# Deep Clustering Based on Sparse Kolmogorov-Arnold Network and Spectral Constraint

**Zixuan Bi, Yang Zhao*, Ganchao Liu**

[1]School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China
[2]School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China
bizixuan@mail.nwpu.edu.cn, {izhaoyang, liuganchao}@nwpu.edu.cn

## Abstract

At present, spectral clustering is an important branch of unsupervised learning, and its application in deep learning has been widely concerned. However, for high-dimensional sparse datasets, the complexity of network scale leads to parameter explosion, and static Gaussian kernel often has wrong preset data structure. To overcome these challenges, we propose a novel deep clustering model, Deep Clustering Based on Sparse Kolmogorov-Arnold Network (KAN) and Spectral Constraint. It contains a deep sparse clustering framework, in which sparse KAN and the orthogonal layer are designed to enhance the sparsity of the activation function matrix, reduce the number of parameters and improve the stability of model convergence. Additionally, we add an adaptive optimized affinity matrix based on spectral constraint, which overcomes the limitations of static Gaussian kernels, and improves the performance and stability of spectral constraint. Experimental results on both synthetic and real datasets demonstrate that our model outperforms existing methods in clustering performance, computational efficiency, and stability.

**Code** — https://github.com/bizixuan/sparse_kan_SC

## Introduction

In the field of machine learning and artificial intelligence, unsupervised learning, has always been one of the core issues of research. With the emergence of large-scale unlabeled datasets, unsupervised clustering has become one of the important methods in data analysis and processing, and occupies an indispensable position in image classification, remote sensing monitoring and other fields. The core task of clustering is to divide samples into several categories according to the inherent similarity or distribution characteristics between data, thus revealing the potential structural patterns and laws of data (Li, Wei, and Zhao 2024).

In the past decades, many classical clustering algorithms have emerged in the field of unsupervised learning, such as K-Means clustering (Nie et al. 2022), spectral clustering (SC) (Wang et al. 2022), hierarchical clustering (Cheung and Zhang 2019) and Gaussian mixture model (Zhao et al. 2021). SC has been widely used in unsupervised learning because of its powerful ability to capture the internal structure of data and flexible affinity measurement. Chen et al. (Chen et al. 2022) reduce the dependence of SC on the traditional affinity matrix by spectral embedding and spectral rotation, accelerate the clustering speed and improve the robustness. However, SC has high time complexity for large-scale datasets, and its performance on high-dimensional and nonlinear datasets is limited. In recent years, the rise of deep learning has brought a new research paradigm for unsupervised clustering (Zhao and Li 2023). Deep fuzzy clustering network (Tan et al. 2024) optimizes the clustering score by introducing mixed matrix norm regularization into the fuzzy clustering network. Spectralnet (Shaham et al. 2018) solves the scalability problem of traditional spectral clustering. However, the affinity matrix construction of deep SC is based on Gaussian kernel, which is effective for datasets with ideal Gaussian distribution, such as noise data and human body data, but it is difficult to adjust according to the requirements of different tasks, especially for high-dimensional sparse datasets, the data structure is usually wrongly preset (Koloskova, Hendrikx, and Stich 2023).

In addition, the full connection layer structure of deep neural network leads to the exponential growth of parameter scale with the increase of network depth and input dimension, which brings significant computational resource expenditure, leads to the slow convergence speed of optimization process and is easy to fall into local optimal solution (Lim 2021). Wan et al. (Wan et al. 2013) introduced sparse connections in the fully connected layer and reduced the parameter amount by randomly discarding the connection by using DropConnect technology. Xception (Chollet 2017) uses deep separable convolution operation, which effectively avoids the parameter expansion of the full connection layer. Kolmogorov-Arnold Networks (KAN) (Liu et al. 2025) can learn the activation function of each element, realize efficient feature extraction, and have pruning ability in the training process, thus reducing the parameter quantity and the training and reasoning time. However, sparse connections and convolution operations will lead to the instabislity of the model, and KAN relies on complex feature interaction in the training process, whose excessive pruning makes the fitting of activation functions too complicated and lead to the increase of parameters.

Inspired by the above methods, we propose a new deep

*Corresponding author

clustering model, called Deep Clustering Based on Sparse Kolmogorov-Arnold Network and Spectral Constraint. Our core strategy is to build a deep sparse clustering framework. The regularization constraints of global group lasso and entropy are added to KAN, which can control the joint sparsity of the activation function matrix and improve the efficiency of the model. In addition, the orthogonal layer is added to the final output of sparse KAN, which meets the requirement of spectral constraint loss for feature independence and improves the stability of the network. In addition, we design an adaptive optimized affinity matrix based on spectral constraint to participate in the loss calculation, which avoids the wrong preset data structure and improves the robustness of the model.

The main contributions of this article are summarized as follows.

- Deep sparse clustering framework is designed, which includes the following two parts: sparse KAN activation function matrix reduces the burden of parameter storage, and the orthogonal layer ensures the independence of features. Our model realizes a significant reduction in the number of parameters, improves the computational efficiency, and enhances the stability.
- Adaptive optimized affinity matrix based on spectral constraint is proposed. Combining with feature mapping, manifold information is dynamically learned, the robustness of the model is ensured and the clustering performance is significantly improved.
- Our model is verified on synthetic and real datasets, showing the improvement of performance, and the effectiveness of the model is proved by the analysis of parameters, convergence and stability.

## Preliminaries

### Kolmogorov-Arnold Network

MLP consists of fully connected layers of interconnected nodes, and each node has a corresponding weight and activation function. However, the weight setting of MLP for all nodes will lead to a sharp increase in its parameters, and MLP does not have the capability of adjusting the network structure during training. KAN (Liu et al. 2025) solves the problem of parameter explosion by pruning. In the training process, the nodes with little influence on the results are removed by pruning, and the network structure is continuously optimized. By changing the weight matrix into heterogeneous activation function matrix, the expression ability is prevented from weakening due to the reduction of parameters. The calculation equation of KAN can be expressed as

$$KAN(\mathbf{x}) = \prod_{l=0}^{L-1} \Phi_l \cdot \mathbf{x}, \tag{1}$$

$$\Phi_l(\cdot) = \begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_{l+1}}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_{l+1}}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{l,n_l,1}(\cdot) & \phi_{l,n_l,2}(\cdot) & \cdots & \phi_{l,n_l,n_{l+1}}(\cdot) \end{pmatrix}. \tag{2}$$

where $\mathbf{x}$ is the data entered into the network, $\Phi_l$ is a function matrix that represents all the activation functions of the $l$-th layer, with each node having its own different activation function, $L$ is the number of network layers.

In addition, KAN overcomes the limitation of the Kolmogorov-Arnold representation theorem. This theorem limits the number of hidden layers to 2, and the number of nodes in each hidden layer is $2n + 1$ ($n$ is the number of nodes in the previous layer). KAN can have any number of layers, and each layer can have any number of nodes. Activation function can be expressed as follows

$$\phi(\cdot) = w_a \alpha(\cdot) + w_b spline(\cdot), \tag{3}$$

where $w_a$ and $w_b$ are the weights of two parts respectively, $\alpha(\cdot)$ refers to sigmoid linear unit function, and $spline(\cdot)$ is composed of multiple B-spline basis functions. Usually the number of knots vectors in a spline function is eight.

When the traditional MLP network calculates the weight of the full connection layer, its time complexity and space complexity are both $O(\sum n_l^{MLP} n_{l+1}^{MLP})$. With the increase of network depth and data dimensions, the parameter scale expands rapidly. KAN significantly reduces the parameter scale of the network by pruning. Pruning makes the number of nodes in each layer much smaller than that in MLP networks, $n_l^{KAN} \ll n_l^{MLP}, l = 1, 2, ...L - 1$. In KAN, the time complexity is about $O(\sum n_l^{KAN} n_{l+1}^{KAN})$ and the space complexity is about $O(10 \sum n_l^{KAN} n_{l+1}^{KAN})$. Therefore, KAN has been greatly optimized in terms of parameters and computational overhead.

## Method

In this section, we will introduce our method in detail, which contains two branches. The first is **Deep Sparse Clustering Framework**. It solves the complexity problem of KAN activation function matrix by global group lasso and entropy regularization, and controls the orthogonality of output by cholesky decomposition which meets the requirement of spectral constraint loss. The second is **Spectral Constraint**, an adaptive optimized affinity matrix based on spectral constraint is designed, which is combined with feature mapping to realize stable and efficient training of the network. The main architecture is shown in Figure 1.

### Deep Sparse Clustering Framework

#### Sparse KAN

We propose a multi-regularization constraint mechanism into KAN, including Global Group Lasso regularization (GGL) and entropy regularization.

GGL regards the weight of each column of the activation function matrix as a group, and forces the parameters of each group to be zero, thus achieving group-level sparsity. At the same time, it ensures that the sparsity of each layer is close through an additional regularization item to avoid the imbalance of sparsity between layers. GGL can be expressed as

$$L_{GGL} = \sum_{l=1}^{L-1} \sum_{j=1}^{n_{l+1}} \sqrt{\sum_{i=1}^{n_l} ||\phi_{l,i,j}||_F^2} + \frac{1}{L-1} \sum_{l=1}^{L-1} (s_l - \bar{s})^2, \tag{4}$$

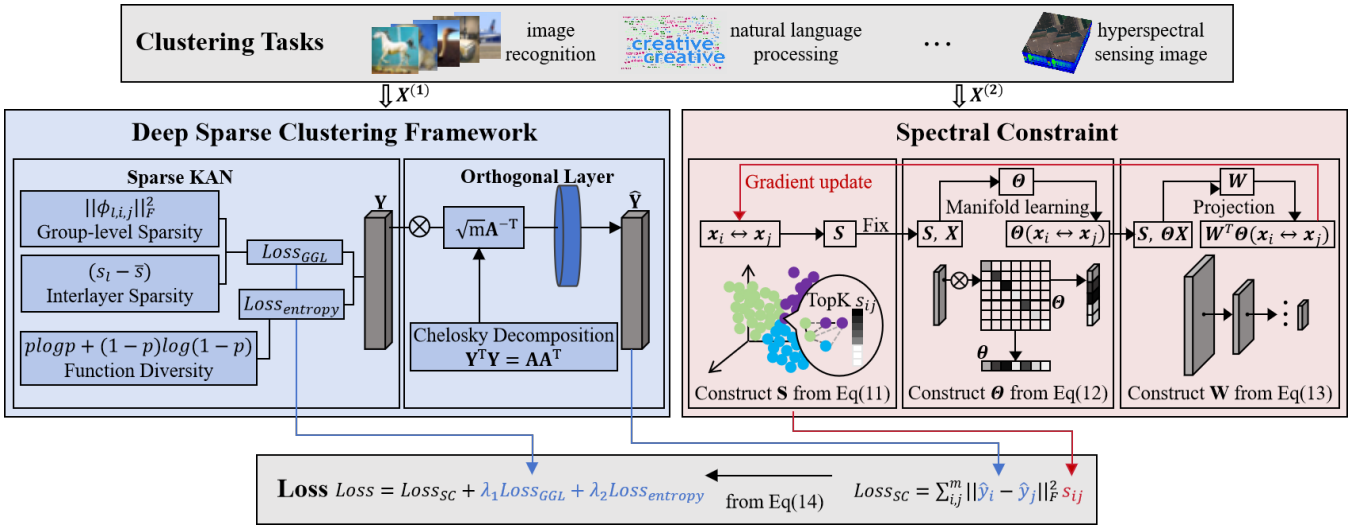



Figure 1: The architecture of our method. The upper branch describes the internal training stage of the network. The network is sparse by regularization constraints, and output to the last orthogonal layer, and orthogonalized by cholesky decomposition. The lower branch represents the affinity matrix training phase. The adaptive optimized affinity matrix is obtained by the projection matrix and feature weighting. Finally, the network loss is calculated and the output is obtained.

where $||\phi_{l,i,j}||$ contains eight B-spline weights and two global weights, $s_l$ is the sparsity of the $l$-th layer and $\bar{s}$ is the average sparsity of all layers. To prevent the decline of expression ability caused by sparse matrix, we use entropy regularization to ensure the distribution diversity of activation functions

$$L_{entropy} = -\sum_{l=1}^{L-1}\sum_{i,j} p_{l,i,j} log(p_{l,i,j}) + (1-p_{l,i,j}) log(1-p_{l,i,j}), \tag{5}$$

where $p_{l,i,j}$ is the normalized value of the $(i,j)$-th activation function of the $l$-th layer

$$p_{l,i,j} = \frac{||\phi_{l,i,j}(\cdot)||_1}{\sum_{k,m}||\phi_{l,k,m}(\cdot)||_1}. \tag{6}$$

Sparse constraint will force column sparsity, so that the parameter group of the whole output node is zero.

Assume the number of invalid nodes is $\rho$, the time complexity of sparse KAN is $O(\sum n_l^{KAN}(n_{l+1}^{KAN}-\rho))$, and the space complexity is $O(10\sum n_l^{KAN}(n_{l+1}^{KAN}-\rho))$. Through sparseness, KAN significantly reduces the amount of parameters while maintaining its expressive ability.

**Orthogonal Layer**

In SC, the solution of spectral embedding depends on calculating the eigenvectors corresponding to the first $c$ smallest eigenvalues of the Laplacian matrix $\mathbf{L}$ to form $\mathbf{F}$. These eigenvectors naturally form orthogonal bases, ensuring that each dimension in the embedded space captures independent cluster structure information (Nie et al. 2024). The objective function can be expressed as

$$\min_{\mathbf{F}} \sum_{j=1}^{c} \mathbf{f}_j^T \mathbf{L}\mathbf{f}_j \Rightarrow tr(\mathbf{F}^T\mathbf{L}\mathbf{F}), \tag{7}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is the Laplacian matrix, $\mathbf{S}$ is the affinity matrix and $\mathbf{D}$ is its degree matrix, $\mathbf{F}$ is the spectral embedding matrix we want to get.

Therefore, spectral constraint loss has a natural demand for feature independence. In deep learning, the mapping relation $\mathbf{Y} = F(\mathbf{X})$ of sparse KAN is used to replace $\mathbf{F}$. Without the orthogonal constraint, $\mathbf{Y}$ can't guarantee the linear independence of each dimension, which leads to the mutual coupling between features. Moreover, $\mathbf{Y}$ will converge to the trivial solution of repeated eigenvectors.

we add an orthogonal constraint to the last layer of the network to improve the stability of network convergence

$$\mathbb{E}(\mathbf{y}\mathbf{y}^T) = \mathbf{I}_{c\times c}, \tag{8}$$

where $c$ is the number of clusters.

To ensure the immediate optimization of the loss, we randomly select $m$ samples as a minibatch to provide data to the network in each iteration. We can get the following optimized orthogonal constraints

$$\frac{1}{m}\mathbf{Y}^T\mathbf{Y} = \mathbf{I}_{c\times c}, \tag{9}$$

where $\mathbf{Y} \in \mathbb{R}^{m\times c}$ consists of $\mathbf{y}^T \in \mathbb{R}^c$ by rows.

Calculate the Gram matrix of $\mathbf{Y}$, and make it positive definite by adding small perturbation. Then we construct $\mathbf{A}$ by Cholesky decomposition $\mathbf{Y}^T\mathbf{Y} = \mathbf{A}\mathbf{A}^T$. The orthogonal matrix $\hat{\mathbf{Y}}$ can be obtained by multiplying $\mathbf{Y}$ from the right by $\sqrt{m}(\mathbf{A})^{-T}$. This is verified in the arxiv.

The orthogonal layer ensures that each dimension of the embedding feature expresses independent information, so that the loss of spectral constraint can more effectively guide the network to learn embedding that conforms to data clustering.

## Spectral Constraint

Define an embedding matrix $\hat{\mathbf{Y}} \in \mathbb{R}^{m\times c}$ of the network output, which consists of $(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_m)^T$. We construct the loss function through spectral constraint:

$$Loss_{SC} = tr(\hat{\mathbf{Y}}^T \mathbf{L} \hat{\mathbf{Y}}) = \sum_{i,j=1}^{m} (||\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_j||_F^2 s_{ij}), \quad (10)$$

where $s_{ij}$ is the $(i, j)$-th element of the affinity matrix $\mathbf{S}$.

Spectral constraint encodes manifold information by the Laplacian matrix $\mathbf{L}$, and uses the value of the affinity matrix $\mathbf{S}$ to ensure that data points with high similarity in the original space remain close in the embedded space, while dissimilar points are pushed away. See arxiv for the specific construction method of the initial affinity matrix $\mathbf{S}$.

The purpose of affinity matrix based on adaptive optimization is to dynamically learn the distance measure inside the original dataset $\mathbf{X}$, so that the loss can learn $\hat{\mathbf{Y}}$ better. To meet the requirements of spectral constraint, we can get the following optimization (Nie et al. 2016)

$$\min_{\forall i, \mathbf{s}_i \mathbf{1}=1, \mathbf{s}_i \geq 0, s_{ii}=0} \sum_{i,j=1}^{m} (||\mathbf{x}_i - \mathbf{x}_j||_F^2 s_{ij} + \gamma s_{ij}^2), \quad (11)$$

where $\gamma > 0$ is a regularization parameter. We prefer to learn that $\mathbf{S}$ is a sparse matrix, so we add $\gamma$ to ensure that the optimal solution of $\mathbf{s}_i$ has exactly $c$ non-zero values. This is verified in the arxiv.

By combining manifold learning (Wang et al. 2023), we weight the original features in Eq (11), which makes the construction of the affinity matrix $\mathbf{S}$ more consistent with the manifold distribution of data and balances the features of different scales,

$$\begin{aligned} &\min \sum_{i,j=1}^{m} (||\mathbf{\Theta}\mathbf{x}_i - \mathbf{\Theta}\mathbf{x}_j||_F^2 s_{ij} + \gamma s_{ij}^2), \\ &s.t. \forall i, \mathbf{s}_i \mathbf{1} = 1, \mathbf{s}_j \geq 0, \\ &\mathbf{\Theta} = diag(\theta), \theta > 0, \theta^T \mathbf{1} = 1. \end{aligned} \quad (12)$$

where $\mathbf{\Theta}$ is the diagonal matrix of $\theta$. It not only captures the local structure of data, but also suppresses noise. Subsequently, using these weighted features, we continue to project them into low-dimensional space and extract more discriminating status feature representations to avoid feature redundancy and facilitate more effective processing and analysis,

$$\begin{aligned} &\min \sum_{i,j=1}^{m} (||\mathbf{W}^T\mathbf{\Theta}\mathbf{x}_i - \mathbf{W}^T\mathbf{\Theta}\mathbf{x}_j||_F^2 s_{ij} + \gamma s_{ij}^2) \\ &- \eta tr(\mathbf{W}^T\mathbf{\Theta}\mathbf{W}), \\ &s.t. \forall i,\ \mathbf{s}_i \mathbf{1} = 1,\ \mathbf{s}_i \geq 0,\ \mathbf{\Theta} = diag(\theta), \\ &\theta > 0,\ \theta^T \mathbf{1} = 1,\ \mathbf{W}^T\mathbf{W} = \mathbf{I}_w. \end{aligned} \quad (13)$$

where $\mathbf{W}^T \in \mathbb{R}^{w\times d}$ is a projection subspace matrix, and the second term is to avoid falling into the local optimal solution $\mathbf{W}^T\mathbf{\Theta} = 0$.

Algorithm 1: Deep Clustering Based on Sparse KAN and Spectral Constraint

**Input:** Dataset $\mathbf{X} \in \mathbb{R}^{n\times d}$, The number of clusters $c$, Batch size $m$, Reduced dimension $w$, Regularization parameters $\lambda_1, \lambda_2, \eta$.
**Output:** Indicator matrices $\hat{\mathbf{Y}} \in \mathbb{R}^{m\times c}$, Cluster assignments $k_1, k_2, ..., k_n, k \in 1, 2, .., c$.
Randomly initialize the network weights;
Initialize $\mathbf{S}$, $\theta = \frac{1}{d}\mathbf{I}_d$;
**While** *Loss not converge* **do**
  **Deep Sparse Clustering Framework**:
    1) Sample a random minibatch $\mathbf{X}$ of size $m$;
    2) Forward propagate $\mathbf{X}$ and compute inputs to the orthogonal output $\hat{\mathbf{Y}}$;
    3) Calculate $Loss_{GGL}$ and $Loss_{entropy}$ by Eq (4) and Eq (5);
    4) Set the weights of the orthogonalization layer through Cholesky decomposition;
    5) Forword propagate $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m$ to $\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_m$;
  **Spectral Constraint**:
    1) Update $\mathbf{L} = \mathbf{D} - \frac{\mathbf{S}^T + \mathbf{S}}{2}$, where $\mathbf{D}$ is a degree matrix;
    2) Update $\mathbf{W}$, $\mathbf{\Theta}$, $\mathbf{S}$ in turn;
    3) Calculate $Loss_{SC}$ by Eq (10);
    4) Calculate $Loss$ by Eq (14);
    5) Use the gradient of $Loss$ to turn all layer weights, except those of the output layer;
**End**
Once training finished, forward propagate data points to get the orthogonal output, and then obtain final cluster assignment matrix through $k$-means.

To obtain the globally optimal solution of Eq (13), we use gradient descent method to optimize $\mathbf{W}$, $\mathbf{\Theta}$ and $\mathbf{S}$ in turn and judge the convexity of the objective function to ensure the effectiveness of the alternating iteration. See arxiv for specific construction methods. Finally, we bring the calculated $\mathbf{S}$ into Eq (10) to calculate the loss.

We successfully add spectral constraint loss to the network, and improve the traditional Gaussian kernel SC from adaptive optimization of the affinity matrix and feature selection. The combination of $\mathbf{\Theta}$ and $\mathbf{W}$ constitutes a joint optimization framework. By learning feature weights and projection subspaces, the interference of noise in high-dimensional data can be effectively alleviated, the vulnerability of Gaussian kernel can be avoided. Eq (13) allows the affinity matrix $\mathbf{S}$ to be adaptively optimized through the task-driven objective function, so as to better match the requirements of clustering tasks, thus avoiding the wrong preset data structure of Gaussian kernel.

Finally, we get the overall loss of the network.

$$Loss = Loss_{SC} + \lambda_1 Loss_{GGL} + \lambda_2 Loss_{entropy}, \quad (14)$$

where $\lambda_1$ and $\lambda_2$ are parameters that control regularization. The detailed algorithm flow can is given in Algorithm 1.

## Experiment

All the experiments are carried out on PC with 2.4GHz Xeon Silver 4210 R CPU, NVIDIA GeForce RTX 4090, Ubuntu 20.04 system and Python 3.8.

**Dataset. Tiny Imagenet** contains 200 categories, with a total of 120,000 images, all of which are reduced to the size of 64*64, which is a computer vision dataset specially designed for image recognition and classification tasks. **Reuters** contains 21,578 news documents, which are divided into 46 subject categories. Text data are converted into word bags for classification, mainly for natural language processing tasks. In this article, we choose 10 categories. **CIFAR-100** contains 60,000 32*32 color images in 100 categories, such as airplanes, cars, birds, ships, etc. It is mainly used in more challenging image classification tasks. **Botswana** contains 189 bands of hyperspectral images, and the scene size is 144*144 pixels. It is an important dataset for hyperspectral image classification and is widely used in the field of remote sensing.

**Metrics.** For the experiment, we use ACC to indicate the matching degree between the clustering result and the real label, NMI to measure the information sharing between the clustering result and the real label. In addition, we compare the parameters of different methods by the histogram, draw the measurement curve of loss to compare the convergence speed of different methods, and plot the box diagram of ACC and NMI to compare the stability of different methods.

**Baselines.** The model is compared with K-Means (Shen et al. 2017), SC (Wang et al. 2022) and KAN that Gaussian kernel $\sigma$ takes different values on two synthetic datasets. And for real datasets, we compare the model with 13 clustering methods, including traditional clustering methods SC (Wang et al. 2022), low-rank representation (LRR) (Fu et al. 2021), low-rank subspace clustering (LRSC) (Tang, Xie, and Zhang 2023) and deep clustering methods, including unsupervised learning variational autoencoder (UL-VAE) (Ambekar and Thokchom 2024), knowledge distillation (KD) (Shao et al. 2023), adaptive adversarial distillation (AdaAD) (Huang et al. 2023), Manifold Projection Enhance (MPE) (Li et al. 2025), deep SC with constrained laplacian rank (DSCCLR) (Li, Wei, and Zhao 2024), deep SC with Projected Adaptive Feature Selection (DSCFS) (Zhao et al. 2025), KAN (Liu et al. 2025) and various variants.

**Parameter Settings.** In the experiment, we set the mini-batch size to 256; SC constructs affinity matrix by calculating the distance metric of neighbors, in which the number of nearest neighbors is set to 10; LRR and LRSC use the default parameter setting; The optimizers of UL-VAE are Adam, the initial learning rate is set to 0.001, and the KL loss weight is set to 0.00025. KD, AdaAD and MPE use the default parameters in the references. The final clustering methods of all deep models adopt K-Means. All the experiments have been repeated 10 times.

### Clustering Performance Evaluation

**Synthetic Datasets.** Figure 2 shows the experimental results of our method and comparison methods on synthetic datasets, and the results show that our model is superior to other methods in performance. According to the results, we have the following observations:

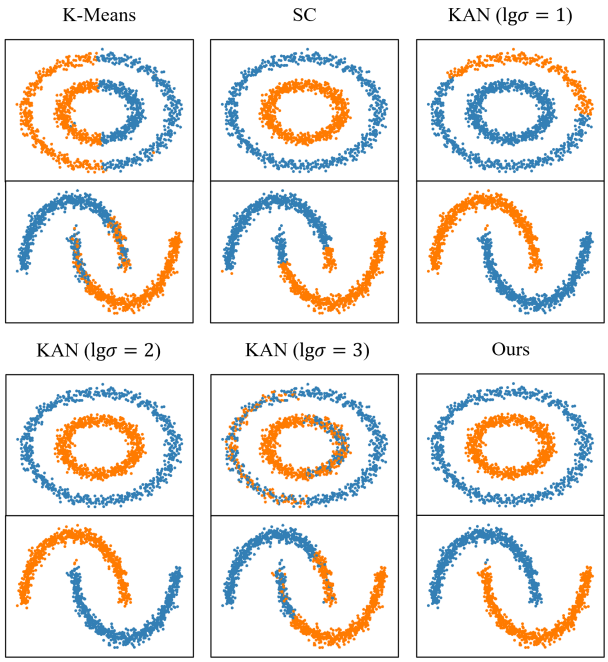



Figure 2: Comparison of different clustering methods on synthetic datasets. Two synthetic datasets are Circle in Circle (CC), Two Moons (TM).

**1)** For KAN network based on Gaussian kernel $\sigma$, the clustering performance is badly influenced by the numerical value of Gaussian kernel, and the robustness of the model is poor;

**2)** This result shows that all deep clustering models can handle this kind of small-scale nonconvex datasets, and our model has better performence.

**Real Datasets.** We have tested the model with other methods on large-scale real datasets respectively. In Table 1, we get the clustering results of different methods, including traditional clustering (SC, LRR, LRSC) and deep clustering (UL-VAE, KD, AdaAD, MPE, DSCCLR, DSCFS, KAN and various variants). Among these methods, we especially compare the sparse module and the spectral module which have influence on the model performance. The parameters of all comparison methods are well adjusted for different samples. It is worth noting:

**1)** Our method achieves better results on four datasets compared with baseline methods. Compared with the best baseline method, ACC increased by 4.86%, 2.7%, 6.42% and 2.98% respectively, NMI increased by 8.28%, 0.67%, 4.51% and 2.46%;

**2)** Our method is improved compared with the original KAN and two variants Ours without SC and Ours without Sparse, which proves that each module of our proposed model has a good performance improvement effect and spectral constraint do better;

**3)** The breakthrough performance on Tiny Imagenet and

| Dataset | Tiny Imagenet | | Reuters | | CIFAR-100 | | Botswana | |
|---|---|---|---|---|---|---|---|---|
| Method | ACC | NMI | ACC | NMI | ACC | NMI | ACC | NMI |
| (A) Traditional Clustering Methods | | | | | | | | |
| SC (Wang et al. 2022) | 25.13 | 28.76 | 42.52 | 18.90 | 18.05 | 19.47 | 68.20 | 74.17 |
| LRR (Fu et al. 2021) | 47.67 | 50.18 | 51.33 | 31.61 | 39.18 | 42.81 | 65.64 | 68.04 |
| LRSC (Tang, Xie, and Zhang 2023) | 51.23 | 61.37 | 61.33 | 44.33 | 52.16 | 55.68 | 66.41 | 64.16 |
| (B) Deep Clustering Methods | | | | | | | | |
| UL-VAE (Ambekar and Thokchom 2024) | 49.61 | 50.27 | 69.18 | 47.52 | 61.59 | 61.77 | 52.25 | 60.22 |
| KD (Shao et al. 2023) | 55.29 | 60.71 | 78.68 | 62.77 | 64.52 | 67.28 | 77.95 | 73.20 |
| AdaAD (Huang et al. 2023) | 53.73 | 57.28 | 75.26 | 61.59 | 62.49 | 65.37 | 74.23 | 79.91 |
| MPE (Li et al. 2025) | 53.58 | 55.76 | 82.05 | 69.70 | 66.58 | 64.23 | 62.54 | 80.19 |
| DSCCLR (Li, Wei, and Zhao 2024) | 45.27 | 44.37 | 80.31 | 70.01 | 52.36 | 55.72 | 81.15 | 81.98 |
| DSCFS (Zhao et al. 2025) | 52.31 | 57.25 | 78.23 | 60.57 | 60.39 | 66.17 | 82.39 | 83.58 |
| KAN (Liu et al. 2025) | 53.12 | 51.24 | 80.37 | 65.28 | 65.21 | 70.36 | 80.19 | 80.79 |
| Ours Without SC | 53.25 | 51.73 | 80.64 | 67.25 | 65.03 | 69.97 | 81.21 | 82.37 |
| Ours Without Sparse | 56.52 | 60.26 | 82.31 | 70.29 | 67.62 | 70.27 | 83.59 | 84.92 |
| Ours | **60.15** | **69.65** | **84.75** | **70.68** | **73.00** | **74.87** | **85.37** | **86.04** |

Table 1. Performance comparisons with 13 clustering methods on real datasets. The figures in boldface are best results

| Dataset | Tiny Imagenet | Reuters | CIFAR -100 | Botswana |
|---|---|---|---|---|
| MLP | 53.16m | 117.62s | 30.26m | 349.91s |
| KAN | 43.10m | 98.91s | 24.26m | 314.28s |
| Ours | **24.15m** | **87.58s** | **20.41m** | **285.99s** |

Table 2. Running time of three network architectures on four datasets. Take the time of one epoch as an example.

CIFAR-100 shows that our model has a better performance for image classification tasks. This proves that our model has stronger learning ability for local features and can better capture complex manifold structures.

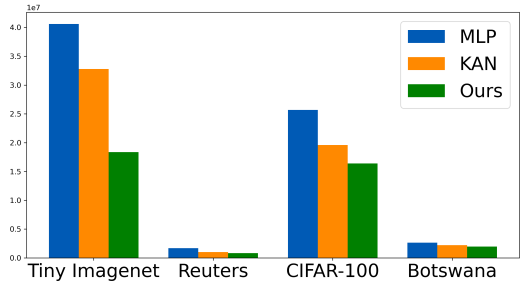


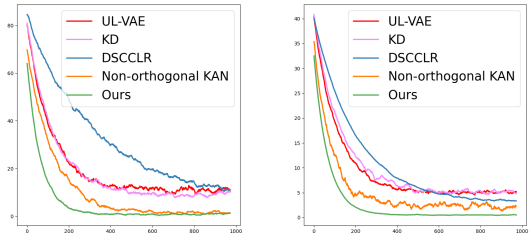

Figure 3: Comparison of parameters of three network architectures on four datasets. MLP adopts the conventional architecture of each dataset

## Parameters Analysis

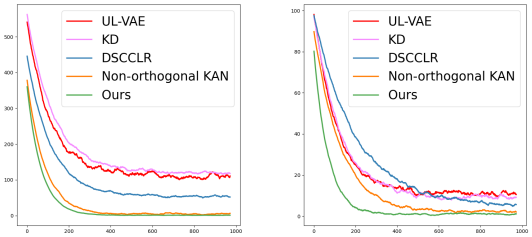
Figure 3 illustrates parameter counts of three methods (MLP, KAN, and Ours) on four real datasets. Table 2 shows the running time of the three methods. While ensuring the experimental accuracy, MLP adopts the conventional architecture of each dataset. The parameter count of MLP is the highest, which is primarily attributed to its use of fully connected layers for constructing the network. In contrast, the original KAN model demonstrates a lower parameter count, indicating that KAN replaces weight matrices with activation function matrices. This allows the network to express more complex data structures using fewer fitting functions. Furthermore, the use of GGL and entropy regularization results in a decrease in the overall network architecture's parameter count. Our model further sparsifies KAN, successfully reduces the number of parameters.

## Convergence Analysis

We have theoretically proved the convexity of the network loss function of the proposed model in the arxiv. In this part, the convergence curve of the mentioned model and the other four methods are tested empirically on the real datasets, and the comparison method with better performance in Table 1 is selected. It is worth noting that we have tested our model and KAN without the orthogonal layer, as shown in Figure 4. For each subgraph, the x-axis is the number of iterations and the y-axis is the network loss value. From the experiment, we can observe that the loss function is monotonically decreasing until the proposed method converges. Because of fewer network parameters, our model can converge at a faster speed, and compared with KAN without the orthogonal layer, our model is more stable on the convergence curve.

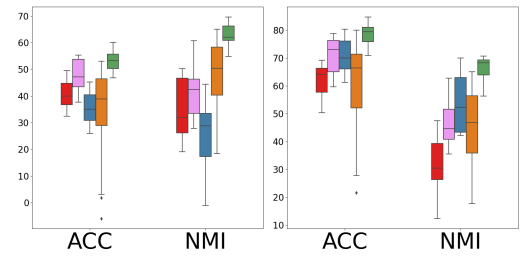



(a) Tiny Imagenet (b) Reuters

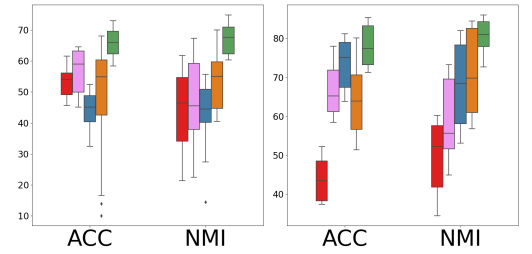



(c) CIFAR-100 (d) Botswana

Figure 4: Convergence curves of five algorithms on four datasets. Non-orthogonal KAN is the result of ours to remove the orthogonal layer.

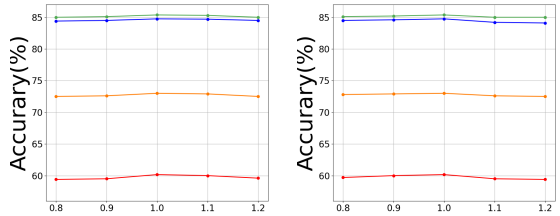



(a) Tiny Imagenet (b) Reuters

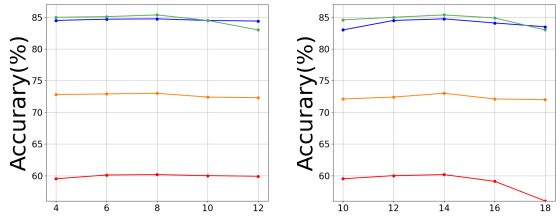



(c) CIFAR-100 (d) Botswana

Figure 5: Boxplot for ACC, NMI and ARI of five methods after ten runs on four datasets. G-KAN stands for SC using Gaussian kernel. The red is UL-VAE, the pink is KD, the blue is DSCCLR, the orange is G-KAN and the green is Ours.




(a) $\lambda_1$ (b) $\lambda_2$




(c) $\eta$ (d) $w$

Figure 6: The impact of two regularization weights and two affinity matrix parameters on four datasets is evaluated. The red is Tiny Imagenet, the blue is Reuters, the orange is CIFAR-100 and the green is Botswana.

## Stability Analysis

Figure 5 shows ACC and NMI statistics of VAE, KD, DSCCLR, G-KAN and our proposed model after fifteen independent experiments on six real datasets. The results show that our model has excellent performance which shows that our model is more robust and stable in dealing with noise after optimizing feature weights and projection subspace.

In Figure 6, we have evaluated the effects of $\lambda_1$, $\lambda_2$, $\eta$ and $w$ respectively. Based on our experimental observations, setting $\lambda_1$ and $\lambda_2$ to 1, $\eta$ in [6,8] and $w$ in [12,14] can achieve good results for all datasets. While these values may vary depending on other datasets, these findings provide useful insights and can serve as a guideline for tuning these parameters in future experiments.

## Conclusion

In this paper, we address the problems existing in previous algorithms, e.g., the huge amount of network training parameters and the strong dependence of SC on Gaussian kernel. We propose Deep Clustering Based on Sparse Kolmogorov-Arnold Network and Spectral Constraint. Specifically, we theoretically prove that adding regularization and orthogonalization to the network structure can effectively reduce the number of parameters needed by the network in the training process and improve the stability. By adaptively optimizing the feature projection and affinity matrix, we successfully avoid the problems that the Gaussian kernel cannot be dynamically matched.

## Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2022ZD0160402), the National Natural Science Foundation of China (No. 62201471, 62273282).

## References

Ambekar, N. G.; and Thokchom, S. 2024. UL-VAE: An Unsupervised Learning Approach for Zero-day Malware Detection Using Variational Autoencoder. In *2024 International Conference on Computational Intelligence and Network Systems*, 1–7.

Chen, J.; Zhu, J.; Xie, S.; Yang, H.; and Nie, F. 2022. FGC_SS: Fast Graph Clustering Method by Joint Spectral Embedding and Improved Spectral Rotation. *Information Sciences*, 613: 853–870.

Cheung, Y.-m.; and Zhang, Y. 2019. Fast and Accurate Hierarchical Clustering Based on Growing Multilayer Topology Training. *IEEE Transactions on Neural Networks and Learning Systems*, 30(3): 876–890.

Chollet, F. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 1800–1807.

Fu, Z.; Zhao, Y.; Chang, D.; Zhang, X.; and Wang, Y. 2021. Double Low-Rank Representation With Projection Distance Penalty for Clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 5320–5329.

Huang, B.; Chen, M.; Wang, Y.; Lu, J.; Cheng, M.; and Wang, W. 2023. Boosting Accuracy and Robustness of Student Models via Adaptive Adversarial Distillation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24668–24677.

Koloskova, A.; Hendrikx, H.; and Stich, S. U. 2023. Revisiting Gradient Clipping: Stochastic bias and tight convergence guarantees. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, 17343–17363.

Li, X.; Wei, T.; and Zhao, Y. 2024. Deep Spectral Clustering With Constrained Laplacian Rank. *IEEE Transactions on Neural Networks and Learning Systems*, 35(5): 7102–7113.

Li, Z.; Yin, S.; Jiang, T.-X.; Hu, Y.; Wu, J.-M.; Yang, G.; and Liu, G. 2025. Enhancing the Adversarial Robustness via Manifold Projection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 451–459.

Lim, H.-i. 2021. A Study on Comparative Analysis of the Effect of Applying DropOut and DropConnect to Deep Neural Network. In *Intelligent Human Computer Interaction*, 42–47.

Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljacic, M.; Hou, T. Y.; and Tegmark, M. 2025. KAN: Kolmogorov-Arnold Networks. In *The Thirteenth International Conference on Learning Representations*.

Nie, F.; Liu, C.; Wang, R.; and Li, X. 2024. A Novel and Effective Method to Directly Solve Spectral Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 10863–10875.

Nie, F.; Wang, X.; Jordan, M.; and Huang, H. 2016. The Constrained Laplacian Rank Algorithm for Graph-Based Clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Nie, F.; Xue, J.; Wu, D.; Wang, R.; Li, H.; and Li, X. 2022. Coordinate Descent Method for $k$-means. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5): 2371–2385.

Shaham, U.; Stanton, K. P.; Li, H.; Basri, R.; Nadler, B.; and Kluger, Y. 2018. SpectralNet: Spectral Clustering using Deep Neural Networks. In *6th International Conference on Learning Representations*.

Shao, R.; Zhang, W.; Yin, J.; and Wang, J. 2023. Data-free Knowledge Distillation for Fine-grained Visual Categorization. In *2023 IEEE/CVF International Conference on Computer Vision*, 1515–1525.

Shen, X.; Liu, W.; Tsang, I.; Shen, F.; and Sun, Q.-S. 2017. Compressed K-Means for Large-Scale Clustering. 1, 2527–2533.

Tan, D.; Huang, Z.; Peng, X.; Zhong, W.; and Mahalec, V. 2024. Deep Adaptive Fuzzy Clustering for Evolutionary Unsupervised Representation Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(5): 6103–6117.

Tang, Y.; Xie, Y.; and Zhang, W. 2023. Affine Subspace Robust Low-Rank Self-Representation: From Matrix to Tensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8): 9357–9373.

Wan, L.; Zeiler, M.; Zhang, S.; Le Cun, Y.; and Fergus, R. 2013. Regularization of Neural Networks using DropConnect. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, 1058–1066.

Wang, Q.; Miao, Y.; Chen, M.; and Yuan, Y. 2022. Spatial-Spectral Clustering With Anchor Graph for Hyperspectral Image. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–13.

Wang, Z.; Cao, L.; Lin, W.; Jiang, M.; and Tan, K. C. 2023. Robust Graph Meta-Learning via Manifold Calibration with Proxy Subgraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 15224–15232.

Zhao, M.; Jia, X.; Fan, L.; Liang, Y.; and Yan, D.-M. 2021. Robust Ellipse Fitting Using Hierarchical Gaussian Mixture Models. *IEEE Transactions on Image Processing*, 30: 3828–3843.

Zhao, Y.; Bi, Z.; Zhu, P.; Yuan, A.; and Li, X. 2025. Deep Spectral Clustering With Projected Adaptive Feature Selection. *IEEE Transactions on Geoscience and Remote Sensing*, 63: 1–12.

Zhao, Y.; and Li, X. 2023. Deep Spectral Clustering With Regularized Linear Embedding for Hyperspectral Image Clustering. *IEEE Transactions on Geoscience and Remote Sensing*, 61: 1–11.