IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 14, 2020, accepted September 24, 2020, date of publication September 28, 2020,
date of current version October 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027481

AAE-SC: A scRNA-Seq Clustering Framework Based on Adversarial Autoencoder

YULUN WU, YANMING GUO, YANDONG XIAO, AND SONGYANG LAO
College of Systems Engineering, National University of Defense Technology, Changsha 410072, China

Corresponding author: Yanming Guo (guoyanming@nudt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61806218, and in part by the National Natural Science of Hunan Province, China, under Grant 2019JJ50722.

ABSTRACT Single-cell RNA sequencing (scRNA-seq) provides the expression profiles of individual cells, and it is expected to provide higher cellular differential resolution than traditional bulk RNA sequencing. In scRNA-seq analysis, clustering is crucial for identifying cell types, and can be potentially exploited to understand high-level biological processes. Recently, autoencoder has been successfully applied in scRNA-seq clustering problem and achieved promising results. Most existing works focus on characterizing the sparsity of data, and directly utilize the bottleneck feature of the autoencoder for clustering might not be optimal. In this paper, a novel framework named Adversarial AutoEncoder ScRNA-seq Clustering (AAE-SC) is proposed to bring an additional constraint on the bottleneck feature. Specifically, AAE-SC adds an AAE module on top of the bottleneck layer, and constrains the bottleneck feature distribution to be aligned with a consistent distribution. Also, the AAE and the reconstructed modules are jointly optimized to generate a highly discriminative and consistent feature, which is further proceeded for clustering. We find that by using AAE-SC to impose certain constraints on the features of the hidden layer, the performance of clustering can be improved. Experimental results on three real-world datasets demonstrate that the proposed AAE-SC framework outperformed state-of-the-art methods by 2% at least and 5% at most. And AAE-SC shows more robustness than the baseline model for downsampled and unbalanced cluster size datasets.

INDEX TERMS Single-cell RNA-seq data, adversarial autoencoder, clustering analysis, unsupervised learning.

I. INTRODUCTION

Technological advances in single-cell RNA sequencing (scRNA-seq) [1]–[5] have revolutionized transcriptomic studies by providing higher resolution of individual cellular differences of transcriptomes than commonly used bulk RNA sequencing. They allow researchers to systematically study the cellular heterogeneity, cellular developmental trajectories and classification of tumor sub-population across a large number of cells [6]. Unsupervised clustering is an essential step in the analysis of scRNA-seq to achieve the above tasks. Only after clustering, the cell types can be identified, and researchers can further depict the cellular functional states and infer the potential cellular dynamics [7].

Although clustering is a traditional machine learning research field [8]–[10] and there have been some representative methods such as *k*-means [11] and spectral clustering [12], clustering analysis on scRNA-seq data is still a challenge due to the *dropout* occurring in the raw data [13]. The *dropout* refers to the fact that there are some false-zero counts and gene count matrix may contain actually no reported data, which are caused by low sequencing depth and other technology limits. As shown in Fig. 1, different heat map colors indicate different gene expression levels (the value in the gene count matrix). It is obvious that most genes in cells have very low expression level and only a few genes express over 0. Therefore the *dropout* makes the scRNA-seq data highly sparse, and traditional clustering approaches fail to deal with this data. To alleviate this problem, several specific clustering algorithms including SNN-Clip [14], single-cell interpretation via multikernel learning (SIMLR) [15] and MPSSC [16] for scRNA-seq data have been proposed. However, their computational cost is huge for large-scale datasets, and the clustering performance is still inferior.

Recently, deep learning technology [17] has made significant breakthroughs in computer vision, natural language processing and other cross domains based on deep neural

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

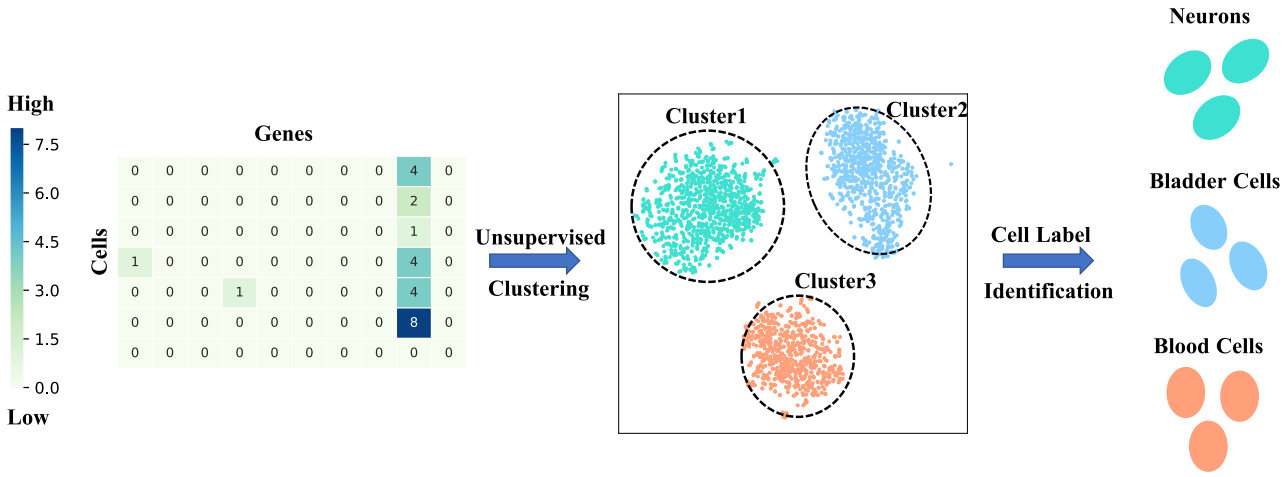


FIGURE 1. Explanation of scRNA-seq clustering task. Because *dropout* causes the gene expression level of the original data to be very low, the data is highly sparse, and it brings difficulties to the subsequent clustering. Therefore, special clustering algorithms are required to process this type of data and correctly assign different cell samples to different And identify the cell type. The heat map on the left of the figure is a visual representation of raw scRNA-seq data, and the numbers in the heat map indicate the expression value of each gene in the cell sample. The color bars in the figure indicate the level of gene expression.

networks (DNN) [18]. DNN can process high-dimensional data and extract efficient and effective features, thus becomes the dominant option for big data analysis. Among a variety of deep learning frameworks, the autoencoder [19] is a classic unsupervised algorithm. It consists of two components: an encoder and a decoder with symmetrical structures. The encoder first projects high-dimensional data to a low-dimensional feature in the hidden layer (i.e. the bottleneck feature), then the decoder attempts to reconstruct the original data from this bottleneck feature. If the bottleneck feature can reconstruct the input data well, or is considered to contains the most informative components of the original data, it is straightforward to be utilized for clustering. Actually, there are several relevant attempts which apply the autoencoder algorithm to scRNA-seq, including scScope [20], scvis [21], deep count autoencoder (DCA) [22] and single-cell-model-based deep embedded clustering (scDeepCluster) [23]. These models improve the clustering performance of scRNA-seq and significantly speed up calculations, but most of them focus on addressing the sparsity problem, and directly utilize the bottleneck feature for clustering. Without constraints on hidden code during feature learning process, the latent features might be noisy and distorted, which are not good for clustering.

In this paper, we propose a novel framework, named Adversarial AutoEncoder ScRNA-seq Clustering (AAE-SC). Our baseline model scDeepCluster lacks constraint on the hidden feature and its performance on clustering is limited. Therefore, inspired by the generative model Adversarial Autoencoder (AAE) [24] which can match the latent feature to any prior distribution while processing the data reconstruction stage, here we add an AAE module on the basis of scDeepCluster to preserve the data structure in hidden layer during the feature learning, forming the AAE-SC framework. Specifically, AAE-SC first trains the additional discriminator network and data reconstruction module by the adversarial loss and the zero-inflated negative binomial (ZINB) loss. After acquiring the constrained initial features, AAE-SC clusters the hidden features by jointly optimizing the reconstruction loss and clustering loss from an improved deep clustering layer. Finally, experiments on several real-world datasets demonstrate that the proposed AAE-SC framework can considerably outperform the state-of-the-art models on three clustering evaluation metrics. Also, subsequent experiments also show that AAE-SC achieves better robustness than the baseline model. Our main contributions are as follows:

- We proposed AAE-SC framework which innovatively utilizes the adversarial autoencoder component to constrain the low-dimensional feature and uses the constrained hidden feature for clustering.
- The proposed AAE-SC framework is evaluated on three real-world scRNA-seq datasets and the clustering results on the three clustering metrics are at least 2% and at most 5% better than that of the state-of-the-art model.
- Furthermore, regarding experiments on datasets with downsampled and unbalanced cluster size, our model also shows better robustness compared to the baseline model. And the effect of clustering coefficient on clustering performance and the network structure selection of AAE-SC is studied in detail.

The remaining parts of this paper are organized as follows: The Section II mainly reviews the representative works of scRNA-seq clustering. The Section III introduces the proposed AAE-SC framework. The Section IV describes the dataset information, the relevant implemental details of the model and evaluation metrics of the experiment. The following section is about the analysis and discussion of the experimental results. Finally, we summarize the work of the paper

TABLE 1. Abbreviations in this article.

Abbreviation	Full name	Abbreviation	Full name
AAE	Adversarial autoencoder	VAE	Variational autoencoder
AAE-SC	Adversatial Autoencoder ScRNA-seq Clustering	DEC	Deep Embedded Clustering
scRNA-seq	Single-cell RNA sequencing	IDEC	Improved Deep Embedded Clustering
SIMLR	Single-cell interpretation via multikernel learning	MSE	Mean Square Error
DNN	Deep Neural Network	KL	Kullback-Leibler
DCA	Deep Count Autoencoder	GAN	Generative Adversarial Network
scDeepCluster	Single-cell-model based deep embedded clustering	DAE	Denoised autoencoder
ZINB	Zero-inflated negative bionomical	ACC	Clustering Accuracy
scCATCH	Single-cell Cluster-based automatic Annotation Toolkit for Cellular Heterogeneity	NMI	Normalized Mutual Information
VASC	Deep variational autoencoder for scRNA-seq data	ARI	Adjusted Rand Index

and the look forward to the future work in the Section VI. Table 1 provides the abbreviation-full name comparison table of the full article.

II. RELATED WORK

In this section, we briefly reviewed and summarized the representative works in scRNA-seq clustering analysis. And we focused on these works from two aspects: traditional clustering methods and deep learning-based methods.

A. TRADITIONAL METHODS

Early researchers applied the traditional clustering algorithms to analyze the scRNA-seq data. The SNN-Clip [14] identified groups of cells that are densely connected by graph-based clique algorithm. It leveraged the concept of shared nearest neighbor to calculate the cell similarity for finding all quasi-cliques. Then several algorithms based on k-means have been proposed. RaceID [25] utilized *k*-means to unravel the heterogeneity of rare intestinal cell types. SAIC [26] applied an iterative *k*-means to identify the optimal subset of signature genes that separate single cells into distinct clusters. Since *k*-means is a greedy algorithm, these methods may fail to find their global optimum. Besides, *k*-means is sensitive to outliers since it tends to identify globular clusters, resulting in the failures in detecting of rare cell types. To overcome the above disadvantages, the RaceID2 [27] replaced *k*-means with *k*-medoids clustering and later the improved version of RaceID3 [28] added the random forest component to ameliorate the clustering accuracy. Some researchers have also tried to add additional constraints to the feature extraction phase before the clustering phase on scRNA-seq data begin. The LAK [29] integrated Lasso penalty into clustering method as the feature selection approaches, and then using *k*-means algorithm based on the selected genes which have an actual effect on clustering.

Some researchers also tried to determine the variety of cell groups by spectral clustering method. The SIMLR [15] used Gaussian Kernel and assisted spectral clustering to learn a better distance metric to model the special data structure. In addition, SIMLR can process the large-scale datasets with heavy noise. MPSSC [16] innovatively used L1 penalty to characterize the sparsity of data with multi-kernel spectral clustering. SinNLRR [30] was proposed to impose a non-negative and a low rank structure on cell similarity matrix and then utilized spectral clustering to detect cell types.

Although these methods have improved clustering performance on scRNA-seq (better performance on cluster metrics, see the Section V for details), they were usually not very scalable and required huge computing resources and storage when dealing with the large-scale dataset (for example, researches [23] have shown that it is hard to run the datasets which contain over 4000 cell samples with even large memory such as 256GB by MPSSC and SIMLR, and the clustering time of some spectral clustering methods on 2000 sample datasets is more than 10 times longer than other algorithms [31]). Some scalable tools like Seurat [32] and SCANPY [33] which utilized Louvain algorithm to detect the community have low time complexity on large-scale datasets, but they may not find small communities and therefore reduce the accuracy of clustering. Some researchers also tried to use existing datasets as reference to identify to cell types of scRNA-seq data. The single-cell Cluster-based automatic Annotation Toolkit for Cellular Heterogeneity (scCATCH) [34] algorithm annotated cell types through the tissue-specific cellular taxonomy reference database and the evidence-based scoring protocol.

B. DEEP LEARNING METHODS

Recently, deep learning has made breakthroughs in many areas of bioinformatics [9], [35], [36]. Among all the deep learning techniques, autoencoder has been the most popular so far. There has been many autoencoder approaches which aim to deal with scRNA-seq data more efficiently and accurately. Lin *et al.* [37] tried to reduce the dimensions of scRNA-seq data by neural networks with prior biological knowledge. The scScope [20] used a stacked auto-encoder to build a recurrent model and conducted batch effect removal, *dropout* imputation and cell subpopulation identification. Inspired by the recent success of autoencoders for sparse matrix imputation in collaborative filtering for recommendation system, Talwar *et al.* [38] proposed AutoImpute, which was also based on autoencoder and aimed to handle the *dropout* in scRNA-seq data. This model utilized over-complete autoencoder to regenerate the imputed expression matrix by focusing on the non-zero entries in the input sparse matrix. Some works like deep variational autoencoder for scRNA-seq data (VASC) [39] and scvis [21] both

utilized variational autoencoder (VAE) [40] to characterize the data structure of scRNA-seq afterwards. VASC modeled the dropout events and attempted to find the nonlinear hierarchical feature representation of original data, while scvis inferred the approximate posterior distribution of low-dimensional latent variables and accordingly learned a parametric mapping from a high-dimensional space to a low-dimensional embedding.

The imputation model DCA [22] adjusted the reconstruction loss of traditional auto-encoder into a special ZINB model-based loss function, and the loss function is the likelihood of the ZINB distribution. DCA constructed a denoising autoencoder with three neuron nodes in the output layer, which represented the mean of denoised data and two parameters of ZINB distribution respectively. It modeled special sparsity structure and inferenced *dropout* events of scRNA-seq data. On the basis of DCA, the scDeepCluster [23] added an extra clustering layer inspired by a deep learning clustering method of improved deep clustering (IDEC) [41], and it can iteratively update clustering assignment after trained the DCA. The scDeepCluster outperformed DCA in the performance of clustering task and became state-of-the-art approach for scRNA-seq clustering.

C. SUMMARY OF EXISTING METHODS

In general, previous researchers have improved traditional clustering algorithms or used deep learning algorithms to implement clustering analysis on scRNA-seq data and achieved better performances.

As for the algorithms using the *k*-means method, the RaceID3 [28] and LAK [29] algorithms have effectively improved the traditional *k*-means algorithm by using the random forest components and the Lasso penalty respectively. Also, they achieved better performance than other *k*-means based algorithms. On the benchmark dataset 10X PBMC, RaceID3 achieved about 69%, 70% and 55% on the three evaluation metrics of Clustering Accuracy (ACC), Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI), respectively. As for the LAK algorithm, the three metrics are improved to 78%, 75% and 68%. Regarding the three spectral clustering based methods, including SIMLR [15], MPSSC [16] and SinNLRR [30], the last algorithm combined the advantages of the first two algorithms and used a unique low rank structure on the cell similarity matrix, which achieved about 77%, 74% and 66% on the above three metrics (The performance of the first two algorithms are: SIMLR: 62%, 72%, 52%, and MPSSC: 76%, 73%, 65%). For other traditional methods, Seurat [32] and SCANPY [33] are more used for scRNA-seq data preprocessing or coarse-grained analysis. And the innovative use of reference database for cell type identification of scCATCH [34] reached 83%, 76% and 73% on the ACC, NMI and ARI under 10X PBMC dataset, which also shows competitive performance compared to the *k*-means based and spectral clustering based algorithms.

The methods based on deep learning mainly use autoencoder as the core component. As a representative of the earlier work, the performance of the AutoImpute [38] and IDEC [41] algorithms on the 10X PBMC dataset reached about 72%, 71%, 61% and 70%, 70%, 55% on the three metrics respectively. Afterwards, some work such as DCA [22], VASC [39] and scvis [21] improved the previous algorithm from different starting points, which made the clustering performance improved. Among them, the scDeepCluster [23] method combined the previous researchers' modeling of the special sparsity and *dropout* noise of scRNA-seq data. Also, scDeepClsuter leveraged the deep embedded clustering method to make the two processes of denoising and clustering can be jointly trained and optimized, achieving the best performance on the scRNA-seq data clustering task. It reached 82%, 77% and 72% on the three metrics under 10X PBMC dataset, and also reached better clustering performance under other scRNA-seq datasets (all the numerical performances for each algorithms are shown in Table 4).

Although the above methods have achieved certain clustering performance, they still suffer from some shortcomings [7], [13], [31], [42]. *K*-means based algorithms are not good at directly determining the optimal value of the number of clusters, and are not good at handling samples with unbalanced data clusters [13], [42]. The computational time complexity and computational space consumption of algorithms based on spectral clustering are very huge, making them not suitable for the current large-scale data analysis [7], [31]. The accuracy of the scCATCH algorithm depends on the selection of good reference datasets, while the scDeepCluster algorithm lacks constraints on hidden layer features, which may be unfavorable for the subsequent clustering.

III. PROPOSED FRAMEWORK

In this section, the baseline model scDeepCluster is introduced first. Then, the proposed **Adversarial AutoEncoder ScRNA Clustering (AAE-SC)** model is described, alongside the training and optimization process of our method.

As shown in Fig. 2, AAE-SC consists of an AAE module with noise, three independent layers at the end of the decoder of AAE to estimate ZINB loss and the improved deep clustering layer.

A. ScDeepCluster METHOD

ScDeepCluster is proposed by Tian [23], which consists of a denoised autoencoder (DAE) with a specific ZINB [22] loss and an IDEC [41] layer.

To make the autoencoder more robust, the DAE incorporates an extra Gaussian noise into the input samples, and attempts to reconstruct the original input from corrupted data. In DAE, both the encoder and decoder are composed of fully connected layers which are low-dimensional compared to the raw data. By reconstructing the clean data, the hidden layer learns effective low-dimensional feature representation.

Although common practice tends to employ the mean-square error (MSE) loss to fulfill the reconstruction process in traditional autoencoder and DAE, the scRNA-seq data is too sparse that the MSE loss cannot rebuild the original data

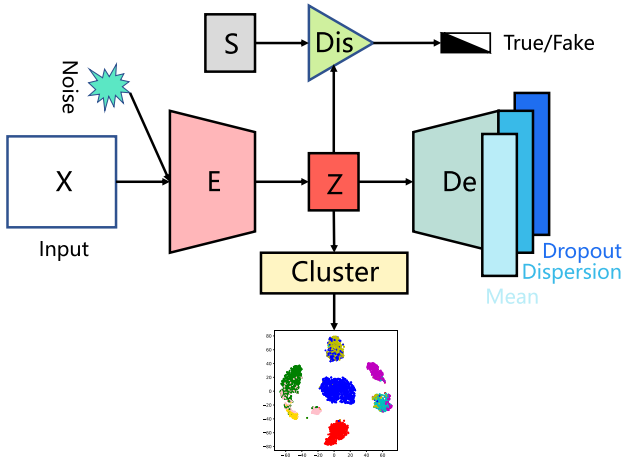


FIGURE 2. The architecture of our AAE-SC. Dis is the discriminator of AAE and S stands for the random samples from the prior distribution. Mean, Dispersion and Dropout are ZINB parameters.

well. Therefore scDeepCluster utilizes a specific loss function based on ZINB distribution from DCA. This distribution has shown its effectiveness to model the highly sparse and overdispersed data. ZINB can be estimated by the mean (μ) and dispersion (θ) of the negative binomial distribution and the additional coefficient (π) which represents the probability of *dropout*:

$$NB(X|\mu,\theta) = \frac{\Gamma(X+\theta)}{X!\Gamma(\theta)}\left(\frac{\theta}{\theta+\mu}\right)^{\theta}\left(\frac{\mu}{\theta+\mu}\right)^{X}, \quad (1)$$

$$ZINB(X|\pi,\mu,\theta) = \pi\delta_0(X) + (1-\pi)NB(X|\mu,\theta), \quad (2)$$

where X stands for the original data. The scDeepCluster uses three independent fully connected layer at the end of decoder to estimate the above parameters.

To better perform the clustering task, scDeepCluster also employs the method of IDEC method in the latent space features instead of using traditional clustering algorithm such as k-means directly. After obtaining the latent space features from the hidden layer of DAE, scDeepCluster uses the same clustering approach with IDEC. The method first computes the distribution Q of soft clustering labels in sample features, and then defines an auxiliary target distribution P based on Q. Finally the clustering loss is defined as Kullback-Leibler (KL) divergence between P and Q, which is illustrated as below:

$$q_{ij} = \frac{(1+\|z_i-\mu_j\|^2)^{-1}}{\sum_{j'}(1+\|z_i-\mu_j\|^2)^{-1}}, \quad (3)$$

$$p_{ij} = \frac{q_{ij}^2/\sum_j q_{ij}}{\sum_{j'}(q_{ij'}^2/\sum_{j'} q_{ij'})}, \quad (4)$$

where q_{ij} is the soft label of embedded sample z_i. This variable is used to measure the similarity between sample z_i and cluster center μ_j by Student's t-distribution. After that, scDeepCluster iteratively uses the self-training strategy to compute the auxiliary target distribution p_{ij} with previous q_{ij}.

B. AAE-SC FRAMEWORK

In addition to modeling and constraining the reconstructed data output by the decoder with a special prior ZINB distribution, we also constrain the prior distribution of the bottleneck feature of DAE to preserve the latent data structure and generate more consistent feature. Recent researches use variational inference like AAE [24] to match the aggregated posterior of the latent features of the autoencoder with an arbitrary prior distribution, and they have been proved to be effective in many fields. Therefore we modify the DAE in scDeepCluster to an AAE by adding a discriminator D on top of the bottleneck layer, and use the original encoder as a generator.

Based on the implementation of DAE in scDeepCluster, the input data X^{input} is corrupted by a zero-mean Gaussian random noise ε and becomes X^{noise}. We define the encoder and decoder functions as $Z = F_{W_E}(X^{noise})$ and $G_{W_D}(Z)$, where Z stands for the latent space feature in the bottleneck layer. The weight W_E and W_D are learning parameters of encoder and decoder respectively. In addition to the raw data, we also add the zero-mean Gaussian random noise to each layer of the encoder and make the model more robust.

Similar to the generative adversarial network (GAN) [43], AAE uses an adversarial training procedure on the autoencoder and a discriminator to match the aggregated posterior of the hidden vector with the prior distribution, which aims to learn a better mapping function and hidden code. However, the purpose of AAE and GAN are completely different. GAN uses an adversarial training method to learn the data distribution of the original data, so the random noise can be converted into new data similar to the raw data through a generator. Whereas, AAE is trained to make the hidden layer feature of the autoencoder conform to a prior distribution. So the purpose of GAN is to generate new data, while the goal of AAE is to restrict the data distribution of existing features. In this article we adopted AAE to make constraint on the hidden feature so that it can be clustering-friendly.

The additional discriminator of AAE is also composed of fully connected layers. Meanwhile, the final layer's output dimension is set to be 1, which is to determine the authenticity of input samples. The input of the discriminator are the latent features from the bottleneck layer of DAE and the random sampling data from the prior distribution with the same dimensions as the former. The generated data from prior distribution is true data and its label is set to 1, while the label of latent feature is set to 0, which is regarded as fake data. The discriminator network utilizes the binary cross entropy loss to train and update parameters:

$$L_d = \frac{1}{n}\sum_{i=1}^{n}[log(D(s_i)) + log(1-D(z_i))], \quad (5)$$

where s_i, z_i and n stand for the generated samples from prior distribution, the latent feature and the batch size respectively. After updating the parameters of discriminator D, all weights inside are frozen.

Unlike the GAN structure, which has an independent generator, the adversarial autoencoder trains the encoder part as a generator to confuse the discriminator D, and let D judge whether the input samples generated by encoder are true samples:

$$L_g = \frac{1}{n}\sum_{i=1}^{n} logD(z_i). \tag{6}$$

Through the above adversarial training process, the hidden features are aligned to the prior distribution and the whole AAE framework learns a good mapping function from input data to a low-dimensional feature which is suitable for the downstream cluster analysis.

In addition to the variance inference by AAE, our method also employs the ZINB loss as the reconstruction loss and utilizes an IDEC layer. To estimate the three parameters (π, μ, θ) of ZINB distribution described above, the last layer of the decoder is replaced with three independent fully connected layers and their dimensions are the same with the input data. Thus the architecture of the decoder is given as below (Z represents the output of bottleneck layer in AAE-SC):

$$De = G_{W_D}(Z), \tag{7}$$

$$\bar{M} = exp(W_M De) \times diag(sf), \tag{8}$$

$$\Phi = sigmoid(W_\pi De), \tag{9}$$

$$\Theta = exp(W_\theta De), \tag{10}$$

where W_M, W_π and W_θ are the learning parameters in the final three fully connected layers respectively. The size factor sf is an independent biological variable that is calculated by the library size and the median of cells. The reconstruction loss function of the ZINB distribution is the negative log transformation of the ZINB likelihood:

$$L_r = -log(ZINB(X|\pi, \mu, \theta)). \tag{11}$$

AAE-SC also has an IDEC layer on top of the hidden layer of AAE. Based on the above description, the clustering loss is computed by KL-divergence between P and Q as below:

$$L_c = KL(P\|Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}. \tag{12}$$

C. TRAINING STRATEGY & OPTIMIZATION

In this paper, our model has two stages: 1) Combination of the adversarial training and reconstruction stage, which aims to constrain the prior distribution of the hidden layer coding while reconstructing the noisy original data. 2) Jointly optimizing the reconstruction loss and clustering loss on the constrained features listed above to iteratively update the clustering label assignment. The objective function of model is defined as below:

$$L_1 = L_r + L_g, \tag{13}$$

$$L_2 = L_r + \alpha L_c, \tag{14}$$

where α is a clustering coefficient to adjust the clustering loss to avoid the clustering space to be distorted. Loss in the pre-training phase corresponding to L_1, and L_2 represents the target function in clustering process.

As for the above loss function, the three types of parameters can be optimized and updated by Stochastic Gradient Descent (SGD) and back propagation.

Specifically, as described in [41], [44], the gradient of L_c with respect to the clustering center μ_j and latent feature sample z_i can be computed as below:

$$\frac{\partial L_c}{\partial \mu_j} = -2\sum_i (1 + \|z_j - \mu_i\|^2)^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j), \tag{15}$$

$$\frac{\partial L_c}{\partial z_i} = 2\sum_j (1 + \|z_j - \mu_i\|^2)^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j), \tag{16}$$

And during the clustering process the clustering center μ_j is updated by:

$$\mu_j = \mu_j - \frac{l}{n}\sum_{i=1}^{n} \frac{\partial L_c}{\partial \mu_j}, \tag{17}$$

where l is the learning rate and n is the value of mini batch.

The decoder weights W_D are updated by:

$$W_D = W_D - \frac{l}{n}\sum_{i=1}^{n} \frac{\partial L_r}{\partial W_D}, \tag{18}$$

In the stage 1 the encoder weights W_E are updated by:

$$W_E = W_E - \frac{l}{n}\sum_{i=1}^{n} (\frac{\partial L_r}{\partial W_E} + \frac{\partial L_g}{\partial W_E}), \tag{19}$$

And in the stage 2 the encoder weights W_E are updated by:

$$W_E = W_E - \frac{l}{n}\sum_{i=1}^{n} (\frac{\partial L_r}{\partial W_E} + \alpha \frac{\partial L_c}{\partial W_E}). \tag{20}$$

IV. EXPERIMENTS

In this section, we provide quantitative comparisons of the proposed AAE-SC model to other state-of-the-art scRNA-seq clustering methods in two categories: traditional clustering models and deep learning models.

A. DATASETS

The proposed AAE-SC model is evaluated on three real-world scRNA-seq datasets coming from different sequencing platforms. All the datasets are publicly available. The statistics of the datasets are summarized in Table 2 and the detailed information is shown as follows:

TABLE 2. Datasets statistics.

Dataset	Cells	Genes	Clusters
10X PBMC [46]	4271	16449	8
Mouse Bladder Cells [47]	2746	19079	16
Worm Neuron Cells [48]	4186	11955	10

TABLE 3. Brief introduction of experimental methods.

Method	Description	Reference
PCA + k-means	Uses PCA to extract the main data features, and uses k-means for clustering	[48]
GAN + k-means	Uses GAN to generate samples similar to the original data, and uses k-means for clustering	[43]
SIMLR	Spectral clustering with multiple Gaussian kernel similarity measures	[15]
MPSSC	Multi-kernel spectral clustering with L1 penalty	[16]
SAIC	Uses iterative k-means algorithm for clustering	[26]
RaceID3	Uses random forest and improved k-means for clustering	[28]
SinNLRR	Spectral clustering with a non-negative and low rank structure on similarity matrix	[30]
AutoImpute	Uses overcomplete autoencoder to regenerate the impute expression matrix	[38]
DEC	Uses encoders with trained parameters to extract features, and uses k-means for clustering	[44]
IDEC	Improve version of DEC, jointly optimize the data clustering and data reconstruction	[41]
scvis	Uses variational autoencoder for clustering	[21]
VASC	Uses variational autoencoder for data reduction and clustering	[39]
DCA	Uses deep count autoencoder and ZINB noised model to characterize data	[22]
LAK	Uses Lasso penalty to select gene features, and uses k-means for clusterng	[29]
scCATCH	Uses reference scRNA-seq database to assist cell type annotation	[34]
scDeepCluster	Combines the advantages of DCA model and IDEC model	[23]
our AAE-SC	Uses AAE module to add additional constraints on hidden features of scDeepCluster	

- **10X PBMC** [45][1]: This dataset is downloaded from the 10X scRNA-seq platform. It measures the transcriptome of the peripheral blood mononuclear cells collected from a healthy donor [45]. There are over 4,000 cells with 16,000 genes in the dataset and it has 8 different clusters.
- **Mouse Bladder Cells** [46][2]: This dataset comes from the Mouse Cell Atlas project by [46]. We select the bladder tissue cell data from overall 400,000 single cells, and they can be divided into 16 different groups.
- **Worm Neuron Cells** [47][3]: It is a worm cell dataset profiled by the sci-RNA sequencing platform. About 50,000 cells from the nematode Caenorhabditis elegans at the L2 larval stage in the dataset have been measured by previous researchers and the corresponding cell types have been identified. Following the approach in [47], we select the subset of these neural cells and remove the unlabeled individuals. Therefore, the dataset we use consists of 4,186 cells with over 10,000 genes and there are 10 different categories in total.

B. EXPERIMENTAL METHODS

To evaluate the performance of our proposed AAE-SC, we compare it with sixteen algorithms, which are the representative and widely used works in scRNA-seq clustering. The descriptions of these methods are summarized in Table 3.

C. EVALUATION METRICS

In our experiments, three metrics of ACC, NMI and ARI are used to evaluate AAE-SC model, which are widely used in unsupervised learning scenario. Introduction of these metrics are as follows:

- **ACC**: The clustering accuracy (ACC) is used to measure the matching level of the clustering labels assigned to the samples and their true labels. Given the sample i, the assignment label p_i and its groundtruth label t_i, the ACC is computed as:

$$ACC = \frac{\sum_{i=1}^{n} \delta(t_i, map(p_i))}{n}, \tag{21}$$

$$\delta(x, y) = \begin{cases} 1, & \text{if } x = y, \\ 0, & \text{else}, \end{cases} \tag{22}$$

  where n is the number of sample points and $map()$ refers to best mapping between assigned labels and true labels. It can be solved by the Hungarian algorithm with polynomial time.
- **NMI**: The Normalized Mutual Information (NMI) measures the similarity of two clusters from the perspective of information theory. It is defined as:

$$NMI = \frac{I(T, P)}{\max\{H(T); H(P)\}}, \tag{23}$$

$$I(T, P) = \sum_i \sum_j \frac{|t_i \cap p_j|}{n} log \frac{n|t_i \cap p_j|}{|t_i||p_j|}, \tag{24}$$

$$H(P) = -\sum_j \frac{|p_j|}{n} log \frac{|p_j|}{n}, \tag{25}$$

  where I(T,P) represents the mutual information between the ground truth label T and the model-predicted assigned label P. $H()$ denotes the entropy of the labels and n is the batch size.
- **ARI**: The Adjusted Rand Index (ARI) evaluates the similarity between two clustering results by computing the pair relationship improved from the original RI (Rand Index). Given ground truth label T and the predicted clustering results assignment P, we first compute the four mathematical quantities:
  - a: the number of sample pairs which are divided into the same cluster in both T and P.
  - b: the number of sample pairs which are divided into different clusters in T and P.
  - c: the number of sample pairs which are divided into the same cluster in P but different in T.

[1]https://support.10xgenomics.com/single-cell-gene-expression/datasets/2.1.0/pbmc4k
[2]https://figshare.com/s/865e694ad06d5857db4b
[3]http://atlas.gs.washington.edu/worm-rna/docs/

TABLE 4. Experimental results on 10X PBMC, Mouse Bladder Cells and Worm Neuron Cells. Best results are shown in bold.

	10X PBMC			Mouse Bladder Cells			Worm Neuron Cells		
Method	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
PCA+k-means [48]	56.93	63.58	48.62	43.71	59.57	37.79	43.02	37.59	22.92
GAN+k-means [43]	53.27	59.73	48.12	40.25	55.93	35.01	41.27	37.19	23.93
SIMLR [15]	62.13	72.29	51.93	52.75	69.36	44.70	68.02	64.11	38.97
MPSSC [16]	76.29	73.59	65.87	60.75	70.52	48.23	63.96	53.87	40.69
SAIC [26]	63.72	65.79	52.30	51.62	62.33	40.25	44.22	39.52	25.79
RaceID3 [28]	69.25	70.53	55.89	61.72	71.32	50.12	61.28	42.60	35.72
SinNLRR [30]	77.65	73.80	65.95	63.29	70.83	49.37	65.26	56.89	45.23
AutoImpute [38]	72.53	71.36	59.28	53.26	60.37	44.63	55.72	50.92	36.89
DEC [44]	61.62	60.53	52.05	49.82	57.89	35.72	37.26	38.92	20.95
IDEC [41]	70.25	69.95	55.97	55.68	61.42	43.76	45.16	43.91	30.28
scvis [21]	85.30	76.35	75.05	49.36	68.56	40.53	56.57	52.35	35.67
VASC [39]	79.68	75.12	73.02	60.75	70.52	48.23	63.96	53.87	40.69
DCA [22]	74.92	73.57	73.26	59.85	65.23	50.79	60.23	51.82	36.15
LAK [29]	78.33	75.68	68.58	65.13	71.58	52.87	62.36	59.35	46.29
scCATCH [34]	83.62	76.50	73.98	62.75	70.95	55.66	66.36	65.32	50.72
scDeepCluster [23]	82.58	77.52	72.91	68.39	74.74	57.73	67.31	**70.24**	52.03
our AAE-SC	**87.26**	**81.31**	**81.32**	**71.03**	**76.99**	**61.06**	**70.52**	68.29	**52.50**

– d: the number of sample pairs which are divided into different clusters in P but the same in T.

Base on the above quantities, the ARI is defined as:

$$ARI = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c)+(d+b)(d+c)}{\binom{n}{2}^2 - [(a+b)(a+c)+(d+b)(d+c)]}. \tag{26}$$

The value range of ACC and NMI is both [0,1], and that of ARI is [-1,1] of ARI. For all the three metrics, a larger score indicates a more accurate clustering result.

D. IMPLEMENTATION DETAILS

In experiments, the proposed AAE-SC network architecture is constructed with the same layers as that of the baseline model, scDeepCluster. Meanwhile, the encoder network dimensions is set to *input* – 128 – 64 – 32, where *input* stands for the dimension of input data, and the decoder has a symmetric structure with the encoder. Besides, the discriminator network is built with dimensions 32 – 128 – 64 – 32 – 1. The reasons of setting the first layer and its symmetrical layer to be 128 nodes will be discussed in the following section. Furthermore, the activation function of the last layer of discriminator is *sigmoid*, while other fully-connected layers are all activated by *ReLU*. In the pretraining stage, we utilize the optimizer Adam with learning rate 0.001 for all the datasets. As for the clustering phase, the optimizer Adadelta is applied with learning rate 1.0. We discussed the choice of initial learning rate in subsequent experiments.

Here the standard normal distribution $N(0, 1)$ is exploited as the prior distribution to align the bottleneck feature. And the corresponding group number is employed to be the number of clusters for the clustering layer as prior information on each dataset. All weights in fully-connected layers of the proposed AAE-SC model are initialized with the Glorot uniform. The whole model is pretrained by 300 iterations first, then the clustering stage start. Meanwhile, special experiments are conducted to determine the more appropriate value of parameter α in the following section. The rest hyperparameters are set to be the same as scDeepCluster.

E. DATA PRE-PROCESSING

All scRNA-seq data were pre-processed using SCANPY [33], following the data pre-processing implementation of the baseline scDeepCluster: For each dataset, genes with expression values less than 5 and cells with expression values less than 1 were first filtered out. Then, the entire reading matrix was normalized so that each cell has a total count equal to the median of the gene counts of per cell before normalization. After that, logarithmic transformation was applied on the data, and it was scaled to unit variance and zero mean.

V. RESULTS AND DISCUSSION

A. QUANTITATIVE ANALYSIS

The clustering analysis results on three real-world scRNA-seq datasets are listed in Table 4. All experimental data are the average of 10 independent experimental results.

The proposed model is first compared with the three traditional methods: PCA+k-means, SIMLR and MPSSC. PCA+k-means is considered as a typical traditional method, both the PCA and k-means are very commonly utilized in clustering process. Compared to the PCA+k-means method, AAE-SC demonstrates huge superiority, with an overall increase of 17%-32% on all of the three datasets. Since the PCA method only focuses on reducing the dimension of the data rather than extract effective features for clustering, resulting in poor final clustering effect. By employing spectral clustering, SIMLR and MPSSC achieve a remarkable improvement over PCA+k-means method. Although spectral clustering is better than the ordinary PCA+k-means method, SIMLR cannot model the large amount of noise and *dropout* events present in scRNA-seq data effectively. The MPSSC adds an additional L1 penalty on the basic of spectral clustering, so its performance is better than SIMLR. However, this artificially designed constraint does not fully

TABLE 5. Wilcoxon Signed-Rank Test results on 10X PBMC, Mouse Bladder Cells and Worm Neuron Cells. The suffix _TS in the header stands for test significance for each metric. Each row in the table represents the test significance of the comparison algorithm and the original experimental data of AAE-SC after the Wilcoxon Signed-Rank Test.

	10X PBMC			Mouse Bladder Cells			Worm Neuron Cells		
Compared Method	ACC_TS	NMI_TS	ARI_TS	ACC_TS	NMI_TS	ARI_TS	ACC_TS	NMI_TS	ARI_TS
PCA+k-means	0.02	0.009	0.008	0.01	0.005	0.015	0.017	0.009	0.011
GAN+k-means	0.023	0.016	0.019	0.021	0.013	0.017	0.012	0.007	0.01
SIMLR	0.02	0.013	0.021	0.024	0.016	0.013	0.019	0.005	0.007
MPSSC	0.01	0.018	0.019	0.022	0.016	0.015	0.011	0.003	0.012
SAIC	0.012	0.021	0.023	0.015	0.013	0.016	0.013	0.029	0.02
RaceID3	0.023	0.015	0.017	0.027	0.018	0.019	0.015	0.019	0.013
SinNLRR	0.025	0.013	0.021	0.011	0.025	0.013	0.012	0.009	0.008
AutoImpute	0.012	0.007	0.015	0.017	0.021	0.011	0.018	0.003	0.006
DEC	0.012	0.018	0.005	0.007	0.011	0.013	0.015	0.008	0.011
IDEC	0.022	0.02	0.013	0.015	0.012	0.011	0.016	0.009	0.01
scvis	0.029	0.02	0.021	0.017	0.019	0.013	0.013	0.007	0.012
VASC	0.025	0.021	0.015	0.012	0.013	0.015	0.013	0.005	0.009
DCA	0.023	0.017	0.012	0.018	0.015	0.013	0.016	0.01	0.012
LAK	0.025	0.018	0.013	0.02	0.018	0.017	0.017	0.008	0.015
scCATCH	0.026	0.02	0.012	0.018	0.02	0.016	0.019	0.009	0.013
scDeepCluster	0.027	0.025	0.021	0.019	0.01	0.019	0.02	0.051	0.013

model the essential characteristics of scRNA-seq data. As a result, their performance is inferior to the method proposed in this paper. Meanwhile, our method is also superior to other algorithms including the RaceID3, LAK, SinNLRR and scCATCH, which are based on *k*-means algorithm, spectral clustering algorithm and reference databases respectively. It can be clearly seen that our algorithm is superior to these algorithms on each dataset.

DEC and IDEC are the early deep learning methods which used autoencoder for clustering. For IDEC, the decoder structure is retained for subsequent clustering on the basis of DEC, and it is apparent that IDEC performs better than DEC on all the three datasets. However, because the scRNA-seq data is quite different from the traditional image data, and these two algorithms are not specifically designed for the task of scRNA-seq data clustering. Experimental results of the two algorithms on this kind of data are even worse than traditional MPSSC method, and this holds true for the AutoImpute algorithm. On the other hand, although DCA, scvis and VASC characterize the scRNA-seq data by a specific ZINB loss and variance inference model VAE respectively, all of them ignore taking the advantage of deep clustering. Therefore, they only achieve the similar performance with traditional spectral clustering algorithm, restraining the ability of deep learning to process big data.

The baseline model, scDeepCluster, follows the method of DEC and IDEC and adds an extra clustering layer connecting the hidden layer of DCA model. In this way, the scDeepCluster not only effectively models and describes scRNA-seq data through ZINB distribution, but also strengthens the effect of subsequent clustering tasks by the cLustering layer. In this case, it outperforms the above methods and becomes the previous state-of-the-art. Compared with scDeepCluster, our improved model constrains the hidden layer data to prevent distortion of the data structure during the learning and clustering process, thus showing remarkable improvement on 10X PBMC and Mouse Bladder Cells dataset. Especially in the experiment of 10X PBMC, our model exceeds the original scDeepCluster by about 5% on both ACC and ARI metrics. This confirms the importance of maintaining data structures in hidden layer and the effect of AAE for improving the clustering performance.

The GAN [43] model with a confrontation training process similar to that of AAE is also compared, and it is found that the GAN model plus *k*-means algorithm performs the worst among all comparison algorithms. The reason is that although the GAN model uses similar adversarial training ideas, the essence of GAN is to generate data rather than extract useful features, and this model is not suitable for cluster analysis.

In addition, all of our original experimental performances have been tested by Wilcoxon Signed-Rank Test with other algorithms. The results are shown in Table 5. It is clear that when the significance level is 0.05, the performance of our algorithm on all metrics is significant different compared with the above algorithms, which also proves statistically that our algorithm is better.

B. QUALITATIVE ANALYSIS

As described above, the scDeepCluster is improved by adding an extra clustering layer on DCA first, and AAE-SC impose constraint on the hidden features of scDeepCluster by AAE. To evaluate clustering effects and effectiveness of AAE-SC versus the baseline methods more intuitively, we visualize the hidden embedded representations of AAE-SC, scDeepCluster and DCA on 10X PBMC dataset using TSNE [49].

In Fig. 3, it is obvious that the samples in the same cell group distribute in a wide range and cannot be well clustered in DCA. Taking advantage of the extra clustering layer, the performance of scDeepCluster is significantly better than DCA. Although similar cells are made compact and dense by scDeepCluster, some different clusters cannot be separated well (such as cluster 2&6 and cluster 4&5). Our AAE-SC overcomes the above problems and clusters the cell samples

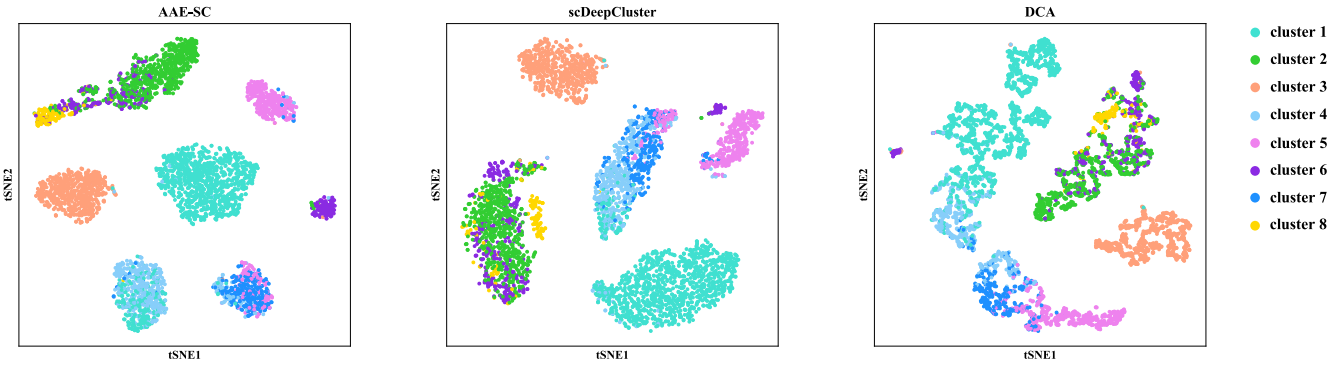


FIGURE 3. Comparison of 3D visualization of hidden embedded representation on the 10X PBMC. From left to right in the picture: AAE-SC, scDeepCluster and DCA. Different colors indicate different cell clusters.

well into different groups, which will be beneficial for the subsequent biological analysis.

C. SELECTIONS OF THE NUMBER OF CLUSTERS

Since the k-means algorithm is utilized in the improved deep embedding clustering (IDEC) method of AAE-SC, and the k-means algorithm demands the number of clusters in the data in advance. In order to determine the optimal number of clusters, additional experiments on the above three datasets were conducted.

For each dataset, the number of clusters used by the k-means algorithm in the experiment was chosen within the range of length 2 around the number of cell types in the dataset. For the above three datasets of 10X PBMC, Mouse Bladder Cells and Worm Neuron Cells, the selection range of the number of clusters fall into $[6, 10]$, $[14, 18]$ and $[8, 12]$ respectively. The clustering performance on these three datasets is shown in Fig. 4. It can be clearly seen that on each dataset, the clustering performance is the best when the number of clusters is equal to that of cell types in the dataset.

We also utilize the metric Davies-Bouldin Index (DBI) [51] to determine the optimal clustering effect when selecting the number of clusters. Introduction of this metric is as follows:

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{avg(C_i)+avg(C_j)}{d_{cen}(C_i, C_j)}\right). \tag{27}$$

where $avg()$ represents average distance between samples in the cluster, and $d_{cen}()$ represents the distance between the center points of two clusters. The smaller the value of DBI, the better the clustering effect can be considered.

Fig. 5 shows the DBI value under different number of clusters on three datasets. It can be obviously seen that on each dataset, the DBI value is the best when the number of clusters is equal to that of cell types in the dataset. This also verifies the results of our above experiment. So, the number of cell types in each dataset is taken as the number of clusters for the algorithm to reach the best clustering performance.

D. THE ROBUSTNESS OF AAE-SC

Robustness of the AAE-SC model was also studied and compared to that of the baseline model, scDeepCluster. SCANPY [33] was exploited to downsample the above three datasets, and limited the total gene counts of each cell sample in the dataset to the range of [500, 1000, 1500], so that the clustering performance of the two models under these noisy data could be observed.

Fig. 6 shows the clustering performances of AAE-SC and scDeepCluster on the downsampled datasets. AAE-SC performs better than the baseline model on three different down-sampled noise data in almost every dataset. Especially when the downsampling reaches only 500 gene counts per cell, the NMI metric of AAE-SC outperforms that of the baseline model by about 10%, and the other two metrics by 5%. The superior performance indicates that the constrained features extracted by AAE-SC model are more robust than the hidden layer features of the scDeepCluster baseline model in case of high-noise data.

E. EXPERIMENTS ON UNBALANCED DATASETS

In addition, the ability of the AAE-SC model to handle unbalanced datasets were studied. The unbalanced dataset means that the number of samples of certain cell types (clusters) in the dataset is much smaller than that of the dataset. In this experiment, three representative unbalanced single-cell datasets extracted from three mouse tissues in the Tabula Muris project [50] have been selected and some rare data clusters exist in these datasets. Relevant information of the datasets is demonstrated in Table 6. Obviously, there are clusters with very few samples in these datasets. The data pre-processing method we use on these three data sets is the same as the method above.

Fig. 7 exhibits the clustering performance of AAE-SC versus that of the baseline model on three unbalanced datasets. It can be observed that the AAE-SC model performs at least 5% better than the baseline model scDeepCluster on all three metrics. Especially on the Skin dataset, AAE-SC outperforms the scDeepCluster model by at least 12% on each metric. The above experimental results can show that the AAE-SC model

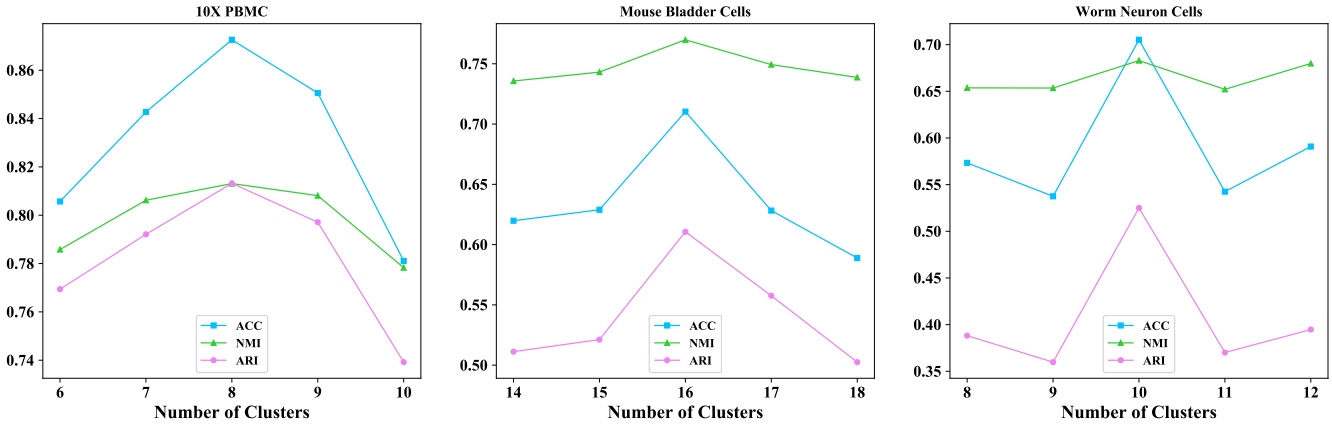


FIGURE 4. Comparison of clustering performance under different numbers of data clusters on three datasets. From left to right in the picture: 10X PBMC, Mouse Bladder Cells and Worm Neuron Cells. Different colors indicate different evaluation metrics.

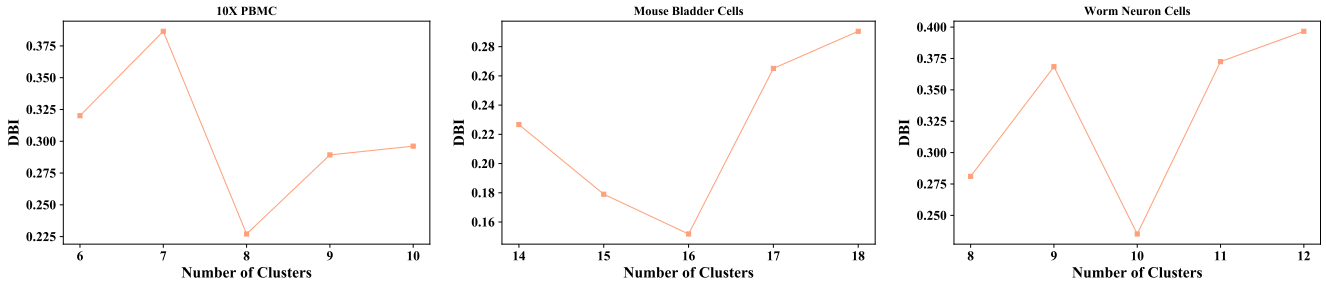


FIGURE 5. Comparison of values of Davies-Bouldin Index under different numbers of data clusters on three datasets. From left to right in the picture: 10X PBMC, Mouse Bladder Cells and Worm Neuron Cells.

TABLE 6. Unbalanced datasets statistics. Rare clusters refer to clusters that account for less than 5% of the dataset.

Dataset	Cells	Clusters	Number of rare clusters	Proportion of rare cluster samples
Thymus [50]	9275	6	1	2.15%
Large Interstine [50]	12295	5	2	9.57%
Mammary Gland [50]	4454	7	3	12.36%

has better clustering performance than the baseline model in case of the unbalanced datasets with scarce data clusters.

F. SELECTIONS OF THE INITIAL LEARNING RATE FOR TWO OPTIMIZERS

The selections of the initial learning rate for the two optimizers in our AAE-SC model were also studied. In addition to the learning rate used in the baseline model scDeepCluster (0.001 for Adam and 1 for Adadelta), we also compared the performance of the two optimizers with other learning rates. The learning rate of Adam is set as 0.0001, 0.001, 0.01 and 0.1 successively, and which of Adadelta is set as 0.01, 0.1, 1 and 10 successively.

Fig. 8 exhibits the clustering performance of AAE-SC under different learning rates for the two optimizers on 10X PBMC dataset. It can be clearly seen that when the learning rates of Adam and Adadelta are 0.001 and 1, respectively, the clustering performance of the model is best. At the same time, the above two learning rates are also the default learning rates of the two optimizers in the baseline model scDeepCluster, so in order to make a fair comparison and achieve the best performance, we recommend using these two values as the initial learning rate of the two optimizers.

G. HYPER-PARAMETER ANALYSIS

The effect of the clustering coefficient α on the clustering performance is further investigated. In this study, we aim to find a suitable value of α, so that the final clustering effect can be improved. Meanwhile, we hope that the final model will not be overly sensitive to the changes of coefficient α and the performance of the model not fluctuates too much. Therefore, the effect of different network widths on the model performance was also studied, by changing the width of first layer of the adversarial autoencoder network.

In order to make the optimization process more efficient and to better explore the solution space, we chose to use

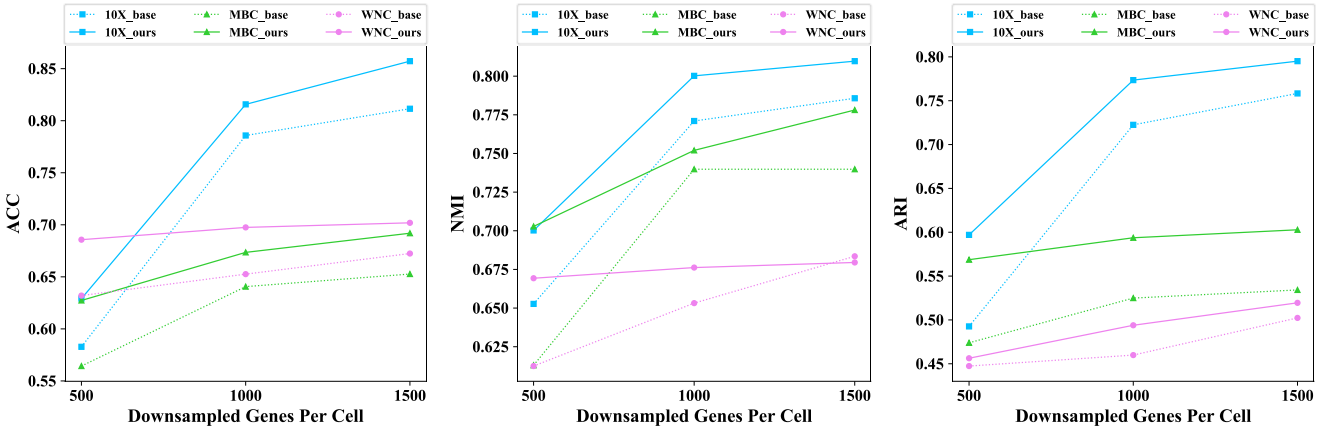


FIGURE 6. Comparison of the clustering performance of AAE-SC and baseline model scDeepCluster on three scRNA-seq datasets which are downsampled. From left to right in the picture are: ACC, NMI and ARI respectively. The solid line represents our model, while the dashed line represents the baseline model scDeepCluster.

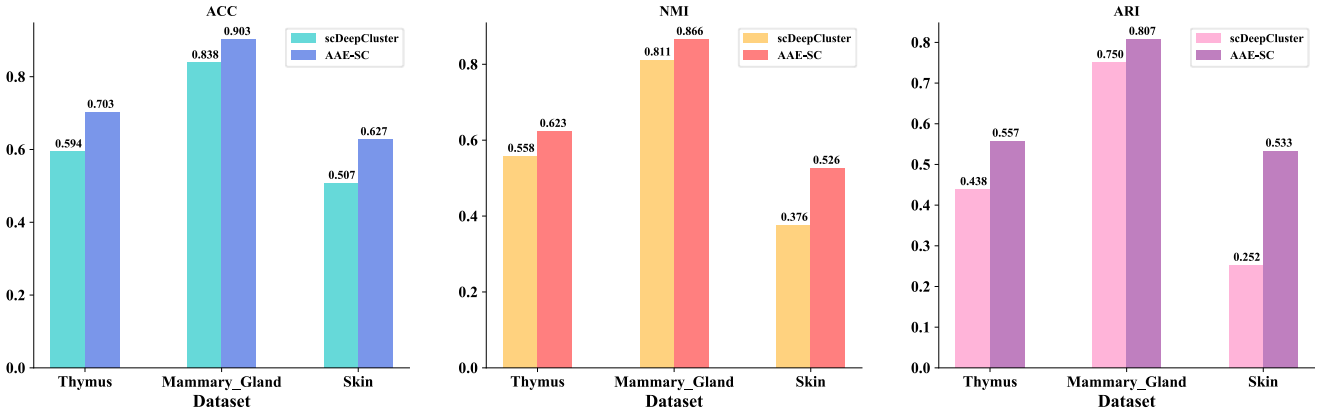


FIGURE 7. Comparison of the clustering performance of AAE-SC and baseline model scDeepCluster on three unbalanced scRNA-seq datasets. From left to right in the picture are: ACC, NMI and ARI respectively.

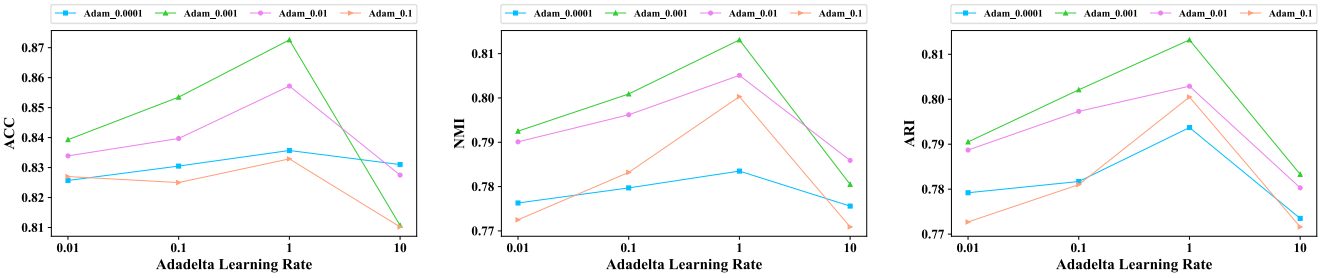


FIGURE 8. CComparison of the clustering performance for 10X PBMC dataset of two optimizers Adam and Adadelta with different initial learning rates. From left to right: ACC, NMI and ARI. The learning rate of Adam is set as 0.0001, 0.001, 0.01 and 0.1 successively, and the learning rate of Adadelta is set as 0.01, 0.1, 1 and 10 successively.

Bayesian optimization to optimize these two parameters. Specifically, we used the python package BayesianOptimization to make the analysis. For clustering coefficient α and the width of first layer in AAE-SC, we defined the search space as [0.1, 10] and [64,512] respectively in advance, and then performed experiments. The result of the experiment is that the optimal parameters of the two under Bayesian optimization are 1.50023 and 128.0076 respectively, so we finally chose to set these two parameters to 1.5 and 128. It is worth mentioning that this setting can make the width of the network narrower than the baseline model.

VI. CONCLUSION

In this paper, we propose AAE-SC, a single-cell RNA-seq data clustering model. The model combines benefits of modeling of specific biological noise modeling, variational inference and deep clustering. The AAE-SC model preserves the data structure and utilizes the constraint bottleneck feature to improve clustering analysis by an AAE module. Experimental results on three real-world scRNA-seq datasets show that AAE-SC achieves considerable better clustering performance than the state-of-the-art on three evaluation metrics.

Although the proposed model achieves superior performance, we believe that it has some limitations, including AAE may not be easy to train, and the training time may be relatively long, etc., so our future work will also focus on improving the above problems. Also, it is significant to explore the utilization of this model for processing other single-cell data, such as single-cell Hi-C data and scATAC-seq data. Besides, using XAI technology to improve the interpretability of this model is worth to explore.

REFERENCES

[1] E. Shapiro, T. Biezuner, and S. Linnarsson, "Single-cell sequencing-based technologies will revolutionize whole-organism science," *Nature Rev. Genet.*, vol. 14, no. 9, pp. 618–630, Sep. 2013.

[2] T. Terabayashi, G. G. Germino, and L. F. Menezes, "Pathway identification through transcriptome analysis," *Cellular Signalling*, vol. 74, Oct. 2020, Art. no. 109701.

[3] S. Baek and I. Lee, "Single-cell ATAC sequencing analysis: From data preprocessing to hypothesis generation," *Comput. Struct. Biotechnol. J.*, vol. 18, pp. 1429–1439, 2020.

[4] I. Govorov, S. Sitkin, T. Pervunina, A. Moskvin, D. Baranenko, and E. Komlichenko, "Metabolomic biomarkers in gynecology: A treasure path or a false path?" *Current Medicinal Chem.*, vol. 27, no. 22, pp. 3611–3622, Jun. 2020.

[5] M. Villar, L. Mateos-Hernandez, and J. de la Fuente, "The impact of post-genomics approaches in neurodegenerative demyelinating diseases: The case of Guillain-Barré syndrome," *Current Medicinal Chem.*, vol. 25, no. 29, pp. 3482–3490, Sep. 2018.

[6] A. A. Kolodziejczyk, J. K. Kim, V. Svensson, J. C. Marioni, and S. A. Teichmann, "The technology and biology of single-cell RNA sequencing," *Mol. Cell*, vol. 58, no. 4, pp. 610–620, May 2015.

[7] M. D. Luecken and F. J. Theis, "Current best practices in single-cell rna-seq analysis: A tutorial," *Mol. Syst. Biol.*, vol. 15, no. 6, 2019, Art. no. e8746.

[8] Z.-X. Guan, S.-H. Li, Z.-M. Zhang, D. Zhang, H. Yang, and H. Ding, "A brief survey for MicroRNA precursor identification using machine learning methods," *Current Genomics*, vol. 21, no. 1, pp. 11–25, Mar. 2020.

[9] G. Bitencourt-Ferreira and W. F. de Azevedo, "Machine Learning to predict binding affinity," in *Docking Screens for Drug Discovery*. New York, NY, USA: Humana, 2019, pp. 251–273.

[10] A. D. da Silva, G. Bitencourt-Ferreira, and W. F. Azevedo, "Taba: A tool to analyze the binding affinity," *J. Comput. Chem.*, vol. 41, no. 1, pp. 69–73, Jan. 2020.

[11] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1. 1967, pp. 281–297.

[12] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. ICML*, 2008, pp. 1096–1103.

[13] V. Y. Kiselev, T. S. Andrews, and M. Hemberg, "Challenges in unsupervised clustering of single-cell RNA-seq data," *Nature Rev. Genet.*, vol. 20, no. 5, pp. 273–282, May 2019.

[14] C. Xu and Z. Su, "Identification of cell types from single-cell transcriptomes using a novel clustering method," *Bioinformatics*, vol. 31, no. 12, pp. 1974–1980, Jun. 2015.

[15] B. Wang, J. Zhu, E. Pierson, D. Ramazzotti, and S. Batzoglou, "Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning," *Nature methods*, vol. 14, no. 4, p. 414, 2017.

[16] S. Park and H. Zhao, "Spectral clustering based on learning similarity matrix," *Bioinformatics*, vol. 34, no. 12, pp. 2069–2076, Jun. 2018.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[18] L. Koumakis, "Deep learning models in Genomics; Are we there yet?" *Comput. Struct. Biotechnol. J.*, vol. 18, pp. 1466–1473, Jun. 2020.

[19] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[20] Y. Deng, F. Bao, Q. Dai, L. F. Wu, and S. J. Altschuler, "Massive single-cell rna-seq analysis and imputation via deep learning," *BioRxiv*, Jan. 2018, Art. no. 315556.

[21] J. Ding, A. Condon, and S. P. Shah, "Interpretable dimensionality reduction of single cell transcriptome data with deep generative models," *Nature Commun.*, vol. 9, no. 1, pp. 1–13, Dec. 2018.

[22] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, "Single-cell RNA-seq denoising using a deep count autoencoder," *Nature Commun.*, vol. 10, no. 1, pp. 1–14, Dec. 2019.

[23] T. Tian, J. Wan, Q. Song, and Z. Wei, "Clustering single-cell RNA-seq data with a model-based deep learning approach," *Nature Mach. Intell.*, vol. 1, no. 4, pp. 191–198, Apr. 2019.

[24] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, *arXiv:1511.05644*. [Online]. Available: http://arxiv.org/abs/1511.05644

[25] D. Grün, A. Lyubimova, L. Kester, K. Wiebrands, O. Basak, N. Sasaki, H. Clevers, and A. van Oudenaarden, "Single-cell messenger RNA sequencing reveals rare intestinal cell types," *Nature*, vol. 525, no. 7568, pp. 251–255, Sep. 2015.

[26] L. Yang, J. Liu, Q. Lu, A. D. Riggs, and X. Wu, "SAIC: An iterative clustering approach for analysis of single cell RNA-seq data," *BMC Genomics*, vol. 18, no. S6, pp. 9–17, Oct. 2017.

[27] D. Grün, M. J. Muraro, J.-C. Boisset, K. Wiebrands, A. Lyubimova, G. Dharmadhikari, M. van den Born, J. van Es, E. Jansen, H. Clevers, E. J. P. de Koning, and A. van Oudenaarden, "De novo prediction of stem cell identity using single-cell transcriptome data," *Cell Stem Cell*, vol. 19, no. 2, pp. 266–277, Aug. 2016.

[28] J. S. Herman and D. Grün, "FateID infers cell fate bias in multipotent progenitors from single-cell RNA-seq data," *Nature Methods*, vol. 15, no. 5, pp. 379–386, May 2018.

[29] J. Hua, H. Liu, B. Zhang, and S. Jin, "LAK: Lasso and K-means based single-cell RNA-seq data clustering analysis," *IEEE Access*, vol. 8, pp. 129679–129688, 2020.

[30] R. Zheng, M. Li, Z. Liang, F.-X. Wu, Y. Pan, and J. Wang, "SinNLRR: A robust subspace clustering method for cell type detection by non-negative and low-rank representation," *Bioinformatics*, vol. 35, no. 19, pp. 3642–3650, Oct. 2019.

[31] S. Zhang, X. Li, Q. Lin, and K.-C. Wong, "Review of single-cell RNA-seq data clustering for cell type identification and characterization," 2020, *arXiv:2001.01006*. [Online]. Available: http://arxiv.org/abs/2001.01006

[32] R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev, "Spatial reconstruction of single-cell gene expression data," *Nature Biotechnol.*, vol. 33, no. 5, pp. 495–502, May 2015.

[33] F. A. Wolf, P. Angerer, and F. J. Theis, "SCANPY: Large-scale single-cell gene expression data analysis," *Genome Biol.*, vol. 19, no. 1, p. 15, Dec. 2018.

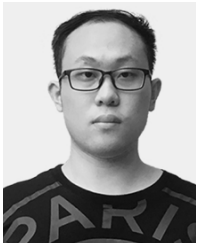
[34] X. Shao, J. Liao, X. Lu, R. Xue, N. Ai, and X. Fan, "ScCATCH: Automatic annotation on cell types of clusters from single-cell RNA sequencing data," *iScience*, vol. 23, no. 3, Mar. 2020, Art. no. 100882.

[35] M. M. Xavier, G. S. Heck, M. B. de Avila, N. M. B. Levin, V. O. Pintro, N. L. Carvalho, and W. F. de Azevedo, "SAnDReS a computational tool for statistical analysis of docking results and development of scoring functions," *Combinat. Chem. High Throughput Screening*, vol. 19, no. 10, pp. 801–812, Dec. 2016.

[36] M. Wójcikowski, P. Siedlecki, and P. J. Ballester, "Building machine-learning scoring functions for structure-based prediction of intermolecular binding affinity," in *Docking Screens for Drug Discovery*. New York, NY, USA: Humana, 2019, pp. 1–12.

[37] C. Lin, S. Jain, H. Kim, and Z. Bar-Joseph, "Using neural networks for reducing the dimensions of single-cell RNA-seq data," *Nucleic Acids Res.*, vol. 45, no. 17, p. e156, Sep. 2017.

[38] D. Talwar, A. Mongia, D. Sengupta, and A. Majumdar, "AutoImpute: Autoencoder based imputation of single-cell RNA-seq data," *Sci. Rep.*, vol. 8, no. 1, pp. 1–11, Dec. 2018.
[39] D. Wang and J. Gu, "VASC: Dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder," *Genomics, Proteomics Bioinf.*, vol. 16, no. 5, pp. 320–331, Oct. 2018.
[40] D. P Kingma and M. Welling, "Auto-encoding variational bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114
[41] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1753–1759.
[42] R. Petegrosso, Z. Li, and R. Kuang, "Machine learning and statistical methods for clustering single-cell RNA-sequencing data," *Briefings Bioinf.*, vol. 21, no. 4, pp. 1209–1223, Jul. 2020.
[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[44] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.
[45] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, and M. T. Gregory, "Massively parallel digital transcriptional profiling of single cells," *Nature Commun.*, vol. 8, no. 1, pp. 1–12, 2017.
[46] X. Han, R. Wang, Y. Zhou, L. Fei, H. Sun, S. Lai, A. Saadatpour, Z. Zhou, H. Chen, and F. Ye, "Mapping the mouse cell atlas by microwell-seq," *Cell*, vol. 172, no. 5, pp. 1091–1107, 2018.
[47] J. Cao, J. S. Packer, V. Ramani, D. A. Cusanovich, C. Huynh, R. Daza, X. Qiu, C. Lee, S. N. Furlan, F. J. Steemers, A. Adey, R. H. Waterston, C. Trapnell, and J. Shendure, "Comprehensive single-cell transcriptional profiling of a multicellular organism," *Science*, vol. 357, no. 6352, pp. 661–667, Aug. 2017.
[48] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 4, pp. 433–459, 2010.
[49] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[50] T. M. Consortium, "Single-cell transcriptomics of 20 mouse organs creates a tabula muris," *Nature*, vol. 562, no. 7727, p. 367, 2018.
[51] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.



YULUN WU received the B.S. degree in systems engineering from the National University of Defense Technology, Changsha, China, in 2018, where he is currently pursuing the M.S. degree in control science and engineering.

His current research interests include unsupervised learning, adversarial machine learning, and deep learning.

YANMING GUO received the B.S. and M.S. degrees from the National University of Defense Technology, Changsha, China, in 2011 and 2013, respectively, and the Ph.D. degree from the Leiden Institute of Advanced Computer Science (LIACS), Leiden University, in 2017.

He is currently an Associate Professor with the College of Systems Engineering, National University of Defense Technology. His current interests include computer vision, natural language processing, and deep learning. He has served as a Reviewer for many journals, such as TIP, TNNLS, TMM, *Neurocomputing,* and MTAP.

YANDONG XIAO received the B.S. degree in information system engineering and the Ph.D. degree in system engineering from the National University of Defense Technology, Changsha, China, in 2012 and 2018, respectively, and the Ph.D. degree from Harvard University, Boston, MA, USA, where he was also a Research Trainee.

He is currently a Lecturer with the College of System Engineering, National University of Defense Technology. His research interests include complex system modeling, artificial intelligence, swarm intelligence, interdisciplinary principles for community ecology, and social networks.

SONGYANG LAO received the B.S. degree in information system engineering and the Ph.D. degree in system engineering from the National University of Defense Technology, Changsha, China, in 1990 and 1996, respectively.

He was a Visiting Scholar with the Dublin City University, Ireland, from 2004 to 2005. He is currently a Professor with the College of Systems Engineering. His current research interests include deep learning, image processing, video analysis, and human–computer interaction.