GatorSC: Multi-Scale Cell and Gene Graphs with Mixture-of-Experts Fusion for Single-Cell Transcriptomics

Yuxi Liu [a,1,*], Zhenhao Zhang[b,1], Mufan Qiu[f], Song Wang[c], Flora D. Salim[d], Jun Shen[e], Tianlong Chen[f,*], Imran Razzak[g,*], Jiang Bian[a,*]

[a] *Biostatistics and Health Data Science, School of Medicine, Indiana University, Indianapolis, 46202, IN, USA*
[b] *College of Life Sciences, Northwest A&F University, Yangling, 712100, Shaanxi, China*
[c] *Department of Computer Science, University of Central Florida, Orlando, 32816, FL, USA*
[d] *School of Computer Science and Engineering, University of New South Wales, Sydney, 2052, NSW, Australia*
[e] *School of Computing and Information Technology, University of Wollongong, Wollongong, 2522, NSW, Australia*
[f] *Department of Computer Science, The University of North Carolina at Chapel Hill, Chapel Hill, 27599, NC, USA*
[g] *Department of Computational Biology, Muhammad bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE*

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.

[*] Corresponding author
Email address: tianlong@cs.unc.edu; Imran.Razzak@mbzuai.ac.ae; yuxiliu@iu.edu; bianji@iu.edu (Jiang Bian)
[1] Equal contribution

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. Across 14 benchmark datasets, **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, Gene expression imputation, Cell type annotation

---

1. 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 contains 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 how to combine heterogeneous graph views into a single, robust representation of each cell.

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 graph neural network 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.

Our contributions are several-fold as follows:

- We treat representation learning for scRNA-seq data as a multi-scale graph information fusion problem and propose **GatorSC**, a unified framework that jointly models a global cell–cell graph, a global gene–gene graph, and a local gene–gene graph derived from neighborhood-specific subgraphs to provide an integrated view of cell populations and gene programs.

- We introduce a modality-specific MoE fusion module that assigns one expert to each graph-derived embedding and uses a gating network to adaptively combine their outputs on a per-cell basis, enabling data-driven fusion of heterogeneous structural signals across scales.

- We develop a unified self-supervised learning objective that couples graph reconstruction with graph contrastive learning on both cell and gene graphs, yielding label-free training signals that jointly preserve structural fidelity, denoise expression, and enforce semantic consistency in the latent space.

- We conduct extensive experiments on 19 publicly available scRNA-seq datasets across three core tasks: cell clustering, gene expression imputation, and cell-type annotation, and further assess **GatorSC** in downstream biological analyses, including trajectory inference, marker gene analysis, and an application to an Alzheimer's disease single-nucleus RNA-seq cohort. The experimental results demonstrate that GatorSC consistently achieves strong, often state-of-the-art performance while providing biologically interpretable fused embeddings.

2. Related works

The Mixture of Experts (MoE) framework was first introduced in the context of traditional machine learning models [13] and subsequently extended to hierarchical formulations [14]. The core idea of MoE is to decompose a complex prediction task into multiple expert networks whose contributions are dynamically weighted by a gating mechanism. Modern deep-learning MoE architectures build on this principle by activating only a subset of experts for each input, thereby achieving scalable specialization and efficient capacity allocation in large vision and language models [15–17]. In practice, MoE has been utilized in several forms, including sparse MoE layers integrated into Transformer blocks [18, 19], large modular MoE systems composed of independent expert models [20], and multimodal MoE architectures in which different experts are specialized for distinct input modalities [21, 22].

In our proposed method, the three hierarchical graph views, global cell–cell, global gene–gene, and local gene–gene, naturally play the role of distinct modalities, each emphasizing a different level of biological organization. We therefore adopt a modality-specific MoE design, in which one expert network is assigned to each graph-derived representation, and a gating network dynamically determines, for every cell, the relative contribution of each expert to the fused embedding. This architecture is particularly suitable for scRNA-seq data analysis, because the three graph levels encode complementary, non-redundant information at different scales (population-level cell neighborhoods, cross-cell gene dependencies, and context-specific gene dependencies), and the informativeness of these signals can vary across datasets and experimental conditions [23–25]. For example, datasets containing well-separated cell populations are often best captured by global cell–cell structure, whereas datasets dominated by closely related subtypes or gradual developmental transitions rely more heavily on fine-grained local gene–gene dependencies. By allowing each expert to specialize in one graph modality while the gate learns cross-modality interactions in a data-driven manner, the MoE provides an adaptive fusion mechanism that yields unified embeddings offering a robust foundation for downstream analyses, including cell clustering, gene expression imputation, and cell-type annotation.

3. Methods

3.1. 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 Figure 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 captures 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

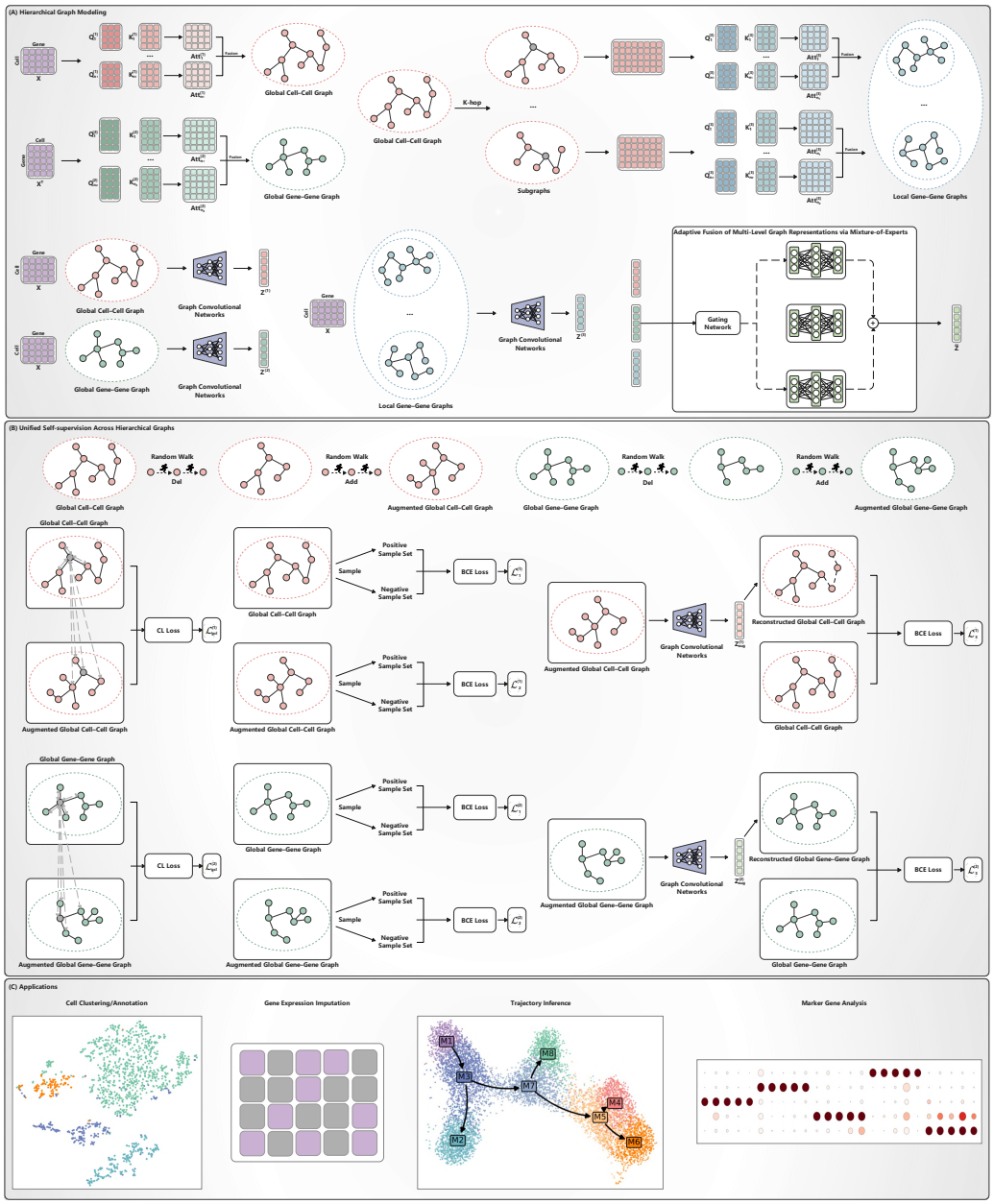

Figure 1: **Overview of the GatorSC framework.**

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.

3.1.1. 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 write

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

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:

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

where all $\mathbf{W}$ is trainable parameters. $\mathrm{head}_i^{(1)}$ is the i-th attention head. 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.

3.1.2. 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 $\mathbf{W}$ is trainable parameters. 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.

3.1.3. 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:

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

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:

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

where all $\mathbf{W}$ is trainable parameters. 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.

3.1.4. Adaptive Fusion of Multi-Level 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) [26] 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. To address this, 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 contributions of the input, while a gating network adaptively controls their weight [27]. 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 [28]. 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 a 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 i-th expert network for a given input $\mathbf{X}$. The aggregated dense representation generated by the MoE can be written as:

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

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 $L = 3$.

3.2. 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 Figure 1B. Self-supervised learning approaches can be broadly categorized into two types [29]. The generative objective focuses on reconstructing the

original signal or graph structure (e.g., graph reconstruction), emphasizing structural fidelity [30, 31]. 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 [32, 33]. This helps the model learn noise-invariant and discriminative representations, improving performance on clustering, classification, and related tasks.

Based on these perspectives, 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:

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

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

Based on these sets, we construct the augmented graph as follows:

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

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. $\mathbf{A}_{\text{aug}}^{(1)}$ is the adjacent matrix of the augmented graph, which we denote by $\mathcal{G}_{\text{aug}}^{(1)}$. Having obtained the original graph $\mathcal{G}^{(1)}$ and its augmented version $\mathcal{G}_{\text{aug}}^{(1)}$,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 $\mathbf{A}_{\text{aug}}^{(1)}$, we define the corresponding positive and negative edge sets as $\mathcal{E}_{\text{aug}}^{(1)+}$, $\mathcal{E}_{\text{aug}}^{(1)-}$, respectively. Based on these positive and negative edge sets, the reconstruction loss can be formulated as follows:

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

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 $\mathcal{G}_{\text{aug}}^{(1)}$. 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. 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:

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

$$\psi_{ij} = \exp\big(\text{Sim}\big(\mathbf{Z}_i^{(1)},\, \mathbf{Z}_j^{(1)}\big)/\tau\big) + \exp\big(\text{Sim}\big(\mathbf{Z}_i^{(1)},\, (\mathbf{Z}_{\text{aug}}^{(1)})_j\big)/\tau\big).$$

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 i-th node in $\mathcal{G}^{(1)}$. $\mathcal{L}\big(\mathbf{Z}^{(1)},\, \mathcal{G}^{(1)}\big)$ 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:

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

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.

3.3. Application tasks

As described above, the model is trained by minimizing the loss function defined in Eq. (15). After training, the matrix $\tilde{\mathbf{Z}}$ in Eq. (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, including cell clustering, gene expression imputation, and cell-type annotation, as illustrated in Figure 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 $\mathbf{W}_x$ 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:

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

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

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.

4. Experiments

4.1. 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, 34–42], as shown in Table 1.

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 [34]	10x 5' v2	3912	19283	6
Chen [35]	10x 3' v3	36313	20043	15
Emont [36]	10x 3' v3	7632	24811	4
Global [34]	10x 5' v2	10831	19283	31
HTAPP-213-SMP-6752 [37]	10x 3' v3	9799	24855	7
HTAPP-313-SMP-932 [37]	10x 3' v3	12494	20938	10
HTAPP-783-SMP-4081 [37]	10x 3' v2	2276	16853	8
Quake-Diaphragm [1]	Smart-seq2	870	23341	5
Quake-Limb Muscle [1]	Smart-seq2	1090	23341	6
Quake-Lung [1]	Smart-seq2	1676	23341	11
Quake-Trachea [1]	Smart-seq2	1350	23341	4
Zeisel [38]	STRT-seq UMI	3005	19972	9
Mouse ES [39]	inDrop	2717	24175	4
Pancreas [40]	inDrop	1937	15575	14
AMB3 [41]	SMART-Seq v4	12832	42625	3
AMB16 [41]	SMART-Seq v4	12832	42625	16
AMB92 [41]	SMART-Seq v4	12832	42625	92
TM [1]	SMART-Seq2	54865	19791	55
Zheng [42]	10x Chromium	65943	20387	11

For clustering, we benchmarked **GatorSC** against a broad spectrum of state-of-the-art scRNA-seq clustering methods. The deep generative and autoencoder-based approaches included DESC [6], scDeepCluster [5], scVI [4], and DeepBID [7], while graph-based methods comprised scGNN [8],

graph-sc [9], and scDSC [10]. We further compared against recent graph contrastive learning frameworks scGPCL [11] and scSimGCL [12], as well as conventional clustering algorithms KMeans, Leiden [43], and Louvain [44].

For imputation, we benchmarked **GatorSC** against a diverse set of state-of-the-art denoising and imputation approaches. Graph-based and graph contrastive learning methods included scSimGCL [12], GE-Impute [45], scGNN [8], MAGIC [46], and scBFP [47], all of which exploit cell–cell or gene–gene graphs to propagate information and recover dropout events. We further considered deep generative and autoencoder-based models, including AutoClass [48], scVI [4], DeepBID [7], and DCA [49], which learn low-dimensional latent representations and distribution-aware reconstructions of the count data. Together, these baselines span both graph- and distribution-based paradigms and represent leading approaches for scRNA-seq data imputation.

For cell-type annotation, we benchmarked **GatorSC** against a panel of widely used supervised and semi-supervised classifiers. Graph neural network–based methods included scHeteroNet [50], which explicitly models heterophily in the cell graph and incorporates a novelty-propagation module for detecting previously unseen cell types, and scDeepSort [51], a pre-trained weighted GNN built on large human and mouse cell atlases. We further compared against ACTINN [52], a feed-forward neural network classifier trained on reference scRNA-seq datasets, and two atlas-based annotation frameworks: CellTypist [53], which uses logistic-regression models trained on a curated immune cell atlas, and scANVI [54], a semi-supervised deep generative model that jointly harmonizes datasets and transfers cell-type labels across experiments.

4.2. Data pre-processing

GatorSC takes the scRNA-seq gene expression matrix as input. The scRNA-seq data are represented as a two-dimensional matrix, where rows correspond to cells, and columns to genes. Each cell is associated with an available label. The matrix is processed using the Scanpy toolkit [55]. Preprocessing involves normalization and logarithmic transformation of the gene expression matrix.

4.3. Implementation and parameter settings

GatorSC was implemented in Python (v3.9.12) with PyTorch (v1.11.2). For each scRNA-seq dataset, we randomly split cells into training/validation/test sets with a ratio of 80%/10%/10%. For hierarchical graph modeling, we used

two attention heads and initialized the learnable thresholds as $\phi^{(1)} = 0.05$, $\phi^{(2)} = 0.02$, and $\phi^{(3)} = 0.03$; the hop number was set to 2. For adaptive fusion of multi-level graph representations, the hidden dimension of both the GCN and MLP was set to 225, and the number of expert networks was set to 3. For unified self-supervision, the Bernoulli deletion and addition probabilities were both set to 0.4, and the maximum path length was set to 2. The temperature was set to $\tau = 0.25$. The loss scaling coefficients were set to $\alpha_1 = 0.55$, $\alpha_2 = 0.53$, and $\alpha_3 = 0.62$; $\beta_1 = 0.90$ and $\beta_2 = 0.85$; $\gamma_1 = 0.76$ and $\gamma_2 = 0.89$; and $\lambda = 0.68$. We trained the model using the Adam optimizer with a learning rate of 10^{-3} and a batch size of 256. For clustering, we applied K-means (scikit-learn default settings) to the learned embeddings, setting the number of clusters to the ground-truth number of cell types for each dataset. For other clustering baselines, we used the default hyperparameters provided by the corresponding packages. All hyperparameters were selected via grid search based on validation performance. All experiments were conducted on a workstation equipped with an NVIDIA RTX 4090 GPU (24 GB memory).

4.4. Evaluation metrics

For cell clustering performance, we utilize adjusted Rand index (ARI), normalized mutual information (NMI), clustering accuracy (ACC), and homogeneity score (Homogeneity) as evaluation metrics. ARI measures the agreement between the predicted and reference clusterings while accounting for randomness, and is therefore particularly useful when the class distributions are unbalanced or when clusters overlap. The ARI between the ground truth $\mathbf{Y}$ and the predicted $\hat{\mathbf{Y}}$ can be written as follows:

$$\mathrm{ARI} = \frac{\sum_{i=1}^{n_{\hat{\mathbf{Y}}}} \sum_{j=1}^{n_{\mathbf{Y}}} \binom{|\hat{\mathbf{Y}}_i \cap \mathbf{Y}_j|}{2} - \left[\sum_{i=1}^{n_{\hat{\mathbf{Y}}}} \binom{|\hat{\mathbf{Y}}_i|}{2} \sum_{j=1}^{n_{\mathbf{Y}}} \binom{|\mathbf{Y}_j|}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_{i=1}^{n_{\hat{\mathbf{Y}}}} \binom{|\hat{\mathbf{Y}}_i|}{2} + \sum_{j=1}^{n_{\mathbf{Y}}} \binom{|\mathbf{Y}_j|}{2}\right] - \left[\sum_{i=1}^{n_{\hat{\mathbf{Y}}}} \binom{|\hat{\mathbf{Y}}_i|}{2} \sum_{j=1}^{n_{\mathbf{Y}}} \binom{|\mathbf{Y}_j|}{2}\right] / \binom{n}{2}}. \tag{20}$$

where $n_{\mathbf{Y}}$ and $n_{\hat{\mathbf{Y}}}$ denote the numbers of clusters in $\mathbf{Y}$ and $\hat{\mathbf{Y}}$, respectively, and n is the total number of cells.

NMI quantifies the mutual dependence between the predicted and ground truth labels, while the normalization alleviates the influence of imbalanced label distributions. The NMI between $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ can be written as follows:

$$\text{NMI} = \frac{\sum_{i=1}^{n_{\hat{\mathbf{Y}}}} \sum_{j=1}^{n_{\mathbf{Y}}} |\hat{\mathbf{Y}}_i \bigcap \mathbf{Y}_j| log \frac{n \times |\hat{\mathbf{Y}}_i \bigcap \mathbf{Y}_j|}{|\hat{\mathbf{Y}}_i| \times |\mathbf{Y}_j|}}{max(-\sum_{i=1}^{n_{\hat{\mathbf{Y}}}} |\hat{\mathbf{Y}}_i| log \frac{|\hat{\mathbf{Y}}_i|}{n}, -\sum_{j=1}^{n_{\mathbf{Y}}} |\mathbf{Y}_j| log \frac{|\mathbf{Y}_j|}{n})}. \tag{21}$$

ACC measures the proportion of correctly assigned cells after optimally matching predicted clusters to ground-truth labels. The ACC between $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ can be written as follows:

$$\text{ACC} = \underset{\phi}{max} \frac{\sum_{i=1}^{n} \mathbb{1}_{[\mathbf{Y}_i = \phi(\hat{\mathbf{Y}}_i)]}}{n}, \tag{22}$$

where ϕ is a mapping from predicted cluster indices to ground-truth labels, and $\mathbb{1}_{[\cdot]}$ denotes the indicator function.

Finally, the homogeneity score evaluates the extent to which each cluster contains only cells from a single ground-truth class as follows:

$$\text{Homogeneity} = \frac{I(\hat{\mathbf{Y}}, \mathbf{Y})}{H(\mathbf{Y})}. \tag{23}$$

where $H(\mathbf{Y})$ is the entropy of $\mathbf{Y}$ and $I(\hat{\mathbf{Y}}, \mathbf{Y})$ is the mutual information between $\hat{\mathbf{Y}}$ and $\mathbf{Y}$. The score ranges from 0 to 1, with 1 indicating perfectly pure clusters (each cluster containing cells from only one class), whereas lower values indicate mixed clusters.

For the gene expression imputation, we adopt widely used metrics: Pearson correlation coefficient (PCC) and L1 loss. Specifically, PCC measures the linear correlation between the imputed values and ground truth expression matrix, with higher scores suggesting better preservation of global structural patterns in the data. The PCC between the ground truth $\mathbf{X}$ and the imputed value $\hat{\mathbf{X}}$ can be written as follows:

$$\text{PCC} = \frac{E(\hat{\mathbf{X}}\mathbf{X}) - E(\hat{\mathbf{X}})E(\mathbf{X})}{\sqrt{E(\hat{\mathbf{X}}^2) - E(\hat{\mathbf{X}})^2} + \sqrt{E(\mathbf{X}^2) - E(\mathbf{X})^2}}, \tag{24}$$

where $E(\cdot)$ is the expected value. The L1 loss captures the average absolute deviation between the predicted and true values, providing a robust measure

of local imputation accuracy. The L1 between $\mathbf{X}$ and $\hat{\mathbf{X}}$ can be written as follows:

$$\mathrm{L1} = \frac{1}{n}\sum_{i=1}^{n} |\hat{\mathbf{X}}_i - \mathbf{X}_i|. \tag{25}$$

For the cell-type annotation, we report the F1 score (F1), classification accuracy (ACC), and Matthews correlation coefficient (MCC), three widely used metrics for evaluating classification performance. Given the numbers of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN), these metrics can be written as:

$$\begin{aligned} \mathrm{F1} &= \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \\ \mathrm{ACC} &= \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}, \\ \mathrm{MCC} &= \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}. \end{aligned} \tag{26}$$

5. Results

5.1. Cell clustering performance on 14 benchmark scRNA-seq datasets

We first evaluated **GatorSC** on the cell clustering task using 14 benchmark scRNA-seq datasets spanning diverse tissues, species, and sequencing platforms. As summarized in Figure 2, **GatorSC** consistently outperforms representative deep generative, graph-based, and conventional clustering methods across these datasets. For all four clustering metrics, adjusted Rand index (ARI), normalized mutual information (NMI), clustering accuracy (ACC), and homogeneity, **GatorSC** generally achieves the highest or second-highest scores on the majority of datasets. 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 perform better than 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 representation-learning–based approaches and exhibit larger performance fluctuations across datasets.

These results indicate that the proposed hierarchical graph modeling and Mixture-of-Experts fusion in **GatorSC** generate more discriminative and biologically meaningful cell representations. By jointly leveraging global cell–cell structure, global gene–gene relationships, and local context-specific gene–gene interactions, **GatorSC** achieves robust clustering performance across diverse experimental settings, without being tailored to a particular platform or tissue type.

5.2. Gene expression imputation performance on 14 benchmark scRNA-seq datasets

We next evaluated the gene expression imputation performance of **GatorSC** on the same panel of 14 benchmark scRNA-seq datasets under simulated dropout rates ranging from 10% to 50%. As shown in Figure 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. The gains are evident both on relatively simple datasets with a small number of cell types (e.g., Emont, Mouse ES) and on highly heterogeneous datasets such as Global and Pancreas, indicating that **GatorSC** can effectively recover expression signals in the presence of complex cell population structure.

Figure 3 further shows that the relative advantage of **GatorSC** becomes more pronounced as the dropout rate increases. At mild dropout levels (10–20%), scSimGCL achieves competitive PCC values on several datasets. However, under more severe dropout (40–50%), the performance of many baselines deteriorates, particularly in terms of PCC, whereas **GatorSC** maintains relatively stable PCC and L1 values across datasets. These results suggest that the hierarchical graph construction and Mixture-of-Experts fusion enable **GatorSC** to exploit both local and global transcriptomic structure to infer missing values, even when the observed expression matrix is highly sparse.

These findings demonstrate that **GatorSC** functions not only as a strong clustering model but also as an effective imputation method for scRNA-seq data. By integrating graph reconstruction and graph contrastive learning objectives during training, **GatorSC** learns representations that preserve global correlation patterns while accurately reconstructing local expression

Figure 2: **Clustering performance of GatorSC and baseline methods on 14 benchmark scRNA-seq datasets. For each dataset, rows show ARI, NMI, ACC, and homogeneity scores for GatorSC and baseline methods; higher values indicate better clustering performance.**

profiles, leading to improved imputation accuracy across a wide range of biological and technical scenarios.

5.3. 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 Figure 4, **GatorSC** consistently matches or outperforms widely used supervised and semi-supervised classifiers, including scHeteroNet, ACTINN, scDeepSort, CellTypist, and scANVI. Across the three evaluation metrics, F1 score, overall accuracy (ACC), and Matthews correlation coefficient (MCC), **GatorSC** generally achieves the highest or near-highest values on the majority of datasets, indicating that the learned representations transfer well to downstream classification.

The comparison with strong graph-based baselines is particularly informative. scHeteroNet is the GNN-based method tailored for cell-type annotation, achieve solid performance on several datasets, yet **GatorSC** still provides consistent gains on many of them, especially on datasets with more heterogeneous cell-type compositions such as Global and Pancreas. On simpler datasets with only a few broad cell types, such as such as BCs-PCs and Emont, **GatorSC** performs on par with the best baselines, showing that it does not sacrifice accuracy on easier tasks.

These results indicate that the hierarchical graph modeling and Mixture-of-Experts fusion in **GatorSC** yield cell representations that are not only informative for unsupervised clustering but also well suited for supervised annotation. By jointly leveraging cell–cell and gene–gene structure in a unified self-supervised framework, **GatorSC** provides a flexible representation that supports accurate and robust cell-type labeling in diverse single-cell transcriptomic studies.

5.4. Cell-type annotation across annotation levels and dataset scales

To further examine the robustness of **GatorSC** for cell-type annotation, we considered datasets that vary in both annotation depth and dataset size. Specifically, we used the Allen Mouse Brain (AMB) dataset, which provides three nested annotation levels with 3, 16, and 92 cell populations (denoted AMB3, AMB16, and AMB92), to study how performance changes as the number of finer-grained cell types within the same dataset increases. This setting allows us to probe whether a model that performs well on coarse labels can still maintain discriminative power when closely related subtypes

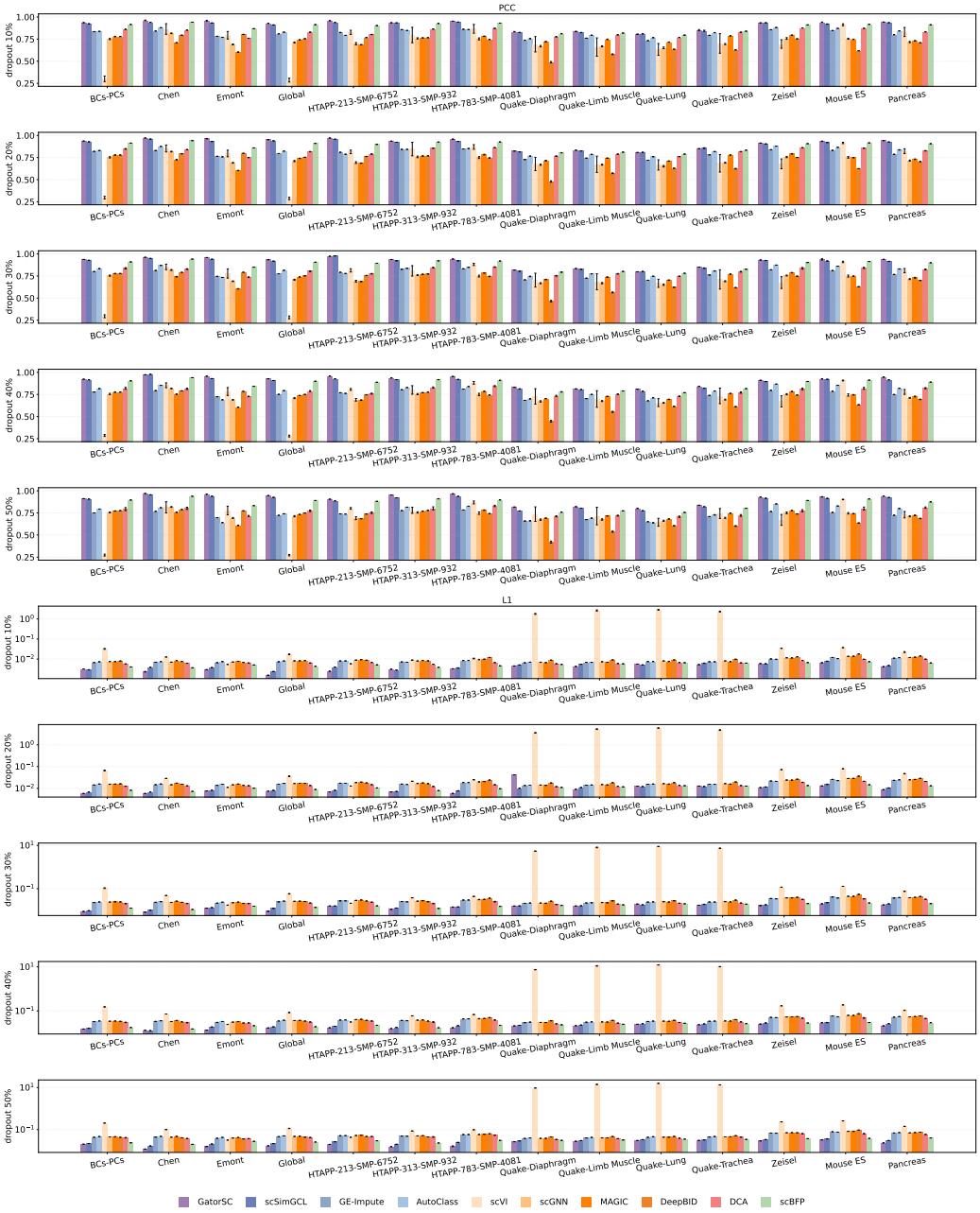

Figure 3: **Gene expression imputation performance of GatorSC and competing methods on 14 scRNA-seq datasets under increasing simulated dropout rates (10–50%). The top panels report Pearson correlation coefficients (PCC) between imputed and ground-truth expression, and the bottom panels report L1 errors. Higher PCC and lower L1 values indicate better imputation performance.**

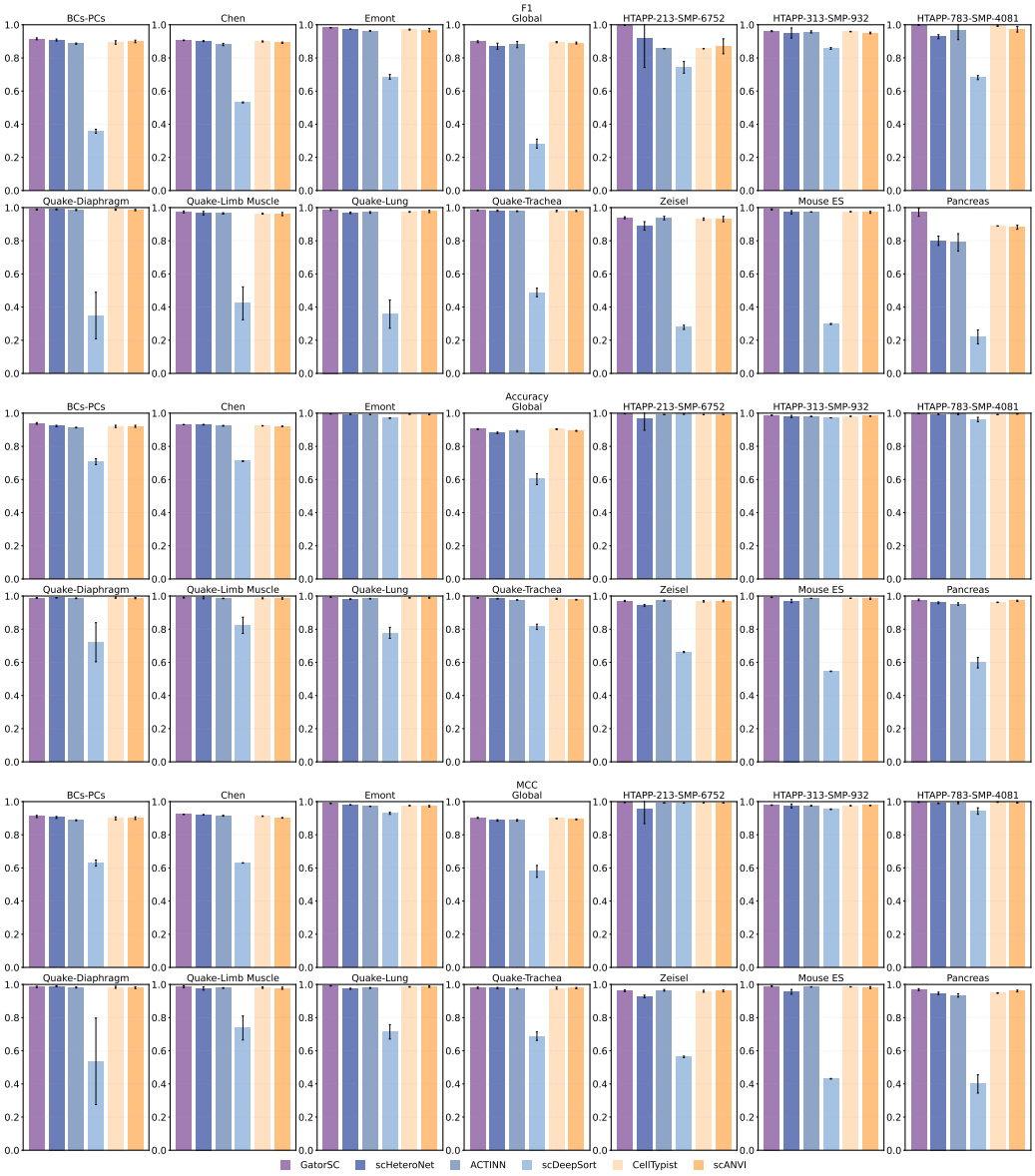

Figure 4: **Cell-type annotation performance of GatorSC and competing methods on 14 scRNA-seq datasets. Bars show F1 score (top row), overall accuracy (middle row), and Matthews correlation coefficient (MCC, bottom row) for GatorSC and competing methods. Higher values of all three metrics indicate better cell-type annotation performance.**

must be distinguished. We included Tabula Muris (TM) and Zheng 68K, two large-scale scRNA-seq datasets with $> 50,000$ cells and complex tissue compositions, to evaluate scalability and stability in high-throughput scenarios. As shown in Figure 5, **GatorSC** achieves state-of-the-art or comparable F1, ACC, and MCC scores across all three AMB annotation levels. Although all methods exhibit a noticeable drop in performance when moving from the coarse AMB3 labels to the most deeply annotated AMB92 setting, **GatorSC** retains a clear advantage or remains among the top-performing approaches, indicating that its representations preserve meaningful structure even under challenging fine-grained label schemes. On the large TM and Zheng 68K datasets, **GatorSC** again matches or outperforms existing classifiers across all three metrics, suggesting that the hierarchical graph representation and Mixture-of-Experts fusion not only support accurate annotation of subtle subpopulations but also scale effectively to datasets with tens of thousands of cells. Collectively, these results demonstrate that **GatorSC** provides a consistent and robust cell-type annotation framework across a spectrum of annotation resolutions and dataset sizes.

*5.5. t-SNE Visualization of Learned Latent Representations for **GatorSC** and Baseline Methods*

Figure 6 presents the t-SNE visualization of the latent embeddings learned by all baseline methods and **GatorSC** on the Emont and Quake-Trachea datasets. For each method, we first extract the intermediate low-dimensional representation of each cell and then apply t-SNE to obtain a two-dimensional embedding, on which cells are colored according to their ground-truth cell-type labels. This allows a direct qualitative comparison of how well different models separate cell types in the latent space. Across both datasets, the clusters generated by **GatorSC** are more compact and clearly separated than those of competing methods. On Emont, **GatorSC** yields distinct islands for most cell types and achieves the highest ARI (0.763), outperforming strong baselines such as scDSC (ARI = 0.715) and scSimGCL (ARI = 0.702). On the Quake-Trachea dataset, **GatorSC** further improves the agreement with ground truth with an ARI of 0.801, again exceeding scSimGCL (ARI = 0.778) and scGPCL (ARI = 0.745).

In contrast, several baselines show noticeable mixing between cell types in the t-SNE space. Methods such as Desc and scDeepCluster form partially entangled clusters, with blurred boundaries between neighboring populations on both datasets, consistent with their moderate ARI values. Latent-variable

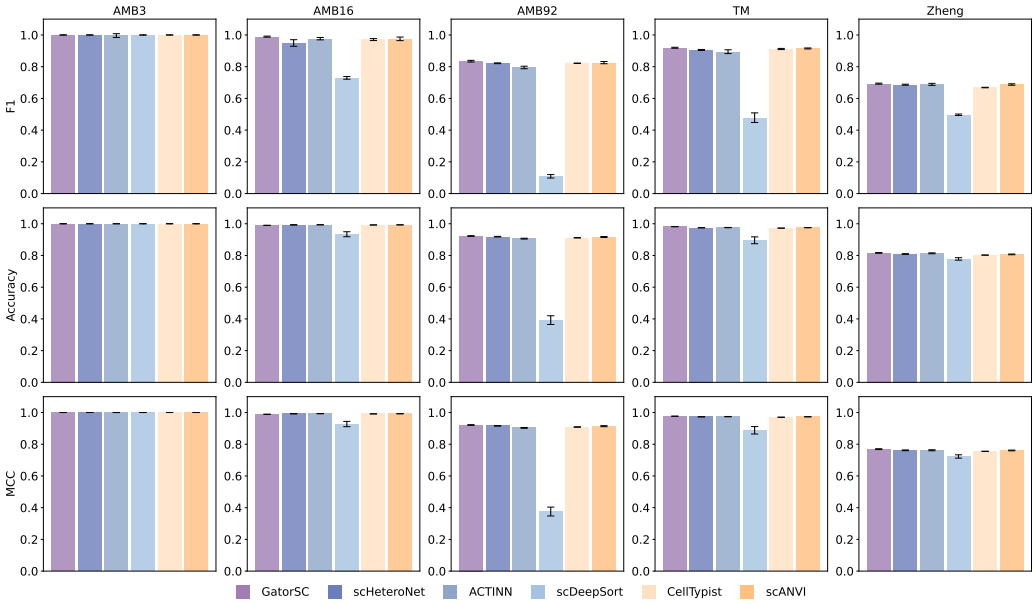

Figure 5: **Cell-type annotation performance of GatorSC and competing methods on five scRNA-seq datasets with varying annotation granularity and dataset size. Columns correspond to the three annotation levels of the Allen Mouse Brain dataset (AMB3, AMB16, AMB92; with 3, 16, and 92 cell populations, respectively), the Tabula Muris (TM) dataset, and the Zheng 68K PBMC dataset. Bars show F1 score (top row), overall accuracy (middle row), and Matthews correlation coefficient (MCC, bottom row) for each method. Across most datasets and metrics, GatorSC achieves state-of-the-art or comparable performance, and higher values of all three metrics indicate better cell-type annotation.**

models including scVI and DeepBID separate some major populations but still show overlap in regions where **GatorSC** produces well-isolated clusters. The graph-based approach graph-sc shows limited separation on Emont, where most cells collapse into a single diffuse cloud, consistent with its lower ARI, and only partially recovers the cluster structure on Quake-Trachea. Even the stronger baselines scDSC, scGPCL, and scSimGCL, while yielding more discernible groups, tend to produce clusters that are less compact and more fragmented than those of **GatorSC**. Taken together, these t-SNE visualizations indicate that **GatorSC** learns a latent space in which biologically meaningful cell populations are more cleanly separated than with existing methods.

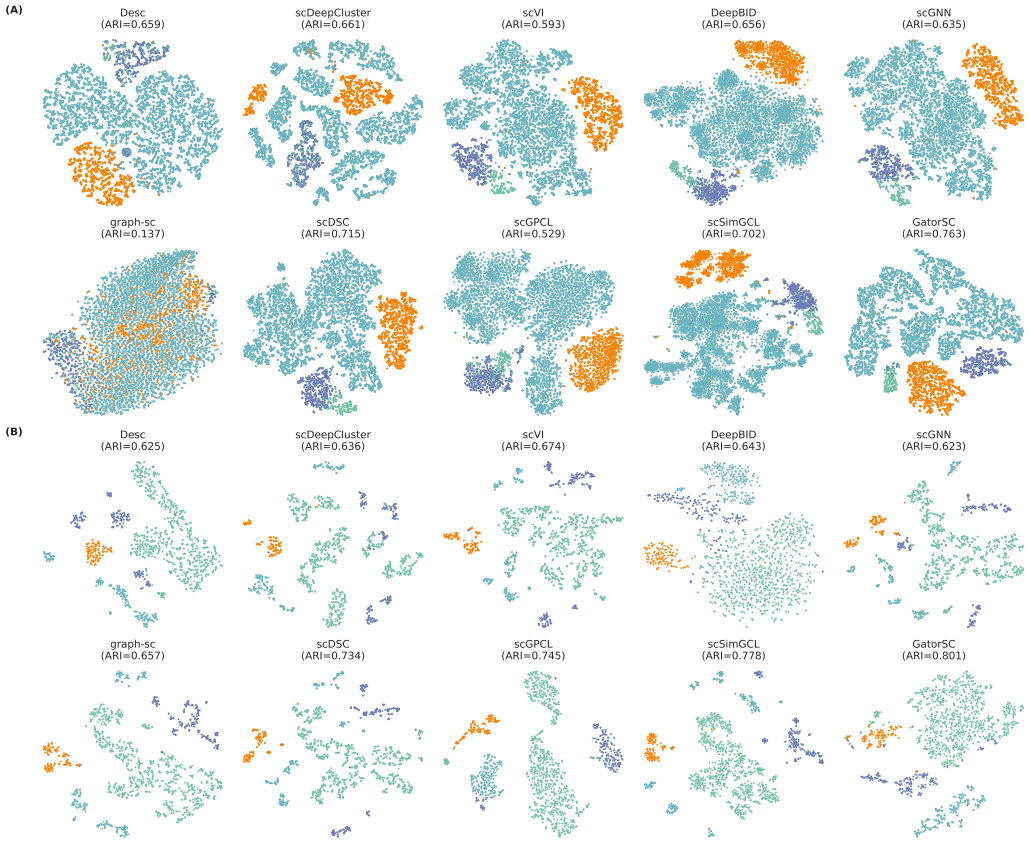

Figure 6: **Visualization analysis results of baselines and GatorSC on the (A) Emont and (B) Quake-Trachea datasets.**

5.6. Trajectory inference on the `binary_tree_8` *benchmark dataset*

Figure 7 summarizes the developmental trajectory inferred by **GatorSC** from the latent representation of the `binary_tree_8` single-cell transcriptomes [56]. **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**.

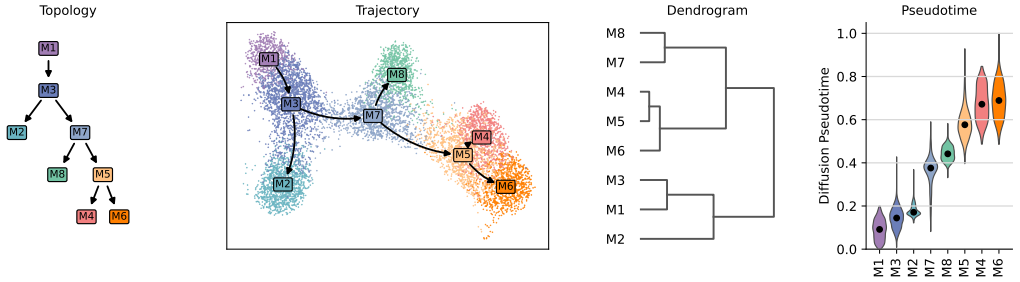

Figure 7: **Trajectory inference on the `binary_tree_8` benchmark dataset by GatorSC. The Topology panel (left) shows the directed tree connecting eight meta-states (M1–M8) and recapitulates the ground-truth branching structure. The Trajectory panel visualizes single cells in the learned low-dimensional manifold, colored by meta-state, with the black curve indicating the principal developmental paths from the upstream state M1 toward downstream branches. The Dendrogram panel depicts hierarchical clustering of meta-states based on their transcriptomic similarity, closely matching the inferred branch structure. The Pseudotime panel displays diffusion pseudotime distributions for each meta-state, revealing a gradual increase from early states (M1, M3, M2) to late states (M5, M4, M6), consistent with the inferred developmental progression.**

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 con-

tinuous 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.

5.7. Biological interpretability and case studies

5.7.1. 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] (Figure 8). In the dot plot (Figure 8A), 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 (Figure 8B) depict the distribution of expression for representative genes across the five **GatorSC**-defined cell types, and a heatmap of normalized marker scores (Figure 8C) 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 two-dimensional embedding of the **GatorSC** representation (Figure 8D), with cells colored by inferred label, reveals well-separated clusters corresponding to the five major cell types. These results demonstrate that **GatorSC** yields coherent cell-type annotations that recapitulate known skeletal muscle and immune cell biology and support marker-based downstream interpretation of single-cell data.

5.7.2. Application to an 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 [57]. In the learned latent space, **GatorSC** robustly segregates cells into five major brain cell populations, microglia, neurons, oligodendrocyte progenitor cells (OPCs), astrocytes, and oligodendrocytes (Figure 9A), 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 (Figure 9B). 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 path-

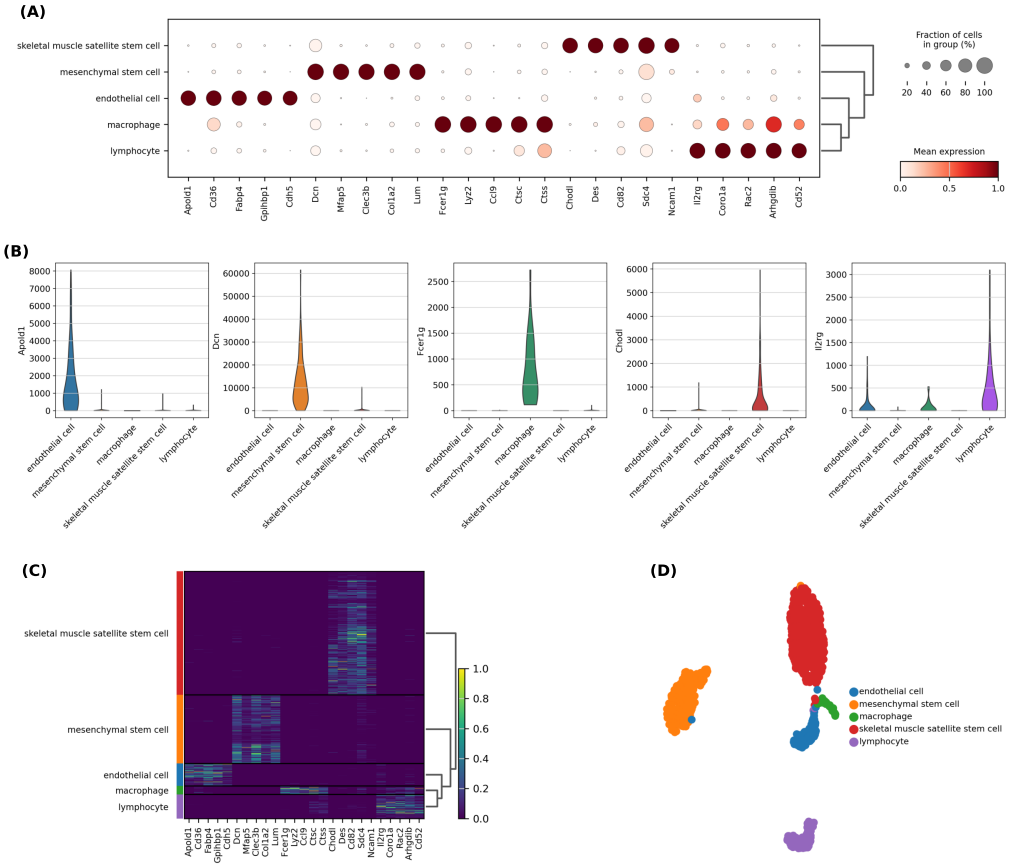

Figure 8: **Marker gene analysis in the Quake Smart-seq2 diaphragm dataset. (a) Dot plot of average expression of selected canonical marker genes across five GatorSC-annotated cell types. (b) Violin plots of representative marker gene expression in each cell type. (c) Heatmap of normalized marker scores summarizing the marker panel. (d) UMAP visualization of the GatorSC latent representation, colored by inferred cell-type labels.**

ways, 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.

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 (Figure 9B). This pattern suggests a relative suppression of MAPK-mediated regulation in myelinating oligodendrocytes compared to 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.

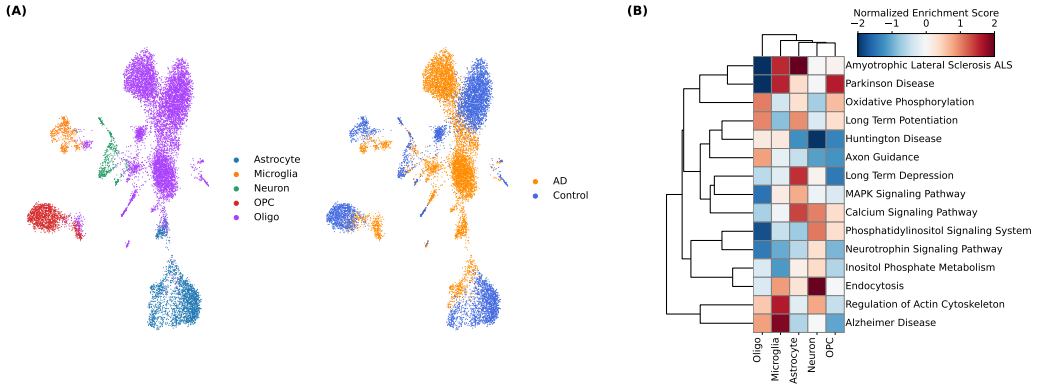

Figure 9: **Application of GatorSC to an Alzheimer's disease single-nucleus RNA-seq dataset. (A) UMAP visualization of single-nucleus profiles, colored by GatorSC-inferred cell clusters (left) and by disease status (AD vs. control; right). (B) Heatmap of normalized enrichment scores for selected AD-related pathways in each major cell type, comparing AD versus control nuclei.**

5.8. Ablation Study

To quantify the contribution of each component in **GatorSC**, we conducted an ablation study comparing the full model with six variants, as shown in Figure 10. In **GatorSC**$_\alpha$, we removed the MoE fusion and instead concatenated $Z^{(1)}$, $Z^{(2)}$, and $Z^{(3)}$, followed by a linear projection back to the original latent space. **GatorSC**$_\beta$ and **GatorSC**$_\gamma$ retained only the contrastive or reconstruction objective, respectively, while discarding the complementary self-supervised loss terms. **GatorSC**$_\delta$ removed the local gene–gene graph and

fused only $Z^{(1)}$ and $Z^{(2)}$, whereas **GatorSC**$_\varepsilon$ removed the global gene–gene graph and fused $Z^{(1)}$ and $Z^{(3)}$. Finally, **GatorSC**$_\zeta$ used only the global cell–cell graph, directly treating $Z^{(1)}$ as the final representation. Across 14 benchmark scRNA-seq datasets, the full **GatorSC** consistently achieves the best ARI, NMI, ACC, and homogeneity scores, whereas all ablated variants exhibit noticeable performance degradation. The performance gap between **GatorSC** and **GatorSC**$_\alpha$ indicates that MoE-based adaptive fusion is more effective than a simple concatenation-plus-linear layer, highlighting the benefit of data-dependent expert weighting. The inferior results of **GatorSC**$_\beta$ and **GatorSC**$_\gamma$ demonstrate that contrastive and reconstruction objectives provide complementary supervisory signals, and that removing either objective harms clustering quality. Moreover, the performance drops observed for **GatorSC**$_\delta$ and **GatorSC**$_\varepsilon$ confirm that both global and local gene–gene graphs contribute non-redundant information beyond the cell–cell graph. Finally, **GatorSC**$_\zeta$, which relies solely on the global cell–cell graph, yields suboptimal performance, highlighting the importance of jointly modeling hierarchical gene–gene structure and employing dual self-supervised learning to obtain robust and discriminative cell representations.

5.9. Hyperparameter sensitivity analysis

We systematically evaluated the sensitivity of **GatorSC** to its key hyperparameters, namely the k-hop radius n_k, the path-wise neighborhood size n_p, the contrastive temperature τ, and the dropout rate p_{drop}. For each hyperparameter, we varied its value over a wide range and assessed the resulting clustering quality using ARI, NMI, ACC, and homogeneity across the 14 benchmark scRNA-seq datasets. As summarized in Figure 11, the performance curves remain largely flat over broad intervals for all four hyperparameters, indicating that **GatorSC** maintains stable clustering accuracy under substantial variation in its configuration. While very extreme settings can lead to a mild decrease in performance on a subset of datasets, we do not observe any abrupt performance collapse or catastrophic degradation. Instead, comparable performance is consistently achieved across a continuum of reasonable choices for n_k, n_p, τ, and p_{drop}, and the qualitative trends are aligned across all evaluation metrics. These results demonstrate that **GatorSC** is robust to hyperparameter variation and does not rely on carefully tuned, dataset-specific configurations, which reduces the burden of hyperparameter selection and facilitates its deployment to new scRNA-seq datasets.

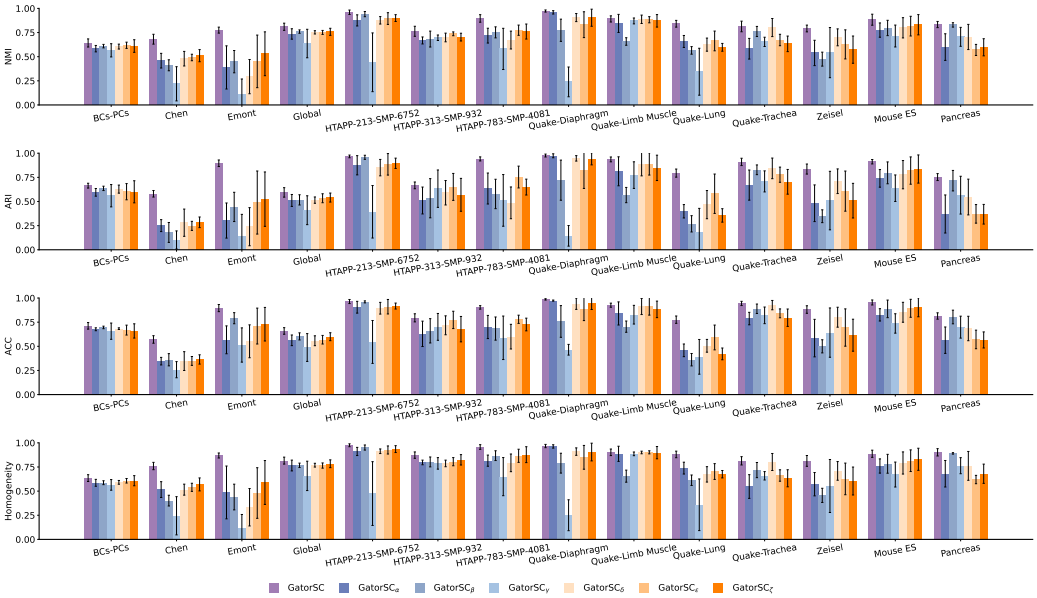

Figure 10: **Clustering performance comparison across 14 scRNA-seq datasets between GatorSC and its six variants. For each dataset, rows show ARI, NMI, ACC, and homogeneity scores; higher values indicate better clustering performance.**

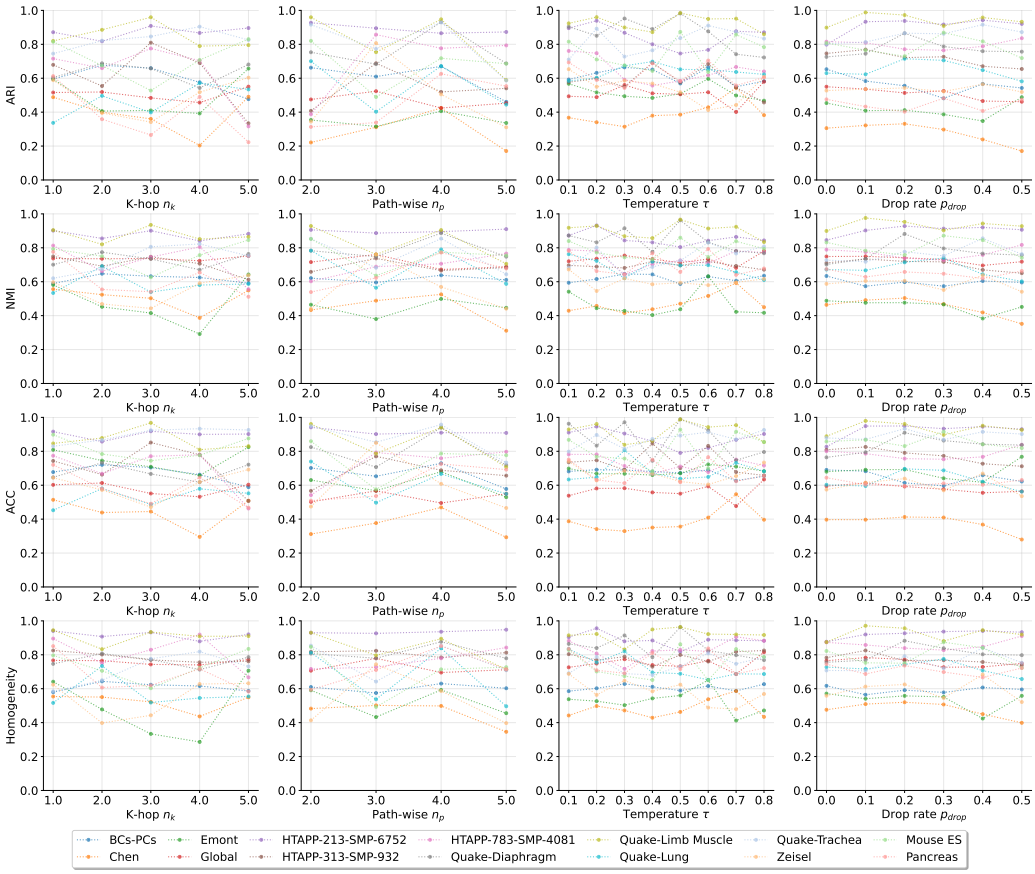

Figure 11: **Hyperparameter analysis results using four clustering evaluation metrics: ARI, NMI, ACC, and Homogeneity on 14 scRNA-seq datasets. Higher values across all metrics indicate better clustering performance.**

6. Conclusion

In this work, we introduced **GatorSC**, a unified representation learning framework that models scRNA-seq data through multi-scale cell and gene graphs and fuses these complementary views with a Mixture-of-Experts architecture. By jointly constructing a global cell–cell graph, a global gene–gene graph, and a local, context-specific gene–gene graph, **GatorSC** explicitly captures population-level cellular organization, cross-cell gene dependencies, and fine-grained gene interactions within local neighborhoods. These hierarchical graphs are coupled with a dual self-supervised objective that combines graph reconstruction and graph contrastive learning, enabling the model to learn robust and biologically meaningful cell embeddings without relying on task-specific supervision.

Comprehensive evaluations across 19 publicly available scRNA-seq datasets demonstrate that **GatorSC** achieves consistently strong, often state-of-the-art, performance on three core tasks: cell clustering, gene expression imputation, and cell-type annotation. On the 14 benchmark datasets used for comparisons, the method maintains high clustering quality on diverse tissues, species, and sequencing platforms, robustly recovers gene expression under increasing levels of simulated dropout, and provides accurate cell-type labels. Analyses on datasets that vary in annotation depth and scale, including the AMB3/16/92 series, Tabula Muris, and Zheng 68K, further show that **GatorSC** scales to large, deeply annotated atlases while preserving fine-grained cellular structure. Additional experiments including t-SNE visualization, trajectory reconstruction on the `binary_tree_8` benchmark, marker gene analysis in the Quake diaphragm dataset, and application to an Alzheimer's disease single-nucleus RNA-seq cohort demonstrate that the learned embeddings preserve known biological structure and support interpretable downstream discoveries. Ablation and hyperparameter sensitivity studies confirm that each component of **GatorSC**, including the hierarchical gene graphs, MoE-based adaptive fusion, and unified self-supervision, contributes to its overall robustness and performance.

GatorSC provides a flexible foundation that can be extended in several directions. Methodologically, incorporating additional modalities such as chromatin accessibility, spatial coordinates, or protein abundance would enable multi-omic and spatially aware variants of the framework. The MoE paradigm could also be utilized for designing task-aware experts or to transfer knowledge across related datasets and perturbation conditions. Additionally,

integrating **GatorSC** with large-scale cell atlases and longitudinal cohorts may facilitate systematic mapping of rare cell states, disease progression, and treatment response. Overall, by combining hierarchical graph modeling with adaptive fusion and unified self-supervision, **GatorSC** offers a powerful tool for comprehensive single-cell transcriptomic analysis.

Code availability

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

Compliance with Ethics Requirements

This article does not contain any studies with human or animal subjects.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

The project is partially supported by NIH grants R01AG083039, RF1AG084178, RF1AG077820, R01AG080991, R01AG080624, and R01AG076234.

References

[1] N. Schaum, J. Karkanias, N. F. Neff, A. P. May, S. R. Quake, T. Wyss-Coray, S. Darmanis, J. Batson, O. Botvinnik, M. B. Chen, et al., Single-cell transcriptomics of 20 mouse organs creates a tabula muris: The tabula muris consortium, Nature 562 (7727) (2018) 367.

[2] A. Alemany, M. Florescu, C. S. Baron, J. Peterson-Maduro, A. Van Oudenaarden, Whole-organism clone tracing using single-cell sequencing, Nature 556 (7699) (2018) 108–112.

[3] W. Chung, H. H. Eum, H.-O. Lee, K.-M. Lee, H.-B. Lee, K.-T. Kim, H. S. Ryu, S. Kim, J. E. Lee, Y. H. Park, et al., Single-cell rna-seq enables comprehensive tumour and immune cell profiling in primary breast cancer, Nature communications 8 (1) (2017) 15081.

[4] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, N. Yosef, Deep generative modeling for single-cell transcriptomics, Nature methods 15 (12) (2018) 1053–1058.

[5] T. Tian, J. Wan, Q. Song, Z. Wei, Clustering single-cell rna-seq data with a model-based deep learning approach, Nature Machine Intelligence 1 (4) (2019) 191–198.

[6] X. Li, K. Wang, Y. Lyu, H. Pan, J. Zhang, D. Stambolian, K. Susztak, M. P. Reilly, G. Hu, M. Li, Deep learning enables accurate clustering with batch effect removal in single-cell rna-seq analysis, Nature communications 11 (1) (2020) 2338.

[7] L. Qin, G. Zhang, S. Zhang, Y. Chen, Deep batch integration and denoise of single-cell rna-seq data, Advanced Science 11 (29) (2024) 2308934.

[8] J. Wang, A. Ma, Y. Chang, J. Gong, Y. Jiang, R. Qi, C. Wang, H. Fu, Q. Ma, D. Xu, scgnn is a novel graph neural network framework for single-cell rna-seq analyses, Nature communications 12 (1) (2021) 1882.

[9] M. Ciortan, M. Defrance, Gnn-based embedding for clustering scrna-seq data, Bioinformatics 38 (4) (2022) 1037–1044.

[10] Y. Gan, X. Huang, G. Zou, S. Zhou, J. Guan, Deep structural clustering for single-cell rna-seq data jointly through autoencoder and graph neural network, Briefings in Bioinformatics 23 (2) (2022).

[11] J. Lee, S. Kim, D. Hyun, N. Lee, Y. Kim, C. Park, Deep single-cell rna-seq data clustering with graph prototypical contrastive learning, Bioinformatics 39 (6) (2023) btad342.

[12] Z. Zhang, Y. Liu, M. Xiao, K. Wang, Y. Huang, J. Bian, R. Yang, F. Li, Graph contrastive learning as a versatile foundation for advanced scrna-seq data analysis, Briefings in Bioinformatics 25 (6) (2024) bbae558.

[13] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, G. E. Hinton, Adaptive mixtures of local experts, Neural computation 3 (1) (1991) 79–87.

[14] M. I. Jordan, R. A. Jacobs, Hierarchical mixtures of experts and the em algorithm, Neural computation (1994).

[15] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, J. Dean, Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, arXiv preprint arXiv:1701.06538 (2017).

[16] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, et al., Mixtral of experts, arXiv preprint arXiv:2401.04088 (2024).

[17] R. Wang, S. An, M. Cheng, T. Zhou, S. J. Hwang, C.-J. Hsieh, One prompt is not enough: Automated construction of a mixture-of-expert prompts, arXiv preprint arXiv:2407.00256 (2024).

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, NeurIPS 30 (2017).

[19] Y. N. Dauphin, A. Fan, M. Auli, D. Grangier, Language modeling with gated convolutional networks, in: International conference on machine learning, 2017.

[20] S. Wang, Z. Chen, C. Shi, C. Shen, J. Li, Mixture of demonstrations for in-context learning, in: NeurIPS, 2024.

[21] H. Yu, Z. Qi, L. K. Jang, R. Salakhutdinov, L.-P. Morency, P. P. Liang, Mmoe: Enhancing multimodal models with mixtures of multimodal interaction experts, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 10006–10030.

[22] X. Han, H. Nguyen, C. Harris, N. Ho, S. Saria, Fusemoe: Mixture-of-experts transformers for fleximodal fusion, Advances in Neural Information Processing Systems 37 (2024) 67850–67900.

[23] K. Lazaros, D. E. Koumadorakis, P. Vlamos, A. G. Vrahatis, Graph neural network approaches for single-cell data: a recent overview, Neural Computing and Applications 36 (17) (2024) 9963–9987.

[24] N. Adossa, S. Khan, K. T. Rytkönen, L. L. Elo, Computational strategies for single-cell multi-omics integration, Computational and Structural Biotechnology Journal 19 (2021) 2588–2596.

[25] S. Wang, Y. Liu, Z. Zhang, Q. Ma, Q. Song, J. Bian, Gatorst: A versatile contrastive meta-learning framework for spatial transcriptomic data analysis, bioRxiv (2025).

[26] T. Kipf, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).

[27] W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, J. Huang, A survey on mixture of experts in large language models, IEEE Transactions on Knowledge and Data Engineering (2025).

[28] H. Wang, Z. Jiang, Y. You, Y. Han, G. Liu, J. Srinivasa, R. Kompella, Z. Wang, et al., Graph mixture of experts: Learning on large-scale graphs with explicit diversity modeling, Advances in Neural Information Processing Systems 36 (2023) 50825–50837.

[29] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, J. Tang, Self-supervised learning: Generative or contrastive, IEEE transactions on knowledge and data engineering 35 (1) (2021) 857–876.

[30] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, J. Tang, Graphmae: Self-supervised masked graph autoencoders, in: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, 2022, pp. 594–604.

[31] J. Li, R. Wu, W. Sun, L. Chen, S. Tian, L. Zhu, C. Meng, Z. Zheng, W. Wang, What's behind the mask: Understanding masked graph modeling for graph autoencoders, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 1268–1279.

[32] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, Y. Shen, Graph contrastive learning with augmentations, Advances in neural information processing systems 33 (2020) 5812–5823.

[33] Z. Zhang, Y. Liu, J. Bian, A. J. Yepes, J. Shen, F. Li, G. Long, F. D. Salim, Boosting patient representation learning via graph contrastive learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2024, pp. 335–350.

[34] Y. Gu, R. Bartolomé-Casado, C. Xu, A. Bertocchi, A. Janney, C. Heuberger, C. F. Pearson, S. A. Teichmann, E. E. Thornton, F. Powrie, Immune microniches shape intestinal treg function, Nature 628 (8009) (2024) 854–862.

[35] C. Chen, W. Yu, F. Alikarami, Q. Qiu, C.-h. Chen, J. Flournoy, P. Gao, Y. Uzun, L. Fang, J. W. Davenport, et al., Single-cell multiomics reveals increased plasticity, resistant populations, and stem-cell–like blasts in kmt2a-rearranged leukemia, Blood, The Journal of the American Society of Hematology 139 (14) (2022) 2198–2211.

[36] M. P. Emont, C. Jacobs, A. L. Essene, D. Pant, D. Tenen, G. Colleluori, A. Di Vincenzo, A. M. Jørgensen, H. Dashti, A. Stefek, et al., A single-cell atlas of human and mouse white adipose tissue, Nature 603 (7903) (2022) 926–933.

[37] J. Klughammer, D. L. Abravanel, Å. Segerstolpe, T. R. Blosser, Y. Goltsev, Y. Cui, D. R. Goodwin, A. Sinha, O. Ashenberg, M. Slyper, et al., A multi-modal single-cell and spatial expression map of metastatic breast cancer biopsies across clinicopathological features, Nature Medicine 30 (11) (2024) 3236–3249.

[38] A. Zeisel, A. B. Muñoz-Manchado, S. Codeluppi, P. Lönnerberg, G. La Manno, A. Juréus, S. Marques, H. Munguba, L. He, C. Betsholtz, et al., Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq, Science 347 (6226) (2015) 1138–1142.

[39] A. M. Klein, L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D. A. Weitz, M. W. Kirschner, Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells, Cell 161 (5) (2015) 1187–1201.

[40] M. D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Müller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché,

et al., Benchmarking atlas-level data integration in single-cell genomics, Nature methods 19 (1) (2022) 41–50.

[41] B. Tasic, Z. Yao, L. T. Graybuck, K. A. Smith, T. N. Nguyen, D. Bertagnolli, J. Goldy, E. Garren, M. N. Economo, S. Viswanathan, et al., Shared and distinct transcriptomic cell types across neocortical areas, Nature 563 (7729) (2018) 72–78.

[42] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, et al., Massively parallel digital transcriptional profiling of single cells, Nature communications 8 (1) (2017) 14049.

[43] V. A. Traag, L. Waltman, N. J. Van Eck, From louvain to leiden: guaranteeing well-connected communities, Scientific reports 9 (1) (2019) 1–12.

[44] J. H. Levine, E. F. Simonds, S. C. Bendall, K. L. Davis, E.-a. D. Amir, M. D. Tadmor, O. Litvin, H. G. Fienberg, A. Jager, E. R. Zunder, et al., Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis, Cell 162 (1) (2015) 184–197.

[45] X. Wu, Y. Zhou, Ge-impute: graph embedding-based imputation for single-cell rna-seq data, Briefings in Bioinformatics 23 (5) (2022) bbac313.

[46] D. Van Dijk, R. Sharma, J. Nainys, K. Yim, P. Kathail, A. J. Carr, C. Burdziak, K. R. Moon, C. L. Chaffer, D. Pattabiraman, et al., Recovering gene interactions from single-cell data using data diffusion, Cell 174 (3) (2018) 716–729.

[47] J. Lee, S. Yun, Y. Kim, T. Chen, M. Kellis, C. Park, Single-cell rna sequencing data imputation using bi-level feature propagation, Briefings in Bioinformatics 25 (3) (2024) bbae209.

[48] H. Li, C. R. Brouwer, W. Luo, A universal deep neural network for in-depth cleaning of single-cell rna-seq data, Nature Communications 13 (1) (2022) 1901.

[49] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, F. J. Theis, Single-cell rna-seq denoising using a deep count autoencoder, Nature communications 10 (1) (2019) 390.

[50] J. Liu, X. Fan, C. Gu, Y. Yang, B. Wu, G. Chen, C.-Y. Hsieh, P.-A. Heng, scheteronet: A heterophily-aware graph neural network for accurate cell type annotation and novel cell detection, Advanced Science 12 (16) (2025) 2412095.

[51] X. Shao, H. Yang, X. Zhuang, J. Liao, P. Yang, J. Cheng, X. Lu, H. Chen, X. Fan, scdeepsort: a pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network, Nucleic acids research 49 (21) (2021) e122–e122.

[52] F. Ma, M. Pellegrini, Actinn: automated identification of cell types in single cell rna sequencing, Bioinformatics 36 (2) (2020) 533–538.

[53] C. Domínguez Conde, C. Xu, L. B. Jarvis, D. B. Rainbow, S. B. Wells, T. Gomes, S. Howlett, O. Suchanek, K. Polanski, H. King, et al., Cross-tissue immune cell analysis reveals tissue-specific features in humans, Science 376 (6594) (2022) eabl5197.

[54] C. Xu, R. Lopez, E. Mehlman, J. Regier, M. I. Jordan, N. Yosef, Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models, Molecular systems biology 17 (1) (2021) e9620.

[55] F. A. Wolf, P. Angerer, F. J. Theis, Scanpy: large-scale single-cell gene expression data analysis, Genome biology 19 (1) (2018) 15.

[56] W. Saelens, R. Cannoodt, H. Todorov, Y. Saeys, A comparison of single-cell trajectory inference methods, Nature biotechnology 37 (5) (2019) 547–554.

[57] A. Grubman, G. Chew, J. F. Ouyang, G. Sun, X. Y. Choo, C. McLean, R. K. Simmons, S. Buckberry, D. B. Vargas-Landin, D. Poppe, et al., A single-cell atlas of entorhinal cortex from individuals with alzheimer's disease reveals cell-type-specific gene expression regulation, Nature neuroscience 22 (12) (2019) 2087–2097.