Parameter-Free Clustering via Self-Supervised Consensus Maximization

Lijun Zhang[1*], Suyuan Liu[1*], Siwei Wang[2], Shengju Yu[1], Xueling Zhu[3†], Miaomiao Li[4†], Xinwang Liu[1]

[1]College of Computer Science and Technology, National University of Defense Technology, Changsha, 410073, China
[2]Academy of Military Sciences, Beijing, 100091, China
[3]Xiangya Hospital, Central South University, Changsha, 410008, China
[4]College of Electronic Information and Electrical Engineering, Changsha University, Changsha, 410022, China
{junliz, suyuanliu, wangsiwei13}@nudt.edu.cn, yu-shengju@foxmail.com, zhuxueling@csu.edu.cn, miaomiaolinudt@gmail.com, xinwangliu@nudt.edu.cn

Abstract

Clustering is a fundamental task in unsupervised learning, but most existing methods heavily rely on hyperparameters such as the number of clusters or other sensitive settings, limiting their applicability in real-world scenarios. To address this long-standing challenge, we propose a novel and fully parameter-free clustering framework via **S**elf-supervised **C**onsensus **Max**imization, named SCMax. Our framework performs hierarchical agglomerative clustering and cluster evaluation in a single, integrated process. At each step of agglomeration, it creates a new, structure-aware data representation through a self-supervised learning task guided by the current clustering structure. We then introduce a nearest neighbor consensus score, which measures the agreement between the nearest neighbor-based merge decisions suggested by the original representation and the self-supervised one. The moment at which consensus maximization occurs can serve as a criterion for determining the optimal number of clusters. Extensive experiments on multiple datasets demonstrate that the proposed framework outperforms existing clustering approaches designed for scenarios with an unknown number of clusters.

Code — https://github.com/ljz441/2026-AAAI-SCMax
Extended Version — https://arxiv.org/abs/2511.09211

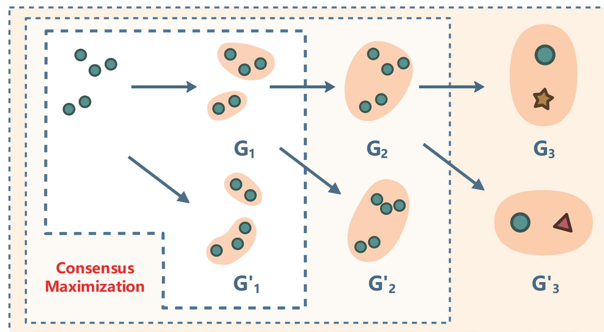

Figure 1: Motivation of Consensus Maximization. Here, $\mathbf{G}_i$ and $\mathbf{G'}_i$ denote the cluster structures from original and self-supervised representations, respectively. The optimal structure is determined when their consensus is maximized. For simplicity, each type of shape in $\mathbf{G}_3$ and $\mathbf{G'}_3$ is represented as a class set.

Introduction

Clustering is a fundamental machine learning task that aims to partition unlabeled data into groups based on intrinsic similarities (Zhang et al. 2025a; Zhou et al. 2025; Liu et al. 2024; Yu et al. 2024; Qin and Qian 2024; Wang et al. 2024). As a typical unsupervised learning method, clustering has been widely applied in various domains such as image processing (Qiu et al. 2024; Zhang et al. 2024), and text analysis (He, Wang, and Zhang 2025; Yang et al. 2025), serving as a crucial tool for understanding data structures.

However, existing clustering methods often heavily rely on hyperparameters such as the number of clusters or other sensitive settings, which poses significant challenges when dealing with tasks involving open and complex data structures. Unlike supervised learning, clustering lacks the guidance of class labels and, in real-world applications, it is generally infeasible to preset any priors. This reality has driven the emergence of parameter-free clustering techniques. In recent years, numerous clustering methods that do not rely on predefined cluster numbers or other sensitive settings have been proposed, including density-based (Abbas, El-Zoghabi, and Shoukry 2021; Rodriguez and Laio 2014), graph-based (Sun et al. 2024; Peng et al. 2019; Shah and Koltun 2017), hierarchy-based (Yang and Lin 2025; Dang et al. 2021; Sarfraz, Sharma, and Stiefelhagen 2019), and probabilistic model-based (Wang et al. 2022; Yang et al. 2019) approaches. Among them, hierarchy-based methods have become a prominent research direction due to their ability to naturally generate multi-level cluster structures during the clustering process, thereby producing multiple candidate values of K with strong interpretability.

Although current parameter-free hierarchical clustering methods attempt to address the core issue of K-value se-

[*]These authors contributed equally.
[†]Corresponding authors.

lection, they do not entirely eliminate the reliance on priors. Many such methods, while not explicitly specifying K, introduce other sensitive parameter settings, such as distance thresholds or probability cutoffs, which still require manual tuning and can affect the stability and generalizability of the clustering results. Overall, these methods can typically be divided into two stages: cluster number generation and cluster structure evaluation. In the generation stage, common strategies often adopt split-and-merge techniques (Ronen, Finder, and Freifeld 2022; Liu et al. 2023; Dai et al. 2024). However, these approaches usually imply assumptions about the initial number of clusters and can only explore a limited set of K values in practice, making it difficult to cover all potential clustering structures. In the evaluation stage, common techniques such as the elbow method (Shi et al. 2021; Schubert 2023) often rely on smoothed or ambiguous evaluation curves that require manually defined thresholds or subjective identification of "elbow points," which can be unstable in high-dimensional or complex data scenarios. Therefore, how to automatically generate and evaluate cluster structures without any prior knowledge or manual settings remains a key challenge in achieving truly parameter-free clustering.

To address the above issues, we propose a fully parameter-free clustering framework via **S**elf-supervised **C**onsensus **Max**imization, named SCMax. Falling under the category of hierarchical clustering, SCMax achieves the complete process from cluster number generation to cluster structure evaluation without relying on any prior settings. Specifically, in the generation stage, SCMax constructs nearest neighbor graphs to guide the cluster merging process and automatically produces a set of candidate cluster structures. Unlike existing methods, SCMax does not assume an initial number of clusters or restrict the search space, enabling the direct and efficient generation of multiple candidates and significantly reducing the K-value search space. Furthermore, based on the principle of nearest neighbor stability, we design a Nearest Neighbor Consensus (NNC) evaluation metric to measure the agreement of merging decisions between the original and self-supervised feature representations. The optimal number of clusters is automatically determined at the point of consensus maximization. Compared to traditional methods, this metric avoids reliance on subjective parameters such as thresholds or elbow points. The entire process is fully automated, achieves strong generalization and robustness, and embodies the essence of "parameter-free" clustering. The motivation is illustrated in Fig. 1. The main contributions of this work are summarized as follows.

- We propose a fully parameter-free clustering framework that performs hierarchical agglomerative clustering and cluster evaluation in a single integrated process. This framework does not require any hyperparameters of the number of clusters or other sensitive settings.
- We design a cluster evaluation metric based on nearest neighbor consensus, which measures the agreement between the nearest neighbor-based merge decisions derived from the original and self-supervised representations. The moment at which consensus maximization occurs can serve as a criterion for determining the optimal number of clusters.
- Extensive experiments on multiple datasets demonstrate that the proposed framework outperforms existing clustering approaches designed for scenarios with an unknown number of clusters.

Related Work

This section focuses on the branch of hierarchical clustering under the umbrella of parameter-free clustering, with particular emphasis on two key aspects: cluster number generation and cluster structure evaluation.

Cluster Number Generation

Within the framework of parameter-free clustering, traditional hierarchical clustering methods mainly adopt two types of cluster number generation strategies: bottom-up and top-down. The former, such as agglomerative clustering (Li et al. 2024; Xing et al. 2021), starts with a larger number of clusters and progressively merges the most similar ones; the latter, such as divisive clustering (Bagirov, Aliguliyev, and Sultanova 2023; Liu et al. 2023; Dai et al. 2024), begins with fewer clusters and recursively splits them into smaller subclusters. In addition, some methods combine the strengths of both approaches, such as split-and-merge strategy (Zhao, Yang, and Deng 2024; Xiao et al. 2023; Ronen, Finder, and Freifeld 2022), which dynamically decides whether to split or merge clusters based on an initial assumption of the number of clusters. While these methods are capable of constructing a hierarchical structure of clustering results, they still face several practical challenges. On one hand, many of them heavily rely on the initial number of clusters. If this initial value deviates significantly from the true number of clusters, it may lead to inefficient clustering or misguide the subsequent structural exploration, resulting in poor candidate cluster structures. On the other hand, most approaches require a predefined search space for the number of clusters in order to avoid the high computational cost of brute-force enumeration, which inevitably limits their flexibility and generalization ability.

To address these challenges, Nearest Neighbor Clustering (NN clustering) methods have attracted increasing attention in recent years (Yang and Lin 2025; Dang et al. 2021; Sarfraz, Sharma, and Stiefelhagen 2019). Compared to traditional strategies, NN clustering offers superior computational efficiency, with complexity reduced to $O(N \log N)$. Without the need to assume an initial number of clusters or constrain the search space, NN clustering can directly and efficiently generate multiple candidate cluster structures, significantly reducing the K-value search space. These methods not only improve runtime performance and clustering quality but also scale well to large datasets, demonstrating broad applicability in hierarchical clustering tasks.

Cluster Structure Evaluation

In hierarchical clustering frameworks, the core challenge of parameter-free clustering lies in how to automatically identify the optimal partition from a set of hierarchical candidate cluster structures, which is crucial for ensuring cluster-

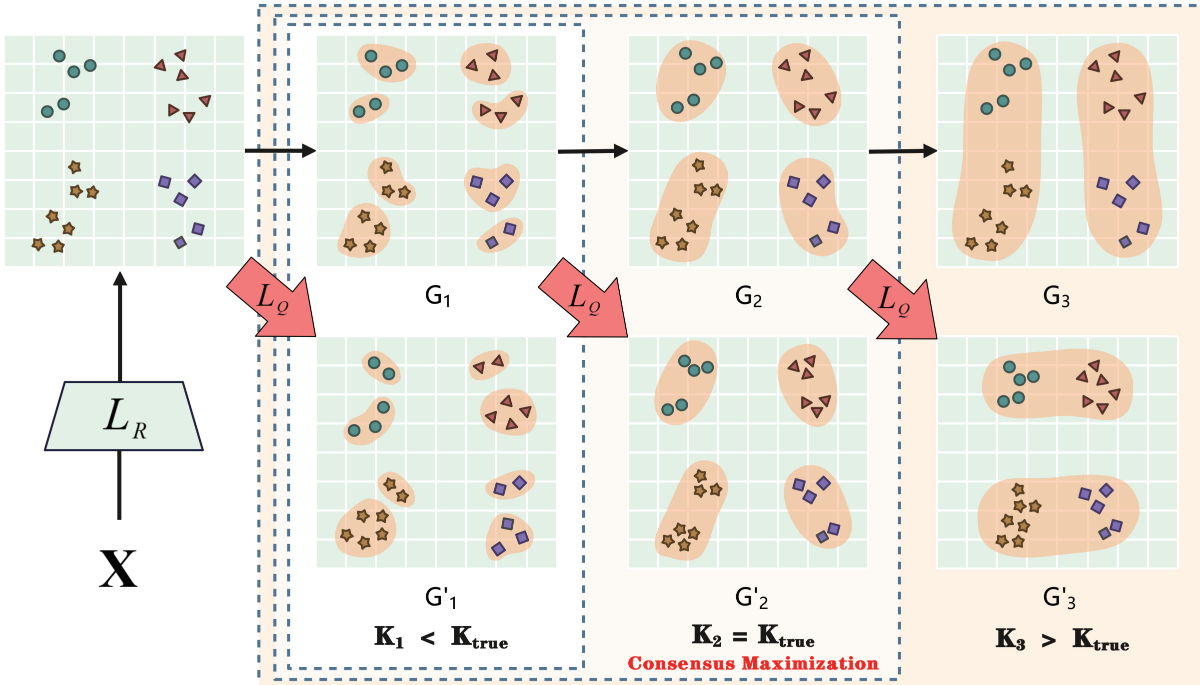

Figure 2: The proposed SCMax framework.

ing performance. Traditional approaches such as the Elbow Method (Kodinariya and Makwana 2013; Shi et al. 2021; Schubert 2023) typically rely on manually set thresholds or visually identified inflection points to determine when to stop the clustering process. However, such methods are often highly sensitive to subjective settings, lack adaptive capabilities, and are vulnerable to noise, which can lead to unstable and unreliable evaluation results.

To overcome these limitations, we observe a phenomenon referred to as nearest-neighbor stability in the actual feature representation space and propose a nearest-neighbor consensus-based cluster structure evaluation metric. This metric measures the consistency of nearest-neighbor merging decisions between the original features and those obtained via self-supervised learning. The moment when this consensus maximization occurs corresponds to the optimal cluster structure. This metric is entirely free of manually set hyperparameters and can adaptively capture the intrinsic structure of the data, enabling truly parameter-free, stable, and reliable cluster structure evaluation.

Methodology

In this section, we propose a fully parameter-free clustering framework termed SCMax. The overall framework is illustrated in Fig. 2.

Autoencoder Module

Given a dataset $\mathbf{X} \in \mathbb{R}^{N \times D}$, where N denotes the number of samples and D denotes the feature dimension. For the original features $\mathbf{X}$, there may exist redundancy and noise. Therefore, we use an autoencoder to nonlinearly map $\mathbf{X}$ into a customizable feature space to extract more representative feature embeddings. Specifically, we denote the encoder and decoder as $E(\mathbf{X}, \theta)$ and $D(\mathbf{X}, \phi)$, where θ and ϕ are network parameters. Based on this, the mapped L-dimensional latent feature representation is $\mathbf{Z} = E(\mathbf{X}) \in \mathbb{R}^{N \times L}$, and the reconstructed representation is $\hat{\mathbf{X}} = D(E(\mathbf{X}))$. The reconstruction loss between input $\mathbf{X}$ and output $\hat{\mathbf{X}}$ is denoted as L_R. Thus, the autoencoder's loss is formulated as:

$$L_R = \sum_{i=1}^{N} \|\mathbf{X}_i - \hat{\mathbf{X}}_i\|_2^2 \, . \tag{1}$$

Cluster Number Generation Module

Based on the dimensionally reduced feature representation $\mathbf{Z}$, we begin cluster number generation. In SCMax, we adopt nearest neighbor merging to generate cluster candidates due to its efficiency, as it aggregates data using only local neighbor relations without constructing a global distance matrix. The resulting partitions have been shown to closely match true clusters and perform well across various tasks (Sarfraz, Sharma, and Stiefelhagen 2019; Dang et al. 2021; Yang and Lin 2025). Specifically, each sample is initially treated as an individual cluster, forming the first-level cluster structure. We then recursively merge these clusters, where at each step, each cluster performs merging operations guided by the nearest neighbor graph, thereby generating a new cluster structure. To efficiently construct the nearest neighbor graph, we employ an approximate nearest neighbor search method (i.e., KD-Tree). Each cluster is represented by the mean of its sample vectors, and the nearest class neighbors are computed based on these mean vectors. Unlike sample-level nearest neighbor merging, this approach performs merging at the class level, which simplifies computation and maintains a complexity of $O(K \log K)$, where K denotes the number of clusters. We denote the nearest neighbor graph as $\mathbf{A} \in \mathbb{R}^{K \times 1}$. Given the integer indices $\mathbf{A}$ representing the nearest neighbor of each class, we can construct a symmetric adjacency matrix $\mathbf{G} \in \mathbb{R}^{K \times K}$ in linear time as follows:

$$\mathbf{G}(i,j) = \begin{cases} 1, & \text{if } j = \mathbf{A}_i \text{ or } \mathbf{A}_j = i \text{ or } \mathbf{A}_i = \mathbf{A}_j \\ 0, & \text{otherwise} \end{cases} , \quad (2)$$

where $\mathbf{A}_i$ and $\mathbf{A}_j$ denote the nearest neighbors of classes i and j, respectively. The adjacency matrix $\mathbf{G}$ captures the inter-class relationships, and its connected components correspond to the resulting clusters. Since $\mathbf{G}$ describes class-level relations, a reverse label mapping is required to propagate the clustering results back to the sample level, thereby obtaining the final cluster label vector $\mathbf{Q} \in \mathbb{R}^{N \times 1}$. This clustering formulation is extremely simple and parameter-free, directly producing a set of candidate cluster structures from the data.

Cluster Structure Evaluation Module

After obtaining multiple hierarchical candidate cluster structures $\mathbf{G}$, the core problem of the evaluation module is how to select the optimal clustering structure from these candidates. Before formally introducing the cluster structure evaluation module in SCMax, we first need to introduce the core concept of "nearest neighbor stability." Specifically, when perturbations are applied to the representation $\mathbf{Z}$, the adjacency relationships of classes in the nearest neighbor graph may change. We define the degree of change in nearest neighbor graph structure before and after perturbation as the classes' nearest neighbor stability, which indicates the robustness of the local structure to perturbations.

Next, we provide the definition and implementation of the perturbation. In SCMax, perturbations are not based on random noise but are implemented by introducing label $\mathbf{Q}$-based contrastive constraints. These constraints impose structural interventions on the fixed feature representation $\mathbf{Z}$, indirectly altering the adjacency relationships among classes to simulate perturbations. Under the contrastive learning constraint, we aim to pull together samples of the same class while pushing apart samples from different classes. Hence, we define the following structural contrastive loss:

$$L_Q = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \mathbf{D}_{ij} - \frac{1}{|\mathcal{N}|} \sum_{(i,j)\in\mathcal{N}} \mathbf{D}_{ij} , \quad (3)$$

where $\mathcal{P} = \{(i,j) \mid q_i = q_j,\ i \neq j\}$ denote the set of positive (same-class) sample pairs, $\mathcal{N} = \{(i,j) \mid q_i \neq q_j\}$ denote the set of negative (different-class) sample pairs, and $\mathbf{D}_{ij}$ represents the Euclidean distance between z_i and z_j, where $\mathbf{D} \in \mathbb{R}^{B \times B}$ and B is the batch size. To avoid $O(N^2)$ complexity in computation, SCMax does not perform full-sample contrastive learning but adopts a mini-batch training strategy. That is, at each training step, a local Euclidean distance matrix is constructed within the current batch, significantly reducing computation and memory costs. This loss function uses label $\mathbf{Q}$ as supervision to guide the fixed representation $\mathbf{Z}$ to undergo structured changes in semantic space, achieving the perturbation goal. Unlike random noise perturbations, this method's intervention in local structure is directional and controllable, effectively influencing nearest neighbor stability and providing a clean and explicit baseline for subsequent cluster structure evaluation.

At this point, after perturbing the fixed representation $\mathbf{Z}$ under the guidance of cluster labels $\mathbf{Q}$, we obtain a set of new self-supervised representations $\mathbf{Z}' = \{\mathbf{Z}'_1, \mathbf{Z}'_2, ..., \mathbf{Z}'_i\}$, where each $\mathbf{Z}'_i$ corresponds to the perturbation guided by $\mathbf{Q}_i$. Based on the fixed representation $\mathbf{Z}$, we find the nearest neighbors within each class in $\mathbf{G}_i$ and perform merging to get the next cluster structure $\mathbf{G}_{i+1}$. Similarly, based on the self-supervised representation $\mathbf{Z}'_i$, we find the nearest neighbors in $\mathbf{G}_i$ and merge to obtain a new cluster structure $\mathbf{G}'_{i+1}$. So far, we have two sets of cluster structures: $\mathbf{G} = \{\mathbf{G}_1, \mathbf{G}_2, ..., \mathbf{G}_i\}$ from the fixed representation $\mathbf{Z}$, and $\mathbf{G}' = \{\mathbf{G}'_2, ..., \mathbf{G}'_i\}$ from the self-supervised representation $\mathbf{Z}'$. By measuring the consistency between the merging decisions of nearest neighbors executed on the original representation $\mathbf{Z}$ and the self-supervised representation $\mathbf{Z}'$—that is, the similarity between cluster structures $\mathbf{G}_i$ and $\mathbf{G}'_i$—we define the Nearest Neighbor Consensus (NNC) metric as follows:

$$\mathrm{NNC}(\mathbf{G}_i, \mathbf{G}'_i) = \max_{\pi\in\mathcal{S}} \frac{1}{N} \sum_{j=1}^{N} \mathbf{1}\left(\mathbf{Q}_i(j) = \pi(\mathbf{Q}'_i(j))\right) , \quad (4)$$

where $\mathbf{Q}_i$ and $\mathbf{Q}'_i$ denote the cluster assignment labels for each sample in the clusters $\mathbf{G}_i$ and $\mathbf{G}'_i$, respectively; the indicator function $\mathbf{1}(\cdot)$ returns 1 if the condition is true and 0 otherwise; π represents the set $\mathcal{S}$ of all possible label mappings under the Hungarian algorithm for label alignment.

Finally, we provide an in-depth discussion and analysis of the NNC metric, explaining why the moment when consensus reaches its maximum corresponds to the optimal clustering structure. The detailed process is illustrated in the Fig. 2.

- **When the cluster number K_i corresponding to $\mathbf{G}_i$ is greater than the true cluster number**: perturbations applied on the fixed representation Z at the previous step significantly affect the nearest neighbor relationships among intra-class samples of the true cluster structure. Since samples in true clusters are close to each other, even slight perturbations can break the original adjacency relations, indicating unstable nearest neighbor relations at K_{i-1}. Consequently, the nearest neighbor graph constructed from the perturbed self-supervised representation $\mathbf{Z}'$ significantly differs from that constructed from $\mathbf{Z}$, resulting in low similarity between $\mathbf{G}_i$ and $\mathbf{G}'_i$.
- **When K_i equals the true cluster number**: perturbations applied on $\mathbf{Z}$ at the previous step do not significantly affect intra-class adjacency. At this point, K_{i-1} approaches the true cluster number, with most intra-class samples merged into the same cluster, greatly reducing the number of intra-class nearest neighbors and leading to a stable adjacency relationships. Therefore, the difference between perturbed and original nearest neighbor

Algorithm 1: SCMax

Input: Dataset $\mathbf{X} \in \mathbb{R}^{N \times D}$
Output: Cluster number K^*, cluster labels $\mathbf{Q}^*$
Compute latent representation $\mathbf{Z} \in \mathbb{R}^{N \times L}$ by Eq.(1)
Initialize each sample as a singleton cluster
Get $\mathbf{G}_1$ via nearest neighbor merging on $\mathbf{Z}$ by Eq.(2)
Derive K_1, $\mathbf{Q}_1$ from connected components in $\mathbf{G}_1$
Let $i = 1$
while at least three clusters exist in $\mathbf{G}_i$ **do**
    Update $\mathbf{G}_{i+1}$ via nearest neighbor merging based on $\mathbf{Z}$ by Eq.(2)
    Obtain K_{i+1} and $\mathbf{Q}_{i+1}$ by computing connected components in $\mathbf{G}_{i+1}$
    Generate perturbed representation $\mathbf{Z}'$ based on $\mathbf{Q}_i$ by Eq.(3)
    Compute $\mathbf{G}'_{i+1}$ via nearest neighbor merging based on $\mathbf{Z}'$ by Eq.(2)
    Compute NNC score between $\mathbf{G}_{i+1}$ and $\mathbf{G}'_{i+1}$ by Eq.(4)
    Record K_{i+1}, $\mathbf{Q}_{i+1}$ and corresponding NNC score
    $i \leftarrow i + 1$
end while
Select K^*, $\mathbf{Q}^*$ with the highest NNC score
return K^*, $\mathbf{Q}^*$

graphs is minimal, and the similarity between $\mathbf{G}_i$ and $\mathbf{G}'_i$ reaches its maximum.

- **When K_i is less than the true cluster number**: perturbations applied on $\mathbf{Z}$ disrupt the inter-class boundary structure of the true cluster. Since true clusters usually have certain gaps, perturbations blur or overlap these gaps, causing samples from different true clusters to be incorrectly grouped together. This breaks the original inter-class boundaries, making nearest neighbor relations unstable again at K_{i-1}. As a result, the difference between perturbed and original nearest neighbor graphs becomes significant again, and the similarity between $\mathbf{G}_i$ and $\mathbf{G}'_i$ decreases.

In summary, when the nearest neighbor consensus reaches its maximum, the corresponding cluster structure $\mathbf{G}^*$ precisely reflects the optimal cluster distribution of the data, and the corresponding cluster number K^* and label set $\mathbf{Q}^*$ can be regarded as the final clustering result. It can be seen that the proposed NNC metric does not depend on any prior hyperparameters and can adaptively evaluate the intrinsic structure of the data.

Computational Complexity Analysis

The pseudo-code of SCMax is presented in Algorithm 1. This subsection focuses on analyzing the main computational and memory overheads. Regarding time complexity, constructing the class-level nearest neighbor index using a KD-Tree for K classes requires $O(K \log K)$ time. The symmetric adjacency matrix $\mathbf{G}$ is then constructed in linear time, $O(K)$. To assess local consistency, SCMax computes the Euclidean distance matrix within each mini-batch of size B, which incurs a cost of $O(B^2)$. During the computation of the NNC score, the Hungarian matching algorithm introduces an additional complexity of $O(K^3)$. Therefore, the total time complexity is $O(K \log K + K + B^2 + K^3)$. As for space complexity, the nearest neighbor index requires $O(K)$ space, the adjacency matrix occupies $O(K^2)$, and the distance matrix within each mini-batch takes $O(B^2)$. In addition, constructing the confusion matrix for computing the NNC score requires $O(K^2)$ space. Consequently, the overall space complexity is $O(K + 2K^2 + B^2)$.

Dataset	Samples	Dimension	Cluster
MSRCv1	210	1302	7
HW2	2000	784	10
Wiki	2866	10	10
MNIST	5000	784	10
Cifar10	50000	1024	10
Cifar100	50000	512	100
Fashion	60000	1280	10
YTF20	63896	512	20

Table 1: Datasets description.

Experiments

Experimental Setup

Regarding the datasets, we employ eight single-view datasets: MSRCv1[1], HW2[2], Wiki[3], MNIST[4], Cifar10[5], Cifar100[6], Fashion (Xiao, Rasul, and Vollgraf 2017), YTF20[7], as shown in Table 1. Regarding the comparison methods, since the proposed algorithm addresses the issue of K-value selection in parameter-free clustering, we select nine Non-K clustering methods: FINCH (Sarfraz, Sharma, and Stiefelhagen 2019), COMIC (Peng et al. 2019), BP (Averbuch-Elor, Bar, and Cohen-Or 2020), DenMune (Abbas, El-Zoghabi, and Shoukry 2021), DeepDPM (Ronen, Finder, and Freifeld 2022), MPAASL (Dai et al. 2024), TC (Yang and Lin 2025), Gauging-δ (Yao, Pan, and Zeng 2025) and AFCL (Zhang et al. 2025b). Moreover, we also selected two Given-K methods for reference comparison: DCGL (Chen, Wang, and Li 2024) and DMAC (Wang et al. 2025). Finally, we adopt four metrics to evaluate clustering quality: Accuracy (ACC), Normalized Mutual Information (NMI), Purity (PUR), and F-score. More information about the datasets, comparison methods, and implementation details is provided in the Extended Version.

Clustering Performance Comparison

As shown in Table 2, 3, SCMax demonstrates superior clustering performance. Compared with Non-K clustering

[1] https://www.microsoft.com/en-us/research/project/image-understanding
[2] https://cs.nyu.edu/roweis/data.html
[3] http://www.svcl.ucsd.edu/projects/crossmodal/
[4] http://yann.lecun.com/exdb/mnist/
[5] http://www.cs.toronto.edu/~kriz/cifar.html
[6] http://www.cs.toronto.edu/~kriz/cifar.html
[7] https://www.cs.tau.ac.il/~wolf/ytfaces/

Datasets	FINCH CVPR'19	COMIC ICML'19	BP TPAMI'20	DenMune PR'21	DeepDPM CVPR'22	MPAASL ICML'24	TC TPAMI'25	Gauging-δ TPAMI'25	AFCL AAAI'25	SCMax
					ACC					
MSRCv1	0.2524	0.1667	0.2762	0.4048	0.1429	0.4476	0.3048	0.2333	0.0310	**0.6238**
HW2	0.1975	0.1045	0.1410	0.2005	0.1000	0.5980	0.4725	0.1000	0.0005	**0.6115**
Wiki	0.5286	0.1207	0.4833	0.1019	0.1574	0.5056	0.4682	0.1574	0.0052	**0.5967**
MNIST	0.1968	0.1028	0.5004	0.0910	0.1000	0.5946	0.4752	0.1000	0.1000	**0.6004**
Cifar10	0.7117	OOM	OOM	0.0193	0.8178	OOM	OOM	OOM	OOM	**0.8703**
Cifar100	0.5999	OOM	OOM	0.0742	0.3100	OOM	OOM	OOM	OOM	**0.9061**
Fashion	0.2933	OOM	OOM	0.0183	0.3652	OOM	OOM	OOM	OOM	**0.7542**
YTF20	0.1238	OOM	OOM	0.0253	0.4890	OOM	OOM	OOM	OOM	**0.6493**
					NMI					
MSRCv1	0.2185	0.0454	0.1335	0.5479	0.0000	**0.5888**	0.5567	0.0986	0.2936	0.5597
HW2	0.3480	0.0128	0.0731	0.5616	0.0000	**0.7459**	0.5998	0.0000	0.2198	0.6436
Wiki	0.5448	0.4654	0.5209	0.4151	0.0000	0.5322	0.5284	0.0000	0.0061	**0.5609**
MNIST	0.3562	0.0109	0.6005	0.4946	0.0000	**0.7385**	0.6321	0.0000	0.0000	0.7021
Cifar10	0.7262	OOM	OOM	0.3707	0.7474	OOM	OOM	OOM	OOM	**0.7714**
Cifar100	0.8246	OOM	OOM	0.6712	0.8368	OOM	OOM	OOM	OOM	**0.9801**
Fashion	0.4830	OOM	OOM	0.3587	0.6511	OOM	OOM	OOM	OOM	**0.7479**
YTF20	0.0503	OOM	OOM	0.4490	**0.8126**	OOM	OOM	OOM	OOM	0.8065
					PUR					
MSRCv1	0.2524	0.1667	0.2810	0.6762	0.1429	0.7762	**0.7905**	0.2333	0.5262	0.6476
HW2	0.1975	0.1065	0.1410	**0.8625**	0.1000	0.6960	0.7785	0.1000	0.2295	0.6115
Wiki	0.6350	**0.9337**	0.6308	0.6933	0.1574	0.6497	0.6643	0.1574	0.2102	0.6403
MNIST	0.1968	0.1052	0.5004	0.8708	0.1000	**0.8738**	0.8228	0.1000	0.2178	0.7658
Cifar10	0.7117	OOM	OOM	0.8361	0.8376	OOM	OOM	OOM	OOM	**0.8922**
Cifar100	0.5999	OOM	OOM	0.9055	0.3100	OOM	OOM	OOM	OOM	**0.9318**
Fashion	0.2933	OOM	OOM	0.8161	0.8358	OOM	OOM	OOM	OOM	**0.8497**
YTF20	0.1238	OOM	OOM	0.8345	**0.9711**	OOM	OOM	OOM	OOM	0.8071
					F-score					
MSRCv1	0.1593	0.0803	0.2029	0.2045	0.0357	0.2066	0.0797	0.1322	0.4419	**0.5698**
HW2	0.0738	0.0247	0.0799	0.0234	0.0182	0.5175	0.2430	0.0182	0.2695	**0.5538**
Wiki	0.3957	0.0011	0.3574	0.0069	0.0272	0.2895	0.2534	0.0272	0.1822	**0.5538**
MNIST	0.0741	0.0121	0.4255	0.0035	0.0182	0.3603	0.2318	0.0182	0.1540	**0.4553**
Cifar10	0.6484	OOM	OOM	0.0001	0.7933	OOM	OOM	OOM	OOM	**0.8001**
Cifar100	0.5778	OOM	OOM	0.0029	0.1691	OOM	OOM	OOM	OOM	**0.8476**
Fashion	0.1854	OOM	OOM	0.0000	0.1550	OOM	OOM	OOM	OOM	**0.6187**
YTF20	0.0332	OOM	OOM	0.0001	0.1543	OOM	OOM	OOM	OOM	**0.5671**

Table 2: Non-K methods clustering performance comparison results, where the best results are highlighted in bold and OOM denotes out of memory error.

methods, it achieves the highest scores in both ACC and F-score across all datasets, indicating its strong capability to uncover the true underlying cluster structures. For the NMI metric, SCMax also exhibits robust performance, significantly outperforming most competitors. Although it does not always achieve the best NMI on a few datasets such as MSRCv1, its results remain very close to the top-performing method, confirming its adaptability and effectiveness across diverse application scenarios. Meanwhile, for the PUR metric—which often favors over-segmentation—SCMax still delivers competitive results, reflecting its balanced performance across multiple evaluation perspectives. Compared with Given-K clustering methods, it is important to note that such a comparison is inherently unfair: Given-K methods operate with the prior knowledge of the cluster number K, whereas Non-K methods must simultaneously perform cluster number estimation and clustering optimization, incurring additional K-value selection overhead. Even under this unfair comparison, our method still achieves highly competitive results and even attains the best performance on several datasets, including Wiki, Cifar10, Cifar100, Fashion, and YTF20. Overall, these results strongly validate the effectiveness and robustness of SCMax in practical tasks.

Due to the page limitation, the comparisons of estimated cluster number, running time and computational complexity for all methods are provided in the Appendix.

Ablation Study

To evaluate the effectiveness and design rationale of each component in SCMax, we conducted ablation studies. Table 4 reports the ACC scores on four representative datasets, complete results are provided in the Extended Version. We

Methods	MSRCv1	HW2	Wiki	MNIST	Cifar10	Cifar100	Fashion	YTF20
				ACC				
DCGL (AAAI'24)	**0.6810**	**0.6185**	0.5534	**0.6246**	OOM	OOM	OOM	OOM
DMAC (AAAI'25)	0.4905	0.5285	0.1574	0.1000	OOM	OOM	OOM	OOM
SCMax	0.6238	0.6115	**0.5967**	0.6004	**0.8703**	**0.9061**	**0.7542**	**0.6493**
				NMI				
DCGL (AAAI'24)	**0.6143**	**0.6804**	0.5356	0.6621	OOM	OOM	OOM	OOM
DMAC (AAAI'25)	0.4841	0.4708	0.0000	0.0000	OOM	OOM	OOM	OOM
SCMax	0.5597	0.6436	**0.5609**	**0.7021**	**0.7714**	**0.9801**	**0.7479**	**0.8065**
				PUR				
DCGL (AAAI'24)	**0.7000**	**0.6940**	0.6179	0.6712	OOM	OOM	OOM	OOM
DMAC (AAAI'25)	0.5381	0.5425	0.1574	0.1000	OOM	OOM	OOM	OOM
SCMax	0.6476	0.6115	**0.6403**	**0.7658**	**0.8922**	**0.9318**	**0.8497**	**0.8071**
				F-score				
DCGL (AAAI'24)	**0.6771**	0.5322	0.5428	**0.6206**	OOM	OOM	OOM	OOM
DMAC (AAAI'25)	0.3776	0.5349	0.0272	0.0182	OOM	OOM	OOM	OOM
SCMax	0.5698	**0.5538**	**0.5538**	0.4553	**0.8001**	**0.8476**	**0.6187**	**0.5671**

Table 3: Given-K methods clustering performance comparison results, where the best results are highlighted in bold and OOM denotes out of memory error.

Datasets	SCMax	Remove CL - I	Remove CL - II	Select Random Noise	Choose $\mathbf{G}'$ as result
MSRCv1	**0.6238**	0.6238	0.6238	0.1429	0.4952
MNIST	**0.6004**	0.3158	0.3902	0.6004	0.5596
Fashion	**0.7542**	0.1980	0.3913	0.7542	0.7194
YTF20	**0.6493**	0.2071	0.6493	0.3421	0.6455

Table 4: The ablation experiment results on four representative datasets are presented, only showing the ACC metric results. The best result is displayed in bold. The complete results can be found in the Extended Version.

designed four ablation settings:

- Remove CL-I: Removing the contrastive learning constraint Part I, retaining only the perturbation method based on pushing apart negative (different-class) samples.
- Remove CL-II: Removing the contrastive learning constraint Part II, retaining only the perturbation method based on pulling together positive (same-class) samples.
- Select Random Noise: Replacing perturbation with random noise sampled from $[-1, 1]$, whose standard deviation matches that of the fixed representation $\mathbf{Z}$.
- Choose $\mathbf{G}'$ as Result: Selecting the cluster structure $\mathbf{G}'$ at the moment of consensus maximization as the final result.

Experimental results show that removing or replacing any component leads to performance degradation, validating the necessity of each design. Notably, the comparison between "Remove CL-I" and "Remove CL-II" indicates that perturbations based on pushing apart negative samples have a more pronounced impact, suggesting that negative pair repulsion plays a more critical role in generating effective perturbations. The "Select Random Noise" setting leads to unstable and highly variable results, revealing the limitations of perturbations without structural constraints. Moreover, the "Choose $\mathbf{G}'$ as Result" setting consistently yields inferior performance compared to choosing $\mathbf{G}$, further indicating that the perturbed candidate cluster structures are less likely to produce optimal clustering results.

Analysis

In this subsection, we analyze SCMax from two perspectives: **the NNC score** and **loss convergence**. The NNC score effectively evaluates the latent cluster structure, reaching its maximum exactly at the closest ground-truth number of clusters, while SCMax exhibits stable and efficient training, with both the autoencoder and contrastive losses showing steady convergence. For details, see the Extended Version.

Conclusion

The proposed SCMax framework addresses the true parameter-free clustering problem. By integrating hierarchical clustering, self-supervised representation learning, and nearest-neighbor-based consensus evaluation within a unified process, SCMax not only dynamically guides the generation of clustering number but also automatically evaluates the optimal clustering structure. Extensive experimental results demonstrate the superior performance of SCMax across multiple datasets. In future work, we aim to explore more advanced cluster number generation strategies to further enhance the accuracy and robustness of the resulting clustering structures. Furthermore, we plan to eliminate the reliance of the autoencoder-based representation learning module on predefined architectural structures.

Acknowledgements

This work is supported by the National Science Fund for Distinguished Young Scholars of China (No. 62325604), and the National Natural Science Foundation of China (No. 62276271 and 62376039).

References

Abbas, M.; El-Zoghabi, A.; and Shoukry, A. 2021. DenMune: Density peak based clustering using mutual nearest neighbors. *Pattern Recognition*, 109: 107589.

Averbuch-Elor, H.; Bar, N.; and Cohen-Or, D. 2020. Border-Peeling Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(07): 1791–1797.

Bagirov, A. M.; Aliguliyev, R. M.; and Sultanova, N. 2023. Finding compact and well-separated clusters: Clustering using silhouette coefficients. *Pattern Recognition*, 135: 109144.

Chen, M.; Wang, B.; and Li, X. 2024. Deep contrastive graph learning with clustering-oriented guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 11364–11372.

Dai, H.; Liu, Y.; Su, P.; Cai, H.; Huang, S.; and Lv, J. 2024. Multi-view clustering by inter-cluster connectivity guided reward. In *International Conference on Machine Learning*.

Dang, Z.; Deng, C.; Yang, X.; Wei, K.; and Huang, H. 2021. Nearest neighbor matching for deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13693–13702.

He, W.; Wang, Z.; and Zhang, Y. 2025. Target Semantics Clustering via Text Representations for Robust Universal Domain Adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 17132–17140.

Kodinariya, T.; and Makwana, P. 2013. Review on Determining of Cluster in K-means Clustering. *International Journal of Advance Research in Computer Science and Management Studies*, 1: 90–95.

Li, D.; Zhou, S.; Zeng, T.; and Chan, R. H. 2024. Multi-Prototypes Convex Merging Based K-Means Clustering Algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 36(11): 6653–6666.

Liu, S.; Zhang, J.; Wen, Y.; Yang, X.; Wang, S.; Zhang, Y.; Zhu, E.; Tang, C.; Zhao, L.; and Liu, X. 2024. Sample-level cross-view similarity learning for incomplete multi-view clustering. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 38, 14017–14025.

Liu, Y.; Liang, K.; Xia, J.; Yang, X.; Zhou, S.; Liu, M.; Liu, X.; and Li, S. Z. 2023. Reinforcement graph clustering with unknown cluster number. In *Proceedings of the ACM International Conference on Multimedia*, 3528–3537.

Peng, X.; Huang, Z.; Lv, J.; Zhu, H.; and Zhou, J. T. 2019. COMIC: Multi-view clustering without parameter selection. In *International Conference on Machine Learning*, 5092–5101. PMLR.

Qin, Y.; and Qian, L. 2024. Fast elastic-net multi-view clustering: a geometric interpretation perspective. In *Proceedings of the 32nd ACM international conference on multimedia*, 10164–10172.

Qiu, L.; Zhang, Q.; Chen, X.; and Cai, S. 2024. Multi-level cross-modal alignment for image clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 14695–14703.

Rodriguez, A.; and Laio, A. 2014. Clustering by fast search and find of density peaks. *science*, 344(6191): 1492–1496.

Ronen, M.; Finder, S. E.; and Freifeld, O. 2022. Deepdpm: Deep clustering with an unknown number of clusters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9861–9870.

Sarfraz, S.; Sharma, V.; and Stiefelhagen, R. 2019. Efficient parameter-free clustering using first neighbor relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8934–8943.

Schubert, E. 2023. Stop using the elbow criterion for k-means and how to choose the number of clusters instead. *ACM SIGKDD Explorations Newsletter*, 25(1): 36–42.

Shah, S. A.; and Koltun, V. 2017. Robust continuous clustering. *Proceedings of the National Academy of Sciences*, 114(37): 9814–9819.

Shi, C.; Wei, B.; Wei, S.; Wang, W.; Liu, H.; and Liu, J. 2021. A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm. *EURASIP Journal on Wireless Communications and Networking*, 2021(1): 31.

Sun, L.; Huang, Z.; Peng, H.; Wang, Y.; Liu, C.; and Yu, P. S. 2024. LSEnet: Lorentz structural entropy neural network for deep graph clustering. In *International Conference on Machine Learning*, 47078–47104.

Wang, B.; Zeng, C.; Chen, M.; and Li, X. 2025. Towards Learnable Anchor for Deep Multi-View Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 21044–21052.

Wang, F.; Jin, J.; Hu, J.; Liu, S.; Yang, X.; Wang, S.; Liu, X.; and Zhu, E. 2024. Evaluate then cooperate: Shapley-based view cooperation enhancement for multi-view clustering. *Advances in Neural Information Processing Systems*, 37: 135355–135379.

Wang, Z.; Ni, Y.; Jing, B.; Wang, D.; Zhang, H.; and Xing, E. 2022. DNB: A Joint Learning Framework for Deep Bayesian Nonparametric Clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12): 7610–7620.

Xiao, A.; Chen, H.; Guo, T.; Zhang, Q.; and Wang, Y. 2023. Deep Plug-and-Play Clustering with Unknown Number of Clusters. *Transactions on Machine Learning Research*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747*.

Xing, Y.; He, T.; Xiao, T.; Wang, Y.; Xiong, Y.; Xia, W.; Wipf, D.; Zhang, Z.; and Soatto, S. 2021. Learning hierarchical graph neural networks for image clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3467–3477.

Yang, J.; and Lin, C.-T. 2025. Autonomous clustering by fast find of mass and distance peaks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yang, X.; Jing, H.; Zhang, Z.; Wang, J.; Niu, H.; Wang, S.; Lu, Y.; Wang, J.; Yin, D.; Liu, X.; et al. 2025. Darec: A disentangled alignment framework for large language model and recommender system. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, 904–917. IEEE.

Yang, X.; Yan, Y.; Huang, K.; and Zhang, R. 2019. Vsb-dvm: an end-to-end bayesian nonparametric generalization of deep variational mixture model. In *IEEE International Conference on Data Mining*, 688–697. IEEE.

Yao, J.; Pan, J.; and Zeng, Y. 2025. Gauging-$delta$: A Non-parametric Hierarchical Clustering Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yu, S.; Wang, S.; Dong, Z.; Tu, W.; Liu, S.; Lv, Z.; Li, P.; Wang, M.; and Zhu, E. 2024. A non-parametric graph clustering framework for multi-view data. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 38, 16558–16567.

Zhang, P.; Pan, Y.; Wang, S.; Yu, S.; Xu, H.; Zhu, E.; Liu, X.; and Tsang, I. 2025a. Max-Mahalanobis Anchors Guidance for Multi-View Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 22488–22496.

Zhang, Y.; Duan, Z.; Lu, M.; Ding, D.; Zhu, F.; and Ma, Z. 2024. Another way to the top: Exploit contextual clustering in learned image coding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 9377–9386.

Zhang, Y.; Zhang, Y.; Lu, Y.; Li, M.; Chen, X.; and Cheung, Y.-m. 2025b. Asynchronous federated clustering with unknown number of clusters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 22695–22703.

Zhao, H.; Yang, X.; and Deng, C. 2024. Parameter-agnostic deep graph clustering. *ACM Transactions on Knowledge Discovery from Data*, 18(3): 1–20.

Zhou, T.; Dong, Z.; Wang, S.; Liang, K.; Li, M.; Liu, X.; Zhu, E.; and Dong, X. 2025. DPFMVC: Dynamic Progressive Fusion for Multi-view Clustering. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 1102–1111.