DEEP SIAMESE GENERATIVE ADVERSARIAL CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep clustering has attracted considerable attention owing to the inspiring performance in mining patterns of data. However, previous deep clustering methods cannot capture the category-invariant structures of data in mining patterns. Moreover, they cannot take into consideration the consistencies of data manifolds in learning clustering dependence between categories and instances. To address those challenges, a deep Siamese generative adversarial clustering method is proposed, which can capture the category-invariant structures and consistent manifolds of data in pattern mining. Specifically, the generative clustering is defined as the generation process of pairwise data within an expectation-maximization scheme, which explores the link information of pairwise data to facilitate consistent structures in clusters and different structures between clusters in fitting data distributions. Furthermore, a deep Siamese generative adversarial clustering network is proposed to implement the definition within a symmetrical architecture that contains generator, discriminator, and cluster networks, which can capture the category-invariant structures and consistencies of data manifolds in clustering. Afterwards, an adversarial clustering loss is defined based on the adversarial game of data distribution and the binary classification task of cluster structures to train the network parameters, which can facilitate the intra-cluster compactness and inter-cluster separation in partitioning data. Finally, extensive experiments are conducted on six datasets and the results illustrate the state-of-the-art performance of the proposed method in comparison with seventeen cutting-edge methods.

1 INTRODUCTION

Clustering, as an essential unsupervised learning method, is attracting plentiful attention in various domains (Singh et al., 2026; Gao et al., 2024). It typically utilizes structure information to partition data into several clusters, such that data of the same cluster are more similar than those of different clusters. Many clustering algorithms are proposed to extract valuable patterns from data. For example, k-means finds out the k centroids that minimize the sum of intra-cluster distances to divide data into non-overlapping groups (Hong et al., 2026). Gaussian Mixture Model fits the joint distribution between categories and instances within the expectation-maximization algorithm to learn data assignment. Those clustering algorithms are the shallow models that focus on linear patterns of data. They usually produce poor performance in mining intrinsic patterns hidden in the non-linear correlations between high-dimension data (Li et al., 2022).

In recent years, deep clustering has been proposed to recognize intrinsic patterns of high-dimension data (Vardakas et al., 2026). It integrates deep learning into the partition of data to reveal non-linear correlations between instances. At present, deep clustering can be approximately grouped into deep discriminative clustering and deep generative clustering. The deep discriminative clustering learns patterns via modeling the partition relation between categories and instances in exploring data structures (Zhou et al., 2025). The deep generative clustering captures patterns via modeling the generation relation between categories and instances. It learns the recovery process of complex high-dimension data to capture structure information of instances (Luo et al., 2026).

Although deep generative clustering produces encouraging performance on the pattern mining against deep discriminative clustering, there are still two vast challenges. First of all, they often utilize the single-flow network architecture in modeling clustering relation between categories and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

instances, which leads to failure in capturing category-invariant structures of instances in mining patterns. Secondly, they only focus on modeling the clustering relation between categories and instances, and neglect consistencies hidden in the manifolds of data, which may degrade intra-cluster compactness and inter-cluster separation in mining patterns.

To this end, the **D**eep **S**iamese **G**enerative **A**dversarial **C**lustering (**DSGAC**) is proposed to partition data, which can capture the category-invariant structures and consistent manifolds of data. Specifically, the clustering is interpreted as a generation process of pairwise instances within an expectation-maximization scheme, which utilizes the links between instances to capture consistent structures in clusters and different structures between clusters in fitting data distributions. In the expectation step, a conditional joint probability matrix of categories is derived to capture clustering relationships between instances, which is used to distinguish if pairwise instances are in the same category based on the link information of generated pairwise instances. In the maximization step, an adversarial game between real instances and generated instances is derived to flexibly fit complex distributions of data, which is used to produce high-quality pairwise instances with link information that indicates whether pairwise instances are in the same cluster. More specifically, clustering is interpreted as a two-step paradigm. In the first step, it produces high-quality pairwise instances with link information whether pairwise instances are in the same cluster. In the second step, it utilizes the link information to partition data via distinguishing if pairwise instances are in the same category.

To implement the definition, a deep Siamese generative adversarial clustering network is devised within a symmetrical architecture, where a generator network, a discriminator network, and a cluster network cooperate to learn clustering-specific representations and partition data in an end-to-end manner. In detail, the generator network captures a probabilistic generation function from noise inputs with category-static information to instances to produce pairwise data with link information. The discriminator network distinguishes between real instances and generated instances to provide supervision information for distribution fitting. The cluster network learns indicator features of data for clustering via detecting the structural differences between pairwise instances of the same cluster and pairwise instances of different clusters. The three networks seamlessly cooperate based on the generated pairwise instances with the link information to capture the category-invariant structures and consistent manifolds of data in clustering. Furthermore, an adversarial clustering loss composed of an adversarial loss and a clustering loss is introduced to train parameters of the deep Siamese generative adversarial clustering network in accordance with the expectation-maximization scheme. It utilizes the adversarial games of data distributions and the binary classification task of cluster structures to facilitate the intra-cluster compactness and inter-cluster difference in partitioning data.

Thus, the significant contributions of this paper can be concluded in three-fold aspects:

- A new clustering method is introduced via defining the generation process of pairwise data from category variables to instance variables, which explore the links between instances to capture consistent structures in clusters and different structures between clusters in fitting data distributions.
- A deep Siamese generative adversarial clustering network is devised to learn clustering-specific representations and partition data in an end-to-end manner, which can capture the category-invariant structures and the consistencies of data manifolds.
- Extensive experiments are conducted on six benchmark datasets, illustrating the superiority of DSGAC compared with seventeen state-of-the-art methods in terms of ACC, NMI, and ARI.

2 Related Work

The deep discriminative clustering models the partition relation between categories and instances in exploring data structures to mine patterns. It can be divided into deep prototype clustering and deep graph clustering (Li et al., 2020; Vardakas et al., 2026). The former minimizes the intra-cluster distance between centroids and instances to produce self-supervised information, such as pseudo labels and soft distributions, which is further used to guide deep neural networks to capture decision boundaries of data partitions. For instance, Guo et al. (2020) designed an adaptive deep clustering model, in which the self-paced learning is used to exclude hard samples in minimizing the intra-cluster distance between centroids and instances. The latter explores a data-driven similarity

graph of instances in the non-linear space. Then, it performs Laplacian embedding to capture the adjacency relation between instances (Zhang et al., 2019). For example, Yu et al. (2022) proposed a self-supervised deep subspace clustering network based on the adversarial data augmentation, which utilizes the consistencies in both representations and data to enhance the robustness of the adjacency matrix of data. Those methods explicitly capture the partition relation between categories and instances in mining data patterns. However, they are confined to the discriminative mapping of deep neural networks, which leads to the failure in capturing clustering information hidden in consistent manifolds of data. Furthermore, they cannot capture category-invariant information of data in modeling the partition relation. Thus, they may degrade the clustering performance when dealing with complex-distribution data.

The deep generative clustering captures the generation relation between categories and instances to mine patterns of data, which is composed of deep variational clustering and generative adversarial clustering (Singh et al., 2026; Wang & Jiang, 2021). The former maximizes an evidence lower bound of the data likelihood that contains hidden category variables to learn data partitions from the generation process from categories to instances. For example, Yang et al. (2020) proposed an adversarial variational embedding model via utilizing the Student's t-distribution to mix multiple neural Gaussian nets, which can learn robust clustering structures from the maximization of data likelihood. The latter mines patterns of data via utilizing the adversarial game to extract pseudo label information from the generation relation between categories and instances (Luo et al., 2026). For instance, de Mello et al. (2022) proposed a top-down generative adversarial clustering model, which utilizes pseudo label information derived from the hierarchical clustering to produce data that belong to one of the predefined categories. Those methods leverage the generation relation between categories and instances to recognize patterns of data, which can capture information hidden in data manifolds. However, they are confined to the single-flow network architecture, which cannot capture category-invariant structures of instances in modeling the generation relation. Moreover, they neglect the consistency of data manifolds, which may degrade intra-cluster compactness and inter-cluster separation. Thus, they may not make full use of data manifolds in pattern mining.

3 The Definition of DSGAC

In this section, clustering is defined as a generation process of pairwise data from category variables to instance variables within an expectation-maximization scheme, which can utilize the information of links between instances in maximizing data likelihood to capture category-invariant structures and consistencies of data manifolds.

3.1 The Generative Process of Pairwise Data

The generative process of pairwise data is introduced along with a pairwise data log-likelihood, which explores the link information that indicates whether pairwise instances are in the same cluster to capture the consistent structures in clusters and different structures between clusters. It can utilize the equal pairwise category variables and unequal pairwise category variables to generate pairwise instances of the same cluster and pairwise instances of the different clusters, respectively. Specifically, given the dataset $\{x_i\}_{i=1}^{N}$ with the K predefined clusters, where N instances $x_i \in \mathbb{R}^d$ are subject to the independent and identical distribution, the generative process is depicted as:

1. Sample two category variables from the category distribution, i.e., $c_i \sim \text{Cat}(K)$ and $c_j \sim \text{Cat}(K)$;
2. Sample two latent representations from the category variables, i.e., $z_i \sim \text{Cluster}(c_i)$ and $z_j \sim \text{Cluster}(c_j)$;
3. Generate two instances from the two latent representations, i.e., $x_i = f(z_i)$ and $x_j = f(z_j)$;
4. Endow x_i and x_j with the link information, i.e., must-link and cannot-link, based on the relationship between c_i and c_j.

where c_* and z_* denote the category variable and the latent representation of the $*$-th sample x_*, respectively, $\text{Cat}(K)$ represents the distribution of categories, $\text{Cluster}(c_*)$ denotes the distribution of the $*$-th cluster, $f()$ is the generation function, $*$ denotes either i or j, and $i, j \in \{1, 2, ..., N\}$.

In the generation process, pairwise generated instances, i.e., x_i and x_j, are endowed with the must-link, if $c_i = c_j$ is chosen. This implies that the pairwise generated instances should be in the same cluster. Otherwise, the pairwise generated instances are endowed with the cannot-link, specifying that pairwise generated instances should belong to different clusters. In addition, the pairwise instances with the must-link indicate consistent structures in clusters, and the pairwise instances with the cannot-link reflect different structures between clusters.

According to the generation process, the pairwise data log-likelihood is introduced to fit the distribution of data, which utilizes the co-occurrence of pairwise instances to reveal the link information of instances. To be specific, it utilizes the joint probability of pairwise instances to rewrite the vanilla log-likelihood of data, as follows:

$$\mathbb{L} = \log \prod P(x_i;\theta) \propto \quad \log \prod P(x_i;\theta)P(x_j;\theta) = \iint \log P(x_i, x_j;\theta) dx_i dx_j \tag{1}$$

where $P(x_i;\theta)$ is the marginal probability of x_i, θ is the model parameter, and $P(x_i, x_j;\theta)$ is the joint probability to denote the co-occurrence between x_i and x_j. Then, the hidden category variables are introduced into the pairwise data log-likelihood to mine intrinsic clustering patterns, which can be further redefined as the lower bound with the following form (The proof is shown in the Appendix A.2):

$$\mathbb{L} = \iint \mathbb{E}_{P(c_i,c_j|x_i,x_j;\theta)} \log P(x_i, x_j, c_i, c_j;\theta) dx_i dx_j \tag{2}$$

where $\mathbb{E}$ is the expectation operation, $P(x_i, x_j, c_i, c_j;\theta)$ is the joint probability between categories and instances, and $P(c_i, c_j|x_i, x_j;\theta)$ is the conditional joint probability of categories given x_i and x_j. $P(c_i, c_j|x_i, x_j;\theta)$ expresses the link information between x_i and x_j, i.e., must-link and cannot-link, and it estimates whether the pairwise instances x_i and x_j are in the same cluster. Thus, the lower bound can ensure the exploration of links between instances in the generation process.

3.2 The Expectation-Maximization Scheme

The expectation-maximization scheme is designed to maximize the pairwise data log-likelihood, which alternates an expectation step (E-step) to mine clustering patterns and a maximization step (M-step) to fit data distributions.

E-step leverages structural differences between data to fit the ground-truth conditional joint probability of categories. In particular, given a pair of instances (x_i, x_j), E-step uses the Bayes rule to compute the conditional joint probability of categories as:

$$P(c_i, c_j|x_i, x_j;\theta) = \frac{P(x_i, x_j|c_i, c_j;\theta)P(c_i, c_j;\theta)}{\sum\limits_{c_i,c_j} P(x_i, x_j|c_i, c_j;\theta)P(c_i, c_j;\theta)} \tag{3}$$

where $P(c_i, c_j;\theta)$ is the joint probability of categories, and $P(x_i, x_j|c_i, c_j;\theta)$ is the conditional joint probability of instances given c_i and c_j. To measure structural differences between instances, E-step factorizes the conditional joint probability of categories into the product of conditional probabilities of categories as:

$$\begin{aligned} P(c_i, c_j|x_i, x_j;\theta) &= \frac{P(x_i|c_i;\theta)P(c_i;\theta)}{\sum\limits_{c_i} P(x_i|c_i;\theta)P(c_i;\theta)} \frac{P(x_j|c_j;\theta)P(c_j;\theta)}{\sum\limits_{c_j} P(x_j|c_j;\theta)P(c_j;\theta)} \\ &= P(c_i|x_i;\theta)P(c_j|x_j;\theta) \end{aligned} \tag{4}$$

where $P(c_*;\theta)$ is the marginal probability of the category c_*. According to the Eq. (4), there exists a conditional joint probability matrix $\mathbf{P} \in \mathbb{R}^{K\times K}$ in which each element $P(c_i, c_j|x_i, x_j;\theta)$ indicates the probability of a possible elementary event about the value of c_i and c_j. In the matrix $\mathbf{P}$, the diagonal elements indicate the probability that the pairwise instances x_i and x_j belong to the same categories; other elements indicate the probability that pairwise instances are in different categories.

Thus, E-step can utilize the conditional joint probability of categories to capture clustering relationships between instances. In detail, E-step forces the diagonal elements of $\mathbf{P}$ to be one, when the instances x_i and x_j are generated from the same cluster, i.e., x_i and x_j with the must-link. Otherwise, E-step forces the diagonal elements to be zero, i.e., x_i and x_j with the cannot-link. In

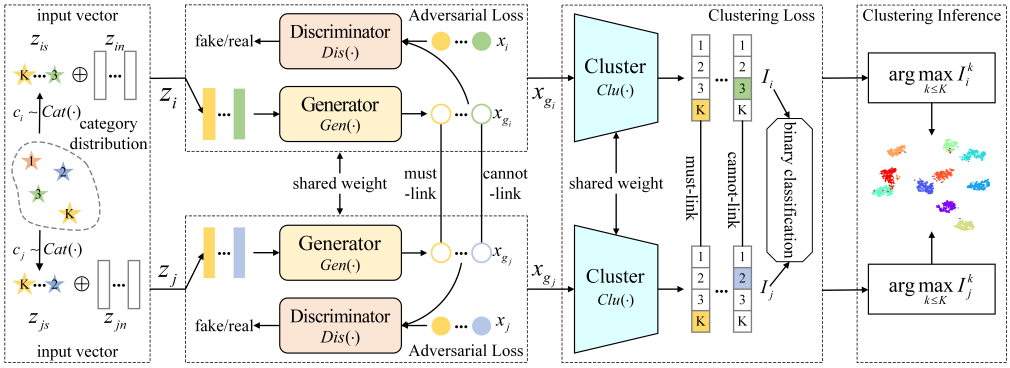


Figure 1: The architecture of DSGACN. Specifically, pairwise input vectors with link information, i.e., z_i and z_j, are sampled from the same category distribution or the different category distributions, and they are input into the two Siamese generator networks to produce pairwise instances with link information, i.e., must-link and cannot-link. Then, the pairwise instances are fed into the two Siamese discriminator networks to perform the adversarial learning with real instances. Meanwhile, they are input into the two Siamese cluster networks to conduct the binary classification of cluster structures with help of the link information. In the clustering inference, the dimension with the maximum activation in the indicator feature is interpreted as the index of the clustering group.

such a manner, E-step strengthens intra-cluster links and weakens inter-cluster links to capture the category-invariant structures of data. Furthermore, it feeds such supervision information back to M-step to enhance the consistencies of data manifolds in fitting data distributions.

M-step maximizes the lower bound of pairwise data log-likelihood to match the joint distribution between categories and instances. In detail, it maximizes the lower bound, as follows (The proof is shown in the Appendix A.2):

$$\max \mathbb{L} \Leftrightarrow \min \sum_{c_i=1}^{K} \mathbb{E}_{P(x_i|c_i;\theta)} \log(1-D) + \mathbb{E}_{P(x_i|c_i)} \log(D) \tag{5}$$

where $P(x_i|c_i;\theta)$ is the distribution of generated data, $P(x_i|c_i)$ denotes the distribution of real data, and D is the probability that data are from the real distribution. The Eq. (5) is equivalent to the adversarial learning between the real data and the generated data. Thus, M-step can utilize the generative adversarial learning to fit the joint distribution, which is flexible enough to model the complex correlations between data.

Overall, the clustering is defined as the novel generation process of pairwise instances with the pairwise data log-likelihood. Then, the expectation-maximization scheme is designed to maximize the pairwise data log-likelihood, where the M-step produces pairwise instances with link information based on the adversarial learning of data distributions and the E-step captures data partitions via utilizing the link information to constrain the conditional joint probability matrix of categories. In other words, the novel clustering utilizes the must-link and the cannot-link to detect the consistent structures in clusters and different structures between clusters in the E-step, respectively, which ensures the capture of invariant structures in clusters. Meanwhile, it leverages the feedback of link information from the E-step to facilitate the compactness of intra-cluster instances and the separation of inter-cluster instances in fitting data distributions, which contributes to the consistencies of data manifolds in the M-step.

4 The Deep Siamese Generative Adversarial Clustering Network

A deep Siamese generative adversarial clustering network (DSGACN) is designed to implement the generation process of pairwise instances within the expectation-maximization scheme, containing the Siamese architecture and the adversarial clustering loss, as shown in Figure 1.

4.1 THE SIAMESE ARCHITECTURE

The Siamese architecture utilizes generator, discriminator, and cluster networks, to learn clustering-specific representations and data partitions in an end-to-end manner. The Siamese architecture produces pairwise instances with link information in accordance with pairwise categories via probabilistic generation functions of the Siamese generator networks, and meanwhile, it improves the generated instances via distinguishing functions of the Siamese discriminator networks in the M-step. Then, it distinguishes whether the pairwise generated instances are in the same cluster via semantic functions of the Siamese cluster networks to learn clustering-specific representations and data partitions in the E-step.

The generator network learns a probabilistic generation function from a noise space to the data space, which can approximate the intractable joint distribution over categories and instances with the help of hidden invariant structures. The probabilistic generation function $Gen\left(\right)$ is defined via:

$$x_{g_i} = Gen\left(z_i; \theta_g\right), z_i = [z_{in}, z_{is}] \tag{6}$$

where x_{g_i} denotes the i-th generated instance, θ_g represents parameters of the generator network, [] is the concatenation of vectors, z_i is the input vector of the latent representation to synthesize the i-th sample, z_{in} is a noise vector to ensure the intra-cluster diversity in each cluster, and z_{is} is a category-static vector sampled from the category c_i, $c_i \in \{1, 2, \cdots, K\}$. z_{is} denotes the hidden invariant structure and is equal for generated instances of the same cluster to endow pairwise instances with the must-link. c_i is subject to the uniform distribution in the network training.

The discriminator network learns a distinguishing function between the real distribution and the synthesized distribution, which extracts information hidden in data distributions for the generator network. The distinguishing function $Dis\left(\right)$ is computed as follows:

$$f = Dis\left(x_i; \theta_d\right), x_i \sim P\left(x\right) \ or \ x_i \sim P\left(x_g\right) \tag{7}$$

where $f \in [0, 1]$ indicates the probability of data from the real distribution, $P\left(x\right)$ is the real data distribution, $P\left(x_g\right)$ is the generated data distribution, and θ_d represents parameters of the discriminator network. x and x_g denote the real data and generated data, respectively.

The cluster network learns a semantic function transforming data into the clustering-oriented space to extract indicator features of data for clustering-specific representation learning and data partitioning. The cluster network $Clu\left(\right)$ is defined as:

$$I_i = Clu\left(x_{g_i}; \theta_c\right) \tag{8}$$

where θ_c represents the parameters. I_i is the indicator feature which is centered around the simplex corner corresponding to the clustering centroid of the instance x_{g_i}. In detail, the indicator feature is expressed as:

$$Cont : \forall i, I_i^k \geq 0, k = 1, 2, \cdots, K \tag{9}$$

where I_i^k is the k-th element of the indicator feature I_i, denoting the probability that the i-th instance belongs to the k-th cluster. $Cont$ denotes the constraint that $\left\|I_i\right\|_1 = \sum_{k=1}^{K} I_i^k = 1$. In experiments, the indicator feature is implemented by the softmax layer.

4.2 THE ADVERSARIAL CLUSTERING LOSS

The adversarial clustering loss is designed in accordance with the expectation-maximization scheme to train the Deep Siamese generative adversarial clustering network. It can capture category-invariant structures and consistencies of manifolds in clustering via utilizing an adversarial loss and a clustering loss to explore link information of pairwise data as:

$$L = L_{al} + L_{cl} \tag{10}$$

where L_{al} is the adversarial loss and L_{cl} is the clustering loss.

The adversarial loss L_{al} instantiates the Eq. (5) to construct the max-min game of distribution between real and generated instances in the M-step, which guides the generator networks to fit the joint distribution between categories and instances with the help of the discriminator networks. Specifically, given the dataset $X = \{x_i\}_{i=1}^N$ and the noise $Z = \{[z_{in}, z_{is}]\}_{i=1}^N$, L_{al} is defined as:

$$\begin{aligned} L_{al} &= \min_{Gen} \max_{Dis} \mathbb{E}_{x_i \sim P(x)} Dis\left(x_i\right) + \mathbb{E}_{x_i \sim P(x_g)} \left[1 - Dis\left(x_i\right)\right] \\ &+ \min_{Gen} \max_{Dis} \mathbb{E}_{x_j \sim P(x)} Dis\left(x_j\right) + \mathbb{E}_{x_j \sim P(x_g)} \left[1 - Dis\left(x_j\right)\right] \end{aligned} \tag{11}$$

where $P(x)$ and $P(x_g)$ represent the distributions of real data and generated data, respectively. The adversarial loss can help the two Siamese generator networks to produce pairwise instances that look like the real ones in the M-step within the deep Siamese generative adversarial clustering architecture. Furthermore, it can ensure that the two Siamese generator networks endow pairwise instances with link information with the help of category-static vectors in the input vectors.

The clustering loss L_{cl} implements the conditional category probability matrix (Eq. 4) of pairwise instances to construct the binary classification task of cluster structures in the E-step. It guides the cluster network to capture the category-invariant structures and impose pairwise link constraints on the likelihood of data. Specifically, given the pairwise generated instances $\left\{\left(x_{g_i}, x_{g_j}\right)\right\}_{i,j}^{N,N}$, the clustering loss utilizes the Cauchy-Schwarz divergence to instantiate the conditional category probability, which measures the structural differences between pairwise instances as:

$$CS(P(c_i|x_{g_i};\theta), P(c_j|x_{g_j};\theta)) = -\log \frac{\sum P(c_i|x_{g_i};\theta)P(c_j|x_{g_j};\theta)}{\sqrt{\sum P(c_i|x_{g_i};\theta)^2 \sum P(c_j|x_{g_j};\theta)^2}} = -\log m(I_i, I_j) \quad (12)$$

where $CS()$ denotes the Cauchy-Schwarz divergence, and $m()$ is the cosine similarity function. $m() \in [0, 1]$, since $P(c_i|x_{g_i};\theta)$ and $P(c_j|x_{g_j};\theta)$ belong to $[0, 1]$. Thus, $m()$ can be interpreted as the probability that x_{g_i} and x_{g_j} belong to the same cluster.

To further strengthen the intra-cluster compactness and inter-cluster separation in mining patterns, the clustering loss Eq. (12) can be redefined based on a binary classification task between intra-cluster structures and inter-cluster structures. In detail, the generator networks try to produce real pairwise instances that follow the must-link and fake pairwise instances that follow the cannot-link in order to confuse the cluster networks. Meanwhile, the cluster networks try to distinguish the real pairwise instances from the fake pairwise instances. The clustering loss L_{cl} is expressed as:

$$\begin{aligned} L_{cl} &= \mathbb{E}_{I_i, I_j \sim P(I|c_i)}[1 - m(I_i, I_j)] + \mathbb{E}_{I_i \sim P(I|c_i), I_j \sim P(I|c_j)} m(I_i, I_j) \\ &= \mathbb{E}_{x_{g_i}, x_{g_j} \sim P(x_g|c_i)}[1 - m(Clu(x_{g_i}), Clu(x_{g_j}))] \\ &+ \mathbb{E}_{x_{g_i} \sim P(x_g|c_i), x_{g_j} \sim P(x_g|c_j)} m(Clu(x_{g_i}), Clu(x_{g_j})) \end{aligned} \quad (13)$$

where $P(I|c_*)$ denotes the distribution of the indicator features of the generated instances from the category c_*, and $P(x_g|c_*)$ denotes the distribution of the generated instances from the category c_*.

After DSGACN converges, the cluster network partitions data into different groups based on the intrinsic similarities of data. To be specific, given the instance x_i, the cluster network interprets the dimension with the maximum activation in the indicator feature as the index of the clustering group:

$$c_i = arg \max_{k \leq K} I_i^k = arg \max_{k \leq K} Clu(x_i; \theta_c) \quad (14)$$

where K denotes the number of predefined clusters and I_i^k is the value of the k-th dimension in the indicator feature of x_i. The training details of DSGAC are outlined in the Appendix A.1.

5 The Experiment and Evaluation

5.1 Experiment Settings

Benchmark datasets: Six benchmark datasets are chosen to validate the model performance. MNIST (LeCun et al., 1998) contains 70000 gray-level images of handwritten digits, which are grouped into 10 categories. CIFAR-10 (Jia et al., 2025) contains 60000 color images which are organized into 10 completely mutually exclusive groups. CIFAR-100 (Jia et al., 2025) contains 60000 color images which are organized into 20 completely mutually exclusive groups. STL-10 (Coates et al., 2011) includes 13000 color images which are grouped into 10 categories. ImageNet-10 (Deng et al., 2009) is composed of 13000 images which are grouped into 10 categories. ImageNet-Dog (Deng et al., 2009) contains 19500 images randomly selected from 15 kinds of dog images.

Clustering metrics: NMI, ARI, and ACC are utilized to illustrate the performance of DSGAC. For the three clustering metrics, the large value indicates the good clustering performance. The results in Table 1 are the average values obtained from 5 independent runs.

Comparison methods: Seventeen representative clustering methods are utilized to evaluate the performance of DSGAC, which can be divided into shallow clustering methods, i.e., k-means (Wang

et al., 2014) and SC (Zelnik-Manor & Perona, 2004), deep representation-based clustering methods, i.e., AE (Bengio et al., 2006), DAE (Vincent et al., 2010), CILR (Chang et al., 2018b), deep discriminative clustering methods, i.e., CDC (Jia et al., 2025), JULE (Yang et al., 2016), DEC (Xie et al., 2016), DSEC (Chang et al., 2018a), PICA (Huang et al., 2020), CC (Li et al., 2021), and SKAN (Bi et al., 2026), and deep generative clustering methods, i.e., VaDE (Jiang et al., 2017), GANEM (Zhao et al., 2019), CGAN (Mukherjee et al., 2019), and ADVAE (Luo et al., 2026). All the experiments follow the setting in the previous articles (Xie et al., 2016). In detail, the shallow clustering uses HOG features and 8×8 color maps of images as input on STL-10, ImageNet-10, and ImageNet-Dog, and uses the pixel intensities of images as input on MNIST, CIFAR-10, and CIFAR-100. The deep clustering utilizes the pixel intensities of images as input on all the datasets. The learning rate and the batch size are set as 0.001 and 256 on all datasets, respectively.

Table 1: The clustering results of the compared methods on the six datasets (%). The best results are marked in bold. The results with $*$ are excerpted from the official articles. The ARI results are provided in Appendix A.3.

Method	MNIST		CIFAR-10		CIFAR-100		STL-10		ImageNet-10		ImageNet-Dog	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
k-means	57.23*	49.97*	22.89*	8.71*	12.97*	8.39*	19.20*	12.45*	24.09*	11.86*	10.54*	5.48*
SC	69.58*	66.26*	24.67*	10.28*	13.60*	9.01*	15.88*	9.78*	27.40*	15.11*	11.11*	3.83*
AE	81.23*	72.57*	31.35*	23.93*	16.45*	10.04*	30.30*	24.96*	31.70*	20.99*	18.51*	10.39*
DAE	83.16*	75.63*	29.71*	25.06*	15.05*	11.05*	30.22*	22.42*	30.44*	20.64*	19.03*	10.43*
CILR	88.55*	79.86*	25.83*	11.05*	15.54*	9.43*	24.52*	13.98*	30.72*	21.35*	18.26*	10.29*
CDC	97.54	94.64	88.64	80.34	50.11	50.32	85.77	84.65	90.32	84.65	44.82	45.81
JULE-S	95.92*	90.63*	26.43*	19.19*	14.34*	9.44*	27.41*	17.54*	29.27*	15.97*	11.15*	4.92*
JULE-R	96.40*	91.30*	27.15*	19.23*	13.67*	10.26*	27.69*	18.15*	30.04*	17.52*	13.77*	5.37*
DEC	84.30*	77.16*	30.10*	25.68*	18.52*	13.58*	35.90*	27.60*	38.09*	28.19*	19.49*	12.16*
DSEC	98.33*	95.22*	47.78*	43.79*	25.52*	21.24*	48.18*	40.33*	67.37*	58.27*	26.44*	23.62*
PICA	94.10	89.80	69.60*	59.10*	33.70*	31.00*	71.30*	61.10*	87.00*	80.20*	35.20*	35.20*
CC	76.52	66.87	79.00*	70.50*	42.90*	43.10*	85.00*	76.40*	89.30*	85.90*	42.90*	44.50*
SKAN	95.06	92.14	82.16	74.12	45.13	46.28	84.27	79.21	90.12	85.17	44.78	43.82
VaDE	94.46*	86.26	86.49	82.79	30.29	36.36	84.45*	75.55	85.72	85.75	35.81	40.36
GANEM	95.80*	90.08	79.83	78.65	29.81	17.32	50.98	67.43	73.94	66.32	19.32	33.28
CGAN	95.00*	89.00*	79.58	77.42	53.24	52.96	53.63	60.58	89.15	85.12	42.66	41.66
ADVAE	96.21	90.57	85.67	80.15	48.79	49.86	85.69	83.15	90.94	85.86	45.26	45.47
DSGAC	**98.75**	**96.44**	**90.03**	**82.84**	**54.55**	**55.58**	**87.46**	**87.82**	**91.75**	**86.28**	**46.13**	**47.71**

5.2 The Numerical Results of Clustering

Table 1 shows the numerical results of eighteen models on six datasets. As shown in Table 1, DSGAC achieves the best clustering performance in terms of NMI, ARI, and ACC, and it outperforms the state-of-the-art methods by a significant margin on most of the six datasets. There are two essential reasons. 1) DSGAC ensures the consistencies of manifolds in fitting the distribution of data via forcing the two Siamese generator networks to generate pairwise instances subject to the link information of pairwise categories. 2) DSGAC can well capture category-invariant structures of data based on the adversarial game between pairwise instances of the same cluster and pairwise instances of different clusters in the two Siamese cluster networks.

In addition, there are three other observations on the results of Table 1. Firstly, the deep clustering produces better results than the shadow clustering. This is because the deep clustering leverages deep neural networks to capture non-linear correlations between data, which are more informative than linear correlations captured by the shallow clustering. Secondly, the deep end-to-end clustering, i.e., deep discriminative clustering and deep generative clustering, achieves more accurate results than the deep representation-based clustering. This observation implies that the deep end-to-end clustering can learn more structure information from data than the deep representation-based clustering via merging clustering-specific constraints into the training of deep neural networks. Thirdly, the deep generative clustering produces similar values to deep discriminative clustering in most cases, since both methods can extract structural information of data in the training of deep neural networks.

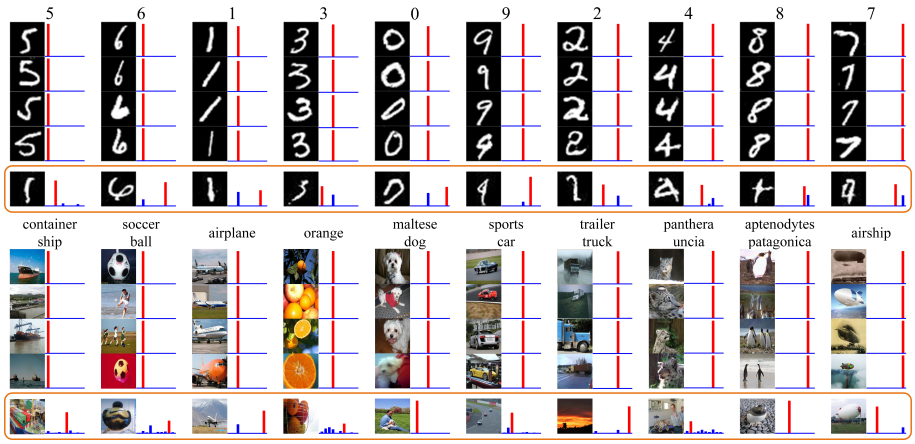
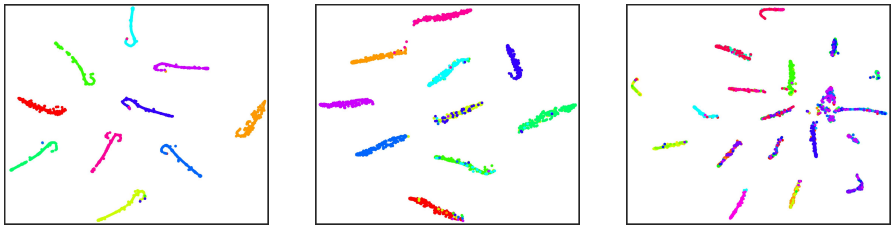


Figure 2: The visualization of indicator features on MNIST and ImageNet-10. The ground-truth labels are written on the upward side, and indicator features are shown on the right side of instances, followed by typical failures at the bottom. The visualizations of indicator features on the other four datasets are shown in Appendix A.3.



(a) MNIST (b) CIFAR-10 (c) CIFAR-100

Figure 3: The t-SNE visualizations of DSGAC on three representative datasets. The visualization results on the remaining three datasets are provided in Appendix A.3.

5.3 The Visualization Results of Clustering

To further validate the performance of DSGAC, the visualization clustering results of six datasets are illustrated in Figure 2 and Figure 3. Figure 2 demonstrates the indicator features of data in the cluster network. As shown in the results, data of the same class activate the same element of indicator features, and there is only one element with high value in indicator features. The results imply that indicator features can well recognize the categories of data. Figure 3 shows t-SNE visualizations of clustering results in the 2-dimensional space. In detail, 2000 instances are randomly selected from each of the three datasets, and input into the t-SNE algorithm. In the t-SNE visualization results, data of different clusters are assigned different colors, and there are few overlaps between different clusters. The results in t-SNE visualizations also illustrate that DSGAC can ensure the intra-cluster compactness and inter-cluster separation in pattern mining. The above visualization results validate the clustering performance of DSGAC.

6 Conclusion

In this paper, a new clustering scheme is proposed via defining a generation process of pairwise data from category variables to instance variables. To implement the clustering scheme, a deep Siamese generative adversarial clustering network is introduced within a Siamese architecture composed of generator, discriminator, and cluster networks. Then, an adversarial clustering loss is devised on the basis of the adversarial game of data distributions and the classification task of cluster structures to train network parameters. Finally, experiments on six benchmark datasets illustrate the significant improvement in comparison with seventeen state-of-the-art clustering models.

AI USE STATEMENT

No generative AI tools are used in any stage of this work, including research design, methodology development, experimental implementation, data analysis, and manuscript preparation. All content is independently produced and reviewed by the authors. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide the model weights trained on the MNIST dataset to facilitate the reproduction of our results in the Supplementary Material. The whole code will be presented after the paper has been accepted.

REFERENCES

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19, 2006.

Zixuan Bi, Yang Zhao, and Ganchao Liu. Deep clustering based on sparse kolmogorov-arnold network and spectral constraint. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19720–19727, 2026.

Jianlong Chang, Gaofeng Meng, Lingfeng Wang, Shiming Xiang, and Chunhong Pan. Deep self-evolution clustering. *IEEE transactions on pattern analysis and machine intelligence*, 42(4): 809–823, 2018a.

Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep unsupervised learning with consistent inference of latent representations. *Pattern Recognition*, 77: 438–453, 2018b.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Daniel PM de Mello, Renato M Assunção, and Fabricio Murai. Top-down deep clustering with multi-generator gans. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7770–7778, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE, 2009.

Jing Gao, Meng Liu, Peng Li, Jianing Zhang, and Zhikui Chen. Deep multiview adaptive clustering with semantic invariance. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9): 12965–12978, 2024.

Xifeng Guo, Xinwang Liu, En Zhu, Xinzhong Zhu, Miaomiao Li, Xin Xu, and Jianping Yin. Adaptive self-paced deep clustering with data augmentation. *IEEE Transactions on Knowledge and Data Engineering*, 32(9):1680–1693, 2020.

Jiazhen Hong, Wei Qian, Yudong Chen, and Yuqian Zhang. A geometric approach to k-means clustering. *IEEE Transactions on Knowledge and Data Engineering*, 38(3):1442–1453, 2026.

Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8846–8855. IEEE, 2020.

Yuheng Jia, Jianhong Cheng, Hui Liu, and Junhui Hou. Towards calibrated deep clustering network. In *International Conference on Learning Representations*, volume 2025, pp. 102968–102992, 2025.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised generative approach to clustering. In *26th International Joint Conference on Artificial Intelligence*, pp. 1965–1972. International Joint Conferences on Artificial Intelligence, 2017.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Peng Li, Jing Gao, Jianing Zhang, Shan Jin, and Zhikui Chen. Deep reinforcement clustering. *IEEE Transactions on Multimedia*, 25:8183–8193, 2022.

Xuelong Li, Rui Zhang, Qi Wang, and Hongyuan Zhang. Autoencoder constrained clustering with adaptive neighbors. *IEEE transactions on neural networks and learning systems*, 32(1):443–449, 2020.

Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8547–8555, 2021.

Zhiwen Luo, Wentao Fan, Manar Amayri, and Nizar Bouguila. Adaptive deep clustering via disentangled hyperspherical vaes with contrastive geometry optimization. *Neurocomputing*, pp. 132625, 2026.

Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4610–4617, 2019.

Samuel V Singh, Shirley Coyle, and Mimi Zhang. Shape-informed clustering of multi-dimensional functional data via deep functional autoencoders. *Advances in Neural Information Processing Systems*, 38:161762–161796, 2026.

Georgios Vardakas, Ioannis Papakostas, and Aristidis Likas. Deep clustering using the soft silhouette score: Towards compact and well-separated clusters. *Machine Learning*, 115(4):81, 2026.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

Jianfeng Wang, Jingdong Wang, Jingkuan Song, Xin-Shun Xu, Heng Tao Shen, and Shipeng Li. Optimized cartesian k-means. *IEEE Transactions on Knowledge and Data Engineering*, 27(1): 180–192, 2014.

Jinghua Wang and Jianmin Jiang. Unsupervised deep clustering via adaptive gmm modeling and optimization. *Neurocomputing*, 433:199–211, 2021.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487. PMLR, 2016.

Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5147–5156, 2016.

Lin Yang, Wentao Fan, and Nizar Bouguila. Clustering analysis via deep generative models with mixture models. *IEEE Transactions on Neural Networks and Learning Systems*, 33(1):340–350, 2020.

Zhiwen Yu, Zhongfan Zhang, Wenming Cao, Cheng Liu, CL Philip Chen, and Hau-San Wong. Gan-based enhanced deep subspace clustering networks. *IEEE Transactions on Knowledge and Data Engineering*, 34(7):3267–3281, 2022.

Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. *Advances in neural information processing systems*, 17, 2004.

Junjian Zhang, Chun-Guang Li, Chong You, Xianbiao Qi, Honggang Zhang, Jun Guo, and Zhouchen Lin. Self-supervised convolutional subspace clustering network. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5468–5477. IEEE, 2019.

Wentian Zhao, Shaojie Wang, Zhihuai Xie, Jing Shi, and Chenliang Xu. Gan-em: Gan based em learning framework. *International Joint Conference on Artificial Intelligence*, pp. 4404–4411, 2019.

Zi-Feng Zhou, Dong Huang, and Chang-Dong Wang. Pyramid contrastive learning for clustering. *Neural Networks*, 185:107217, 2025.

A Appendix

A.1 Algorithm

The training details of DSGAC are outlined in Algorithm 1.

Algorithm 1 Deep Siamese Generative Adversarial Clustering

Input: Input dataset $X = \{x_i\}_{i=1}^{N}$, predefined clusters K, and training epochs n.
Randomly initialize parameters $\theta = \{\theta_g, \theta_d, \theta_c\}$;
for $t = 1, ...n$ **do**
E-step:
Sample pairwise input vectors z_i and z_j from the category distributions;
Generate pairwise instances:
$x_{g_i} = Gen(z_i; \theta_g)$, $x_{g_j} = Gen(z_j; \theta_g)$;
Calculate clustering assignment:
$I_i^k = Clu(x_{g_i}; \theta_c)$, $I_j^k = Clu(x_{g_j}; \theta_c)$,
$c_i = arg\max_{k \le K} I_i^k$, $c_j = arg\max_{k \le K} I_j^k$;
Calculate the adversarial clustering loss L;
Update cluster network parameters θ_c by ∇L;
Update generator network parameters θ_g by ∇L;
M-step:
Sample pairwise input vectors z_i and z_j from the category distributions;
Generate pairwise instances:
$x_{g_i} = Gen(z_i; \theta_g)$, $x_{g_j} = Gen(z_j; \theta_g)$;
Distinguish real data and generated data:
$f = Dis(x; \theta_d), x \in \{x_i, x_{g_i}\, x_j, x_{g_j}\}$;
Calculate the adversarial clustering loss L;
Update discriminator network parameters θ_d by ∇L;
Update generator network parameters θ_g by ∇L;
end for
for $x_i \in X$ **do**
$c_i = arg\max_{k \le K} Clu(x_i; \theta_c)$;
end for
Output: Clustering assignment $\{c_i\}_{i=1}^{N}$.

A.2 Theoretical Proof

Proposition 1 *The pairwise log-likelihood with hidden category variables can be maximized by the lower bound as:*

$$\mathbb{L} = \iint \mathbb{E}_{P(c_i, c_j | x_i, x_j; \theta)} \log P(x_i, x_j, c_i, c_j; \theta) dx_i dx_j \tag{15}$$

where $\mathbb{E}$ is the expectation operation, $P(x_i, x_j, c_i, c_j; \theta)$ is the joint probability between categories and instances, and $P(c_i, c_j | x_i, x_j; \theta)$ is the conditional probability of categories given x_i and x_j.

Proof 1 *The hidden category variables are introduced into the pairwise log-likelihood as:*

$$\mathbb{L} = \iint (\sum_{c_i=1,c_j=1}^{K,K} P(c_i, c_j|x_i, x_j; \theta)) \log P(x_i, x_j; \theta) dx_i dx_j \tag{16}$$

When x_i and x_j are known, $\log P(x_i, x_j; \theta)$ is a constant. The inner of Eq. (16) can be expanded as:

$$\mathbb{L} = \iint (\sum_{c_i=1,c_j=1}^{K,K} P(c_i, c_j|x_i, x_j; \theta) \log P(x_i, x_j; \theta)) dx_i dx_j \tag{17}$$

and it is further re-written as the expectation operation, as follows:

$$\mathbb{L} = \iint \mathbb{E}_{P(c_i,c_j|x_i,x_j;\theta)} \log P(x_i, x_j; \theta) dx_i dx_j \tag{18}$$

Then, $\log P(x_i, x_j; \theta)$ is split into two parts according to the definition of conditional probability:

$$\begin{aligned} \mathbb{L} = & \underbrace{\iint \mathbb{E}_{P(c_i,c_j|x_i,x_j;\theta)} \log P(x_i, x_j, c_i, c_j; \theta) dx_i dx_j}_{\textit{Lower Bound}} \\ & - \underbrace{\iint \mathbb{E}_{P(c_i,c_j|x_i,x_j;\theta)} \log P(c_i, c_j|x_i, x_j; \theta) dx_i dx_j}_{\textit{Negative Entropy}} \end{aligned} \tag{19}$$

In the Eq. (19), the second term is the negative entropy that is less than zero, and the first term is the lower bound of the pairwise likelihood. Thus, the maximization of the pairwise likelihood is equivalent to the maximization of the lower bound, i.e., Eq. (15).

Proposition 2 *The optimization of $\mathbb{L}$ is equivalent to an adversarial learning process between real data and generated data as:*

$$\max \mathbb{L} \Leftrightarrow \min_G \max_D \sum_{c_i=1}^{K} \mathbb{E}_{P(x_i|c_i;\theta)} \log(1 - D) + \mathbb{E}_{P(x_i|c_i)} \log(D) \tag{20}$$

where $P(x_i|c_i; \theta)$ is the distribution of generated data x_i modelled by the generator G, $P(x_i|c_i)$ the distribution of real data, and D is the output of the discriminator, representing the probability that data are from the real distribution.

Proof 2 *The data likelihood $\mathbb{L}$ is split into two terms in accordance with the definition of the condition probability, as follows:*

$$\begin{aligned} \mathbb{L} &= \iint \mathbb{E}_{P(c_i,c_j|x_i,x_j;\theta)} \log P(x_i, x_j, c_i, c_j; \theta) dx_i dx_j \\ &= \iint \mathbb{E}_{P(c_i,c_j|x_i,x_j;\theta)} \log P(c_i, c_j; \theta) dx_i dx_j \\ &+ \iint \mathbb{E}_{P(c_i,c_j|x_i,x_j;\theta)} \log P(x_i, x_j|c_i, c_j; \theta) dx_i dx_j \end{aligned} \tag{21}$$

where $P(c_i, c_j; \theta) = \frac{1}{K^2}$ for each pair of c_i, c_j in the uniform sampling. Hence,the first term of $\mathbb{L}$ (Eq. (21)) can be omitted in the optimization process, and $\mathbb{L}$ is rewritten in the form of the negative cross entropy between real and generated distributions of data:

$$
\begin{aligned}
\mathbb{L} &= \iint \mathbb{E}_{P(c_i,c_j|x_i,x_j;\theta)} \log P(x_i,x_j|c_i,c_j;\theta) dx_i dx_j \\
&= \iint \sum_{c_i=1,c_j=1}^{K,K} P(c_i|x_i;\theta)P(c_j|x_j;\theta) \log(P(x_i|c_i;\theta)P(x_j|c_j;\theta)) dx_i dx_j \\
&= \int \sum_{c_i=1}^{K} \int \sum_{c_j=1}^{K} P(c_j|x_j;\theta) dx_j P(c_i|x_i;\theta) \log P(x_i|c_i;\theta) dx_i \\
&+ \int \sum_{c_j=1}^{K} \int \sum_{c_i=1}^{K} P(c_i|x_i;\theta) dx_i P(c_j|x_j;\theta) \log P(x_j|c_j;\theta) dx_j \\
&= 2N \sum_{c_i=1}^{K} \int P(c_i|x_i;\theta) \log P(x_i|c_i;\theta) dx_i \\
&= 2N^2 \sum_{c_i=1}^{K} \int P(x_i) P(c_i|x_i;\theta) \log P(x_i|c_i;\theta) dx_i \\
&= 2N^2 \sum_{c_i=1}^{K} \int P(c_i) P(x_i|c_i) \log P(x_i|c_i;\theta) dx_i \\
&= 2N^2/K \sum_{c_i=1}^{K} \int P(x_i|c_i) \log P(x_i|c_i;\theta) dx_i \\
&= -2N^2/K \sum_{c_i=1}^{K} CE(P(x_i|c_i)||P(x_i|c_i;\theta))
\end{aligned}
\tag{22}
$$

where $CE(||)$ represents the cross entropy between distributions. In the optimization, there exist the following equivalence relations:

$$
\begin{aligned}
&\max \mathbb{L} \Leftrightarrow \min_G \sum_{c_i=1}^{K} CE(P(x_i|c_i)||P(x_i|c_i;\theta)) \\
&\Leftrightarrow \min_G \sum_{c_i=1}^{K} JS(P(x_i|c_i)||P(x_i|c_i;\theta)) \\
&\Leftrightarrow \min_G 2 \sum_{c_i=1}^{K} JS(P(x_i|c_i)||P(x_i|c_i;\theta)) - \log 4 \\
&= \min_G \sum_{c_i=1}^{K} KL(P(x_i|c_i;\theta)||\frac{P(x_i|c_i;\theta)+P(x_i|c_i)}{2}) + \sum_{c_i=1}^{K} KL(P(x_i|c_i)||\frac{P(x_i|c_i;\theta)+P(x_i|c_i)}{2}) \\
&= \min_G \sum_{c_i=1}^{K} \mathbb{E}_{P(x_i|c_i;\theta)} \log \frac{P(x_i|c_i;\theta)}{P(x_i|c_i;\theta)+P(x_i|c_i)} + \sum_{c_i=1}^{K} \mathbb{E}_{P(x_i|c_i)} \log \frac{P(x_i|c_i)}{P(x_i|c_i;\theta)+P(x_i|c_i)}
\end{aligned}
\tag{23}
$$

where the last line in Eq. (23) is equivalent to the adversarial learning process: $\min_G \max_D \sum_{c_i=1}^{K} \mathbb{E}_{P(x_i|c_i;\theta)} \log(1-D) + \mathbb{E}_{P(x_i|c_i)} \log(D)$ *with optimal discriminator* $D =$

$\frac{P(x_i|c_i)}{P(x_i|c_i;\theta)+P(x_i|c_i)}$, *as shown in Eq. (24):*

$$\begin{aligned} & \min_G \max_D \sum_{c_i=1}^{K} \mathbb{E}_{P(x_i|c_i;\theta)} \log(1-D) + \mathbb{E}_{P(x_i|c_i)} \log(D) \\ = \quad & \min_G \sum_{c_i=1}^{K} \mathbb{E}_{P(x_i|c_i;\theta)} \log \frac{P(x_i|c_i;\theta)}{P(x_i|c_i;\theta)+P(x_i|c_i)} \\ + \quad & \sum_{c_i=1}^{K} \mathbb{E}_{P(x_i|c_i)} \log \frac{P(x_i|c_i)}{P(x_i|c_i;\theta)+P(x_i|c_i)} \end{aligned} \tag{24}$$

Hence, the optimization of $\mathbb{L}$ *is equivalent to an adversarial learning process between real data and generated data:*

$$\max \mathbb{L} \Leftrightarrow \min_G \max_D \sum_{c_i=1}^{K} \mathbb{E}_{P(x_i|c_i;\theta)} \log(1-D) + \mathbb{E}_{P(x_i|c_i)} \log(D) \tag{25}$$

A.3 Experiment and Evaluation

Benchmark datasets: Six benchmark datasets are chosen to validate the model performance, and the statistic details are shown in Table 2.

- MNIST contains 70000 gray-level images of handwritten digits, which have been size-normalized and centered in 28 $\times$ 28 dimensions. The images are grouped into 10 categories.
- CIFAR-10 contains 60000 color images of 32 $\times$ 32 $\times$ 3 dimensions. The images are organized into 10 completely mutually exclusive groups, such as airplane and truck.
- CIFAR-100 contains 60000 color images of 32 $\times$ 32 $\times$ 3 dimensions. The images are organized into 20 completely mutually exclusive groups, such as cat, dog, and bird.
- STL-10 includes 13000 color images of 96 $\times$ 96 $\times$ 3 dimensions. The images are grouped into 10 categories and of a complex distribution.
- ImageNet-10 is an image dataset which contains 10 categories randomly chosen from ILSVRC2012-1K. It is composed of 13000 images resized to 96 $\times$ 96 $\times$ 3.
- ImageNet-Dog contains 19500 images randomly selected from 15 kinds of dog images in ILSVRC2012-1K, and the dog images are resized to 96 $\times$ 96 $\times$ 3.

Table 2: The details of datasets in the experiments.

Dataset	Number	Clusters	Dimensions
MNIST	70000	10	784
CIFAR-10	60000	10	3072
CIFAR-100	60000	20	3072
STL-10	13000	10	27648
ImageNet-10	13000	10	27648
ImageNet-Dog	19500	15	27648

Clustering metrics: Three representative clustering metrics are utilized in the experiments to illustrate the performance of DSGAC.

- Normalized Mutual Information (NMI) is used to measure dependence between the clustering-inference and ground-truth distributions of data.
- Adjusted Rand Index (ARI) is used to compute similarities between the clustering-inference and ground-truth clusters.

- Accuracy (ACC) is used to compute the ratio of the instances in correct clusters to the total instances.

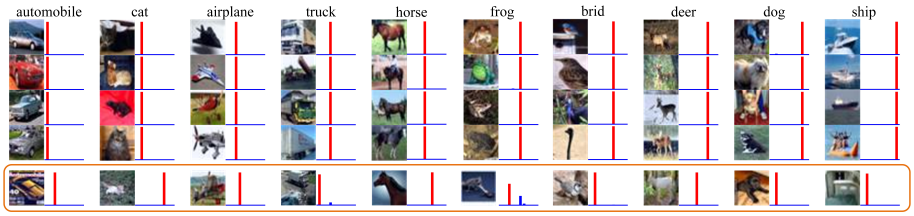
For the three clustering metrics, the large value indicates the good clustering performance.

Experiment results: Table 3 shows the numerical results of eighteen models on six datasets in terms of ACC, NMI, and ARI. In Table 3, the results marked with $*$ are excerpted from the official articles, and the best clustering results are highlighted in bold. Figure 4 presents the visualizations of indicator features on four representative datasets, including CIFAR-10, CIFAR-100, STL-10, and ImageNet-Dog. Figure 5 shows the t-SNE visualization results of DSGAC on the remaining three datasets, including STL-10, ImageNet-10, and ImageNet-Dog.

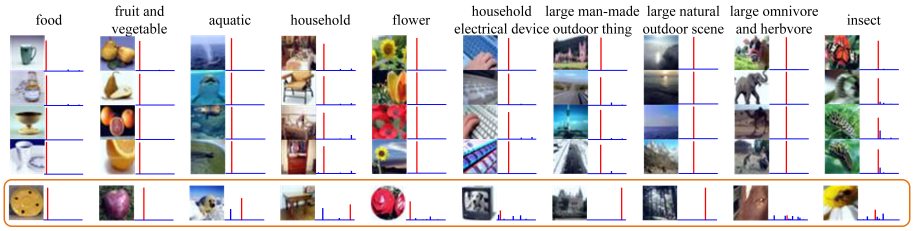
Table 3: The clustering results of the compared methods on the six datasets (%). The best results are marked in bold. The results with $*$ are excerpted from the official articles.

Dataset	MNIST			CIFAR-10			CIFAR-100		
Metric	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC
k-means	49.97*	36.52*	57.23*	8.71*	4.87*	22.89*	8.39*	2.80*	12.97*
SC	66.26*	52.14*	69.58*	10.28*	8.53*	24.67*	9.01*	2.18*	13.60*
AE	72.57*	61.39*	81.23*	23.93*	16.89*	31.35*	10.04*	4.76*	16.45*
DAE	75.63*	64.67*	83.16*	25.06*	16.27*	29.71*	11.05*	4.60*	15.05*
CILR	79.86*	77.56*	88.55*	11.05*	6.12*	25.83*	9.43*	4.53*	15.54*
CDC	94.64	96.70	97.54	80.34	78.73	88.64	50.32	35.74	50.11
JULE-S	90.63*	91.39*	95.92*	19.19*	13.57*	26.43*	9.44*	3.52*	14.34*
JULE-R	91.30*	92.70*	96.40*	19.23*	13.77*	27.15*	10.26*	3.27*	13.67*
DEC	77.16*	74.14*	84.30*	25.68*	16.07*	30.10*	13.58*	4.95*	18.52*
DSEC	95.22*	96.31*	98.33*	43.79*	33.99*	47.78*	21.24*	11.00*	25.52*
PICA	89.80	87.80	94.10	59.10*	51.20*	69.60*	31.00*	17.10*	33.70*
CC	66.87	61.11	76.52	70.50*	63.70*	79.00*	43.10*	26.60*	42.90*
SKAN	92.14	93.48	95.06	74.12	68.25	82.16	46.28	29.17	45.13
VaDE	86.26	84.22	94.46*	82.79	78.22	86.49	36.36	19.42	30.29
GANEM	90.08	89.35	95.80*	78.65	76.13	79.83	17.32	19.18	29.81
CGAN	89.00*	89.00*	95.00*	77.42	67.43	79.58	52.96	37.03	53.24
ADVAE	90.57	91.53	96.21	80.15	74.89	85.67	49.86	33.84	48.79
DSGAC	**96.44**	**97.23**	**98.75**	**82.84**	**80.50**	**90.03**	**55.58**	**39.62**	**54.55**

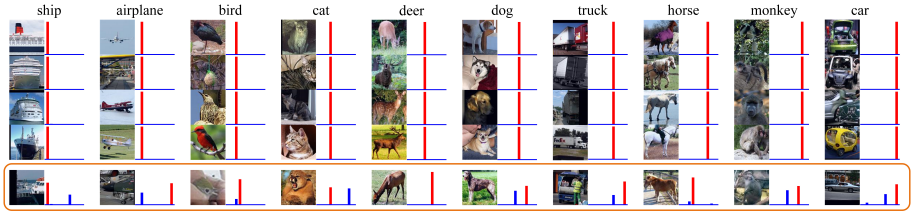
Dataset	STL-10			ImageNet-10			ImageNet-Dog		
Metric	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC
k-means	12.45*	6.08*	19.20*	11.86*	5.71*	24.09*	5.48*	2.04*	10.54*
SC	9.78*	4.79*	15.88*	15.11*	7.57*	27.40*	3.83*	1.33*	11.11*
AE	24.96*	16.10*	30.30*	20.99*	15.16*	31.70*	10.39*	7.28*	18.51*
DAE	22.42*	15.19*	30.22*	20.64*	13.76*	30.44*	10.43*	7.79*	19.03*
CILR	13.98*	6.52*	24.52*	21.35*	13.57*	30.72*	10.29*	7.03*	18.26*
CDC	84.65	75.72	85.77	84.65	83.42	90.32	45.81	30.92	44.82
JULE-S	17.54*	16.22*	27.41*	15.97*	12.05*	29.27*	4.92*	2.61*	11.15*
JULE-R	18.15*	16.43*	27.69*	17.52*	13.82*	30.04*	5.37*	2.84*	13.77*
DEC	27.60*	18.61*	35.90*	28.19*	20.31*	38.09*	12.16*	7.88*	19.49*
DSEC	40.33*	28.64*	48.18*	58.27*	52.18*	67.37*	23.62*	12.41*	26.44*
PICA	61.10*	53.10*	71.30*	80.20*	76.10*	87.00*	35.20*	20.10*	35.20*
CC	76.40*	72.60*	85.00*	85.90*	82.20*	89.30*	44.50*	27.40*	42.90*
SKAN	79.21	70.48	84.27	85.17	81.84	90.12	43.82	28.65	44.78
VaDE	75.55	72.23	84.45*	85.75	79.10	85.72	40.36	23.84	35.81
GANEM	67.43	34.92	50.98	66.32	63.10	73.94	33.28	30.22	19.32
CGAN	60.58	68.45	53.63	85.12	78.91	89.15	41.66	22.51	42.66
ADVAE	83.15	73.64	85.69	85.86	83.71	90.94	45.47	30.48	45.26
DSGAC	**87.82**	**76.25**	**87.46**	**86.28**	**85.27**	**91.75**	**47.71**	**32.20**	**46.13**



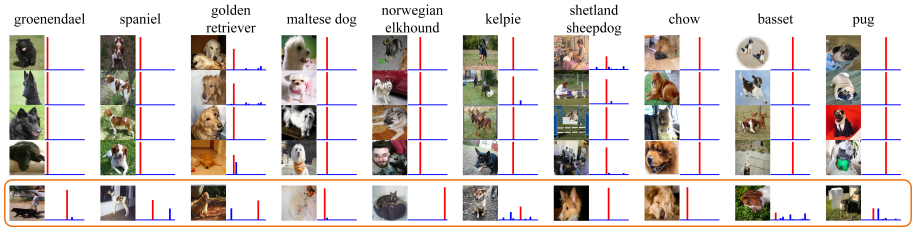
(a) CIFAR-10



(b) CIFAR-100

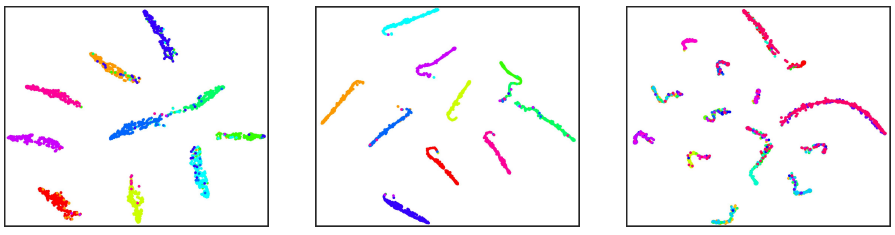


(c) STL-10



(d) ImageNet-Dog

Figure 4: The visualizations of indicator features on four representative datasets.



(a) STL-10 (b) ImageNet-10 (c) ImageNet-Dog

Figure 5: The t-SNE visualizations of DSGAC on three representative datasets.