Computational and Structural Biotechnology Journal 27 (2025) 1090–1102

Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Research Article

scCCTR: An iterative selection-based semi-supervised clustering model for single-cell RNA-seq data

Jie Chen, Qiucheng Sun *, Chunyan Wang *, Changbo Gao

School of Computer Science and Technology, Changchun Normal University, Changchun, 130032, China

ARTICLE INFO

Keywords:
scRNA-seq data
Clustering
Low-rank representation
Attention mechanism
Consensus constraint

ABSTRACT

Single-cell RNA sequencing (scRNA-seq) enables the analysis of the genome, transcriptome, and epigenome at the single-cell level, providing a critical tool for understanding cellular heterogeneity and diversity. Cell clustering, a key step in scRNA-seq data analysis, reveals population structure by grouping cells with similar expression patterns. However, due to the high dimensionality and sparsity of scRNA-seq data, the performance of existing clustering algorithms remains suboptimal. In this study, we propose a novel clustering algorithm, scCCTR, which performs semi-supervised classification by guiding a deep learning model through iterative selection of high-confidence cells and labels. The algorithm consists of two main components: an iterative selection module and a semi-supervised classification module. In the iterative selection module, scCCTR progressively selects high-confidence cells that exhibit core group features and iteratively optimizes feature representations, constructing a consensus clustering result throughout the iterations. In the semi-supervised classification module, scCCTR uses the selected core data to train a Transformer neural network, which leverages a multi-head attention mechanism to focus on critical information, thereby achieving higher clustering precision. We compared scCCTR with several established cell clustering methods on real datasets, and the results demonstrate that scCCTR outperforms existing methods in terms of accuracy and effectiveness for both cell clustering and visualization. (The code of scCCTR is free available for academic https://github.com/chenjiejie387/scCCTR).

1. Introduction

Single-cell RNA sequencing (scRNA-seq) enables high-throughput analysis of mRNA expression at the single-cell level, providing researchers with a high-resolution tool for comprehensive gene expression profiling [1]. This technology reveals gene expression differences between cell population [2], allowing not only the identification of cell type-specific molecular characteristics but also the discovery of novel cell subtypes and rare cell types [3] [4]. It serves as a foundation for in-depth understanding of cellular functional states and regulatory mechanisms [5] [6]. Additionally, scRNA-seq is widely applied in tumor microenvironment and immune system research, uncovering complex intercellular interaction networks and their roles in disease progression, thus supporting disease mechanism elucidation and the discovery of novel therapeutic targets, thereby advancing precision medicine [7–9].

Despite the tremendous potential of scRNA-seq in studying cellular heterogeneity, its clustering analysis still faces several challenges [10]. The high dimensionality and sparsity of scRNA-seq data [11] limit the effectiveness of traditional clustering algorithms [12] [13]. Furthermore, technical noise [14] and batch effects [15] further compromise the accuracy of clustering. These noise sources are due to differences in sample processing and sequencing platforms, often obscuring true biological variation among cells. Therefore, developing algorithms capable of effectively handling these data characteristics is crucial for improving clustering accuracy and for a deeper exploration of cellular biological features.

With the rapid development of single-cell RNA sequencing (scRNA-seq) technology, dimensionality reduction techniques and cell similarity metrics have been widely applied in various clustering methods. Linear dimensionality reduction methods, such as Principal Component Analysis (PCA), effectively simplify the feature space by extracting major sources of variance [16]. Building on this foundation, the Seurat toolkit constructs cell similarity networks using graph theory and employs community detection algorithms, such as Louvain [17] and Leiden [18], to achieve efficient clustering. Linear dimensionality reduction methods, such as PCA, are unable to capture the nonlinear structures

* Corresponding authors.
E-mail addresses: startcj@163.com (J. Chen), sunqiucheng@ccsfu.edu.cn (Q. Sun), wangchunyan@ccsfu.edu.cn (C. Wang), QX202200148@stu.ccsfu.edu.cn (C. Gao).

https://doi.org/10.1016/j.csbj.2025.03.018
Received 14 December 2024; Received in revised form 28 February 2025; Accepted 10 March 2025
Available online 14 March 2025

inherent in high-dimensional single-cell sequencing data. To address this, pcaReduce employs recursive PCA decomposition and clustering, progressively grouping cells to enhance sensitivity to subtle differences between populations [19]. Similarly, CIDR, introduced by Lin et al., integrates PCA with hierarchical clustering to capture multi-layered similarity structures [20]. Researchers have also explored combining multiple similarity metrics to address data heterogeneity and multimodal characteristics. For example, SIMLR leverages multi-kernel learning to combine diverse similarity metrics, effectively handling heterogeneous and multimodal scRNA-seq data [21]. SC3 integrates the outputs of multiple clustering algorithms, using multi-view aggregation to mitigate biases from individual approaches [22].

Although these methods have significantly improved clustering performance, they remain limited in handling the nonlinear features of scRNA-seq data, prompting the development of more sophisticated nonlinear dimensionality reduction and clustering strategies. Building on the traditional autoencoder (AE) framework, the DCA model uses a denoising autoencoder to remove technical noise [23]. Due to the limited interpretability of AE-based dimensionality reduction, Lopez et al. proposed scVI, which incorporates variational inference to model scRNA-seq data as a latent probabilistic distribution. This approach not only introduces a probabilistic framework but also enables scVI to effectively address batch effects and technical noise [24]. Further advancements include VASC, which combines sparse coding with variational autoencoders (VAE) to better capture sparse data features [25]. To overcome the independence of dimensionality reduction and clustering processes, Tian et al. proposed scDeepCluster, which jointly optimizes dimensionality reduction and clustering, allowing the reduced space and clustering results to dynamically adjust in tandem [26].

As the potential of graph neural networks (GNNs) in single-cell analysis continues to grow, the integration of autoencoders with GNNs has significantly enhanced the ability to capture complex intercellular relationships. scGNN maps scRNA-seq data to graph structures and use graph autoencoders to aggregate and refine cell network information, optimizing the construction of cell similarity matrices [27]. Notably, cellVGAE combines variational autoencoders with attention mechanisms, dynamically adjusting weights to enhance the interpretability of k-nearest neighbor (KNN) graphs and the robustness of the model [28]. However, the performance of these methods heavily depends on the quality of the KNN graph, inaccurate graph construction can compromise the model's reliability. To address the dropout issue caused by technical noise [29], Hu et al. proposed scDSC, which integrates a zero-inflated negative binomial (ZINB) autoencoder with GNNs to capture the global graph structure and improve robustness against technical noise [30]. Additionally, Wang et al. developed the scGCL model, which applies contrastive learning strategies to optimize cell embeddings from multiple perspectives [31].

While unsupervised clustering methods based on deep learning have improved clustering accuracy, relying solely on unsupervised approaches often fails to fully exploit the latent information in high-noise and highly heterogeneous single-cell data. In contrast, semi-supervised clustering methods leverage partially labeled data as prior information to enhance clustering precision. scANVI, an extension of scVI, incorporates partially labeled data to refine cell type predictions, thereby enhancing performance in cell annotation tasks and indirectly improving clustering accuracy [32]. Berthelot et al.'s MixMatch method reduces label uncertainty through label smoothing while integrating data augmentation and consistency loss to align labeled and unlabeled data in the feature space [33]. Additionally, Liger utilizes non-negative matrix factorization to integrate labeled and unlabeled data, identifying and comparing the biological states of different cell types [34]. SDEC combines deep embedding with partial supervision, performing semi-supervised clustering directly in the embedding space by optimizing a joint loss function [35]. However, accurately labeling single cells requires complex experimental procedures and is cost-prohibitive, making large-scale annotation increasingly challenging as single-cell sequencing datasets continue to grow.

To address the limitations of existing single-cell clustering methods in handling high-sparsity and high-noise data while fully exploring the deep information embedded in gene expression data, we propose a semi-supervised clustering model named scCCTR. This model replaces external prior information by iteratively selecting high-confidence cells and labels. scCCTR comprises two main modules. In the iterative selection module, a residual connection encoder is first employed to perform dimensionality reduction and denoising on high-dimensional scRNA-seq data, followed by K-means clustering to quickly identify the basic population structure. Subsequently, scCCTR selects a subset of cells that represent the core characteristics of each cluster and uses sub-autoencoders to reconstruct the core gene expression data. The process captures shared local expression patterns within clusters through iterations, achieving convergence when cell clustering no longer undergoes significant changes, and optimizes core cluster features through data reconstruction. The process also optimizes the core feature expression of clusters through data reconstruction. Additionally, scCCTR introduces a consensus constraint strategy, which constructs a consensus matrix combined with hierarchical clustering to provide consistency constraints for clustering results during iterations. Experimental results demonstrate that this design effectively enhances the model's ability to capture data structures, reducing instability caused by random initialization or noise. In the semi-supervised classification module, to overcome the limitations of autoencoders in capturing global features, scCCTR integrates a Transformer-based semi-supervised classification module leveraging a multi-head attention mechanism [36]. This module uses the core cell data and corresponding cluster labels identified during the iterative selection to train a Transformer neural network, improving clustering accuracy. The main contributions of scCCTR are as follows:

1) Iteratively selecting differential cells representing core population characteristics, effectively filtering technical noise and improving data quality through sub-autoencoder reconstruction.

2) Integrating a consensus constraint module to consolidate unstable solutions during iterations, reducing uncertainty caused by random initialization or data noise.

3) Utilizing a multi-head attention mechanism to deeply explore cell expression differences, effectively handling long-range feature interactions and compensating for the inability of traditional autoencoders to capture global features.

To evaluate the model's performance, we conducted experiments on sixteen real datasets. The results demonstrate that compared to eight existing clustering methods, scCCTR exhibits significant advantages in handling high-sparsity and high-noise data, achieving higher clustering accuracy and showing great potential in uncovering cellular developmental trajectories.

2. Methods

As shown in Fig. 1, scCCTR consists of two main modules. The function of the first module is to select data and corresponding labels that can represent the core characteristics of each cluster, while the second module utilizes this data to train a Transformer neural network, achieving more accurate clustering results. Throughout the process, scCCTR fully explores and leverages the latent information within the data itself, gradually optimizing data representation to achieve precise classification. The corresponding algorithm is presented in Algorithm 1.

2.1. Iterative selection module

This module iteratively selects subsets of cells from the complete dataset that represent the core characteristics of each cell population. The iterative selection process includes the initial clustering phase, consensus clustering phase, and data reconstruction phase.

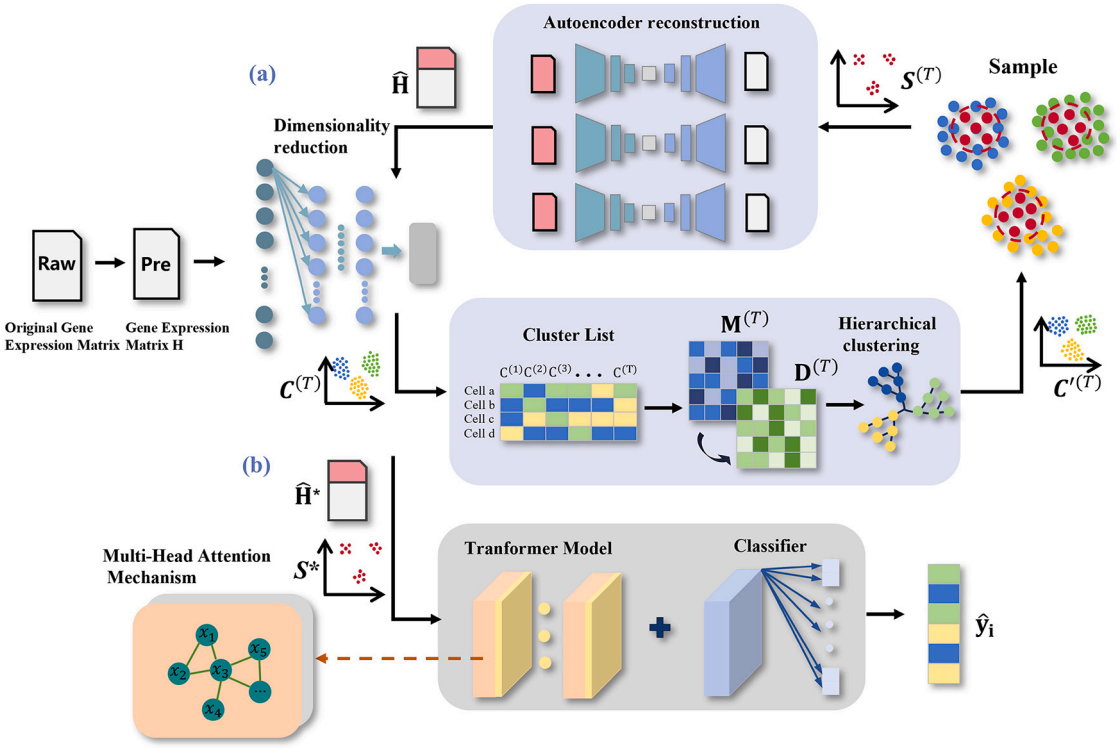


Fig. 1. Workflow of scCCTR. (a) The initial data undergoes Preprocessing to obtain the input matrix **H**. For T iterations, dimensionality reduction is first performed on **H**, followed by K-means clustering to obtain the clustering set $C^{(T)}$ for the T-th iteration. Next, the sequence of T clustering sets is integrated to compute the consensus matrix $\mathbf{M}^{(T)}$ and distance matrix $\mathbf{D}^{(T)}$, which are used to iteratively update the clustering results $C^{(T)}$, replacing $C^{(T)}$ in each iteration. Finally, core subsets $S^{(T)}$ are refined for each cluster, and data reconstruction is performed by matching subsets with sub-autoencoders. The updated matrix $\hat{\mathbf{H}}$ is used for dimensionality reduction. This iterative process is designed to capture high-confidence data. (b) After selection, the semi-supervised classification module first uses the core clustering set S^* from (a) to select core base data for pre-training a Transformer neural network. Then, the reconstructed matrix $\hat{\mathbf{H}}^*$ is used as input to predict the final clustering labels.

Algorithm 1 Pseudocode of scCCTR Algorithm.

Input: Gene expression matrix **H**
Output: Clustering labels y_i'
Stage 1
1: Compute **Z** by (1);
2: Calculate $C^{(t)}$ using the k-means algorithm;
3: Aggregate the clustering ensemble list $\{C^{(1)}, C^{(2)}, \ldots, C^{(T)}\}$, compute $\mathbf{M}^{(T)}$ and $\mathbf{D}^{(T)}$ by (2) to (3);
4: Compute $C'^{(T)}$ by (4) and calculate $\mu = \{\mu_1, \mu_2, \ldots, \mu_k\}$;
5: Filter $S^{(T)}$ and $\mathbf{H}'$ by (6);
6: Reconstruct and impute **H** by (7) and (8) to obtain $\hat{\mathbf{H}}$;
7: Iterate until convergence to output S^* and $\hat{\mathbf{H}}^*$;
Stage 2
8: Select $\hat{\mathbf{H}}'$ based on the cell indices S^* and $\hat{\mathbf{H}}^*$;
9: Train the Transformer model using S^* and $\hat{\mathbf{H}}'$ by (10) to (15);
10: Generate clustering labels y_i' by (16).

2.1.1. *Initial clustering phase*

In the initial stage of data selection, scCCTR employs a multi-layer neural network with a residual autoencoder to project the original high-dimensional data into a nonlinear embedding space. To mitigate gradient vanishing issues in deep networks and improve feature stability, residual connections are introduced in each layer of the autoencoder. These connections not only provide transformed feature representations but also retain the core information from the previous layer.

The input matrix for the autoencoder is defined as $\mathbf{H}=[x_1, x_2, \ldots, x_N]$, where x_i represents the i-th cell sample, and N is the number of cells. The dimensionality reduction representation at the l-th layer of the autoencoder is formulated as:

$$z_i^{(l)} = f\left(W^{(l)} z_i^{(l-1)} + b^{(l)}\right) + z_i^{(l-1)} \tag{1}$$

Where $z_i^{(l-1)}$ is the feature representation of x_i at layer $l-1$, $\mathbf{W}^{(l)}$ and $b^{(l)}$ are the weight matrix and bias term at layer l, and f denotes the nonlinear activation function PReLU [37]. Through dimensionality reduction, the model generates low-dimensional embeddings $\mathbf{Z} = \{z_1, z_2, \ldots, z_N\}$, significantly reducing data dimensionality and suppressing noise.

To progressively construct clustering structures, the features **Z** are subjected to K-means clustering to achieve efficient partitioning. This step provides an interpretable initial clustering structure, represented as $C = \{C_1, C_2, \ldots, C_k\}$.

2.1.2. *Consensus clustering phase*

To further improve the consistency and robustness of clustering, scCCTR introduces a consensus clustering module that fully leverages the iterative deep representation learning of the autoencoder. This module constructs a consensus clustering result by integrating clustering outcomes across multiple iterations, thereby improving consistency and stability.

Specifically, in the t-th iteration, we apply K-means clustering on low-dimensional embeddings **Z** of the data to obtain clustering results $C^{(t)} = \{C_1^{(t)}, C_2^{(t)}, \ldots, C_k^{(t)}\}$, where $C_k^{(t)}$ represents the k-th cluster obtained in the t-th iteration.

After T rounds of selection, we integrate the clustering results from each iteration $\{C^{(1)}, C^{(2)}, \ldots, C^{(T)}\}$ and construct a consensus matrix $\mathbf{M}^{(T)} \in \mathbb{R}^{N \times N}$. The element $\mathbf{M}_{ij}^{(T)}$ represents the frequency with which samples x_i and x_j are assigned to the same cluster across T iterations, and is defined as:

$$\mathbf{M}^{(T)} = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}\left(C^{(t)} 1^{\top} = 1 \left(C^{(t)}\right)^{\top}\right) \tag{2}$$

Where $\mathbf{1} \in \mathbb{R}^{1\times N}$ is a row vector of all ones. $\mathbb{I}(\cdot)$ is an indicator function that equals 1 when $C_i^t = C_j^t$, indicating that samples x_i and x_j belong to the same cluster in the t-th iteration, and 0 otherwise.

The consensus matrix is invariant to permutations of cluster labels. For instance, relabeling clusters (e.g., swapping labels 1 and 2) in any iteration does not alter the value of $\mathbb{I}(C_i^t = C_j^t)$, since the indicator function depends solely on the equivalence of labels, not their absolute values.

Next, the consensus matrix $M^{(T)}$ is transformed into a distance matrix $D^{(T)}$, where $D_{ij}^{(T)}$ measures the similarity between samples x_i and x_j in the T-th iteration. The distance matrix is defined as:

$$\mathbf{D}^{(T)} = 1 - \mathbf{M}^{(T)} \tag{3}$$

Based on the distance matrix $D^{(T)}$, iteratively calculate pairwise cluster distances to solidify and refine clusters until the target number of k clusters is achieved [38]. The consensus clustering outcome for the T-th iteration is $C'^{(T)} = \{C_1'^{(T)}, C_2'^{(T)}, \ldots, C_k'^{(T)}\}$. The distance between clusters A and B is defined as the average of the pairwise distances between all samples in A and B:

$$d(A,B) = \frac{1}{|A||B|}\sum_{i\in A}\sum_{j\in B}\mathbf{D}_{ij}^{(T)} \tag{4}$$

Where $|A|$ and $|B|$ denote the number of samples in clusters A and B, respectively, and $\mathbf{D}_{ij}^{(T)}$ represents the distance between samples x_i and x_j in the T-th iteration.

Finally, the consensus clustering result $C'^{(T)}$ replaces the K-means clustering result $C^{(T)}$, enabling stable and consistent clustering across multiple iterations, thereby enhancing model robustness in noisy and sparse data scenarios.

To avoid excessive computational costs during iterative selection, an ARI-based termination criterion is introduced for the consensus clustering process. After obtaining the consensus clustering result for the T-th iteration, we compute the similarity between the current and previous consensus clustering results, $C'^{(T)}$ and $C'^{(T-1)}$. A threshold δ is defined to measure the difference between two consecutive iterations. The process terminates when:

$$\left|\mathrm{ARI}\left(C'^{(T)}, C'^{(T-1)}\right)\right| < \delta \tag{5}$$

If the ARI difference between the current and previous consensus clustering results is smaller than δ, the clustering result is considered stable, and the selection process can terminate. Otherwise, iterations continue until the maximum number is reached or the condition is satisfied.

2.1.3. Data reconstruction phase

To reduce noise and improve clustering accuracy, this method selects high-confidence cells located near the cluster centroids to represent the core characteristics of each population. Specifically, for the j-th cluster $C_j'^{(T)}$, where $j = 1, \cdots, k$. We calculate the Euclidean distance between each sample's low-dimensional embedding z_i and the centroid μ_j:

$$d\left(z_i, \mu_j\right) = \left\|z_i - \mu_j\right\| \tag{6}$$

Based on the Euclidean distance, we select a certain proportion of samples closest to the centroid to construct the core clustering subset $S^{(T)} = \{S_1^{(T)}, S_2^{(T)}, \ldots, S_k^{(T)}\}$. Next, based on the selected core indices, we extract submatrices from the original matrix $\mathbf{H} = \{\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_k\}$ to form the core matrix set $\mathbf{H}' = \{\mathbf{H}_1', \mathbf{H}_2', \cdots, \mathbf{H}_k'\}$, and establish an independent autoencoder for each submatrix $\mathbf{H}_j'$ to perform reconstruction, where $j = 1, \cdots, k$. The reconstruction process is defined as follows:

$$z_i^{(l)} = f(\mathbf{W}^{(l)} z_i^{(l-1)} + b^{(l)}) \tag{7}$$

$$h_i^{(l)} = g(\mathbf{V}^{(l)} h_i^{(l-1)} + c^{(l)}) \tag{8}$$

Where $\mathbf{W}^{(l)}$ and $b^{(l)}$ are the weight matrix and bias term of the l-th encoder layer, $\mathbf{V}^{(l)}$ and $c^{(l)}$ are the weight matrix and bias term of the l-th decoder layer. f and g are nonlinear activation functions. We optimize the parameters of the autoencoder by minimizing the mean squared error (MSE) loss between the reconstructed matrix and the input matrix, with the loss function defined as:

$$L = \frac{1}{N}\sum_{i=1}^{N}\left\|x_i - \hat{x}_i\right\|^2 \tag{9}$$

Finally, the reconstructed core data are reinserted into the original matrix $\mathbf{H}$, generating an updated high-dimensional matrix $\hat{\mathbf{H}}$ for further dimensionality reduction and clustering. Through this iterative selection process, the model can learn the core characteristics of the data. Upon completion of this phase, the final core clustering result S^* and reconstructed complete basis matrix $\hat{\mathbf{H}}^* = \left\{\hat{\mathbf{H}}_1^*, \hat{\mathbf{H}}_2^*, \cdots, \hat{\mathbf{H}}_k^*\right\}$ are obtained.

2.2. Transformer semi-supervised module

To effectively extract features from complex single-cell core basis data for classification, this module establishes a Transformer-based semi-supervised classification framework. By pre-training on a small set of labeled core samples, the model is capable of extracting effective features from unlabeled data for classification. After the selection phase, the core indices of the core clustering set $S^* = \{S_1^*, S_2^*, \ldots, S_k^*\}$ are used to extract the core basis matrix $\hat{\mathbf{H}}'$ from the full matrix $\hat{\mathbf{H}}^*$, where $\hat{\mathbf{H}}' \in \mathbb{R}^{N'\times D}$, N' denotes the number of core cells, and D is the number of features, $\hat{\mathbf{H}}'$ will be used for pre-training of the Transformer model.

To further adapt the Transformer model for processing high-dimensional data, the input features are first mapped to a latent dimension d_{model} through a linear transformation. Additionally, the transformed feature matrix undergoes layer normalization to eliminate discrepancies in feature distributions and improve model efficiency. The input matrix $\tilde{\mathbf{X}}$ after linear transformation and layer normalization is represented as::

$$\tilde{\mathbf{X}} = \mathrm{LayerNorm}\left(\mathbf{X}\mathbf{W}_e + b_e\right) \tag{10}$$

Where $W_e \in \mathbb{R}^{D\times d_{\text{model}}}$ is the weight matrix, and b_e is the bias term.

The normalized feature matrix is then fed into the Transformer encoder to capture latent complex relationships between samples. The encoder consists of multi-head self-attention layers and feed-forward neural network (FFN) layers. Each attention head computes the similarity between the query ($\mathbf{Q}$), key ($\mathbf{K}$), and value($\mathbf{V}$) matrices to capture complex dependencies between input samples. The attention calculation is defined as:

$$\mathrm{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_K}}\right)\mathbf{V} \tag{11}$$

Where $\mathbf{Q} = \tilde{\mathbf{X}}\mathbf{W}_Q, \mathbf{K} = \tilde{\mathbf{X}}\mathbf{W}_K, \mathbf{V} = \tilde{\mathbf{X}}\mathbf{W}_V$, and $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d_{\text{model}}\times d_k}$ are the projection matrices, and d_k is the dimension of each attention head. The multi-head attention mechanism uses multiple attention heads in parallel to capture different dependencies in the input features. The concatenated outputs from all heads are then linearly transformed to obtain a comprehensive feature representation:

$$\mathrm{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{Concat}(\mathrm{head}_1, \cdots, \mathrm{head}_h)\mathbf{W}_O \tag{12}$$

Where $\mathbf{W}_O \in \mathbb{R}^{h\cdot d_h\times d_{\text{model}}}$ is the linear projection matrix that maps the multi-head attention outputs back to the original feature dimension. The feed-forward neural network (FFN) captures both local and global features by applying linear transformations and non-linear activations to the output of the attention layer. The computation is defined as:

$$FNN(\mathbf{Z}) = \mathrm{ReLU}(\mathbf{Z}\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2 \tag{13}$$

Where $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{model}}\times d_f}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_{\text{model}}\times d_f}$ are the weight matrices of the feed-forward network, b_1 and b_2 are the bias terms, and d_{ff} is the hidden dimension of the feed-forward network. Finally, the en-

Table 1
Summary of the sixteen scRNA-seq datasets used in experiments.

Datasets	Cells	Genes	Cell Type	Species	Protocol
Baise	56	25734	4	Mouse	NCBI
Goolam	124	23386	5	Mouse	EMBL_EBI
Klein	2717	24175	4	Mouse	NCBI
Zeisel	3005	19972	9	Mouse	NCBI
Muraro	2122	19140	9	Human	NCBI
Baron	8569	20125	14	Human	NCBI
Shiokawa	4449	17712	10	Mouse	NCBI
Sun	6361	26552	7	Mouse	NCBI
Human kidney	8814	20007	20	Human	NCBI
Mouse heart	7713	16354	13	Mouse	10x Genomics
4K PBMCs1	3996	16889	7	Human	10x Genomics
4K PBMCs2	4352	15402	9	Human	10x Genomics
Mouse kidney	1385	17073	9	Mouse	10x Genomics
5K PBMCs	5025	33538	8	Human	10x Genomics
Human embryos	1529	20667	5	Mouse	EMBL_EBI
Mouse lymph node	1489	23323	12	Mouse	EMBL_EBI

coder output $\mathbf{Z}$ is linearly projected to the classifier, generating class probabilities $\hat{\mathbf{Y}} = \{\hat{y}_1, \cdots, \hat{y}_N\}^\top$:

$$\hat{\mathbf{Y}} = \text{softmax}\left(\mathbf{Z}\mathbf{W}_C + b_C\right) \tag{14}$$

Where $\mathbf{W}_C \in \mathbb{R}^{d_{\text{model}} \times C}$ is the classification weight matrix, b_C is the bias term, and C is the number of classes. The *softmax* layer normalizes the output of the linear transformation to a probability distribution, yielding the probability that each sample belongs to each class. The model optimizes the prediction results by minimizing the cross-entropy loss between the predictions and true labels. The loss function is defined as:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{N_K} y_i \log \hat{y}_i \tag{15}$$

Where y_i is the true label of sample i, $\hat{y}_i$ is the predicted probability, and N_K is the number of labeled samples. The cross-entropy loss ensures that the model fits the labeled data accurately, while maximizing prediction confidence to improve generalization ability.

After pre-training, the complete basis matrix $\widehat{\mathbf{H}}^*$ with reconstructed embeddings is input into the Transformer model. Consistent feature embeddings are then projected to the classifier for prediction, ensuring the model's robustness to new data. Specifically, for each cell sample $x_i \in \widehat{\mathbf{H}}^*$, the model generates a class probability distribution through the softmax layer, and the final predicted class label y_i' is determined by selecting the maximum value from the argmax output:

$$y_i' = \arg\max(\hat{y}_i) \tag{16}$$

3. Experimental results

3.1. Dataset

To evaluate the performance of our model, we utilized sixteen datasets with ground truth labels for clustering analysis, pseudotime analysis, differential gene expression analysis, and other downstream analyses. These datasets were sourced from various platforms, including 10x Genomics, the National Center for Biotechnology Information (NCBI), and the European Molecular Biology Laboratory - European Bioinformatics Institute (EMBL-EBI). Additionally, these datasets encompass different species, cell types, and dynamic processes of cell development. Detailed information related to the datasets is provided in Table 1.

The scRNA-seq data used in this study were obtained from public data platforms. The dataset Baise [39], Klein [40], Zeisel [41], Muraro [42], Baron [43], Shiokawa [44], Sun [45], Human kidney [46] comes from the Gene Expression Omnibus (GEO) databases, with accession numbers GSE57249, GSE65525, GSE60361, GSE74672, GSE84133, GSE213755, GSE246147, GSE114569.

The mouse heart sample, human PBMCs sample, and mouse kidney cell sample are from 10x Genomics. The accessible datasets include the following URLs:

Mouse Heart: https://www.10xgenomics.com/datasets/10-k-heart-cells-from-an-e-18-mouse-v-3-chemistry-3-standard-3-0-0

4K PBMCs1: https://www.10xgenomics.com/datasets/peripheral-blood-mononuclear-cells-pbm-cs-from-a-healthy-donor-chromium-connect-channel-1-3-1-standard-3-1-0

4K PBMCs2: https://www.10xgenomics.com/datasets/4-k-pbm-cs-from-a-healthy-donor-2-standard-1-3-0

Mouse Kidney: https://www.10xgenomics.com/datasets/1k-mouse-kidney-nuclei-isolated-with-chromium-nuclei-isolation-kit-3-1-standard

5K PBMCs: https://www.10xgenomics.com/datasets/5-k-peripheral-blood-mononuclear-cells-pbm-cs-from-a-healthy-donor-v-3-chemistry-3-1-standard-3-0-2

The Goolam [47], mouse lymph node cell sample [48] and human preimplantation embryo sample [49] were sourced from EMBL-EBI, with accession numbers E-MTAB-3321, E-MTAB-10434 and E-MTAB-3929, respectively.

3.2. Data preprocessing

In the data preprocessing phase of our experiments, we applied normalization, logarithmic transformation, and high-variance gene selection to enhance the quality of the single-cell sequencing data. First, we standardized the gene expression levels of each cell using Total Count Normalization. Specifically, we employed the *normalize_total* function in the Scanpy library to adjust the RNA counts of each cell to a uniform total count level [50]. This step mitigates intercellular differences due to varying sequencing depths, ensuring comparability of gene expression levels across cells.

Next, we performed logarithmic transformation on the normalized data using the *log1p* function in Scanpy [51]. This log transformation, defined as $\log(x + 1)$, effectively handles zero and low-expression values, reducing the dynamic range of the data. This adjustment brings the distribution of gene expression levels closer to a normal distribution, thereby enhancing the statistical properties and stability of the data for analysis.

Finally, to focus on biologically relevant variability, we calculated the expression variance of each gene and selected the top 2,000 genes with the highest variance as high-variance genes (HVGs) [52]. These HVGs capture the variability across different cellular states and serve as a crucial basis for downstream clustering analysis. Through these preprocessing steps, we effectively improved data quality, strengthening both the reliability of clustering results and their biological interpretability.

3.3. Results on real single-cell datasets

In this section, we validate the proposed scCCTR model on fourteen real-world datasets and compare its performance with eight state-of-the-art clustering methods, demonstrating its effectiveness.

3.3.1. Analysis of clustering results

We evaluated the performance of the proposed scCCTR method on fourteen real-world single-cell datasets (listed in Table 1) and systematically compared it with eight widely used single-cell clustering algorithms: Scanpy [53], SC3 [22], Seurat [50], SIMLR [21], scGNN [27], scDeepCluster [26], scDSC [30], and cellVGAE [28]. To objectively assess the clustering performance, we employed six metrics: Adjusted Rand Index (ARI) [54], Normalized Mutual Information (NMI) [55], Accuracy (ACC) [56], Fowlkes-Mallows Index (FMI) [57], Davies-Bouldin Index (DBI) [58], and Graph cLISI [59]. These metrics comprehensively

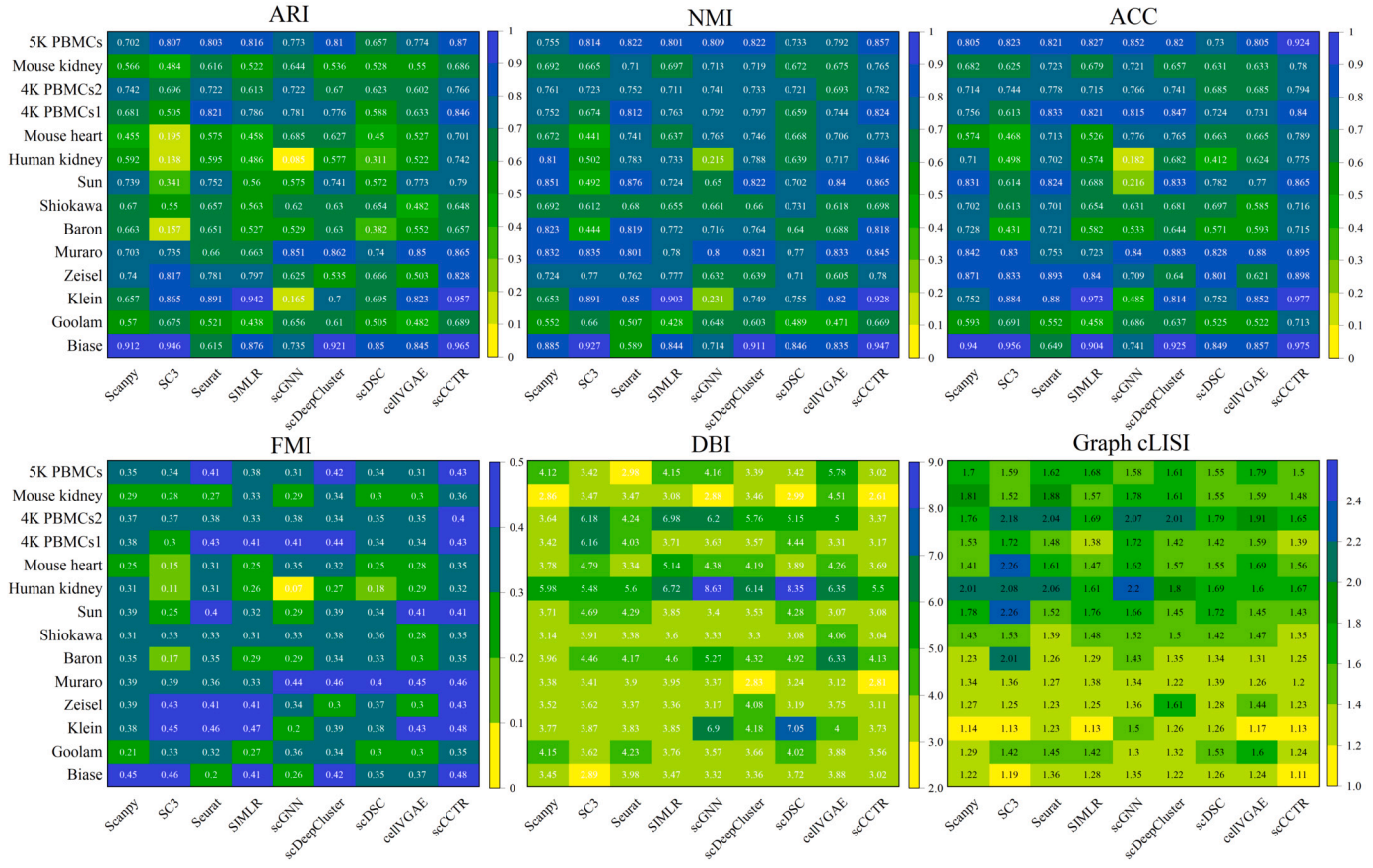


Fig. 2. The ARI, NMI, ACC, FMI, DBI and Gragh cLISI scores of scCCTR and the other 8 clustering algorithms on 12 real-world datasets.

evaluate clustering results in terms of label consistency, cluster separation quality, and local cell-type homogeneity within graph structures. All comparative methods were implemented using their original publicly available codes. To ensure fairness and reproducibility, we applied the optimal parameter configurations for each method on each dataset and reported their best performance.

Fig. 2 illustrates the clustering performance of scCCTR and baseline methods across the twelve datasets. The results demonstrate that scCCTR either outperforms or matches all other methods in all evaluation metrics. Specifically, ARI, NMI, ACC, and FMI access the consistency between clustering labels and ground-truth annotations. On average, scCCTR achieved ARI = 0.786, NMI = 0.814, ACC = 0.832, and FMI = 0.4, surpassing the average scores of the eight baseline methods by 15.2%, 10.0%, 12.3%, and 6.4%, respectively. Notably, scCCTR exhibited superior performance to mainstream algorithms such as SC3 and SIMLR. For cluster compactness and separation, scCCTR achieved significantly lower DBI scores (indicating better clustering quality) than other methods on most datasets. Furthermore, in terms of Graph cLISI, which quantifies local cell-type consistency in graph neighborhoods, scCCTR attained the highest scores on ten datasets and performed comparably to other methods on the remaining four.

The scCCTR method consistently demonstrated strong advantages across both small-scale and highly heterogeneous datasets. These results highlight its robustness and effectiveness in handling single-cell data characterized by high dimensionality, sparsity, and technical noise.

3.3.2. *Visualization of clustering results*

In this study, we employed t-Distributed Stochastic Neighbor Embedding (t-SNE) [60] to visualize the intrinsic structure of high-dimensional single-cell RNA sequencing (scRNA-seq) data. To evaluate the clustering performance of scCCTR in biologically complex scenarios, we selected two representative datasets: the Mouse kidney dataset (characterized by high cellular heterogeneity) and the 4K PBMCs1 dataset (a smaller-scale peripheral blood mononuclear cell dataset). These datasets exemplify clustering challenges across distinct biological systems.

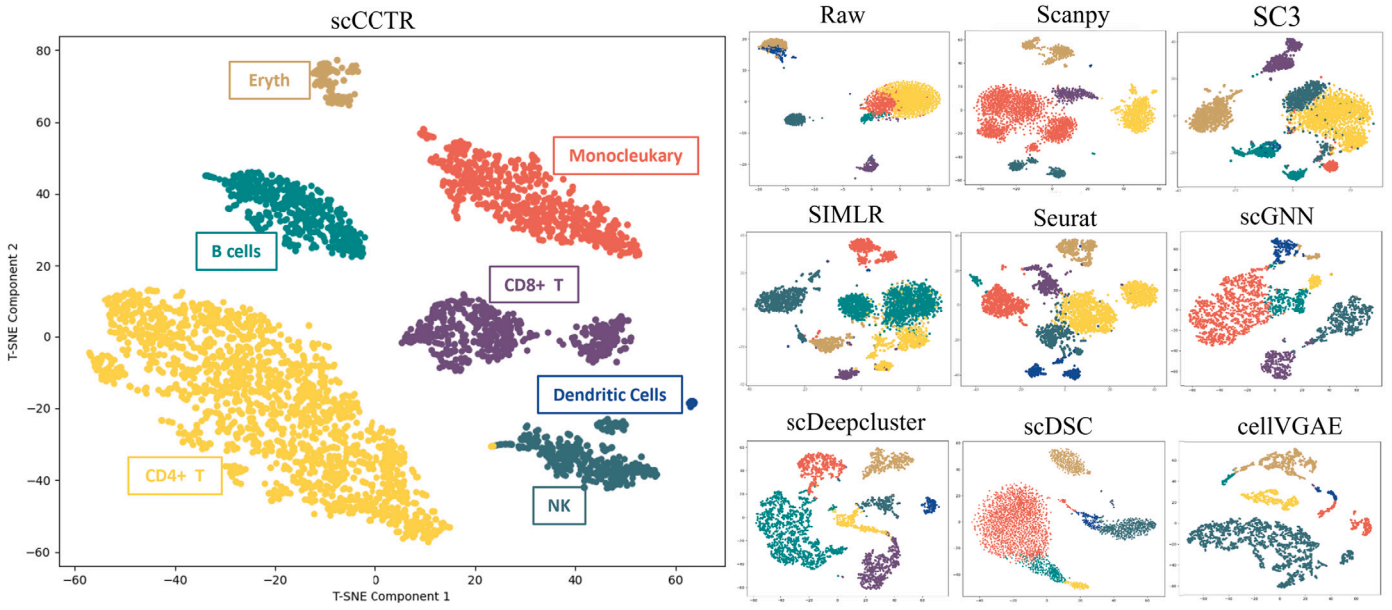
We conducted a comparative visualization analysis between scCCTR and eight baseline methods on these two datasets. For the 4K PBMCs1 dataset, the visualization results are shown in Fig. 3(a). Methods like scCCTR, cellVGAE, and scDeepCluster successfully identified seven cell clusters with high classification accuracy and well-defined inter-cluster boundaries. In contrast, the remaining six methods exhibited overlapping clusters and insufficient separation, complicating the discrimination of distinct cell populations.

On the Mouse kidney dataset, as shown in Fig. 3(b), scCCTR demonstrated clear separation of nine cell clusters with minimal overlap. However, Seurat and Scanpy produced ambiguous cluster boundaries, while SC3 and scGNN struggled to maintain intra-cluster cohesion, as evidenced by scattered cell distributions within clusters.

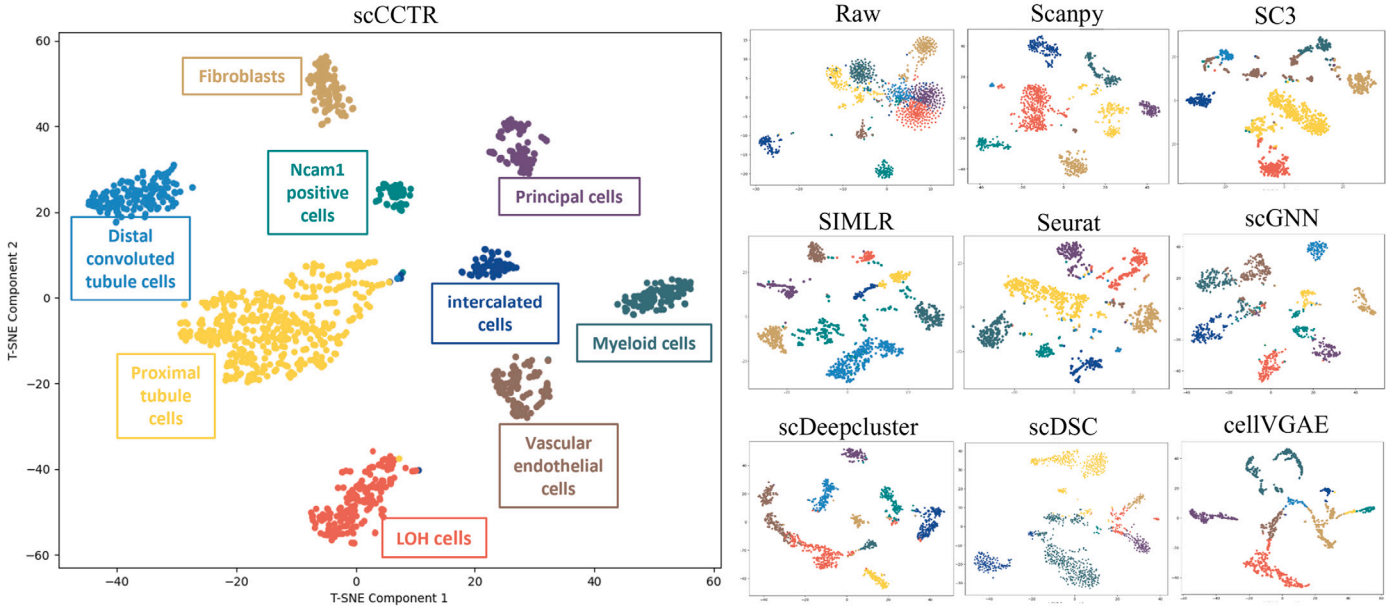
Notably, scCCTR exhibited outstanding and consistent performance across both datasets. Whether applied to the smaller 4K PBMCs1 dataset or the more heterogeneous and complex Mouse kidney dataset, scCCTR maintained superior clustering precision and visualization clarity. These findings further validate scCCTR's broad applicability and robustness in single-cell clustering tasks, particularly for datasets with high dimensionality, sparsity, and biological variability.

3.4. *Parameter analysis*

The scCCTR method optimizes core feature representation by iteratively selecting and reconstructing key data. In this section, we systematically evaluated two critical parameters—reconstruction ratio (α) and iteration count (β)—using the 5K PBMCs dataset. This experiment focused on analyzing the effect of varying α on clustering accuracy (ARI, NMI, and ACC) while incrementally increasing β to assess model stabil-



(a) 4K PBMCs1



(b) Mouse kidney

Fig. 3. The clustering visualization results of scCCTR and eight other clustering algorithms. (a) t-SNE visualization of the 4K PBMCs1 dataset. (b) t-SNE visualization of the Mouse kidney dataset. Each point represents a cell, and different colors indicate different cell clusters.

ity. All results were averaged over multiple runs to ensure robustness and reliability.

As shown in Fig. 4, clustering accuracy metrics (ARI, NMI, and ACC) exhibited a pronounced increase as α rose from 0.1 to 0.5, with the most notable growth occurring in the range of 0.2 to 0.4. When α further increased from 0.6 to 1.0, the results indicated that the improvement in clustering performance becomes gradual, with diminishing gains in some metrics. Particularly, as α approached 1.0, the reconstructed features stabilized, indicating saturation in noise reduction and information retention.

These results suggest that lower reconstruction ratios ($\alpha < 0.5$) enhance the model's ability to capture discriminative cellular features by selectively reconstructing data near core clusters, effectively filtering technical noise while preserving biologically critical patterns. Conversely, higher reconstruction ratios ($\alpha > 0.6$) introduce redundant information and noise, reducing the marginal utility of additional re-

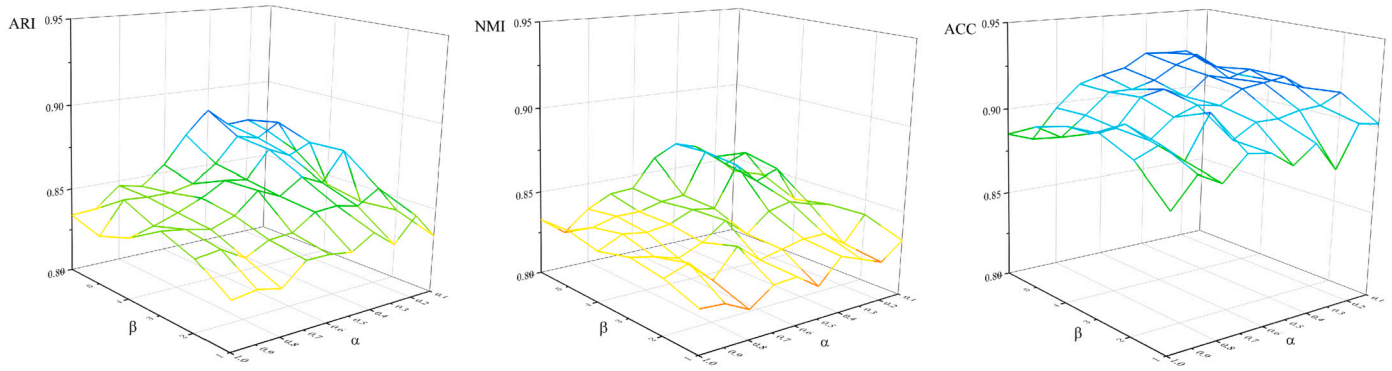


Fig. 4. Analysis of parameters α and β using the 5K PBMCs dataset. α ranges from 0.1 to 1.0, with β set from 1 to 6.

construction. Furthermore, a controlled increase in iteration count (β) improved clustering performance within a bounded range, though excessive iterations risked overfitting without commensurate gains.

Thus, for complex single-cell RNA sequencing data, optimizing α (ideally within 0.2–0.5) and carefully tuning β are essential to balance computational efficiency and clustering efficacy.

3.5. Downstream analysis

In addition to clustering analysis, single-cell sequencing data is widely applied in various downstream analyses such as developmental trajectory analysis, pseudotime analysis, differential gene expression analysis and functional enrichment analysis, aiding in the exploration of complex biological processes. These downstream analyses serve as powerful tools for deciphering cellular heterogeneity and understanding complex biological systems.

In this study, we employed the Monocle [61] algorithm to perform pseudotime analysis on the Human embryos dataset. This dataset comprises 88 samples from human pre-implantation embryos (days 3 to 7), containing 1,529 single-cell RNA sequencing profiles. It provides crucial biological information for understanding cellular differentiation during early embryonic development.

To assess the effectiveness of scCCTR in dimensionality reduction and core feature reconstruction, we conducted pseudotime analysis on both the original data and data augmented by scCCTR, NMF, and KNN methods to compare their performance in cell developmental trajectory inference. The experimental results, as shown in Fig. 5, indicate that the augmented data more accurately captures the continuous developmental process of cells, with smoother transitions along the pseudotime axis compared to the original data. Moreover, the developmental trajectories reconstructed by scCCTR exhibit high consistency with KNN and NMF methods, outperforming better or equivalently to other methods in arranging cells at different developmental stages. These findings suggest that scCCTR's data reconstruction capability plays an active role in gene expression data analysis. The scCCTR model not only excels in noise reduction and data reconstruction but also retains cellular developmental features, effectively reducing errors caused by the sparsity and noise of the original data.

In addition, we conducted further biological information analysis using the Mouse lymph node dataset, which includes 753 inguinal lymph node cells from adult mice exposed to extracellular vesicles (EVs) derived from melanoma, along with 736 unaffected cells from the same site. This dataset provides a valuable resource for studying the impact of melanoma-derived EVs on lymph node cells, which consist primarily of lymph node endothelial cells, macrophages, T cells, NK cells, and other immune cells.

In our experiments, we used Monocle3 [62] for trajectory inference to evaluate the model's data reconstruction and dimensionality reduction capabilities. As shown in Fig. 6(a), types 1 and 2 represent EV-exposed and normal LECs, respectively, while types 3 and 4 represent EV-exposed and normal lymph node immune cells. Compared to the original data, we identified three LEC subtypes (ACKR4$^+$ cLECs, MADCAM1$^+$ fLECs, and Mrc1$^+$ mLECs) using subtype markers from Noriki Fujimoto's study [63]. This indicates that scCCTR significantly enhances high-quality data reconstruction.

Furthermore, during trajectory inference of LECs, we observed a clear developmental trajectory distinction between EV-exposed lymph node endothelial cells (EV_LN_LECs) and normal cells. Therefore, we performed further pseudotime analysis on the reconstructed data using Monocle. Fig. 6(b) shows that, compared to normal cells, EV_LN_LECs exposed to melanoma EVs exhibit delayed developmental and differentiation processes. This finding suggests that melanoma-derived EVs may play a critical role in the early stages of tumor invasion and metastasis by influencing LEC development and function.

To explore the biological effects of melanoma EVs, we used scCCTR-reconstructed gene expression data to generate volcano plots for analyzing differential gene expression in LECs. The x-axis represents log2 fold changes (avg_log2FC) to display relative expression changes between the two groups, and the y-axis represents the negative log10 of the adjusted p-value (-log10), indicating statistical significance of differential expression. As shown in Fig. 6(c), red points on the left represent significantly differentially expressed genes in Naive_LN_LECs (p-val_adj < 0.05 and $|\log 2\text{FC}| > 1$), while red points on the right indicate significantly differentially expressed genes in EV_LN_LECs. Gray points denote non-significant genes. To interpret the function of differentially expressed genes, we performed functional enrichment analysis for downregulated and upregulated genes, listing the top 10 enriched GO terms for each.

As illustrated in Fig. 6(d), downregulated genes are primarily enriched in GO terms related to protein translation and synthesis, as well as tissue and organ development and differentiation. These downregulated GO terms suggest that EV_LN_LECs may experience developmental inhibition, affecting their normal tissue formation and functional differentiation. Upregulated genes are mainly associated with cell proliferation, angiogenesis, and morphogenesis, indicating that EV_LN_LECs may possess enhanced supportive functions in tumor growth, facilitating conditions for tumor expansion and metastasis.

In summary, both pseudotime analysis and gene enrichment analysis reveal adaptive changes in LECs influenced by EVs. While reducing their own metabolism and proliferation, they enhance supportive functions, such as angiogenesis and lymphangiogenesis, providing structural support within the tumor microenvironment, which may facilitate tumor progression.

3.6. Stability and efficiency analysis

In single-cell RNA sequencing, dropout events, which are a consequence of technical limitations, often result in a significant number of expression values being recorded as zero, obscuring the true gene expression and introducing noise. To assess the robustness of the model

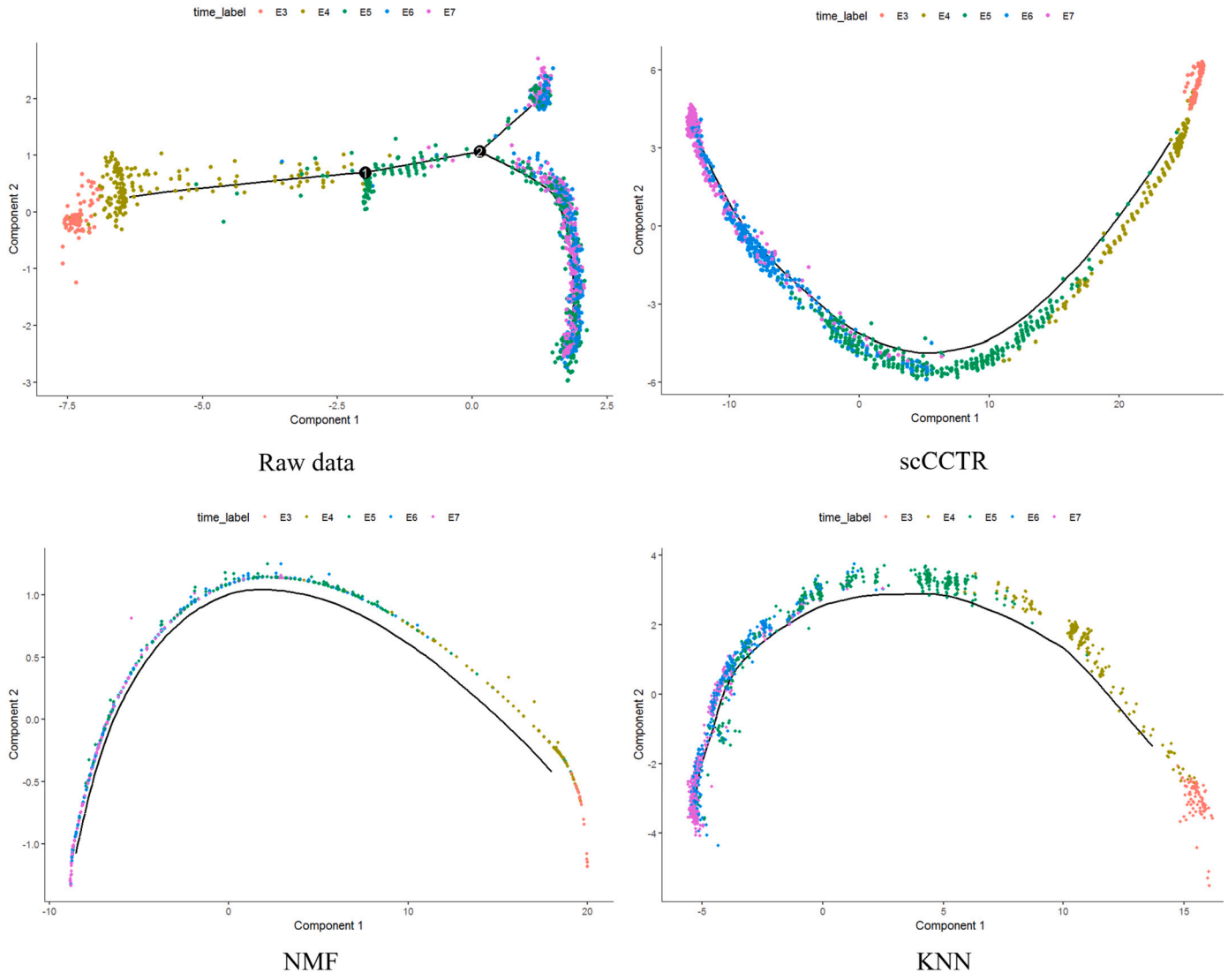


Fig. 5. Pseudotime series analysis of original data and data augmented by scCCTR, NMF, and KNN.

against dropout issues, we evaluated the performance of scCCTR under varying dropout rates and compared it with several other single-cell clustering methods. Using the Mouse heart dataset, we evaluated stability based on ARI and NMI. In this experiment, we adjusted the parameters of various methods across multiple independent trials. For scCCTR, we tuned multiple parameters in the experiments, including the dimensionality reduction, the number of iterations, and the number of attention heads. Boxplot metrics, including upper and lower bounds, interquartile range (IQR), median, and mean, were used to quantify stability across replicates.

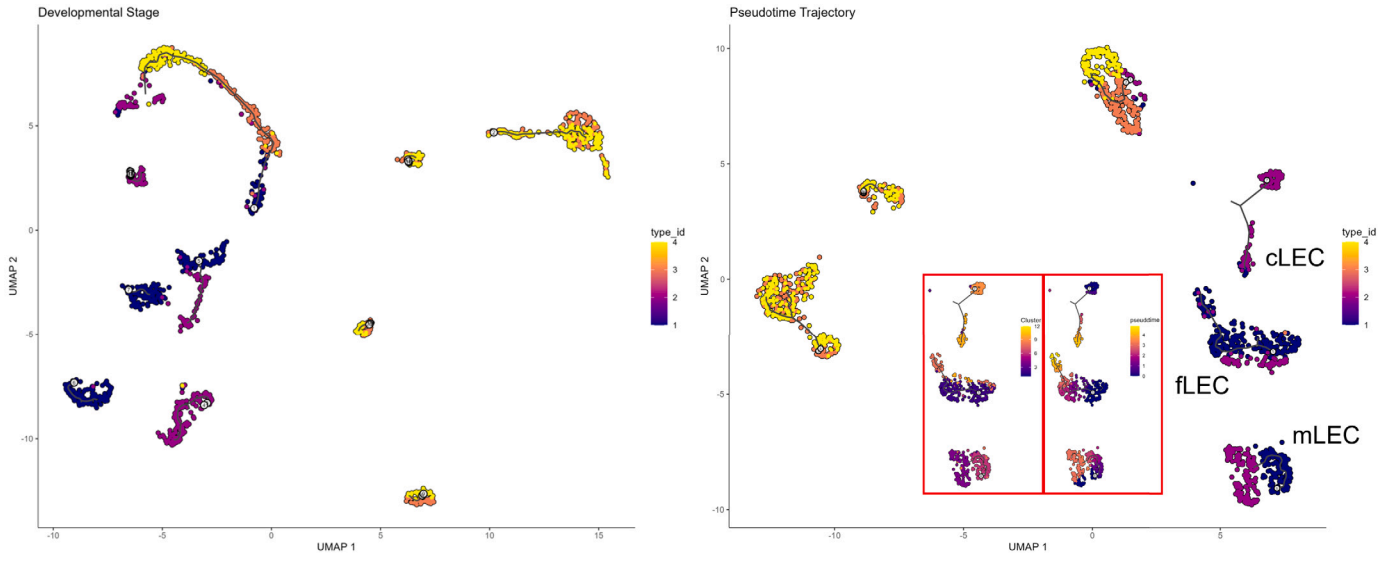
As shown in Fig. 7(a), scCCTR exhibits superior stability at dropout rates of 0.1, 0.3, and 0.5. Compared to methods such as Seurat, scDSC, and cellVGAE, scCCTR has a narrower boxplot for the ARI metric, with a median close to 0.7 and smaller upper and lower bounds, indicating higher consistency of clustering results across different experimental replicates. For the NMI metric, scCCTR also demonstrates the same advantage, with a higher median and smaller interquartile range, showing significant stability compared to other methods.

The experimental results indicate that scCCTR not only achieves high clustering accuracy on single-cell data but also effectively addresses dropout issues in single-cell sequencing, providing robust technical support for the analysis of complex data in single-cell transcriptomics studies.

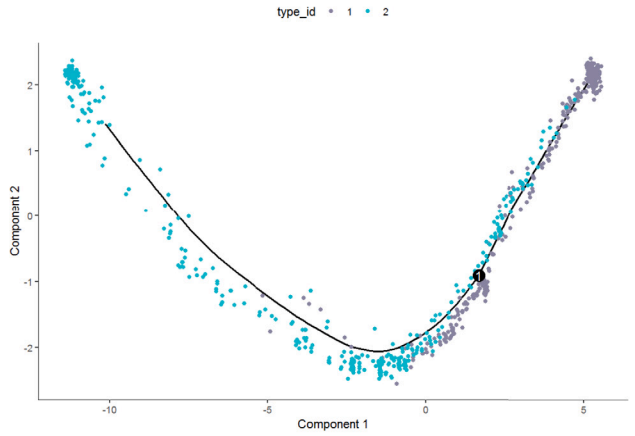
To systematically evaluate the computational efficiency of scCCTR against eight benchmark methods, we generated test datasets ranging from 2,000 to 14,000 cells using the Baron dataset, with gene features fixed at 20,000. All experiments were executed on a CPU under single-threaded conditions, and total runtime was measured as the sum of preprocessing, model training, and clustering phases.

As illustrated in Fig. 7(b), community detection-based methods (Scanpy and Seurat) exhibited the shortest runtimes, maintaining efficient performance even at 14,000 cells (10^2–10^3 s). While scCCTR showed higher runtime than scDeepCluster and scDSC, it significantly outperformed SC3, SIMLR, scGNN, and cellVGAE. Notably, SIMLR failed to complete at 14,000 cells due to excessive memory consumption caused by its multi-kernel learning framework.

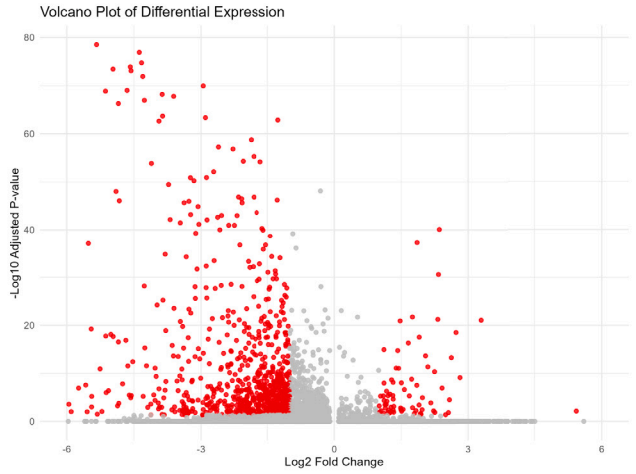
The iterative consensus mechanism of scCCTR contributed to increased training time compared to lightweight methods. However, its computational overhead remains manageable through two key optimizations: (1) aggressive feature selection during preprocessing to reduce dimensionality, and (2) GPU acceleration during training, which we demonstrate improves scalability for large-scale datasets.



(a) Developmental trajectory inference



(b) Pseudotime analysis



(c) Differential gene expression analysis

Fig. 6. (a) Developmental trajectory inference using original and reconstructed data on the Mouse lymph node dataset. (b) Pseudotime analysis of Naive LN_LEC and EV_LN_LEC. (c) Differential gene expression analysis between Naive LN_LEC and EV_LN_LEC. (d) Downregulated and upregulated GO enrichment analysis for Naive LN_LEC and EV_LN_LEC.

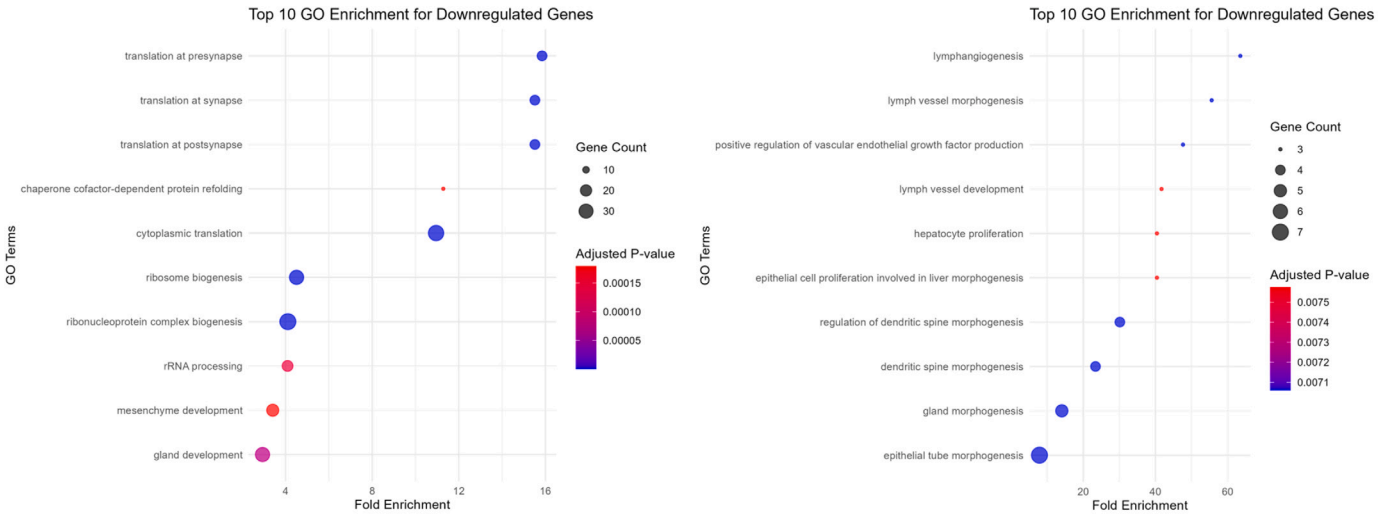
3.7. Ablation study

In this study, we conducted a series of ablation experiments to systematically evaluate the impact of key modules in the scCCTR model—consensus constraints, cyclic reconstruction, and the Transformer module–on clustering performance. By removing these modules and comparing their performance with the complete model across multiple datasets, we analyzed their contributions to model stability and clustering accuracy.

First, to assess the role of the consensus constraint module, we performed five independent experiments on the 4K PBMCs2 dataset and compared the results with those of the complete model. As shown in Fig. 8(a), removing the consensus constraint led to significant fluctuations during cyclic reconstruction, resulting in unstable clustering accuracy. In contrast, the complete model, supported by the consensus constraint module, exhibited higher stability and clustering accuracy. This demonstrates that the consensus constraint plays a crucial role in integrating clustering results across iterations, suppressing noise interference, and avoiding local optima, thereby significantly improving model stability and global optimization capabilities.

Next, we conducted ablation experiments on the cyclic reconstruction and Transformer modules across six datasets, using ARI as the evaluation metric. The results, shown in Fig. 8(b), compare three groups: NC, NT, and FULL. Removing the cyclic reconstruction module resulted in a significant decline in clustering performance. This indicates that the cyclic reconstruction module plays a crucial role in iteratively refining high-confidence data, enabling the model to capture core structural features across multiple iterations. Through this module, the model gradually enhances its ability to recognize cellular heterogeneity and subtle differences, while avoiding the information loss associated with directly relying on autoencoder outputs.

The Transformer module is designed to capture global dependencies in high-dimensional data, thereby enhancing the model's adaptability to complex single-cell datasets. Removing the Transformer module weakened the model's ability to learn the internal structure of datasets and limited its capacity to utilize core cells for global feature learning, ulti-



(d) GO enrichment analysis

Fig. 6. (*continued*)

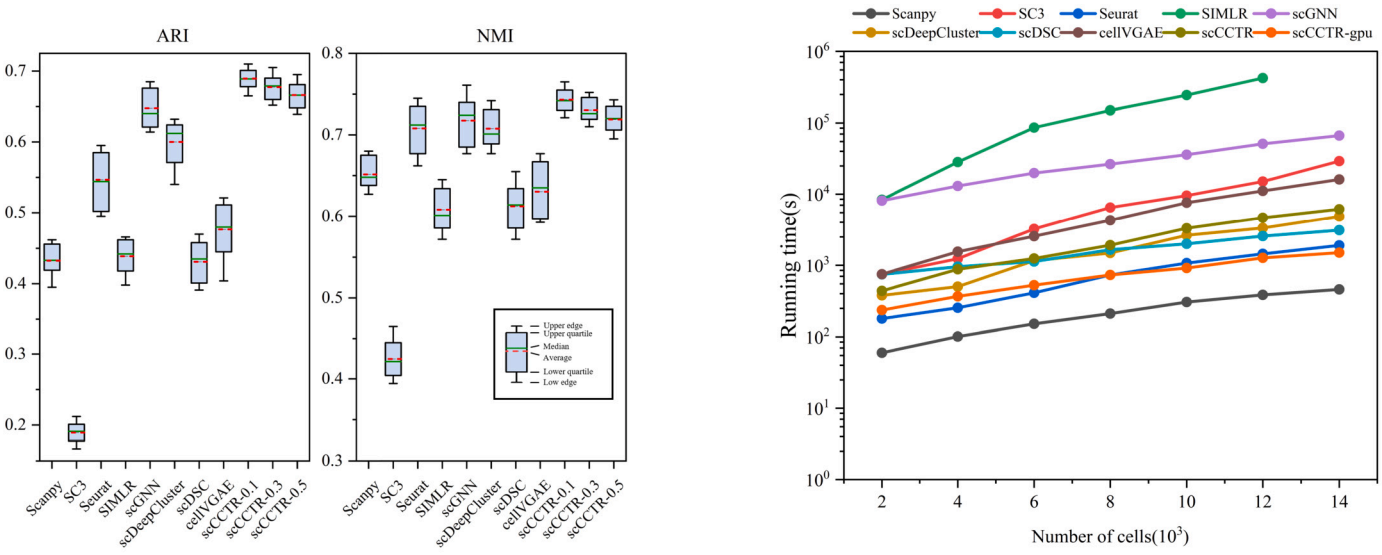


Fig. 7. Ablation Study (a) For dropout rates of 0.1, 0.3, and 0.5, the stability analysis results of scCCTR compared with different methods. (b) Comparison of scCCTR with each benchmark method in terms of cell count and runtime, as well as the runtime of scCCTR after applying GPU.

mately resulting in reduced clustering accuracy. These results validate the importance of the Transformer module in improving global clustering performance.

4. Conclusion

In recent years, the rapid advancement of single-cell sequencing technology has enabled researchers to dissect cellular heterogeneity in complex biological systems with unprecedented precision. However, the inherent characteristics of single-cell sequencing data, including high dimensionality, sparsity, and significant heterogeneity, present considerable challenges for dimensionality reduction, clustering analysis, and trajectory reconstruction.

This study introduces scCCTR, a semi-supervised model based on iterative selection of high-confidence data. The model extracts stable representative features through iterative refinement and employs a consensus strategy to eliminate unstable solutions during the process. By leveraging iteratively selected core data as a substitute for external prior information, scCCTR overcomes the limitations of traditional semi-supervised learning, which often relies on hard-to-acquire prior knowledge. Comprehensive evaluations on eight real-world datasets demonstrate the significant advantages of scCCTR in handling high-dimensional, sparse, and heterogeneous scRNA-seq data. Compared to state-of-the-art methods, scCCTR achieves higher accuracy in clustering analysis and successfully reconstructs developmental trajectories in pseudotime analysis, highlighting its broad applicability and strong potential in single-cell studies.

Despite its effectiveness and robustness in clustering analysis, scCCTR relies on the quality of core data for optimal performance. Future work will focus on integrating multi-modal single-cell data and developing a more flexible analytical framework to accommodate diverse experimental conditions and data characteristics. This will further enhance scCCTR's adaptability and scalability, providing robust support for understanding complex biological systems and advancing single-cell research.

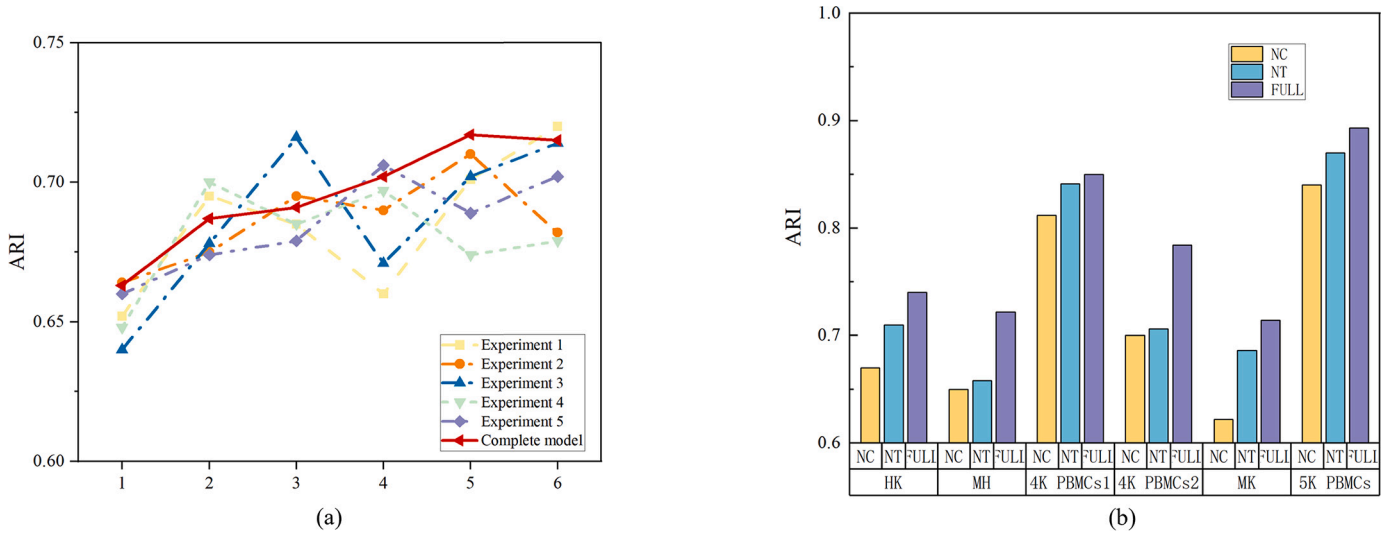


Fig. 8. Ablation Study (a) Ablation analysis of the consensus constraint module. Experiment 1-5 represent the trajectories without the constraint conditions, and the "Complete model" shows the trajectory of the full model. (b) NC shows the ablation results of the iterative reconstruction module, and NT shows the ablation results of the Transformer module.

CRediT authorship contribution statement

Jie Chen: Writing – original draft, Formal analysis, Conceptualization. **Qiucheng Sun:** Writing – review & editing, Project administration, Methodology, Funding acquisition. **Chunyan Wang:** Software, Resources. **Changbo Gao:** Visualization, Validation, Formal analysis.

Declaration of competing interest

The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

[1] Macosko EZ, Basu A, Satija R, Nemesh J, Shekhar K, Goldman M, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell 2015;161(5):1202–14.
[2] Zheng GX, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun 2017;8(1):14049.
[3] Zhao J, Jaffe A, Li H, Lindenbaum O, Sefik E, Jackson R, et al. Detection of differentially abundant cell subpopulations in scrna-seq data. Proc Natl Acad Sci 2021;118(22):e2100293118.
[4] Liu C, Zhang Y, Gao X, Wang G. Identification of cell subpopulations associated with disease phenotypes from scrna-seq data using pacsi. BMC Biol 2023;21(1):159.
[5] Ruan Z, Cao G, Qian Y, Fu L, Hu J, Xu T, et al. Single-cell rna sequencing unveils lrg1's role in cerebral ischemia–reperfusion injury by modulating various cells. J Neuroinflamm 2023;20(1):285.
[6] Qiu X, Mao Q, Tang Y, Wang L, Chawla R, Pliner HA, et al. Reversed graph embedding resolves complex single-cell trajectories. Nat Methods 2017;14(10):979–82.
[7] Tirosh I, Izar B, Prakadan SM, Wadsworth MH, Treacy D, Trombetta JJ, et al. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq. Science 2016;352(6282):189–96.
[8] Jaitin DA, Kenigsberg E, Keren-Shaul H, Elefant N, Paul F, Zaretsky I, et al. Massively parallel single-cell rna-seq for marker-free decomposition of tissues into cell types. Science 2014;343(6172):776–9.
[9] Huang J, Liu L, Qin L, Huang H, Li X. Single-cell transcriptomics uncovers cellular heterogeneity, mechanisms, and therapeutic targets for Parkinson's disease. Front Genet 2022;13:686739.
[10] Lun AT, McCarthy DJ, Marioni JC. A step-by-step workflow for low-level analysis of single-cell rna-seq data with bioconductor. F1000Res 2016;5.
[11] Bouland GA, Mahfouz A, Reinders MJ. Consequences and opportunities arising due to sparser single-cell rna-seq datasets. Genome Biol 2023;24(1):86.
[12] Reid AJ, Talman AM, Bennett HM, Gomes AR, Sanders MJ, Illingworth CJ, et al. Single-cell rna-seq reveals hidden transcriptional variation in malaria parasites. eLife 2018;7:e33105.
[13] Zou Z, Hua K, Zhang X. Hgc: fast hierarchical clustering for large-scale single-cell data. Bioinformatics 2021;37(21):3964–5.
[14] Tangherloni A, Ricciuti F, Besozzi D, Liò P, Cvejic A. Analysis of single-cell rna sequencing data based on autoencoders. BMC Bioinform 2021;22(1):309.
[15] Hicks SC, Townes FW, Teng M, Irizarry RA. Missing data and technical variability in single-cell rna-sequencing experiments. Biostatistics 2018;19(4):562–78.
[16] Habib N, Avraham-Davidi I, Basu A, Burks T, Shekhar K, Hofree M, et al. Massively parallel single-nucleus rna-seq with dronc-seq. Nat Methods 2017;14(10):955–8.
[17] Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008;2008(10):P10008.
[18] Traag VA, Waltman L, Van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. Sci Rep 2019;9(1):1–12.
[19] Žurauskienė J, Yau C. pcareduce: hierarchical clustering of single cell transcriptional profiles. BMC Bioinform 2016;17:1–11.
[20] Lin P, Troup M, Ho JW. Cidr: ultrafast and accurate clustering through imputation for single-cell rna-seq data. Genome Biol 2017;18:1–11.
[21] Wang B, Zhu J, Pierson E, Ramazzotti D, Batzoglou S. Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning. Nat Methods 2017;14(4):414–6.
[22] Kiselev VY, Kirschner K, Schaub MT, Andrews T, Yiu A, Chandra T, et al. Sc3: consensus clustering of single-cell rna-seq data. Nat Methods 2017;14(5):483–6.
[23] Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ. Single-cell rna-seq denoising using a deep count autoencoder. Nat Commun 2019;10(1):390.
[24] Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. Nat Methods 2018;15(12):1053–8.
[25] Wang D, Gu J. Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. Genomics Proteomics Bioinform 2018;16(5):320–31.
[26] Tian T, Wan J, Song Q, Wei Z. Clustering single-cell rna-seq data with a model-based deep learning approach. Nat Mach Intell 2019;1(4):191–8.
[27] Wang J, Ma A, Chang Y, Gong J, Jiang Y, Qi R, et al. scgnn is a novel graph neural network framework for single-cell rna-seq analyses. Nat Commun 2021;12(1):1882.
[28] Buterez D, Bica I, Tariq I, Andrés-Terré H, Liò P. Cellvgae: an unsupervised scrna-seq analysis workflow with graph attention networks. Bioinformatics 2022;38(5):1277–86.
[29] Svensson V, Vento-Tormo R, Teichmann SA. Exponential scaling of single-cell rna-seq in the past decade. Nat Protoc 2018;13(4):599–604.
[30] Gan Y, Huang X, Zou G, Zhou S, Guan J. Deep structural clustering for single-cell rna-seq data jointly through autoencoder and graph neural network. Brief Bioinform 2022;23(2):bbac018.
[31] Xiong Z, Luo J, Shi W, Liu Y, Xu Z, Wang B. Scgcl: an imputation method for scrna-seq data based on graph contrastive learning. Bioinformatics 2023;39(3):btad098.
[32] Xu C, Lopez R, Mehlman E, Regier J, Jordan MI, Yosef N. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. Mol Syst Biol 2021;17(1):e9620.
[33] Berthelot D, Carlini N, Goodfellow I, Papernot N, Oliver A, Raffel CA. Mixmatch: a holistic approach to semi-supervised learning. Adv Neural Inf Process Syst 2019;32.
[34] Duren Z, Chen X, Zamanighomi M, Zeng W, Satpathy AT, Chang HY, et al. Integrative analysis of single-cell genomics data by coupled nonnegative matrix factorizations. Proc Natl Acad Sci 2018;115(30):7723–8.
[35] Ren Y, Hu K, Dai X, Pan L, Hoi SC, Xu Z. Semi-supervised deep embedded clustering. Neurocomputing 2019;325:121–30.
[36] Vaswani A. Attention is all you need. Adv Neural Inf Process Syst 2017.

[37] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 1026–34.
[38] Pollen AA, Nowakowski TJ, Shuga J, Wang X, Leyrat AA, Lui JH, et al. Low-coverage single-cell mrna sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. Nat Biotechnol 2014;32(10):1053–8.
[39] Biase FH, Cao X, Zhong S. Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell rna sequencing. Genome Res 2014;24(11):1787–96.
[40] Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, Li V, et al. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell 2015;161(5):1187–201.
[41] Zeisel A, Muñoz-Manchado AB, Codeluppi S, Lönnerberg P, La Manno G, Juréus A, et al. Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. Science 2015;347(6226):1138–42.
[42] Muraro MJ, Dharmadhikari G, Grün D, Groen N, Dielen T, Jansen E, et al. A single-cell transcriptome atlas of the human pancreas. Cell Syst 2016;3(4):385–94.
[43] Baron M, Veres A, Wolock SL, Faust AL, Gaujoux R, Vetere A, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. Cell Syst 2016;3(4):346–60.
[44] Shiokawa D, Sakai H, Koizumi M, Okimoto Y, Mori Y, Kanda Y, et al. Elevated stress response marks deeply quiescent reserve cells of gastric chief cells. Commun Biol 2023;6(1):1183.
[45] Sun W, Liu Z, Jiang X, Chen MB, Dong H, Liu J, et al. Spatial transcriptomics reveal neuron–astrocyte synergy in long-term memory. Nature 2024;627(8003):374–81.
[46] Kim AD, Lake BB, Chen S, Wu Y, Guo J, Parvez RK, et al. Cellular recruitment by podocyte-derived pro-migratory factors in assembly of the human renal filter. iScience 2019;20:402–14.
[47] Goolam M, Scialdone A, Graham SJ, Macaulay IC, Jedrusik A, Hupalowska A, et al. Heterogeneity in oct4 and sox2 targets biases cell fate in 4-cell mouse embryos. Cell 2016;165(1):61–74.
[48] Leary N, Walser S, He Y, Cousin N, Pereira P, Gallo A, et al. Melanoma-derived extracellular vesicles mediate lymphatic remodelling and impair tumour immunity in draining lymph nodes. J Extracell Vesicl 2022;11(2):e12197.
[49] Petropoulos S, Edsgärd D, Reinius B, Deng Q, Panula SP, Codeluppi S, et al. Single-cell rna-seq reveals lineage and x chromosome dynamics in human preimplantation embryos. Cell 2016;165(4):1012–26.
[50] Lun ATL, Bach K, Marioni JC. Pooling across cells to normalize single-cell rna sequencing data with many zero counts. Genome Biol 2016;17:1–14.
[51] Satija R, Farrell JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol 2015;33(5):495–502.
[52] Brennecke P, Anders S, Kim JK, Kołodziejczyk AA, Zhang X, Proserpio V, et al. Accounting for technical noise in single-cell rna-seq experiments. Nat Methods 2013;10(11):1093–5.
[53] Wolf FA, Angerer P, Theis FJ. Scanpy: large-scale single-cell gene expression data analysis. Genome Biol 2018;19:1–5.
[54] Van der Maaten L, Hinton G. Visualizing data using t-sne. J Mach Learn Res 2008;9(11).
[55] Trapnell C, Cacchiarelli D, Grimsby J, Pokharel P, Li S, Morse M, et al. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nat Biotechnol 2014;32(4):381–6.
[56] Bendall SC, Davis KL, Amir E-aD, Tadmor MD, Simonds EF, Chen TJ, et al. Single-cell trajectory detection uncovers progression and regulatory coordination in human b cell development. Cell 2014;157(3):714–25.
[57] Li G, Fu S, Wang S, Zhu C, Duan B, Tang C, et al. A deep generative model for multi-view profiling of single-cell rna-seq and atac-seq data. Genome Biol 2022;23(1):20.
[58] Lim HS, Qiu P. Quantifying the clusterness and trajectoriness of single-cell rna-seq data. PLoS Comput Biol 2024;20(2):e1011866.
[59] Luecken MD, Büttner M, Chaichoompu K, Danese A, Interlandi M, Müller MF, et al. Benchmarking atlas-level data integration in single-cell genomics. Nat Methods 2022;19(1):41–50.
[60] Santos JM, Embrechts M. On the use of the adjusted rand index as a metric for evaluating supervised classification. In: International conference on artificial neural networks. Springer; 2009. p. 175–84.
[61] Strehl A, Ghosh J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J Mach Learn Res Dec. 2002;3:583–617.
[62] Huang L, Yan D, Taft N, Jordan M. Spectral clustering with perturbed data. Adv Neural Inf Process Syst 2008;21.
[63] Fujimoto N, He Y, D'Addio M, Tacconi C, Detmar M, Dieterich LC. Single-cell mapping reveals new markers and functions of lymphatic endothelial cells in lymph nodes. PLoS Biol 2020;18(4):e3000704.