scCDCG: Efficient Deep Structural Clustering for single-cell RNA-seq via Deep Cut-informed Graph Embedding

Ping Xu[1,2*], Zhiyuan Ning[1,2*], Meng Xiao[1,2], Guihai Feng[4,5,6], Xin Li[4,5,6], Yuanchun Zhou[1,2,3], and Pengfei Wang[1,2(✉)]

[1] Computer Network Information Center, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3] Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences
[4] State Key Laboratory of Stem Cell and Reproductive Biology, Institute of Zoology, Chinese Academy of Sciences
[5] Institute for Stem Cell and Regenerative Medicine, Chinese Academy of Science
[6] Beijing Institute for Stem Cell and Regenerative Medicine, Chinese Academy of Science

{xuping,ningzhiyuan,shadow,zyc}@cnic.cn, {fenggh,xinli}@ioz.ac.cn, pfwang@cnic.com

Abstract. Single-cell RNA sequencing (scRNA-seq) is essential for unraveling cellular heterogeneity and diversity, offering invaluable insights for bioinformatics advancements. Despite its potential, traditional clustering methods in scRNA-seq data analysis often neglect the structural information embedded in gene expression profiles, crucial for understanding cellular correlations and dependencies. Existing strategies, including graph neural networks, face challenges in handling the inefficiency due to scRNA-seq data's intrinsic high-dimension and high-sparsity. Addressing these limitations, we introduce scCDCG (**s**ingle-**c**ell RNA-seq **C**lustering via **D**eep **C**ut-informed **G**raph), a novel framework designed for efficient and accurate clustering of scRNA-seq data that simultaneously utilizes intercellular high-order structural information. scCDCG comprises three main components: (i) *A graph embedding module utilizing deep cut-informed techniques*, which effectively captures intercellular high-order structural information, overcoming the over-smoothing and inefficiency issues prevalent in prior graph neural network methods. (ii) *A self-supervised learning module guided by optimal transport*, tailored to accommodate the unique complexities of scRNA-seq data, specifically its high-dimension and high-sparsity. (iii) *An autoencoder-based feature learning module* that simplifies model complexity through effective dimension reduction and feature extraction. Our extensive experiments on 6 datasets demonstrate scCDCG's superior performance and efficiency compared to 7 established models, underscoring scCDCG's potential as a transformative tool in scRNA-seq data analysis. Our code is available at: https://github.com/XPgogogo/scCDCG.

Keywords: Bioinformatics · scRNA-seq data · Clustering · Deep cut-informed graph embedding · Self-supervised learning.

[*] These authors contributed equally to this work.

1 Introduction

Single-cell RNA sequencing (scRNA-seq) data analysis is a groundbreaking advancement in the field of bioinformatics that enables researchers to explore the complex structure and heterogeneity within cell populations and reveal different cell types, states, or subpopulations [18]. Clustering is an important tool for analyzing scRNA-seq data, which separates the cells into different groups with similar cells in the same group, thus can reveal the complex characteristics of different cell populations [12]. Clustering of scRNA-seq data has proven to be particularly valuable in elucidating cell development, understanding disease mechanisms, and identifying novel cell types and subtypes [8].

Early classical scRNA-seq clustering methods rely on hard clustering algorithms. For example, pcaReduce [33] combines two methods, PCA and K-means clustering, and iteratively merges clusters based on the associated probability density function; SUSSC [26] combines Uniform Manifold Approximation and Projection (UMAP) with K-means clustering to merge clusters. However, the inherent characteristics of scRNA-seq data include high-dimension, high-sparsity, noise, nonlinearity, and frequent dropout events, which make the relationships between variables complex and nonlinear [5]. This will bring a common limitation to these methods: they use hard clustering algorithms that are difficult to deal with high-dimension and capture complex nonlinear data features, and are susceptible to interference from the inherent noise of the data. Another line of research focuses on applying deep learning to the task of clustering scRNA-seq data. For example, DEC [30] uses neural networks to learn feature representations and cluster assignments; scDeepCluster [22] uses the ZINB model [5] to simulate the distribution of scRNA-seq data and combines it with deep learning embedding clustering. However, all the above approaches focus on learning the features of a single cell while ignoring the intercellular structural information, which is important for describing the differences between cells.

To bridge this gap, a series of methods that create cell-cell graphs based on relationships between cells have emerged to incorporate intercellular structural information [7]. Moreover, in order to introduce high-order structural information, the latest scRNA-seq data clustering methods have started to use Graph Neural Networks (GNNs). Specifically, scGNN [27] utilizes GNNs and three multi-modal autoencoders to formulate and aggregate cell-cell relationships; scDSC [6] combines a ZINB-based autoencoder and GNNs in a mutually supervised manner, enhancing the clustering performance. Despite considering intercellular structure information in scRNA-seq data, the GNN-based clustering still has some shortcomings: (i) As single-cell transcriptome sequencing technology continues to advance, the scale of scRNA-seq data is getting increasingly larger. However, GNNs have high complexity and poor efficiency, i.e., longer training/inference time, especially when facing large-scale datasets [29]. (ii) GNNs tend to generate similar representations for all nodes [17,32,28], which will even lead to the over-smoothing issue [4] (i.e., indistinguishable representations of nodes in different classes) when GNNs become deeper (i.e., more layers). This problem will become more severe when GNNs face the scRNA-seq data, because scRNA-seq

data's high-dimension and high-sparsity will result in the similarity between all cell being close to 0, thus all cells are more poorly distinguishable. Therefore, GNN-based clustering methods face challenges in handling the inefficiency due to scRNA-seq data's intrinsic characteristics.

Thus, it is crucial to develop a scRNA-seq data clustering method that simultaneously utilizes intercellular high-order structural information, handles high-dimensional and high-sparse scRNA-seq datasets, and demonstrates high efficiency. In order to achieve this goal, we introduce **scCDCG** (**s**ingle-**c**ell RNA-seq **C**lustering via **D**eep **C**ut-informed **G**raph). The scCDCG consists of three main modules, specifically: (i) scCDCG incorporates complex intercellular high-order structural information through a novel and simple method of graph embedding module based on deep cut-informed techniques [19], while avoiding the efficiency limitations and over-smoothing issue of GNN-based methods. It is well-suited for high-dimensional and high-sparse scRNA-seq data, demonstrating enhanced efficiency. (ii) scCDCG utilizes a new self-supervised learning module based on optimal transport to guide the model learning process [15], which is more adaptive to high-dimensional and high-sparse scRNA-seq data and enhances clustering accuracy and robustness. (iii) scCDCG employs an autoencoder-based feature learning module for dimension reduction and feature extraction, effectively reducing the model complexity. To fully verify the capacity and effectiveness of scCDCG, we conduct extensive experiments with 7 established models on 6 datasets, the results demonstrate the superior clustering performance and efficiency of scCDCG. In addition, we conduct sufficient ablation and analysis experiments to validate the effectiveness of each module of scCDCG.

Our key contributions can be summarized as follows:

- We propose scCDCG, modeled from a graph cutting perspective, which is more suitable for high-dimensional, high-sparse scRNA-seq data, exploiting the complex intercellular higher-order structural information while avoiding the over-smoothing problem and efficiency limitations of previous GNN-based methods.
- We introduce a self-supervised learning module via optimal transport to guide the model learning process, which can better adapt to the complex structure of scRNA-seq data, especially the properties of high-dimension and high-sparsity.
- Extensive experiments are conducted on various real-world scRNA-seq datasets to demonstrate the excellent efficiency and superior performance of scCDCG. We also perform comprehensive ablation experiments to demonstrate that each component of our method is indispensable.

2 Related Work

2.1 Classical Clustering Methods for ScRNA-seq Data

In scRNA-seq data analysis, early clustering methods primarily employed hard clustering algorithms. pcaReduce [33] is a notable example, combining Principal

Component Analysis (PCA) with K-means clustering. This method iteratively merges clusters based on probability density functions, capturing key data variances. Similarly, SUSSC [26] integrates Uniform Manifold Approximation and Projection (UMAP) with K-means, enhancing data dimensionality reduction and cluster accuracy. Deep learning also gained prominence in scRNA-seq clustering. DEC [30] employs neural networks for feature representation and clustering, adapting to complex gene expression profiles. Meanwhile, scDeepCluster [22] utilizes the Zero-Inflated Negative Binomial (ZINB) model [5] in conjunction with deep learning, addressing data sparsity and over-dispersion for more precise clustering. However, all the above approaches focus on learning the features of a single cell while ignoring the intercellular structural information, which is important for describing the differences between cells.

2.2 Deep Structural Clustering Methods for ScRNA-seq Data

In the field of single-cell RNA sequencing (scRNA-seq) data analysis, deep structural clustering methods have gained prominence. Notably, scGNN employs Graph Neural Networks (GNNs) [27] and multi-modal autoencoders to aggregate cell-cell relationships and model gene expression patterns using a left-truncated mixture Gaussian model. scDSC [6], on the other hand, combines a ZINB model-based autoencoder with GNN modules, integrating these using a mutually supervised strategy for enhanced data representation. GraphSCC [31] utilizes a graph convolutional network to capture structural cell-cell relationships, optimizing the network's output with a self-supervised module. Lastly, scGAE [13] leverages a graph autoencoder to maintain both feature and topological structure information of scRNA-seq data, illustrating the diversity and sophistication of current approaches in the field.

3 Methodology

3.1 Definition

Assume the scRNA-seq data as $\boldsymbol{X} \in \mathbb{R}^{n\times m}$, where $\boldsymbol{x}_{ij}\,(1 \le i \le n, 1 \le j \le m)$ shows j-th gene's expression in the i-th cell. Denote $\boldsymbol{H}_{n\times d} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n]^T$ is the latent embedding of $\boldsymbol{X}$ and $\boldsymbol{c} = [c_1, ..., c_n]$ is the clustering assignment. We will simultaneously learn the embedding $\boldsymbol{H}$ and clustering assignments $\boldsymbol{c}$ in this paper.

3.2 The framework of scCDCG

We propose scCDCG, a deep cut-informed graph model-based single cell clustering method (see Fig. 1 for its architecture), which includes (i) an autoencoder-based feature learning module for learning gene expression embeddings, (ii) a graph embedding module based on deep cut-informed techniques for capturing intercellular high-order structural information, and (iii) a self-supervision learning module via optimal transport for generating clustering assignments.

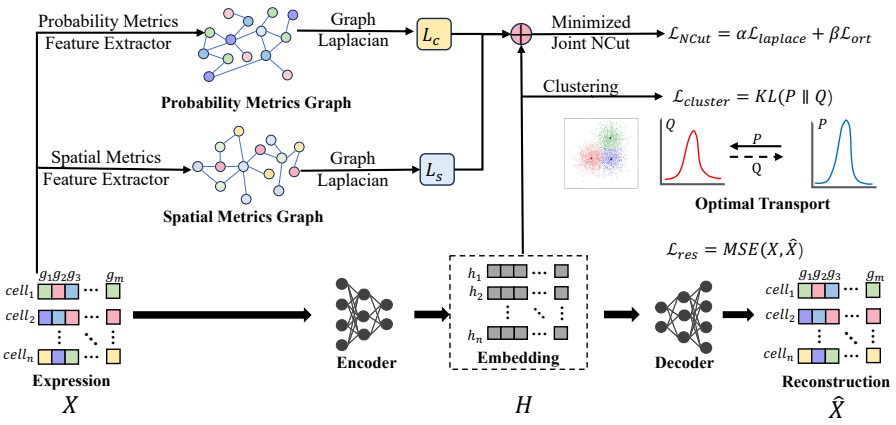


Fig. 1: The model architecture of scCDCG. The scCDCG contains three core modules: (i) an autoencoder, (ii) a two-channel graph embedding module, and (iii) an optimal transport-based self-supervision learning module.

Feature learning module based an autoencoder. To reduce complexity (i.e., using the mean square error loss function), which can map high-dimensional scRNA-seq data to a low-dimensional feature space. We adopt an unsupervised autoencoder [11]. The encoder of the autoencoder maps $\boldsymbol{X}$ to the latent representation H as:

$$\boldsymbol{H} = f_{enc}(W\boldsymbol{X} + b). \tag{1}$$

The decoder maps H to a reconstruction $\hat{\boldsymbol{X}}$ of the raw data, as:

$$\hat{\boldsymbol{X}} = f_{dec}(W^{'}\boldsymbol{H} + b^{'}), \tag{2}$$

where b and $b^{'}$, respectively, represent the bias vectors of the encoder and the decoder, W and $W^{'}$ are the weight matrix of the encoder and the decoder.

Thus, the reconstruction loss function can be written as:

$$\mathcal{L}_{res} = \frac{1}{2N}\sum_{i=1}^{N}\|x_i - \hat{x_i}\|_2^2 = \frac{1}{2N}\|X - \hat{X}\|_F^2. \tag{3}$$

Graph embedding module based on deep cut-informed techniques. This module can leverage intercellular high-order structural information and avoid the over-smoothing issue and scalability limitations of GNN-based methods, making it well-suited for high-dimension and high-sparsity scRNA-seq data and demonstrating enhanced scalability.

Specifically, we employ a dual-channel approach, processing the raw data X concurrently through both probability metrics and spatial metrics feature extractors, which yields the probability metrics graph $\mathcal{G}_{\mathcal{C}}$ and the spatial metrics graph $\mathcal{G}_{\mathcal{S}}$. Here, $\mathcal{V}$ is a set of n nodes in the graph, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set

of graph and $\boldsymbol{A} \in \{0,1\}^{n\times n}$ is the adjacency matrix of graph. The two channels are as follows:

Channel 1: The probability metrics feature matrix $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ is calculated as $\boldsymbol{C} = \boldsymbol{X}\boldsymbol{X}^T$. Each entry $\boldsymbol{C}_{ij}$ in the matrix represents the covariance between cells i and j. Then we use the probability metrics matrix to construct a probability metrics graph $\mathcal{G}_{\mathcal{C}} = (\mathcal{V}, \boldsymbol{C}, \mathcal{E}, \boldsymbol{A}_C)$, each node represents a cell.

Channel 2: The spatial metrics feature matrix $\boldsymbol{S} \in \mathbb{R}^{n\times n}$is computed as $\boldsymbol{S}_{ij} = \frac{\boldsymbol{x}_i^T \boldsymbol{x}_j}{|\boldsymbol{x}_i|_2 |\boldsymbol{x}_j|_2}$. Each entry $\boldsymbol{S}_{ij}$ in the matrix represents the pairwise cosine similarity between cells i and j. Then we utilize it to obtain a spatial metrics graph $\mathcal{G}_{\mathcal{S}} = (\mathcal{V}, \boldsymbol{S}, \mathcal{E}, \boldsymbol{A}_S)$.

Meanwhile, we proposed a graph embedding module based on deep cut-informed to fuse the probability metrics graph and spatial metrics graph by minimizing their joint normalized cut [19]. For graphs $\mathcal{G}_{\mathcal{C}}$ and $\mathcal{G}_{\mathcal{S}}$, $\boldsymbol{D}_{\boldsymbol{C}}$, $\boldsymbol{D}_{\boldsymbol{S}}$ are their diagonal matrices respectively. Here, the normalized graph Laplacian of $\boldsymbol{C}$ and $\boldsymbol{S}$ can be formulated as:

$$\begin{aligned} L_{\boldsymbol{C}} &= \boldsymbol{I} - \boldsymbol{D}_{\boldsymbol{C}}^{-1/2} \boldsymbol{C} \boldsymbol{D}_{\boldsymbol{C}}^{-1/2}, \\ L_{\boldsymbol{S}} &= \boldsymbol{I} - \boldsymbol{D}_{\boldsymbol{S}}^{-1/2} \boldsymbol{S} \boldsymbol{D}_{\boldsymbol{S}}^{-1/2}. \end{aligned} \tag{4}$$

Given two subsets of $\mathcal{V}$ denoted as $\mathcal{V}_1$ and $\mathcal{V}_2$, we define $\boldsymbol{A}(\mathcal{V}_1, \mathcal{V}_2) = \sum_{i\in\mathcal{V}_1, j\in\mathcal{V}_2} A_{ij}$. Then for a partition $\mathcal{V} = \mathcal{V}_1 \bigcup \cdots \bigcup \mathcal{V}_K$, its normalized cut [19] can be defined as follows:

$$\mathrm{Ncut}(\mathcal{V}_1, \cdots, \mathcal{V}_K) = \frac{1}{2} \sum_{i=1}^{K} \frac{\boldsymbol{A}(\mathcal{V}_i, \bar{\mathcal{V}}_i)}{\mathrm{vol}(\mathcal{V}_i)}, \tag{5}$$

where $\bar{\mathcal{V}}_i$ is the complementary set of $\mathcal{V}_i$ and $\mathrm{vol}(\mathcal{V}_i) = \boldsymbol{A}(\mathcal{V}_i, \mathcal{V}_i) + \boldsymbol{A}(\mathcal{V}_i, \bar{\mathcal{V}}_i)$ is the total edges of $\mathcal{V}_i$. According to this criterion, an optimal graph partition should have minimized Ncut. However, it is NP-hard to find the optimal graph partition. Inspired by covariates-assisted spectral clustering, we can alternatively solve it with the following optimization strategy:

$$\begin{aligned} &\arg\min_{\boldsymbol{H}} \mathrm{Tr}(\boldsymbol{H}^T L_{\boldsymbol{C}} \boldsymbol{H}) \quad \text{subject to } \boldsymbol{H}^T \boldsymbol{H} = \boldsymbol{I}, \\ &\arg\min_{\boldsymbol{H}} \mathrm{Tr}(\boldsymbol{H}^T L_{\boldsymbol{S}} \boldsymbol{H}) \quad \text{subject to } \boldsymbol{H}^T \boldsymbol{H} = \boldsymbol{I}, \end{aligned} \tag{6}$$

which is a relaxation of minimizing the normalized cut. $\boldsymbol{H}$ is a soft partition of graph $\mathcal{G}$ in terms of minimizing normalized cut, which implies that $\boldsymbol{H}$ can be viewed as the node embeddings for clustering. It is a natural idea to make the learned embedding $\boldsymbol{H}$ by Equation 6 contain the information of both probability metrics and spatial metrics graphs, so the optimization function is following:

$$\begin{aligned} &\arg\min_{\boldsymbol{H}} \mathrm{Tr}(\boldsymbol{H}^T (\alpha L_{\boldsymbol{C}} + (1-\alpha) L_{\boldsymbol{S}}) \boldsymbol{H}) \\ &\text{subject to } \boldsymbol{H}^T \boldsymbol{H} = \boldsymbol{I}, \end{aligned} \tag{7}$$

where α is the balance parameter between probability metrics graph and spatial metrics graph. So the NCut loss function can be written as:

$$\mathcal{L}_{\mathrm{NCut}} = \beta \mathrm{Tr}(\boldsymbol{H}^T (\alpha L_{\mathcal{G}} + (1-\alpha) L_{\boldsymbol{S}}) \boldsymbol{H}) + \gamma \|\boldsymbol{H}^T \boldsymbol{H} - \boldsymbol{I}\|_F, \tag{8}$$

where β, γ are both tuning parameters and $\|\boldsymbol{H}^T\boldsymbol{H}-\boldsymbol{I}\|_F$ is the loss to guarantee the orthogonality.

Note that, the learned embedding $\boldsymbol{H}$ is better for clustering from two viewpoints: (i) The loss function is a relaxation of minimizing the joint normalized cut of probability metrics graph and spatial metrics graph, which means $\boldsymbol{H}$ is a soft assignment of the graph partition from the perspective of minimizing normalized cut. (ii) The trade-off between the probability metrics graph and spatial metrics graph makes $\boldsymbol{H}$ consistent with both two graphs, which maintains the structure supported by both two graphs.

Self-supervision learning module via optimal transport. To better adapt to the complex structure of scRNA-seq data, especially the two properties of high-dimension and high-sparsity, we adopt a self-supervised approach introduced by [30], which can generate learning assignments. It employs the Student's t-distribution [23] as a kernel to measure the similarity between the embedded point h_i and the clustering center c_j.

$$q_{ij} = \frac{(1+\|h_i - c_j\|^2/\theta)^{-\frac{1+\theta}{2}}}{\sum_{j'}(1+\|h_i - c_{j'}\|^2/\theta)^{-\frac{1+\theta}{2}}}, \tag{9}$$

where $h_i = f(\boldsymbol{x}_i) \in \boldsymbol{H}$ corresponds to $\boldsymbol{x}_i \in \boldsymbol{X}$ after embedding, θ is the degrees of freedom of the Student's t-distribution, and q_{ij} can be interpreted as the probability of assigning sample i to cluster j (i.e., a soft assignment). Then, we define $Q = [q_{ij}]$ as the distribution of the assignments of all samples.

After obtaining the clustering result distribution Q, we aim to optimize the data representation by learning the high-confidence assignments. This approach aims to minimize the difference between the data representations and the cluster centers, thereby improving the cohesion of the clusters. For instance, SDCN [2] calculated a target distribution $P = [p_{ij}]$:

$$p_{ij} = \frac{q_{ij}^2/f_j}{\sum q_{ij'}^2/f_{j'}}, \tag{10}$$

where $f_j = \sum_i a_{ij}$ are soft cluster frequencies. In the target distribution P, each assignment in Q is squared and normalized to enhance assignment confidence [2]. However, to prevent the emergence of degenerate solutions that assign all data points to a single (arbitrary) label, we introduce constraints that align the label distribution with the mixing proportions. This approach ensures improved clustering accuracy and the equalized contribution of each data point to the loss calculation. Therefore, we construct the target probability matrix P by solving the following optimal transport strategy:

$$\begin{aligned} \min_{P} \quad & -P * (\log Q) \\ \text{s.t.} \quad & P \in \mathbb{R}_+^{N\times C}, \\ & P\,\mathbf{1}_C = \mathbf{1}_N \ \text{ and } \ P^T\mathbf{1}_N = N\boldsymbol{\pi}. \end{aligned} \tag{11}$$

Here, we consider the target distribution P as the transport plan matrix within the optimal transport theory and $-\log Q$ as the associated cost matrix. The constraint $P^T\mathbf{1}_N = N\boldsymbol{\pi}$ is imposed, with π representing the proportion of points in each cluster, which can be estimated by the intermediate clustering result. This step successfully adds the constraint that the result cluster distribution must be consistent with the mixing proportions. Given the computational burden of direct optimization, we leverage the Sinkhorn distance [20] for rapid optimization through an entropic constraint. The optimization problem with an entropy constraint Lagrange multiplier is as follows:

$$\begin{aligned} \min_{P} \quad & -P * (\log Q) - \frac{1}{\lambda}\mathrm{H}(P) \\ \text{s.t.} \quad & P \in \mathbb{R}_{+}^{N\times C}, \\ & P\ \mathbf{1}_C = \mathbf{1}_N \ \text{ and } \ P^T\mathbf{1}_N = N\boldsymbol{\pi}, \end{aligned} \tag{12}$$

where $\boldsymbol{H}$ is the entropy function and λ is the smoothness parameter controlling the equilibrium of clusters. The existence and unicity of P are guaranteed, and can be effciently solved by Sinkhorn's [20] fixed point iteration:

$$\boldsymbol{T}^* = \mathrm{diag}(\boldsymbol{u}^{(t)})Q^{\lambda}\mathrm{diag}(\boldsymbol{v}^{(t)}), \tag{13}$$

where t denotes the iteration and in each iteration: $\boldsymbol{u}^{(t)} = \mathbf{1}_N/(Q^{\lambda}\boldsymbol{v}^{(t-1)})$ and $\boldsymbol{v} = N_{\pi}/(Q^{\lambda}\boldsymbol{u}^{(t)})$, with the initiation $\boldsymbol{v}^{(0)} = \mathbf{1}_N$. After the fixed point iteration, we can get the optimal transport plan matrix $\hat{P}$, i.e., the solution of (14).

During the training process, we fix $\hat{P}$ and make Q be consistent with $\hat{P}$, so the clustering loss function can be formulated as:

$$\mathcal{L}_{\mathrm{KL}} = \mathrm{KL}(\hat{P}\|Q). \tag{14}$$

Thus, the overall loss function is

$$\mathcal{L} = \mu\mathcal{L}_{res} + \sigma\mathcal{L}_{NCut} + \tau\mathcal{L}_{KL}, \tag{15}$$

where μ, σ and τ are tuning parameters to balance the three kinds of loss, $\mathcal{L}_{res}$, $\mathcal{L}_{NCut}$ and $\mathcal{L}_{KL}$.

3.3 Overall algorithm

In practice, we first train the model with both NCut loss and reconstruction loss by 200 epochs. After that, we apply K-means to the embeddings to obtain the initial centroids and clustering sizes. Then, we train the model with 200 epochs each for NCut loss, reconstruction loss, and clustering loss. Ultimately, clustering results are obtained at the last epoch.

4 Results

4.1 Experimental Setup

Dataset. We assessed the performance of the model using six scRNA-seq datasets, each of which was accompanied by provided cell labels. Of the six datasets, four

Table 1: Summary of the scRNA-seq datasets.

Datasets	Sample Size	Gene Number	Group Number
Human Pancreas cells 2 [1]	1724	20125	14
Meuro Human Pancreas cells [16]	2122	19046	9
Mouse Bladder cells [9]	2746	20670	16
Worm Neuron cells [3]	4186	13488	10
CITE-CMBC [9]	8617	2000	15
Human Liver cells [14]	8444	4999	11

are derived from human samples, one from mouse samples, and one from worm samples. The datasets cover a wide range of cell types, including pancreas, bladder, neurons, liver, and peripheral blood mononuclear cells. A detailed description of these datasets is provided in Table 1.

Evalution metrics. To show the effectiveness of the proposed method, the clustering performance is evaluated by three established metrics from the public domain: Accuracy (ACC), Normalized Mutual Information (NMI) [21], and Adjusted Rand Index (ARI) [24].

Competing Methods. We compare our model with the 7 competing methods to further evaluate its clustering performance. Detailed categorization and description are as follows:

- **Early Clustering Methods**: **pcaReduce** [33] iteratively combines clusters based on relevant probability density functions; **SUSSC** [26] uses stochastic gradient descent and nearest neighbor search for clustering.
- **Deep Clustering Methods**: **DEC** [30] is deep-embedded clustering, employing neural networks to learn both feature representations and cluster assignments; **scDeepCluster** [22] utilizes the ZINB model to simulate the distribution of scRNA-seq data and combines deep embedding clustering to learn feature representation and clustering simultaneously; **scNAME** [25] integrates a mask estimation task and a neighborhood contrastive learning framework for denoising and clustering scRNA-seq data.
- **Deep Structural Clustering Methods**: **scGNN** [27] utilizes GNNs and three multi-modal autoencoders to formulate and aggregate cell-cell relationships and models heterogeneous gene expression patterns using a left-truncated mixture Gaussian model; **scDSC** [6] learns data representations using the ZINB model-based autoencoder and GNN modules, and uses mutual supervised strategy to unify these two architectures.

Implementation Details. In the experiments, we implement scCDCG in Python 3.7 based on PyTorch. For baseline approaches, we report the results stated in their papers. The training phase uses the same hyperparameters as the pre-training phase, such as learning rate, weight decay, and other parameters. The

Table 2: Clustering performances of all models on ten datasets (mean ± std). The best results are highlighted in **bold**, and the runner-up results are highlighted in underline. (Higher values indicate better performance.)

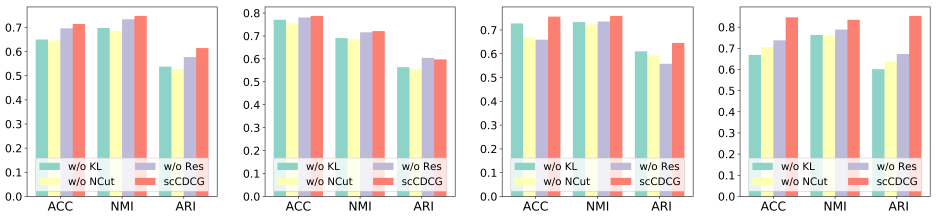
Dataset	Metric	pcaReduce	SUSSC	DEC	scDeep-Cluster	scNAME	scGNN	scDSC	scCDCG
Human Pancreas cells 2	ACC	37.56 ± 0.4	46.75 ± 0.2	45.26 ± 1.8	62.88 ± 5.5	63.33 ± 2.0	57.62 ± 2.3	80.40 ± 4.0	**84.66** ± 1.1
	NMI	50.81 ± 0.3	68.62 ± 0.0	66.32 ± 2.1	73.97 ± 2.2	58.05 ± 1.6	79.32 ± 2.3	79.44 ± 1.7	**83.44** ± 2.6
	ARI	19.23 ± 0.5	38.45 ± 0.4	39.14 ± 1.8	64.56 ± 7.5	32.12 ± 1.5	79.96 ± 1.7	82.85 ± 6.5	**85.30** ± 1.6
Meuro Human Pancreas cells	ACC	50.56 ± 3.4	76.26 ± 0.1	72.22 ± 5.5	60.89 ± 3.1	80.36 ± 7.9	75.12 ± 4.2	69.18 ± 3.7	**92.65** ± 1.9
	NMI	54.83 ± 1.7	62.25 ± 0.0	76.29 ± 2.7	70.02 ± 2.0	79.12 ± 2.7	72.15 ± 2.2	64.31 ± 1.3	**86.81** ± 1.0
	ARI	36.61 ± 2.3	64.23 ± 0.0	61.76 ± 5.6	55.14 ± 2.4	76.6 ± 10.35	72.13 ± 3.1	56.63 ± 1.2	**91.37** ± 1.2
Mouse Bladder cells	ACC	31.75 ± 0.5	50.60 ± 1.6	50.72 ± 2.9	74.30 ±3.8	62.45 ± 4.0	54.21 ± 3.2	74.32 ± 3.9	**75.61** ± 1.2
	NMI	44.15 ± 0.3	65.35 ± 0.1	62.52 ± 2.2	74.99 ± 2.0	61.96 ± 1.9	69.86 ± 1.7	74.42 ± 2.4	**75.91** ± 0.9
	ARI	36.61 ± 2.3	39.35 ± 1.0	41.86 ± 6.28	62.33 ± 5.4	39.51 ± 1.6	56.45 ± 5.3	62.52 ± 4.7	**64.55** ± 2.3
Worm Neuron cells	ACC	29.30 ± 3.7	54.94 ± 0.8	40.96 ± 3.9	65.71 ± 4.0	66.89 ± 2.0	59.32 ± 2.1	60.93 ± 3.7	**78.73** ± 2.9
	NMI	25.45 ± 14.4	43.84 ± 0.1	38.61 ± 2.5	67.58 ± 1.1	64.23 ± 2.1	48.73 ± 1.4	61.63 ± 3.7	**72.11** ± 2.0
	ARI	8.65 ± 8.8	32.69 ± 5.5	19.99 ± 1.5	49.91 ± 4.3	55.59 ± 1.1	30.68 ± 1.8	54.10 ± 4.2	**59.78** ± 1.9
CITE-CMBC	ACC	28.19 ± 0.1	47.22 ± 0.1	41.88 ± 5.4	70.80 ± 2.5	68.06 ± 13.7	65.32 ± 2.0	67.69 ± 2.8	**71.45** ± 1.8
	NMI	28.36 ± 0.2	60.14 ± 0.0	46.87 ± 9.2	72.53 ± 0.7	63.23 ± 13.2	61.42 ± 3.1	63.80 ± 2.7	**74.77** ± 1.7
	ARI	5.10 ± 0.1	35.56 ± 1.6	23.55 ± 10.3	56.23 ± 4.1	58.99 ± 21.3	49.56 ± 1.6	50.09 ± 1.6	**61.46** ± 1.4
Human Liver cells	ACC	34.92 ± 0.2	46.62 ± 1.6	46.32 ± 5.5	63.44 ± 0.7	73.55 ± 4.8	68.65 ± 3.2	67.90 ± 3.8	**75.34** ± 1.7
	NMI	32.07 ± 0.6	67.32 ± 0.0	52.24 ± 5.0	74.60 ± 2.5	75.99 ± 2.0	62.33 ± 2.1	57.93 ± 2.6	**79.34** ± 2.6
	ARI	6.34 ± 14.4	35.12 ± 0.0	31.04 ± 4.7	58.58 ± 10.1	72.40 ± 9.4	65.41 ± 1.3	57.34 ± 2.7	**81.26** ± 2.7

encoder network contains layers of $[256, 16]$ to produce a latent representation of 16 dimensions, and the decoder network has a symmetric structure. The smoothness parameter λ of the optimal transport strategy is set as 5 for all datasets; the degrees of freedom of the Student's t-distribution θ is set as 1 for all datasets; besides that, each dataset has unique hyperparameters, which are achieved by the grid search. The network is trained by the Adam optimizer with the initial learning rate and weight decay to update variables. Furthermore, hyperparameter settings for all datasets are available in the code. During network training, we train the scCDCG for 200 epochs. In order to ensure the accuracy of the experimental results, we conducted each experiment 10 times with random seeds on all the datasets and reported the mean and variance of the results.

4.2 Overall Clustering Performance

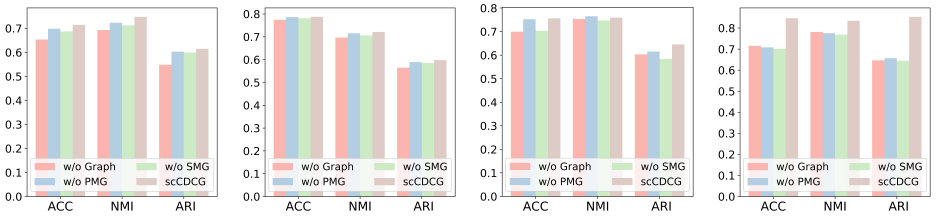
To verify the superior performance of scCDCG, we benchmarked it against 7 competing methods on 6 real-world datasets. In the comparative analysis, we utilize three widely used metrics (ACC, NMI, and ARI) to evaluate the clustering performance of each method, as summarized in Table 2.

From these results, we note several observations: For each metric, our approach scCDCG is significantly better than the best results of competing methods on all datasets, achieving a significant improvement of 5.58% on ACC, 3.79% on NMI, 5.79% on ARI averagely. The deep learning-based methods and the



(a) CITE_CMBC (b) Worm Neuron Cells (c) Mouse Bladder Cells (d) Human Pancreas Cells 2

Fig. 2: Ablation comparisons of all components on four datasets.



(a) CITE_CMBC (b) Worm Neuron Cells (c) Mouse Bladder Cells (d) Human Pancreas Cells 2

Fig. 3: Effect of various graph information on model performance.

GNN-based methods significantly outperform the traditional ones. Combined with Table 1, we can also observe that scCDCG achieves good results on datasets with feature dimensions of different sizes, which contain genes ranging from 2,000 to 20,670, and demonstrates higher efficiency on scRNA-seq data with high-dimension and high-sparsity. Accordingly, in the clustering task of scRNA-seq data, scCDCG can extract intricate intercellular high-order structural information and successfully apply it to clustering methods, compared to comparative methods, outperforming the existing state-of-the-art baselines.

4.3 Ablation Study

Effect of each main components. We conduct ablation studies to evaluate the contributions of different components in our method. Here, we denote the model without NCut loss as the *scCDCG-w/o NCut*. *scCDCG-w/o KL* and *scCDCG-w/o Res* denote the model without clustering loss and reconstruction loss, respectively. Likewise, scCDCG denotes our model that includes all components. Based on the results shown in Fig. 2, we can conclude that each component of our model plays an important role and they improve the performance of scCDCG, making it more efficient.

Effect of graph information. To comprehensively demonstrate the significance of both the probability metrics graph (PMG) and spatial metrics graph

Table 3: Ablation study of orthogonality regularization and optimal transport.

Dataset	Metric	*scCDCG -w/o Ort*	*scCDCG -w/o OT*	scCDCG
CITE-CMBC	ACC	56.82	69.45	**71.45**
	NMI	57.99	71.83	**79.34**
	ARI	45.47	81.26	**81.26**
Worm Neuron cells	ACC	63.16	70.18	**78.73**
	NMI	53.82	65.35	**72.11**
	ARI	40.43	50.88	**59.78**
Mouse Bladder cells	ACC	62.97	70.89	**75.61**
	NMI	69.05	65.21	**75.91**
	ARI	54.55	58.45	**64.55**
Human Pan -creas cells 2	ACC	58.77	70.03	**84.66**
	NMI	68.14	70.87	**83.44**
	ARI	55.31	62.41	**85.30**

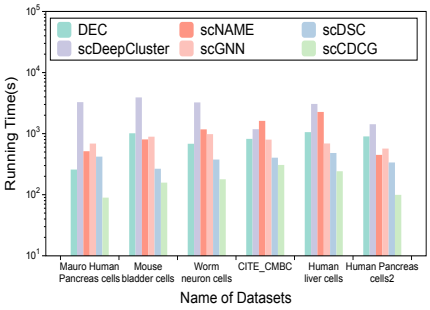


Fig. 4: The running time of scCDCG and five baselines.

(SMG), we conduct ablation studies on four distinct datasets with three metrics. Here, we denote the model without cut-informed graph embedding as *scCDCG-w/o Graph* since it does not include probability metrics graph and spatial metrics graph. *scCDCG-w/o PMG* and *scCDCG-w/o SMG* denote the model without the the probability metrics graph and spatial metrics graph, respectively. And scCDCG denotes our model that who contains these two graphs. Notably, Fig. 3 shows, each graph improves the performance compared with the baseline. *scCDCG-w/o PMG* outperforms *scCDCG-w/o SMG* across all metrics, indicating that SMG can better capture intercellular high-order structural information between cells, with greater gains compared to SMG in the clustering task of scRNA-seq data. In addition, the final model scCDCG with both PMG and SMG shows superior performance compared to these baseline models, suggesting that the two graphs can interact to obtain more complex high-order intercellular structural information, crucial to account for intercellular heterogeneity in the clustering task of scRNA-seq data.

Effect of optimal transport. We use the optimal transport theorem to guarantee the clustering process and improve the clustering assignments. To illustrate our optimal transport based self-supervised clustering module is better than the traditional approaches (e.g., DEC [30]), which adopts a clustering guided loss function to force the generated sample embeddings to have the minimum distortion against the pre-learned clustering center, we design this experiment. In Table 3, *scCDCG-w/o OT* denotes that the baseline does not use the proposed optimal transport based self-supervised clustering module but instead uses the traditional clustering guided loss function. The results show that scCDCG outperforms *scCDCG-w/o OT* in terms of three metrics, indicating that introducing optimal transport can better adapt to the complex structure of scRNA-seq data, especially the two properties of high-dimension and high-sparsity.

Effect of orthogonality regularization. From the perspective of minimizing normalized cut, the orthogonality of $\boldsymbol{H}$ is necessary. We experimentally compare our method with baselines to explore the superiority of the proposed orthogo-

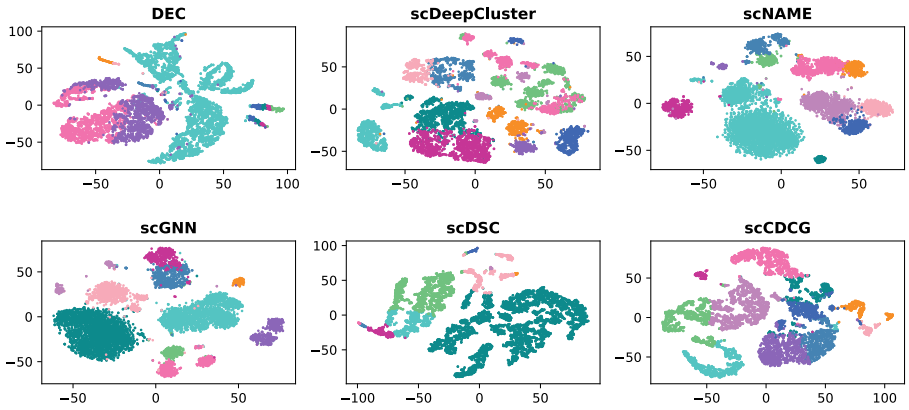


Fig. 5: The visualization of scCDCG and five baselines on *human liver cells*.

nality regularization. Here, *scCDCG-w/o Ort* is denoted as the baseline without the proposed orthogonality regularization. Table 3 presents the results, scCDCG consistently outperforms *scCDCG-w/o Ort* across all four datasets. Such results indicate that the introduction of the orthogonality regularization in the graph embedding module is necessary to make the learned node embeddings sufficiently and thus avoid over-similarity and thus achieve better clustering results when processing scRNA-seq data.

4.4 Efficiency Study

With the increasing number of cells profiled in scRNA-seq experiments, there is an urgent need to develop analysis methods that can effectively handle large datasets. To evaluate the efficiency of scCDCG, we measured the runtime of five competing models (both deep learning-based and GNN-based approaches) and scCDCG on five datasets. As shown in Fig. 4, scCDCG exhibits a significant advantage in terms of time efficiency on datasets of varying sizes, while competing methods exhibit varying degrees of computational burden. This shows that scCDCG remains efficient in terms of runtime despite changes in dataset size. The excellent performance on large-scale datasets further validates the potential of scCDCG as a powerful tool for processing biological data of different sizes.

4.5 Visualization

The latent space in the proposed scCDCG model is an ideal low-dimensional embedded representation of the high-dimensional input data. To demonstrate the effectiveness of the latent space in the scCDCG model, we employ t-SNE [10] for visualizing the final embedded points in the two-dimensional(2D) space, which are reflected in H. Each point represents a cell, while each color represents a predicted cell type. No method uses the true label information. To verify the

superiority of scCDCG over competing methods, we plot the 2D space representations of all methods on *human liver cells*, which contains 8,444 cells and 11 cell-types(see Fig. 5). scCDCG has clear boundaries between various cell subtypes, where cells of the same type are separated well, and much better than competing methods. We can see that the representations obtained by scCDCG are discriminative, and each cluster is compact. Overall, the results in Fig. 5 show that scCDCG performs well in the discrimination of cell subtypes, which can effectively capture the intercellular high-order structural information of the scRNA-seq data.

5 Conclusion

In conclusion, our study introduces scCDCG, an innovative framework for the efficient and accurate clustering of single-cell RNA sequencing (scRNA-seq) data. scCDCG successfully navigates the challenges of high-dimension and high-sparsity through a synergistic combination of a graph embedding module with deep cut-informed techniques, a self-supervised learning module guided by optimal transport, and an autoencoder-based feature learning module. Our extensive evaluations on six datasets confirm scCDCG's superior performance over seven established models, marking it as a transformative tool for bioinformatics and cellular heterogeneity analysis. Looking forward, we aim to extend scCDCG's capabilities to integrate multi-omics data, enhancing its applicability in more complex biological contexts. Additionally, further exploration into the interpretability of the clustering results generated by scCDCG will be crucial for providing deeper biological insights and facilitating its adoption in clinical research settings. This future work will continue to expand the frontiers of scRNA-seq data analysis and its impact on understanding the complexities of cellular systems.

6 Acknowledgements

This work is partially supported by the Strategic Priority Research Program of the Chinese Academy of Sciences XDB38030300, the Informatization Plan of Chinese Academy of Sciences (CAS-WX2021SF-0101, CAS-WX2021SF-0111), the Postdoctoral Fellowship Program of CPSF (No.GZC20232736), the China Postdoctoral Science Foundation Funded Project (No.2023M743565) and the X-Compass Project.

References

1. Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M Klein, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell systems*, 3(4):346–360, 2016.

2. Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural deep clustering network. In *Proceedings of the web conference 2020*, pages 1400–1410, 2020.
3. Junyue Cao, Jonathan S Packer, Vijay Ramani, Darren A Cusanovich, Chau Huynh, Riza Daza, Xiaojie Qiu, Choli Lee, Scott N Furlan, Frank J Steemers, et al. Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science*, 357(6352):661–667, 2017.
4. Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3438–3445, 2020.
5. Gökcen Eraslan, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis. Single-cell rna-seq denoising using a deep count autoencoder. *Nature communications*, 10(1):390, 2019.
6. Yanglan Gan, Xingyu Huang, Guobing Zou, Shuigeng Zhou, and Jihong Guan. Deep structural clustering for single-cell rna-seq data jointly through autoencoder and graph neural network. *Briefings in Bioinformatics*, 23(2):bbac018, 2022.
7. Yanglan Gan, Ning Li, Guobing Zou, Yongchang Xin, and Jihong Guan. Identification of cancer subtypes from single-cell rna-seq data using a consensus clustering method. *BMC medical genomics*, 11:65–72, 2018.
8. Laleh Haghverdi, Maren Büttner, F Alexander Wolf, Florian Buettner, and Fabian J Theis. Diffusion pseudotime robustly reconstructs lineage branching. *Nature methods*, 13(10):845–848, 2016.
9. Xiaoping Han, Renying Wang, Yincong Zhou, Lijiang Fei, Huiyu Sun, Shujing Lai, Assieh Saadatpour, Ziming Zhou, Haide Chen, Fang Ye, et al. Mapping the mouse cell atlas by microwell-seq. *Cell*, 172(5):1091–1107, 2018.
10. G Hinton and L van der Maaten. Visualizing data using t-sne journal of machine learning research. 2008.
11. Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.
12. Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.
13. Zixiang Luo, Chenyu Xu, Zhen Zhang, and Wenfei Jin. A topology-preserving dimensionality reduction method for single-cell rna-seq data using graph autoencoder. *Scientific reports*, 11(1):20028, 2021.
14. Sonya A MacParland, Jeff C Liu, Xue-Zhong Ma, Brendan T Innes, Agata M Bartczak, Blair K Gage, Justin Manuel, Nicholas Khuu, Juan Echeverri, Ivan Linares, et al. Single cell rna sequencing of human liver reveals distinct intrahepatic macrophage populations. *Nature communications*, 9(1):4383, 2018.
15. Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pages 666–704, 1781.
16. Mauro J Muraro, Gitanjali Dharmadhikari, Dominic Grün, Nathalie Groen, Tim Dielen, Erik Jansen, Leon Van Gurp, Marten A Engelse, Francoise Carlotti, Eelco Jp De Koning, et al. A single-cell transcriptome atlas of the human pancreas. *Cell systems*, 3(4):385–394, 2016.
17. Zhiyuan Ning, Pengfei Wang, Pengyang Wang, Ziyue Qiao, Wei Fan, Denghui Zhang, Yi Du, and Yuanchun Zhou. Graph soft-contrastive learning via neighborhood ranking. *arXiv preprint arXiv:2209.13964*, 2022.

18. Ehud Shapiro, Tamir Biezuner, and Sten Linnarsson. Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nature Reviews Genetics*, 14(9):618–630, 2013.
19. Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
20. Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967.
21. Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.
22. Tian Tian, Ji Wan, Qi Song, and Zhi Wei. Clustering single-cell rna-seq data with a model-based deep learning approach. *Nature Machine Intelligence*, 1(4):191–198, 2019.
23. Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
24. Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning*, pages 1073–1080, 2009.
25. Hui Wan, Liang Chen, and Minghua Deng. scname: neighborhood contrastive clustering with ancillary mask estimation for scrna-seq data. *Bioinformatics*, 38(6):1575–1583, 2022.
26. Hai-Yun Wang, Jian-ping Zhao, and Chun-Hou Zheng. Suscc: secondary construction of feature space based on umap for rapid and accurate clustering large-scale single cell rna-seq data. *Interdisciplinary Sciences: Computational Life Sciences*, 13:83–90, 2021.
27. Juexin Wang, Anjun Ma, Yuzhou Chang, Jianting Gong, Yuexu Jiang, Ren Qi, Cankun Wang, Hongjun Fu, Qin Ma, and Dong Xu. scgnn is a novel graph neural network framework for single-cell rna-seq analyses. *Nature communications*, 12(1):1882, 2021.
28. Pengfei Wang, Daqing Wu, Chong Chen, Kunpeng Liu, Yanjie Fu, Jianqiang Huang, Yuanchun Zhou, Jianfeng Zhan, and Xiansheng Hua. Deep adaptive graph clustering via von mises-fisher distributions. *ACM Transactions on the Web*, 18(2):1–21, 2024.
29. Zhaokang Wang, Yunpan Wang, Chunfeng Yuan, Rong Gu, and Yihua Huang. Empirical analysis of performance bottlenecks in graph neural network training and inference with gpus. *Neurocomputing*, 446:165–191, 2021.
30. Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.
31. Yuansong Zeng, Xiang Zhou, Jiahua Rao, Yutong Lu, and Yuedong Yang. Accurately clustering single-cell rna-seq data by capturing structural relations between cells through graph convolutional network. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 519–522. IEEE, 2020.
32. Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. Graph neural networks with heterophily. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11168–11176, 2021.
33. Justina Žurauskienė and Christopher Yau. pcareduce: hierarchical clustering of single cell transcriptional profiles. *BMC bioinformatics*, 17:1–11, 2016.