua et al., 2017), a collection containing 2000 samples of handwritten digits from '0' to '9'. We use the Fourier coefficient features in the experiments. (4) `MS1M` (Guo et al., 2016), a substantial benchmark dataset commonly used in face recognition tasks, and we sampled two large subsets: `MS1M-10k`, `MS1M-100k`. The details of these datasets are shown in Table 1. Regarding the commonly used benchmarks in previous constrained clustering methods (e.g., UCI datasets (Asuncion & Newman, 2007)), they are not appropriate for our problems due to the very small sample and class sizes. Consequently, we have not evaluated the performance on those benchmarks. We seek to demonstrate that our method is generally workable for different types of data/applications with a wide range of cluster numbers and sample numbers.

Baselines. To validate the performance of our methods, a set of baselines and state-of-the-art algorithms are compared. **Random** (Basu et al., 2003) randomly selects pairwise constraints. **FFQS** (Basu et al., 2004) uses the farthest-first scheme to acquire diversified samples and pairwise constraints. **NPU** (Xiong et al., 2013) uses the classic entropy-based principle to select informative samples to construct pairwise constraints. We use PCKMeans (Basu et al., 2003) as the semi-supervised clustering algorithm for these three methods, because PCKMeans best suits the pairwise constraints manner, and is suitable for large data sets with sparse high-dimensional data (Cai et al., 2023). **URASC** (Xiong et al., 2016) aims to iteratively query pairwise constraints that can maximally reduce the uncertainty of spectral clustering. **COBRA** over-cluster a dataset with K-means, then iteratively selects the closest cluster pairs for querying.

Implementation. For the baseline methods, we maintain the same hyperparameter settings as reported in their original papers to ensure fairness in the comparison. Note that

Algorithm 1 Adaptive Active Aggregation and Splitting

Input: Data X, query limit $Q_{\max}$, index $q = 0$, clustering algorithm $\mathcal{A}$
Initialization: Using $\mathcal{A}$ to generate initial clustering Ω
for *iter in* $1 : L_2$ **do**
  Get a batch of k candidate cluster pairs whose aggregation probability ranks top-k as $\mathcal{C} = \{(w_i, w_j) | w_i, w_j \in \Omega\}$
  Use Eq. (4) to measure cluster pairs in $\mathcal{C}$ and choose the top pair
  if $q < Q_{\max}$ **then**
    Implement Purity Test on w_1 and w_2 with Eq. (6)
    if *Both clusters pass the test* **then**
      Select their central samples as x_1 and x_2
      Require oracle to query (x_1, x_2)
      Aggregate w_1 and w_2 if (x_1, x_2) is must-linked
    else
      Split w_1 or/and w_2 with Algorithm 2
    end if
    Update constraints set with Algorithm 3
    Update q by adding the newly invested number of queries
  else
    Terminate and return the result
  end if
end for

Algorithm 2 Subcluster Partition

Input: Cluster w; subcluster lists $\mathcal{N} = \{\}$
Sort samples in w in ascending order by their distance to the centroid
for *i in w* **do**
  Select one samples from each subcluster in $\mathcal{N}$, and get S
  Query i with sample in S till a must-link is reached or all samples in S have been queried
  if *i is must-linked to j in S* **then**
    Move i from w to the subcluster $\mathcal{N}_u$, $j \in \mathcal{N}_u$
  else
    Move i from w to an empty subcluster, and add the new subcluster $\mathcal{N}_i = \{i\}$ to $\mathcal{N}$
  end if
end for

Algorithm 3 Fast Transitive Inference

Input: State matrix S, new constraints (s, t).
for *i in* (s, t) **do**
  Get $\mathcal{ML} = \{j | S[i, j] = 1\}$
  Get $\mathcal{CL} = \{j | S[i, j] = -1\}$
  Let $S[p, q] = 1$, for $p, q \in \mathcal{ML}$
  Let $S[p, q] = -1$, for $p \in \mathcal{ML}, q \in \mathcal{CL}$
end for

Table 1. Statistical information about six datasets. N, K, and D represent the sample numbers, class numbers, and the dimension of features. b indicates whether the sample quantities between different classes are balanced.

	MK20	MK100	Handwritten	Humbi-Face	MS1M-10k	MS1M-100k
N	351	1650	2000	5600	10000	100000
K	20	100	10	100	146	1469
D	256	256	76	256	512	512
b	✗	✗	✓	✓	✗	✗

these baseline methods require the real cluster number as input, which is provided in our experiments. In many clustering applications, however, this number is typically not known beforehand, thus they are at an advantage. For the initialization of A3S[1], we use isotonic regression to learn pairwise probability and utilize Fast Probabilistic Clustering (FPC) (Liu et al., 2022) as the adaptive clustering method. Here, we choose FPC because it only requires the pairwise probability for automatic data grouping, which is easy to implement. **COBRA** cannot assign a proper initial cluster number itself, so we use the same k as A3S for a fair comparison. More details are presented in Appendix B.

Evaluation. As discussed in Vinh et al. (2009), NMI can exhibit bias towards fine-grained clustering. Therefore, in addition to NMI, we employ the Adjusted Rand Index (ARI) (Hubert & Arabie, 1985) to evaluate the performance. NMI and ARI fall within the range of (0,1] and [-1,1] respectively, with larger values indicating superior clustering performance. To further investigate whether A3S effectively resolves the category fission and recover the real clustering structure, we introduce two supplementary metrics: (1) the Fission Rate ($\Upsilon = \frac{k}{K}$), where K is the real class number and k is the resulting cluster number; and (2) the entropy ratio between resulting clusters (Ω) and real class partitions (C), $r = \frac{\mathbb{H}(\Omega)}{\mathbb{H}(C)}$. When Υ and r approaches 1, we conclude that A3S has effectively mitigated the category fission problem, and successfully recovered the true structure of C.

[1]The code is available at https://github.com/xiangtanshi/A3S.

3.2. Comparison with SOTA Active Clustering Methods

We compare A3S and five baseline methods on four datasets with different numbers of queries in terms of both NMI and ARI. The results are shown in Figure 3. Overall, A3S has higher NMI and ARI values than other methods on these data sets when setting the same number of queries, and A3S requires only a small amount of queries to improve the NMI and ARI values significantly. In addition, we observe that both A3S and **COBRA** (cluster-based methods) improve steadily with the increase of queries, while **Random**, **FFQS**, **NPU** and **URASC** (semi-supervised clustering based methods) show fluctuations on all data sets. This is because genuine supervisory information can sometimes be detrimental to clustering, as it may introduce violations (e.g., (x_i, x_j) is cannot-linked, but their similarity to x_k is both very high). This problem also exists for the latest semi-supervised clustering methods such as PCSKM (Vouros & Vasilaki, 2021). However, it does not mean that this line of work is not applicable. One common advantage of them is that they can ultimately improve the NMI and ARI value to 1.0 if enough queries can be provided (typically less than $N \times \log(N)$). They are a good choice when the target is to reveal the cluster identity for all samples accurately, or only low-quality features are available and the NMI of the initial clustering is lower than 0.2.

The detailed running result of A3S is in Table 2. A3S

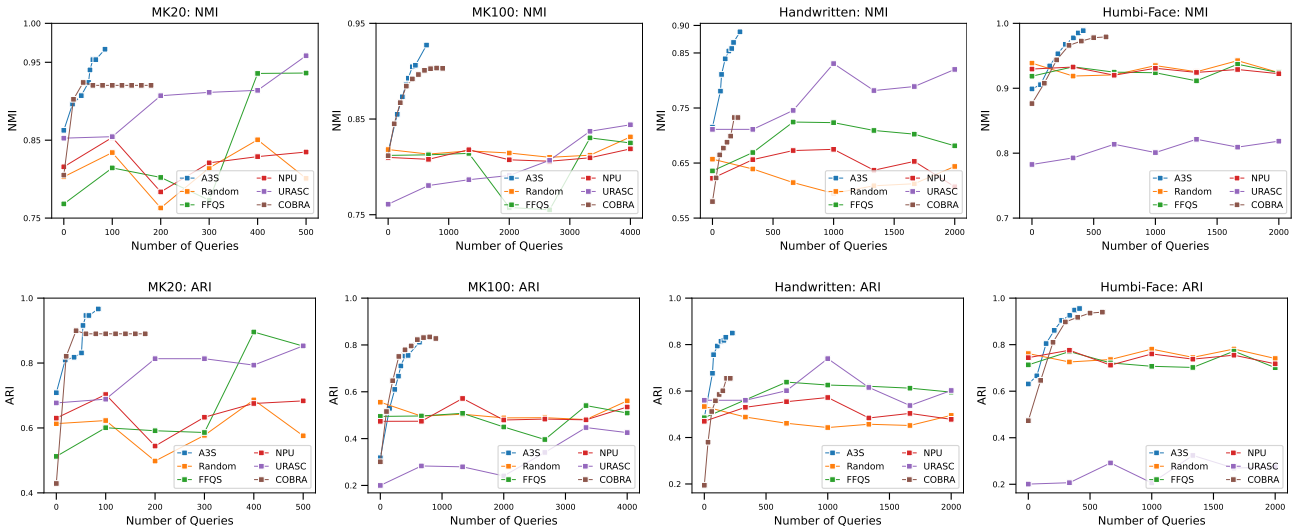

Figure 3. Comparing the performance of A3S and baselines on four datasets in terms of query count. More queries are invested for baselines to illustrate their characteristics.

significantly reduces the fission rate and the entropy ratio to almost 1.0 on all datasets, with a high clustering purity. This validates that A3S can effectively reveal the true clustering structure of data. It's important to highlight that these results are obtained without any prior knowledge about the number of classes or class distribution. Additionally, A3S exhibits robustness to the choice of adaptive cluster number. As detailed in Appendix C, increasing the adaptive number does not significantly alter the number of queries needed to obtain the desired result.

Next, we delineate the distinctions in the results yielded by A3S and **COBRA**. Although **COBRA** quickly improves the NMI and ARI values, it cannot further enhance the clustering even when more queries are invested (all must-link clusters have already been discovered), leading to a performance ceiling dictated by cluster purities. Recent methods like COBRAS (Van Craenendonck et al., 2018) and AQM+MEE (Deng et al., 2023b) also encounter similar issues. In contrast, A3S not only delivers high-quality clusters but also identifies a subset of outlier samples. This approach enables continued augmentation of the NMI and ARI values through strategic querying of these outliers in combination with existing clusters. For instance, A3S and **COBRA** reach NMI values of 0.93 and 0.90 on `MK100` separately, but the overall clustering purity of A3S is 0.9752, far surpassing **COBRA** that is 0.8509. Further, by querying the outlier samples of A3S with their neighbor clusters until 'must-link' is observed and they are aggregated to the corresponding clusters, A3S can reach an NMI of 0.99 with less than 500 more queries.

Table 2. Detailed results of A3S in Figure 3: the total running time (seconds), initial and ending fission rate (Υ) and entropy ratio (r), and the final clustering purity.

	Time	Υ_{init}	Υ_{end}	r_{init}	r_{end}	purity
`MK20`	0.93	2.05	0.95	1.25	1.01	0.9886
`MK100`	5.58	2.83	0.94	1.30	1.08	0.9752
`Handwritten`	6.39	8.6	1.00	1.55	1.00	0.9165
`Humbi-Face`	37.62	3.48	1.01	1.19	1.0	0.9745

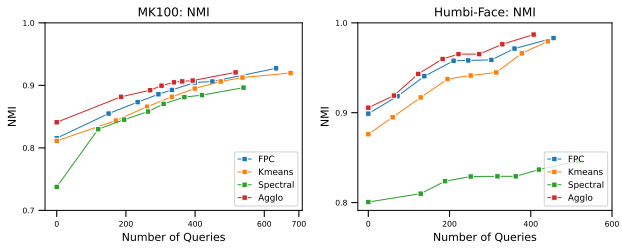

Figure 4. Performance of A3S on `MK100` and `Humbi-Face` when utilizing different clustering algorithms to generate the initial clustering result.

3.3. A3S for Different Clustering Algorithms

To assess the compatibility of A3S on different clustering algorithms, we additionally use three classic clustering algorithms to generate the initial clustering (adaptive cluster number is provided by FPC): K-means clustering (Choo et al., 2020), Spectral clustering (Von Luxburg, 2007) and Agglomerative Clustering (Murtagh & Legendre, 2014). We test these versions of A3S on `MK100` and `Humbi-Face`, and the results are shown in Figure 4. Besides, we quantify the difference between these initial clustering results with their mutual ARI value in Table 3. We have two observations: first, the initial clustering outcomes derived from different algorithms exhibit substantial variability (mutual ARI value is typically lower than 0.6); second, A3S demon-

Table 3. The ARI value between the clustering results of different clustering algorithms. We use F, K, S, and A to represent Fast Probabilistic Clustering, K-means Clustering, Spectral Clustering, and Agglomerative Clustering.

	MK100				Humbi-Face			
	F	K	S	A	F	K	S	A
F	1.000	0.581	0.187	0.597	1.000	0.600	0.272	0.805
K	0.581	1.000	0.140	0.408	0.600	1.000	0.178	0.573
S	0.187	0.140	1.000	0.249	0.272	0.178	1.000	0.276
A	0.597	0.408	0.249	1.000	0.805	0.573	0.276	1.000

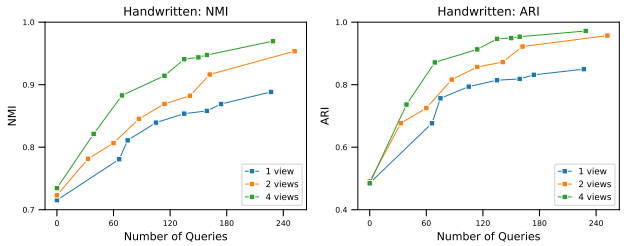

Figure 5. Performance of A3S on `Handwritten` when 1, 2, or 4 views of the feature are used.

strates a consistent ability to enhance the clustering performance across various algorithms. The results validate that A3S is robust to the initial clustering results, and can be effectively applied to other clustering algorithms without modification. In contrast, previous active clustering methods are typically designed and applicable to a specific clustering algorithm like DBSCAN (Mai et al., 2013) and Spectral Clustering (Xiong et al., 2016).

3.4. Influence of Estimated Pairwise Probability Quality

Better pairwise probability can lead to improved clustering performance, but its impact on A3S is unexplored. Multi-view clustering (Yang & Wang, 2018) is the major approach in this domain, and we utilize `Handwritten` to investigate this aspect. The complete `Handwritten` dataset comprises four distinct feature types for each sample. Following the setup in Liu et al. (2022), we initially learn the pairwise probabilities using features from each view. Subsequently, pairwise probabilities are aggregated across V different views employing the following formula (Liu et al., 2022): $\mathbb{P}(e_{ij} = 1|d_{ij}^1, \cdots, d_{ij}^V) = \frac{\prod_{m=1}^V \mathbb{P}(e_{ij}=1|d_{ij}^m)}{\prod_{m=1}^V \mathbb{P}(e_{ij}=1|d_{ij}^m)+\prod_{m=1}^V \mathbb{P}(e_{ij}=0|d_{ij}^m)}$, where d_{ij}^m is the pairwise distance in the m-th view.

In the investigation of A3S's performance with varying views (one/two/four) as illustrated in Figure 5, we observed that employing multi-view clustering slightly improves the initial clustering performance, but significantly boosts A3S's overall effectiveness. This approach leads to quicker convergence and nearly perfect NMI and ARI values. The improvement is largely attributed to multi-view aggregated pairwise probabilities, which make outlier samples distinguishable from neighbor samples that are from different classes, thereby preventing them from being in one cluster during initial clustering. Empirically, In four-view scenarios, despite a higher initial fission rate of 10.1, the initial clustering purity improves from 0.916 to 0.989.

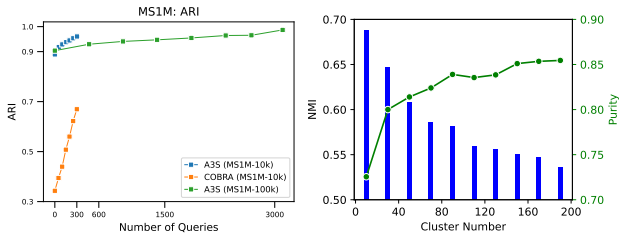

Figure 6. Left: performance of A3S on `MS1M-10k` and `MS1M-100k`. Right: trade-off between initial clustering quality and initial clustering purity for **COBRA** on `Handwritten`.

3.5. Performance of A3S on Large datasets.

We further test A3S on two large datasets `MS1M-10k` and `MS1M-100k`, where the initial cluster number is determined by FPC for both A3S and **COBRA**. The result is in Figure 6. We have two observations: (1) A3S can reach near-optimal ARI value (0.995) with only 3100 queries for `MS1M-100k` in 8581.86 seconds, which validates that it is scalable to large datasets. As far as we know, this is the largest dataset ever used in active clustering research. (2) **COBRA** is sensitive to the initial cluster number and can waste too many queries to get a bad clustering result.

3.6. Case Study for A3S

We present a case study in Figure 2 to demonstrate A3S's mechanism using a two-class dataset. Initially, data points are clustered into seven groups, including two outlier clusters. In the first iteration, clusters w_6 and w_7 are selected for their largest expected impact on the NMI value. During the purity test, w_6 fails the density test, leading to a subsequent query between its central sample (j_0) and the margin sample ($j_{j_0,0.7}$) of the sphere that is centered at j_0 and contains 70% samples of w_6. The cannot-link result indicates w_6 fails the purity test, so it is split into two sub-clusters using Algorithm 2, which costs 12 queries. In the second iteration, clusters w_9 and w_7 are selected, both passing the purity test. Their central samples are queried, and they are merged into w_{10} following the must-link result.

A3S addresses outlier detection through purity tests and subcluster partitioning. However, the purity test can discover most but not all low-quality clusters whose purity is lower than 0.7. As a remedy, we could resort to multi-view features which can significantly improve the clustering purity and reduce the number of outliers, as shown in Section 3.4. We remark that when multi-view

data is available, the purity test and subcluster partition of A3S can be omitted, streamlining the procedure. In contrast, **COBRA** attempts to mitigate outlier issues by opting for a larger cluster number to enhance initial clustering purity, which is inefficient as depicted in the right image of Figure 6 (the clustering purity increases very slowly).

4. Related Work

Query Strategy in Active Clustering. Recent Active clustering methods have embraced a trend of incorporating sample uncertainty into their query strategies. These methods frequently utilize entropy to quantify uncertainty (Abin, 2016; Xiong et al., 2016; Shi et al., 2020). A common task involves estimating the probability of a sample belonging to different clusters or neighborhoods (Xiong et al., 2013). Additionally, alternative criteria such as maximum expected error reduction (Wang & Davidson, 2010) and maximum expected clustering change (Biswas & Jacobs, 2014) have been proposed to assess the stability of clustering results when perturbing the similarity values between two samples.

Constraints in Semi-supervised Clustering. When using the must-link and cannot-link constraints to perform SSC, two aspects are usually taken into consideration: the transitive inference of constraints and the combination of constraints to specific clustering algorithms. A few studies (Lutz et al., 2021) address transitive inference with graph-based techniques instead of a brute-force manner. In addition, recent studies that optimize clustering results with constraints in SSC have explored various approaches, Vouros & Vasilaki (2021) explores Kmeans clustering for high dimensional data; Yang et al. (2022) attempts to correctly infer the number of clusters for hierarchical clustering; Ren et al. (2018) tries to utilize prior knowledge to determine a proper cluster number for density-based clustering; Chen & Zhong (2022) develops a graph-based SSC that is robust to noise, but is not suitable for large datasets. However, there remains a relatively unexplored potential in integrating these modern SSC approaches with active clustering, which presents a promising avenue for future research.

5. Conclusion

This paper studies the cluster-adjustment scheme in active clustering, offering theoretical guidance for non-deteriorating cluster aggregation and quantifying the impact of human queries and aggregation operations. We then propose A3S, a general framework that does not rely on the dataset prior. Through extensive testing, A3S demonstrates its effectiveness on diverse real-world datasets with varying class numbers and distributions. We will explore the application of A3S on more complex multi-view datasets and gigantic datasets at the million level in the future.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (2022YFB3104701) and Alibaba Group through Alibaba Research Intern Program. We appreciate the reviewers for their insightful feedback and advice, these constructive criticism and recommendations have been invaluable in helping us improve the quality of this work.

Impact Statement

This paper presents a work that aims to advance the field of Active Clustering. There are some potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Abin, A. A. Clustering with side information: Further efforts to improve efficiency. *Pattern Recognition Letters*, 84:252–258, 2016.

Anand, S., Mittal, S., Tuzel, O., and Meer, P. Semi-supervised kernel mean shift clustering. *TPAMI*, pp. 1201–1215, Jun 2014. doi: 10.1109/tpami.2013.190. URL http://dx.doi.org/10.1109/tpami.2013.190.

Asuncion, A. and Newman, D. Uci machine learning repository, 2007.

Basu, S., Banerjee, A., and Mooney, R. Semi-supervised clustering by seeding. *ICML*, Jul 2002.

Basu, S., Bilenko, M., and Mooney, R. J. Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering. In *ICML*, pp. 42–49. Citeseer, 2003.

Basu, S., Banerjee, A., and Mooney, R. J. Active semi-supervision for pairwise constrained clustering. In *ICDM*, pp. 333–344. SIAM, 2004.

Bilenko, M., Basu, S., and Mooney, R. J. Integrating constraints and metric learning in semi-supervised clustering. In *ICML*, pp. 11, 2004.

Biswas, A. and Jacobs, D. Active image clustering with pairwise constraints from humans. *IJCV*, 108:133–147, 2014.

Cai, J., Hao, J., Yang, H., Zhao, X., and Yang, Y. A review on semi-supervised clustering. *Information Sciences*, 2023.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by

contrasting cluster assignments. *NeurIPS*, 33:9912–9924, 2020.

Celebi, M. E. and Aydin, K. *Unsupervised learning algorithms*, volume 9. Springer, 2016.

Chen, L. and Zhong, Z. Adaptive and structured graph learning for semi-supervised clustering. *Information Processing & Management*, 59(4):102949, 2022.

Choo, D., Grunau, C., Portmann, J., and Rozhon, V. k-means++: few more steps yield constant approximation. In *ICML*, pp. 1909–1917. PMLR, 2020.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019.

Deng, X., Wang, W., Feng, F., Zhang, H., He, X., and Liao, Y. Counterfactual active learning for out-of-distribution generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11362–11377, 2023a.

Deng, Y., Yuan, Y., Fu, H., and Qu, A. Query-augmented active metric learning. *Journal of the American Statistical Association*, 118(543):1862–1875, 2023b.

Dua, D., Graff, C., et al. Uci machine learning repository, 2017. *URL http://archive. ics. uci. edu/ml*, 7(1), 2017.

González-Almagro, G., Peralta, D., De Poorter, E., Cano, J.-R., and García, S. Semi-supervised constrained clustering: An in-depth overview, ranked taxonomy and future research directions. *arXiv preprint arXiv:2303.00522*, 2023.

Guo, Y., Zhang, L., Hu, Y., He, X., and Gao, J. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, pp. 87–102. Springer, 2016.

He, Y., Reinert, G., Wang, S., and Cucuringu, M. Sssnet: semi-supervised signed network clustering. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pp. 244–252. SIAM, 2022.

Hubert, L. and Arabie, P. Comparing partitions. *Journal of classification*, 2:193–218, 1985.

Khan, M. M. R., Siddique, M. A. B., Arif, R. B., and Oishe, M. R. Adbscan: Adaptive density-based spatial clustering of applications with noise for identifying clusters with varying densities. In *2018 4th international conference on electrical engineering and information & communication technology (iCEEiCT)*, pp. 107–111. IEEE, 2018.

Khanum, M., Mahboob, T., Imtiaz, W., Ghafoor, H. A., and Sehar, R. A survey on unsupervised machine learning algorithms for automation, classification and maintenance. *International Journal of Computer Applications*, 119(13), 2015.

Kvalseth, T. O. Entropy and correlation: Some comments. *IEEE Transactions on Systems, Man, and Cybernetics*, 17 (3):517–519, 1987.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Liu, J., Liu, J., Yan, S., Jiang, R., Tian, X., Gu, B., Chen, Y., Shen, C., and Huang, J. Mpc: Multi-view probabilistic clustering. In *CVPR*, pp. 9509–9518, 2022.

Liu, J., Liu, J., Jiang, R., Liu, X., Gu, B., Chen, Y., Shen, C., and Ye, J. Self-learning symmetric multi-view probabilistic clustering. *arXiv preprint arXiv:2305.07307*, 2023.

Lu, Z. and Leen, T. Semi-supervised learning with penalized probabilistic clustering. *NeurIPS*, 17, 2004.

Lutz, Q., De Panafieu, E., Stein, M., and Scott, A. Active clustering for labeling training data. *NeurIPS*, 34:8469–8480, 2021.

Mai, S. T., He, X., Hubig, N., Plant, C., and Bohm, C. Active density-based clustering. In *ICDM*, Dec 2013. doi: 10.1109/icdm.2013.39. URL http://dx.doi.org/10.1109/icdm.2013.39.

Murtagh, F. and Legendre, P. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, 31:274–295, 2014.

Ren, Y., Hu, X., Shi, K., Yu, G., Yao, D., and Xu, Z. Semi-supervised denpeak clustering with pairwise constraints. In *PRICAI 2018: Trends in Artificial Intelligence: 15th Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, August 28–31, 2018, Proceedings, Part I 15*, pp. 837–850. Springer, 2018.

Shi, Y., Yu, Z., Cao, W., Chen, C. P., Wong, H.-S., and Han, G. Fast and effective active clustering ensemble based on density peak. *TNNLS*, 32(8):3593–3607, 2020.

Van Craenendonck, T., Dumancic, S., and Blockeel, H. Cobra: A fast and simple method for active clustering with pairwise constraints. *IJCAI*, 2017.

Van Craenendonck, T., Dumančić, S., Van Wolputte, E., and Blockeel, H. Cobras: Interactive clustering with pairwise queries. In *ISIDA*, pp. 353–366. Springer, 2018.

Vinh, N. X., Epps, J., and Bailey, J. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *ICML*, pp. 1073–1080, 2009.

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

Vouros, A. and Vasilaki, E. A semi-supervised sparse k-means algorithm. *Pattern Recognition Letters*, 142:65–71, 2021.

Wang, X. and Davidson, I. Active spectral clustering. In *ICDM*, pp. 561–568. IEEE, 2010.

Xiong, C., Johnson, D. M., and Corso, J. J. Active clustering with model-based uncertainty reduction. *TPAMI*, 39(1): 5–17, 2016.

Xiong, S., Azimi, J., and Fern, X. Z. Active learning of constraints for semi-supervised clustering. *TKDE*, 26(1): 43–54, 2013.

Yang, T., Pasquier, N., and Precioso, F. Semi-supervised consensus clustering based on closed patterns. *Knowledge-Based Systems*, 235:107599, 2022.

Yang, Y. and Wang, H. Multi-view clustering: A survey. *Big Data Mining and Analytics*, 1(2):83–107, 2018.

Yu, Z., Yoon, J. S., Lee, I. K., Venkatesh, P., Park, J., Yu, J., and Park, H. S. Humbi: A large multiview dataset of human body expressions. In *CVPR*, pp. 2990–3000, 2020.

Zhang, R., Du, T., Qu, S., and Sun, H. Adaptive density-based clustering algorithm with shared knn conflict game. *Information Sciences*, 565:344–369, 2021.

Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., and Tian, Q. Scalable person re-identification: A benchmark. In *ICCV*, pp. 1116–1124, 2015.

A. Proofs

A.1. Proof of Theorem 2.5

Proof of Theorem 2.5. Let p and q denote the sizes of w_1 and w_2, respectively. We further assume that the class index of these outliers is $i_j \in \{1, 2, \cdots, K\}$, where $j \in \{1, 2, \cdots, (1-t_1)p + (1-t_2)q\}$. For ease of presentation, for any class index $i \in \{1, 2, \cdots, K\}$, we use s_i to denote the class size, i.e., $|c_i| = s_i$. Without loss generality, we assume that $q \geq p$ throughout this proof.

By the definition of mutual information, we have

$$\begin{aligned}
\mathbb{I}(\Omega^*; C) &= \mathbb{I}(\Omega; C) + \sum_{\tau=1}^{K} \mathbb{P}(w_{1,2} \cap c_\tau) \log \frac{\mathbb{P}(w_{1,2} \cap c_\tau)}{\mathbb{P}(w_{1,2})\mathbb{P}(c_\tau)} - \sum_{\tau=1}^{K} \mathbb{P}(w_1 \cap c_\tau) \log \frac{\mathbb{P}(w_1 \cap c_\tau)}{\mathbb{P}(w_1)\mathbb{P}(c_\tau)} \\
&\quad - \sum_{\tau=1}^{K} \mathbb{P}(w_2 \cap c_\tau) \log \frac{\mathbb{P}(w_2 \cap c_\tau)}{\mathbb{P}(w_2)\mathbb{P}(c_\tau)} \\
&= \mathbb{I}(\Omega; C) + \underbrace{\sum_{\tau=1}^{K} \frac{|w_{1,2} \cap c_\tau|}{N} \log \frac{N \cdot |w_{1,2} \cap c_\tau|}{|w_{1,2}| \cdot |c_\tau|}}_{\text{(I)}} - \underbrace{\sum_{\tau=1}^{K} \frac{|w_1 \cap c_\tau|}{N} \log \frac{N \cdot |w_1 \cap c_\tau|}{|w_1| \cdot |c_\tau|}}_{\text{(II)}} \\
&\quad - \underbrace{\sum_{\tau=1}^{K} \frac{|w_2 \cap c_\tau|}{N} \log \frac{N \cdot |w_2 \cap c_\tau|}{|w_2| \cdot |c_\tau|}}_{\text{(III)}}.
\end{aligned} \tag{7}$$

Then we bound these three terms respectively. For Term (I), we have

$$\begin{aligned}
\text{(I)} &= \frac{|w_{1,2} \cap c_1|}{N} \log \frac{N \cdot |w_{1,2} \cap c_1|}{|w_{1,2}| \cdot |c_1|} + \sum_{\tau=2}^{K} \frac{|w_{1,2} \cap c_\tau|}{N} \log \frac{N \cdot |w_{1,2} \cap c_\tau|}{|w_{1,2}| \cdot |c_\tau|} \\
&= \frac{t_1 p + t_2 q}{N} \log \frac{N(t_1 p + t_2 q)}{(p+q)s_1} + \sum_{j=1}^{(1-t_1)p+(1-t_2)q} \frac{1}{N} \log \frac{N \cdot |w_{1,2} \cap c_{i_j}|}{(p+q)s_{i_j}}.
\end{aligned} \tag{8}$$

Similarly, we have

$$\begin{aligned}
\text{(II)} &= \frac{|w_1 \cap c_1|}{N} \log \frac{N \cdot |w_1 \cap c_1|}{|w_1| \cdot |c_1|} + \sum_{\tau=2}^{K} \frac{|w_1 \cap c_\tau|}{N} \log \frac{N \cdot |w_1 \cap c_\tau|}{|w_1| \cdot |c_\tau|} \\
&= \frac{t_1 p}{N} \log \frac{N t_1}{s_1} + \sum_{j=1}^{(1-t_1)p} \frac{1}{N} \log \frac{N \cdot |w_1 \cap c_{i_j}|}{p s_{i_j}},
\end{aligned} \tag{9}$$

and

$$\begin{aligned}
\text{(III)} &= \frac{|w_2 \cap c_1|}{N} \log \frac{N \cdot |w_2 \cap c_1|}{|w_2| \cdot |c_1|} + \sum_{\tau=2}^{K} \frac{|w_2 \cap c_\tau|}{N} \log \frac{N \cdot |w_2 \cap c_\tau|}{|w_2| \cdot |c_\tau|} \\
&= \frac{t_2 q}{N} \log \frac{N t_2}{s_2} + \sum_{j=1}^{(1-t_2)q} \frac{1}{N} \log \frac{N \cdot |w_2 \cap c_{i_j}|}{q s_{i_j}}.
\end{aligned} \tag{10}$$

Plugging Eq. (8), Eq. (9), and Eq. (10) into Eq. (7), together with the fact that

$$|w_{1,2} \cap c_{i_j}| \geq \max\{|w_1 \cap c_{i_j}|, |w_2 \cap c_{i_j}|\}, \quad \forall j \in \{1, 2, \cdots, (1-t_1)p + (1-t_2)q\},$$

we obtain that

$$\mathbb{I}(\Omega^*;C) \geq \mathbb{I}(\Omega;C) + \underbrace{\frac{t_1 p}{N}\log\frac{t_1 p + t_2 q}{t_1(p+q)} + \frac{t_2 q}{N}\log\frac{t_1 p + t_2 q}{t_2(p+q)}}_{\text{(IV)}} \tag{11}$$

$$- \underbrace{\Big(\frac{(1-t_1)p + (1-t_2)q}{N}\log(p+q) - \frac{(1-t_1)p}{N}\log p - \frac{(1-t_2)q}{N}\log q\Big)}_{\text{(V)}}. \tag{12}$$

For Term (IV) in Eq. (11), by the Taylor expansion

$$\log(1+x) = \sum_{u=1}^{\infty}\frac{(-1)^{u-1}}{u}\cdot x^u,$$

we have

$$\begin{aligned}
\text{(IV)} &= \frac{t_1 p}{N}\sum_{u=1}^{\infty}\frac{(-1)^{u-1}}{u}\cdot\Big(\frac{(t_2-t_1)q}{t_1(p+q)}\Big)^u + \frac{t_2 q}{N}\sum_{u=1}^{\infty}\frac{(-1)^{u-1}}{u}\cdot\Big[\frac{(t_1-t_2)p}{t_2(p+q)}\Big]^u \\
&= \sum_{v=1}^{\infty}\frac{1}{2v-1}\cdot\Big[\frac{t_1 p}{N}\cdot\Big(\frac{(t_2-t_1)q}{t_1(p+q)}\Big)^{2v-1} + \frac{t_2 q}{N}\Big(\frac{(t_1-t_2)p}{t_2(p+q)}\Big)^{2v-1}\Big] \\
&\quad - \sum_{v=1}^{\infty}\frac{1}{2v}\cdot\Big[\frac{t_1 p}{N}\cdot\Big(\frac{(t_2-t_1)q}{t_1(p+q)}\Big)^{2v} + \frac{t_2 q}{N}\Big(\frac{(t_1-t_2)p}{t_2(p+q)}\Big)^{2v}\Big].
\end{aligned} \tag{13}$$

For ease of presentation, we denote $m = q/p \geq 1$. Then for any $v \geq 1$, we have

$$\begin{aligned}
&\sum_{v=1}^{\infty}\frac{1}{2v}\cdot\Big[\frac{t_1 p}{N}\cdot\Big(\frac{(t_2-t_1)q}{t_1(p+q)}\Big)^{2v} + \frac{t_2 q}{N}\Big(\frac{(t_1-t_2)p}{t_2(p+q)}\Big)^{2v}\Big] \\
&\quad = \sum_{v=1}^{\infty}\frac{pq(t_1-t_2)^{2v}}{2vN(p+q)^{2v}}\cdot\Big[\frac{q^{2v-1}}{t_1^{2v-1}} + \frac{p^{2v-1}}{t_2^{2v-1}}\Big] \\
&\quad \leq \sum_{v=1}^{\infty}\frac{mp^2(3/10)^{2v}}{2vNp^{2v}(1+m)^{2v}}\cdot(1+m^{2v-1})\cdot\Big(\frac{10p}{7}\Big)^{2v-1} \\
&\quad \leq \frac{3p}{20N}\sum_{v=1}^{\infty}\frac{1}{v}\cdot\Big(\frac{3}{7}\Big)^{2v-1} \leq \frac{0.0716p}{N},
\end{aligned} \tag{14}$$

where the first inequality uses $m = q/p$ and the assumption that $t_1, t_2 \in [0.7, 1]$, the second inequality follows the fact that $m(1+m^{2v-1}) \leq (1+m)^{2v}$, and the last inequality follows that

$$\sum_{v=1}^{\infty}\frac{1}{v}\cdot\Big(\frac{3}{7}\Big)^{2v-1} \leq \frac{3}{7} + \sum_{v=2}^{\infty}\frac{1}{2}\cdot\Big(\frac{3}{7}\Big)^{2v-1} = \frac{267}{560}$$

and simple calculations.

On the other hand, for any $v \geq 1$, we have

$$\begin{aligned}
&\sum_{v=1}^{\infty}\frac{1}{2v-1}\cdot\Big[\frac{t_1 p}{N}\cdot\Big(\frac{(t_2-t_1)q}{t_1(p+q)}\Big)^{2v-1} + \frac{t_2 q}{N}\Big(\frac{(t_1-t_2)p}{t_2(p+q)}\Big)^{2v-1}\Big] \\
&\quad = \sum_{v=1}^{\infty}\frac{pq(t_1-t_2)^{2v-1}}{(2v-1)N(p+q)^{2v-1}}\cdot\Big[\frac{p^{2v-2}}{t_2^{2v-2}} - \frac{q^{2v-2}}{t_1^{2v-2}}\Big] \\
&\quad = \sum_{v=1}^{\infty}\frac{pq(t_1-t_2)^{2v+1}}{(2v+1)N(p+q)^{2v+1}}\cdot\Big[\frac{p^{2v}}{t_2^{2v}} - \frac{q^{2v}}{t_1^{2v}}\Big].
\end{aligned} \tag{15}$$

Furthermore, we have

$$\begin{aligned}\sum_{v=1}^{\infty}\frac{pq(t_1-t_2)^{2v+1}}{(2v+1)(p+q)^{2v+1}}\cdot\Big[\frac{p^{2v}}{t_2^{2v}}-\frac{q^{2v}}{t_1^{2v}}\Big] &\geq -\sum_{v=1}^{\infty}\frac{pq|t_1-t_2|^{2v+1}}{(2v+1)N(p+q)^{2v+1}}\cdot\Big[\frac{p^{2v}}{t_2^{2v}}+\frac{q^{2v}}{t_1^{2v}}\Big]\\ &\geq -\frac{p}{N}\sum_{v=1}^{\infty}\frac{1}{2v+1}\cdot\Big(\frac{3}{10}\Big)^{2v+1}\\ &\geq -\frac{0.0096p}{N},\end{aligned}\tag{16}$$

where the second inequality uses the facts that $(p+q)^{2v+1}\geq q^{2v+1}+qp^{2v}$ and $t_1,t_2\in[0.7,1]$, and the last inequality follows that

$$\sum_{v=1}^{\infty}\frac{1}{2v+1}\Big(\frac{3}{10}\Big)^{2v+1}\leq\frac{9}{1000}+\frac{1}{5}\sum_{v=2}^{\infty}\Big(\frac{3}{10}\Big)^{2v+1}=\frac{9}{1000}+\frac{243}{455000}<0.0096.$$

Combining Eq. (13), Eq. (14), Eq. (15), and Eq. (16), we obtain that

$$\text{(IV)}\geq-\frac{0.0812p}{N}.\tag{17}$$

Term (V) in Eq. (12). For Term (V) in Eq. (12), we have

$$\begin{aligned}\text{(V)} &= \frac{(1-t_1)p+(1-t_2)q}{N}\log(p+q)-\frac{(1-t_1)p}{N}\log p-\frac{(1-t_2)q}{N}\log q\\ &= \frac{(1-t_1)p}{N}\log\frac{p+q}{p}+\frac{(1-t_2)q}{N}\log\frac{p+q}{q}.\end{aligned}\tag{18}$$

Furthermore, we have

$$\frac{(1-t_1)p}{N}\log\frac{p+q}{p}+\frac{(1-t_2)q}{N}\log\frac{p+q}{q}\tag{19}$$

$$\begin{aligned}&\leq\frac{(1-\min\{t_1,t_2\})p}{N}\log\frac{p+q}{p}+\frac{(1-\min\{t_1,t_2\})q}{N}\log\frac{p+q}{q}\\ &=\frac{(1-\min\{t_1,t_2\})}{N}\cdot\Big[p\log\frac{p+q}{p}+q\log\frac{p+q}{q}\Big].\end{aligned}\tag{20}$$

Let

$$\Delta h=\frac{1}{N}\cdot\Big[p\log\frac{p+q}{p}+q\log\frac{p+q}{q}\Big].\tag{21}$$

Combining Eq. (18), Eq. (19), and Eq. (21), we obtain

$$\text{(V)}\leq(1-\min\{t_1,t_2\})\cdot\Delta h.\tag{22}$$

Putting Together. When $t_1,t_2\geq0.7$, plugging Eq. (17) and Eq. (22) into Eq. (11) and Eq. (12), we have

$$\mathbb{I}(\Omega^*;C)\geq\mathbb{I}(\Omega;C)-\frac{0.0812p}{N}-(1-\min\{t_1,t_2\})\cdot\Delta h.\tag{23}$$

Recall that the Δh defined in Eq. (21) takes the form

$$\begin{aligned}\Delta h &= \frac{1}{N}\cdot\Big[p\log\frac{p+q}{p}+q\log\frac{p+q}{q}\Big]\\ &= \frac{p}{N}\cdot\Big(\log(1+m)+m\log\big(1+\frac{1}{m}\big)\Big)\\ &\geq 2\log 2\cdot\frac{p}{N},\end{aligned}\tag{24}$$

where the second equality uses $m = q/p$, the last inequality uses $m \geq 1$. Putting Eq. (23) and Eq. (24) together, we have

$$\mathbb{I}(\Omega^*; C) \geq \mathbb{I}(\Omega; C) - (1.0586 - \min\{t_1, t_2\}) \cdot \Delta h. \tag{25}$$

Then, we calculate the entropy after fusion.

$$\begin{aligned}
\mathbb{H}(\Omega^*) &= \mathbb{H}(\Omega) - \frac{p+q}{N}\log\frac{p+q}{N} + \frac{p}{N}\log\frac{p}{N} + \frac{q}{N}\log\frac{q}{N} \\
&= \mathbb{H}(\Omega) - \frac{p+q}{N}\log(p+q) + \frac{p}{N}\log p + \frac{q}{N}\log q \\
&= \mathbb{H}(\Omega) - \Delta h
\end{aligned} \tag{26}$$

Recall that

$$n_1 = \frac{2\mathbb{I}(\Omega; C)}{\mathbb{H}(\Omega) + \mathbb{H}(C)}, \quad n_2 = \frac{2\mathbb{I}(\Omega^*; C)}{\mathbb{H}(\Omega^*) + \mathbb{H}(C)}.$$

By Eq. (25) and Eq. (26), we know that the sufficient condition of $n_2 \geq n_1$ is

$$\frac{2[\mathbb{I}(\Omega; C) - (1.0586 - \min\{t_1, t_2\}) \cdot \Delta h]}{\mathbb{H}(\Omega) + \mathbb{H}(C) - \Delta h} \geq \frac{2\mathbb{I}(\Omega; C)}{\mathbb{H}(\Omega) + \mathbb{H}(C)},$$

which is equivalent to

$$n_1 = \frac{2\mathbb{I}(\Omega; C)}{\mathbb{H}(\Omega) + \mathbb{H}(C)} \geq 2 \cdot (1.0586 - \min\{t_1, t_2\}),$$

which concludes the proof of Theorem 2.5.

□

A.2. Derivation of Eq. (2)

Derivation of Eq. (2). We consider the queried result of "must-link" ($c_m = c_n$) between w_i and w_j as a conditional event, which is expressed by $\{\forall s \in w_i, \forall t \in w_j, e_{st} = 1 \mid \forall (s,t) \in w_i \text{ or } w_j, e_{st} = 1\}$. Here the condition implies that samples within a cluster are assigned to the same class. Further, we follow the setup in probabilistic clustering (Lu & Leen, 2004; Liu et al., 2022) that models the distribution of clustering based on the pairwise probability. Then, the joint probability density of a clustering $\pi = [z_1, z_2, \cdots, z_m]$ for m samples is expressed as $\mathbb{P}(\pi) = \frac{1}{\alpha}\prod_{s,t\in[1,2,\cdots,m]} \mathbb{P}(e_{st} = 1)^{I(z_s = z_t)} \times \mathbb{P}(e_{st} = 0)^{I(z_s \neq z_t)}$, where $I(\cdot)$ is the indicator function, and α is the normalization factor.

Under the condition of $\forall (s,t) \in w_i$ or $w_j, e_{st} = 1$, the two events $\{\forall s \in w_i, \forall t \in w_j, e_{st} = 1\}$ and $\{\forall s \in w_i, \forall t \in w_j, e_{st} = 0\}$ are mutually exclusive. And we denote them as $y(w_i) = y(w_j)$ and $y(w_i) \neq y(w_j)$ for simplicity. Therefore, by the formula of conditional probability, we can obtain:

$$\begin{aligned}
\mathbb{P}(c_m = c_n) &= \mathbb{P}(\forall s \in w_i, \forall t \in w_j, e_{st} = 1 \mid \forall (s,t) \in w_i \text{ or } w_j, e_{st} = 1) \\
&= \frac{\mathbb{P}(y(w_i) = y(w_j), \forall (s,t) \in w_i \text{ or } w_j, e_{st} = 1)}{\mathbb{P}(\forall (s,t) \in w_i \text{ or } w_j, e_{st} = 1))} \\
&= \frac{\mathbb{P}(y(w_i) = y(w_j), \forall (s,t) \in w_i \text{ or } w_j, e_{st} = 1))}{\mathbb{P}(y(w_i) = y(w_j), \forall (s,t) \in w_i \text{ or } w_j, e_{st} = 1)) + \mathbb{P}(y(w_i) \neq y(w_j), \forall (s,t) \in w_i \text{ or } w_j, e_{st} = 1))} \\
&= \frac{\frac{1}{A}\prod_{s\in w_i, t\in w_j} \mathbb{P}(e_{st} = 1) \prod_{s,t\in w_i, s,t\in w_j} \mathbb{P}(e_{st} = 1)}{\frac{1}{A}[\prod_{s\in w_i, t\in w_j} \mathbb{P}(e_{st} = 1) + \prod_{s\in w_i, t\in w_j} \mathbb{P}(e_{st} = 0)] \prod_{s,t\in w_i, s,t\in w_j} \mathbb{P}(e_{st} = 1)} \\
&= \frac{\prod_{s\in w_i, t\in w_j} \mathbb{P}(e_{st} = 1)}{\prod_{s\in w_i, t\in w_j} \mathbb{P}(e_{st} = 1) + \prod_{s\in w_i, t\in w_j} \mathbb{P}(e_{st} = 0)},
\end{aligned}$$

where A is the normalization factor of the joint probability density for samples in $\{w_i, w_j\}$. □

Remark on the condition. The assignment of classes for samples is subject to clustering. For instance, to categorize a group of sheep, various attributes like gender, age, health, and weight can be the basis for clustering. In our scenario to measure the likelihood of "must-link", what we indeed care about is the probability that the two clusters should be merged during the clustering, rather than the actual classes of these samples (this is the mission of oracles). After all, each class assignment corresponds to a specific meaning in real applications.

Remark on the aggregation probability. Previous studies often evaluate the likelihood of $c_m = c_n$ with heuristic strategies. Common strategies include using the distance between the central samples of two clusters or the closest distance between the clusters themselves to determine which cluster pairs to query. However, these methods that focus on a single sample from a cluster, may become less effective for clusters with irregular shapes, and Eq. (2) provides a better estimation that takes into account the influence of all samples.

Remark on the query style. A3S requires oracles to provide pairwise comparison results for each selected sample pair, which demands minimal domain-specific knowledge and is easier to implement (Xiong et al., 2016), especially when the number of classes is large. In contrast, traditional active learning necessitates that the oracle assigns specific labels to the selected samples or annotates them according to predefined rules (Deng et al., 2023a), causing much heavier costs during the annotation process.

A.3. Justification of Approximation

Regarding the Approximation of Entropy. If $(p+q) \ll N$, then

$$\begin{aligned}\Delta h &= \frac{p+q}{N}\log(p+q) - \frac{p}{N}\log p - \frac{q}{N}\log q \\ &< \frac{p+q}{N}\log(p+q) \\ &< \frac{p+q}{N}\log\frac{N}{p+q}.\end{aligned}$$

Note that $\mathbb{H}(\Omega) = \sum_s \frac{s}{N}\log\frac{N}{s}$, where s is the cluster size like p and q. Together with $p+q \ll N$, we have

$$\Delta h \ll \mathbb{H}(\Omega) < \mathbb{H}(\Omega) + \mathbb{H}(C).$$

Regarding the Mutual Information. If the purity of w_1 and w_2 is 1, and they belong to the same class c_τ, then we have

$$\mathbb{P}(w_1 \cap c_\tau) = \mathbb{P}(w_1), \quad \mathbb{P}(w_2 \cap c_\tau) = \mathbb{P}(w_2), \quad \mathbb{P}(w_{1,2} \cap c_\tau) = \mathbb{P}(w_{1,2}) = \mathbb{P}(w_1) + \mathbb{P}(w_2).$$

Hence, and we have $\mathbb{I}(\Omega^*; C) = \mathbb{I}(\Omega; C)$. This is because

$$\begin{aligned}\mathbb{I}(\Omega^*; C) &= \mathbb{I}(\Omega; C) + \mathbb{P}(w_{1,2} \cap c_\tau)\log\frac{\mathbb{P}(w_{1,2} \cap c_\tau)}{\mathbb{P}(w_{1,2})\mathbb{P}(c_\tau)} \\ &\quad - \mathbb{P}(w_1 \cap c_\tau)\log\frac{\mathbb{P}(w_1 \cap c_\tau)}{\mathbb{P}(w_1)\mathbb{P}(c_\tau)} - \mathbb{P}(w_2 \cap c_\tau)\log\frac{\mathbb{P}(w_2 \cap c_\tau)}{\mathbb{P}(w_2)\mathbb{P}(c_\tau)} \\ &= \mathbb{I}(\Omega; C) + \mathbb{P}(w_{1,2})\log\frac{1}{\mathbb{P}(c_\tau)} - \mathbb{P}(w_1)\log\frac{1}{\mathbb{P}(c_\tau)} - \mathbb{P}(w_2)\log\frac{1}{\mathbb{P}(c_\tau)} \\ &= \mathbb{I}(\Omega; C).\end{aligned}$$

When the purity is less than 1.0, considering Eq. (25), and fact that $\Delta h \ll \min\{\mathbb{H}(\Omega), \mathbb{H}(C)\}$, we have

$$|\mathbb{I}(\Omega^*; C) - \mathbb{I}(\Omega; C)| < (1.0586 - \min\{t_1, t_2\}) \cdot \Delta h \ll \min\{\mathbb{H}(\Omega), \mathbb{H}(C)\}. \tag{27}$$

A.4. Fast Transitive Inference

FTI involves expanding the set of constraints based on the information within the original set. For example, if (x_1, x_2) and (x_2, x_3) are must-link constraints, and (x_1, x_4) is a cannot-link constraint, it implies that (x_1, x_3) must be a must-link constraint, while (x_2, x_4) and (x_3, x_4) must be cannot-link constraints.

To facilitate the process, we present an efficient method, Fast Transitive Inference (FTI), which is designed to discover the transitive closure for the constraint set in A3S. The implementation is shown in Algorithm 3.

The performance of FTI is guaranteed by Theorem A.1.

Theorem A.1 (Completeness of FTI)**.** *By executing the FTI algorithm every time a new human query is made, we can always get the latest transitive closure.*

Proof of Theorem A.1. Suppose the must-link sample sets with samples i and j are denoted as G_i and G_j, respectively. And the sample sets that are cannot-link with i and j are denoted as g_i and g_j. When the constraints between i and j are queried, only the constraints of sample pairs within $\{G_i, G_j, g_i, g_j\}$ may change, as no must-link constraints are built between them and the rest samples. We discuss the two cases where i and j are must-linked or cannot-linked:

1 (i, j) is must-linked. First, FTI updates the constraints for sample pairs related to i, then $G'_i = G_i \cup \{j\}$, $G'_j = G_j \cup G_i$, and $g'_j = g_j \cup g_i$. The constraints between i and G_j, g_j have not yet been updated. Then, FTI updates the constraints for sample pairs related to j, then we have $G'_i = G_i \cup G_j$ and $g'_i = g_i \cup g_j$. And this means all ml constraints between G_i and G_j, and cl constraints between $\{G_i, G_j\}$ and $\{g_i, g_j\}$ are updated and stored in the state matrix S.

2 (i, j) is cannot-linked. First, FTI updates the constraints for sample pairs related to i, then $g'_i = g_i \cup \{j\}$ and $g'_j = g_j \cup G_i$. Then FTI updates the constraints for sample pairs related to j, then we have $g'_i = g_i \cup G_j$. And this means all cl constraints between G_i and G_j are updated and stored in S.

Combining these two scenarios, we finish the proof of Theorem A.1 □

B. Implementation Details of A3S

B.1. Estimating Pairwise Probability

Following the setup in (Liu et al., 2022), we use isotonic regression to learn a regressor that estimates the pairwise posterior probability $\mathbb{P}(e_{ij} = 1|d_{ij})$, where d_{ij} is the Euclidean distance between samples i and j. The estimation encompasses three steps:

(1) Since ground truth labels are unavailable, we employ K-means clustering to generate pseudo labels for the samples. Alternatively, Fast Probabilistic Clustering (FPC) can also be used for this purpose, where similarity values between samples serve as a rudimentary approximation of pairwise probabilities. In practical applications, cosine similarity is particularly well-suited for FPC.

(2) Next, we generate the training data for isotonic regression. We utilize the k-nearest neighbors of each sample to form sample pairs. The independent variable of isotonic regression is the Euclidean distance between two samples in the sample pair. Each pair is labeled as 0 or 1, indicating whether the two samples in each pair share the same pseudo label. The label serves as the dependent variable of isotonic regression.

(3) Finally, we conduct isotonic regression on the gathered data, learning a function that maps the Euclidean distance between two samples to the pairwise probability, i.e., $\mathbb{P}(e_{st} = 1|d_{st})$.

Liu et al. (2022) also proposed Graph-context-aware refinement to enhance the quality of the posterior probability, but it is not an essential component of A3S. Therefore, we did not include it in our experiments. However, incorporating them would further enhance the performance of A3S, as they can improve the quality of the estimated merging probability between cluster pairs.

B.2. Calculating Aggregation Probability

We derive the aggregation probability based on the condition that the purity of two clusters are 1.0. In practical applications, the assumption that both clusters have a purity of 1.0 might not strictly hold. Specifically, the pairwise probability between major samples and outlier samples from w_i and w_j will degrade the aggregation probability (i.e., making the result biased towards 0). To deal with this issue, we propose a variant of $\mathbb{P}(c_m = c_n)$, denoted as $\mathbb{P}_{\text{knn}}(c_m = c_n)$. This variant considers

only the sample pairs within the k-nearest neighbors. Assume that $|w_i| \leq |w_j|$, the formulation is as follows:

$$\mathbb{P}_{\text{knn}}(c_m = c_n) = \frac{\prod\limits_{s \in w_i t \in \text{knn}_{w_j}(s)} \mathbb{P}(e_{st} = 1)}{\prod\limits_{s \in w_i t \in \text{knn}_{w_j}(s)} \mathbb{P}(e_{st} = 1) + \prod\limits_{s \in w_i t \in \text{knn}_{w_j}(s)} \mathbb{P}(e_{st} = 0)}, \tag{28}$$

where $\text{knn}_{w_j}(s)$ denotes the nearest neighbors of s in w_j. Empirical results show that A3S is not sensitive to the number of neighbors, and we consider 4 neighbors for each sample in our experiments.

B.3. Feature Extraction

The facial characteristics in the `Humbi-Face` dataset are extracted using a face recognition model. Similarly, the `MK20` and `MK100` datasets utilize a person re-identification model for body feature extraction (Liu et al., 2022). In the case of `MS1M-10k` and `MS1M-100k`, facial features are extracted using the Arcface model (Deng et al., 2019). The `Handwritten` dataset encompasses four types of features: average pixel features, Fourier coefficient features, Zernike moments features, and Karhunen-Loève coefficient features.

B.4. Hyperparameter Setting

The implementation of A3S involves two hyperparameters: the threshold for density test and the number of neighbors considered in Fast Probabilistic Clustering. We report our choice of these two parameters in Table 4. In particular, the hyper-parameter τ is used to filter out the clusters with low density, and it is set to be slightly lower than the average density among all clusters. For the selection of τ, we first compute density using the formula in Eq. (5) for all clusters, then calculate the mean value as d, and set τ as $d - 0.1$. The final results of A3S are not sensitive to this value, and perturbing it to d or $d - 0.05$ has a negligible impact on the final clustering result.

Table 4. Hyperparameter setting of A3S.

dataset	`MK20`	`MK100`	`Handwritten`	`Humbi-Face`	`MS1M-10k`	`MS1M-100k`
τ	0.5	0.8	0.8	0.5	0.5	0.5
neighbors	50	50	50	50	50	50

B.5. Computing Resources

We utilize a [GeForce RTX 3090 Ti] for feature extraction using DNN models. For the implementation of baseline methods and A3S, we perform the experiments on a machine equipped with an Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz.

C. More Experiment Results

In our ablation study, we examine the impact of varying the adaptive cluster number on A3S. We define this number as $r \cdot k$, where k is the adaptive number determined by FPC and r is a ratio factor set to values in the set 1, 1.5, 2. Utilizing the K-means algorithm, we generate initial clustering with the cluster number $r \cdot k$, and present the corresponding results of A3S in Figure 7. It's observed that the performance of the initial clustering is relatively sensitive to the adaptive cluster number. However, the selection of this adaptive number has a minimal effect on the number of queries required to achieve the desired clustering outcome. This demonstrates A3S's robustness to variations in the adaptive cluster number.

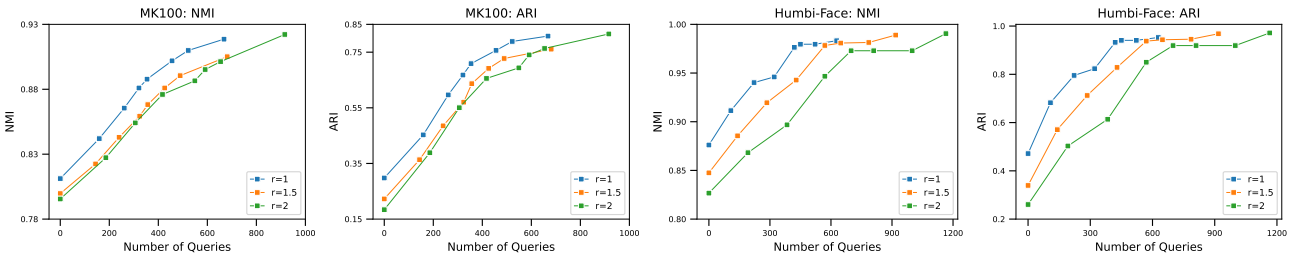

Figure 7. The NMI and ARI performance of A3S when the adaptive cluster number is set as $r \cdot k$, where k is the adaptive number generated by FPC and r is the ratio factor.