

Soft Graph Clustering for single-cell RNA Sequencing Data

Ping Xu^{1,2}, Pengfei Wang^{1,2}, Zhiyuan Ning^{1,2}, Meng Xiao^{1,2},
Min Wu⁴, Yuanchun Zhou^{1,2,3*}

¹*Computer Network Information Center, Chinese Academy of Sciences,
100083, Beijing, China.

²University of Chinese Academy of Sciences, 100864, Beijing, China.

³Hangzhou Institute for Advanced Study, Chinese Academy of Sciences,
310024, Hangzhou, China.

⁴Institute for Infocomm Research (I2R), Agency for Science, Technology
and Research (A*STAR), 138632, Singapore.

*Corresponding author(s). E-mail(s): zyc@cnic.cn;
Contributing authors: xuping@cnic.con; pfwang@cnic.cn;
ningzhiyuan@cnic.cn; shaow@cnic.cn; wumin@i2r.a-star.edu.sg;

Abstract

Background: Clustering analysis is fundamental in single-cell RNA sequencing (scRNA-seq) data analysis for elucidating cellular heterogeneity and diversity. Recent graph-based scRNA-seq clustering methods, particularly graph neural networks (GNNs), have significantly improved in tackling the challenges of high-dimension, high-sparsity, and frequent dropout events that lead to ambiguous cell population boundaries. However, one major challenge for GNN-based methods is their reliance on hard graph constructions derived from similarity matrices. These constructions introduce difficulties when applied to scRNA-seq data due to: (i) The simplification of intercellular relationships into binary edges (0 or 1) by applying thresholds, which restricts the capture of continuous similarity features among cells and leads to significant information loss. (ii) The presence of significant inter-cluster connections within hard graphs, which can confuse GNN methods that rely heavily on graph structures, potentially causing erroneous message propagation and biased clustering outcomes.

Results: To tackle these challenges, we introduce **scSGC**, a Soft Graph Clustering for single-cell RNA sequencing data, which aims to more accurately characterize continuous similarities among cells through non-binary edge weights,

thereby mitigating the limitations of rigid data structures. The scSGC framework comprises three core components: (i) a zero-inflated negative binomial (ZINB)-based feature autoencoder designed to effectively handle the sparsity and dropout issues in scRNA-seq data; (ii) a dual-channel cut-informed soft graph embedding module, constructed through deep graph-cut information, capturing continuous similarities between cells while preserving the intrinsic data structures of scRNA-seq; and (iii) an optimal transport-based clustering optimization module, achieving optimal delineation of cell populations while maintaining high biological relevance.

Conclusion: By integrating dual-channel cut-informed soft graph representation learning, a ZINB-based feature autoencoder, and optimal transport-driven clustering optimization, scSGC effectively overcomes the challenges associated with traditional hard graph constructions in GNN methods. Extensive experiments across ten datasets demonstrate that scSGC outperforms 13 state-of-the-art clustering models in clustering accuracy, cell type annotation, and computational efficiency. These results highlight its substantial potential to advance scRNA-seq data analysis and deepen our understanding of cellular heterogeneity.

Keywords: Bioinformatics, scRNA-seq data, soft graph clustering, deep cut-informed graph embedding

1 Background

Single-cell RNA sequencing (scRNA-seq) technology precisely captures gene expression differences at the single-cell level, revealing cellular heterogeneity and diversity, and providing crucial insights for bioinformatics [1]. Clustering cells based on gene expression patterns is essential in scRNA-seq analysis, aiding in cell type annotation and marker gene identification [2, 3]. Accurately identifying pathological cell types through clustering provides important information for clinical diagnosis and personalized treatment, thereby significantly advancing precision medicine [4].

Cell clustering has garnered significant research interest, resulting in the development of various methods such as K-means and hierarchical clustering. Advanced techniques like Phenograph [5], MAGIC [6], and Seurat [7] employ k-nearest neighbor (KNN) algorithms to model cellular relationships, with MAGIC pioneering explicit genome-wide imputation in single-cell expression profiles. In contrast, CIDR [8] uses implicit imputation to mitigate the effects of dropout events. Multi-kernel spectral clustering approaches, including SIMLR [9] and MPSSC [10], utilize multiple kernel functions for robust similarity measure learning; MPSSC is noted for its effectiveness in handling scRNA-seq data sparsity. Despite these advancements, scRNA-seq data still face challenges like high-sparsity and frequent dropout events, where many genes are undetected or incorrectly detected, leading to blurred cell boundaries. Additionally, advances in sequencing technologies have increased data volume and dimensionality, which constrain existing methods' capability to accurately capture intercellular similarities in high-dimensional space [3].

With the growing focus on cellular heterogeneity, deep learning has become a central technique in scRNA-seq clustering. Techniques like DCA leverage zero-inflated negative binomial (ZINB) autoencoders to model scRNA-seq data distribution, while scDeepCluster [11] integrates ZINB-based autoencoders with deep-embedded clustering to optimize latent feature learning and improve clustering performance. AttentionAE-sc [12] employs attention mechanisms to combine denoised representation learning with clustering-friendly feature extraction. Despite these advancements, many methods overlook intercellular structural information [13]. To address this, algorithms such as Louvain and Leiden [14] utilize graph structures to better capture cellular similarities. Approaches like scGAE [15] and graph-sc [16] employ graph autoencoders to embed data while preserving topology. Similarly, scGNN [17] merges graph neural networks (GNNs) with multimodal autoencoders to aggregate cellular relationships and model gene expression patterns using a left-truncated Gaussian mixture model. Likewise, scDSC [18] integrates a ZINB-based autoencoder with a GNN module, using a mutual supervision strategy to optimize data representation.

Although GNN-based methods have advanced general graph encoding [19], they face specific challenges when applied to scRNA-seq data: (i) **Dependence on hard graph constructions:** GNN-based methods typically rely on hard graph constructions, where edges between nodes (cells) are defined using strict criteria or thresholds, i.e., edges are represented as binary (0/1) connections [20–23]. This binary approach oversimplifies the complexity of cellular similarities by disregarding transitional states between cells. (ii) **Similar representations for nearby nodes:** GNN-based methods rely heavily on the underlying graph structure when processing data, aggregating information from neighboring nodes through inherent message-passing mechanisms to generate similar representations [24–27]. However, hard graphs constructed from high-dimensional and high-sparse scRNA-seq data often exhibit significant inter-cluster connections, potentially resulting in erroneous information propagation between cells that should belong to distinct clusters.

To tackle these challenges, we propose a **Soft Graph Clustering** for single-cell RNA sequencing data, namely **scSGC**, which aims to leverage soft graph construction to more accurately capture the continuous similarities between cells through non-binary edge weights. By overcoming the limitations of rigid binary structures, scSGC facilitates improved identification of distinct cellular subtypes and clearer delineation of cell populations, thereby advancing our understanding of cellular heterogeneity. In scSGC, we first utilize a ZINB autoencoder to handle the sparsity and dropout issues inherent in scRNA-seq data, generating robust cellular representations. Then, two soft graphs are constructed using the input data, and their corresponding laplacian matrices are computed [23, 28, 29]. These matrices undergo a minimum jointly normalized cut through a graph-cut strategy to optimize the representation of cell-cell relationships [30]. Finally, an optimal transport-based self-supervised learning approach is employed to refine the clustering, ensuring accurate partitioning of cell populations in high-dimensional and high-sparse data [31]. Experimental results across ten datasets demonstrate that **scSGC** outperforms eleven state-of-the-art (SOTA) single-cell clustering models and exhibits competitive performance across various downstream tasks, highlighting its robustness and effectiveness in diverse analytical scenarios.

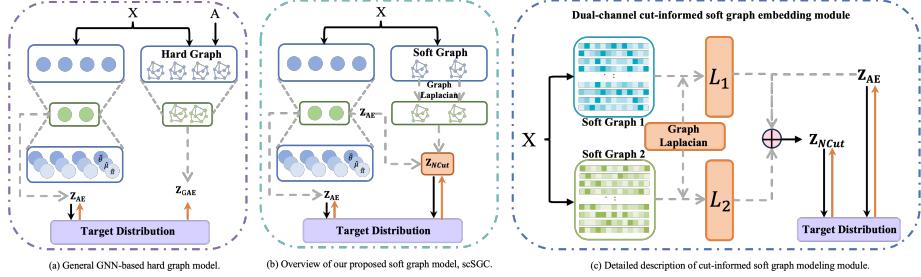


Fig. 1: Comparison for diagram of graph-based scRNA-seq data clustering framework and detailed description of cut-informed soft graph modeling module.

2 Methods

2.1 The framework of scSGC

Fig. 1(a) depicts a hard graph GNN-based framework for scRNA-seq clustering, while Fig. 1(b) illustrates the framework of our proposed method, **scSGC**. Unlike traditional methods, **scSGC** provides two key advantages: (i) It tightly integrates two key modules within the graph-based scRNA-seq clustering framework, i.e., the feature autoencoder and the graph autoencoder, allowing both modules to optimize the final embedding collaboratively; (ii) By employing a soft graph construction strategy, it eliminates reliance on hard graph structures, enabling more effective capture of intracellular structural information and fully utilizing continuous similarities between cells. Specifically, our proposed method, scSGC, comprises three key modules: (i) **ZINB-based feature autoencoder**, which employs a ZINB model to characterize scRNA-seq data specifically to address high sparsity and dropout rates in gene expression data; (ii) **Cut-informed soft graph modeling** (see Fig. 1(c) for its architecture), leverages dual-channel cut-informed soft graph construction to generate consistent and optimized cellular representations, facilitating smoother capture of intercellular continuous structural information; (iii) **Optimization via optimal transport**, utilizing optimal transport theory, achieving optimal partitioning of cell populations at minimal transport cost and ensuring stable clustering results within complex data structures.

2.2 Methodology

To begin, denote the preprocessed input data as \mathbf{X} , where x_{ij} ($1 \leq i \leq N, 1 \leq j \leq D$) represents the expression value of j -th gene for the i -th cell in the dataset. In this study, we aim to simultaneously learn the feature representation \mathbf{Z} for each cell and the clustering assignment \mathbf{C} .

2.2.1 ZINB-based feature autoencoder

To accurately model the distribution of scRNA-seq data and address the challenges posed by its high sparsity and frequent dropout events, we introduce a ZINB-based autoencoder, which utilizes the log-likelihood function of the ZINB distribution as its

loss function.

$$\text{NB}(\mathbf{X}_{ij}; \mu_{ij}, \theta_{ij}) = \frac{\Gamma(\mathbf{X}_{ij} + \theta_{ij})}{\Gamma(\theta_{ij})} \left(\frac{\theta_{ij}}{\theta_{ij} + \mu_{ij}} \right)^{\theta_{ij}} \left(\frac{\mu_{ij}}{\theta_{ij} + \mu_{ij}} \right)^{\mathbf{X}_{ij}}, \quad (1)$$

$$\text{ZINB}(\mathbf{X}_{ij}; \pi_{ij}, \mu_{ij}, \theta_{ij}) = \pi_{ij} \delta_0(\mathbf{X}_{ij}) + (1 - \pi_{ij}) \text{NB}(\mathbf{X}_{ij}; \mu, \theta), \quad (2)$$

where dropout (π_{ij}) is the probability of dropout events, mean (μ_{ij}) and dispersion (θ_{ij}) parameters, respectively, represent the negative binomial component.

Then, we define the encoder function as $\mathbf{Z} = f_e(\mathbf{X})$ and the decoder function as $\mathbf{X}' = f_d(\mathbf{Z})$, where both the encoder and decoder are implemented as fully connected neural networks. The decoder has three output layers to estimate parameters π , μ and θ , which are

$$\begin{aligned} \hat{\pi} &= \text{sigmoid}(W_\pi \cdot \mathbf{Z}), \\ \hat{\mu} &= \text{diag}(s_i) \times \exp(W_\mu \cdot \mathbf{Z}), \\ \hat{\theta} &= \exp(W_\theta \cdot \mathbf{Z}). \end{aligned} \quad (3)$$

Similarly, W_π , W_μ and W_θ are the corresponding weights, the size factors s_i is the ratio of the total cell count to the median S . Finally, we minimize the overall negative likelihood loss of the ZINB model-based autoencoder,

$$\mathcal{L}_{ZINB} = -\log \left(\text{ZINB} \left(\mathbf{X}_{ij} | \hat{\pi}_{ij}, \hat{\mu}_{ij}, \hat{\theta}_{ij} \right) \right) \quad (4)$$

2.2.2 Cut-informed soft graph modeling

We propose the dual-channel cut-informed soft graph embedding module to capture the continuous similarities between cells from high-dimensional and high-sparse scRNA-seq data. As depicted in Fig. 1(c), this module constructs dual-channel soft graphs and computes the corresponding laplacian matrices, then applies a minimum jointly normalized cut strategy to fuse the matrices from both channels.

Initially, using the preprocessed scRNA-seq data matrix \mathbf{X} as input, we adopt a dual-channel strategy to construct two distinct similarity soft graphs: the feature similarity graph \mathcal{G}_1 and the cosine similarity graph \mathcal{G}_2 . In both graphs, \mathcal{V} denotes the set of cell nodes, and \mathcal{E} represents the edges indicating similarity between cells. Notably, the adjacency matrix of \mathcal{G}_1 , \mathbf{A}_1 , is computed via the inner product of \mathbf{X} , capturing feature-space similarity between cells, formulated as $\mathbf{A}_1 = \mathbf{X} \cdot \mathbf{X}^T$. Concurrently, the adjacency matrix of \mathcal{G}_2 , \mathbf{A}_2 , is constructed based on cosine similarity, which is computed for each pair of cells x_i and x_j using $\cos(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|}$, representing the angular similarity between the vectors of gene expression profiles of the two cells. Subsequently, additional matrix multiplication is applied to emphasize the similarity information and capture the directional differences in gene expression patterns between cells, defined as:

$$A_2 = |\cos(\mathbf{X}, \mathbf{X})| \cdot |\cos(\mathbf{X}, \mathbf{X})|^T. \quad (5)$$

For graphs \mathcal{G}_1 and \mathcal{G}_2 , \mathbf{D}_1 , \mathbf{D}_2 are their diagonal matrices respectively. This dual-channel graph construction enables the effective capture of both feature-based and

angular similarities between cells, improving the model's robustness to high-dimension, high-sparsity, and dropout events inherent in scRNA-seq data, thus facilitating more accurate cellular representation and downstream clustering tasks.

To effectively integrate continuous cell similarities within the dual-channel soft graph, we employ a strategy that minimizes the Joint Normalized Cut (Joint NCut). This approach involves calculating the normalized laplacian matrices, \mathbf{L}_1 and \mathbf{L}_2 , for the two soft graphs, which together capture continuous structural information to identify optimal partitioning points in high-dimensional space:

$$\begin{aligned}\mathbf{L}_1 &= \mathbf{I} - \mathbf{D}_1^{-1/2} \mathbf{A}_1 \mathbf{D}_1^{-1/2}, \\ \mathbf{L}_2 &= \mathbf{I} - \mathbf{D}_2^{-1/2} \mathbf{A}_2 \mathbf{D}_2^{-1/2}.\end{aligned}\quad (6)$$

To fuse the similarity information from both soft graphs, we minimize the following Joint NCut objective:

$$\text{Joint NCut}(\mathcal{V}) = \frac{1}{2} \sum_{i=1}^k \left(\frac{\text{cut}(\mathcal{V}_i, \mathcal{V} \setminus \mathcal{V}_i; \mathbf{L}_1)}{\text{vol}(\mathcal{V}_i; \mathbf{A}_1)} + \frac{\text{cut}(\mathcal{V}_i, \mathcal{V} \setminus \mathcal{V}_i; \mathbf{L}_2)}{\text{vol}(\mathcal{V}_i; \mathbf{A}_2)} \right), \quad (7)$$

\mathcal{V}_i refers to the set of nodes in cluster i . The term $\text{cut}(\mathcal{V}_i, \mathcal{V} \setminus \mathcal{V}_i)$ represents the weight of edges cut between the cluster \mathcal{V}_i and the rest of the graph, while $\text{vol}(\mathcal{V}_i)$ is the volume of cluster \mathcal{V}_i . By jointly optimizing normalized cuts for both graphs, we achieve a consistent and optimized cellular representation. This strategy integrates continuous structural information from multiple similarity measures, enhancing the model's robustness and accuracy in handling high-dimensional, sparse scRNA-seq data.

Next, we formulate the objective function to minimize the Joint NCut, namely:

$$\arg \min_{\mathbf{Z}} \text{Tr}(\mathbf{Z}^T (\alpha \mathbf{L}_1 + (1 - \alpha) \mathbf{L}_2) \mathbf{Z}) \quad \text{subject to } \mathbf{Z}^T \mathbf{Z} = \mathbf{I}, \quad (8)$$

where α is a weight parameter that balances the contributions of the two graphs to the optimization outcome. To ensure orthogonality in the cellular representations, we incorporate a regularization term, resulting in our loss function:

$$\mathcal{L}_{\text{NCut}} = \text{Tr}(\mathbf{Z}^T (\alpha \mathbf{L}_1 + (1 - \alpha) \mathbf{L}_2) \mathbf{Z}) + \beta \|\mathbf{Z}^T \mathbf{Z} - \mathbf{I}\|_F. \quad (9)$$

Here, β is a regularization parameter and $\|\cdot\|_F$ denotes the Frobenius norm.

Note that our method promotes the clustering of scRNA-seq data from multiple perspectives. First, the dual-channel soft graph construction effectively captures the continuous similarities between cells, enhancing robustness against high-dimension and high-sparsity. Second, the Joint NCut strategy integrates cellular relationships across similarity measures to improve clustering accuracy, while the optimized loss function ensures consistency of the learned embeddings across diverse graph structures, further enhancing performance.

2.2.3 Optimization via optimal transport

To ensure stable clustering results in complex data structures, we propose an optimal transport-based clustering optimization module that optimizes clustering assignments by minimizing the transportation cost. The objective is to minimize the cost of transferring the probability mass from an initial clustering assignment, represented by matrix $Q = [q_{ij}]$, to an optimal target distribution $P = [p_{ij}]$. Here, q_{ij} denotes the probability of assigning cell i to cluster j , and P represents the optimal transport plan. The transport optimization problem is formulated as minimizing the divergence between Q and P , constrained by the requirement that the cluster assignments respect the underlying data distribution. Specifically, we aim to minimize the cost matrix $-\log Q$, which quantifies the divergence between the observed data and the cluster centers while ensuring a balanced distribution of cluster assignments:

$$\begin{aligned} \min_P \quad & -P * (\log Q) \\ \text{s.t.} \quad & P \in \mathbb{R}_+^{N \times C}, \\ & P \mathbf{1}_C = \mathbf{1}_N \text{ and } P^T \mathbf{1}_N = N\pi, \end{aligned} \tag{10}$$

here, $\mathbf{1}_C$ and $\mathbf{1}_N$ are vectors of ones, and π represents the proportions of points in each cluster, which can be estimated from intermediate clustering results. The optimization ensures that the target distribution P aligns with the proportions π , preventing degenerate solutions where the clustering assignments become unbalanced.

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

where $H(P)$ is the entropy of the transport plan P , and λ balances the accuracy of the assignment and the smoothness of the clustering. This optimization problem is solved through Sinkhorn's fixed-point iteration method, which updates the transport matrix iteratively as follows:

$$\hat{P}^{(t)} = \text{diag}(\mathbf{u}^{(t)}) Q^\lambda \text{diag}(\mathbf{v}^{(t)}), \tag{12}$$

where $\mathbf{u}^{(t)} = \mathbf{1}_N / (Q^\lambda \mathbf{v}^{(t-1)})$ and $\mathbf{v} = N\pi / (Q^\lambda \mathbf{u}^{(t)})$. After several iterations, we obtain the optimal transport matrix \hat{P} .

Once \hat{P} is fixed, we align Q with \hat{P} , ensuring that the clustering assignment is consistent with the optimal transport plan. The clustering loss is defined as the Kullback-Leibler (KL) divergence between the estimated transport matrix \hat{P} and the target distribution Q :

$$\mathcal{L}_{\text{KL}} = \text{KL}(\hat{P} \| Q). \tag{13}$$

Overall, this module improves clustering accuracy by balancing the assignments across clusters and maintaining consistency with the data structure. It enhances scSGC's ability to handle high-dimensional, high-sparse scRNA-seq data, ensuring more stable and reliable clustering results.

Table 1: Summary of the scRNA-seq datasets.

Datasets	Sequencing platform	Sequencing method	Sample size	No. of genes	No. of groups	Cluster sizes	Sparsity rate(%)
Mouse Pancreas cells 1 [32]	Illumina HiSeq 2500	CEL-seq	822	14878	13	(343,29,55,3,29,72,1,9,17,4,2,4)	90.48
Mouse Pancreas cells 2 [32]	Illumina HiSeq 2500	CEL-seq	1064	14878	13	(551,39,133,182,27,67,8,19,4,18,3,3)	87.81
Human Pancreas cells 2 [32]	Illumina HiSeq 2500	CEL-seq	1724	20125	14	(125,676,81,301,371,17,22,86,23,2,6,9,3,2)	90.59
Human Pancreas cells 1 [32]	Illumina HiSeq 2500	CEL-seq	1937	20125	14	(110,872,21,4,51,120,236,13,70,130,92,14,5,8,2)	90.45
Muraro Human Pancreas cells [33]	Illumina NextSeq 500	CEL-seq2	2122	19046	9	(21,812,448,193,245,21,3,10,10,80)	73.02
Human Pancreas cells 3 [32]	Illumina HiSeq 2500	CEL-seq	3605	20125	14	(843,161,376,130,75,97,100,5,36,14,7,2,2,1)	91.30
CITE-CMBC [34]	Illumina NextSeq 500	10X Genomics	8617	2000	15	(1791,2293,1248,1404,100,5,36,14,7,2,2,1)	93.26
Human Liver cells [35]	Illumina HiSeq 2500	10X Genomics	8344	4999	11	(844,119,110,60,5,88,59,50,129,37,51,93)	90.77
Talabia Muris Limb Muscle cells [36]	Illumina NovaSeq 6000	Smart-seq2	3855	21069	6	(1833,935,453,31,1187,190)	91.28
Talabia Sapiens Liver cells [36]	Illumina NovaSeq 6000	Smart-seq2	2152	61759	15	(741,323,177,172,129,98,81,77,74,74,66,56,33,29,22)	95.42

2.2.4 Joint optimiztion

The overall optimization objective of scSGC combines three main components: NCut loss, ZINB loss, and clustering loss, formulated as:

$$\mathcal{L} = \mathcal{L}_{NCut} + \gamma \mathcal{L}_{ZINB} + \mu \mathcal{L}_{KL}, \quad (14)$$

where the coefficients γ and μ serve as balancing factors. Each term plays a crucial role in fine-tuning scSGC's performance.

2.3 Datasets

We evaluated the performance of scSGC on 10 scRNA-seq datasets comprising both human and mouse samples, covering diverse cell types such as pancreas, liver, and muscle cells. These datasets were generated using various sequencing platforms including Illumina HiSeq 2500, NextSeq 500, and NovaSeq 6000, and employed sequencing methods such as CEL-seq, 10X Genomics, and Smart-seq2. The number of cells per dataset ranges from 822 to 8617, with gene counts varying between 2,000 and 61,759. The datasets include between 6 and 15 annotated cell groups, and exhibit sparsity rates ranging from 73.02% to 95.42%. Detailed information for each dataset is summarized in Table1.

2.4 Baselines

To highlight the advantages of our method, we compared it with 11 prominent existing approaches, including traditional clustering methods, deep learning-based clustering algorithms, and deep structured clustering approaches. Among the **traditional methods**, **pcaReduce** [37] performs clustering by iteratively combining probability density functions, and **SUSSC** [38] uses stochastic gradient descent and nearest neighbor search for clustering.

In the realm of **deep learning clustering**, we considered several foundational models. **DEC** utilizes deep embedded clustering, employing neural networks to simultaneously learn feature representations and cluster assignments [39]. **contrastive-sc** adapts self-supervised contrastive learning to scRNA-seq data, decoupling embedding learning and clustering for flexible and robust cell type identification [40]. The method using contrastive-sc embeddings followed by K-Means clustering is referred to as **contrastive-sc K-means**, while that using Leiden clustering is referred to as **contrastive-sc Leiden**. **scDeepCluster** leverages the ZINB model to simulate the

distribution of scRNA-seq data while concurrently learning feature representations and clustering [11]. **scDSSC** enhances clustering performance through explicit modeling of scRNA-seq data generation [41]. Additionally, **scNAME** integrates mask estimation and neighborhood contrastive learning to achieve denoising and clustering [42], while **AttentionAE-sc** employs attention mechanisms to enhance representation learning [12]. **seziDesk** introduces a denoising autoencoder and optimizes the clustering results in the latent space through a soft self-training K-means algorithm [43].

For **deep structured clustering methods**, we compared scGNN [17], scDSC [18], and scCDCG [30]. **scGNN** utilizes GNNs and multi-modal autoencoders to capture intercellular relationships and processes heterogeneous gene expression patterns through a left-truncated mixture Gaussian model [17]. **scDSC** combines a ZINB model with GNN modules for data representation learning and uses a mutual supervision strategy to unify these architectures [18]. Lastly, **scCDCG** effectively handles high-dimensional and high-sparsity scRNA-seq data by leveraging higher-order structural information, demonstrating exceptional efficiency [30].

3 Results

3.1 Experimental Setup

3.1.1 Dataset Preprocessing

Raw scRNA-seq data were preprocessed using the SCANPY package [44], resulting in the processed dataset \mathbf{X} as input for scSGC. Initially, genes expressed in fewer than three cells were filtered out, and genes with zero counts across all cells were removed. The remaining data were then normalized and log-transformed using TPM. Finally, the top n most highly expressed genes were selected from the remaining genes to construct the preprocessed dataset, with the specific value of n for each dataset outlined in Table 2.

For baseline methods, we adhered to the preprocessing steps specified in their respective publications to ensure consistency with recommended practices and allow for fair comparison while preserving the integrity of each method's design.

3.1.2 Evaluation Metrics

To assess the efficacy of the proposed method, we evaluate its clustering performance using three widely recognized metrics from public domain: Accuracy (ACC), Normalized Mutual Information (NMI) [45], and Adjusted Rand Index (ARI) [46]. Higher values for these metrics indicate better clustering results.

3.1.3 Training Procedure

The training process of scSGC includes two steps: (i) we pre-train scSGC with both NCut loss and ZINB loss by 200 epochs; (ii) under the guidance of Eq. 14, we train the whole network with 200 epochs. As the number of cell clusters k is unknown in real applications, we apply K-means to the embeddings to obtain the optimal value of k .

Table 2: Hyperparameter settings for different datasets. The symbol ‘–’ indicates that this operation was not performed for the respective dataset.

Datasets	α	β	γ	Weight decay	No. of highly expression genes
Mouse Pancreas cells 2	0.5	20	25	1e-3	1500
Mouse Pancreas cells 1	0.7	25	20	5e-3	1500
Human Pancreas cells 2	0.7	10	25	5e-4	2000
Human Pancreas cells 1	0.7	30	25	5e-4	2000
Mauro Human Pancreas cells	0.5	10	100	5e-3	2000
Human Pancreas cells 3	0.1	30	25	5e-3	2000
CITE-CMBC	0.7	10	50	5e-3	1500
Human Liver cells	0.7	15	25	5e-3	1500
Tabula Muris Limb Muscle cells	0.75	20	25	1e-4	–
Tabula Spaeis Liver cells	0.5	20	100	5e-3	8000

3.1.4 Parameters Setting

In practice, we implement scSGC in Python 3.7 based on PyTorch. The embedding size d is fixed at 16 for all datasets. Furthermore, we conduct meticulous parameter tuning across different datasets to ensure optimal model performance. The specific parameter settings are as follows: the model training is optimized using the Adam optimizer with a uniformly set learning rate of 1e-3 across all datasets and supplemented by a weight decay strategy to further enhance scSGC’s generalization capability, ensuring stable convergence; for all datasets, the hyperparameters λ , θ and μ are initially set to 5, 1, and 1e-3, respectively; for additional details on the other hyperparameter settings, please see Table 2.

For baseline methods, we strictly followed the parameter settings provided in their respective publications. In cases where parameter settings were not clearly specified, we conducted preliminary experiments to determine the optimal values, ensuring each tool operated under its best conditions.

To ensure the accuracy of the experimental results, we repeat each experiment 10 times with varying random seeds across all datasets, reporting the mean and variance of the outcomes.

3.2 Overall Performance

3.2.1 Quantitative Analysis

Table 3 presents a comparative analysis of clustering performance across ten benchmark datasets, evaluating different models based on ACC, NMI, and ARI. The results highlight three key observations: (i) scSGC outperforms all other models across all three metrics, demonstrating its effectiveness in tackling scRNA-seq clustering challenges. On average, scSGC exhibits improvements of 4.06%, 3.725%, and 4.03% in ACC, NMI, and ARI, respectively, when compared to the second-best approach, highlighting its comprehensive advantage across multiple evaluation dimensions. (ii) Deep

Table 3: Clustering performances of all models on 10 benchmark datasets (mean \pm standard). The **bold** values represent the best results, and underline values represent the runner-up results.

Datasets	Metric	pcaReduce	SUSSC	DEC	contrastive -sc	contrastive -sc	scDeep- Cluster	scDSSC	scNAME	sczIDesk	Attention -AE	scDSC	scGNN	scCDCG	scSGC	
Mouse Pancreas cells 1	ACC	42.50 \pm 0.1	40.00 \pm 0.8	47.79 \pm 0.7	57.58 \pm 4.2	57.53 \pm 4.7	65.18 \pm 0.7	71.51 \pm 1.3	75.05 \pm 5.5	78.73 \pm 4.3	81.70 \pm 4.2	69.91 \pm 3.8	47.93 \pm 4.9	<u>\$2.64 \pm 0.1\$</u>	91.24 \pm 1.0	
	NMI	47.19 \pm 0.1	61.61 \pm 0.1	55.31 \pm 1.9	65.62 \pm 2.1	57.20 \pm 14.0	65.92 \pm 0.9	75.63 \pm 4.6	72.25 \pm 1.2	73.00 \pm 7.5	<u>79.92 \pm 3.0</u>	62.04 \pm 5.4	58.77 \pm 1.7	<u>77.66 \pm 2.0</u>	<u>85.78 \pm 0.5</u>	
	ARI	18.32 \pm 0.1	30.73 \pm 1.0	33.04 \pm 2.3	47.01 \pm 4.3	55.95 \pm 9.3	55.47 \pm 2.2	62.88 \pm 1.3	64.28 \pm 6.9	75.24 \pm 7.5	69.94 \pm 5.6	64.78 \pm 8.4	36.33 \pm 5.5	<u>\$0.60 \pm 5.8\$</u>	91.63 \pm 1.0	
Mouse Pancreas cells 2	ACC	28.97 \pm 0.1	36.41 \pm 0.1	34.19 \pm 2.3	59.55 \pm 5.7	47.01 \pm 3.6	69.91 \pm 2.2	89.11 \pm 1.4	82.36 \pm 8.0	85.71 \pm 2.0	81.92 \pm 5.6	77.35 \pm 2.3	43.03 \pm 3.2	<u>93.97 \pm 0.2</u>	94.45 \pm 0.2	
	NMI	48.93 \pm 0.1	60.31 \pm 0.2	54.08 \pm 1.1	62.34 \pm 1.1	53.42 \pm 1.0	59.58 \pm 2.2	83.75 \pm 1.7	78.00 \pm 3.9	79.91 \pm 2.1	81.88 \pm 1.2	69.50 \pm 2.0	55.87 \pm 1.9	88.02 \pm 0.2	<u>88.45 \pm 0.9</u>	
	ARI	12.80 \pm 0.1	24.90 \pm 0.1	18.54 \pm 1.1	49.75 \pm 8.5	37.85 \pm 1.3	60.14 \pm 3.6	89.66 \pm 0.8	76.56 \pm 11.4	81.19 \pm 4.3	82.14 \pm 7.8	65.81 \pm 1.4	26.40 \pm 2.9	<u>92.61 \pm 0.3</u>	92.64 \pm 1.0	
Human Pancreas cells 2	ACC	37.56 \pm 0.1	46.75 \pm 0.2	45.26 \pm 1.8	60.01 \pm 5.7	50.31 \pm 2.0	62.88 \pm 5.5	80.02 \pm 9.4	63.33 \pm 2.0	72.72 \pm 6.2	81.80 \pm 12.1	80.40 \pm 4.0	57.62 \pm 1.1	<u>84.66 \pm 1.1</u>	91.77 \pm 0.5	
	NMI	50.81 \pm 0.1	68.62 \pm 0.1	66.32 \pm 2.1	62.25 \pm 2.0	51.99 \pm 11.2	73.97 \pm 2.2	75.34 \pm 5.5	58.05 \pm 1.6	76.42 \pm 3.9	82.16 \pm 10.0	79.44 \pm 1.7	79.32 \pm 2.2	<u>83.44 \pm 2.6</u>	91.21 \pm 0.5	
	ARI	19.23 \pm 0.5	38.45 \pm 0.4	39.14 \pm 1.8	47.10 \pm 5.3	47.12 \pm 13.2	64.56 \pm 7.5	67.06 \pm 14.4	32.12 \pm 1.5	57.07 \pm 9.7	73.20 \pm 16.5	82.85 \pm 6.5	79.96 \pm 1.6	85.30 \pm 1.6	<u>92.62 \pm 0.4</u>	
Human Pancreas cells 1	ACC	23.63 \pm 0.2	40.65 \pm 0.1	40.64 \pm 1.9	69.67 \pm 7.8	60.68 \pm 2.8	66.39 \pm 3.8	79.29 \pm 9.6	84.20 \pm 6.0	86.60 \pm 7.0	81.56 \pm 1.8	73.10 \pm 2.3	55.10 \pm 2.1	<u>92.15 \pm 0.8</u>	96.25 \pm 0.8	
	NMI	42.58 \pm 0.1	67.95 \pm 0.1	62.35 \pm 2.3	69.21 \pm 2.1	65.61 \pm 2.3	69.42 \pm 12.5	83.84 \pm 6.4	65.74 \pm 4.0	84.99 \pm 2.5	82.62 \pm 9.4	60.50 \pm 2.0	64.03 \pm 2.1	<u>86.35 \pm 0.0</u>	91.42 \pm 0.9	
	ARI	1.98 \pm 0.5	31.16 \pm 0.1	26.07 \pm 1.3	52.00 \pm 8.7	37.11 \pm 4.0	47.92 \pm 3.0	72.07 \pm 14.4	63.31 \pm 7.4	78.16 \pm 11.6	71.84 \pm 1.8	69.81 \pm 1.6	38.32 \pm 0.0	<u>92.83 \pm 0.0</u>	94.89 \pm 0.2	
Mauro Human Pancreas cells	ACC	50.56 \pm 0.1	76.26 \pm 0.1	72.22 \pm 5.5	81.85 \pm 4.6	67.22 \pm 4.2	74.70 \pm 2.7	80.10 \pm 4.8	80.36 \pm 7.9	91.53 \pm 8.9	93.84 \pm 2.7	70.42 \pm 1.4	79.32 \pm 4.0	92.05 \pm 1.5	96.03 \pm 0.2	
	NMI	54.83 \pm 1.7	82.25 \pm 0.0	76.29 \pm 2.7	77.73 \pm 1.5	73.31 \pm 1.7	79.32 \pm 6.0	82.16 \pm 4.7	79.12 \pm 2.7	84.49 \pm 4.6	87.80 \pm 2.3	75.89 \pm 6.6	87.76 \pm 4.6	86.81 \pm 1.0	<u>90.23 \pm 1.7</u>	
	ARI	36.61 \pm 2.3	64.23 \pm 0.1	61.76 \pm 2.6	75.22 \pm 10.1	60.04 \pm 4.9	64.59 \pm 2.5	85.76 \pm 7.3	76.64 \pm 10.4	88.03 \pm 11.8	90.74 \pm 4.1	73.42 \pm 12.2	78.87 \pm 3.2	<u>91.37 \pm 1.2</u>	92.39 \pm 1.9	
Human Pancreas cells 3	ACC	35.92 \pm 0.2	45.16 \pm 0.1	43.56 \pm 15	57.47 \pm 5.0	52.77 \pm 3.6	73.65 \pm 3.3	71.61 \pm 9.3	83.20 \pm 8.7	84.13 \pm 9.3	90.96 \pm 2.4	80.12 \pm 10.0	67.94 \pm 4.6	88.04 \pm 0.3	94.80 \pm 0.8	
	NMI	51.22 \pm 0.0	69.50 \pm 0.1	64.51 \pm 0.3	66.84 \pm 2.3	67.39 \pm 2.5	75.02 \pm 1.2	75.77 \pm 6.2	80.56 \pm 5.6	82.02 \pm 6.6	87.00 \pm 1.8	75.87 \pm 8.4	62.60 \pm 1.6	82.21 \pm 1.1	<u>88.81 \pm 0.5</u>	
	ARI	26.40 \pm 0.4	43.53 \pm 0.1	42.73 \pm 1.0	47.9 \pm 3.4	46.15 \pm 3.9	63.31 \pm 2.3	64.86 \pm 13.4	81.08 \pm 5.7	72.77 \pm 10.0	89.78 \pm 2.4	82.85 \pm 6.5	58.41 \pm 1.9	<u>90.78 \pm 0.8</u>	93.58 \pm 0.1	
CITE-CMBC	ACC	28.19 \pm 0.1	47.22 \pm 0.1	41.88 \pm 1.4	52.70 \pm 5.0	49.84 \pm 5.7	70.80 \pm 2.5	63.14 \pm 8.0	68.06 \pm 13.2	80.12 \pm 7.9	60.79 \pm 6.7	66.71 \pm 4.0	<u>21.15 \pm 1.1</u>	75.89 \pm 0.3		
	NMI	28.36 \pm 0.2	60.14 \pm 0.1	46.87 \pm 2.2	64.83 \pm 6.4	63.57 \pm 8.7	72.53 \pm 6.7	64.04 \pm 9.9	62.02 \pm 9.6	64.21 \pm 3.3	61.70 \pm 4.4	24.22 \pm 1.7	76.18 \pm 8.6			
	ARI	5.10 \pm 0.1	35.56 \pm 1.6	23.35 \pm 0.9	47.88 \pm 8.7	52.27 \pm 8.8	56.23 \pm 4.3	55.28 \pm 10.6	58.99 \pm 21.0	56.38 \pm 8.8	44.20 \pm 11.6	53.51 \pm 2.2	60.25 \pm 1.1	<u>61.46 \pm 1.1</u>	64.47 \pm 0.9	
Human Liver cells	ACC	34.92 \pm 0.2	46.62 \pm 0.1	46.32 \pm 5.5	70.31 \pm 2.1	47.18 \pm 6.4	63.44 \pm 6.7	70.60 \pm 7.1	74.55 \pm 5.3	74.01 \pm 3.4	81.34 \pm 3.4	53.32 \pm 4.9	53.35 \pm 11.7	49.02 \pm 2.0	<u>60.21 \pm 4.0</u>	82.25 \pm 7.3
	NMI	32.07 \pm 0.6	67.32 \pm 0.0	52.24 \pm 5.0	85.11 \pm 6.1	75.48 \pm 1.4	74.60 \pm 2.0	78.94 \pm 11.3	77.42 \pm 12.0	73.59 \pm 3.7	82.86 \pm 4.5	71.63 \pm 2.8	62.33 \pm 2.1	79.34 \pm 2.6	<u>89.17 \pm 0.6</u>	
	ARI	6.34 \pm 14.4	35.12 \pm 0.8	31.04 \pm 4.7	80.96 \pm 6.1	35.18 \pm 4.1	58.58 \pm 10.1	73.11 \pm 11.8	79.03 \pm 2.9	77.44 \pm 12.9	79.16 \pm 11.8	73.34 \pm 2.8	85.41 \pm 1.3	81.26 \pm 2.7	<u>93.05 \pm 0.5</u>	
Tabula Muris Limb Muscle cells	ACC	29.19 \pm 1.1	29.71 \pm 2.4	50.70 \pm 7.2	46.49 \pm 1.3	32.37 \pm 4.5	59.57 \pm 4.3	59.47 \pm 4.7	61.34 \pm 3.4	53.31 \pm 4.9	53.35 \pm 11.7	49.02 \pm 5.2	<u>60.21 \pm 4.0</u>	82.25 \pm 7.3	<u>68.68 \pm 2.3</u>	
	NMI	19.26 \pm 1.2	8.6 \pm 0.9	54.40 \pm 2.1	31.71 \pm 1.5	33.12 \pm 0.9	34.55 \pm 5.5	40.88 \pm 6.04	<u>63.23 \pm 1.4</u>	36.66 \pm 6.1	12.55 \pm 3.8	22.11 \pm 5.2	61.70 \pm 3.1	56.54 \pm 8.3	<u>71.23 \pm 4.7</u>	
	ARI	15.20 \pm 1.1	5.94 \pm 1.2	21.26 \pm 2.1	18.75 \pm 2.1	35.93 \pm 3.4	42.97 \pm 5.7	54.70 \pm 2.8	32.59 \pm 1.1	13.64 \pm 7.5	23.90 \pm 3.5	<u>60.25 \pm 1.8</u>	53.37 \pm 9.1	<u>69.01 \pm 1.9</u>		
Tabula Spates Liver cells	ACC	31.23 \pm 0.3	41.25 \pm 6.5	63.76 \pm 3.6	45.50 \pm 3.3	46.93 \pm 3.5	49.08 \pm 2.3	47.49 \pm 4.7	63.33 \pm 1.9	66.68 \pm 3.3	53.28 \pm 12.5	<u>73.83 \pm 3.4</u>	68.65 \pm 3.2	48.86 \pm 1.6	73.90 \pm 0.6	
	NMI	41.34 \pm 4.7	57.63 \pm 5.3	71.92 \pm 21	48.48 \pm 12.8	54.59 \pm 2.0	60.98 \pm 2.5	44.56 \pm 3.4	74.83 \pm 0.8	75.10 \pm 1.3	51.17 \pm 9.4	<u>72.71 \pm 2.2</u>	62.33 \pm 2.1	62.54 \pm 3.6	<u>77.83 \pm 2.0</u>	
	ARI	33.40 \pm 6.7	28.82 \pm 5.4	63.45 \pm 4.5	36.06 \pm 6.8	37.74 \pm 3.4	34.52 \pm 2.2	42.88 \pm 1.7	58.94 \pm 2.8	66.12 \pm 2.3	42.41 \pm 4.2	51.11 \pm 5.0	65.41 \pm 1.3	41.11 \pm 5.0	72.86 \pm 0.9	

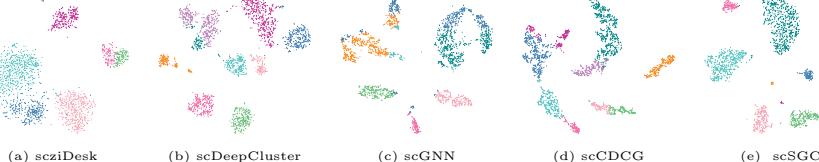


Fig. 2: Visualization of scSGC and four typical baselines on *Mauro Human Pancreas cells* dataset in 2D t-SNE projection. Each point represents a cell, while each color represents a predicted cell type.

learning-based methods, such as Attention-AE, primarily rely on node features while disregarding complex structural relationships, making them ineffective in handling the high sparsity and dropout effects in scRNA-seq data. scSGC addresses this by integrating a ZINB-based feature autoencoder to enhance feature extraction and an optimal transport-based clustering module to refine cluster assignments by minimizing transport cost, leading to more stable and biologically meaningful clustering. (iii) GNN-based methods, which rely on predefined hard graph structures, often struggle with representation collapse and fail to capture the continuous similarities between cells. This limitation arises because hard graphs enforce binary connections, leading to excessive discretization and the loss of nuanced intercellular relationships. In contrast, scSGC introduces a dual-channel soft graph construction with a minimized Joint NCut strategy, effectively capturing continuous cell relationships, reducing rigid graph dependencies, and improving representation learning.

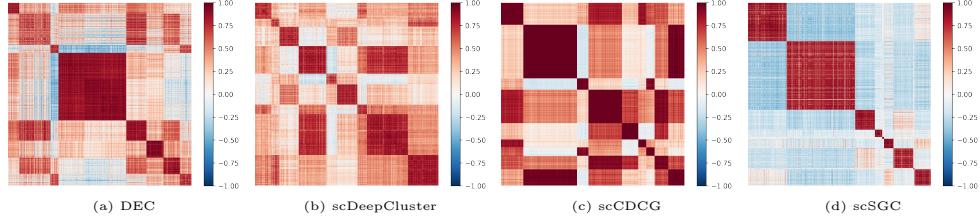


Fig. 3: The heat maps of node similarity matrices in the latent space of DEG, scDeepCluster, scGNN, and our proposed method scSGC on the *Mauro Human Pancreas cells* dataset.

3.2.2 Qualitative Analysis

To intuitively interpret the clustering results from a biological perspective, we use t-SNE to extract and visualize the distribution of clustering embeddings on the *Mauro Human Pancreas cells* dataset. The embeddings, represented by the matrix \mathbf{Z} in a two-dimensional space, are obtained from several classical baseline methods. As shown in Fig. 2, scSGC effectively differentiates cell types by learning compact, well-separated representations that form distinct boundaries between populations, thereby better capturing the underlying clustering structure in scRNA-seq data. In contrast, other methods exhibit suboptimal clustering performance, with noticeable overlap between clusters and more scattered results, failing to group similar cell types effectively.

3.2.3 Similarity Analysis

Taking the *Mauro Human Pancreas cells* dataset as an example, we extract representation matrices from scSGC and three representative methods, construct their element similarity matrices, and visualize the results as heatmaps. As shown in Fig. 3, scSGC effectively captures the three-dimensional clustering structure in the latent space, forming more compact and well-organized clusters, which indicates its advantage in preserving continuous cell similarities and revealing the underlying biological structure. In contrast, the structural information of other methods is more scattered, making it difficult to accurately characterize potential biological features. This comparison further demonstrates the superiority of scSGC in enhancing intercellular similarity representation and maintaining structural integrity.

3.3 Biological Analysis

Following the validation of the clustering performance of scSGC, it is essential to further interpret its biological significance. Clustering single-cell RNA sequencing data enables the identification of gene expression patterns, offering valuable insights into cellular heterogeneity. Such analysis is crucial for understanding the tissue microenvironment and advancing precision medicine applications. In this section, we use the *Mauro Human Pancreas cells* dataset as a case study to perform biological validation based on the clustering labels generated by scSGC. Specifically, we conduct

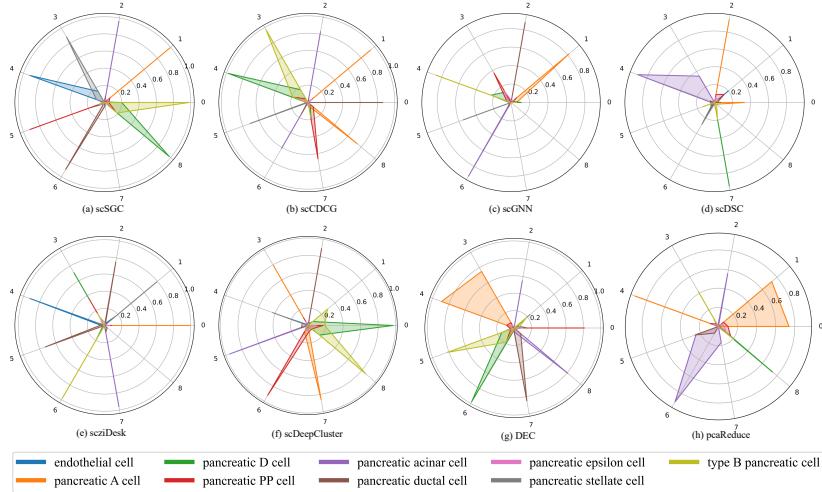


Fig. 4: Cell type annotation. Overlap of top 100 DEGs in clusters detected by ten methods with gold standard cell types (similarity = number of overlapped DEGs/100).

marker gene identification and cell type annotation to clarify the biological relevance of the clustering results.

3.3.1 Cell Type Annotation

Differentially expressed genes (DEGs) serve as a critical indicator for cell type annotation, as they allow the marker gene identification within each cluster based on gene expression matrices and predicted labels. We employ the “FindAllMarkers” function from the Seurat [7] package to identify the DEGs for each cluster. Taking the *Mauro Human Pancreas cells* as an example, we first obtained the top 100 DEGs within gold standard clusters for each annotated cell type, which served as a baseline for evaluating various methods. We apply the same approach to both scSGC and other competing methods, identifying the top 100 marker genes for each cluster and calculating the overlap with the gold standard DEGs, as illustrated in Fig. 4. scSGC accurately assigned cell types to specific clusters, with DEGs overlap exceeding 90% for all clusters except cluster 7. For example, clusters 2, 1, 0, 8, 6, 4, 5, and 3 were annotated as ‘pancreatic acinar cell’, ‘pancreatic A cell’, ‘type B pancreatic cell’, ‘pancreatic D cell’, ‘pancreatic ductal cell’, ‘endothelial cell’, ‘pancreatic PP cell’, and ‘pancreatic stellate cell’, respectively. The corresponding sample sizes for these eight cell types in the gold standard were 219, 812, 448, 193, 245, 21, 101, and 80, while ‘pancreatic epsilon cell’ had only 3 samples. Due to the extremely small sample size of ‘pancreatic epsilon cell’ in the gold standard, scSGC did not provide a clear annotation for this type but successfully identified the other eight cell types, demonstrating high accuracy and reliability.

Fig. 4(b)-(h) presents the cell type annotation results of seven other methods. Specifically, scGNN failed to identify the cell types of clusters 0, 7, and 8, scDeepCluster could not accurately partition cluster 1, and the DEG overlap for cluster 4 was

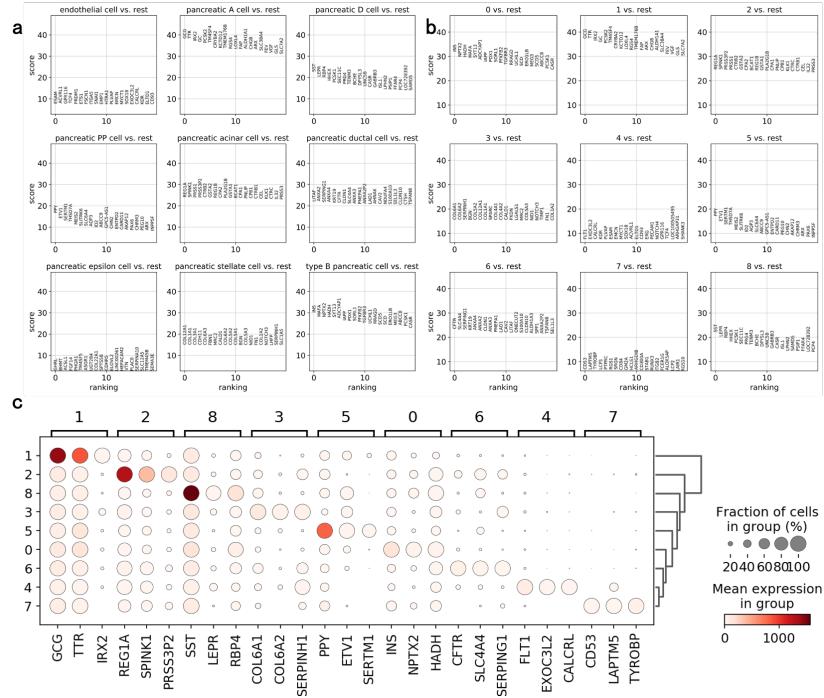


Fig. 5: Marker gene identification. (a) Top 20 DEGs in different cell types by gold standard. (b) Top 20 DEGs in different cell types by scSGC. (c) Dot plot of the top 3 DEGs within each cell type by scSGC.

below 40%. DEC misannotated clusters 3 and 4 as 'pancreatic A cell'. This demonstrates scSGC's superior ability to accurately match clusters to known cell types, highlighting the lack of consistency and accuracy in competing methods.

3.3.2 Marker Gene Identification

Additionally, we conducted DEGs analysis and marker gene identification based on clusters identified by scSGC. Using the Wilcoxon Rank Sum test, Fig. 5(a) and (b) present the top 20 DEGs for different cell types identified by scSGC and the gold standard. This allows us to see which genes overlap between the clusters identified by scSGC and the gold standard, further supporting the accuracy of the cell type annotations. Fig. 5(c) illustrates a dot plot of the top three DEGs for each of the nine clusters. These genes represent the most characteristic DEGs identified by scSGC for each cluster and can be regarded as marker genes. For instance, the marker genes for cluster 1 are GCG, TTR, and IRX2, which serve as defining characteristics of the cell type in this cluster.

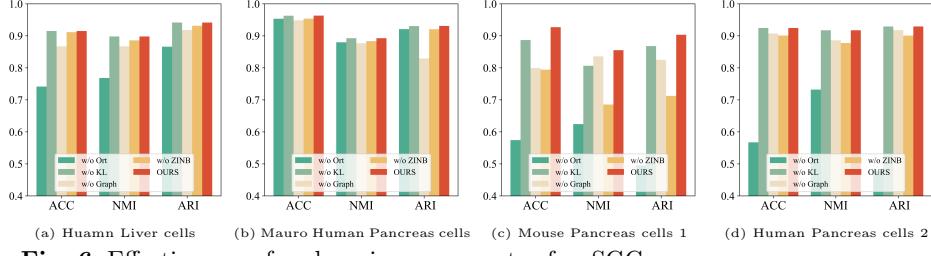


Fig. 6: Effectiveness of each main components of scSGC.

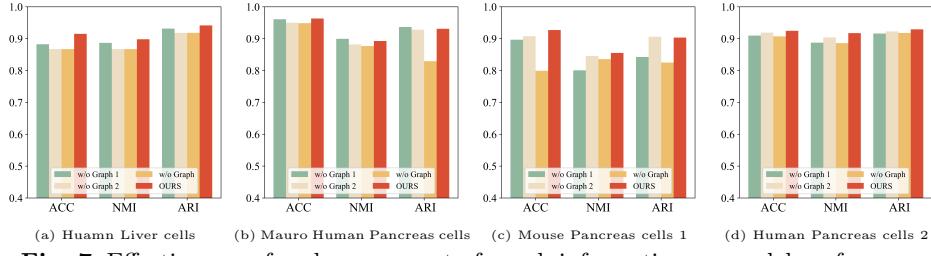


Fig. 7: Effectiveness of each component of graph information on model performance.

3.4 Ablation Study

3.4.1 Effectiveness of each main components

To validate the effectiveness of each component in scSGC, we performed ablation studies on four datasets, assessing the contributions of the three main modules related to the NCut loss, ZINB loss, and clustering loss. The NCut loss includes two sub-losses: the graph structure information and the orthogonality term.

Here, scSGC without the ZINB loss is denoted as *scSGC-w/o ZINB*, while scSGC without KL loss is referred to as *scSGC-w/o KL*. Furthermore, *scSGC-w/o Graph* and *scSGC-w/o Ort* represent models that exclude the graph information and orthogonality components of NCut loss, respectively. The complete model, incorporating all components, is referred to as *OURS*, i.e., scSGC. As demonstrated in Fig. 6, the systematic ablation study reveals that each components of scSGC contribute synergistically to performance optimization. Compared to other variations, these components significantly improve scSGC's efficiency and robustness.

3.4.2 Effectiveness of each component of graph information

As shown in Fig. 6, the incorporation of soft graph information significantly contributes to the overall performance improvement. To further validate the importance of different components within the soft graph, we conducted experiments comparing scSGC (denoted as *OURS*) with its various ablated versions. Specifically, *scSGC-w/o*

Table 4: Ablation comparisons of optimal transport on two datasets (mean \pm standard deviation).

Dataset	Metric	DEC	OURS
Human liver cells	ACC	$76.87_{\pm 1.3}$	$91.16_{\pm 0.3}$
	NMI	$72.74_{\pm 2.5}$	$89.17_{\pm 0.6}$
	ARI	$83.47_{\pm 2.2}$	$93.65_{\pm 0.5}$
Mauro Human Pancreas cells	ACC	$88.52_{\pm 2.5}$	$96.03_{\pm 0.2}$
	NMI	$83.68_{\pm 2.0}$	$90.23_{\pm 1.7}$
	ARI	$86.19_{\pm 5.4}$	$92.39_{\pm 1.9}$

Graph 1 and *scSGC-w/o Graph 2* represent the models without the feature similarity graph \mathcal{G}_1 and the cosine similarity graph \mathcal{G}_2 , respectively. Notably, Fig. 7 demonstrates that both types of similarity graphs contribute to the final model performance, with \mathcal{G}_1 showing a more substantial impact. The findings indicate that scSGC effectively captures subtle, continuous similarities between cells using the dual-channel cut-informed soft graph embedding module. By minimizing Joint NCut, it effectively integrates cellular relationships across multiple similarity measures, robustly uncovering latent cellular relationships and enhancing clustering precision.

3.4.3 Effectiveness of optimal transport

To illustrate our optimal transport-based clustering optimization module is better than the traditional approaches (e.g., DEC [39], 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. Fig. 6 demonstrates scSGC’s superiority over *scSGC-w/o KL* across four datasets, highlighting the importance of optimal transport in the model. We further substituted DEC for our optimal transport-based clustering optimization module within scSGC and compared the results. The results in Table 4 show that optimal transport significantly outperforms DEC across three key metrics. This indicates that optimal transport effectively maintains the balance of proximity guidance to pre-learned cluster centers, thus preventing degenerate solutions and achieving superior outcomes in the clustering tasks involving scRNA-seq data.

3.5 Parameter Sensitivity

Our work introduces five hyperparameters: α , β , γ , μ , and λ . The specific roles of these hyperparameters are as follows:

- α : A weight parameter that balances the contributions of the two graphs.
- β : A regularization parameter used to adjust the impact of the regularization term.
- γ and μ : Parameters that control the relative weights of the loss functions \mathcal{L}_{ZINB} and \mathcal{L}_{KL} , respectively.
- λ : A parameter that governs the trade-off between the accuracy of the assignment and the smoothness of the clustering in optimal transport.

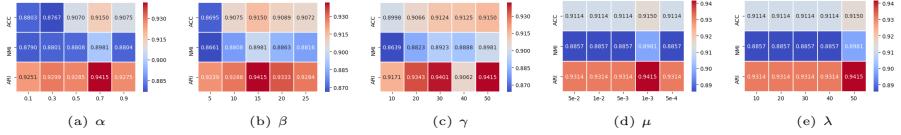


Fig. 8: Variation of scSGC performance on *Human Liver cells* dataset across different hyperparameters, including balance parameter α , tuning parameters β , γ , μ , and smoothness parameter λ .

We conducted a sensitivity analysis of the hyperparameters using the *Human Liver cells* dataset to investigate their impact on model performance. As shown in Fig. 8, the experimental results demonstrate that scSGC maintains stable performance across a wide range of hyperparameter values. For instance, when μ varies between 0.1 and 0.9, the overall ACC value fluctuates around 0.91, indicating the robustness of this hyperparameter. Similarly, adjustments to other hyperparameters also lead to gradual changes in model performance, further confirming the model’s stability with respect to hyperparameter settings.

To enhance the reproducibility of the experiments and the generalizability of the model, we recommend an initial set of hyperparameters: $\alpha = 0.7$, $\beta = 15$, $\gamma = 50$, $\mu = 1e - 3$, and $\lambda = 50$. These values serve as starting points for hyperparameter tuning, and adjusting them based on the specific characteristics of the data is necessary when applying scSGC to new datasets. Although our sensitivity analysis indicates that these settings generally yield near-optimal results across various datasets, fine-tuning the parameters to accommodate different datasets helps ensure the model’s stability and optimal performance.

3.6 Computational Efficiency

With the rapid advancement of scRNA-seq technology, the increasing size of datasets necessitates the development of highly efficient analysis methods. To systematically assess the computational efficiency of scSGC, we performed runtime comparison experiments across three datasets, benchmarking against nine competing methods, including deep learning and graph neural network approaches. All experiments were conducted on a server equipped with an EYPC 7742 processor, 1024GB RAM, and 8 Tesla A100 40GB GPUs. Fig. 9 shows that scSGC consistently demonstrates superior computational efficiency across all datasets, while baseline models exhibit varying degrees of computational burden. scSGC maintains a relatively low and stable runtime across the tested datasets, and its scalability to substantially larger datasets (e.g., $> 100k$ cells) requires further evaluation. Notably, scSGC’s performance may vary depending on the computational hardware, and further optimization could be explored for environments with different resource constraints.

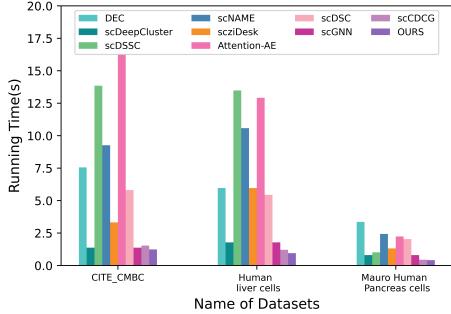


Fig. 9: The running time of scSGC and nine baselines.

3.7 Analysis of Simulated Data

To comprehensively assess the robustness and generalization ability of the model across varying data conditions, we generated ten simulated datasets and conducted a systematic comparison between scSGC and its baseline method, scCDCG, which serves as the current benchmark with the best performance. Specifically, we generated ten simulated scRNA-seq datasets using the R package Splatter [47], simulating both balanced datasets, where cell clusters are of approximately equal size, and imbalanced datasets, characterized by uneven cluster size distributions, under varying levels of dropout noise. Each dataset consists of 3,000 cells, 2,500 genes, and six distinct cell clusters. To simulate varying levels of technical noise (i.e., dropout effects), we indirectly controlled the degree of sparsity by adjusting the mid parameter in Splatter (set to -12, -4, -2, -1.3, and -0.6, with shape = -1). These settings correspond to approximate dropout rates of 5%, 10%, 15%, 20%, and 25%, respectively.

Notably, scSGC was developed based on the scCDCG framework, with hyperparameters fixed according to the configuration recommended in the "Parameter Sensitivity" section. These hyperparameters remained unchanged throughout the experiment to eliminate the potential influence of additional tuning on the final performance results. As shown in Table 5, scSGC consistently outperformed scCDCG on all simulated datasets in terms of ACC, NMI, and ARI, thus validating the robustness and generalization capability of the proposed method under controlled conditions. A more detailed analysis of the results reveals the following key observations: (i) On balanced datasets, the overall performance of scSGC and scCDCG was generally better compared to imbalanced datasets, indicating that clustering methods are sensitive to class distribution. (ii) scSGC demonstrated stable performance advantages even under challenging conditions, such as high dropout noise or severe class imbalance, reflecting its strong noise robustness and adaptability. (iii) Particularly in imbalanced data scenarios, the performance improvement of scSGC over scCDCG was more pronounced, suggesting that the improvements made to the model effectively mitigated the negative impacts of data sparsity and class imbalance on clustering outcomes.

Table 5: Performance comparison between scSGC and scCDCG on ten simulated datasets (mean \pm std).

Method	Metric	Balanced data					Unbalanced data				
		5%	10%	15%	20%	25%	5%	10%	15%	20%	25%
scSGC	ACC	82.7 \pm 0.7	75.2 \pm 4.4	82.6 \pm 0.7	76.8 \pm 0.7	73.8 \pm 2.1	72.1 \pm 2.1	84.4 \pm 3.5	47.1 \pm 0.8	74.0 \pm 1.4	62.5 \pm 4.3
	NMI	97.7 \pm 1.9	74.2 \pm 12.9	96.4 \pm 0.7	76.4 \pm 0.7	67.5 \pm 2.1	60.9 \pm 6.4	82.7 \pm 1.3	27.5 \pm 1.0	67.0 \pm 1.4	60.0 \pm 2.3
	ARI	98.2 \pm 1.6	68.9 \pm 12.0	97.1 \pm 0.7	77.3 \pm 0.7	66.1 \pm 2.1	61.0 \pm 2.1	85.0 \pm 4.4	24.5 \pm 4.4	61.6 \pm 1.4	50.0 \pm 6.0
scCDCG	ACC	62.1 \pm 2.8	61.1 \pm 6.6	60.9 \pm 1.0	64.6 \pm 4.1	69.3 \pm 8.0	67.0 \pm 10.6	46.7 \pm 2.1	40.9 \pm 0.5	59.8 \pm 5.8	58.3 \pm 1.8
	NMI	62.7 \pm 0.2	54.2 \pm 4.9	50.7 \pm 5.2	56.9 \pm 9.6	67.3 \pm 12.1	59.8 \pm 16.6	37.1 \pm 4.9	22.0 \pm 0.3	50.0 \pm 6.0	47.5 \pm 5.8
	ARI	56.5 \pm 2.7	46.7 \pm 5.0	48.3 \pm 5.0	52.8 \pm 7.6	65.7 \pm 10.6	56.5 \pm 15.6	28.7 \pm 3.2	19.1 \pm 0.7	45.5 \pm 8.0	41.3 \pm 5.5

4 Conclusion

In conclusion, we propose scSGC, an efficient and accurate framework for clustering single-cell RNA sequencing data. By integrating dual-channel soft graph representation learning with deep cut-informed techniques and incorporating ZINB-based feature autoencoder and optimal transport-driven clustering optimization, scSGC effectively addresses the critical challenges associated with traditional hard graph constructions, improving clustering accuracy while preserving biological relevance. This unified design enables scSGC to model complex, continuous relationships between cells while preserving biologically meaningful structures across diverse datasets.

The key contributions of this study are multifold. First, we formulate the clustering problem through a deep soft graph-cut perspective, enabling the development of a more flexible and accurate scRNA-seq clustering framework. Second, we propose a dual-channel, cut-informed soft graph learning module that effectively captures continuous similarities between cells. This design, in combination with the Joint NCut strategy, promotes consistency across learned graph structures and boosts downstream clustering performance. Third, we incorporate an optimal transport mechanism to refine clustering assignments, enabling minimal transport cost solutions that support both stability and biological coherence in cell population partitioning. Finally, we perform extensive benchmarking on ten diverse datasets, where scSGC demonstrates superior performance over eleven state-of-the-art clustering models across multiple criteria, including clustering accuracy, cell type annotation, and computational efficiency. These results highlight the robustness and generalizability of our framework in practical single-cell analysis settings.

Looking ahead, we intend to further strengthen scSGC by incorporating advanced graph learning techniques and adapting the framework to accommodate other high-dimensional omics data. Further, we also plan to improve its scalability and efficiency to handle increasingly large and complex datasets. These developments will likely open new avenues for understanding intricate cellular mechanisms and drive innovations in personalized medicine.

5 Abbreviations

scRNA-seq: single-cell RNA sequencing;
ZINB: zero-inflated negative binomial;

GNN: graph neural network;
SOTA: state-of-the-art;
KL: Kullback-Leibler;
ACC: Accuracy;
NMI: Normalized Mutual Information;
ARI: Adjusted Rand Index.

6 Declarations

6.1 Ethics approval and consent to participate

Not applicable.

6.2 Consent for publication

Not applicable.

6.3 Availability of data and materials

This manuscript does not report data generation, and all datasets used in this work are publicly available from open sources. All data and code required to reproduce the results presented in SGC will be made available on GitHub (<https://github.com/XPgogogo/seSGC>). In addition, concerning the baseline methods, we have provided download links for the relevant code in the “`readme.md`” file on GitHub to facilitate access and verification.

6.4 Competing interests

The authors declare that they have no competing interests.

6.5 Funding

This work is partially supported by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No.XDA0460104, National Natural Science Foundation of China (No.92470204) and the Beijing Natural Science Foundation (No.4254089).

6.6 Authors' contributions

PX conceptualized and designed the methodology, conducted all experiments, analyzed the results, and wrote the main manuscript text. PW contributed to parts of the manuscript writing and provided guidance on model design. Other authors participated in discussions on the methodology and offered guidance for the method design. All authors reviewed and approved the final manuscript.

6.7 Acknowledgements

Not applicable.

References

- [1] Yang, X., Liu, G., Feng, G., Bu, D., Wang, P., Jiang, J., Chen, S., Yang, Q., Miao, H., Zhang, Y., et al.: Genecompass: deciphering universal gene regulatory mechanisms with a knowledge-informed cross-species foundation model. *Cell Research*, 1–16 (2024)
- [2] Kiselev, V.Y., Andrews, T.S., Hemberg, M.: Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics* **20**(5), 273–282 (2019)
- [3] Wang, P., Liu, W., Wang, J., Liu, Y., Li, P., Xu, P., Cui, W., Zhang, R., Long, Q., Hu, Z., et al.: secompass: An integrated multi-species scrna-seq database for ai-ready. *Advanced Science*, 2500870 (2025)
- [4] Haghverdi, L., Büttner, M., Wolf, F.A., Buettner, F., Theis, F.J.: Diffusion pseudotime robustly reconstructs lineage branching. *Nature methods* **13**(10), 845–848 (2016)
- [5] Levine, J.H., Simonds, E.F., Bendall, S.C., Davis, K.L., El-ad, D.A., Tadmor, M.D., Litvin, O., Fienberg, H.G., Jager, A., Zunder, E.R., et al.: Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis. *Cell* **162**(1), 184–197 (2015)
- [6] Van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A.J., Burdziak, C., Moon, K.R., Chaffer, C.L., Pattabiraman, D., et al.: Recovering gene interactions from single-cell data using data diffusion. *Cell* **174**(3), 716–729 (2018)
- [7] Butler, A., Hoffman, P., Smibert, P., Papalexi, E., Satija, R.: Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology* **36**(5), 411–420 (2018)
- [8] Lin, P., Troup, M., Ho, J.W.: Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome biology* **18**(1), 1–11 (2017)
- [9] Wang, B., Ramazzotti, D., De Sano, L., Zhu, J., Pierson, E., Batzoglou, S.: Simlr: A tool for large-scale genomic analyses by multi-kernel learning. *Proteomics* **18**(2), 1700232 (2018)
- [10] Park, S., Zhao, H.: Spectral clustering based on learning similarity matrix. *Bioinformatics* **34**(12), 2069–2076 (2018)
- [11] Tian, T., Wan, J., Song, Q., Wei, Z.: Clustering single-cell rna-seq data with a model-based deep learning approach. *Nature Machine Intelligence* **1**(4), 191–198 (2019)
- [12] Li, S., Guo, H., Zhang, S., Li, Y., Li, M.: Attention-based deep clustering method

for scRNA-seq cell type identification. *PLOS Computational Biology* **19**(11), 1011641 (2023)

- [13] Gan, Y., Li, N., Zou, G., Xin, Y., Guan, J.: Identification of cancer subtypes from single-cell RNA-seq data using a consensus clustering method. *BMC Medical Genomics* **11**, 65–72 (2018)
- [14] Traag, V.A., Waltman, L., Van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. *Scientific reports* **9**(1), 1–12 (2019)
- [15] Luo, Z., Xu, C., Zhang, Z., Jin, W.: A topology-preserving dimensionality reduction method for single-cell RNA-seq data using graph autoencoder. *Scientific reports* **11**(1), 20028 (2021)
- [16] Ciortan, M., Defrance, M.: Gnn-based embedding for clustering scRNA-seq data. *Bioinformatics* **38**(4), 1037–1044 (2022)
- [17] Wang, J., Ma, A., Chang, Y., Gong, J., Jiang, Y., Qi, R., Wang, C., Fu, H., Ma, Q., Xu, D.: scgnn is a novel graph neural network framework for single-cell RNA-seq analyses. *Nature Communications* **12**(1), 1882 (2021)
- [18] 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. *Briefings in Bioinformatics* **23**(2), 018 (2022)
- [19] Zhou, Y., Wang, P., Dong, H., Zhang, D., Yang, D., Fu, Y., Wang, P.: Make graph neural networks great again: A generic integration paradigm of topology-free patterns for traffic speed prediction. *arXiv preprint arXiv:2406.16992* (2024)
- [20] Ning, Z., Qiao, Z., Dong, H., Du, Y., Zhou, Y.: Lightcake: A lightweight framework for context-aware knowledge graph embedding. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 181–193 (2021). Springer
- [21] Alekseev, V.E., Bolia, R., Korobitsyn, D.V., Lozin, V.V.: Np-hard graph problems and boundary classes of graphs. *Theoretical Computer Science* **389**(1-2), 219–236 (2007)
- [22] Tang, Y., Huang, Z., Cheng, J., Zhou, G., Feng, S., Zheng, H.: Graph neural network-based node classification with hard sample strategy. In: *2021 International Conference on Cyber-Physical Social Intelligence (ICCSI)*, pp. 1–4 (2021). IEEE
- [23] WU, Y., HUANG, H., SONG, Y., JIN, H.: Soft-gnn: towards robust graph neural networks via self-adaptive data utilization. *Frontiers of Computer Science* **19**(4), 194311 (2025)
- [24] Ning, Z., Wang, P., Wang, P., Qiao, Z., Fan, W., Zhang, D., Du, Y., Zhou,

- Y.: Graph soft-contrastive learning via neighborhood ranking. arXiv preprint arXiv:2209.13964 (2022)
- [25] Ning, Z., Wang, Z., Zhang, R., Xu, P., Liu, K., Wang, P., Ju, W., Wang, P., Zhou, Y., Cambria, E., et al.: Deep cut-informed graph embedding and clustering. arXiv preprint arXiv:2503.06635 (2025)
 - [26] Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., Sun, X.: Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3438–3445 (2020)
 - [27] Ning, Z., Wang, P., Qiao, Z., Wang, P., Zhou, Y.: Rethinking graph contrastive learning through relative similarity preservation. arXiv preprint arXiv:2505.05533 (2025)
 - [28] Thumbakara, R.K., George, B.: Soft graphs. General mathematics notes **21**(2) (2014)
 - [29] Wang, Z., Wang, P., Liu, K., Wang, P., Fu, Y., Lu, C.-T., Aggarwal, C.C., Pei, J., Zhou, Y.: A comprehensive survey on data augmentation. arXiv preprint arXiv:2405.09591 (2024)
 - [30] Xu, P., Ning, Z., Xiao, M., Feng, G., Li, X., Zhou, Y., Wang, P.: sccdeg: Efficient deep structural clustering for single-cell rna-seq via deep cut-informed graph embedding. In: International Conference on Database Systems for Advanced Applications, pp. 172–187 (2024). Springer
 - [31] Xu, P., Ning, Z., Li, P., Liu, W., Wang, P., Cui, J., Zhou, Y., Wang, P.: scsiamese-clu: A siamese clustering framework for interpreting single-cell rna sequencing data. arXiv preprint arXiv:2505.12626 (2025)
 - [32] Baron, M., Veres, A., Wolock, S.L., Faust, A.L., Gaujoux, R., Vetere, A., Ryu, J.H., Wagner, B.K., Shen-Orr, S.S., Klein, A.M., 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)
 - [33] Muraro, M.J., Dharmadhikari, G., Grün, D., Groen, N., Dielen, T., Jansen, E., Van Gurp, L., Engelse, M.A., Carlotti, F., De Koning, E.J., et al.: A single-cell transcriptome atlas of the human pancreas. Cell systems **3**(4), 385–394 (2016)
 - [34] Zheng, G.X., Terry, J.M., Belgrader, P., Ryvkin, P., Bent, Z.W., Wilson, R., Ziraldo, S.B., Wheeler, T.D., McDermott, G.P., Zhu, J., et al.: Massively parallel digital transcriptional profiling of single cells. Nature communications **8**(1), 14049 (2017)
 - [35] MacParland, S.A., Liu, J.C., Ma, X.-Z., Innes, B.T., Bartczak, A.M., Gage, B.K.,

- Manuel, J., Khuu, N., Echeverri, J., Linares, I., *et al.*: Single cell rna sequencing of human liver reveals distinct intrahepatic macrophage populations. *Nature communications* **9**(1), 4383 (2018)
- [36] A single-cell transcriptomic atlas characterizes ageing tissues in the mouse. *Nature* **583**(7817), 590–595 (2020)
- [37] Žurauskienė, J., Yau, C.: pcareduce: hierarchical clustering of single cell transcriptional profiles. *BMC bioinformatics* **17**, 1–11 (2016)
- [38] Wang, H.-Y., Zhao, J.-p., Zheng, C.-H.: 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)
- [39] Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International Conference on Machine Learning, pp. 478–487 (2016). PMLR
- [40] Ciortan, M., Defrance, M.: Contrastive self-supervised clustering of scRNA-seq data. *BMC bioinformatics* **22**(1), 280 (2021)
- [41] Wang, H., Zhao, J., Zheng, C., Su, Y.: scdssc: Deep sparse subspace clustering for scRNA-seq data. *PLOS Computational Biology* **18**(12), 1010772 (2022)
- [42] Wan, H., Chen, L., Deng, M.: scname: neighborhood contrastive clustering with ancillary mask estimation for scRNA-seq data. *Bioinformatics* **38**(6), 1575–1583 (2022)
- [43] Chen, L., Wang, W., Zhai, Y., Deng, M.: Deep soft k-means clustering with self-training for single-cell rna sequence data. *NAR genomics and bioinformatics* **2**(2), 039 (2020)
- [44] Wolf, F.A., Angerer, P., Theis, F.J.: Scanpy: large-scale single-cell gene expression data analysis. *Genome biology* **19**, 1–5 (2018)
- [45] Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research* **3**(Dec), 583–617 (2002)
- [46] Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1073–1080 (2009)
- [47] Zappia, L., Phipson, B., Oshlack, A.: Splatter: simulation of single-cell rna sequencing data. *Genome biology* **18**(1), 174 (2017)