electronics

Article

Self-Supervised Clustering Models Based on BYOL Network Structure

Xuehao Chen [1,2], Jin Zhou [1,2,*], Yuehui Chen [1,*], Shiyuan Han [1], Yingxu Wang [1,2], Tao Du [1], Cheng Yang [1,2] and Bowen Liu [1,2]

[1] Shandong Provincial Key Laboratory of Network-Based Intelligent Computing, University of Jinan, Jinan 250022, China
[2] Quancheng Laboratory, Jinan 250103, China
* Correspondence: ise_zhouj@ujn.edu.cn (J.Z.); yhchen@ujn.edu.cn (Y.C.)

Abstract: Contrastive-based clustering models usually rely on a large number of negative pairs to capture uniform representations, which requires a large batch size and high computational complexity. In contrast, some self-supervised methods perform non-contrastive learning to capture discriminative representations only with positive pairs, but suffer from the collapse of clustering. To solve these issues, a novel end-to-end self-supervised clustering model is proposed in this paper. The basic self-supervised learning network is first modified, followed by the incorporation of a Softmax layer to obtain cluster assignments as data representation. Then, adversarial learning on the cluster assignments is integrated into the methods to further enhance discrimination across different clusters and mitigate the collapse between clusters. To further encourage clustering-oriented guidance, a new cluster-level discrimination is assembled to promote clustering performance by measuring the self-correlation between the learned cluster assignments. Experimental results on real-world datasets exhibit better performance of the proposed model compared with the existing deep clustering methods.

Keywords: deep clustering; contrastive learning; self-supervised learning; adversarial learning; self-correlation

check for updates

Citation: Chen, X.; Zhou, J.; Chen, Y.; Han, S.; Wang, Y.; Du, T.; Yang, C.; Liu, B. Self-Supervised Clustering Models Based on BYOL Network Structure. *Electronics* **2023**, *12*, 4723. https://doi.org/10.3390/electronics12234723

Academic Editor: Ping-Feng Pai

Received: 17 October 2023
Revised: 18 November 2023
Accepted: 19 November 2023
Published: 21 November 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As an effective machine learning technique, clustering plays an important role in data mining [1–3], statistical analysis [4–6], and pattern recognition [7–9]. It aims to partition the data into different clusters according to the similarity between the data samples [10]. Therefore, various clustering methods have been developed over the past decades to extract the inherent features and structures of the data [11,12]. In the current era of big data, more and more high-dimensional data pose huge challenges to traditional clustering due to insufficient representability. For this reason, some dimensionality reduction [13] and representation transformation [14] techniques have been widely studied to map the original data into a new feature space, where the data representation is easier to separate by the existing classifiers. Nevertheless, limited to their high computational complexity, the traditional data transformation methods [15–17] fail to process large-scale and high-dimensional data. Although some random feature [18] methods and random projection [19] methods can yield a low-dimensional representation and a better approximation of user-specified kernel, the representation ability of features learned from these shallow models is generally limited.

In recent decades, deep learning [20] based on neural networks has been widely studied to discover good representation of the data. Meanwhile, the optimization of the deep neural network along with unsupervised clustering has exhibited great promise and excellent clustering performance, which is referred to as deep clustering [21]. Most deep clustering methods can be categorized as either generative models [22] or discriminative

models [23]. Generative models aim to learn the embedding representation or distribution of the original data through the generative process. The clustering then processes the learned distribution or representation in a simultaneous (end-to-end) or asynchronous fashion. Some of the prominent techniques that have a significant impact are deep clustering methods based on autoencoder (AE) [24], deep clustering methods based on variational autoencoder (VAE) [25], and deep clustering methods based on generative adversarial network (GAN) [26]. However, these clustering methods, which rely on generative models, necessitate complex data generation procedures, which can be computationally expensive and may not be necessary for both clustering and representation learning purposes.

Different from generative models, discriminative models, such as contrastive learning-based methods, remove the costly generation step and directly discriminate the representation by learning the decision boundary. As the most representative contrastive learning method, Simple Framework for Contrastive Learning of Representations (SimCLR) [27] exploits the representation between different views of samples, wherein the similarities between different views of one sample (positive pairs) are maximized and those between different samples (negative pairs) are minimized. Based on this idea, some two-step clustering models have been designed. Supervised Contrastive Learning for Pretrained Visual Representations (SCAN) [28] mines the nearest neighbors of each image as prior guidance to optimize the cluster network, while Semantic Pseudo-labeling for Image Clustering (SPICE) [29] and Robust learning for Unsupervised Clustering (RUC) [30] generate pseudo-labels via self-learning methods to guide the clustering. These methods employed a two-stage operation where the clustering and the representation learning were decoupled. They focus more on the optimization of the neural networks to achieve more discriminative representations but suffer from a lack of clustering-oriented guidance, which results in suboptimal clustering performance.

Recently, more contrastive learning-based models have been constructed to excavate representation and perform clustering in an end-to-end fashion. Among these methods, Contrastive Clustering (CC) [31] performs both instance-level contrastive learning for exploiting the discriminative representations and clustering-level contrastive learning for separating different clusters. Following this idea, Graph Contrastive Clustering (GCC) [32] proposes a graph Laplacian-based contrastive loss to enhance the discriminative and clustering-specific characteristics of features. To further improve the quality of learned representations, Cross-instance guided Contrastive Clustering (C3) [33] takes into account the cross-sample relationships, thereby increasing the number of positive pairs and reducing the impact of false negatives. Even though the contrastive models above yield excellent clustering results, they usually rely on a large number of negative pairs to capture the uniform representations, which requires a large batch size and high computational complexity. Moreover, different instances from the same cluster are regarded as negative pairs and wrongly pushed away, which may inevitably lead to the cluster collision issue.

Different from these traditional contrastive learning-based models, some self-supervised methods, such as Bootstrap Your Own Latent (BYOL) [34], perform non-contrastive learning to capture discriminative representations only with positive pairs. However, the absence of negative pairs in contrastive learning hinders the ability of self-supervised representation learning methods to achieve uniform representations across clusters, which may lead to the issue of the collapse of clustering [35], i.e., assigning all data samples into fewer clusters than desired. Therefore, it is crucial to introduce an effective clustering enhancement method to improve the quality of the clustering assignment.

To solve these issues, a novel end-to-end Self-supervised Clustering model based on BYOL network structure with Instance-level and Cluster-level discriminations (BSC-IC) is proposed in this paper to perform clustering and representation learning simultaneously only with positive pairs. Taking inspiration from the concept of "cluster assignments as representations" [36], we enhance the original BYOL network by incorporating a Softmax layer to convert representations into cluster assignments. Subsequently, we also integrate adversarial learning [37] into cluster assignments not only to improve discrimination

among clusters but also to mitigate the issue of collapsed clusters. To mitigate the high interdependence between the target and online networks in BYOL, we propose a novel self-enhancement loss. This loss evaluates the similarity of cluster assignments among positive pairs within a mini-batch across the online network itself. To further enhance the clustering-oriented guidance. A new cluster-level discrimination is integrated into the discriminative network to promote clustering performance by measuring the self-correlation between the learned cluster assignments.

The rest of this paper is organized as follows. The related work is presented in Section 2. The contrastive clustering model with instance-level and cluster-level discrimination is designed in Section 3. Experiments are performed in Section 4. The ablation study and its analysis are provided in Section 5. Conclusions are given in Section 6.

2. Related Work

2.1. Contrastive Clustering

CC [31] is a contrastive learning-based clustering method that aims to discover meaningful groups or patterns in a given dataset by emphasizing the dissimilarity or contrast between data points. In CC, instance-level and cluster-level contrastive learning are respectively conducted in the row and column spaces by maximizing the similarities of positive pairs while minimizing those of negative ones. However, this method usually relies on a large number of negative pairs to capture the uniform representations, which requires a large batch size and high computational complexity.

2.2. Bootstrap Your Own Latent

BYOL [34] is a self-supervised deep learning method used for representation learning. It is designed to learn meaningful representations from unlabeled data, allowing the model to capture useful patterns and information without the need for negative samples. BYOL consists of two identical neural networks called the online network and the target network. From an augmented view of a data sample, BYOL trains the online network to predict the representation of the target network from a different augmented view of the same data sample.

3. BSC with Instance-Level and Cluster-Level Discriminations

The contrastive-based clustering models usually rely on a large number of negative pairs to capture uniform representations, which requires a large batch size and high computational complexity. In contrast, some self-supervised methods perform non-contrastive learning to capture discriminative representations with only positive pairs but suffer from the collapse of clustering. To solve these issues, a novel end-to-end Self-supervised Clustering model based on BYOL network structure with Instance-level and Cluster-level discriminations (BSC-IC) is designed in this section. Figure 1 illustrates the framework of the BSC-IC model, which consists of three joint learning components: the self-supervised learning network, the instance-level discriminative network, and the cluster-level discriminative network. The self-supervised learning network adopts a similar structure to BYOL to capture the good cluster assignments of the data with only positive pairs, which includes an online-target network and a target network. A little different from BYOL, the Softmax layer is equipped to convert the representation to the cluster assignment. The novel instance-level discriminative network and cluster-level discriminative network are designed to provide clustering-oriented guidance for self-supervised learning.

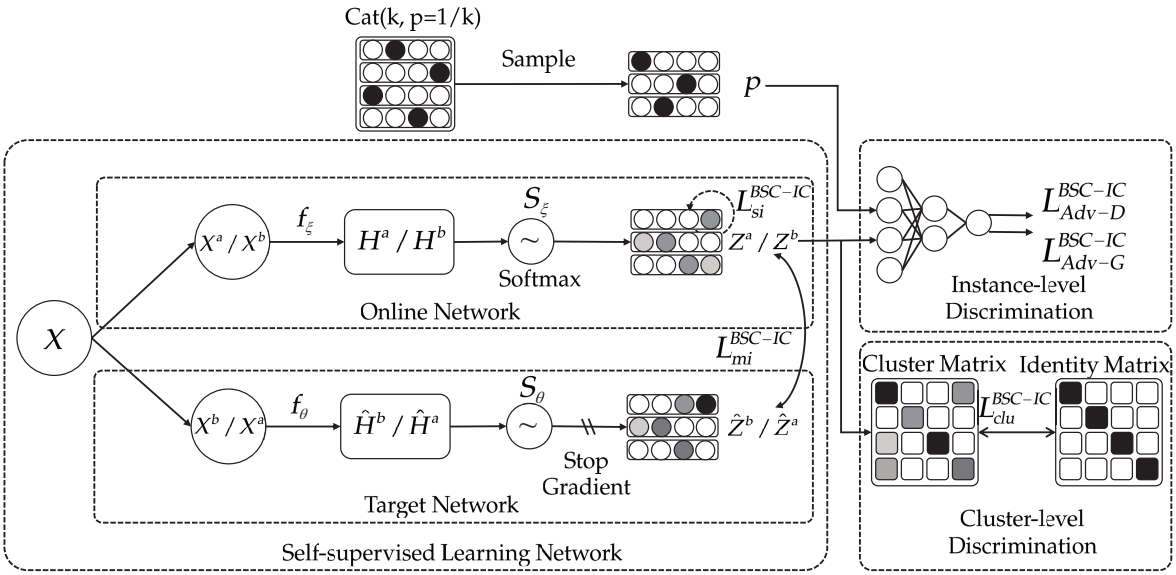

Figure 1. The framework of the proposed BSC-IC model.

3.1. The Self-Supervised Learning Network for Representation Capturing

The self-supervised learning network in BSC-IC is designed for representation learning, and contains an online network and a target network. The online network with parameters ξ is defined by an encoder, f_ξ, to extract the representation features, which is followed by a Softmax layer, S_ξ, to convert the representation to the cluster assignment of the input data. The target network has the same architecture as the online network but adopts a different set of parameters, θ.

In detail, given a set of data $X = \{x_i | 1 \leq i \leq N\} \in \mathbb{R}^{N \times D}$ in a mini-batch, N is the batch size and D is the dimension of the data. Data augmentations are first conducted to obtain two augmented views of the original data $X^a = \{x_i^a | 1 \leq i \leq N\}$ and $X^b = \{x_i^b | 1 \leq i \leq N\}$ as positive pairs. The first augmented view X^a then feeds into the online network to outputs the cluster assignment $Z^a = \{z_i^a | 1 \leq i \leq N\} \in \mathbb{R}^{N \times K}$. Simultaneously, the second augmented view X^b is fed into the target network to generate the cluster assignment $\hat{Z}^b = \{\hat{z}_i^b | 1 \leq i \leq N\} \in \mathbb{R}^{N \times K}$, where K is the number of clusters.

Self-supervised learning is then performed to maximize the similarity of positive pairs and realize the mutual optimization between the target and online networks. Unlike the cosine distance metric used in BYOL, the similarity of cluster assignments for positive pairs is measured using Kullback–Leibler (KL) divergence. The KL divergence is more suitable for capturing the difference between probability distributions. The loss for the mutual improvement in the self-supervised learning network is defined as (1).

$$L_{\text{mi}} = KL(Z^a, \hat{Z}^b) \tag{1}$$

In order to calculate the overall mutual-improvement loss of BSC-IC, we symmetrize the loss L_{mi} by separately inputting X^a into the target network and inputting X^b into the online network to compute $\tilde{L}_{mi} = KL(Z^b, \hat{Z}^a)$. Finally, the overall mutual-improvement loss of BSC-IC is denoted as (2).

$$L_{mi}^{BSC-IC} = L_{mi} + \tilde{L}_{mi} = KL(Z^a, \hat{Z}^b) + KL(Z^b, \hat{Z}^a) \tag{2}$$

The self-supervised learning network above is made up of two highly interdependent networks, in which the poor optimization of any network can deteriorate the whole structure. Particularly, the subsequent clustering may corrupt the quality of representation space and destroy the preservation of local structure. Moreover, to break the highly mutual interdependence across online and target networks, we define a novel loss, named the

self-improvement loss, as (3) to evaluate the similarity of the cluster assignments between positive pairs of the online network itself.

$$L_{si}^{BSC-IC} = KL(Z^a, Z^b) \tag{3}$$

where Z^a and Z^b indicate the cluster assignments obtained by the online network itself from two augmented views, respectively.

3.2. The Instance-Level Discriminative Network for Data Clustering

To alleviate the collapse of clustering, the instance-level discriminative network $D(\cdot)$ with parameters η in BSC-IC is constructed to provide clustering-oriented guidance for the self-supervised learning network.

Given the data X in a mini-batch, we input their two augmented views X^a and X^b to the online network, and obtain the corresponding cluster assignments Z^a and Z^b. Then, a one-hot-style prior distribution $P \sim Cat(K, p = 1/K\ )$ is imposed on the learned cluster assignments Z (the alternative to Z^a or Z^b), and the adversarial learning between Z and P is conducted to make Z closer to the form of one-hot, so as to enhance the discrimination of clusters and alleviate the collapse problem. Referring to the WGAN-GP method [38], the adversarial losses of the instance-level discriminative network for the generator $L_{Adv\text{-}G}^{BSC-IC}$ and the discriminator $L_{Adv\text{-}D}^{BSC-IC}$ are defined as (4) and (5), respectively.

$$L_{Adv\text{-}G}^{BSC-IC} = -\mathbb{E}_{z\sim Z}[D(z)] \tag{4}$$

$$L_{Adv\text{-}D}^{BSC-IC} = \mathbb{E}_{z\sim Z}[D(z)] - \mathbb{E}_{p\sim P}[D(p)] + \delta\mathbb{E}_{r\sim R}(\|\nabla_r D(r)\|_2 - 1)^2 \tag{5}$$
$$\text{where } r = \epsilon p + (1-\epsilon)z \text{ and } \epsilon \sim U[0,1]$$

where $r = \epsilon p + (1-\epsilon)z$ subject to $\epsilon \sim U[0,1]$ is a representation sampled uniformly along straight lines between the prior distribution P and the soft assignments Z, $(\|\nabla_r D(r)\|_2 - 1)^2$ is the one-centered gradient penalty that limits the gradient of the instance-level discriminative network to be around 1, and δ is the gradient penalty coefficient.

Here, the adversarial loss for the generator $L_{Adv\text{-}G}^{BSC-IC}$ is designed to minimize the Wasserstein distance between the generated assignments and the one-hot distribution, which encourages the generator network to generate more sharp cluster assignments. In contrast, the adversarial loss for the discriminator $L_{Adv\text{-}D}^{BSC-IC}$ is formulated to maximize the Wasserstein distance between the generated assignments and the one-hot distribution. Both adversarial losses train the model through the competitive process between the generator and the discriminator.

3.3. Cluster-Level Discriminations Network for Clusters Enhancement

To further benefit from the strength of capturing clustering-oriented information, a new cluster-level discrimination is integrated into the discriminative network to promote clustering performance by measuring the self-correlation between the learned cluster assignments.

Specifically, given a set of data $X = \{x_i | 1 \le i \le N\}$ in a mini-batch, the online network takes in two augmented views as input, denoted as X^a and X^b. Subsequently, the cluster assignments $Z^a \in \mathbb{R}^{N\times K}$ and $Z^b \in \mathbb{R}^{N\times K}$ are obtained, where N is the batch size and K is the number of clusters. Each column of the cluster assignments can be regarded as the representation of one cluster. Let y_i^a and y_i^b be the i-th column of Z^a and Z^b for $1 \le i \le K$, and we combine y_i^a with y_i^b to form the same cluster pair (y_i^a, y_i^b) and leave the other $K-1$ pairs as (y_i^a, y_j^b) for $\forall j \ne i$ to be different cluster pairs. A cluster-level similarity matrix

$C^{clu} = [c_{ij}^{clu}]$ with size K is defined in the column space of the cluster assignments, where c_{ij}^{clu} is measured by the cosine distance as (6)

$$c_{ij}^{clu} = \frac{(y_i^a)^T (y_j^b)}{\|y_i^a\|_2 \|y_j^b\|_2} \tag{6}$$

Then, the cluster-level discriminative loss L_{clu}^{BSC-IC} is defined as (7).

$$L_{clu}^{BSC-IC} = \sum_i (1 - c_{ii}^{clu})^2 + \lambda_{clu} \sum_i \sum_{j \neq i} (c_{ij}^{clu})^2 \tag{7}$$

where the diagonal elements as c_{ii}^{clu} are restricted to 1 to maximize the similarity between the same clusters, the non-diagonal elements as c_{ij}^{clu} for $\forall i \neq j$ are restricted to 0 to minimize the similarity between different clusters, and λ_{clu} is a positive constant to trade off two terms.

3.4. Training of the BSC-IC

Integrating the self-supervised learning network and the instance-level discriminative network, the final loss function of BSC-IC is defined as (8).

$$L^{BSC-IC} = L_{Adv-G}^{BSC-IC} + \alpha_{clu} \cdot L_{clu}^{BSC-IC} + \alpha_{si} \cdot L_{si}^{BSC-IC} + \alpha_{mi} \cdot L_{mi}^{BSC-IC} \tag{8}$$

The parameters α_{clu}, α_{si}, and α_{mi} are used to balance the significance of different loss terms. We use the adaptive moment estimation (Adam) to optimize the parameters of both the self-supervised learning network and the instance-level discriminative network. Notably, the self-supervised learning network is optimized specifically for minimizing L^{BSC-IC} in respect of the online network only while keeping the target network unchanged. This is indicated by the stop-gradient operation in Figure 1. Consequently, Equation (9) is only used to update the parameters of the online network ξ.

$$\xi = \xi - \alpha \frac{\partial L^{BSC-IC}}{\partial \xi} \tag{9}$$

where α is the learning rate. Drawing inspiration from BYOL, the target network's parameters θ are updated using a weighted moving average of the online parameters ξ. This update process can be performed using Equation (10).

$$\theta \leftarrow \tau\theta + (1 - \tau)\xi \tag{10}$$

where $\tau \in [0, 1]$ represent the target decay rate that controls the moving rate of parameters updating.

Similar to the online network, Equation (11) is employed to update the parameters of the instance-level discriminative network η. And the overall algorithm of BSC-IC is presented in Algorithm 1.

$$\eta = \eta - \alpha \frac{\partial L_{Adv\text{-}D}^{BSC-IC}}{\partial \eta} \tag{11}$$

Algorithm 1 BSC-IC

Input: Input data X, the batch size N, the number of clusters K, the maximum iterations $MaxIter$, the hyperparameters α_{mi}, α_{si} and α_{clu}.
  for epoch $\in \{0, 1, ..., MaxIter\}$ **do**
    for each batch **do**
      Calculate the mutual-improvement loss L_{mi}^{BSC-IC} by (2), the self-improvement loss L_{si}^{BSC-IC} by (3), and the instance-level discriminative losses $L_{Adv\text{-}G}^{BSC-IC}$ by (4) and $L_{Adv\text{-}D}^{BSC-IC}$ by (5), the cluster-level discriminative loss L_{clu}^{BSC-IC} by eq (7);
      Calculate the self-supervised learning network loss L^{BSC-IC} by (8);
      Update the parameter of online network ξ by (9);
      Update the parameter of target network θ by (10);
      Update the parameter of discriminative network η by (11);
    end for
  end for
Output: The online network as clustering network.

4. Experiments

In this section, we perform experiments on six well-known real-world datasets to verify the efficiency of the presented model. All the datasets, methods of comparison, evaluation metrics, implementation details, and experimental results are elaborated.

4.1. Datasets, Methods in Comparison, and Evaluation Metrics

For our evaluation, we assess the effectiveness of the proposed method using six image datasets that are divided into two distinct categories. The first category consists of low-detailed grayscale images like Fashion-MNIST and MNIST. Meanwhile, the second category includes high-detailed color images, such as ImageNet-10, CIFAR-10, CIFAR-100, and Tiny-ImageNet. Table 1 provides a concise description of these datasets.

Table 1. Brief description of datasets used in our experiments.

Datasets	Samples Size	Classes	Image Size
MNIST	70,000	10	$28 \times 28 \times 1$
Fashion-MNIST	70,000	10	$28 \times 28 \times 1$
CIFAR-10	60,000	10	$32 \times 32 \times 3$
CIFAR-100	60,000	20	$32 \times 32 \times 3$
ImageNet-10	13,000	10	$96 \times 96 \times 3$
Tiny-ImageNet	100,000	100	$64 \times 64 \times 3$

Twenty-two mainstream clustering methods as the baseline are adopted for the comparative analysis, including traditional distance-based clustering methods, like K-means [39], SC [40], AC [41], and NMF [42]; deep generative clustering methods, such as AE [43], DEC [44], JULE [45], DEPICT [46], DAC [47], VAE [48], and GAN [37]; and contrastive-based clustering models, such as IIC [49], BYOL [34], DCCM [50], DCCS [51], DHOG [52], GATCluster [53], DRC [54], PICA [55], CC [31], GCC [32], and C3 [33]. It is important to mention that clustering results for the NMF, SC, AE, GAN, VAE, and BYOL methods are obtained by applying k-means on the extracted image features.

Three metrics, i.e., the clustering accuracy (ACC), the normalized mutual information (NMI), and the adjusted rand index (ARI), are utilized to evaluate the clustering performance of different algorithms. For all metrics, a higher value is better. All clustering algorithms are conducted on a computer with two Nvidia TITAN RTX 24G GPUs.

4.2. Implementation Details

Similar image augmentations as DCCS [51] and CC [31] are conducted first to obtain the augmented samples. For low-detailed grayscale image datasets, cropping and horizon-

tal flipping are employed as the augmentation strategies. For high-detailed color image datasets, color distortion and grayscale conversion are incorporated. Specifically, the color distortion method alters various attributes of the image, including contrast, saturation, brightness, and hue, while the grayscale conversion step transforms the color image into a grayscale format.

ResNet-18 is employed to extract the representation for the self-supervised learning network of BSC-IC. A Softmax layer is used to convert the representation into the cluster assignment of data with a dimension of cluster number K. A three-layer fully connected network is utilized as the instance-level discriminative network to divide the data samples into different clusters, and the dimensions of various layers are set to K-1024-512-1.

The Adam optimizer with a learning rate of 0.0003 is adopted to simultaneously optimize the self-supervised learning network and the discriminative network. The moving average parameter τ in the self-supervised learning network is set to 0.99, the discriminative network's gradient penalty coefficient δ is set to 10, and the default batch size N is set to 64. The BSC-IC model involves three control parameters, which are utilized to trade off the effects of different terms in the total loss function. The recommended values of various parameters on different datasets are listed in Table 2.

Table 2. The recommended values of the parameters on different datasets.

Parameter	MNIST, Fashion-MNIST	CIFAR-10, CIFAR-100, ImageNet-10, Tiny-ImageNet
α_{si}	2	4
α_{mi}	1	2
α_{clu}	1	1

Table 3 lists the number of hyperparameters of different models. It can be seen that the proposed BSC-IC model has fewer hyperparameters compared with other models, which indicates a simpler model architecture and ease for parameter tuning in BSC-IC.

Table 3. The number of hyperparameters on different methods.

Methods	Number of Hyperparameters
BSC-IC	**3**
DCCS	4
DCCM	4
GCC	3
GatCluster	4

4.3. Experimental Results

The clustering results of the testing algorithms on six datasets in terms of ACC, NMI, and ARI are listed in Table 4, Table 5, and Table 6, separately, and reveal some interesting observations. The best results are shown in bold.

Table 4. Clustering results of tested algorithms in term of ACC on six datasets.

Method	MNIST	Fashion-MNIST	CIFAR-10	ImageNet-10	CIFAR-100	Tiny-ImageNet
K-means [39]	0.572	0.474	0.229	0.241	0.130	0.025
SC [40]	0.696	0.508	0.247	0.274	0.136	0.022
AC [41]	0.695	0.500	0.228	0.242	0.138	0.027
NMF [42]	0.545	0.434	0.190	0.230	0.118	0.029
AE [43]	0.812	0.563	0.314	0.317	0.165	0.041
DEC [44]	0.843	0.590	0.301	0.381	0.185	0.037
JULE [45]	0.964	0.563	0.272	0.300	0.137	0.033
VAE [48]	0.945	0.578	0.291	0.381	0.152	0.036

Table 4. *Cont.*

Method	MNIST	Fashion-MNIST	CIFAR-10	ImageNet-10	CIFAR-100	Tiny-ImageNet
DEPICT [46]	0.965	0.392	0.279	0.363	0.137	-
GAN [37]	0.736	0.558	0.315	0.346	0.151	0.039
DAC [47]	0.978	0.615	0.522	0.527	0.238	0.066
IIC [49]	0.992	0.657	0.617	0.701	0.257	-
BYOL [34]	0.985	0.703	0.658	0.834	0.334	0.053
DCCS [51]	0.989	0.756	0.656	0.737	0.315	0.106
DCCM [50]	0.982	0.753	0.623	0.710	0.327	0.108
DHOG [52]	0.954	0.658	0.666	-	0.261	-
GATCluster [53]	0.943	0.618	0.610	0.739	0.281	-
DRC [54]	0.961	0.694	0.727	0.884	0.367	-
PICA [55]	0.951	0.683	0.696	0.870	0.337	0.098
CC [31]	0.966	0.708	0.790	0.893	0.429	0.140
GCC [32]	0.987	0.768	**0.856**	0.901	**0.472**	0.138
C3 [33]	0.993	0.773	0.836	**0.943**	0.456	0.140
BSC-IC (ours)	**0.996**	**0.782**	0.753	0.901	0.403	**0.157**

Table 5. Clustering results of tested algorithms in term of NMI on six datasets.

Method	MNIST	Fashion-MNIST	CIFAR-10	ImageNet-10	CIFAR-100	Tiny-ImageNet
K-means [39]	0.500	0.512	0.087	0.119	0.084	0.065
SC [40]	0.663	0.575	0.103	0.151	0.090	0.063
AC [41]	0.609	0.564	0.105	0.138	0.098	0.069
NMF [42]	0.608	0.425	0.081	0.132	0.079	0.072
AE [43]	0.725	0.561	0.239	0.210	0.100	0.131
DEC [44]	0.772	0.601	0.257	0.282	0.136	0.115
JULE [45]	0.913	0.608	0.192	0.175	0.103	0.102
VAE [48]	0.876	0.630	0.245	0.282	0.108	0.113
DEPICT [46]	0.917	0.392	0.237	0.242	0.094	-
GAN [37]	0.763	0.584	0.265	0.225	0.120	0.127
DAC [47]	0.935	0.632	0.396	0.394	0.185	0.190
IIC [49]	0.979	0.634	0.513	0.598	0.198	-
BYOL [34]	0.968	0.653	0.548	0.734	0.305	0.103
DCCS [51]	0.970	0.704	0.569	0.640	0.278	0.219
DCCM [50]	0.951	0.684	0.496	0.608	0.285	0.224
DHOG [52]	0.921	0.632	0.585	-	0.258	-
GATCluster [53]	0.896	0.614	0.475	0.594	0.215	-
DRC [54]	0.923	0.667	0.621	0.830	0.356	-
PICA [55]	0.891	0.653	0.591	0.802	0.310	0.277
CC [31]	0.932	0.675	0.705	0.859	0.431	0.340
GCC [32]	0.975	0.709	**0.764**	0.842	**0.472**	0.347
C3 [33]	0.978	0.715	0.743	**0.905**	0.435	0.335
BSC-IC (ours)	**0.982**	**0.723**	0.681	0.861	0.397	**0.352**

Table 6. Clustering results of tested algorithms in term of ARI on six datasets.

Method	MNIST	Fashion-MNIST	CIFAR-10	ImageNet-10	CIFAR-100	Tiny-ImageNet
K-means [39]	0.365	0.348	0.049	0.057	0.028	0.005
SC [40]	0.521	0.382	0.085	0.076	0.022	0.004
AC [41]	0.481	0.371	0.065	0.067	0.034	0.005
NMF [42]	0.430	0.321	0.034	0.065	0.026	0.005
AE [43]	0.613	0.379	0.169	0.152	0.048	0.007
DEC [44]	0.741	0.446	0.161	0.203	0.050	0.007
JULE [45]	0.927	0.439	0.138	0.138	0.033	0.006
VAE [48]	0.884	0.542	0.167	0.203	0.040	0.006
DEPICT [46]	0.094	0.357	0.171	0.197	0.041	-

Table 6. *Cont.*

Method	MNIST	Fashion-MNIST	CIFAR-10	ImageNet-10	CIFAR-100	Tiny-ImageNet
GAN [37]	0.827	0.631	0.176	0.157	0.045	0.007
DAC [47]	0.949	0.502	0.306	0.302	0.088	0.017
IIC [49]	0.978	0.524	0.411	0.549	0.096	-
BYOL [34]	0.965	0.585	0.468	0.554	0.147	0.028
DCCS [51]	0.976	0.623	0.469	0.560	0.168	0.032
DCCM [50]	0.954	0.602	0.408	0.555	0.173	0.038
DHOG [52]	0.917	0.534	0.492	-	0.118	-
GATCluster [53]	0.887	0.522	0.402	0.552	0.116	-
DRC [54]	0.924	0.551	0.547	0.798	0.208	-
PICA [55]	0.854	0.545	0.512	0.761	0.171	0.040
CC [31]	0.931	0.565	0.637	0.822	0.266	0.071
GCC [32]	0.967	0.625	**0.728**	0.822	**0.305**	0.075
C3 [33]	0.973	0.629	0.703	**0.860**	0.274	0.064
BSC-IC (ours)	**0.979**	**0.639**	0.592	0.829	0.232	**0.085**

First and foremost, compared with the traditional distance-based clustering methods, like K-means, AC, NMF, and SC, all the deep clustering methods show obvious advantages. This emphasizes that deep clustering has the ability to enhance clustering performance by the capturing semantic information of samples through deep neural networks.

Secondly, BSC-IC significantly outperforms most deep clustering methods on all six datasets. This demonstrates the efficiency of self-supervised representation learning only with positive pairs in our model, which helps to extract the similarities and dissimilarities between different views of samples and capture important clustering-orientated information. It is worth noting that GCC achieves the best performance on the CIFAR-10 and CIFAR-100 datasets. But this relies on a large number of negative pairs to capture the uniform representations, which requires a large batch size, like 256, and high computational complexity. In our model, a smaller batch size, like 64, and only positive pairs can also achieve good clustering performance. Figure 2 shows the ACC curves obtained by CC, GCC, and our model with different batch sizes on the CIFAR-10 and CIFAR-100 datasets. It can be seen that the ACCs of CC and GCC drop sharply with the decrease in batch size. Specifically, when the batch size changes from 256 to 64, the ACC of GCC drops by approximately 18 percentage points on the CIFAR-10 dataset and 9 percentage points on the CIFAR-100 dataset. Similarly, the ACC of CC drops by about 20 percentage points on the CIFAR-10 dataset and 6 percentage points on the CIFAR-100 dataset. In contrast, our model yields a more stable ACC without the influence of the value of the batch size.

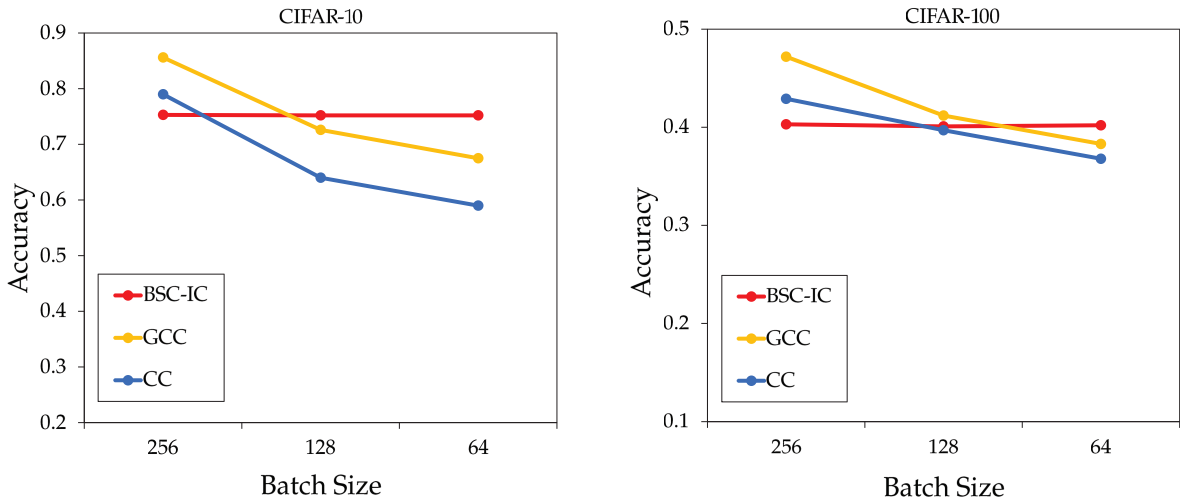

Figure 2. The impact of batch size on accuracy in CIFAR-10 and CIFAR-100 datasets.

5. Ablation Study and Analysis

The ablation study and analysis are carried out in this section to further understand the effect of each term in the loss function, including the self-improvement term (denoted as SI), the mutual-improvement term (denoted as MI), the instance-level discriminative term (denoted as IL), and the cluster-level discriminative term (denoted as CL). The ablation study of BSC-IC on the MNIST and ImageNet-10 datasets is presented in Table 7. The check mark and the cross mark respectively represent the inclusion and exclusion of each terms.

Table 7. The results of the ablation study.

	SI Term	MI Term	IL Term	CL Term	ACC on MNIST	ACC on ImageNet-10
Baseline	✓	✓	✓	✓	0.996	0.901
①	✓	✓	×	✓	0.943	0.884
②	✓	✓	✓	×	0.995	0.853
③	✓	✓	×	×	0.112	0.104
④	×	✓	✓	✓	0.979	0.751
⑤	✓	×	✓	✓	0.965	0.745
⑥	×	×	✓	✓	0.105	0.103

In the discriminative network, the instance-level discriminative term focuses on optimizing the assignment of instances within clusters, while the cluster-level discriminative term aims to optimize the relationships between clusters. Together, they provide effective clustering guidance for self-supervised learning. From ① and ② in Table 7, it can be seen that the absence of any of them will lead to a suboptimal solution for cluster assignments. The most fatal is that the absence of both of them will lead to a collapse of the clustering as ③ in 7.

In the self-supervised learning network, the self-improvement term aims to ensure the stability of the network structure, while the mutual-improvement term provides the alignment between positive pairs for the capture of uniform representations. Together, they provide effective optimization over the online and target networks for the capture of discriminative representations. From ④ and ⑤ in Table 7, it can be seen that the absence of any of them will lead to a decrease in cluster accuracy. Moreover, the absence of both terms as ⑥ will disrupt the optimization over the online and target networks and prevent our method from performing clustering.

6. Conclusions

This paper develops a novel end-to-end self-supervised clustering model based on the BYOL network structure method to jointly seek high-quality representation and perform clustering. The basic self-supervised learning network is first modified, followed by the incorporation of a Softmax layer to capture the cluster assignments as data representation. The mutual-improvement loss and the self-improvement loss together provide effective optimization over online and target networks in BYOL for the capture of discriminative representations. Then, adversarial learning and self-correlation measuring are performed on the learned cluster assignments to promote clustering. The instance-level discriminative loss and the cluster-level discriminative loss together provide effective clustering guidance for self-supervised learning. Experimental results on real-world datasets show the efficiency of the proposed model.

Author Contributions: All the authors contributed extensively to the manuscript. X.C.: Contributed to algorithm development, programming, paper writing and revision, investigation, and methodology. J.Z.: Contributed to project administration, resources, supervision, and paper revisions and suggestions. Y.C.: Helped with formatting, resources, supervision, and review and editing of the paper. S.H.: Contributed to resources, supervision, and paper writing and revision. Y.W.: Contributed to resources and paper revisions and suggestions. T.D.: Contributed to resources and supervision. C.Y.: Contributed to resources, supervision, and helped with formatting. B.L.: Contributed to resources,

supervision, and helped with grammar correction. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under grants No. 62273164 and No. 62373164, the Key Research Project of Quancheng Laboratory, China, under grant No. QCLZD202303, and the Research Project of Provincial Laboratory of Shandong, China, under grant No. SYS202201.

Data Availability Statement: The data presented in this study are openly available in: https://github.com/FrostaQ31/BSC.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Krishnapuram, R.; Joshi, A.; Nasraoui, O.; Yi, L. Low-complexity fuzzy relational clustering algorithms for web mining. *IEEE Trans. Fuzzy Syst.* **2001**, *9*, 595–607. [Crossref] [CrossRef]
2. Berkhin, P. A survey of clustering data mining techniques. In *Grouping Multidimensional Data: Recent Advances in Clustering*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 25–71. [Crossref]
3. Gulati, H.; Singh, P.K. Clustering techniques in data mining: A comparison. In Proceedings of the 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 11–13 March 2015; pp. 410–415.
4. Norberg, P.; Baugh, C.M.; Gaztanaga, E.; Croton, D.J. Statistical analysis of galaxy surveys—I. Robust error estimation for two-point clustering statistics. *Mon. Not. R. Astron. Soc.* **2009**, *396*, 19–38. [Crossref] [CrossRef]
5. Dransfield, E.; Morrot, G.; Martin, J.F.; Ngapo, T. The application of a text clustering statistical analysis to aid the interpretation of focus group interviews. *Food Qual. Prefer.* **2004**, *15*, 477–488. [Crossref] [CrossRef]
6. Srivastava, A.; Joshi, S.H.; Mio, W.; Liu, X. Statistical shape analysis: Clustering, learning, and testing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 590–602. [Crossref] [CrossRef]
7. Baraldi, A.; Blonda, P. A survey of fuzzy clustering algorithms for pattern recognition. I. *IEEE Trans. Syst. Man Cybern. Part B* **1999**, *29*, 778–785. [Crossref] [CrossRef]
8. Diday, E.; Govaert, G.; Lechevallier, Y.; Sidi, J. Clustering in pattern recognition. In Proceedings of the Digital Image Processing: Proceedings of the NATO Advanced Study Institute, Bonas, France, 23 June–4 July 1980; Springer: Berlin/Heidelberg, Germany, 1981; pp. 19–58.
9. Namratha, M.; Prajwala, T. A comprehensive overview of clustering algorithms in pattern recognition. *IOSR J. Comput. Eng.* **2012**, *4*, 23–30.
10. Bicego, M.; Murino, V.; Figueiredo, M.A. Similarity-based clustering of sequences using hidden Markov models. In Proceedings of the International Workshop on Machine Learning and Data Mining in Pattern Recognition, Leipzig, Germany, 5–7 July 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 86–95.
11. Guyon, I.; Elisseeff, A. An introduction to feature extraction. In *Feature Extraction: Foundations and Applications*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1–25. [Crossref]
12. Salahat, E.; Qasaimeh, M. Recent advances in features extraction and description algorithms: A comprehensive survey. In Proceedings of the 2017 IEEE International Conference on Industrial Technology (ICIT), Toronto, ON, Canada, 22–25 March 2017; pp. 1059–1063. [Crossref]
13. Cohen, M.B.; Elder, S.; Musco, C.; Musco, C.; Persu, M. Dimensionality reduction for k-means clustering and low rank approximation. In Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, Portland, OR, USA, 14–17 June 2015; pp. 163–172. [Crossref]
14. Tian, D.P. A review on image feature extraction and representation techniques. *Int. J. Multimed. Ubiquitous Eng.* **2013**, *8*, 385–396.
15. Bro, R.; Smilde, A.K. Principal component analysis. *Anal. Methods* **2014**, *6*, 2812–2831. [Crossref] [CrossRef]
16. Hofmann, T.; Schölkopf, B.; Smola, A.J. Kernel methods in machine learning. *Ann. Statist.* **2008**, *36*, 1171–1220. [Crossref] [CrossRef]
17. Saul, L.K.; Weinberger, K.Q.; Sha, F.; Ham, J.; Lee, D.D. Spectral methods for dimensionality reduction. *Semi-Supervised Learn.* **2006**, *3*.
18. Wang, Y.; Dong, J.; Zhou, J.; Xu, G.; Chen, Y. Random feature map-based multiple kernel fuzzy clustering with all feature weights. *Int. J. Fuzzy Syst.* **2019**, *21*, 2132–2146. [Crossref] [CrossRef]
19. Fern, X.Z.; Brodley, C.E. Random projection for high dimensional data clustering: A cluster ensemble approach. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), Washington, DC, USA, 21–24 August 2003; pp. 186–193.
20. Aljalbout, E.; Golkov, V.; Siddiqui, Y.; Strobel, M.; Cremers, D. Clustering with deep learning: Taxonomy and new methods. *arXiv* **2018**, arXiv:1801.07648. [Crossref]
21. Ren, Y.; Pu, J.; Yang, Z.; Xu, J.; Li, G.; Pu, X.; Yu, P.S.; He, L. Deep clustering: A comprehensive survey. *arXiv* **2022**, arXiv:2210.04142. [Crossref]
22. Zhong, S.; Ghosh, J. Generative model-based document clustering: A comparative study. *Knowl. Inf. Syst.* **2005**, *8*, 374–384. [Crossref] [CrossRef]

23. Tu, Z. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, Beijing, China, 17–21 October 2005; Volume 2, pp. 1589–1596. [Crossref]
24. Yang, Z.; Xu, B.; Luo, W.; Chen, F. Autoencoder-based representation learning and its application in intelligent fault diagnosis: A review. *Measurement* **2022**, *189*, 110460. [Crossref] [CrossRef]
25. Yang, L.; Cheung, N.M.; Li, J.; Fang, J. Deep clustering by gaussian mixture variational autoencoders with graph embedding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6440–6449.
26. Mukherjee, S.; Asnani, H.; Lin, E.; Kannan, S. Clustergan: Latent space clustering in generative adversarial networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4610–4617. [Crossref]
27. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 12–18 July 2020; pp. 1597–1607.
28. Van Gansbeke, W.; Vandenhende, S.; Georgoulis, S.; Proesmans, M.; Van Gool, L. Scan: Learning to classify images without labels. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 268–285. [Crossref]
29. Niu, C.; Shan, H.; Wang, G. Spice: Semantic pseudo-labeling for image clustering. *IEEE Trans. Image Process.* **2022**, *31*, 7264–7278. [Crossref] [CrossRef]
30. Park, S.; Han, S.; Kim, S.; Kim, D.; Park, S.; Hong, S.; Cha, M. Improving unsupervised image clustering with robust learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12278–12287.
31. Li, Y.; Hu, P.; Liu, Z.; Peng, D.; Zhou, J.T.; Peng, X. Contrastive clustering. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; Volume 35, pp. 8547–8555. [Crossref]
32. Zhong, H.; Wu, J.; Chen, C.; Huang, J.; Deng, M.; Nie, L.; Lin, Z.; Hua, X.S. Graph contrastive clustering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 9224–9233.
33. Sadeghi, M.; Hojjati, H.; Armanfard, N. C3: Cross-instance guided contrastive clustering. *arXiv* **2022**, arXiv:2211.07136. [Crossref]
34. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. Bootstrap your own latent-a new approach to self-supervised learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21271–21284.
35. Oord, A.v.d.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
36. Pedrycz, W.; Gomide, F. *An Introduction to Fuzzy Sets: Analysis and Design*; MIT Press: Cambridge, MA, USA, 1998.
37. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [Crossref] [CrossRef]
38. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved Training of Wasserstein GANs. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
39. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, 1967; Volume 1, pp. 281–297.
40. Ng, A.; Jordan, M.; Weiss, Y. On spectral clustering: Analysis and an algorithm. *Adv. Neural Inf. Process. Syst.* **2001**, *14*.
41. Gowda, K.C.; Krishna, G. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognit.* **1978**, *10*, 105–112. [Crossref] [CrossRef]
42. Cai, D.; He, X.; Wang, X.; Bao, H.; Han, J. Locality preserving nonnegative matrix factorization. In Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence, Pasadena, CA, USA, 11–17 July 2009.
43. Bengio, Y.; Lamblin, P.; Popovici, D.; Larochelle, H. Greedy layer-wise training of deep networks. *Adv. Neural Inf. Process. Syst.* **2006**, *19*.
44. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised deep embedding for clustering analysis. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 478–487.
45. Yang, J.; Parikh, D.; Batra, D. Joint unsupervised learning of deep representations and image clusters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5147–5156.
46. Ghasedi Dizaji, K.; Herandi, A.; Deng, C.; Cai, W.; Huang, H. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5736–5745.
47. Chang, J.; Wang, L.; Meng, G.; Xiang, S.; Pan, C. Deep adaptive image clustering. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5879–5887.
48. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114. [Crossref]
49. Ji, X.; Henriques, J.F.; Vedaldi, A. Invariant information clustering for unsupervised image classification and segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9865–9874.
50. Wu, J.; Long, K.; Wang, F.; Qian, C.; Li, C.; Lin, Z.; Zha, H. Deep comprehensive correlation mining for image clustering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8150–8159.

51. Zhao, J.; Lu, D.; Ma, K.; Zhang, Y.; Zheng, Y. Deep image clustering with category-style representation. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XIV 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 54–70. [Crossref]
52. Darlow, L.N.; Storkey, A. Dhog: Deep hierarchical object grouping. *arXiv* **2020**, arXiv:2003.08821. [Crossref]
53. Niu, C.; Zhang, J.; Wang, G.; Liang, J. Gatcluster: Self-supervised gaussian-attention network for image clustering. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXV 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 735–751. [Crossref]
54. Zhong, H.; Chen, C.; Jin, Z.; Hua, X.S. Deep robust clustering by contrastive learning. *arXiv* **2020**, arXiv:2008.03030. [Crossref]
55. Huang, J.; Gong, S.; Zhu, X. Deep semantic clustering by partition confidence maximisation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8849–8858.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.