Briefings in Bioinformatics, 2026, 27, bbag330
https://doi.org/10.1093/bib/bbag330
Published: 17 June 2026
Problem Solving Protocol

OXFORD

GatorSC: multi-scale cell and gene graphs with mixture-of-experts fusion for single-cell transcriptomics

Yuxi Liu[1,‡], Zhenhao Zhang[2,‡], Mufan Qiu[3], Song Wang[4], Flora D. Salim[5], Jun Shen[6], Tianlong Chen[3], Imran Razzak[7], Fuyi Li[8,9,*], Jiang Bian[1,*]

[1]Department of Biostatistics and Health Data Science, Indiana University School of Medicine, Indianapolis, 410 W 10th St, IN 46202, United States
[2]College of Life Sciences, Northwest A&F University, No. 3 Taicheng Road, Yangling, Shaanxi 712100, China
[3]Department of Computer Science, The University of North Carolina at Chapel Hill, 232 S Columbia St, NC 27599, United States
[4]Department of Computer Science, University of Central Florida, 4000 Central Florida Blvd., FL 32816, United States
[5]School of Computer Science and Engineering, Faculty of Engineering, University of New South Wales, High St, Kensington, NSW 2052, Australia
[6]School of Computing and Information Technology, Faculty of Engineering and Information Sciences, University of Wollongong, Wollongong, NSW 2522, Australia
[7]Division of Biology and Life Science, Muhammad bin Zayed University of Artificial Intelligence, Building 1B, Masdar City 20302, Abu Dhabi, UAE
[8]South Australian immunoGENomics Cancer Institute (SAiGENCI), The University of Adelaide, AHMS Building, North Terrace, SA 5005, Australia
[9]College of Information Engineering, Northwest A&F University, Yangling, Shaanxi 712100, China
*Corresponding authors. Fuyi Li, South Australian immunoGENomics Cancer Institute (SAiGENCI), The University of Adelaide, AHMS Building, North Terrace, SA 5005, Australia. E-mail: fuyi.li@adelaide.edu.au; Jiang Bian, Biostatistics and Health Data Science, School of Medicine, Indiana University, 410 W 10th St, IN 46202, United States. E-mail: bianji@iu.edu.
‡Yuxi Liu and Zhenhao Zhang contributed equally to this work.

Abstract

Single-cell RNA sequencing (scRNA-seq) enables high-resolution characterization of cellular heterogeneity, but its rich, complementary structure across cells and genes remains underexploited, especially in the presence of technical noise and sparsity. Effectively leveraging this multi-scale structure is essentially an information fusion problem that requires integrating heterogeneous graph-based views of cells and genes into robust low-dimensional representations. In this paper, we introduce GatorSC, a unified representation learning framework that models scRNA-seq data through multi-scale cell and gene graphs and fuses them with a mixture-of-experts architecture. GatorSC constructs a global cell–cell graph, a global gene–gene graph, and a local gene–gene graph derived from neighborhood-specific subgraphs, and learns graph neural network embeddings that are adaptively fused by a gating network. To learn noise-robust and structure-preserving embeddings without labels, we couple graph reconstruction and graph contrastive learning in a unified self-supervised objective applied to both cell- and gene-level graphs. We evaluate GatorSC on 19 publicly available scRNA-seq datasets covering diverse tissues, species, and sequencing platforms. Experiments showed that GatorSC consistently outperforms state-of-the-art deep generative, graph-based, and contrastive methods for cell clustering, gene expression imputation, and cell-type annotation. The learned embeddings are used for accurate trajectory inference, recovery of canonical marker gene programs, and cell-type-specific pathway signatures in an Alzheimer's disease single-nucleus dataset. GatorSC provides a flexible foundation for comprehensive single-cell transcriptomic analysis and can be readily extended to multi-omic and spatial modalities.

Keywords scRNA-seq data, contrastive learning, mixture-of-experts, cell clustering, cell type annotation

Introduction

Single-cell RNA sequencing (scRNA-seq) has become a key technology for profiling transcriptomes at cellular resolution and for dissecting heterogeneous tissues, developmental processes, and disease-associated cell states [1–3]. However, scRNA-seq data are notoriously challenging to analyze: expression matrices are high-dimensional, extremely sparse due to pervasive dropout, and subject to substantial technical noise. scRNA-seq data inherently contain complementary structure at multiple scales, including global relationships between cells, functional dependencies among genes, and context-specific gene programs that are active only in particular cellular neighborhoods. Effectively exploiting this rich multi-scale structure is fundamentally an information fusion problem that requires combining heterogeneous graph views into a single, robust representation of each cell.

Received: February 9, 2026. **Revised:** April 4, 2026. **Accepted:** May 26, 2026

A large body of work has applied deep representation learning to scRNA-seq data. Deep generative and autoencoder-based models such as scVI [4], scDeepCluster [5], DESC [6], and DeepBID [7] model count distributions with variational or deterministic autoencoders and couple reconstruction losses with clustering objectives, providing powerful latent representations for normalization, clustering, and denoising. Graph-based methods including scGNN [8], graph-sc [9], and scDSC [10] explicitly construct cell or gene graphs and utilize graph neural networks (GNNs) to propagate information across local neighborhoods, often achieving more accurate clustering and imputation than purely feature-based models. More recently, contrastive learning frameworks such as scGPCL [11] and scSimGCL [12] employ graph augmentations and instance- or prototype-level contrastive losses to learn noise-robust embeddings that are less tied to raw reconstruction errors. These approaches have substantially improved single-cell analysis, but they typically focus on one or two graph views and adopt relatively simple strategies for combining them.

Despite this progress, several important gaps remain from an information fusion perspective. First, most existing methods do not jointly model multi-scale structure across both cells and genes within a single representation. They may use a global cell–cell graph or a gene–cell bipartite graph, but rarely integrate a global cell manifold, a global gene network, and local, context-dependent gene dependencies. This makes it difficult to associate specific gene programs with particular regions of the cell manifold or to resolve closely related cell subtypes that differ by subtle shifts in gene expression programs. Second, when multiple graph views are considered, they are usually combined by fixed fusion schemes such as concatenation or uniform averaging, which cannot adapt the relative contribution of each view across datasets or even across individual cells. Third, most self-supervised objectives emphasize either reconstruction of noisy counts or contrastive agreement between graph views, but not both, potentially limiting the ability to simultaneously preserve graph topology, denoise expression, and enforce semantic consistency in the latent space.

In this study, we introduce **GatorSC**, a unified representation learning framework designed to fill these gaps by combining multi-scale graph modeling with a mixture-of-experts (MoE) fusion architecture and dual self-supervision. **GatorSC** constructs three hierarchical graphs from scRNA-seq data, including a global cell–cell graph capturing population-level organization, a global gene–gene graph encoding cross-cell functional dependencies, and a local gene–gene graph derived from neighborhood-specific subgraphs that emphasize context-dependent interactions, and then learns GNN embeddings on each graph. A modality-specific MoE assigns one expert to each graph view and uses a gating network to adaptively fuse expert outputs into a unified cell representation, allowing different datasets and cell populations to draw more heavily on the most informative structural signals. Finally, a unified self-supervised objective couples graph reconstruction losses with graph contrastive learning across the hierarchical graphs, jointly preserving structural fidelity and semantic consistency without relying on cell-type labels. Through extensive experiments on diverse scRNA-seq datasets, we empirically demonstrate that **GatorSC** yields robust gains in cell clustering, gene expression imputation, and cell-type annotation compared with state-of-the-art deep generative, graph-based, and contrastive methods, thereby establishing it as a flexible foundation for single-cell transcriptomic analysis.

Methods

Hierarchical graph modeling for scRNA-seq data

In this paper, we propose a hierarchical graph modeling framework that progressively captures biological structure in single-cell data, ranging from local gene-level interactions within individual cells to global cellular population architectures, as illustrated in Fig. 1a. Each layer builds upon the previous one while offering complementary biological insights. Accordingly, our framework consists of three hierarchical graphs that encode distinct yet synergistic levels of biological organization: a global cell–cell graph, a global gene–gene graph, and a local gene–gene graph.

Global cell–cell graph (population-level structure)

Starting from the gene-expression matrix, we compute pairwise similarities between cells to construct a global cell–cell graph, which delineates the population-level architecture of the dataset. This graph captures global similarity patterns and large-scale cellular organization, serving as the structural backbone for downstream representation learning.

Global gene–gene graph (cross-cell dependency network)

To capture gene-level dependencies that span multiple cells, we construct a global gene–gene graph that models cross-cell functional relationships among genes. This graph encodes robust gene–gene associations that remain stable across diverse cellular contexts. The resulting global gene network complements the cell–cell graph by providing a broader functional perspective, thereby bridging gene-level and cell-level representations.

Local gene–gene graph (context-specific dependency)

Building upon the global cell graph, we extract multiple cell subgraphs, each representing a localized cellular neighborhood. Within each subgraph, we recompute gene–gene similarities to construct context-specific gene graphs that capture dependency patterns unique to specific cellular contexts. These subgraph-level gene graphs are then integrated to form a local gene–gene graph that emphasizes fine-grained, context-dependent associations among genes, complementing the global graph structure with local specificity.

By integrating these three hierarchical levels of graphs (global cell–cell, global gene–gene, and local gene–gene), our framework offers a multi-scale, multi-view representation of scRNA-seq data. This design unifies both global and local dependencies across cells and genes, providing richer structural priors for deep learning models. Consequently, the learned representations are more biologically meaningful and better suited for downstream analyses, including cell clustering, cell type annotation, trajectory inference, and gene expression denoising.

Global cell–cell graph

To formalize the above design, we introduce a unified notation for the three graphs. Let $\mathbf{X} \in \mathbb{R}^{M \times N}$ denote the gene expression matrix, where each row corresponds to a cell and each column to a gene, with M being the number of cells and N the number of genes. From $\mathbf{X}$, we construct three graphs: the global cell–cell graph $\mathcal{G}^{(1)}$, the global gene–gene graph $\mathcal{G}^{(2)}$, and the local gene–gene graph $\mathcal{G}^{(3)}$. For $i = 1, 2, 3$, we

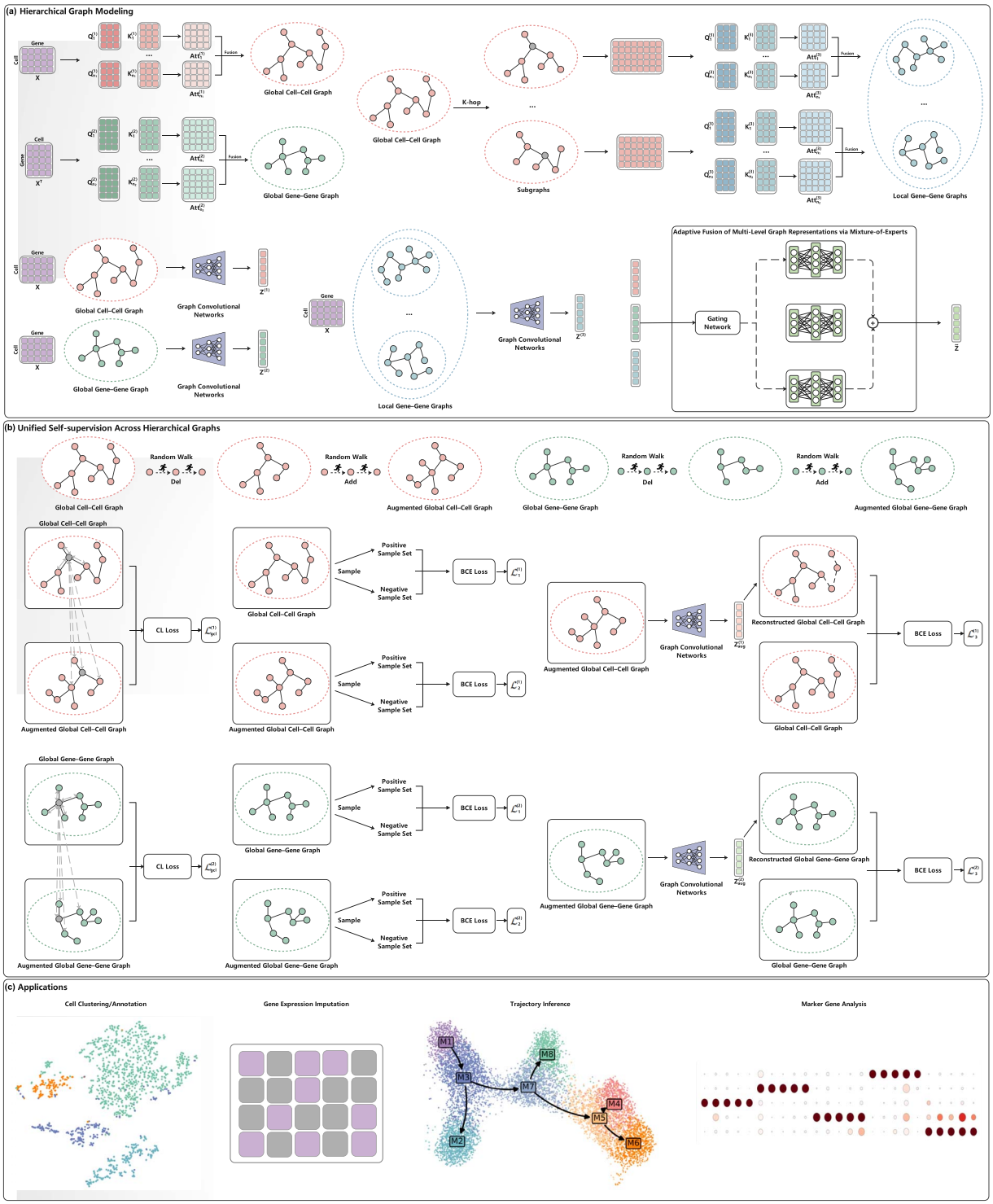

Figure 1 Overview of the **GatorSC** framework.

write

$$\mathcal{G}^{(i)} = \left(\mathcal{V}^{(i)}, \mathcal{E}^{(i)}, \mathbf{X}\right) = \left(\mathbf{A}^{(i)}, \mathbf{X}\right), \tag{1}$$

where $\mathcal{V}^{(i)} = \{v_1^{(i)}, \ldots, v_{N_i}^{(i)}\}$ is the node set, $\mathcal{E}^{(i)}$ is the edge set, and $\mathbf{A}^{(i)} \in \mathbb{R}^{N_i \times N_i}$ is the corresponding adjacency matrix.

To construct the global cell–cell graph $\mathcal{G}^{(1)}$, we compute its adjacency matrix $\mathbf{A}^{(1)}$ using a multi-head attention mechanism to learn cell–cell similarities. Given the gene expression matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$, we perform linear projections to obtain the query and key for each attention head. Then, scaled dot-product attention is applied to compute pairwise similarities between cells, followed by Softmax normalization to obtain attention weights. Each attention head is computed

independently, and the resulting outputs are concatenated and linearly transformed to obtain the adjacency matrix. The mathematical formulation can be written as follows:

$$\begin{aligned}\tilde{\mathbf{A}}^{(1)} &= \mathrm{MultiHeadAtt}(\mathbf{X}) \\ &= \left[\mathrm{head}_1^{(1)}(\mathbf{X}) \oplus \cdots \oplus \mathrm{head}_{n_1}^{(1)}(\mathbf{X})\right]\mathbf{W}_h^{(1)}, \\ \mathrm{head}_i^{(1)}(\mathbf{X}) &= \mathrm{Softmax}\left(\frac{\mathbf{Q}_i^{(1)\top}\mathbf{K}_i^{(1)}}{\sqrt{d_{k_1}}}\right), \\ \mathbf{Q}_i^{(1)} &= (\mathbf{W}_q^{(1)})_i\mathbf{X}^\top, \quad \mathbf{K}_i^{(1)} = (\mathbf{W}_k^{(1)})_i\mathbf{X}^\top,\end{aligned} \tag{2}$$

where all $\mathbf{W}$ are trainable parameters. $\mathbf{head}_i^{(1)}$ is the ith attention head. $\mathbf{Q}_i^{(1)}$ and $\mathbf{K}_i^{(1)}$ denote the query and key matrices of the ith attention head for graph $\mathcal{G}^{(1)}$, obtained via learnable linear projections $(\mathbf{W}_q^{(1)})_i$ and $(\mathbf{W}_k^{(1)})_i$, respectively. $\mathbf{W}_h^{(1)}$ denotes the output projection matrix used to aggregate the multi-head outputs. n_1 is the number of heads. $\oplus$ is the concatenation. d_{k_1} is the dimension of $\mathbf{K}^{(1)}$. Accordingly, a sparse and non-negative adjacency matrix $\mathbf{A}^{(1)}$ can be obtained using the intermediate matrix $\tilde{\mathbf{A}}^{(1)}$ with numerical constraints as follows:

$$\mathbf{A}^{(1)} = \begin{cases} 1, & \tilde{\mathbf{A}}^{(1)} \geq \varphi^{(1)}, \\ 0, & \tilde{\mathbf{A}}^{(1)} < \varphi^{(1)}, \end{cases} \tag{3}$$

where $\varphi^{(1)}$ is a learnable threshold that excludes the values lower than that.

Global gene–gene graph

We construct a global gene–gene graph $\mathcal{G}^{(2)}$ by computing its adjacency matrix $\mathbf{A}^{(2)}$ at the gene level. To learn the intrinsic relationships among genes, we use the transposed gene expression matrix $\mathbf{X}^\top \in \mathbb{R}^{N\times M}$, and apply a multi-head scaled dot-product attention mechanism, consistent with that used in the previous section, to compute pairwise similarity weights among genes. The outputs from all attention heads are aggregated to obtain the adjacency matrix. The mathematical formulation can be written as follows:

$$\begin{aligned}\tilde{\mathbf{A}}^{(2)} &= \mathrm{MultiHeadAtt}(\mathbf{X}^\top) \\ &= \left[\mathrm{head}_1^{(2)}(\mathbf{X}^\top) \oplus \cdots \oplus \mathrm{head}_{n_2}^{(2)}(\mathbf{X}^\top)\right]\mathbf{W}_h^{(2)}, \\ \mathrm{head}_i^{(2)}(\mathbf{X}) &= \mathrm{Softmax}\left(\frac{\mathbf{Q}_i^{(2)\top}\mathbf{K}_i^{(2)}}{\sqrt{d_{k_2}}}\right), \\ \mathbf{Q}_i^{(2)} &= (\mathbf{W}_q^{(2)})_i\mathbf{X}, \quad \mathbf{K}_i^{(2)} = (\mathbf{W}_k^{(2)})_i\mathbf{X}, \\ \mathbf{A}^{(2)} &= \begin{cases} 1, & \tilde{\mathbf{A}}^{(2)} \geq \varphi^{(2)}, \\ 0, & \tilde{\mathbf{A}}^{(2)} < \varphi^{(2)}, \end{cases}\end{aligned} \tag{4}$$

where all $\mathbf{W}$ are trainable parameters. $\mathbf{Q}_i^{(2)}$ and $\mathbf{K}_i^{(2)}$ denote the query and key matrices of the ith attention head for graph $\mathcal{G}^{(2)}$, obtained via learnable linear projections $(\mathbf{W}_q^{(2)})_i$ and $(\mathbf{W}_k^{(2)})_i$, respectively. $\mathbf{W}_h^{(2)}$ denotes the output projection matrix used to aggregate the multi-head outputs. n_2 is the number of attention heads. d_{k_2} is the dimension of $\mathbf{K}^{(2)}$. $\varphi^{(2)}$ is a learnable threshold that excludes the values lower than that.

Local gene–gene graph

Based on the foundation established by the global cell–cell graph, a subgraph is extracted for each node to represent its structural relationships. In particular, the subgraph for a given node consists of its k-hop neighboring nodes. We take an example for node $v_e^{(1)}$, and its subgraph $\mathcal{G}_e^{(1)}$ can be defined as follows:

$$\begin{aligned}\mathcal{G}_e^{(1)} &= (\mathcal{V}_e^{(1)}, \mathcal{E}_e^{(1)}, \mathbf{X}_e), \\ \mathcal{V}_e^{(1)} &= \{e\} \cup \mathcal{N}_e^{(1)}, \\ \mathcal{N}_e^{(1)} &= \left\{r \,\middle|\, 1 \leq d(e,r) \leq k\right\}, \\ \mathcal{E}_e^{(1)} &= \left\{(i,j) \,\middle|\, \mathbf{A}_{i,j}^{(1)} = 1,\ e_i, e_j \in \mathcal{V}_e^{(1)}\right\},\end{aligned} \tag{5}$$

where $\mathcal{N}_e^{(1)}$ is the set of k-hop neighboring nodes of $v_e^{(1)}$. Through the above steps, given a node $v_e^{(1)}$, we can obtain its subgraph $\mathcal{G}_e^{(1)}$ and the corresponding set of k-hop neighbors $\mathcal{N}_e^{(1)}$. Based on $v_e^{(1)}$ and $\mathcal{N}_e^{(1)}$, we can extract the subset of gene expression data corresponding to node $\mathbf{X}_{v_e}$. Using $\mathbf{X}_{v_e}$ as input, we then apply the multi-head scaled dot-product attention mechanism again, to compute the adjacent matrix. The mathematical formulation can be written as follows:

$$\begin{aligned}\mathbf{A}_{v_e}^{\prime(3)} &= \mathrm{MultiHeadAtt}(\mathbf{X}_{v_e}) \\ &= \left[\mathrm{head}_1^{(3)}(\mathbf{X}_{v_e}) \oplus \cdots \oplus \mathrm{head}_{n_3}^{(3)}(\mathbf{X}_{v_e})\right]\mathbf{W}_h^{(3)}, \\ \mathrm{head}_i^{(3)}(\mathbf{X}_{v_e}) &= \mathrm{Softmax}\left(\frac{\mathbf{Q}_i^{(3)\top}\mathbf{K}_i^{(3)}}{\sqrt{d_{k_3}}}\right), \\ \mathbf{Q}_i^{(3)} &= (\mathbf{W}_q^{(3)})_i\mathbf{X}_{v_e}, \quad \mathbf{K}_i^{(3)} = (\mathbf{W}_k^{(3)})_i\mathbf{X}_{v_e}, \\ \tilde{\mathbf{A}}^{(3)} &= \begin{cases} 1, & \mathbf{A}_{v_e}^{\prime(3)} \geq \varphi^{(3)}, \\ 0, & \mathbf{A}_{v_e}^{\prime(3)} < \varphi^{(3)}, \end{cases} \\ \mathbf{A}^{(3)} &= \left\{\tilde{\mathbf{A}}_{v_e}^{(3)}\right\}_{e=1}^{n_{\mathrm{sub}}},\end{aligned} \tag{6}$$

where all $\mathbf{W}$ are trainable parameters. $\mathbf{Q}_i^{(3)}$ and $\mathbf{K}_i^{(3)}$ denote the query and key matrices of the ith attention head for graph $\mathcal{G}^{(3)}$, obtained via learnable linear projections $(\mathbf{W}_q^{(3)})_i$ and $(\mathbf{W}_k^{(3)})_i$, respectively. $\mathbf{W}_h^{(3)}$ denotes the output projection matrix used to aggregate the multi-head outputs. n_3 is the number of attention heads. d_{k_3} is the dimension of $\mathbf{K}^{(3)}$. n_{sub} is the number of subgraphs. $\varphi^{(3)}$ is a learnable threshold that excludes the values lower than that. Here, the local gene–gene component is maintained as a batch of cell-centered local graphs rather than explicitly collapsed into a single adjacency matrix. Each extracted cell subgraph gives rise to one corresponding local gene–gene graph, and the local branch encodes these graphs in batch form to produce cell-specific local representations. Their integration is performed at the embedding level in the downstream fusion module.

Adaptive fusion of multilevel graph representations via mixture-of-experts

After obtaining the three graphs $\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \mathcal{G}^{(3)}$, the next step is to learn the node representations $\mathcal{V}^{(i)}$ in each graph, using graph convolutional networks (GCNs) and multilayer perceptron (MLP), as formulated

below:

$$\mathbf{Z}^{(i)} = \mathrm{MLP}^{(i)}\big(\mathrm{GCN}^{(i)}(\mathbf{A}^{(i)}, \mathbf{X})\big), \quad i = 1, 2, 3, \tag{7}$$

where $\mathbf{Z}^{(i)}$ is the learned representation of $\mathbf{X}$ and $\mathcal{G}^{(i)}$. The obtained representations $\mathbf{Z}^{(1)}$, $\mathbf{Z}^{(2)}$, $\mathbf{Z}^{(3)}$ capture three distinct levels of intrinsic data dependencies. These representations are complementary rather than redundant. Since each representation emphasizes different structural aspects, a naive fusion strategy such as simple concatenation or averaging would ignore the varying relative importance of these levels. We introduce an adaptive fusion mechanism based on the MoE paradigm. MoE is a modular neural architecture in which multiple specialized experts are trained to model different input contributions, while a gating network adaptively controls their weight [13]. Instead of relying on a single network to handle all variations in the data, MoE introduces a gating network that dynamically assigns input samples to different experts based on their characteristics [14]. The gating network produces a probability distribution over experts, enabling the model to adaptively weigh the contributions of each expert and combine heterogeneous representations in a flexible and data-driven manner. Formally, we employ an MoE framework that consists of a set of L expert networks (with $L = 3$ in this study) and a gating network. Let $G(\mathbf{X})$ denote the output of the gating network and $f_i(\mathbf{X})$ the output of the ith expert network for a given input $\mathbf{X}$. The aggregated dense representation generated by the MoE can be written as

$$\begin{aligned}
\tilde{\mathbf{Z}} &= \mathcal{F}\Big(\mathbf{X}; \mathbf{W}_g, \{\mathbf{W}_i\}_{i=1}^{L}\Big) = \sum_{i=1}^{L} G(\mathbf{X}; \mathbf{W}_g)_i f_i(\mathbf{X}; \mathbf{W}_i), \\
f_i(\mathbf{X}; \mathbf{W}_i) &= \mathbf{Z}^{(i)} = \mathrm{MLP}^{(i)}\left(\mathrm{GCN}^{(i)}\left(\mathbf{A}^{(i)}, \mathbf{X}\right)\right), \\
G(\mathbf{X}; \mathbf{W}_g)_i &= \mathrm{Softmax}\big(g(\mathbf{X}; \mathbf{W}_g)\big)_i, \\
g(\mathbf{X}; \mathbf{W}_g) &= \big[f_1(\mathbf{X}; \mathbf{W}_1) \oplus \cdots \oplus f_L(\mathbf{X}; \mathbf{W}_L)\big]\mathbf{W}_g,
\end{aligned} \tag{8}$$

where $\tilde{\mathbf{Z}}$ is the dense representation learned. All $\mathbf{W}$ are trainable parameters. L is the number of expert networks, which is set to $L = 3$. To further clarify the implementation of the MoE fusion in Equation (8), we describe it as follows. **GatorSC** first obtains three branch-specific cell embeddings from the global cell-cell, global gene-gene, and local gene-gene graphs, and projects them to the same latent dimension. For each cell, the three embeddings are concatenated and passed through a gating network, and the resulting scores are normalized by a softmax function to produce cell-specific fusion weights. The final representation is then computed as a weighted combination of the three embeddings. In our implementation, the number of experts is three and the gating function is a single linear layer.

Unified self-supervision across hierarchical graphs

After obtaining the learned dense representation, we further optimize it in an end-to-end manner through a unified self-supervised learning framework that integrates both *generative* and *contrastive* paradigms, as shown in Fig. 1b. Self-supervised learning approaches can be broadly categorized into two types. The generative objective focuses on reconstructing the original signal or graph structure (e.g. graph reconstruction), emphasizing structural fidelity [15]. By recovering missing or corrupted connections, this approach enhances the model's robustness to incomplete or noisy data and benefits downstream tasks such as denoising, imputation, and link prediction. In contrast, the contrastive objective encourages consistent representation among semantically identical instances under different perturbations or graph views, while maintaining uniformity in the embedding space [16, 17]. This helps the model learn noise-invariant and discriminative representations, improving performance on clustering, classification, and related tasks.

We design two self-supervised objectives: graph reconstruction learning and graph contrastive learning. Before defining these objectives, we introduce graph augmentation, a critical step that generates semantically consistent yet structurally diverse graph views. Graph augmentation serves two primary purposes: (i) providing controlled corrupted inputs for generative tasks (e.g. node masking or edge removal), to enhance structural learning, and (ii) creating positive and negative sample pairs for contrastive learning to promote invariance and prevent representation collapse under perturbations.

We modify the graph structure by performing path-level operations, including edge deletion and addition along the sampled paths, to generate structurally perturbed views of the original graph. Given the graph $\mathcal{G}^{(1)}$, the augmentation process can be defined as follows:

$$\begin{aligned}
\varepsilon_{\mathrm{del}}^{(1)} &\sim \mathrm{RandomWalk}\big(\mathcal{V}_{\mathrm{del}}^{(1)}, l_{\mathrm{del}}^{(1)}\big), \\
\varepsilon_{\mathrm{add}}^{(1)} &\sim \mathrm{RandomWalk}\big(\mathcal{V}_{\mathrm{add}}^{(1)}, l_{\mathrm{add}}^{(1)}\big),
\end{aligned} \tag{9}$$

where $\mathcal{V}_{\mathrm{del}}^{(1)}$, $\mathcal{V}_{\mathrm{add}}^{(1)} \subseteq \mathcal{V}^{(1)}$ are sets of root nodes sampled from graph $\mathcal{G}^{(1)}$ following Bernoulli distributions, i.e. $\mathcal{V}_{\mathrm{del}}^{(1)} \sim \mathbf{Bernoulli}\big(p_{\mathrm{del}}^{(1)}\big)$, $\mathcal{V}_{\mathrm{add}}^{(1)} \sim \mathbf{Bernoulli}\big(p_{\mathrm{add}}^{(1)}\big)$, where $p_{\mathrm{del}}^{(1)}$, $p_{\mathrm{add}}^{(1)} \in (0, 1)$ are the deletion and addition probabilities, and $l_{\mathrm{del}}^{(1)}$, $l_{\mathrm{add}}^{(1)}$ are the maximum path lengths. Each set $\varepsilon_{\mathrm{del}}^{(1)}$, $\varepsilon_{\mathrm{add}}^{(1)}$ consists of edges/paths encountered by random walks starting from the sampled root nodes. A path is a sequence of edges; therefore, $\varepsilon_{\mathrm{del}}^{(1)}, \varepsilon_{\mathrm{add}}^{(1)}$ can be viewed as set of edges. Based on these sets, we construct the augmented graph as follows:

$$\begin{aligned}
\big(M_{\mathrm{del}}^{(1)}\big)_{ij} &= \mathbf{1}\big\{(i,j) \in \mathcal{E}_{\mathrm{del}}^{(1)}\big\}, \quad \big(M_{\mathrm{add}}^{(1)}\big)_{ij} = \mathbf{1}\big\{(i,j) \in \mathcal{E}_{\mathrm{add}}^{(1)}\big\}, \\
\mathbf{A}^{(1)-} &= \mathbf{A}^{(1)} \circ \big(\mathbf{1} - M_{\mathrm{del}}^{(1)}\big), \\
\mathbf{A}_{\mathrm{aug}}^{(1)} &= \mathbf{A}^{(1)-} + \big(\mathbf{1} - \mathbf{A}^{(1)-}\big) \circ M_{\mathrm{add}}^{(1)},
\end{aligned} \tag{10}$$

where the symbol $\circ$ denotes the Hadamard (element-wise) product; $\mathbf{1}$ is the all-ones matrix of the same size as $\mathbf{A}^{(1)}$; all additions/subtractions are element-wise. $\mathbf{A}_{\mathrm{aug}}^{(1)}$ is the adjacent matrix of the augmented graph, which we denote by $\mathcal{G}_{\mathrm{aug}}^{(1)}$. Having obtained the original graph $\mathcal{G}^{(1)}$ and its augmented version $\mathcal{G}_{\mathrm{aug}}^{(1)}$,we can now define two self-supervised tasks: graph reconstruction and graph contrastive learning.

We first introduce the graph reconstruction task. Specifically, the set of observed edges in the original graph $\mathcal{G}^{(1)}$ is denoted as $\mathcal{E}^{(1)+}$, which serves as the positive sample set, representing pairs of nodes that should be placed closer together in the embedding space. Conversely, a set of randomly sampled unobserved node pairs from $\mathcal{G}^{(1)}$ is used as the negative sample set, denoted as $\mathcal{E}^{(1)-}$. Similarly, for the augmented graph $\mathbf{A}_{\mathrm{aug}}^{(1)}$, we define the corresponding positive and negative edge sets as $\mathcal{E}_{\mathrm{aug}}^{(1)+}$, $\mathcal{E}_{\mathrm{aug}}^{(1)-}$, respectively. Based on these positive and negative

edge sets, the reconstruction loss can be formulated as follows:

$$\begin{aligned}
\mathbf{Z}_{\text{aug}}^{(1)} &= \text{MLP}(\text{GCN}(\mathbf{A}_{\text{aug}}^{(1)}, \mathbf{X})),\\
\mathcal{L}_1^{(1)} &= \text{BCE}\Big(\{s_{ij}\}_{(i,j)\in\mathcal{E}^{(1)+}}, 1\Big) + \text{BCE}\Big(\{s_{uv}\}_{(u,v)\in\mathcal{E}^{(1)-}}, 0\Big)\\
\mathcal{L}_2^{(1)} &= \text{BCE}\Big(\{s_{ij}\}_{(i,j)\in\mathcal{E}_{\text{aug}}^{(1)+}}, 1\Big) + \text{BCE}\Big(\{s_{uv}\}_{(u,v)\in\mathcal{E}_{\text{aug}}^{(1)-}}, 0\Big),\\
\mathcal{L}_3^{(1)} &= \text{BCE}\Big(\text{Sigmoid}(\mathbf{Z}_{\text{aug}}^{(1)}\mathbf{Z}_{\text{aug}}^{(1)\top}), \mathbf{A}^{(1)}\Big),\\
\mathcal{L}_{\text{rec}}^{(1)} &= \alpha_1\mathcal{L}_1^{(1)} + \alpha_2\mathcal{L}_2^{(1)} + \alpha_3\mathcal{L}_3^{(1)},
\end{aligned} \tag{11}$$

where $s_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$ is the dot-product similarity. $\mathbf{Z}_{\text{aug}}^{(1)}$ is the learned representation of $\mathbf{X}$ in the augmented graph $\mathcal{G}_{\text{aug}}^{(1)}$. $\text{BCE}(\cdot)$ represents the binary loss of cross-entropy. $\mathcal{L}_1^{(1)}$ and $\mathcal{L}_2^{(1)}$ correspond to the reconstruction losses for $\mathcal{G}^{(1)}$ and $\mathcal{G}_{\text{aug}}^{(1)}$, respectively. These loss terms encourage the model to assign higher similarity scores to observed (positive) edges while assigning lower scores to randomly sampled non-edges. $\mathcal{L}_3^{(1)}$ denotes the loss in reconstructing $\mathcal{G}_{\text{aug}}^{(1)}$ from $\mathcal{G}^{(1)}$. α_1, α_2, α_3 are coefficients that balance the contributions of the three losses. The total reconstruction loss is defined as $\mathcal{L}_{\text{rec}}^{(1)}$.

Next, we introduce the graph contrastive learning task. The core idea of contrastive learning is to define positive and negative sample pairs and optimize the model to maximize the similarity of positive pairs while minimizing the similarity of negative pairs [18, 19]. Based on $\mathcal{G}^{(1)}$ and its augmented counterpart $\mathcal{G}_{\text{aug}}^{(1)}$, we define the positive samples as follows: a node from $\mathcal{G}^{(1)}$ is selected as an anchor. The corresponding positive samples include (i) the nodes directly connected to the anchor in $\mathcal{G}^{(1)}$, (ii) its counterpart node in $\mathcal{G}_{\text{aug}}^{(1)}$, and (iii) the nodes connected to the anchor from $\mathcal{G}_{\text{aug}}^{(1)}$, while all remaining nodes are treated as negative samples. Given these definitions, the contrastive loss can be formulated as

$$\begin{aligned}
\mathcal{L}(\mathbf{Z}^{(1)}, \mathcal{G}^{(1)}) &= -\frac{1}{2|\mathcal{N}_{v_i}|+1}\log\frac{\exp(\text{Sim}(\mathbf{Z}_i^{(1)}, (\mathbf{Z}_{\text{aug}}^{(1)})_i)/\tau) + \sum_{v_j\in\mathcal{N}_i}\psi_{ij}}{\sum_{j\neq i}\psi_{ij}},\\
\psi_{ij} &= \exp(\text{Sim}(\mathbf{Z}_i^{(1)}, \mathbf{Z}_j^{(1)})/\tau) + \exp(\text{Sim}(\mathbf{Z}_i^{(1)}, (\mathbf{Z}_{\text{aug}}^{(1)})_j)/\tau),
\end{aligned} \tag{12}$$

where $\text{sim}(\cdot)$ denotes the similarity function, implemented as a dot product, and τ is a temperature parameter that controls the strength of the penalty for negative pairs. $\mathcal{N}_{v_i}$ is a set of positive neighbors for the ith node in $\mathcal{G}^{(1)}$. $\mathcal{L}(\mathbf{Z}^{(1)}, \mathcal{G}^{(1)})$ represents the contrastive loss when the anchor is selected from $\mathcal{G}^{(1)}$. Similarly, when the anchor is selected from $\mathcal{G}_{\text{aug}}^{(1)}$, the contrastive loss is denoted as $\mathcal{L}\Big(\mathbf{Z}_{\text{aug}}^{(1)}, \mathcal{G}_{\text{aug}}^{(1)}\Big)$. The overall contrastive loss $\mathcal{L}_{\text{gcl}}^{(1)}$ is then defined as

$$\mathcal{L}_{\text{gcl}}^{(1)} = \beta_1\,\mathcal{L}(\mathbf{Z}^{(1)}, \mathcal{G}^{(1)}) + \beta_2\,\mathcal{L}(\mathbf{Z}_{\text{aug}}^{(1)}, \mathcal{G}_{\text{aug}}^{(1)}), \tag{13}$$

where β_1, β_2 are coefficients that balance the contributions of the two contrastive losses.

Based on the above steps, the overall self-supervised loss for each hierarchical graph $\mathcal{G}^{(1)}$ can be defined as

$$\mathcal{L}_{\mathcal{G}^{(1)}} = \gamma_1\,\mathcal{L}_{\text{rec}}^{(1)} + \gamma_2\,\mathcal{L}_{\text{gcl}}^{(1)}, \tag{14}$$

where γ_1, γ_2 are coefficients that balance the generative (reconstruction) and contrastive objectives.

In summary, $\mathcal{G}^{(1)}$ models cell–cell relationships from a global population perspective, while $\mathcal{G}^{(2)}$ captures gene–gene dependencies from a global gene-level viewpoint. Based on $\mathcal{G}^{(1)}$, we can obtain the corresponding self-supervised loss $\mathcal{L}_{\mathcal{G}^{(1)}}$; and based on $\mathcal{G}^{(2)}$, we can obtain $\mathcal{L}_{\mathcal{G}^{(2)}}$. Finally, the overall self-supervised loss that integrates both cell and gene perspectives is formulated as

$$\mathcal{L} = \lambda\,\mathcal{L}_{\mathcal{G}^{(1)}} + (1-\lambda)\,\mathcal{L}_{\mathcal{G}^{(2)}}, \tag{15}$$

where λ is a trade-off coefficient that controls the relative importance of the two components. This unified objective allows the model to simultaneously preserve global structural fidelity (through reconstruction) and semantic consistency (through contrastive learning) across both cell- and gene-level graphs. It is worth noting that Equation (15) explicitly supervises only the global cell and global gene branches. Although the local gene component is encoded as the third representation branch in Equation (8) and contributes to the final fused embedding, it is represented as a collection of cell-specific local graphs rather than a single globally aligned graph; therefore, we did not impose a separate self-supervised objective on this branch in the current formulation. Its contribution is instead reflected through the fused representation, which is consistent with the performance drop observed in the ablation study when the local branch is removed.

Application tasks

As described above, the model is trained by minimizing the loss function defined in Equation (15). After training, the matrix $\tilde{\mathbf{Z}}$ in Equation (8) serves as the learned low-dimensional representation of the cells. Building on this unified representation, our framework can be readily applied to a variety of downstream single-cell analysis tasks, as illustrated in Fig. 1c.

For the clustering task, we feed $\tilde{\mathbf{Z}}$ into a general clustering method to obtain cluster assignments, where we use k-means as our clustering algorithm.

For the imputation task, we use $\tilde{\mathbf{Z}}$ as input to a regressor. Specifically, we employ a fully connected layer as the regressor to predict and impute missing gene expression values as

$$\hat{\mathbf{X}} = \mathbf{W}_x\tilde{\mathbf{Z}} + \mathbf{b}_x \tag{16}$$

$$\mathcal{L}_{\text{MAE}} = \frac{1}{M}\sum_{i=1}^{M}\left|\mathbf{X}_i - \hat{\mathbf{X}}_i\right|, \tag{17}$$

where $\mathbf{W}_x$ and $\mathbf{b}_x$ are trainable parameters. $\hat{\mathbf{X}}$ denotes the imputed gene expression matrix. The mean absolute error was used as the loss function to optimize the imputation task.

For the cell-type annotation task, we use $\tilde{\mathbf{Z}}$ as input to a multi-class classifier. Specifically, we employ a fully connected layer to predict cell-type labels as

$$\hat{\mathbf{Y}} = \mathbf{W}_y\tilde{\mathbf{Z}} + \mathbf{b}_y \tag{18}$$

$$\mathcal{L}_{\text{CE}} = -\frac{1}{M}\sum_{i=1}^{M}\left(\mathbf{Y}_i^\top\log\left(\hat{\mathbf{Y}}_i\right) + (\mathbf{1}-\mathbf{Y}_i)^\top\log\left(\mathbf{1}-\hat{\mathbf{Y}}_i\right)\right), \tag{19}$$

Table 1 Summary of scRNA-seq benchmark datasets, sequencing platforms, and basic statistics.

Dataset	Sequencing platform	# of cells	# of genes	# of cell types
BCs-PCs [20]	10x 5′ v2	3912	19 283	6
Chen [21]	10x 3′ v3	36 313	20 043	15
Emont [22]	10x 3′ v3	7632	24 811	4
Global [20]	10x 5′ v2	10 831	19 283	31
HTAPP-213-SMP-6752 [23]	10x 3′ v3	9799	24 855	7
HTAPP-313-SMP-932 [23]	10x 3′ v3	12 494	20 938	10
HTAPP-783-SMP-4081 [23]	10x 3′ v2	2276	16 853	8
Quake-Diaphragm [1]	Smart-seq2	870	23 341	5
Quake-Limb Muscle [1]	Smart-seq2	1090	23 341	6
Quake-Lung [1]	Smart-seq2	1676	23 341	11
Quake-Trachea [1]	Smart-seq2	1350	23 341	4
Zeisel [24]	STRT-seq UMI	3005	19 972	9
Mouse ES [25]	inDrop	2717	24 175	4
Pancreas [26]	inDrop	1937	15 575	14
AMB3 [27]	SMART-Seq v4	12 832	42 625	3
AMB16 [27]	SMART-Seq v4	12 832	42 625	16
AMB92 [27]	SMART-Seq v4	12 832	42 625	92
TM [1]	SMART-Seq2	54 865	19 791	55
Zheng [28]	10x Chromium	65 943	20 387	11

where $\mathbf{W}_y$ and $\mathbf{b}_y$ are trainable parameters, and $\hat{\mathbf{Y}}$ denotes the predicted cell-type labels. The cross-entropy loss was used as the loss function to optimize the cell-type annotation task.

Results

Benchmark datasets and baselines

We evaluated **GatorSC** on three key single-cell analysis tasks, including cell clustering, gene expression imputation, and automated cell-type annotation, using multiple publicly available scRNA-seq datasets spanning diverse tissues, species, and experimental platforms [1, 20–28], as shown in Table 1. **Details on data preprocessing, implementation and parameter settings, evaluation metrics, visualization results, and additional experimental results, as well as ablation studies and hyperparameter sensitivity analyses, are provided in the Supplementary Materials.**

Across clustering, imputation, and annotation, we benchmarked **GatorSC** against representative deep generative/autoencoder models (DESC [6], scDeepCluster [5], scVI [4], DeepBID [7], DCA [29], AutoClass [30]), graph-based methods (scGNN [8], graph-sc [9], scDSC [10], GE-Impute [31], MAGIC [32], scBFP [33]) and graph contrastive learning methods (scGPCL [11], scSimGCL [12]), classical clustering algorithms (KMeans, Leiden [34], Louvain [35]), and supervised/atlas-based annotators (scHeteroNet [36], scDeepSort [37], ACTINN [38], CellTypist [39], scANVI [40]).

Cell clustering performance on 14 benchmark scRNA-seq datasets

We evaluated **GatorSC** on the cell clustering task using 14 benchmark scRNA-seq datasets spanning diverse tissues, species, and sequencing platforms. As summarized in Fig. 2, **GatorSC** consistently outperforms representative deep generative, graph-based, and conventional clustering methods across these datasets. Across four metrics, adjusted Rand index (ARI), normalized mutual information (NMI), clustering accuracy (ACC), and homogeneity, **GatorSC** achieves the best or second-best scores. The advantage is especially notable on heterogeneous datasets with many cell types, such as Global (31 cell types) and Pancreas (14 cell types). Graph-based baselines, such as scGPCL and scSimGCL, often outperform purely autoencoder-based approaches (DESC, scDeepCluster, DeepBID), highlighting the importance of explicitly modeling cell–cell structure. Nevertheless, no single baseline dominates across all datasets or metrics, whereas **GatorSC** remains among the top-performing methods in most cases. In contrast, classical clustering algorithms (KMeans, Leiden, and Louvain) generally underperform and show larger performance fluctuations. Overall, these results suggest that the hierarchical graph modeling and MoE fusion in **GatorSC** yield more discriminative and biologically meaningful representations by integrating global cell–cell structure, global gene–gene relationships, and local context-specific gene–gene interactions, leading to robust performance across diverse settings.

Gene expression imputation performance on 14 benchmark scRNA-seq datasets

We evaluated the gene expression imputation performance of **GatorSC** on the same 14 benchmark scRNA-seq datasets under simulated dropout rates from 10% to 50%. As shown in Fig. 3, **GatorSC** consistently achieves higher Pearson correlation coefficients (PCC) and lower L1 errors between imputed and ground-truth expression than competing denoising and imputation methods across most datasets and dropout levels. Improvements are observed in relatively simple datasets with few cell types (e.g. Emont, Mouse ES) and highly heterogeneous datasets such as Global and Pancreas, indicating that **GatorSC** can recover expression signals under complex population structure. Notably, the advantage becomes more pronounced as dropout increases: at mild dropout (10%–20%), scSimGCL is competitive on several datasets, whereas under severe dropout (40%–50%) many baselines degrade substantially, particularly in PCC, while **GatorSC** remains relatively stable across datasets. These results suggest that the hierarchical graph construction and MoE fusion enable **GatorSC** to leverage both local and global transcriptomic structure to infer missing values, even when the expression matrix is highly sparse. Overall, **GatorSC** serves as a strong clustering model

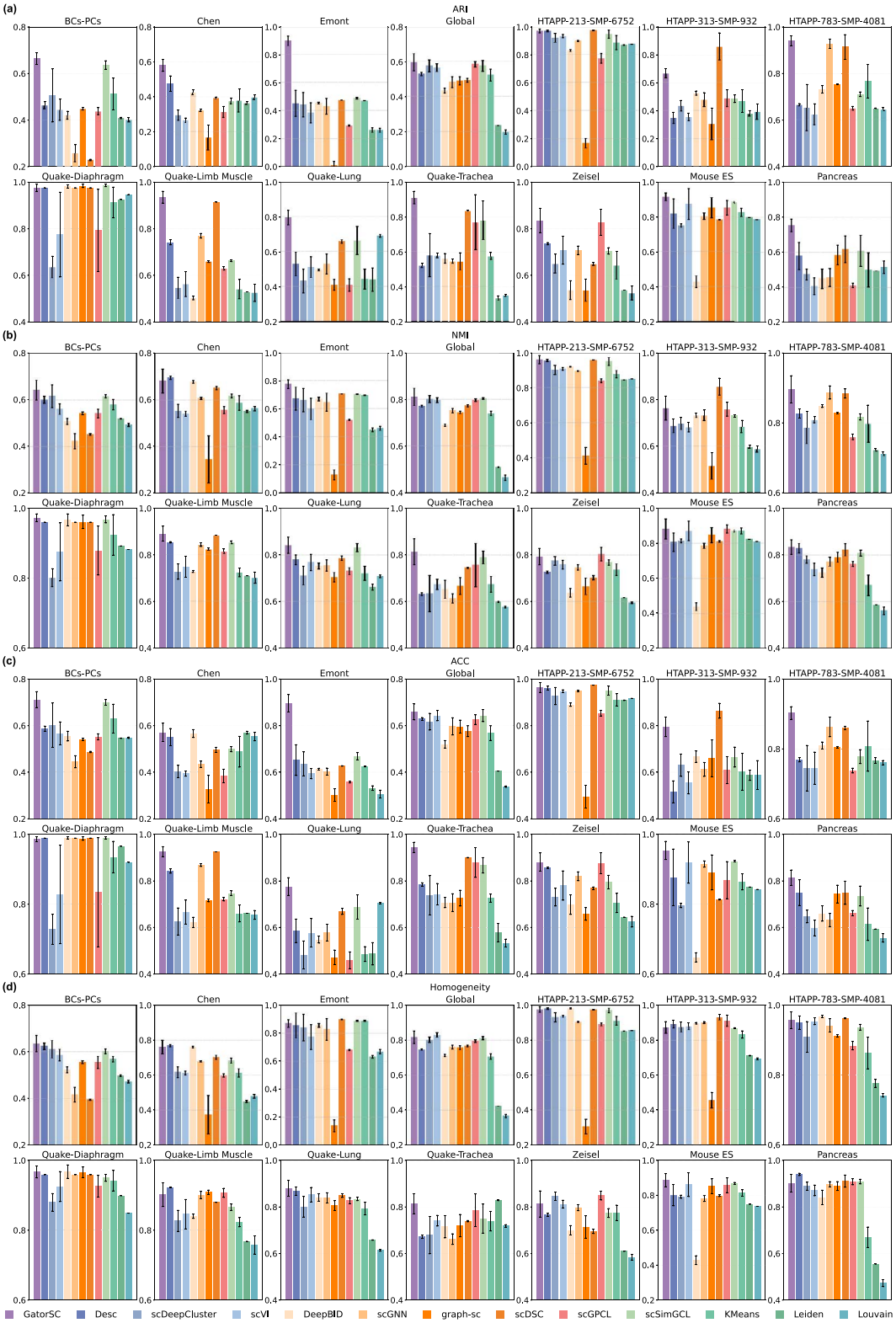

Figure 2 Clustering performance of GatorSC and baseline methods on 14 benchmark scRNA-seq datasets, with rows showing ARI, NMI, ACC, and homogeneity scores, where higher values indicate better clustering performance.

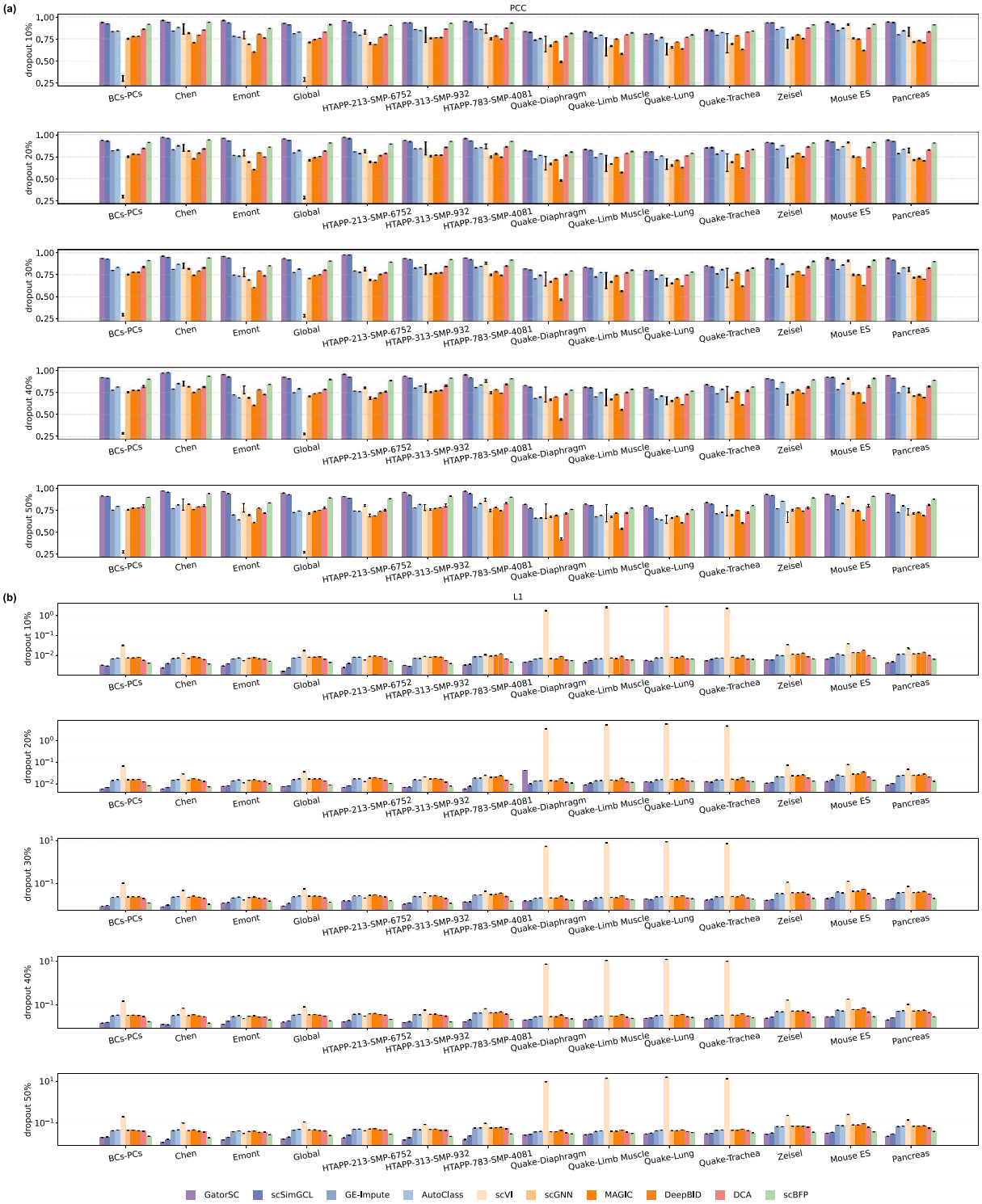

Figure 3 Gene expression imputation performance of GatorSC and competing methods on 14 scRNA-seq datasets under simulated dropout rates from 10% to 50%, with the top panels reporting PCC between imputed and ground-truth expression and the bottom panels reporting L1 errors, where higher PCC and lower L1 values indicate better imputation performance.

Figure 4 Cell-type annotation performance of GatorSC and competing methods on 14 scRNA-seq datasets, with bars showing F1 score, overall accuracy, and MCC, where higher values indicate better annotation performance.

and an effective imputation method, learning representations that preserve global correlation patterns while accurately reconstructing local expression profiles.

Cell-type annotation performance on 14 benchmark scRNA-seq datasets

We further evaluated **GatorSC** on the cell-type annotation task using 14 benchmark scRNA-seq datasets with ground-truth labels. As shown in Fig. 4, **GatorSC** consistently matches or outperforms widely used supervised and semi-supervised classifiers, including scHeteroNet, ACTINN, scDeepSort, CellTypist, and scANVI. Across three metrics, F1 score, overall accuracy (ACC), and Matthews correlation coefficient (MCC), **GatorSC** achieves the best or near-best performance on most datasets, suggesting that the learned representations transfer well to downstream classification. Compared with strong graph-based baselines, scHeteroNet attains solid performance on several datasets, yet **GatorSC** provides consistent gains on many of them, particularly on heterogeneous datasets such as Global and Pancreas. On simpler datasets with only a few broad cell types (e.g. BCs-PCs

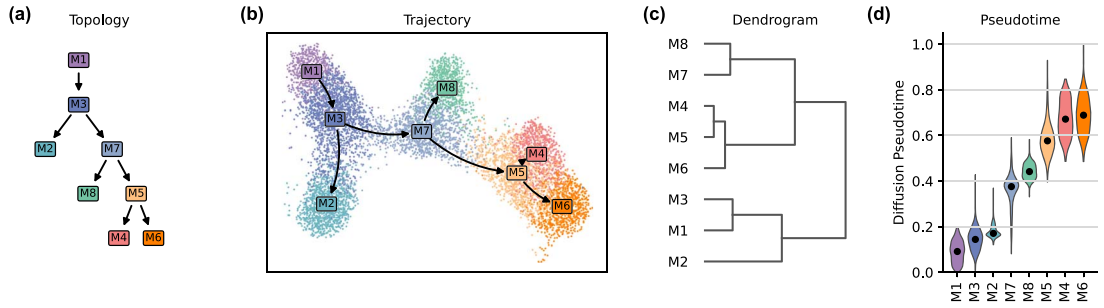

Figure 5 Trajectory inference on the binary_tree_8 benchmark dataset by GatorSC, including the inferred directed tree of eight meta-states, the learned lowdimensional cell trajectory colored by meta-state, hierarchical clustering of meta-states based on transcriptomic similarity, and diffusion pseudotime distributions showing progression from early to late states.

and Emont), **GatorSC** performs comparably to the best baselines. These results indicate that the hierarchical graph modeling and MoE fusion in **GatorSC** yield representations that are effective for both unsupervised clustering and supervised annotation. By jointly leveraging cell–cell and gene–gene structure in a unified self-supervised framework, **GatorSC** supports accurate and robust cell-type labeling across diverse single-cell transcriptomic studies.

Trajectory inference on the `binary_tree_8` benchmark dataset

Figure 5 summarizes the developmental trajectory inferred by **GatorSC** from the latent representation of the `binary_tree_8` single-cell transcriptomes [41]. **GatorSC** first aggregates cells into eight meta-states (M1–M8) and then orders both meta-states and individual cells along a branching developmental continuum. In the Topology panel, the meta-states are arranged as a directed tree that captures the global branch structure of the trajectory. M1 is identified as an upstream progenitor-like state that flows into M3 and subsequently bifurcates into several branches. From M3, one branch terminates in M2, whereas another proceeds through M7 to M5, which further splits into the terminal states M4 and M6; a separate branch from M3 ends in M8. This topology is consistent with the underlying design of the `binary_tree_8` benchmark and highlights the main decision points and terminal fates inferred by **GatorSC**.

In the Trajectory panel, each point represents a single cell in the low-dimensional manifold and colors denote the inferred meta-states. The black curve traces the principal trajectory learned by **GatorSC**, with arrows indicating progression from M1 toward downstream branches. Cells form continuous clouds that smoothly connect neighboring meta-states along each branch, and branch junctions are populated by cells assigned to intermediate states such as M3 and M7. This spatial organization indicates that the discrete meta-state structure is coherent with an underlying continuous manifold of cell states rather than being driven by isolated clusters.

The Dendrogram panel depicts the hierarchical relationships among meta-states based on their transcriptomic similarity. Early states (M1, M3, and M2) cluster together and are clearly separated from the group comprising late states (M5, M4, and M6), whereas M7 and M8 form an intermediate branch. This clustering pattern closely mirrors the branching structure in the topology panel, supporting the view that the inferred branches correspond to genuine transcriptional diversification of cell populations.

The Pseudotime panel shows the distribution of diffusion pseudotime values for each meta-state. Early meta-states (M1, M3, and M2) exhibit low pseudotime values, intermediate states (M7 and M8) occupy central pseudotime ranges, and late states (M5, M4, and M6) concentrate at high pseudotime values. The monotonic increase in median pseudotime from M1 to downstream meta-states, with limited overlap between early and late distributions, suggests that **GatorSC** recovers a temporally consistent ordering of cell states that is aligned with the inferred trajectory topology.

These results demonstrate that **GatorSC** reconstructs a developmental continuum consistent with the known branching structure of the `binary_tree_8` dataset, in which an upstream progenitor-like state (M1) progresses through intermediate states and diverges into multiple terminal branches. The agreement between topology, low-dimensional embedding, hierarchical relationships, and pseudotime supports the robustness of the inferred trajectory.

Marker gene analysis on the Quake Smart-seq2 diaphragm dataset

To evaluate whether **GatorSC** recovers biologically meaningful structure, we examined the expression of canonical marker genes across major cell populations in the Quake Smart-seq2 diaphragm dataset [1] (Fig. 6). In the dot plot (Fig. 6a), a curated panel of lineage-specific markers is shown for **GatorSC**-annotated skeletal muscle satellite stem cells, mesenchymal stem cells, endothelial cells, macrophages, and lymphocytes. Dot size encodes the fraction of cells in each group expressing a given gene, and dot color represents the mean expression level. Stromal and mesenchymal markers such as MFAP5, CLEC3B, COL1A2, LUM, and DCN are enriched in the mesenchymal stem cell population, whereas endothelial-associated genes including CD36, FABP4, GPIHBP1, and CDH5 show selective expression in the endothelial compartment. Myeloid markers such as FCER1G, CCL9, CTSC, CTSS, and RAC2 are concentrated in the macrophage cluster, while the lymphocyte marker CD52 is largely restricted to the lymphocyte population, indicating largely lineage-restricted expression patterns.

To provide complementary summaries of these markers, violin plots (Fig. 6b) depict the distribution of expression for representative genes across the five **GatorSC**-defined cell types, and a heatmap of normalized marker scores (Fig. 6c) aggregates signal across the full marker panel. In each case, the corresponding cell type shows higher expression and a larger fraction of marker-positive cells than unrelated populations, which show low signal. Finally, a 2D embedding of the **GatorSC** representation (Fig. 6d), with cells colored by inferred label, reveals well-separated clusters corresponding to the five major cell types. These results demonstrate that **GatorSC** yields coherent

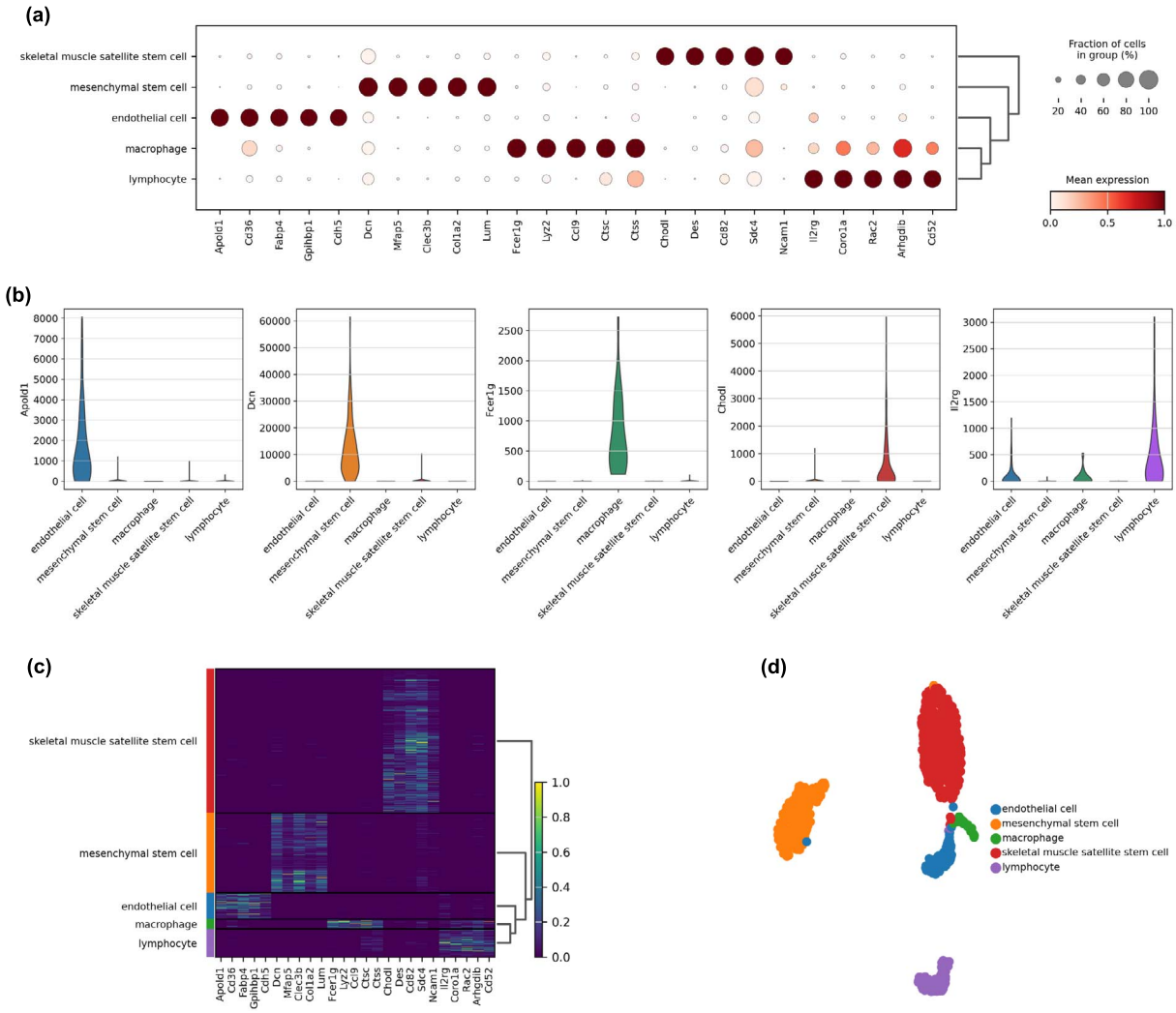

Figure 6 Marker gene analysis in the Quake Smartseq2 diaphragm dataset, including (a) a dot plot of canonical marker genes across five GatorSC-annotated cell types, (b) violin plots of representative marker gene expression, (c) a heatmap of normalized marker scores, and (d) a UMAP visualization of the GatorSC latent representation colored by inferred cell-type labels.

cell-type annotations that recapitulate known skeletal muscle and immune cell biology and support marker-based downstream interpretation of single-cell data.

Application to Alzheimer's disease single-nucleus RNA-seq dataset

To demonstrate its practical utility on real disease data, we applied **GatorSC** to a single-nucleus RNA-seq dataset from human entorhinal cortex (GEO: GSE138852) comprising 13 214 nuclei profiled from six individuals with AD and six age-matched controls [42]. In the learned latent space, **GatorSC** robustly segregates cells into five major brain cell populations, microglia, neurons, oligodendrocyte progenitor cells (OPCs), astrocytes, and oligodendrocytes (Fig. 7a), in agreement with the original annotation of this dataset. The corresponding UMAP embedding shows that AD and control nuclei are well intermingled within each cell type, indicating that **GatorSC** captures biologically meaningful cell identities rather than trivially separating samples by disease status.

Within each **GatorSC**-defined cell type, we then contrasted AD and control nuclei and performed pathway enrichment analysis (Fig. 7b). Across all five cell populations, AD cells show strong positive enrichment for oxidative phosphorylation and for hallmark neurodegenerative disease pathways, including those associated with Alzheimer's disease, Parkinson's disease, Huntington's disease, and amyotrophic lateral sclerosis. In addition, we observe pronounced cell-type-specific modulation of neuronal signaling pathways, such as axon guidance, long-term potentiation and depression, calcium signaling, and regulation of the actin cytoskeleton, indicating widespread perturbation of synaptic and neurodevelopmental programs in AD.

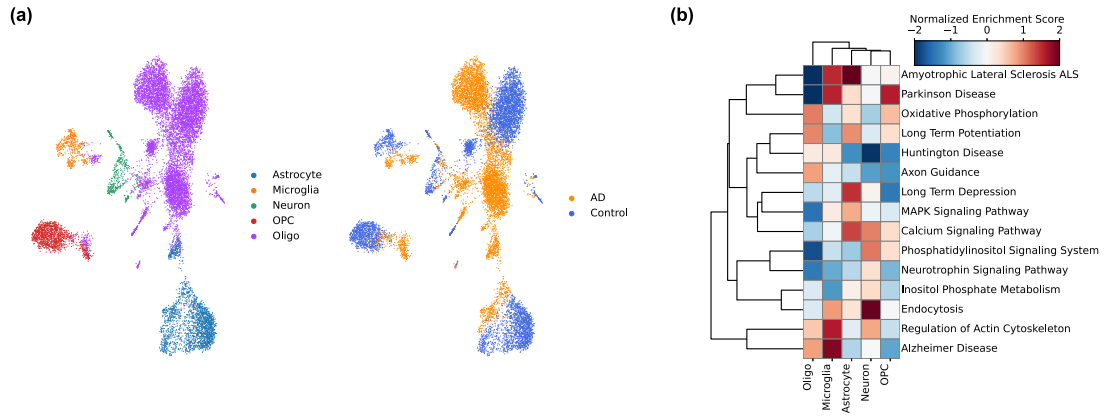

Figure 7 Application of GatorSC to an Alzheimer's disease single-nucleus RNA-seq dataset, including (a) UMAP visualizations colored by GatorSC-inferred cell clusters and disease status, and (b) a heatmap of normalized enrichment scores for selected Alzheimer's disease-related pathways comparing Alzheimer's disease and control nuclei across major cell types.

Notably, the MAPK (mitogen-activated protein kinase) signaling pathway shows a negative enrichment specifically in oligodendrocytes, whereas it is weakly perturbed or positively enriched in the other cell types (Fig. 7b). This pattern suggests a relative suppression of MAPK-mediated regulation in myelinating oligodendrocytes compared with neurons, astrocytes, microglia, and OPCs, pointing to a distinct regulatory response of the oligodendrocyte lineage in AD. Together, these results demonstrate that **GatorSC** can be effectively applied to complex human snRNA-seq datasets to recover major brain cell classes and to dissect cell-type-specific pathway alterations and regulatory programs associated with Alzheimer's disease.

Conclusion

We introduced **GatorSC**, a unified representation learning framework for scRNA-seq that models multi-scale cell and gene structures with hierarchical graphs and fuses complementary views via a MoE architecture. Trained with a dual self-supervised objective combining graph reconstruction and contrastive learning, **GatorSC** learns robust, biologically meaningful cell embeddings without task-specific supervision. Across 19 public datasets, **GatorSC** achieves consistently state-of-the-art performance on cell clustering, gene expression imputation, and cell-type annotation, and scales well to large, deeply annotated atlases while preserving fine-grained cellular structure. Beyond quantitative benchmarks, biological analyses (trajectory inference, marker-gene analysis, and a disease case study) further support the biological fidelity and interpretability of the learned embeddings, and ablation and sensitivity studies validate the contributions of hierarchical gene graphs, adaptive MoE fusion, and unified self-supervision. While **GatorSC** shows strong performance across diverse public datasets, the current study focuses primarily on scRNA-seq data, and extension to other modalities such as spatial transcriptomics and multi-omics remains future work. In addition, the hierarchical graph construction and MoE fusion increase model complexity and may introduce additional computational cost for very large datasets.

Key Points

- GatorSC represents scRNA-seq with three hierarchical graphs to capture cell–cell and gene–gene structure at global and local scales.
- GatorSC uses a mixture-of-experts gating network to fuse multi-graph embeddings, trained with self-supervised reconstruction and contrastive learning.
- Across 19 public datasets, GatorSC delivers state-of-the-art clustering, imputation, and annotation, and supports trajectory, marker, and pathway analyses.

Author contributions

Yuxi Liu (Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Writing—original draft), Zhenhao Zhang (Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing—original draft), Mufan Qiu (Formal analysis, Software, Validation), Song Wang (Writing—original draft), Flora D. Salim (Writing—review & editing), Jun Shen (Writing—review & editing), Tianlong Chen (Writing—review & editing), Imran Razzak (Writing—review & editing), Fuyi Li (Conceptualization, Validation, Writing—review & editing), and Jiang Bian (Conceptualization, Funding acquisition, Project administration, Resources, Supervision, Writing—review & editing)

Supplementary material

Supplementary material is available at *Briefings in Bioinformatics* online.

Conflicts of interest

The authors declare that they have no competing interests.

Funding

F.L. is supported by the Australia National Health and Medical Research Council (NHMRC) Investigator Fellowship (GNT2041439).

Data availability

The source code of **GatorSC** is available at https://github.com/zhangzh1328/GatorSC.

References

1. Schaum N, Karkanias J, Neff NF *et al.* Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris: the Tabula Muris Consortium. *Nature* 2018;**562**:367.
2. Alemany A, Florescu M, Baron CS *et al.* Whole-organism clone tracing using single-cell sequencing. *Nature* 2018;**556**:108–12.
3. Chung W, Eum HH, Lee H-O *et al.* Single-cell RNA-seq enables comprehensive tumour and immune cell profiling in primary breast cancer. *Nat Commun* 2017;**8**:15081.
4. Lopez R, Regier J, Cole MB *et al.* Deep generative modeling for single-cell transcriptomics. *Nat Methods* 2018;**15**:1053–8.
5. Tian T, Wan J, Song Q *et al.* Clustering single-cell RNA-seq data with a model-based deep learning approach. *Nat Mach Intell* 2019;**1**: 191–8.
6. Li X, Wang K, Lyu Y *et al.* Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. *Nat Commun* 2020;**11**:2338.
7. Qin L, Zhang G, Zhang S *et al.* Deep batch integration and denoise of single-cell RNA-seq data. *Adv Sci* 2024;**11**:2308934.
8. Wang J, Ma A, Chang Y *et al.* scGNN is a novel graph neural network framework for single-cell RNA-seq analyses. *Nat Commun* 2021;**12**:1882.
9. Ciortan M, Defrance M. GNN-based embedding for clustering scRNA-seq data. *Bioinformatics* 2022;**38**:1037–44.
10. Gan Y, Huang X, Zou G *et al.* Deep structural clustering for single-cell RNA-seq data jointly through autoencoder and graph neural network. *Brief Bioinform* 2022;**23**:bbac018.
11. Lee J, Kim S, Hyun D *et al.* Deep single-cell RNA-seq data clustering with graph prototypical contrastive learning. *Bioinformatics* 2023;**39**:btad342.
12. Zhang Z, Liu Y, Xiao M *et al.* Graph contrastive learning as a versatile foundation for advanced scRNA-seq data analysis. *Brief Bioinform* 2024;**25**:bbae558.
13. Cai W, Jiang J, Wang F *et al.* A survey on mixture of experts in large language models. *IEEE Trans Knowl Data Eng* 2025;**37**:3896–915.
14. Wang H, Jiang Z, You Y *et al.* Graph mixture of experts: learning on large-scale graphs with explicit diversity modeling. *Adv Neural Inf Proces Syst* 2023;**36**:50825–37.
15. Hou Z, Liu X, Cen Y *et al.* GraphMAE: self-supervised masked graph autoencoders. In: *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 594–604. Washington DC, USA: Association for Computing Machinery, 2022.
16. Liu Y, Zhang Z, Qin S *et al.* Contrastive learning-based imputation-prediction networks for in-hospital mortality risk modeling using EHRs. In: De Francisci Morales Gianmarcoand Perlich, Claudiaand Ruchansky, Nataliand Kourtellis, Nicolasand Baralis, Elenaand Bonchi, Francesco (eds.), *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 428–43. Cham: Springer Nature Switzerland, 2023.
17. Zhang Z, Liu Y, Bian J *et al.* Boosting patient representation learning via graph contrastive learning. In: Bifet, Albertand Krilavičius, Tomasand Miliou, Ioannaand Nowaczyk, Slawomir (eds.), *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 335–50. Cham: Springer, 2024.
18. Liu Y, Zhang Z, Mi J *et al.* GatorCLR: personalized predictions of patient outcomes on electronic health records using self-supervised contrastive graph representation. *J Biomed Inform* 2025;**168**:104851.
19. Wang S, Liu Y, Zhang Z *et al.* GatorST: a versatile contrastive meta-learning framework for spatial transcriptomic data analysis. *Small Methods* 2026;**10**:e00006.
20. Yisu G, Bartolomé-Casado R, Chuan X *et al.* Immune microniches shape intestinal T_{reg} function. *Nature* 2024;**628**:854–62.
21. Chen C, Wenbao Y, Alikarami F *et al.* Single-cell multiomics reveals increased plasticity, resistant populations, and stem-cell–like blasts in KMT2A-rearranged leukemia. *Blood* 2022;**139**:2198–211.
22. Emont MP, Jacobs C, Essene AL *et al.* A single-cell atlas of human and mouse white adipose tissue. *Nature* 2022;**603**:926–33.
23. Klughammer J, Abravanel DL, Segerstolpe Å *et al.* A multimodal single-cell and spatial expression map of metastatic breast cancer biopsies across clinicopathological features. *Nat Med* 2024;**30**:3236–49.
24. Zeisel A, Muñoz-Manchado AB, Codeluppi S *et al.* Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* 2015;**347**:1138–42.
25. Klein AM, Mazutis L, Akartuna I *et al.* Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* 2015;**161**:1187–201.
26. Luecken MD, Büttner M, Chaichoompu K *et al.* Benchmarking atlas-level data integration in single-cell genomics. *Nat Methods* 2022;**19**:41–50.
27. Tasic B, Yao Z, Graybuck LT *et al.* Shared and distinct transcriptomic cell types across neocortical areas. *Nature* 2018;**563**:72–8.
28. Zheng GXY, Terry JM, Belgrader P *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat Commun* 2017;**8**:14049.
29. Eraslan G, Simon LM, Mircea M *et al.* Single-cell RNA-seq denoising using a deep count autoencoder. *Nat Commun* 2019;**10**:390.
30. Li H, Brouwer CR, Luo W. A universal deep neural network for in-depth cleaning of single-cell RNA-seq data. *Nat Commun* 2022;**13**:1901.
31. Xiaobin W, Zhou Y. GE-Impute: graph embedding-based imputation for single-cell RNA-seq data. *Brief Bioinform* 2022;**23**:bbac313.
32. Van Dijk D, Sharma R, Nainys J *et al.* Recovering gene interactions from single-cell data using data diffusion. *Cell* 2018;**174**:716–29.
33. Lee J, Yun S, Kim Y *et al.* Single-cell RNA sequencing data imputation using bi-level feature propagation. *Brief Bioinform* 2024;**25**:bbae209.
34. Traag VA, Waltman L, Van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. *Sci Rep* 2019;**9**:1–12.
35. Levine JH, Simonds EF, Bendall SC *et al.* Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. *Cell* 2015;**162**:184–97.
36. Liu J, Fan X, Chunbin G *et al.* scHeteroNet: aheterophily-aware graph neural network for accurate cell type annotation and novel cell detection. *Adv Sci* 2025;**12**:2412095.
37. Shao X, Yang H, Zhuang X *et al.* scDeepSort: a pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network. *Nucleic Acids Res* 2021;**49**:e122–2.
38. Ma F, Pellegrini M. ACTINN: automated identification of cell types in single cell RNA sequencing. *Bioinformatics* 2020;**36**:533–8.
39. Domínguez Conde C, Chao X, Jarvis LB *et al.* Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science* 2022;**376**:eabl5197.
40. Chenling X, Lopez R, Mehlman E *et al.* Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Mol Syst Biol* 2021;**17**:e9620.
41. Saelens W, Cannoodt R, Todorov H *et al.* A comparison of single-cell trajectory inference methods. *Nat Biotechnol* 2019;**37**: 547–54.
42. Grubman A, Chew G, Ouyang JF *et al.* A single-cell atlas of entorhinal cortex from individuals with Alzheimer's disease reveals cell-type-specific gene expression regulation. *Nat Neurosci* 2019;**22**:2087–97.

Briefings in Bioinformatics, 2026, **27**, bbag330
https://doi.org/10.1093/bib/bbag330
Problem Solving Protocol