Deep single-cell RNA-seq data clustering with graph prototypical contrastive learning

Anonymous Authors[1]

Abstract

In single-cell RNA sequencing analysis, despite the importance of identifying cell types through clustering techniques for downstream analysis, challenges of scRNA-seq data, such as pervasive dropout phenomena, hinder obtaining robust clustering outputs. Although existing studies try to alleviate these problems, they mainly rely on reconstruction-based losses that highly depend on the data quality, which is sometimes noisy. This work proposes a graph-based prototypical contrastive learning method, named scGPCL. Specifically, scGPCL encodes the cell representations using Graph Neural Networks on cell-gene graph that captures the relational information inherent in scRNA-seq data and introduces prototypical contrastive learning to learn cell representations by pushing apart semantically dissimilar pairs and pulling together similar ones. Through extensive experiments on both simulated and real scRNA-seq data, we demonstrate the effectiveness and efficiency of scGPCL. Code is available at https://github.com/Junseok0207/scGPCL

1. Introduction

By measuring transcriptome-wide gene expression at single cell level, single-cell RNA sequencing (scRNA-seq) studies have helped researchers to better understand complex biological questions, such as exploring cellular heterogeneity. To this end, clustering techniques that identify cell types of cells have been widely studied. Early studies mainly relied on dimensionality reduction techniques, such as PCA, t-SNE (Maaten, 2008), and UMAP (McInnes et al., 2018), however, they fall short of handling scRNA-seq data that typically contains tens of thousands dimensional features, which incurs the curse of dimensionality leading to poor clustering performance of traditional clustering algorithms. Moreover, a considerable fraction of truly expressed genes is not well observed in scRNA-seq data owing to the pervasive dropout phenomenon, which results in false zero counts incurring further difficulties in analyzing scRNA-seq data.

Recently, Deep Neural Networks (DNN) have emerged as powerful feature extractors for dimensionality reduction or clustering, and several recent methods have translated this success to scRNA-seq data. Most of them, such as DCA (Eraslan et al., 2019), and scDeepCluster (Tian et al., 2019), leverage an autoencoder network that learns cell representations with compression and reconstruction schemes. However, these methods are hard to learn cell representations for accurate clustering when input features are not informative enough (e.g., input gene expression matrix is highly sparse) because they do not leverage any relational information between cells.

Although reconstruction-based representation learning is a dominant way of learning cell representations in scRNA-seq data, contrastive learning-based representation learning methods have also been investigated for the scRNA-seq data. We argue that the contrastive learning framework is especially well-suited for analyzing highly sparse scRNA-seq data, since contrastive learning generates cell representations that are more tolerant to noise than the reconstruction-based representation learning framework. This is because the contrastive learning framework aims to learn cell representations by comparing the similarity between a positive pair and negative pairs in the representation space, whereas the reconstruction-based approaches solely rely on reconstructing the input matrix that is highly sparse in nature due to the inevitable dropout phenomenon. Recently proposed contrastive-sc (Ciortan & Defrance, 2021) adopts instance-wise contrastive learning where a positive pair is defined by randomly masking some gene expression values of a cell, while all other cells are considered as negative pairs. However, we argue that such a simple augmentation scheme is less sufficient to fully leverage the benefit of contrastive learning as it fails to incorporate any relational information between cells.

To leverage the relational information between cells, several existing studies construct a cell-cell graph on which Graph

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by *the 2023 ICML Workshop on Computational Biology*. Do not distribute.

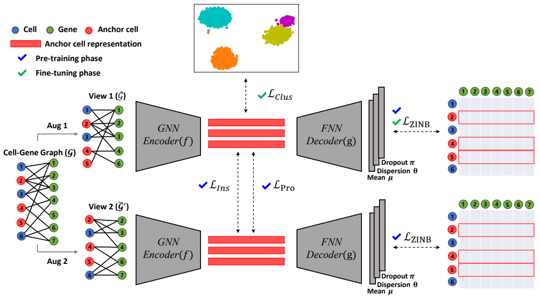

Figure 1. The overall architecture of scGPCL.

Neural Networks (GNNs) are applied. Although the existing studies have shown the effectiveness of reflecting the relational information between cells for the cell clustering task, we argue that since they construct a cell-cell graph based on the learned cell representations or raw expression value, the inherent sparseness of the input gene expression matrix used to compute the similarity between cells leads to a low-quality cell-cell graph, which eventually hinders the construction of high-quality cell representations[1].

In this paper, we propose a graph-based prototypical contrastive learning method aiming at clustering cells in the scRNA-seq data that fully leverages the relational information between cells. We introduce a *bipartite cell-gene graph*, which is constructed by connecting two nodes (i.e., cell and gene) if a particular gene is expressed in that cell on the given input gene expression matrix. This preserves the inherent relationship in the given data, which eventually maintains the quality of the constructed graph. Then, we conduct instance-wise contrastive learning with augmentation techniques that mimics the nature of scRNA-seq data to better capture the characteristics of scRNA-seq data. Moreover, we adopt the prototypical contrastive learning scheme to help our model learn cluster (i.e., cell type) specific information and alleviate the sampling bias (Chuang et al., 2020) by pulling together an anchor cell and its corresponding cluster prototype. Hence, we name our proposed method as single-cell Graph Prototypical Contrastive Learning (scGPCL). Through extensive experiments on both simulated and real scRNA-seq data, we demonstrate the robustness and efficacy of scGPCL compared with state-of-the-art methods.

2. Methods

scGPCL is a graph-based prototypical contrastive learning method designed for clustering cells in the scRNA-seq data, and its overall architecture is shown in Figure 1. First of all, we define a bipartite cell-gene graph denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents a set of nodes consisting of cell and gene nodes and $\mathcal{E}$ represents set of edges, where a cell node and a gene node are connected with an associated expression value (i.e., edge weight) if the gene is expressed in that cell. Note that scGPCL leverages the *cell-gene bipartite graph* obtained from the original gene expression matrix to preserve the natural relationship between cells inherent in the given data rather than leveraging a *cell-cell graph* constructed based on the pre-calculated cell-cell similarity, as it may incur information loss if the similarity values are inaccurate leading to a noisy cell-cell graph.

2.1. Phase 1: Pre-training

The learning strategy of scGPCL is divided into the pre-training and fine-tuning phases. In the pre-training phase, we generate two augmented views $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$ and $\tilde{\mathcal{G}}' = (\tilde{\mathcal{V}}', \tilde{\mathcal{E}}')$ from the original cell-gene graph by applying two different stochastic augmentation functions composed of subgraph sampling and feature masking. This augmentation process is designed to mimic the technical limitations in sequencing, where only a fraction of the gene expression is detected for each cell. Then, scGPCL obtains anchor cell representations $\tilde{H} = f(\tilde{\mathcal{G}}) \in \mathbb{R}^{N_b \times d}$ and $\tilde{H}^{'} = f(\tilde{\mathcal{G}}') \in \mathbb{R}^{N_b \times d}$ from two differently augmented graphs by passing them through a GNN encoder f, where d is the dimensionality of the cell representation and N_b is a number of anchor nodes in the current batch.

Instance-wise Contrastive Loss. After the encoding process described above, we apply the contrastive learning framework whose overview can be found in Figure 6 in the appendix. More precisely, scGPCL computes the infoNCE (Oord et al., 2018) objective for each positive cell node pair $(\tilde{h}_i, \tilde{h}_i^{'})$, where $\tilde{h}_i$ and $\tilde{h}_i^{'}$ are the i-th row of the $\tilde{H}$ and $\tilde{H}^{'}$, respectively, which denote the representation of cell i from the two views:

$$l_{\text{Ins}}(\tilde{h}_i, \tilde{h}_i^{'}) = \log \frac{e^{(\mathsf{sim}(\tilde{h}_i, \tilde{h}_i^{'})/\tau)}}{\sum_{j=1}^{N_b} \mathbb{1}_{[i \neq j]} e^{(\mathsf{sim}(\tilde{h}_i, \tilde{h}_j)/\tau)} + \sum_{j=1}^{N_b} e^{(\mathsf{sim}(\tilde{h}_i, \tilde{h}_j^{'})/\tau)}} \tag{1}$$

where $\mathsf{sim}(\cdot, \cdot)$ is the cosine similarity between two vectors, $\mathbb{1}_{[\cdot]}$ is the indicator function, and τ is the temperature hyperparamter. The overall instance-wise contrastive loss is calculated by

$$\mathcal{L}_{\text{Ins}} = -\frac{1}{2N_b} \sum_{i=1}^{N_b} [l_{\text{Ins}}(\tilde{h}_i, \tilde{h}_i^{'}) + l_{\text{Ins}}(\tilde{h}_i^{'}, \tilde{h}_i)]. \tag{2}$$

By minimizing the above contrastive loss, scGPCL learns the cell representations by pulling together positive pairs and pushing apart negative pairs in the cell representation space. Note that such a contrastive learning scheme is especially beneficial for scRNA-seq data, because it is hard to learn cell representations with only reconstruction-based loss when the given input matrix is highly sparse due to the pervasive dropout phenomenon.

[1] The experiments about this argument can be found in the appendix A

Prototypical Contrastive Learning Framework. However, the instance-wise contrastive loss exhibits an inherent limitation, called sampling bias. In other words, given an anchor cell, since all other cells apart from the augmented version of the anchor cell are considered as negative instances, it is highly likely that negative instances contain cells that belong to the same cell type as the anchor cell, which is undesirably pushed apart from the anchor cell. To alleviate this problem, scGPCL adopts the prototypical contrastive learning framework that treats the pairs of cells assigned to the same prototype as positive pairs and the remaining pairs as negative pairs. The loss for a particular cell i is given as follows:

$$l_{\text{Pro}}(\tilde{h}_i) = \frac{1}{T}\sum_{t=1}^{T}\sum_{s=1}^{K_t} \mathbb{1}_{(\tilde{h}_i \in z_s^t)} \log \frac{e^{(\mathsf{sim}(\tilde{h}_i, z_s^t)/\tau)}}{\sum_{j=1}^{K_t} e^{(\mathsf{sim}(\tilde{h}_i, z_j^t)/\tau)}} \tag{3}$$

where T is the number of clustering rounds to provide prototypes in various granularities, K_t denotes the number of prototypes in t-th iteration, and $z_s^t \in \mathbb{R}^d$ denotes the representation of the prototype s in the t-th iteration. The indicator function $\mathbb{1}_{(\tilde{h}_i \in z_s^t)}$ is defined as 1 if the cell i belongs to the cluster represented by z_s^t, and 0 otherwise. Note that the prototypical loss defined as above is especially beneficial for clustering task, because it groups cells that belong to the same cell type together by minimizing the distance between each cell and the corresponding cluster prototype. The overall prototypical contrastive loss is given as follows:

$$\mathcal{L}_{\text{Pro}} = -\frac{1}{N_b}\sum_{i=1}^{N_b} l_{\text{Pro}}(\tilde{h}_i). \tag{4}$$

ZINB-based Reconstruction Loss. Following existing studies (Tian et al., 2019; Gan et al., 2022), we assume that the gene expression matrix follows a zero-inflated negative binomial (ZINB) distribution and estimate the parameters of the ZINB distribution, namely, the mean (μ), dispersion (θ), and dropout probability (π) by passing through the output of the decoder to the additional layer for each of the three parameters. The overall ZINB-based reconstruction loss of scGPCL from the two views is given by:

$$\mathcal{L}_{\text{ZINB}}^{\text{Pre}} = \frac{1}{2}[l_{\text{ZINB}}(\tilde{\Pi}, \tilde{M}, \tilde{\Theta}) + l_{\text{ZINB}}(\tilde{\Pi}^{'}, \tilde{M}^{'}, \tilde{\Theta}^{'})] \tag{5}$$

where l_{ZINB} is negative log-likelihood of ZINB distribution and $\tilde{\Pi}$, $\tilde{M}$, and $\tilde{\Theta}$ represent the estimated parameters from the view 1, and $\tilde{\Pi}^{'}$, $\tilde{M}^{'}$, $\tilde{\Theta}^{'}$ denote those from the view 2. More details about reconstruction loss can be found in Appendix C.

Final Objectives of Pre-training Phase. Finally, scGPCL combines $\mathcal{L}_{\text{Ins}}$, $\mathcal{L}_{\text{Pro}}$, and $\mathcal{L}_{\text{ZINB}}^{\text{Fine}}$ with balance coefficients λ_1 and λ_2 to learn cell representations in the pre-training phase as follows:

$$\mathcal{L}_{\text{Pre}} = \lambda_1\mathcal{L}_{\text{Ins}} + \lambda_2\mathcal{L}_{\text{Pro}} + \mathcal{L}_{\text{ZINB}}^{\text{Pre}} \tag{6}$$

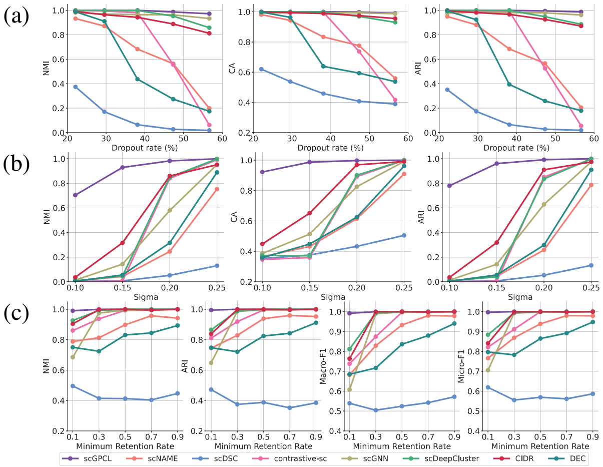

Figure 2. Performance comparisons of scGPCL and other baselines on the simulated dataset. (a), (b), and (c) represent the performance over the various dropout rates, sigmas (small sigma indicates low signals for clustering), and minimum retention rates (small value indicates more imbalanced data), respectively.

2.2. Phase 2: Fine-tuning

Clustering Task-Oriented Loss. In the fine-tuning phase, scGPCL adopts a self-training scheme to encourage each cell to be assigned to a cluster of high confidence. Specifically, it minimizes the Kullback-Leibler (KL) divergence between the soft cluster assignment distribution Q, calculated based on the similarity between the cell representations and the cluster centroids, and the target distribution P, which is obtained by sharpening Q. Formally, this objective is denoted as $\mathcal{L}_{\text{Cluster}} = D_{\text{KL}}(P\|Q)$. More details about clustering loss can be found in Appendix D.

Final Objectives of Fine-tuning Phase. Finally, the overall loss of scGPCL in the fine-tuning phase is defined by combining $\mathcal{L}_{\text{Cluster}}$ and $\mathcal{L}_{\text{ZINB}}^{\text{Fine}}$ with a balance coefficient λ_3 as follows:

$$\mathcal{L}_{\text{Fine}} = \mathcal{L}_{\text{Cluster}} + \lambda_3\mathcal{L}_{\text{ZINB}}^{\text{Fine}} \tag{7}$$

Note that scGPCL maintains the reconstruction-based loss $\mathcal{L}_{\text{ZINB}}^{\text{Fine}} = l_{\text{ZINB}}(\tilde{\Pi}, \tilde{M}, \tilde{\Theta})$ during the fine-tuning phase to preserve the local structure of data.

3. Results

3.1. Evaluation of scGPCL on simulated data

To demonstrate the effectiveness of scGPCL, we simulate scRNA-seq data with Splatter (Zappia et al., 2017) package assuming three situations in which learning cell representations may be challenging: **Case 1:** Gene expression matrix is highly sparse due to the dropout phenomena (Figure 2a), **Case 2:** Gene expression values contain relatively low signal strength required for clustering (Figure 2b), and **Case 3:** The size of cell clusters is imbalanced in number (Fig-

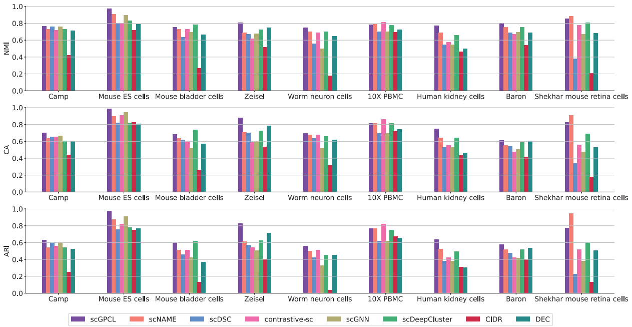

Figure 3. Performance comparisons of scGPCL and other baselines on the nine real scRNA-seq datasets.

ure 2c). To evaluate the clustering performance, we compare scGPCL with seven state-of-the art baselines using three standard clustering evaluation metrics, i.e., normalized mutual information (NMI), clustering accuracy (CA), and adjusted rand index (ARI) and we replace CA with Macro-F1 and Micro-F1 score both of which are well suited for the imbalance cases in Case 3. Through all these results, we demonstrate that scGPCL can robustly separate the cluster of the cells in challenging scenarios, even when the gene expression matrix exhibits severe dropout phenomena, low signal strength, and highly imbalanced cell types.

3.2. Evaluation of scGPCL on real scRNA-seq datasets

To verify the effectiveness of scGPCL on real-world applications, we conduct experiments on real scRNA-seq datasets over various sequencing platforms [2].

Figure 3 shows the overall clustering performance on all nine real-world datasets. Through these experiments, we have the following observations: 1) scGPCL consistently outperforms the state-of-the-art baselines on six datasets, and achieves competitive scores on the three remaining datasets. 2) It is worth noting that scGPCL outperforms contrastive-sc that only leverages instance-wise contrastive learning with a naive augmentation strategy that simply masks some gene expression values. We argue that more advanced augmentation strategy is required to fully leverage the benefit of contrastive learning, and that using the relational information between cells is beneficial. 3) scDSC (Gan et al., 2022) that enhances scDeepCluster using GNNs with a cell-cell graph performs worse than scDeepCluster, implying that simply infusing relational information through a cell-cell graph cannot generally achieve positive effects in many cases. However, scGPCL generally outperforms scDSC by introducing a bipartite cell-gene graph by reflecting the inherent relational information between cells.

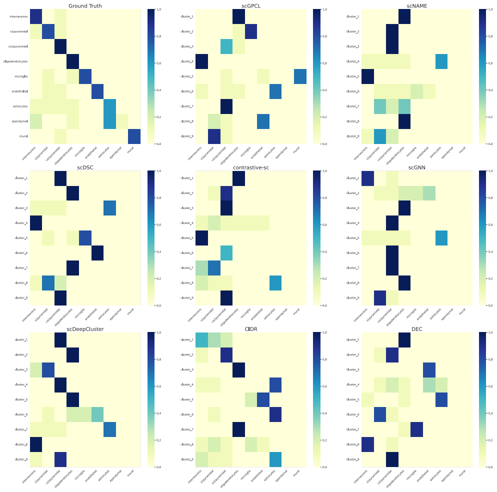

Figure 4. Overlap between gold standard cell types and the top 10 DEGs in clusters detected by ground truth cell type on scGPCL, and baseline methods.

3.3. Marker gene identification

To provide the biological interpretability of the clustering results obtained from scGPCL, we report the marker gene identification result on the Zeisel dataset (Zeisel et al., 2015). Specifically, we find the top 10 differentially expressed genes (DEGs) using the Wilcoxon rank sum test based on the clusters detected by each method and compute the overlap with the gold standard cell types. In Figure 4, DEGs computed based on clustering results obtained from scGPCL highly focus on one cell type except for the 'ependymal' cell type that is hard to detect with the ground truth cell type (upper left). Note that scGPCL succeeds in learning clusters with 'mural' cell type that belongs to a minority class with a small number of cells, whereas other baseline methods fail to do so. Through this result, we can demonstrate that scGPCL can learn cells that belong to the minority cell type from a biological perspective.

4. Conclusion

In this paper, we propose a graph-based prototypical contrastive learning method aiming at clustering cells in the scRNA-seq data that fully leverages the relational information between cells. Instead of relying on the feature information of each cell, scGPCL learns the cell representations using GNNs applied on a bipartite cell-gene graph to reflect the natural relationship between cells inherent in the scRNA-seq data. Moreover, scGPCL adopts instance-wise contrastive learning scheme to fully leverage the relational information as well as prototypical contrastive loss to alleviate the limitation of instance-wise contrastive loss. Through extensive experiments on both simulated and real scRNA-seq datasets, we demonstrate the effectiveness and robustness of scGPCL under real-world challenging scenarios.

[2] Detailed descriptions regarding the datasets are summarized in Appendix F.

References

Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., and Jegelka, S. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

Ciortan, M. and Defrance, M. Contrastive self-supervised clustering of scrna-seq data. *BMC bioinformatics*, 22(1): 1–27, 2021.

Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S., and Theis, F. J. Single-cell rna-seq denoising using a deep count autoencoder. *Nature communications*, 10(1):1–14, 2019.

Gan, Y., Huang, X., Zou, G., Zhou, S., and Guan, J. Deep structural clustering for single-cell rna-seq data jointly through autoencoder and graph neural network. *Briefings in Bioinformatics*, 23(2):bbac018, 2022.

Lin, P., Troup, M., and Ho, J. W. Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome biology*, 18(1):1–11, 2017.

Maaten, L. v. d. Visualizing datausing t-sne. *Journal of machine learning research*, 9:2579–2605, 2008.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

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

Wan, H., Chen, L., and Deng, M. scname: neighborhood contrastive clustering with ancillary mask estimation for scrna-seq data. *Bioinformatics*, 38(6):1575–1583, 2022.

Wang, J., Ma, A., Chang, Y., Gong, J., Jiang, Y., Qi, R., Wang, C., Fu, H., Ma, Q., and Xu, D. scgnn is a novel graph neural network framework for single-cell rna-seq analyses. *Nature communications*, 12(1):1–11, 2021.

Xie, J., Girshick, R., and Farhadi, A. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487. PMLR, 2016.

Zappia, L., Phipson, B., and Oshlack, A. Splatter: simulation of single-cell rna sequencing data. *Genome biology*, 18(1):1–15, 2017.

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

A. Performance comparison of different encoder structures

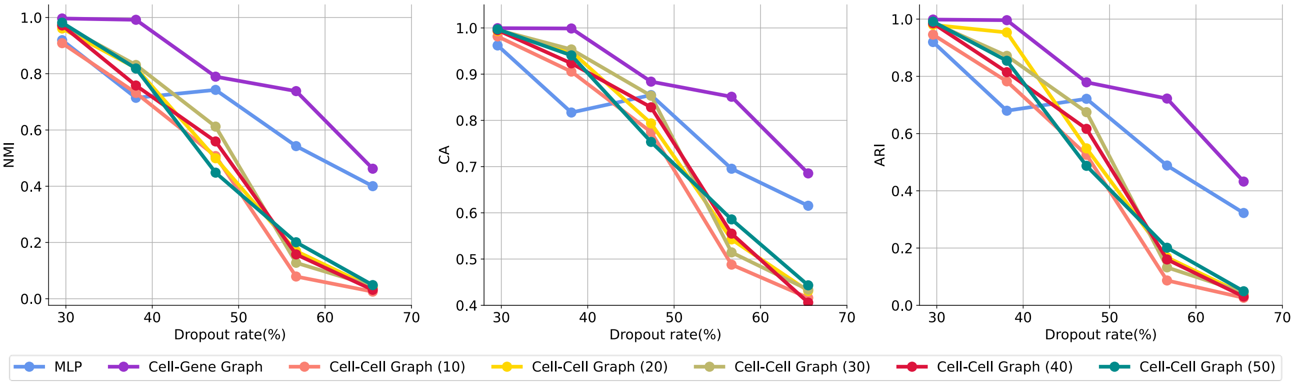

Figure 5. Performance comparison of different encoder structures on the simulated dataset over various dropout rates. We construct the autoencoder style models which have the same 2-layer MLP decoders which output the three parameters of ZINB distribution (i.e., mean, dispersion, and dropout probability) and three different encoders that are MLP, GNNs on a cell-cell graph, and GNNs on a cell-gene graph, respectively. More precisely, pearson correlation is used to calculate the similarity between cells to construct cell-cell graphs following scGNN (Wang et al., 2021) and scDSC (Gan et al., 2022), and we conduct experiments by varying the number of nearest neighbors (i.e., Cell-Cell Graph (10) represent the 10 nearest neighbors graph based on pearson correlation). The cell-gene graph is constructed by connecting two nodes, if a particular gene is expressed in that cell as proposed on scGPCL.

To validate the impact of the input data types for the cell clustering task, in Figure 5, we compare the clustering performance of encoders applied on three different input data types (i.e., Multi-layer Perceptron (MLP) on cell features, GNNs on a cell-cell graph, and GNNs on a cell-gene graph). We observe that GNNs on both the cell-cell graph and the cell-gene graph show good performance over relatively low dropout rates (i.e., $< 40\%$) thanks to the reflection of the relational information between cells, while the performance of GNNs on the cell-cell graph significantly degrades when the dropout rate is higher than 40%. We argue that this is because the highly sparse gene expression matrix makes it hard to accurately compute the similarities between cells, which eventually leads to the drop in the quality of the cell-cell graph. In consequence, this incurs a negative effect on the neighborhood aggregation scheme of GNNs, which eventually results in a poor performance of GNNs on the cell-cell graph.

B. Prototypical Contrastive Learning

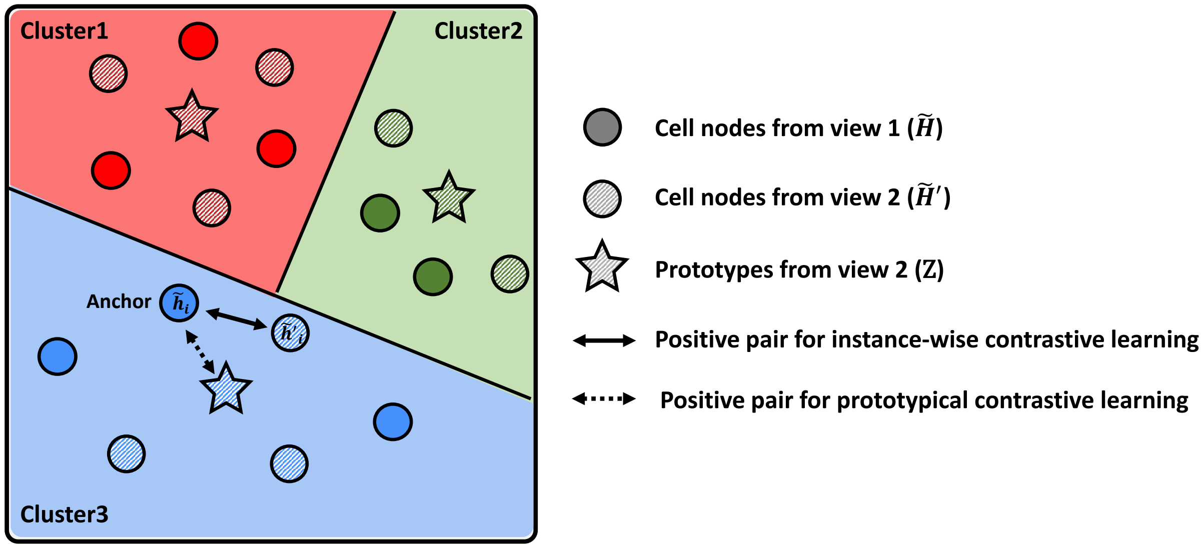
Cluster1
Cluster2
Cluster3
Anchor
$\tilde{h}_i$
$\tilde{h}'_i$
Cell nodes from view 1 ($\tilde{H}$)
Cell nodes from view 2 ($\tilde{H}'$)
Prototypes from view 2 (Z)
Positive pair for instance-wise contrastive learning
Positive pair for prototypical contrastive learning

Figure 6. An overview of prototypical contrastive learning of cell node i on the representation space. Given an anchor node representation of cell i from view 1 denoted by $\tilde{h}_i$, it pulls another cell node representations of the same cell i from view 2 (i.e., $\tilde{h}'_i$), and pushes all other cell representations from both view 1 and view 2 using contrastive loss to learn a generic representation of each cell. In addition, $\tilde{h}_i$ pulls the prototype assigned for the same cell i from view 2, and also pushes all other prototypes to complement the limitation of contrastive loss which suffers from sampling bias.

C. ZINB-based Reconstruction Loss

As we mentioned before, we assume that the gene expression matrix follows the ZINB distribution to capture the characteristic of the scRNA-seq data. Specifically, the ZINB distribution is defined as:

$$\begin{aligned} NB(x|\mu,\theta) &= \frac{\Gamma(x+\theta)}{x!\Gamma(\theta)}\left(\frac{\theta}{\theta+\mu}\right)^{\theta}\left(\frac{\mu}{\theta+\mu}\right)^{x} \\ ZINB(x|\pi,\mu,\theta) &= \pi\delta_0(x)+(1-\pi)NB(x|\mu,\theta) \end{aligned} \tag{8}$$

where μ, θ, and π represent the parameters of ZINB distribution that are mean, dispersion, and dropout probability, respectively. To estimate these parameters, we introduce a shared feed-forward decoder layer g, and an additional layer for each of the three parameters. Specifically, the output of the decoder $D = g(H) \in \mathbb{R}^{N_b \times N_g}$ is independently fed into additional layers for three parameters (i.e., μ, θ, and π) as follows:

$$M = S \times \exp(DW_{\mu}),\ \ \Theta = \exp(DW_{\theta}),\ \ \Pi = \text{sigmoid}(DW_{\pi}) \tag{9}$$

where $S \in \mathbb{R}^{N_b \times N_b}$ is a diagonal matrix whose diagonal element for the i-th row is the size factor (i.e., s_i) of cell i, $M \in \mathbb{R}^{N_b \times N_g}$, $\Theta \in \mathbb{R}^{N_b \times N_g}$, and $\Pi \in \mathbb{R}^{N_b \times N_g}$ are the matrix representation of estimated mean, dispersion, and dropout probability, respectively, and $W_{\mu} \in \mathbb{R}^{N_g \times N_g}$, $W_{\theta} \in \mathbb{R}^{N_g \times N_g}$, and $W_{\pi} \in \mathbb{R}^{N_g \times N_g}$ are trainable parameters. Note that the exponential function is adopted for M and Θ due to the non-negative range of mean and dispersion, whereas the sigmoid function is adopted for Π as the dropout probability lies between 0 and 1. The ZINB-based reconstruction loss for the estimated parameters given by Eqn. 9 is calculated based on the negative log-likehlihood of ZINB distribution as follows:

$$l_{\text{ZINB}}(\Pi, M, \Theta) = \frac{1}{N_b \times N_g}\sum_{i=1}^{N_b}\sum_{j=1}^{N_g} -\log(\text{ZINB}(X^{\text{count}}_{ij} \mid \Pi_{ij}, M_{ij}, \Theta_{ij})) \tag{10}$$

where X^{count} denotes the raw read count matrix, and X^{count}_{ij}, Π_{ij}, M_{ij}, and Θ_{ij} denote the element at the i-th row and the j-th column for each matrix. The overall ZINB-based reconstruction loss of scGPCL from the two views is given by:

$$\mathcal{L}^{\text{Pre}}_{\text{ZINB}} = \frac{1}{2}[l_{\text{ZINB}}(\tilde{\Pi}, \tilde{M}, \tilde{\Theta}) + l_{\text{ZINB}}(\tilde{\Pi}^{'}, \tilde{M}^{'}, \tilde{\Theta}^{'})] \tag{11}$$

where $\tilde{\Pi}$, $\tilde{M}$, and $\tilde{\Theta}$ represent the estimated parameters from the view 1, and $\tilde{\Pi}'$, $\tilde{M}'$, $\tilde{\Theta}'$ denote those from the view 2.

D. Clustering Task-Oriented Loss.

Given a soft cluster assignment distribution matrix $Q \in \mathbb{R}^{N_b \times K}$, where K represents the number of clusters, with each row denoting the soft cluster assignment distribution of each cell, we introduce a target distribution matrix $P \in \mathbb{R}^{N_b \times K}$ that is obtained by sharpening Q, and minimize the Kullback-Leibler (KL) divergence between the two distributions as follows:

$$\mathcal{L}_{\text{Cluster}} = D_{\text{KL}}(P \| Q) = \sum_{i=1}^{N_b} \sum_{k=1}^{K} p_{ik} \log \frac{p_{ik}}{q_{ik}} \tag{12}$$

where q_{ik} and p_{ik} are the assignment probabilities of cell i to cluster k in terms of the soft cluster assignment distribution matrix Q and the target distribution matrix P. Note that q_{ik} is calculated by measuring the similarity between the representation of cell i, (i.e., h_i), and the centroid of cell i, (i.e., c_k), based on the Student's t-distribution as follows:

$$q_{ik} = \frac{(1 + \|h_i - c_k\|_2^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j=1}^{K}(1 + \|h_i - c_j\|_2^2/\alpha)^{-\frac{\alpha+1}{2}}} \tag{13}$$

where α is the degree of freedom of the Student's t-distribution. Then, the target distribution p_{ik} is calculated by normalizing the second power of the soft assignment distribution by the frequency per cluster as follows:

$$p_{ik} = \frac{q_{ik}^2 / f_k}{\sum_{j=1}^{K} q_{ij}^2 / f_j} \tag{14}$$

where $f_k = \sum_{i=1}^{N_b} q_{ik}$ is the soft cluster frequencies used to prevent degenerate solutions in which case some clusters are not assigned any instances at all. In other words, Eqn. 14 sharpens q_{ik} (i.e., p_{ik} is a sharpened version of q_{ik}), by making a large value to be larger and a small value to be smaller. As a result, by minimizing the KL divergence defined in Eqn.12, in which Q and P are defined as in Eqn. 13 and Eqn. 14, respectively, we aim to provide more confident cluster assignments to each cell, which in turn explicitly optimizes the cell representations for the cell clustering task. Note that we run K-means clustering only once before starting the fine-tuning phase to initialize the cluster centroids (i.e., $\{c_k\}_{k=1}^{K}$), and K-means clustering is not performed anymore thereafter. Moreover, to improve the robustness of the clustering assignments, we use $\tilde{H}$ obtained from an augmented graph $\tilde{G}$ (i.e., $\tilde{H} = f(\tilde{G})$) to compute the soft cluster assignment matrix $Q \in \mathbb{R}^{N_b \times K}$.

E. Evaluation metrics

To compare the clustering performance of the scGPCL with the state-of-the-art baselines, we use four standard evaluation metrics, i.e., NMI, CA, ARI, and F1-score, defined as follows:

1) Normalized Mutual Information (NMI) evaluates the clustering quality by measuring the uncertainty of predicted class labels. Specifically, when ground-truth cell type S and the cluster assignment of models C are given, NMI is calculated as follows:

$$NMI = \frac{2 \times I(S;C)}{[H(S) + H(C)]} \tag{15}$$

where $I(\cdot,\cdot)$ denotes the mutual information between two distributions, and H denotes the entropy function. NMI is ranged 0.0 and 1.0 and becomes higher when the predicted cluster assignment and ground-truth are well aligned.

2) Clustering Accuracy (CA) measures the clustering performance in a manner similar to that of the supervised classification. More precisely, we find the best matching function which maps the predicted cluster assignments to the ground-truth cell types, and then calculate the alignment between them. CA is computed as follows:

$$CA = \max_{m} \frac{\sum_{i=1}^{N} \mathbb{1}_{[s_i = m(c_i)]}}{N} \tag{16}$$

where N is the number of instances and m is a matching function which maps the predicted cluster assignments to the ground-truth cell types, and s_i and c_i denote the ground-truth cell type and predicted cluster assignment of the i-th cell, respectively.

3) Adjusted Rand Index (ARI) adjusts the Rand Index (RI) which is defined as

$$RI = \frac{a+b}{{}_{N}C_2} \tag{17}$$

where a represents the number of pairs that successfully belong to the same cluster, while b represents the number of pairs correctly labeled as not belonging to the same cluster. ARI is computed as follows:

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]} \tag{18}$$

where $E[RI]$ means the expectation of RI. ARI is ranged between -1 and 1, and becomes larger when the agreement between ground-truth cell types and predicted cluster assignment is similar.

4) F1-score is the harmonic mean of precision and recall which is suitable for the imbalanced data where the precision is the proportion of positive instances out of positively predicted instances and recall is the proportion of positively predicted instances out of all positive instances on the binary classification setting. F1-score is computed as follows:

$$F1\text{-}score = \frac{2(Precision \times Recall)}{Precision + Recall} \tag{19}$$

To measure the clustering performance, we also find the best matching function to map the predicted cluster assignments to the ground truth cell types and generalize to the multi-class setting by applying micro and macro average scheme (i.e., Micro-F1 and Macro-F1, respectively).

F. Data Statistics.

Data	Sequencing platform	# of Cells	# of Genes	# of Subgroups
Camp	SMARTer	777	19,020	7
Mouse ES cells	inDrop	2,717	24,047	4
Mouse bladder cells	Microwell-seq	2,746	19,771	16
Zeisel	STRT-seq UMI	3,005	19,972	9
Worm neuron cells	sci-RNA-seq	4,186	13,488	10
10X PBMC	10X	4,340	19,773	8
Human kidney cells	10X	5,685	25,215	11
Baron	inDrop	8,569	20,125	14
Shekhar mouse retina cells	Drop-seq	27,499	13,166	19

Table 1. Statistics for real datasets used for experiments.

G. Baseline methods

The cluserting performance of scGPCL is compared with eight state-of-the-arts baseline methods incorporating graph based methods, instance-wise contrastive learning method, ZINB-based autoencoder methods, graph based deep learning methods, and non-deep learning based methods.

- Clustering through imputation and dimensionality reduction (CIDR) (Lin et al., 2017) implicitly imputes gene expression data to alleviate the impact of dropout and then calculates dissimilarity matrix. After that, it performs PCoA and clustering using the first few principal coordinates.
- Deep embedded clustering (DEC) (Xie et al., 2016) jointly optimizes the feature representation and cluster assignments using the self-training strategy.
- Single-cell model-based deep embedded clustering (scDeepCluster) (Tian et al., 2019) simultaneously learns the cell representation and clustering assignment following DEC, and replaces the MSE loss with negative likelihood of ZINB distribution.
- Single-cell graph neural networks (scGNN) (Wang et al., 2021) aggregates the relationship between cells in cell-cell graph using GNNs and adopts left-truncated mixture Gaussian (LTMG) model to reflect the heterogeneous gene expression patterns. In addition, it infuses cell-type specific information using cluster autoencoder.
- Contrastive self-supervised clustering of scRNA-seq data (contrastive-sc) (Ciortan & Defrance, 2021) adopts instance-wise contrastive learning scheme for the scRNA-seq data by randomly generating two differently augmented views in terms of cell features.
- Structural deep clustering for single-cell RNA-seq data (scDSC) (Gan et al., 2022) enhances scDeepCluster by jointly optimizing ZINB-based autoencoder and the GNNs on cell-cell graph to aggregate cell-cell relationship.
- scNAME (Wan et al., 2022) utilizes an auxiliary mask estimation task to grasp the gene pertinence by distinguishing the uncorrupted structure and achieve intra-cluster compacteness and inter-cluster separation using neighborhood contrastive loss that enhances similarity between k-nearest neighbors.

H. Visualization

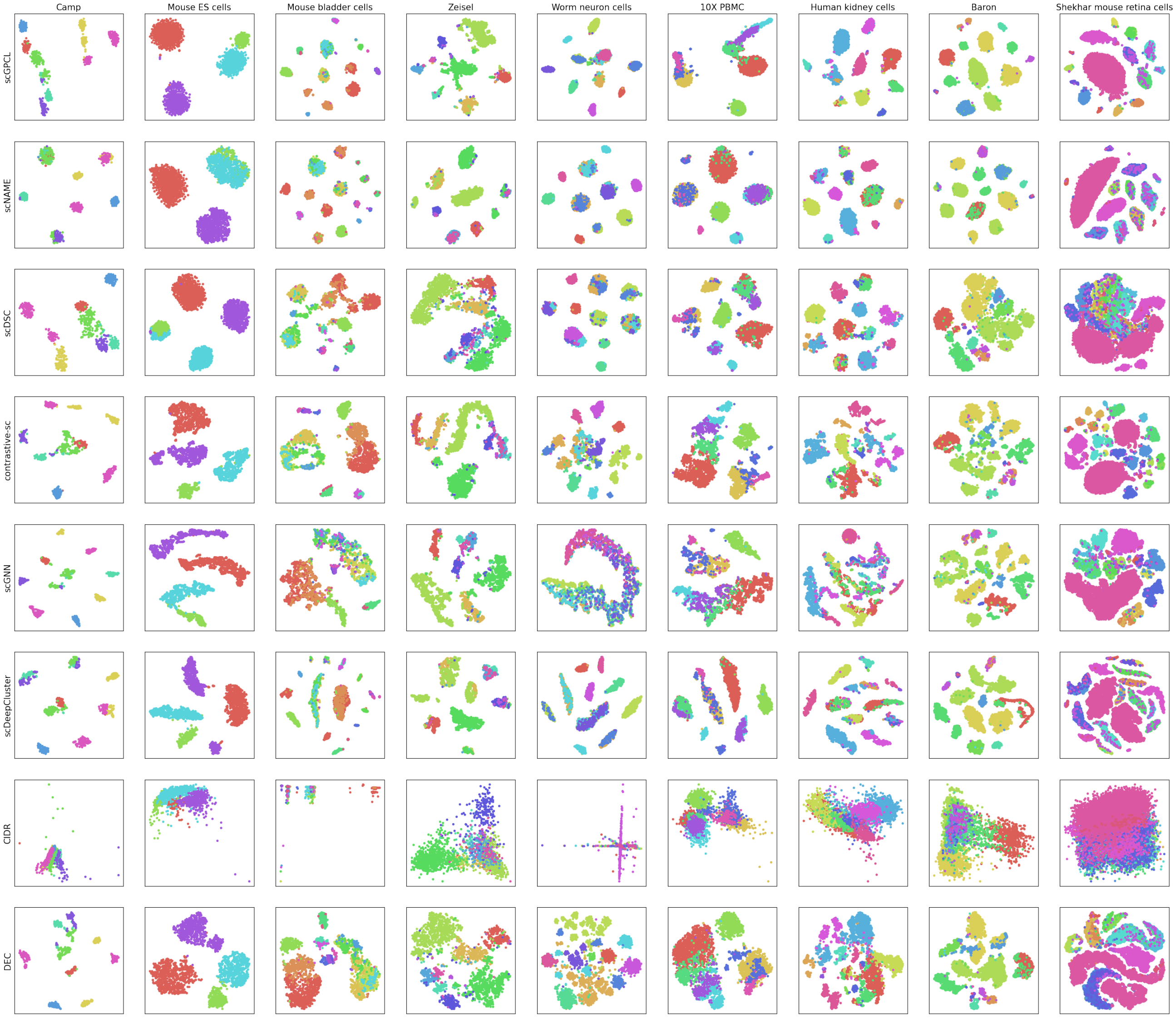

Figure 7. Visualization of cell representations obtained from various methods including scGPCL.

I. Model Analysis

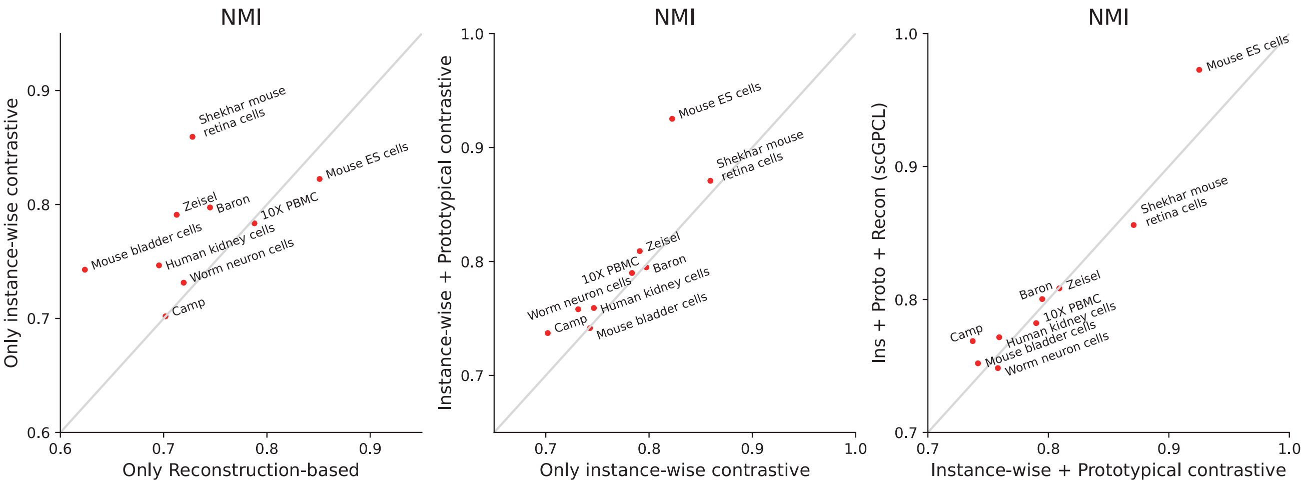

Figure 8. Ablation studies regarding each component in scGPCL.

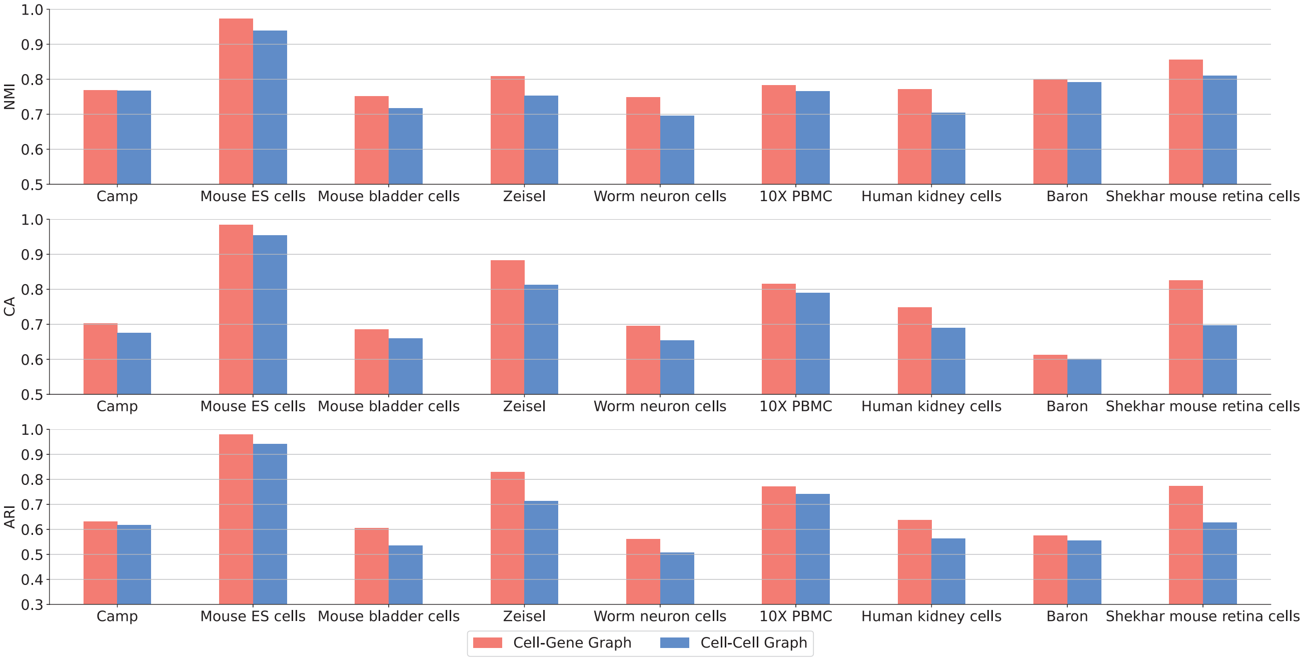

Figure 9. Ablation studies regarding the types of the underlying graph (i.e., cell-gene graph vs. cell-cell graph) for scGPCL.

We conduct ablation studies to clarify the benefits of each component of scGPCL. In Figure 8, we test each loss function in the pre-training phase and have the following observations: 1) Contrastive learning scheme consistently shows an increased performance compared with the one that only uses the reconstruction-based loss (i.e., Recon only) except for 10X PBMC and Worm neuron cells datasets. 2) Using prototypical contrastive loss (i.e., Ins+Proto) is more beneficial than using only instance-wise contrastive learning (i.e., Ins only) because it can alleviate the sampling bias and help to infuse cell type information during the pre-training phase. 3) Adding the reconstruction loss (i.e., scGPCL) is beneficial in some cases, however, it does not show consistent performance improvements. Through these results, we argue that the reconstruction loss can be considered as an auxiliary loss that is helpful in stabilizing the performance, but not the main component of scGPCL.

Furthermore, we conduct ablation studies on the types of the underlying graphs (i.e., cell-gene graph vs. cell-cell graph). Our goal is to verify our claim that it is better to leverage a cell-gene graph to maintain the quality of the constructed graph compared with a cell-cell graph which may have an adverse effect when the quality of constructed graph dropped due to the instability of the pre-computed cell-cell similarity. To this end, in Figure 9, we conduct experiments by changing the type of the underlying input graph of scGPCL to the cell-cell graph, which is constructed based on 10 nearest neighbors of each cell based on the pearson correlation as the similarity measure following scGNN and scDSC, and compare its performance with that of the original scGPCL that uses a cell-gene graph. More precisely, both of them have same decoder structures, but encode the cell representations using GNNs on cell-cell graph and GNNs on cell-gene graph, respectively. We observe that scGPCL with a cell-gene graph as the input consistently outperforms that with a cell-cell graph, which demonstrates that the cell-gene graph better helps to infuse the inherent relational information between cells.