scCGNet: Jointly Optimizing Copula Mixtures and Graph Autoencoders for Single-Cell Clustering

Anonymous authors
Paper under double-blind review

Abstract

Clustering is central to single-cell RNA sequencing (scRNA-seq) analysis but remains difficult: the data are high-dimensional, sparse, and correlated. Recent deep methods increasingly model this correlation to learn better representations-for example, through gene-gene structure or masked reconstruction. The clustering step, however, does not. After embedding, cells are grouped by K-means, deep embedded clustering, or contrastive objectives that rely on distances to cluster centroids. These objectives treat the latent dimensions as independent within a cluster, and cannot capture the non-Gaussian, asymmetric, and tail dependence that transcriptomic data exhibit. We propose scCGNet, which models dependence at the clustering stage. A variational graph autoencoder encodes a cell-cell k-nearest-neighbor graph into topology-aware latent representations. A Gaussian mixture copula model then clusters cells by explicitly modeling the dependence structure within each cluster. A zero-inflated negative binomial layer accounts for sparsity and over-dispersion. The three components share one objective and are trained jointly. On fifteen real scRNA-seq datasets, scCGNet attains the highest and most consistent accuracy, and ablations show that the copula-based cluster model drives the gain.

1 Introduction

Clustering is a standard step in scRNA-seq analysis, and the resulting partition is the basis for cell-type annotation, differential expression analysis, and the detection of rare cell populations Krzak et al. (2019); Chen et al. (2019). scRNA-seq count matrices are high-dimensional and sparse, and gene expression levels are correlated across cells Wu et al. (2024); Angerer et al. (2017).

Deep learning methods model these dependencies to learn informative cell representations. Some reconstruct masked expression values Fang et al. (2024), while others use gene–gene and cell–cell structure with graph neural networks Peng et al. (2026); Van Vinh et al. (2025). ZINB-based models are also widely used to account for sparsity and over-dispersion Tian et al. (2019); Eraslan et al. (2019). However, this dependence is mainly used during representation learning. Once the embedding is obtained, cells are typically grouped using geometric methods such as K-means, deep embedded clustering Tian et al. (2019), or optimal transport Xu et al. (2024). These methods do not explicitly model the joint distribution of latent representations within each cluster. Thus, the dependence captured during representation learning is not fully carried into the clustering step. Mixture models provide a statistical alternative to distance-based clustering. scGMAI Yu et al. (2021) and scGMM-VGAE Lin et al. (2023) integrate Gaussian mixtures into deep representations for clustering. However, Gaussian mixtures tie the marginal distributions to the dependence structure. Copulas separate these two components, allowing dependence to be modeled independently of the marginals. Gaussian mixture copula models (GMCM) extend this idea by capturing richer dependence through a Gaussian mixture Bilgrau et al. (2016); Kasa et al. (2020). Although copula-based models have been used for single-cell factor analysis Choi et al. (2025) and network inference Tomilina et al. (2026), they have not been used as the cluster model itself.

We propose scCGNet, an unsupervised framework that integrates copula-based dependence modeling into deep clustering. A variational graph autoencoder learns a topology-aware representation

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

from a cell–cell k-nearest-neighbor graph, while a Gaussian mixture copula models cluster structure in the latent space. A zero-inflated negative binomial layer models the sparsity and over-dispersion of the raw counts. These components are jointly optimized under a single objective, allowing dependence-aware clustering to shape the learned representation. Unlike scGMM-VGAE, which uses a Gaussian mixture, scCGNet uses a Gaussian mixture copula to separate marginal behavior from dependence at the clustering stage. To the best of our knowledge, it is the first end-to-end deep graph clustering framework to incorporate a copula-based clustering model.

Our contributions are threefold: (1) we identify the gap between dependence-aware representation learning and distance-based clustering; (2) we introduce scCGNet, which jointly learns graph representations, a copula-based cluster model, and ZINB reconstruction; and (3) we evaluate it on fifteen real scRNA-seq datasets against eight recent deep clustering methods, with ablations examining the contribution of copula normalization.

2 Related Work

We group deep clustering methods for scRNA-seq by the objective that determines cluster membership.

Centroid-based clustering. These methods learn a latent representation and assign cells by distance or similarity to cluster centroids. scDeepCluster (Tian et al., 2019) combines a ZINB autoencoder with deep embedded clustering, and scziDesk (Chen et al., 2020) uses weighted soft K-means with self-training. FSSC (Zhong et al., 2026), scAce (He et al., 2023), and scMAE (Fang et al., 2024) differ in representation learning but also use centroid- or partition-based assignment. None of these methods specifies a distribution over the latent variables within a cluster.

Graph-based clustering. These methods encode cell–cell relationships in a graph. scRCL (Peng et al., 2026), scAFGCC (Tian et al., 2025), and scAGCL (Van Vinh et al., 2025) use graph contrastive learning, scCDCG (Xu et al., 2024) combines graph structure with an optimal-transport objective, and other methods combine graph encoders with ZINB reconstruction (Gan et al., 2022; Yu et al., 2022). Their objectives are based on pairwise similarity, connectivity, or transport costs, not on a probabilistic model of within-cluster dependence.

Probabilistic mixture-based clustering. scGMAI (Yu et al., 2021) fits a Gaussian mixture model (GMM) to an autoencoder embedding, and scGMM-VGAE (Lin et al., 2023) integrates a GMM into a variational graph autoencoder. In a GMM, within-cluster dependence is represented only by each component's covariance matrix and is therefore symmetric and determined by linear correlation. scCGNet instead defines cluster distributions with a Gaussian mixture copula model, which specifies the marginal distributions and the dependence structure separately.

3 Method

We propose scCGNet, an end-to-end framework that performs dependence-aware clustering of scRNA-seq data within a learned latent space. A variational graph autoencoder (VGAE) encodes the cell–cell graph into a low-dimensional latent representation $\mathbf{Z}$; on this representation, a Gaussian mixture copula model (GMCM) models cluster structure through the copula density of the latent mixture, a zero-inflated negative binomial (ZINB) head reconstructs the raw counts, and an inner-product decoder reconstructs the graph. Graph reconstruction, KL regularization, GMCM clustering, and the ZINB likelihood are optimized jointly through a single objective. An overview is shown in Figure 1. Algorithm 1 (Appendix E) summarizes training.

3.1 Notation and setup

We are given the raw count matrix $\mathbf{X}^{\mathrm{raw}} \in \mathbf{R}^{n \times p}$ over n cells and p genes, together with its preprocessed (normalized and log-transformed) counterpart $\mathbf{X}^{\mathrm{proc}}$. The encoder reads $\mathbf{X}^{\mathrm{proc}}$, whereas the ZINB head reconstructs $\mathbf{X}^{\mathrm{raw}}$. We form an undirected cell–cell graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ with cells $\mathbf{V} = \{v_1, \ldots, v_n\}$ as nodes, building $\mathbf{E}$ with the k-nearest-neighbor (kNN) algorithm Soucy

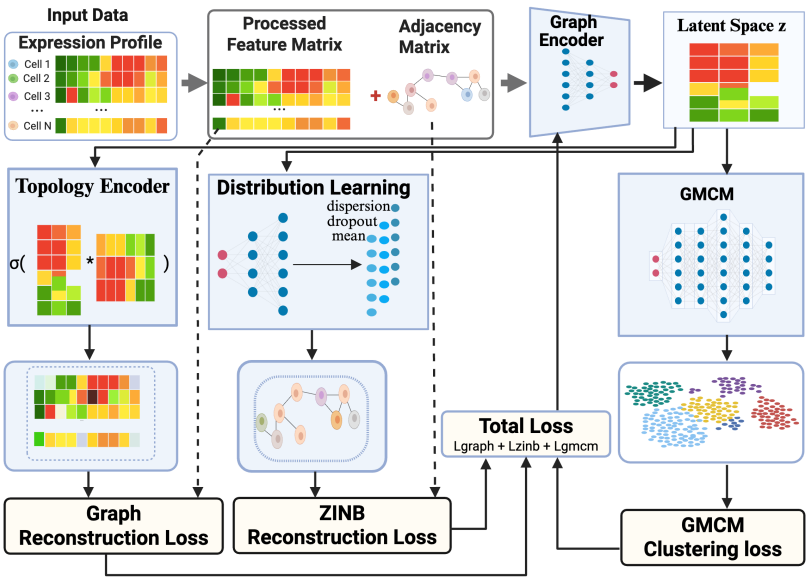


Figure 1: Architecture of scCGNet. A GCN encoder maps the processed counts $\mathbf{X}^{\text{proc}}$ and adjacency $\mathbf{A}$ to a latent representation $\mathbf{Z}$, on which three heads operate jointly: a GMCM head clusters cells, a topology head reconstructs $\mathcal{G}$, and a ZINB head models the raw counts $\mathbf{X}^{\text{raw}}$. The losses form a single end-to-end objective.

& Mineau (2001); Zhang et al. (2018) on $\mathbf{X}^{\text{proc}}$ Amezquita et al. (2020): cells v_i and v_j are connected when one lies among the other's k nearest neighbors, giving the adjacency matrix $\mathbf{A} = [\alpha_{ij}] \in \mathbf{R}^{n\times n}$ ($\alpha_{ij} = 1$ if connected, else 0). We set $k = 15$ Buterez et al. (2022); its effect is analyzed in Section 4.3. The number of clusters K is treated as given; for benchmarking, K is set to the number of annotated cell types in each dataset, and these annotations are used only to fix K, never to train the model.

3.2 Variational Graph Autoencoder (VGAE)

The VGAE encodes $\mathbf{X}^{\text{proc}}$ and $\mathbf{A}$ into a latent matrix $\mathbf{Z} \in \mathbf{R}^{n\times p_L}$ of embedding size p_L. A graph convolutional network (GCN) layer with weights $\mathbf{W}$ and activation ϕ is

$$\text{GCN}_\phi(\mathbf{H}, \mathbf{A} \mid \mathbf{W}) = \phi\Big(\bar{\mathbf{D}}^{-\frac{1}{2}}\bar{\mathbf{A}}\bar{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}\,\mathbf{W}\Big), \qquad \bar{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n, \;\; \bar{\mathbf{D}}_{ii} = \textstyle\sum_j \bar{\mathbf{A}}_{ij}. \tag{1}$$

A shared layer produces a hidden representation $\mathbf{H}$, and two parallel graph-convolution heads with separate weights produce the parameters of the approximate posterior:

$$\mathbf{H} = \text{GCN}_{\text{ReLU}}(\mathbf{X}^{\text{proc}}, \mathbf{A} \mid \mathbf{W}_H), \quad \boldsymbol{\mu} = \text{GCN}_{\text{Lin}}(\mathbf{H}, \mathbf{A} \mid \mathbf{W}_\mu), \quad \log \boldsymbol{\sigma}^2 = \text{GCN}_{\text{Lin}}(\mathbf{H}, \mathbf{A} \mid \mathbf{W}_\sigma), \tag{2}$$

so that

$$q(\mathbf{Z} \mid \mathbf{X}^{\text{proc}}, \mathbf{A}) = \prod_{i=1}^{n} \mathcal{N}\big(\mathbf{z}_i \mid \boldsymbol{\mu}_i, \text{diag}(\boldsymbol{\sigma}_i^2)\big), \qquad \mathbf{z}_i = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}_i, \;\; \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3}$$

where $\odot$ is the elementwise product. All three downstream heads operate on $\mathbf{Z}$; at inference the sampling is disabled and $\mathbf{z}_i = \boldsymbol{\mu}_i$, so the reported clusters are obtained from the posterior mean embedding.

Graph decoder. An inner-product decoder reconstructs the graph from the latent vectors Kipf & Welling (2016a), using $\ell_{ij} = \mathbf{z}_i^\top \mathbf{z}_j$ as the logit of edge (i, j), so that $p(A_{ij} = 1 \mid \mathbf{z}_i, \mathbf{z}_j) = \text{Sigmoid}(\ell_{ij})$. Because $\mathbf{A}$ is sparse, the graph-reconstruction loss is a binary cross-entropy in which present edges are up-weighted by $w_+ = (n^2 - m)/m$, with $m = \sum_{i,j} A_{ij}$ the number of edges:

$$\mathcal{L}_{\text{adj}} = -\frac{1}{n^2}\sum_{i,j}\Big[w_+\, A_{ij} \log \text{Sigmoid}(\ell_{ij}) + (1 - A_{ij}) \log\big(1 - \text{Sigmoid}(\ell_{ij})\big)\Big]. \tag{4}$$

The latent is regularized by the VGAE Kullback–Leibler term Kullback et al. (1987) against the factorized standard normal prior $p(\mathbf{z}_i) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. We scale it by $1/n^2$, which down-weights the KL by a further factor $1/n$ relative to the usual per-cell average:

$$\mathcal{L}_{\mathrm{KL}} = -\frac{1}{2n^2} \sum_{i=1}^{n} \sum_{j=1}^{p_L} \left(1 + \log \sigma_{ij}^2 - \mu_{ij}^2 - \sigma_{ij}^2\right). \tag{5}$$

3.3 Gaussian mixture copula clustering

scCGNet clusters the latent $\mathbf{Z}$ with a Gaussian mixture copula model, capturing the dependence induced by the latent mixture while separating this dependence from the marginal distributions. During training the GMCM operates on the sampled latent $\mathbf{Z}$; at inference it operates on the posterior mean $\boldsymbol{\mu}$.

Copula. For a continuous d-variate distribution with density, Sklar's theorem Sklar (1959) gives the factorization $f(z_1, \ldots, z_d) = c\big(u_1, \ldots, u_d\big) \prod_{j=1}^{d} f_j(z_j)$, with $u_j = F_j(z_j)$, where the copula density c carries the dependence structure and the f_j are the marginal densities. The copula thus isolates the dependence among coordinates from their marginal distributions.

Latent Gaussian mixture. We place a K-component Gaussian mixture with *diagonal* covariance on the latent space. For cluster indicator $C \sim \mathrm{Categorical}(\pi_1, \ldots, \pi_K)$ ($\pi_k > 0$, $\sum_k \pi_k = 1$), component means $\boldsymbol{\eta}_k$, and per-dimension variances σ_{kj}^2, the joint mixture density, its j-th marginal density, and the corresponding marginal CDF are

$$g(\mathbf{z}; \theta) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{p_L} \mathcal{N}(z_j; \eta_{kj}, \sigma_{kj}^2), \quad g_j(z_j; \theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(z_j; \eta_{kj}, \sigma_{kj}^2), \quad G_j(z_j; \theta) = \int_{-\infty}^{z_j} g_j(t; \theta)\, dt, \tag{6}$$

with $\theta = \{\pi_k, \boldsymbol{\eta}_k, \sigma_k^2\}_{k=1}^{K}$. Although each component is diagonal, the *mixture* g is not a product of its marginals: the cluster structure induces dependence across latent dimensions. This mixture-induced dependence, rather than any within-component covariance, is what the copula models.

Copula density and clustering objective. The Gaussian mixture copula model Bilgrau et al. (2016); Tewari et al. (2011); Kasa et al. (2020); Tewari (2023) is the copula of this mixture. Writing $u_j = G_j(z_j; \theta)$ for the probability-integral transform of coordinate j, its density is obtained by normalizing the joint density by its marginals,

$$c(\mathbf{u}; \theta) = \frac{g(\mathbf{z}; \theta)}{\prod_{j=1}^{p_L} g_j(z_j; \theta)}, \qquad \log c(\mathbf{u}; \theta) = \log g(\mathbf{z}; \theta) - \sum_{j=1}^{p_L} \log g_j(z_j; \theta). \tag{7}$$

The marginal normalization factors out the per-dimension marginal densities, leaving the copula density to characterize the dependence structure of the latent mixture; this is what distinguishes the objective from a plain Gaussian mixture. Because the latent representation $\mathbf{Z}$ is available directly, the copula likelihood is evaluated through this density-ratio form without explicitly inverting the marginal CDFs, and the resulting objective is differentiable with respect to both the neural and mixture parameters. The clustering loss is the negative log copula likelihood over the n cells,

$$\mathcal{L}_{GMCM} = -\frac{1}{n} \sum_{i=1}^{n} \log c(\mathbf{u}_i; \theta) = -\frac{1}{n} \sum_{i=1}^{n} \Big[\log g(\mathbf{z}_i; \theta) - \sum_{j=1}^{p_L} \log g_j(z_{ij}; \theta) \Big], \qquad u_{ij} = G_j(z_{ij}; \theta). \tag{8}$$

Using only the joint mixture likelihood, $-\frac{1}{n} \sum_i \log g(\mathbf{z}_i; \theta)$, yields the corresponding GMM ablation (Section 4.4), which assesses the contribution of the copula normalization.

Cluster assignment. Because the marginal denominator in Eq. 7 is common to all mixture components, it cancels when computing component posterior probabilities, so the responsibilities take the ordinary mixture form

$$\gamma_{ik} = \frac{\pi_k \prod_j \mathcal{N}(z_{ij}; \eta_{kj}, \sigma_{kj}^2)}{\sum_{k'=1}^{K} \pi_{k'} \prod_j \mathcal{N}(z_{ij}; \eta_{k'j}, \sigma_{k'j}^2)}, \qquad \hat{k}_i = \arg\max_k \gamma_{ik}. \tag{9}$$

Dependence therefore enters during training, through the copula-normalized objective (Eq. 8) that shapes the mixture parameters θ and the latent representation, while the final component posterior probabilities retain the standard mixture-responsibility form. The parameters θ are learnable and optimized jointly with the neural parameters by gradient descent; the mixture weights and per-dimension log-variances are initialized to uniform and zero, and the component means $\boldsymbol{\eta}_k$ are warm-started from k-means centroids on a pretrained embedding.

3.4 Count modeling and joint optimization

To account for count sparsity and over-dispersion, a ZINB decoder Yau et al. (2003); Moghimbeigi et al. (2008); Preisser et al. (2016); Nguyen et al. (2023) maps the latent $\mathbf{Z}$, through a hidden layer, to per-gene parameters—dropout rate π_{ij}, dispersion θ_{ij}, and mean $\mu_{ij}^{(x)} = S_i \exp(\cdot)$—and reconstructs the raw counts $\mathbf{X}^{\text{raw}}$. The scalar size factor $S_i = \sum_j X_{ij}^{\text{raw}} / \text{median}_{i'} \sum_j X_{i'j}^{\text{raw}}$ normalizes for library size. Assuming conditional independence across genes given these parameters, the per-gene ZINB likelihood is

$$p(x_{ij}^{\text{raw}} \mid \pi_{ij}, \mu_{ij}^{(x)}, \theta_{ij}) = \pi_{ij}\, \delta_0(x_{ij}^{\text{raw}}) + (1 - \pi_{ij})\, \text{NB}(x_{ij}^{\text{raw}} \mid \mu_{ij}^{(x)}, \theta_{ij}), \tag{10}$$

with δ_0 the point mass at zero and $\text{NB}(x \mid \mu, \theta) = \frac{\Gamma(x+\theta)}{x!\,\Gamma(\theta)} \left(\frac{\theta}{\theta+\mu}\right)^{\theta} \left(\frac{\mu}{\theta+\mu}\right)^{x}$, giving the loss

$$\mathcal{L}_{\text{ZINB}} = -\frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} \log p(x_{ij}^{\text{raw}} \mid \pi_{ij}, \mu_{ij}^{(x)}, \theta_{ij}). \tag{11}$$

The four terms are combined into a single objective, optimized end to end by gradient descent. Each term carries a weight, all set to one by default, so the total loss is

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{adj}} + \mathcal{L}_{\text{KL}} + \mathcal{L}_{GMCM} + \mathcal{L}_{\text{ZINB}}. \tag{12}$$

4 Experiment

4.1 Datasets and settings

Datasets We evaluate all methods on 15 publicly available scRNA-seq datasets that span a wide range of cell counts, feature dimensions, and numbers of annotated cell types. Following their scale, we group them into *small* (fewer than 5,000 cells), *medium* (5,000–18,000 cells), and *large* (more than 18,000 cells) datasets. Seven of the tissue subsets–QS-Diaphragm, QS-Limb Muscle, QS-Trachea, QS-Lung, Qx-Bladder, Qx-Limb Muscle, and Qx-Spleen –are drawn from the mouse Tabula Muris atlas (Schaum et al., 2018), where 'QS' and 'Qx' denote the Quake Smart-seq2 and Quake 10x protocols, respectively. The remaining datasets cover human pancreas (Muraro, Muraro et al., 2016), mouse hypothalamus (Romanov, Romanov et al., 2017), human kidney (Young, Young et al., 2018), mouse mammary epithelium (Bach, Bach et al., 2017), and turtle pallium (Tosches, Tosches et al., 2018), together with the Adam, Cao_2020 Spleen, and Wang Large Intestine datasets. Across the benchmark, dataset sizes range from 870 to 23,184 cells and from 4 to 11 annotated cell types, providing a diverse testbed for evaluating clustering performance. To obtain settings with known cluster structure, we use additional nine simulated datasets generated with SPARSim Baruzzo et al. (2020) using the Chu Chu et al. (2016) and Zheng Zheng et al. (2017) templates, spanning 379 to 50,820 cells for a controlled assessment of stability and scalability. Dataset characteristics are given in Appendix A (Table 5)

Evaluation metrics We assess clustering performance using the adjusted Rand index (ARI) Yeung & Ruzzo (2001) and normalized mutual information (NMI) Kraskov & Grassberger (2009). Higher values indicate better agreement, with 1 denoting perfect agreement. Formal definitions are provided in Appendix F.

Baseline methods We compare scCGNet against eight recent deep clustering methods: scDeepCluster (Tian et al., 2019), scziDesk (Chen et al., 2020), FSSC (Zhong et al., 2026), scAce (He et al., 2023), scMAE (Fang et al., 2024), scRCL (Peng et al., 2026),scCDCG (Xu et al., 2024), scAGCL (Van Vinh et al., 2025). Full descriptions are given in Appendix G.

Table 1: ARI on 15 scRNA-seq datasets grouped by size. Best results are in bold and second-best results are underlined. Avg. denotes the average ARI across datasets. Dataset abbreviations: Dia, Diaphragm; QSM, QS-Muscle; Tra, Trachea; Lung, Lung; Mur, Muraro; Bla, Bladder; Rom, Romanov; Adam, Adam; 10xM, 10x-Muscle; Young, Young; Spl, Spleen; CSpl, Cao Spleen; Wang, Wang; Tos, Tosches; Bach, Bach.

	Small									Medium				Large		
Method	Dia	QSM	Tra	Lung	Mur	Bla	Rom	Adam	10xM	Young	Spl	CSpl	Wang	Tos	Bach	Avg.
scDeepCluster	0.6618	0.4497	0.6794	0.6181	0.9339	0.7507	0.5490	0.7231	0.7765	0.4601	0.5854	**0.8162**	0.2702	0.6599	0.8105	0.6377
scziDesk	0.9650	0.6632	0.5238	0.5882	0.9084	0.7354	0.6807	0.7392	0.9813	0.5816	0.8874	0.0073	0.2470	0.6282	0.8368	0.7119
scAce	0.9787	0.6380	0.5655	0.7666	0.9240	0.4899	0.4064	**0.9255**	0.9922	0.6471	0.2956	0.8001	0.3042	0.8216	0.6533	0.6720
FSSC	0.9742	0.8936	0.7465	0.7888	0.9241	0.7538	0.6856	0.6649	0.9608	0.6234	0.6659	0.7956	0.3684	0.6742	**0.8604**	0.7560
scMAE	0.9592	**0.9761**	0.5565	0.7541	0.8917	0.7033	0.6398	0.7016	0.8065	0.3645	0.9114	0.4181	0.1780	0.7222	0.8227	0.7134
scRCL	0.9852	0.9673	0.5152	0.5956	0.8785	0.9965	0.7886	0.5562	0.9523	0.5486	0.9018	0.4518	0.4277	0.5766	0.7199	0.7436
scCDCG	0.4151	0.3295	0.6587	0.2336	**0.9436**	0.4033	0.5636	0.2348	0.5102	0.5401	0.2993	0.7131	0.2444	0.8180	0.6778	0.4909
scAGCL	0.8245	0.5813	0.2135	0.3535	0.9370	0.5104	0.4326	0.2411	0.3710	0.3910	0.2307	0.4246	0.2515	0.5548	0.8116	0.4789
scCGNet	**0.9911**	0.9652	**0.7960**	**0.9128**	0.9285	**0.9967**	**0.8256**	0.9058	**0.9959**	**0.7489**	**0.9426**	0.7120	**0.4892**	**0.8463**	0.8383	**0.8702**

Table 2: NMI on 15 scRNA-seq datasets grouped by size. Best results are in bold and second-best results are underlined. Avg. denotes the average NMI across datasets. Dataset abbreviations are as in Table 1.

	Small									Medium				Large		
Method	Dia	QSM	Tra	Lung	Mur	Bla	Rom	Adam	10xM	Young	Spl	CSpl	Wang	Tos	Bach	Avg.
scDeepCluster	0.8109	0.5862	0.5636	0.7401	0.8967	0.8042	0.5833	0.8142	0.8727	0.6234	0.5733	**0.8196**	0.3739	0.7816	0.8089	0.7102
scziDesk	0.9427	0.8322	0.5979	0.7639	0.8575	0.7662	0.6891	0.8232	0.9607	0.7342	0.7946	0.0566	0.3364	0.7172	0.8388	0.7141
scAce	0.9579	0.7756	0.7091	0.8014	0.8788	0.6751	0.5540	**0.9049**	0.9849	0.7586	0.4693	0.7953	0.4250	0.7757	0.7853	0.7501
FSSC	0.9588	0.8914	0.7354	0.8176	0.8831	0.8112	0.6878	0.6963	0.9425	0.6652	0.6281	0.8000	0.3940	0.6586	0.8403	0.7607
scMAE	0.9159	**0.9607**	0.5978	0.8164	0.8486	0.7457	0.6727	0.7864	0.8836	0.4742	0.8252	0.5431	0.2531	0.7886	0.8179	0.7287
scRCL	0.9726	0.9453	0.6148	0.7389	0.8261	0.9837	0.7213	0.6448	0.9267	0.5224	0.8158	0.6122	0.4307	0.6918	0.8005	0.7498
scCDCG	0.6121	0.3274	0.5574	0.4449	**0.9002**	0.6127	0.5708	0.4715	0.7333	0.6772	0.5185	0.7615	0.3828	0.7552	0.7778	0.6069
scAGCL	0.7924	0.6977	0.4598	0.5760	0.8948	0.6124	0.4840	0.3448	0.5518	0.5135	0.4099	0.5251	0.3986	0.6301	0.7990	0.5793
scCGNet	**0.9826**	0.9500	**0.7790**	**0.8919**	0.8929	**0.9905**	**0.7574**	0.8876	**0.9895**	**0.8153**	**0.8444**	0.7128	**0.4966**	**0.8283**	**0.8454**	**0.8443**

settings scCGNet is implemented in PyTorch and trained full-batch with Adam. Inputs are the 2,000 most variable genes, and the cell–cell graph connects each cell to its 15 nearest neighbours under cosine distance. Training consists of reconstruction pretraining, k-means initialization of the copula component means, and joint optimization of all losses with equal weights. The number of clusters is set to the number of annotated cell types for all methods. Full details and hyperparameters are given in Appendix K.

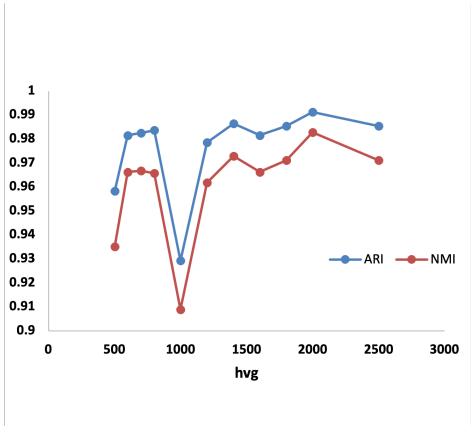
4.2 Quantitative results

Averaged over the 15 datasets, scCGNet obtains the highest ARI (0.870) and NMI (0.844) among the ten methods compared (Tables 1 and 2; Appendix H). The most competitive baselines, FSSC (0.756 / 0.760) and scRCL (0.724 / 0.749), are outperformed by a substantial margin of approximately 0.11 in each metric, whereas scCDCG and scAGCL are the least effective, with mean ARI of 0.506 and 0.475. At the level of individual datasets, scCGNet ranks first on 10 of the 15 datasets in ARI and 11 of the 15 in NMI (Tables 1 and 2), and attains at least the second-best result on most of the remaining datasets. Averages over the fifteen datasets are also shown in Figure 9 (Appendix H). Results on simulated data are sumarized in Appendix C (Table 7, Figure 7)

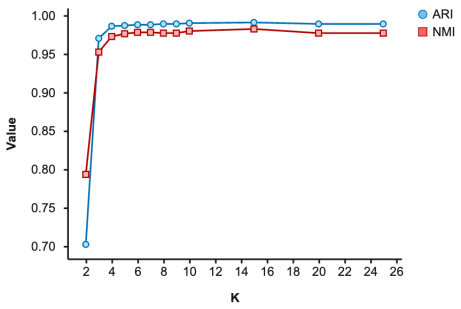
4.3 Parameter analysis

We examine scCGNet's sensitivity to its two preprocessing hyperparameters: the number of highly variable genes (HVGs) and the graph connectivity k (Figure 2).

Highly variable genes. Varying the HVG cut-off from 500 to 2500, accuracy peaks at 2000 HVGs (ARI ≈ 0.991, NMI ≈ 0.983), aside from a transient dip at 1000; we therefore retain 2000 HVGs.

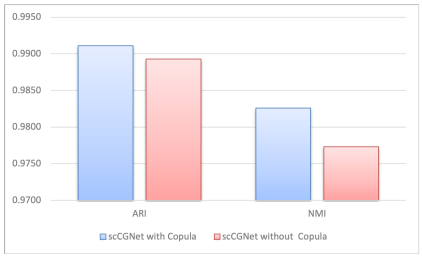


(a) Number of highly variable genes.

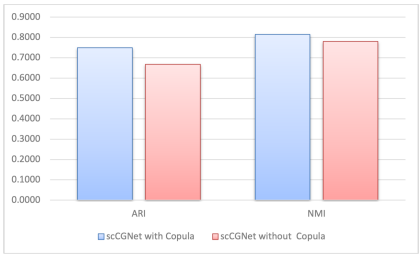


(b) Neighbourhood size k.

Figure 2: Parameter sensitivity of scCGNet using QS-Diaphragm dataset, reporting ARI and NMI as a function of (a) the number of highly variable genes retained and (b) the k-nearest-neighbour graph connectivity.



(a) Diaphragm.



(b) Young.

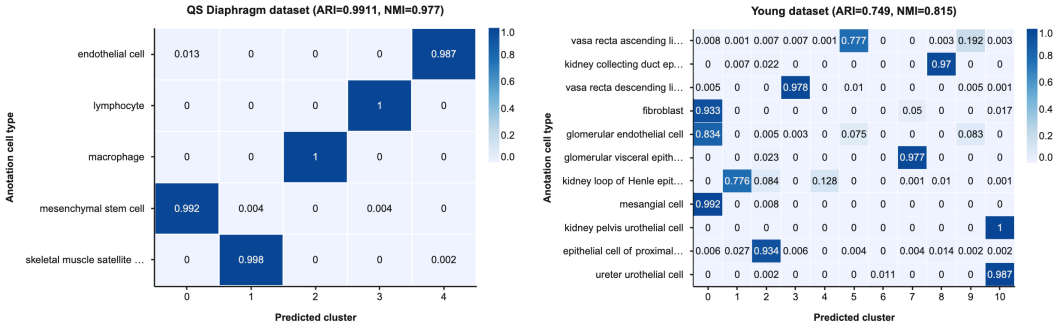
Figure 3: Effect of the copula normalization. scCGNet with the full copula objective versus its Gaussian-mixture variant (marginal term removed) on Diaphragm and Young.

Neighbourhood size k. Sweeping $k \in \{2, \ldots, 25\}$, performance is depressed only at $k = 2$ (ARI 0.702) and saturates from $k = 3$ onward, reaching its maximum at $k = 15$ (ARI 0.991, NMI 0.983); we set $k = 15$.

4.4 Ablation Study

Effect of the copula term. To isolate the contribution of the copula normalization, we compare the full objective (Eq. 8) against its ablated form, which drops the marginal term $\sum_j \log g_j$ and reduces the clustering loss to an ordinary Gaussian mixture. The copula term improves both ARI and NMI on both datasets (Figure 3): ARI rises from 0.9893 to 0.9911 on Diaphragm and from 0.6673 to 0.7489 on Young, with consistent gains in NMI. The improvement is uniform in direction and larger on the harder Young dataset, where more headroom remains. Additional loss-combination ablations are in Appendix B.

Cluster model on fixed embeddings. To examine the cluster model separately from representation learning, we fitted a standalone GMCM, a GMM, and k-means to fixed embeddings and to synthetic data (Appendix J). On synthetic clusters whose marginals were made skewed or heavy-tailed by strictly increasing transformations, the GMCM kept a mean ARI of 0.999, whereas the GMM fell to 0.800–0.882 and k-means to 0.425–0.748. On the embeddings of four real datasets, the within-type marginals were skewed and heavy-tailed, but the standalone GMCM did not improve on the GMM (ARI lower on three of four datasets, by at most 0.031). The benefit of scCGNet is therefore not explained by the copula cluster model applied to a fixed embedding.

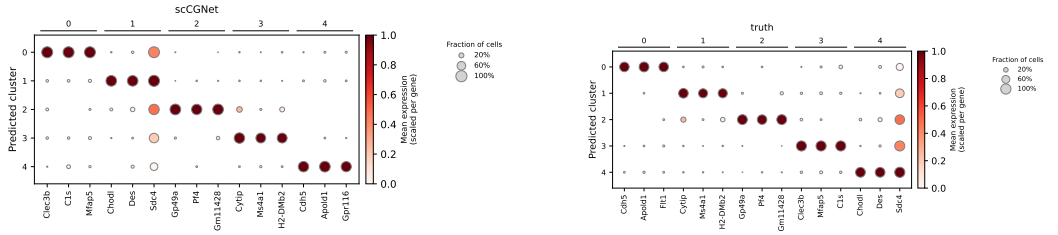


(a) QS-Diaphragm. (b) Young.

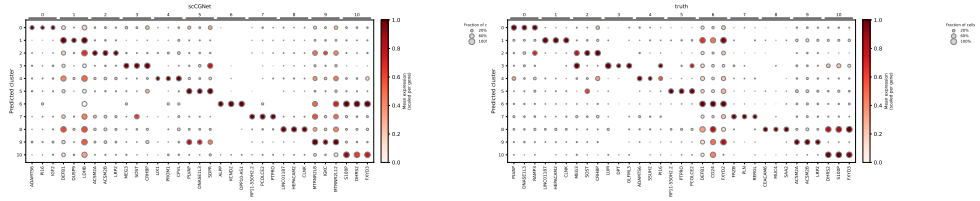
Figure 4: Cell-type recovery. Row-normalized confusion between annotated cell types (rows) and scCGNet clusters (columns) on Diaphragm and Young.

4.5 Biological analysis

Cell-type recovery. To assess the biological relevance of the recovered clusters, we compare scCGNet's predictions with the annotated cell types on QS-Diaphragm and Young (Figure 4). On Diaphragm, all five cell types form distinct clusters with purities of at least 98.7%. On the more complex Young kidney dataset (11 cell types, ARI 0.749), most populations are also recovered with high purity, including kidney pelvis and ureter urothelial cells, mesangial cells, vasa recta, collecting-duct, glomerular-epithelial, and proximal-tubular populations, all exceeding 93%. The main disagreements occur among transcriptionally related populations, with mesangial cells, fibroblasts, and glomerular endothelial cells partially grouped together, and some mixing between adjacent nephron segments. These patterns are consistent with the known biological relatedness of these populations and suggest that the embedding captures meaningful structure in the kidney tissue.



(a) QS-Diaphragm, scCGNet clusters. (b) QS-Diaphragm, ground-truth labels.



(c) Young, scCGNet clusters. (d) Young, ground-truth labels.

Figure 5: Expression of the top three marker genes per group on QS-Diaphragm and Young. Rows are groups and columns are marker genes, arranged by the group for which they were selected. Dot size is the fraction of cells in the group in which the gene is detected; color is the mean log-normalized expression, scaled to $[0, 1]$ per gene.

Marker genes. To check whether scCGNet clusters correspond to annotated cell types in gene expression, we identified the top three marker genes of each cluster and of each ground-truth label with the same procedure (Appendix I). We show the two datasets with gene identifiers, QS-Diaphragm and Young (Figure 5; Tables 4 and 3). On QS-Diaphragm, each of the five clusters matches one

label, with sizes differing by at most one cell; four matched pairs share all three markers and the fifth shares two. On Young, five of the eleven labels share all three markers with a cluster. The remaining differences occur where clusters merge or split labels: cluster 0 merges labels 4, 7, and 3, cluster 10 merges labels 10 and 8, and labels 0 and 6 are each split between two clusters. The mean difference between marker detection rates inside and outside a group is 0.824 for both scCGNet and the ground truth on QS-Diaphragm, and 0.641 versus 0.652 on Young. Across four datasets, 16 of 28 labels share all three markers with a cluster and 24 share at least two (Table 8).

Table 3: Young: correspondence between scCGNet clusters and ground-truth labels, from the confusion matrix. Contributions of more than 50 cells are listed, and all cells for cluster 6.

Cluster	Ground-truth label(s) (cells)	scCGNet markers	Shared with label
3	2 (715)	*MEG3, SOST, CRHBP*	3/3
7	5 (253)	*RP11-550H2.2, PCOLCE2, PTPRO*	3/3
8	1 (260)	*LINC01187, HEPACAM2, CLNK*	3/3
2	9 (451), 6 (126)	*ACSM2A, ACSM2B, LRP2*	3/3 (label 9)
10	10 (613), 8 (73)	*S100P, DHRS2, FXYD3*	3/3 (label 10)
5	0 (933)	*PLVAP, DNASE1L3, SDPR*	2/3
0	4 (311), 7 (117), 3 (56)	*ADAMTS6, PI16, IGF2*	2/3 (label 4)
1	6 (1,163)	*DEFB1, DUSP9, LDHB*	1/3
4	6 (191)	*LIX1, PROM1, CPVL*	0/3
9	0 (230)	*MTRNR2L8, IGKC, MTRNR2L12*	0/3
6	10 (7)	*ALPP, KCND2, DPP10-AS1*	0/3

Table 4: QS-Diaphragm: correspondence between scCGNet clusters and ground-truth labels.

Cluster	Size	Label	Size	scCGNet markers	Shared
0	240	3	241	*Clec3b, C1s, Mfap5*	3/3
1	439	4	439	*Chodl, Des, Sdc4*	3/3
2	31	2	31	*Gp49a, Pf4, Gm11428*	3/3
3	82	1	81	*Cytip, Ms4a1, H2-DMb2*	3/3
4	78	0	78	*Cdh5, Apold1, Gpr116*	2/3

5 Conclusion

We presented scCGNet, an end-to-end framework for clustering scRNA-seq data by integrating a GMCM in the latent space of a variational graph autoencoder with a zero-inflated negative binomial decoder. Across fifteen real scRNA-seq datasets, scCGNet achieves the highest and most consistent clustering accuracy, while ablation studies indicate that the copula-based clustering model is the primary contributor to this improvement. These results show that modeling dependencies among latent representations can improve cluster separation beyond the representation learning component alone. Overall, scCGNet provides a unified framework that combines graph-based representation learning, count-aware reconstruction, and flexible probabilistic clustering for scRNA-seq analysis. The limitations of our method are discussed in Appendix L.

6 Ethics Statement

This work proposes a clustering method for single-cell RNA sequencing (scRNA-seq) data. All experiments use publicly available scRNA-seq datasets generated and released by their original studies; we use only the published count matrices and cell-type annotations. No new data were collected, and no human-subject, animal, or wet-lab experiments were conducted. The datasets contain no

personally identifiable information. A potential misuse of any unsupervised clustering method is to treat inferred clusters as validated cell types; the clusters produced by scCGNet are computational hypotheses and should be confirmed by marker-gene analysis or experimental validation before biological conclusions are drawn.

7 Reproducibility Statement

The source code of scCGNet, implemented in PyTorch, is provided at `ANONYMOUS_LINK`, together with the preprocessing, training, and evaluation scripts. Section 3 and the appendix specify the model: the ZINB variational graph autoencoder, the Gaussian mixture copula clustering objective, and the full training procedure (Algorithm 1). The appendix also describes the preprocessing pipeline (gene filtering, highly variable gene selection, and cell-graph construction), lists the sources of the 15 public datasets, and reports the hyperparameters used for each dataset. The number of clusters K is set to the number of annotated cell types. All baselines are run with their publicly released implementations under the settings described in the appendix.

References

Mike Adam, Andrew S Potter, and S Steven Potter. Psychrophilic proteases dramatically reduce single-cell rna-seq artifacts: a molecular atlas of kidney development. *Development*, 144(19): 3625–3632, 2017.

Robert A Amezquita, Aaron TL Lun, Etienne Becht, Vince J Carey, Lindsay N Carpp, Ludwig Geistlinger, Federico Marini, Kevin Rue-Albrecht, Davide Risso, Charlotte Soneson, et al. Orchestrating single-cell analysis with bioconductor. *Nature methods*, 17(2):137–145, 2020.

Philipp Angerer, Lukas Simon, Sophie Tritschler, F Alexander Wolf, David Fischer, and Fabian J Theis. Single cells make big data: new challenges and opportunities in transcriptomics. *Current opinion in systems biology*, 4:85–91, 2017.

Karsten Bach, Sara Pensa, Marta Grzelak, James Hadfield, David J. Adams, John C. Marioni, and Walid T. Khaled. Differentiation dynamics of mammary epithelial cells revealed by single-cell rna sequencing. *Nature Communications*, 8(1):1–11, 2017.

Giacomo Baruzzo, Ilaria Patuzzi, and Barbara Di Camillo. Sparsim single cell: a count data simulator for scrna-seq data. *Bioinformatics*, 36(5):1468–1475, 2020.

Anders Ellern Bilgrau, Poul Svante Eriksen, Jakob Gulddahl Rasmussen, Hans Erik Johnsen, Karen Dybkær, and Martin Bøgsted. Gmcm: Unsupervised clustering and meta-analysis using gaussian mixture copula models. *Journal of Statistical Software*, 70:1–23, 2016.

David Buterez, Ioana Bica, Ifrah Tariq, Helena Andrés-Terré, and Pietro Liò. Cellvgae: an unsupervised scrna-seq analysis workflow with graph attention networks. *Bioinformatics*, 38(5): 1277–1286, 2022.

Junyue Cao, Diana R O'Day, Hannah A Pliner, Paul D Kingsley, Mei Deng, Riza M Daza, Michael A Zager, KA Aldinger, R Blecher-Gonen, F Zhang, et al. A human cell atlas of fetal gene expression. science 370, eaba7721, 2020.

Geng Chen, Baitang Ning, and Tieliu Shi. Single-cell rna-seq technologies and related computational data analysis. *Frontiers in genetics*, 10:317, 2019.

Liang Chen, Weinan Wang, Yuyao Zhai, and Minghua Deng. Deep soft k-means clustering with self-training for single-cell rna sequence data. *NAR genomics and bioinformatics*, 2(2):lqaa039, 2020.

Junsouk Choi, Hee Cheol Chung, Irina Gaynanova, and Yang Ni. Bayesian segmented gaussian copula factor model for single-cell sequencing data. *Bayesian Analysis*, 1(1):1–26, 2025.

Li-Fang Chu, Ning Leng, Jue Zhang, Zhonggang Hou, Daniel Mamott, David T Vereide, Jeea Choi, Christina Kendziorski, Ron Stewart, and James A Thomson. Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome biology*, 17(1):173, 2016.

Gökcen Eraslan, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis. Single-cell rna-seq denoising using a deep count autoencoder. *Nature communications*, 10(1):390, 2019.

Zhaoyu Fang, Ruiqing Zheng, and Min Li. scmae: a masked autoencoder for single-cell rna-seq clustering. *Bioinformatics*, 40(1):btae020, 2024.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Yanglan Gan, Xingyu Huang, Guobing Zou, Shuigeng Zhou, and Jihong Guan. Deep structural clustering for single-cell rna-seq data jointly through autoencoder and graph neural network. *Briefings in Bioinformatics*, 23(2):bbac018, 2022.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M Mauck, Shiwei Zheng, Andrew Butler, Maddie J Lee, Aaron J Wilk, Charlotte Darby, Michael Zager, et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

Yuhan Hao, Tim Stuart, Madeline H Kowalski, Saket Choudhary, Paul Hoffman, Austin Hartman, Avi Srivastava, Gesmira Molla, Shaista Madad, Carlos Fernandez-Granda, et al. Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature biotechnology*, 42(2): 293–304, 2024.

Xinwei He, Kun Qian, Ziqian Wang, Shirou Zeng, Hongwei Li, and Wei Vivian Li. scace: an adaptive embedding and clustering method for single-cell gene expression data. *Bioinformatics*, 39(9):btad546, 2023.

Siva Rajesh Kasa, Sakyajit Bhattacharya, and Vaibhav Rajan. Gaussian mixture copulas for high-dimensional clustering and dependency-based subtyping. *Bioinformatics*, 36(2):621–628, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016a.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016b.

Alexander Kraskov and Peter Grassberger. Mic: Mutual information based hierarchical clustering. In *Information theory and statistical learning*, pp. 101–123. Springer, 2009.

Monika Krzak, Yordan Raykov, Alexis Boukouvalas, Luisa Cutillo, and Claudia Angelini. Benchmark and parameter sensitivity analysis of single-cell rna sequencing clustering methods. *Frontiers in genetics*, 10:1253, 2019.

Solomon Kullback et al. The kullback-leibler distance. *SID*, 1987.

Xiangjie Li, Kui Wang, Yafei Lyu, Huize Pan, Jingxiao Zhang, Dwight Stambolian, Katalin Susztak, Muredach P Reilly, Gang Hu, and Mingyao Li. Deep learning enables accurate clustering with batch effect removal in single-cell rna-seq analysis. *Nature communications*, 11(1):2338, 2020.

Eric Lin, Boyuan Liu, Leann Lac, Daryl LX Fung, Carson K Leung, and Pingzhao Hu. scgmm-vgae: a gaussian mixture model-based variational graph autoencoder algorithm for clustering single-cell rna-seq data. *Machine Learning: Science and Technology*, 4(3):035013, 2023.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

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

Volodymyr Melnykov and Ranjan Maitra. Finite mixture models and model-based clustering. *Statistics Surveys*, 2010.

Abbas Moghimbeigi, Mohammed Reza Eshraghian, Kazem Mohammad, and Brian Mcardle. Multilevel zero-inflated negative binomial regression modeling for over-dispersed count data with extra zeros. *Journal of Applied Statistics*, 35(10):1193–1202, 2008.

Mauro J Muraro, Gitanjali Dharmadhikari, Dominic Grün, Nathalie Groen, Tim Dielen, Erik Jansen, Leon Van Gurp, Marten A Engelse, Francoise Carlotti, Eelco Jp De Koning, et al. A single-cell transcriptome atlas of the human pancreas. *Cell systems*, 3(4):385–394, 2016.

Thi Kim Hue Nguyen, Koen Van den Berge, Monica Chiogna, and Davide Risso. Structure learning for zero-inflated counts with an application to single-cell rna sequencing data. *The Annals of Applied Statistics*, 17(3):2555–2573, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Liang Peng, Haopeng Liu, Yixuan Ye, Cheng Liu, Wenjun Shen, Si Wu, and Hau-San Wong. Refinement contrastive learning of cell–gene associations for unsupervised cell type identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 908–916, 2026.

John S Preisser, Kalyan Das, D Leann Long, and Kimon Divaris. Marginalized zero-inflated negative binomial regression with application to dental caries. *Statistics in medicine*, 35(10):1722–1735, 2016.

Roman A Romanov, Amit Zeisel, Joanne Bakker, Fatima Girach, Arash Hellysaz, Raju Tomer, Alan Alpar, Jan Mulder, Frederic Clotman, Erik Keimpema, et al. Molecular interrogation of hypothalamic organization reveals distinct dopamine neuronal subtypes. *Nature neuroscience*, 20 (2):176–188, 2017.

Nicholas Schaum, Jim Karkanias, Norma F Neff, Andrew P May, Stephen R Quake, Tony Wyss-Coray, Spyros Darmanis, Joshua Batson, Olga Botvinnik, Michelle B Chen, et al. Single-cell transcriptomics of 20 mouse organs creates a tabula muris: The tabula muris consortium. *Nature*, 562(7727):367, 2018.

M Sklar. Fonctions de répartition à n dimensions et leurs marges. In *Annales de l'ISUP*, volume 8, pp. 229–231, 1959.

Pascal Soucy and Guy W Mineau. A simple knn algorithm for text categorization. In *Proceedings 2001 IEEE international conference on data mining*, pp. 647–648. IEEE, 2001.

Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *cell*, 177(7):1888–1902, 2019.

Alexander J Tarashansky, Yuan Xue, Pengyang Li, Stephen R Quake, and Bo Wang. Self-assembling manifolds in single-cell rna sequencing data. *Elife*, 8:e48994, 2019.

Ashutosh Tewari. On the estimation of gaussian mixture copula models. In *International Conference on Machine Learning*, pp. 34090–34104. PMLR, 2023.

Ashutosh Tewari, Michael J Giering, and Arvind Raghunathan. Parametric characterization of multimodal distributions with non-gaussian modes. In *2011 IEEE 11th international conference on data mining workshops*, pp. 286–292. IEEE, 2011.

Shengwen Tian, Yu Wang, Yutian Wang, Cunmei Ji, and Jiancheng Ni. scafgcc: An augmentation-free graph contrastive clustering method for scrna-seq data analysis. *Neurocomputing*, pp. 131698, 2025.

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

Ekaterina Tomilina, Gildas Mazo, and Florence Jaffrézic. Multi-omics network inference with a gaussian copula model. *BMC bioinformatics*, 2026.

Maria Antonietta Tosches, Tracy M. Yamawaki, Robert K. Naumann, Ariel A. Jacobi, Georgi Tushev, and Gilles Laurent. Evolution of pallium, hippocampus, and cortical cell types revealed by single-cell transcriptomics in reptiles. *Science*, 360(6391):881–888, 2018.

Le Van Vinh, Tran Nhat Quang, Lai Hoang Hiep, Pham Nhat Phuong, and Tran Van Hoai. Deep clustering of single-cell rna-seq using adversarial graph contrastive learning. *Briefings in bioinformatics*, 26(4):bbaf423, 2025.

Yalong Wang, Wanlu Song, Jilian Wang, Ting Wang, Xiaochen Xiong, Zhen Qi, Wei Fu, Xuerui Yang, and Ye-Guang Chen. Single-cell transcriptome analysis reveals differential nutrient absorption functions in human intestine. *Journal of Experimental Medicine*, 217(2):e20191130, 2019.

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):15, 2018.

Xiangyu Wu, Xin Yang, Yunhan Dai, Zihan Zhao, Junmeng Zhu, Hongqian Guo, and Rong Yang. Single-cell sequencing to multi-omics: technologies and applications. *Biomarker research*, 12 (1):110, 2024.

Ping Xu, Zhiyuan Ning, Meng Xiao, Guihai Feng, Xin Li, Yuanchun Zhou, and Pengfei Wang. sccdcg: efficient deep structural clustering for single-cell rna-seq via deep cut-informed graph embedding. In *International Conference on Database Systems for Advanced Applications*, pp. 172–187. Springer, 2024.

Kelvin KW Yau, Kui Wang, and Andy H Lee. Zero-inflated negative binomial mixed regression modeling of over-dispersed count data with extra zeros. *Biometrical Journal: journal of mathematical methods in biosciences*, 45(4):437–452, 2003.

Ka Yee Yeung and Walter L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

Matthew D Young, Thomas J Mitchell, Felipe A Vieira Braga, Maxine GB Tran, Benjamin J Stewart, John R Ferdinand, Grace Collord, Rachel A Botting, Dorin-Mirel Popescu, Kevin W Loudon, et al. Single-cell transcriptomes from human kidneys reveal the cellular identity of renal tumors. *science*, 361(6402):594–599, 2018.

Bin Yu, Chen Chen, Ren Qi, Ruiqing Zheng, Patrick J Skillman-Lawrence, Xiaolin Wang, Anjun Ma, and Haiming Gu. scgmai: a gaussian mixture model for clustering single-cell rna-seq data based on deep autoencoder. *Briefings in bioinformatics*, 22(4):bbaa316, 2021.

Zhuohan Yu, Yifu Lu, Yunhe Wang, Fan Tang, Ka-Chun Wong, and Xiangtao Li. Zinb-based graph embedding autoencoder for single-cell rna-seq interpretations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 4671–4679, 2022.

Shichao Zhang, Debo Cheng, Zhenyun Deng, Ming Zong, and Xuelian Deng. A novel knn algorithm with data-driven k parameter computation. *Pattern Recognition Letters*, 109:44–54, 2018.

GXY Zheng, JM Terry, P Belgrader, P Ryvkin, ZW Bent, R Wilson, SB Ziraldo, TD Wheeler, GP McDermott, J Zhu, et al. Massively parallel digital transcriptional profiling of single cells. nat. commun. 8, 14049, 2017.

Cheng Zhong, Siqi Jiang, and Zhi Wei. Integrating feature selection with unsupervised deep embedding for clustering single-cell rna-seq data. *Briefings in Bioinformatics*, 27(2):bbag082, 2026.

APPENDIX- SCCGNET: JOINTLY OPTIMIZING COPULA MIXTURES AND GRAPH AUTOENCODERS FOR SINGLE-CELL CLUSTERING

APPENDIX TABLE OF CONTENTS

A Dataset details **15**

B Importance of end-to-end integration **15**

C Clustering scalability and stability in simulated datasets **15**

D Visualization of the learned representation **16**

E Algorithm **17**

F Evaluation metrics **17**

G Baseline models **19**

H Average performance **20**

I Marker genes of the recovered clusters **20**

J Latent-space analysis of the cluster model **23**

K Implementation details **25**

L Limitations and future work **27**

M LLM Usage **27**

A Dataset details

We use the 15 benchmark datasets described above for the main evaluation. Their detailed characteristics are provided in Table 5.

Table 5: Summary of the 15 scRNA-seq benchmark datasets.

Dataset	Cells	Genes	Cell types	Source
QS-Diaphragm	870	23,341	5	Tabula Muris (Schaum et al., 2018)
QS-Limb Muscle	1,090	23,341	6	Tabula Muris (Schaum et al., 2018)
QS-Trachea	1,350	23,341	4	Tabula Muris (Schaum et al., 2018)
QS-Lung	1,676	23,341	11	Tabula Muris (Schaum et al., 2018)
Muraro	2,122	19,046	9	(Muraro et al., 2016)
Qx-Bladder	2,500	23,341	4	Tabula Muris (Schaum et al., 2018)
Romanov	2,881	21,143	7	(Romanov et al., 2017)
Adam	3,660	23,797	8	Adam et al. (2017)
Qx-LimbMuscle	3,909	23,341	6	Tabula Muris (Schaum et al., 2018)
Young	5,685	33,658	11	Young et al. (2018)
Qx-Spleen	9,552	23,341	5	Tabula Muris (Schaum et al., 2018)
Cao_2020 Spleen	13,180	56,622	8	(Cao et al., 2020)
Wang Large Intestine	14,537	19,525	6	(Wang et al., 2019)
Tosches turtle	18,664	23,500	8	Tosches et al. (2018)
Bach	23,184	19,965	8	Bach et al. (2017)

To assess clustering stability and scalability under controlled conditions, we further generate nine simulated datasets using SPARSim (Baruzzo et al., 2020). We use the Chu (Chu et al., 2016) and Zheng (Zheng et al., 2017) preset templates, which contain 11,782 genes, 758 cells, and 6 cell types, and 15,401 genes, 3,388 cells, and 4 cell types, respectively. For all simulated datasets, we retain the top 1,200 highly variable genes and vary the number of cells from 379 to 50,820. The resulting datasets and their corresponding numbers of clusters are summarized in Table 6.

B Importance of end-to-end integration

In this experiment, we demonstrate the importance of end-to-end integration in comparison to multi-stage alternative in combining statistical GMCM and deep learning VGAE. We also analyze the positive effect in adding ZINB layer. We consider different cases in this ablation study: 1) two-stage integrating GMC and VGAE, using only VGAE loss to optimize VGAE, and the result from VGAE will be passed to GMC for clustering; 2) similar to case (1), but adding ZINB loss in the optimizing of VGAE; 3) end-to-end integrating VGAE and GMC, using both VGAE and GMC losses to optimize the entire algorithm, clustering is conducted simultaneously with optimization; and 4) similar to case (3) but adding ZINB loss in the optimization.

The results in Fig 6 show that end-to-end integrating GMCM and VGAE outperforms its multi-stage alternative. However, adding ZINB loss has a significant in improving the clustering performance, especially for the large-scale dataset.

C Clustering scalability and stability in simulated datasets

The simulation study was designed to assess stability and scalability under known cluster structure and to compare scCGNet with its statistical components, GMM and GMCM. It therefore uses a different baseline set from the real-data benchmark (Appendix G), with scDeepCluster and scAce common to both. We generate nine datasets with the Chu and Zheng presets, ranging from 379 to 50,820 cells (Table 6). The comparison of ARI and NMI among scCGNet with the additional baselines applying on Chu preset simulated data is summarized in Table 7. As we can see from Table 7, when the number of cells is small (smaller or around 1500), although ARI and NMI of all methods are smaller than 0.9, scCGNet still obtain comparble results. As illustrated in Figure 7, scDeepCluster and scAce have good performance in large-scaled datasets with 0.75-0.85 for ARI

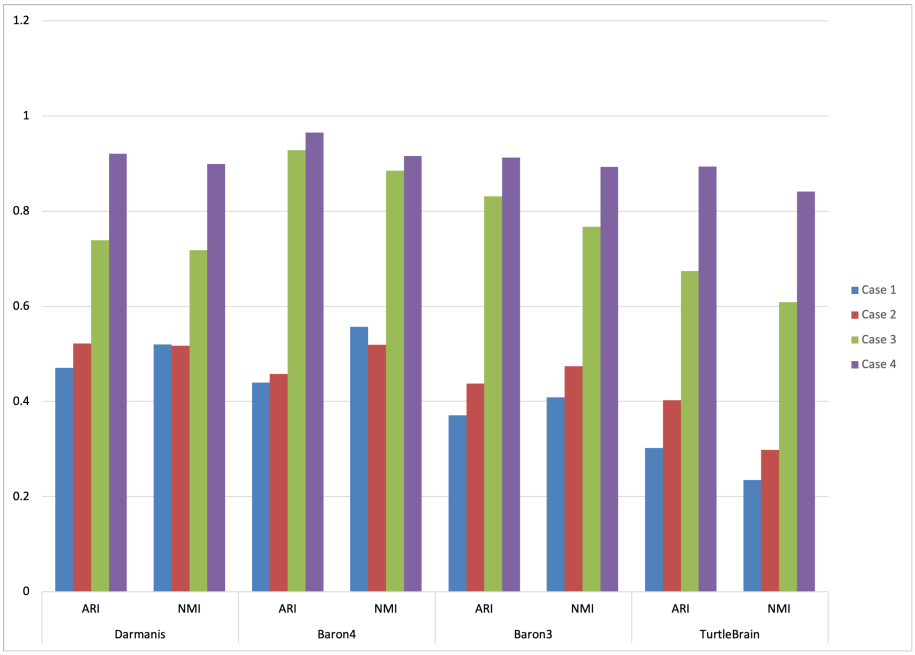


Figure 6: Ablation study of scCGNet integration using different combinations of clustering and reconstruction losses. Results are reported as ARI and NMI on four independent scRNA-seq datasets: Darmanis, Baron4, Baron3, and TurtleBrain. Higher values indicate better agreement with the reference cell-type annotations. Cases 1–4 represent the evaluated loss configurations, with Case 4 corresponding to the complete integration.

Table 6: Summary of simulated scRNA-seq datasets.

Preset	Cells	Genes	HVG	Clusters
Chu's template	379	11,782	1,200	6
	758	11,782	1,200	6
	1,516	11,782	1,200	6
Zheng's template	1,694	15,401	1,200	4
	3,388	15,401	1,200	4
	6,776	15,401	1,200	4
	20,328	15,401	1,200	4
	33,880	15,401	1,200	4
	50,820	15,401	1,200	4

and NMI. However, scCGNet consistently outperforms other baseline methods when the number of cells increases. This demonstrates the stability in clustering performance of scCGNet.

D VISUALIZATION OF THE LEARNED REPRESENTATION

To assess whether scCGNet recovers biologically coherent structure, we project its latent embedding to two dimensions with UMAP (McInnes et al., 2018) (Figure 8). For each dataset the same embedding is coloured by the ground-truth cell type (left) and by scCGNet's predicted cluster (right); agreement between the two panels indicates that the predicted partition recovers the annotated populations. Because cluster indices are assigned independently in the two panels, corresponding populations may appear in different colours; what matters is that their spatial grouping coincides. We notice that scCGNet's embedding resolves the cells into well-separated groups whose predicted colouring closely mirrors the ground-truth colouring: distinct annotated types occupy distinct regions of the embedding, and few cells cross population boundaries. On the datasets with a small number of well-separated populations the correspondence is near-exact, each predicted cluster mapping onto a

Table 7: Clustering accuracy (ARI and NMI) on Chu preset simulated datasets, for increasing numbers of cells n. Best result per column in bold.

	$n = 379$		$n = 758$		$n = 1516$	
Method	ARI	NMI	ARI	NMI	ARI	NMI
GMM	0.30	0.52	0.31	0.54	0.35	0.57
GMCM	0.45	0.66	0.42	0.64	0.52	0.71
Seurat	0.46	0.66	0.53	0.70	0.48	0.70
SAM+Leiden	0.67	0.74	0.69	0.75	0.70	0.77
SAM+K-means	0.74	0.77	0.78	0.80	0.75	0.78
SAM+DBSCAN	0.54	0.73	0.43	0.68	0.43	0.68
DESC	0.73	**0.88**	**0.86**	0.83	0.77	0.78
scDeepCluster	0.52	0.76	0.70	0.81	0.66	0.82
scVI	0.56	0.69	0.59	0.71	0.61	0.72
scAce	0.55	0.71	0.59	0.72	0.59	0.73
scCGNet	**0.80**	0.85	0.79	**0.88**	**0.82**	**0.86**

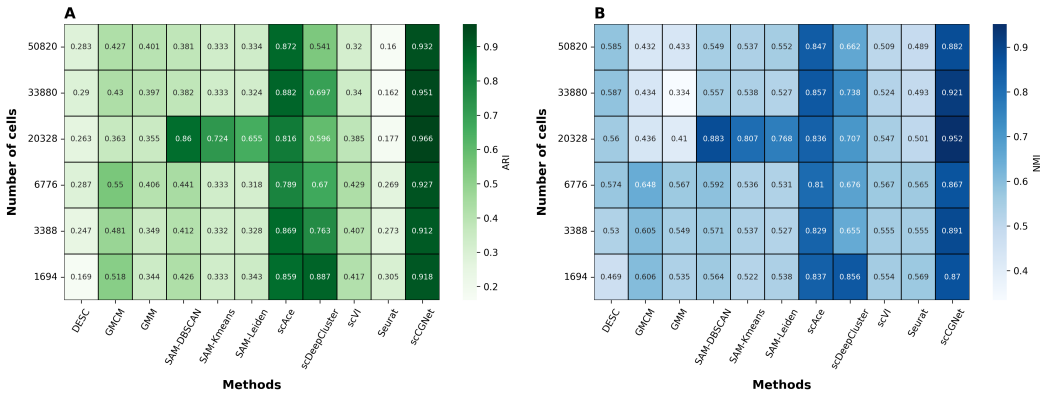


Figure 7: Clustering performance of scCGNet and other baselines on Zheng Preset generated scRNA-seq datasets

single annotated type. On the datasets with more cell types and finer structure, the major populations remain clearly separated, with residual mixing confined to the most transcriptionally similar types.

E Algorithm

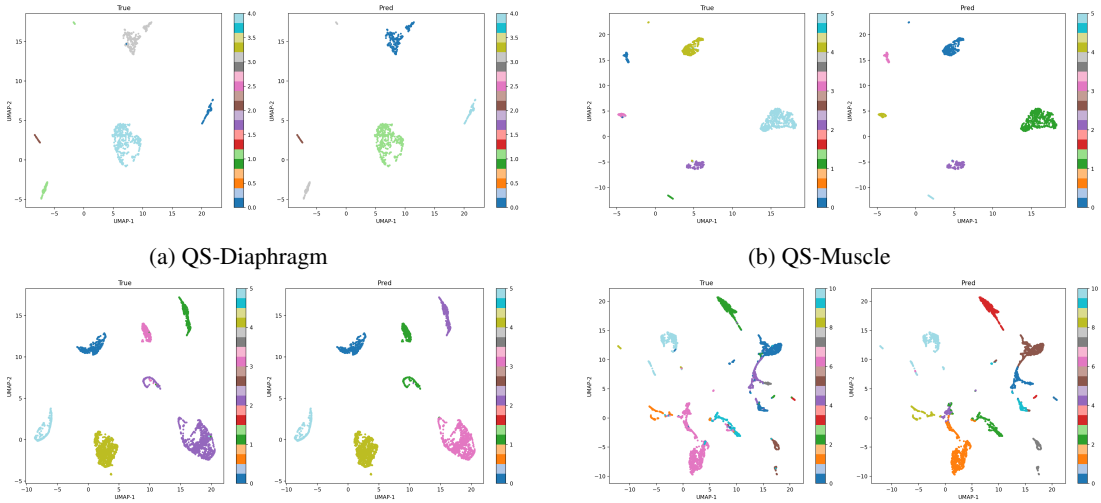
Algorithm 1 summarizes the end-to-end training of scCGNet. The encoder, graph decoder, ZINB decoder, and Gaussian mixture copula parameters are optimized jointly under the single objective of Eq. 12; the copula parameters $\theta = \{\pi_k, \boldsymbol{\eta}_k, \sigma_k^2\}$ are learnable and updated by the same gradient step as the network weights.

F Evaluation metrics

To evaluate the clustering performance of scCGNet in comparison to baseline models, we consider two widely used metrics: adjusted Rand index (ARI) and normalized mutual information (NMI).

Adjusted Rand index

ARI calculates the similarities between ground truth and the predicted labels of any given cluster. The values of ARI near 0 indicates a random clustering and 1 demonstrates a total agreement among clusters (Yeung & Ruzzo, 2001). ARI can be calculated as follows.



(a) QS-Diaphragm (b) QS-Muscle

(c) Qx-Muscle (d) Young

Figure 8: UMAP of scCGNet's latent embedding on six datasets. In each pair, cells are coloured by ground-truth cell type (left) and by scCGNet's predicted cluster (right).

Algorithm 1 scCGNet training

Require: preprocessed counts $\mathbf{X}^{\text{proc}}$, raw counts $\mathbf{X}^{\text{raw}}$, number of clusters K, neighbourhood size k, epochs T

Ensure: cluster labels $\{\hat{k}_i\}_{i=1}^{n}$

build the kNN cell–cell graph $\mathcal{G}$ (adjacency $\mathbf{A}$) from $\mathbf{X}^{\text{proc}}$
pretrain the VGAE encoder–decoder briefly; obtain an initial embedding $\boldsymbol{\mu}$
initialise $\boldsymbol{\eta}_k$ from k-means centroids on $\boldsymbol{\mu}$; set $\pi_k \leftarrow 1/K$, $\log \sigma_{kj}^2 \leftarrow 0$
for $t = 1$ to T **do**
  $\mathbf{H} \leftarrow \text{GCN}_{\text{ReLU}}(\mathbf{X}^{\text{proc}}, \mathbf{A})$; $\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2 \leftarrow \text{GCN}_{\text{Lin}}(\mathbf{H}, \mathbf{A})$
  sample $\mathbf{Z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathcal{L}_{\text{adj}} \leftarrow$ weighted BCE of $\text{Sigmoid}(\mathbf{Z}\mathbf{Z}^\top)$ vs. $\mathbf{A}$ ▷ Eq. 4
  $\mathcal{L}_{\text{KL}} \leftarrow$ KL of $q(\mathbf{Z})$ to $\mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Eq. 5
  $\mathcal{L}_{\text{GMCM}} \leftarrow -\frac{1}{n}\sum_i \left[\log g(\mathbf{z}_i; \theta) - \sum_j \log g_j(z_{ij}; \theta)\right]$ ▷ Eq. 8
  decode ZINB $(\pi, \theta_B, \mu^{(x)})$ from $\mathbf{Z}$; $\mathcal{L}_{\text{ZINB}} \leftarrow$ ZINB NLL vs. $\mathbf{X}^{\text{raw}}$ ▷ Eq. 11
  $\mathcal{L}_{\text{Total}} \leftarrow \mathcal{L}_{\text{adj}} + \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{GMCM}} + \mathcal{L}_{\text{ZINB}}$ ▷ Eq. 12
  update all parameters (network weights and θ) by an Adam step on $\mathcal{L}_{\text{Total}}$
  record ARI; keep the best-ARI checkpoint
end for
at inference, set $\mathbf{z}_i = \boldsymbol{\mu}_i$ and assign $\hat{k}_i = \arg\max_k \gamma_{ik}$ ▷ Eq. 9

$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}\right] - \left[\sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}\right] / \binom{n}{2}}. \tag{13}$$

where n_{ij} represents the number of cells with true label $\mathbf{y}_i$ assigned to cluster of labels $\mathbf{y}_j$. n_i represents the number of cells assigned in cluster of labels $\mathbf{y}_i$, n_j represents the number of cells with true label $\mathbf{y}_j$.

Normalized mutual information

The mutual information score is the measurement of similarity and is used to explore the grouping property (Kraskov & Grassberger, 2009). In contrast, NMI is a normalization of mutual information

where the mutual information score is normalized by a generalized mean of the true labels and predicted labels. NMI can be defined as

$$\mathrm{NMI}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{2 \times \mathrm{MI}(\hat{\mathbf{y}}, \mathbf{y})}{[\mathcal{L}(\hat{\mathbf{y}}) + \mathcal{L}(\mathbf{y})]}, \tag{14}$$

G BASELINE MODELS

To evaluate the performance of scCGNet, we conducted a comprehensive experiment comparing our proposed method against a variety of clustering methods. Two baseline groups are used: eight recent deep clustering methods on the fifteen real scRNA-seq datasets, and a set of statistical, workflow, and deep methods on the simulated datasets. Where a method requires the number of clusters, it is set to the number of annotated cell types, as for scCGNet.

DEEP CLUSTERING BASELINES (REAL DATASETS)

- **scDeepCluster:** A model-based deep embedded clustering method for scRNA-seq data that simultaneously learns the feature representation and performs clustering using an explicit ZINB data-generation model (Tian et al., 2019).
- **scziDesk:** A deep soft K-means clustering method that couples a denoising ZINB autoencoder with a weighted soft K-means objective and a self-training scheme over the learned embedding (Chen et al., 2020).
- **FSSC:** A feature-selection clustering method that incorporates group-Lasso gene selection into a ZINB-autoencoder clustering framework, jointly selecting informative genes and assigning clusters (Zhong et al., 2026).
- **scAce:** An adaptive embedding and clustering method that constructs a modified variational autoencoder to jointly learn the cell embedding and assign clusters. It does not require a predefined number of clusters, as scAce updates and enhances cluster assignments through adaptive cluster merging (He et al., 2023).
- **scMAE:** A masked-autoencoder method that learns representations by reconstructing masked gene-expression values, then clusters the learned embedding with K-means or spectral clustering (Fang et al., 2024).
- **scRCL:** A graph-based method that fuses graph- and feature-based representations through refinement contrastive learning over cell and gene graphs (Peng et al., 2026).
- **scCDCG:** A graph-based method that learns a cut-informed graph representation optimized with an optimal-transport clustering objective (Xu et al., 2024).
- **scAGCL:** A graph contrastive clustering method that strengthens graph-based contrastive learning through adversarial perturbations of the cell–cell graph (Van Vinh et al., 2025).

BASELINES FOR THE SIMULATED DATASETS

For the simulated datasets we compare against scCGNet's statistical bases, GMM (Melnykov & Maitra, 2010) and GMCM (Bilgrau et al., 2016), together with several ML and deep learning methods commonly used for scRNA-seq analysis. GMM and GMCM are included here as an ablation: they quantify what the bare statistical models achieve on their own, relative to the improvement obtained by embedding them in scCGNet's end-to-end framework.

- **Gaussian mixture model (GMM):** GMM is a statistical probabilistic model that represents a mixture of several Gaussian distributions. GMM assumes that the data is generated from a combination of multiple Gaussian distributions (Melnykov & Maitra, 2010).
- **Gaussian mixture copula model (GMCM):** GMCM is a statistical model that combines GMM and copula models to capture the dependency structure between variables in multivariate data. GMCM leverages copulas to account for the dependences between variables while also incorporating the concept of mixture models to handle multiple components or clusters in the data (Bilgrau et al., 2016).

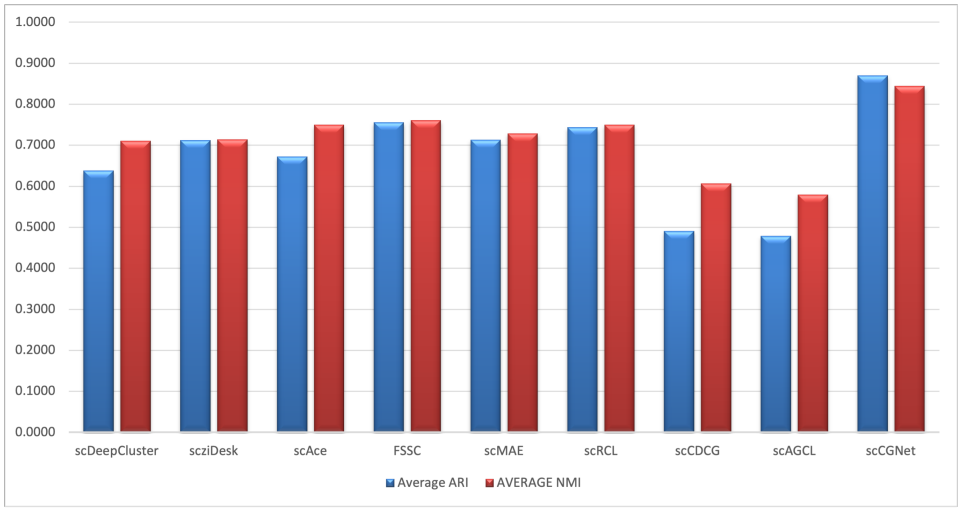


Figure 9: Average clustering performance across the fifteen scRNA-seq datasets. Bars report the mean adjusted Rand index (ARI) and normalized mutual information (NMI) for scCGNet and the eight baselines; scCGNet attains the highest mean on both metrics.

- **Seurat:** This clustering method includes a graph-based clustering analysis tool for scRNA-seq data, which applies a shared nearest neighbor (SNN) modularity optimization-based clustering algorithm to identify the cell clusters (Stuart et al., 2019; Hao et al., 2021; 2024).
- **SAM:** The algorithm is proposed to improve feature selection by assigning weights to genes. The method first creates a kNN graph and refines the graph iteratively by updating the weighting on the genes. The iteration continues until the algorithm converges. SAM includes several built-in clustering methods such as Leiden, Louvain, K-means, DBSCAN, etc. (Tarashansky et al., 2019)
- **DESC:** This is an unsupervised deep learning algorithm that clusters scRNA-seq data. In DESC, a non-linear mapping function is constructed from the original scRNA-seq data space to a low-dimensional feature space by iteratively learning cluster-specific gene expression representation and assigning clusters using a deep neural network (Li et al., 2020).
- **Single-cell variational inference (scVI):** scVI is introduced as a deep neural network which is used to create conditional distributions for a hierarchical Bayesian model. The encoder employs nonlinear transformation to encode cell information. The decoder uses another nonlinear transformation to estimate gene posteriors (Lopez et al., 2018).

H AVERAGE PERFORMANCE

Figure 9 summarizes the clustering results by averaging each method's scores over the fifteen datasets. scCGNet obtains a mean NMI of 0.844, compared with 0.761 for the next-best method, FSSC. The averages complement the per-dataset results in Table 2: on NMI, scCGNet ranks first on 11 of the 15 datasets and second on two.

I MARKER GENES OF THE RECOVERED CLUSTERS

We examined whether the clusters found by scCGNet correspond to the annotated cell types at the level of gene expression. For each dataset, marker genes were identified with the same procedure for the scCGNet clusters and for the ground-truth labels, and the two sets of markers were compared. The analysis is descriptive: it compares two partitions of the same data and does not attribute any result to an individual component of scCGNet.

Procedure. Raw counts were normalized to 10^4 per cell and log-transformed ($\log(1+x)$); genes detected in fewer than three cells were removed. For each cluster, every gene was tested against all

Table 8: Marker-gene summary. *Labels*: number of ground-truth labels; *Clusters*: number of non-empty scCGNet clusters; *3/3* and *≥2/3*: labels matched by a cluster sharing all three, or at least two, of their top-three markers; *Specificity*: mean difference between the detection rate inside and outside the group over the selected markers; *log_2FC*: mean $\log_2$ fold change of the selected markers. GT: ground truth.

Dataset	**Labels**	**Clusters**	**Labels matched**		**Specificity**		**$\log_2$FC**	
			3/3	≥2/3	GT	scCGNet	GT	scCGNet
Young	11	11	5	7	0.652	0.641	6.47	5.36
QS-Diaphragm	5	5	4	5	0.824	0.824	9.50	9.37
Qx-Muscle	6	6	4	6	0.854	0.784	7.04	6.75
QS-Muscle	6	6	3	6	0.827	0.812	10.01	10.20
Total	28		16	24				

remaining cells with a two-sided Wilcoxon rank-sum test (normal approximation with tie correction), and p-values were adjusted with the Benjamini–Hochberg procedure within each cluster. We retained genes with adjusted $p < 0.05$, a $\log_2$ ratio of mean normalized expression above 0.25, and detection in at least 10% of the cells in the cluster. Each gene was assigned only to the cluster with its largest test statistic, and the three genes with the largest statistics were selected per cluster. Because the clusters are estimated from the same expression data, these p-values are not calibrated for inference and are used only to rank and filter genes. A label is counted as matched when a scCGNet cluster shares its top-three markers.

Datasets. We analysed Young, QS-Diaphragm, Qx-Muscle and QS-Muscle. The processed files of Qx-Muscle and QS-Muscle do not contain gene identifiers. For these datasets, markers are identified by column index, which allows the two partitions to be compared but not the markers to be interpreted biologically, and no dot plots are shown. The tables and dot plots of Young and QS-Diaphragm are in the main text (Tables 3 and 4, Figure 5). Labels are reported by their numeric codes.

I.1 Summary across datasets

Table 8 summarizes the results. Of the 28 annotated labels, 16 are matched by a scCGNet cluster with the same three markers, and 24 by a cluster sharing at least two of the three. Averaged over the selected markers, the difference between the detection rate inside and outside a group is within 0.02 of the ground-truth value on Young, QS-Diaphragm and QS-Muscle, and lower by 0.07 on Qx-Muscle.

I.2 Young

Table 3 relates the two partitions using the confusion matrix, and Figure 5 shows the markers. Five of the eleven labels (1, 2, 5, 9 and 10) are matched by a cluster with the same three markers. Cluster 0 contains cells of labels 4, 7 and 3 (311, 117 and 56 cells); its markers are those of label 4, and the markers of labels 3 and 7 are not recovered. Cluster 10 contains cells of labels 10 and 8 (613 and 73 cells); its markers are those of label 10, which are also highly expressed in label 8 (Figure 5d). Label 6 is divided between clusters 1 and 4 (1,163 and 191 cells), and label 0 between clusters 5 and 9 (933 and 230 cells). The markers of cluster 9 (*MTRNR2L8*, *IGKC*, *MTRNR2L12*) are detected in 25–31% of cells outside the cluster, the highest rate among all clusters except cluster 1. Cluster 6 contains seven cells of label 10 and is too small for further interpretation.

I.3 QS-Diaphragm

Each of the five scCGNet clusters matches one label, and the cluster sizes equal the label sizes to within one cell (Table 4, Figure 5). Four labels are matched with the same three markers; for label

Table 9: Datasets without gene identifiers: each scCGNet cluster, its size, the ground-truth label with which it shares the most top-three markers, and the number of shared markers. On Qx-Muscle, cluster 1 also contains cells of label 2, as inferred from marker detection rates.

Dataset	Cluster (cells)	Best-matching label (cells)	Shared
Qx-Muscle	0 (461)	0 (461)	3/3
	2 (322)	1 (320)	3/3
	4 (1,136)	4 (1,136)	3/3
	5 (353)	5 (354)	3/3
	3 (1,135)	2 (1,330)	2/3
	1 (502)	3 (308)	2/3
QS-Muscle	1 (525)	5 (540)	3/3
	4 (49)	3 (45)	3/3
	5 (36)	1 (35)	3/3
	0 (264)	4 (258)	2/3
	2 (145)	2 (141)	2/3
	3 (71)	0 (71)	2/3

0, two markers are shared (*Cdh5*, *Apold1*), and *Gpr116* replaces *Flt1*. The marker *Sdc4*, selected for label 4 and cluster 1, is detected in 82% of cells outside the group and is therefore not specific.

I.4 Datasets without gene identifiers

For the two datasets below, markers are identified by column index. Table 9 lists, for each scCGNet cluster, its size, the label with which it shares the most top-three markers, and the number of shared markers. Group sizes are exact; statements about the label composition of clusters are inferred from the detection rates of each label's markers within the scCGNet clusters.

Qx-Muscle. Clusters 0, 2, 4 and 5 match labels 0, 1, 4 and 5 with the same three markers and sizes within two cells. Cluster 1 (502 cells) shares two markers with label 3 (308 cells), and cluster 3 (1,135 cells) shares two markers with label 2 (1,330 cells). Within cluster 1, the markers of label 3 are detected in 54–62% of cells and those of label 2 in 36–42%, consistent with cluster 1 containing the cells of label 3 together with about 195 cells of label 2 (the difference between the sizes of label 2 and cluster 3).

QS-Muscle. Each of the six clusters matches one label, with sizes within 15 cells. Three labels (1, 3 and 5) are matched with the same three markers and the other three with two of three.

I.5 Internal clustering metrics

To complement the marker analysis, we scored the scCGNet partitions and the ground-truth partitions with three internal metrics in a common expression space: silhouette (higher is better), Calinski–Harabasz index (higher is better) and Davies–Bouldin index (lower is better), all with Euclidean distance. The space was built independently of any clustering model: counts were normalized to 10^4 per cell and log-transformed, the 2,000 genes with the largest variance were retained and z-scored (clipped at ± 10), and the first 50 principal components were used. The Calinski–Harabasz and Davies–Bouldin indices were computed on all cells. The silhouette was computed on all cells for datasets with at most 5,000 cells, and for Young (5,685 cells) it was averaged over five random subsamples of 5,000 cells (standard deviation 0.0010 for the ground truth and 0.0009 for scCGNet).

Table 10 reports the scores. The silhouette of the scCGNet partition differs from that of the ground-truth partition by at most 0.022. On Young and QS-Diaphragm, the scCGNet partition scores better than the ground truth on all three metrics, by small margins. On QS-Muscle, it scores slightly worse

Table 10: Internal clustering metrics of the ground-truth (GT) and scCGNet partitions in a common PCA space of expression (50 components from the 2,000 most variable genes). Sil.: silhouette (↑); CH: Calinski–Harabasz index (↑); DB: Davies–Bouldin index (↓). Both partitions have the same number of clusters on every dataset.

Dataset	**Clusters**	**Sil. (↑)**		**CH (↑)**		**DB (↓)**	
		GT	scCGNet	GT	scCGNet	GT	scCGNet
Young	11	0.130	0.143	464.3	520.3	2.059	2.043
QS-Diaphragm	5	0.303	0.305	180.1	182.0	1.255	1.251
Qx-Muscle	6	0.295	0.273	953.9	896.0	1.312	1.610
QS-Muscle	6	0.281	0.278	187.5	185.2	1.364	1.401

on all three. On Qx-Muscle, it also scores worse on all three, most clearly on the Davies–Bouldin index (1.610 against 1.312). These metrics measure the compactness and separation of clusters in the chosen space; they do not measure agreement with the annotation, and they favour compact, well-separated clusters.

I.6 Limitations

The clusters and the markers are estimated from the same data, so the analysis describes the partitions rather than testing hypotheses about them. The internal metrics depend on the choice of reference space and are reported only for scCGNet and the ground truth, not for other methods. Two of the four datasets lack gene identifiers, so their markers cannot be related to known cell-type markers, and labels are reported as numeric codes. For three datasets, the label composition of clusters is inferred from marker detection rates rather than read from a confusion matrix. On Young, the smallest annotated populations (labels 3, 7 and 8, with 60, 118 and 73 cells) are not recovered as separate clusters but are absorbed into larger ones.

J Latent-space analysis of the cluster model

scCGNet replaces the Gaussian mixture of a GMM with a Gaussian mixture copula model (GMCM). This section examines the cluster model separately from representation learning: we fit a standalone GMCM, a full-covariance GMM, and k-means to fixed embeddings and to synthetic data, and describe the distribution of the embeddings within annotated cell types.

Setup. The standalone GMCM is fitted by pseudo-expectation–maximization on rank-based pseudo-observations (Tewari et al., 2011; Bilgrau et al., 2016): the marginal cumulative distribution functions of the Gaussian mixture are inverted to map the pseudo-observations to the latent scale, and one expectation–maximization step then updates the mixture parameters. It is initialized from a GMM fitted to normal scores. The GMM and the GMCM are each fitted with 5 restarts and k-means with 10, keeping the fit with the highest (pseudo-)likelihood or lowest inertia, and the covariance regularization is 10^{-4} for both mixture models. The number of clusters equals the number of annotated cell types, each latent dimension is standardized, and every fit is repeated with five seeds. This GMCM is a standalone implementation for fixed embeddings; it is not the clustering layer of scCGNet, which is trained jointly with the encoder.

J.1 Cluster models on fixed embeddings

Table 11 reports ARI and NMI on the embeddings of the four datasets used in Appendix I. On QS-Diaphragm, the GMCM obtains ARI 0.990, against 0.989 for the GMM and k-means. On the other three datasets, the GMCM obtains a lower ARI than the GMM: by 0.002 on QS-Muscle, 0.006 on Qx-Muscle and 0.031 on Young. On Young, k-means obtains the highest ARI (0.714), and the

GMCM has the largest variation across seeds (standard deviation 0.031). On these embeddings, the standalone GMCM therefore does not improve on the GMM.

Table 11: Cluster models fitted to fixed embeddings: mean (standard deviation) over five seeds. Standard deviations below 0.0005 are shown as 0.000.

Dataset	ARI			NMI		
	k-means	GMM	GMCM	k-means	GMM	GMCM
QS-Diaphragm	0.989 (0.000)	0.989 (0.000)	0.990 (0.001)	0.977 (0.000)	0.977 (0.000)	0.979 (0.003)
QS-Muscle	0.958 (0.000)	0.958 (0.000)	0.956 (0.000)	0.939 (0.000)	0.939 (0.000)	0.937 (0.000)
Qx-Muscle	0.891 (0.000)	0.895 (0.000)	0.889 (0.000)	0.917 (0.000)	0.919 (0.000)	0.918 (0.000)
Young	0.714 (0.000)	0.702 (0.003)	0.671 (0.031)	0.796 (0.000)	0.802 (0.000)	0.791 (0.005)

J.2 Held-out density of the embeddings

We compared the GMM and the GMCM as density models of the embeddings by five-fold cross-validation repeated with five seeds, with the same number of components as above. For the GMCM, the copula parameters are estimated on the training fold, test cells are mapped to $(0, 1)$ with the training-fold marginal distributions, and the reported quantity is the held-out log-density $\log c\big(F_1(x_1), \ldots, F_d(x_d)\big) + \sum_j \log f_j(x_j)$, where each marginal F_j, f_j is a kernel density estimate mixed with a fitted normal distribution (weight 0.05). Standardization is estimated on the training fold. Table 12 reports the results. The GMM attains a higher held-out log-density than the GMCM in all 25 folds of every dataset. A higher held-out density and a higher agreement with the annotation are different properties, so this result does not by itself determine clustering performance.

Table 12: Held-out log-density per cell of the GMM and the GMCM on fixed embeddings (five-fold cross-validation, five seeds). Gain: GMCM minus GMM, mean (standard deviation) over the 25 folds.

Dataset	GMM	GMCM	Gain	Folds with GMCM higher
QS-Diaphragm	130.27	-29.07	-159.35 (31.29)	0/25
QS-Muscle	51.66	7.85	-43.82 (13.83)	0/25
Qx-Muscle	357.54	62.63	-294.92 (32.86)	0/25
Young	109.83	67.95	-41.88 (1.76)	0/25

J.3 Distribution of the embeddings within cell types

Table 13 describes the standardized embeddings within annotated cell types. Types with fewer than 30 cells or fewer than three cells per latent dimension were not analysed, which leaves 16 of the 28 types. Within the analysed types, the mean absolute skewness of the latent dimensions ranges from 0.59 to 4.62 and the mean absolute excess kurtosis from 1.1 to 78.3, so the within-type marginals depart from normality (Figure 10). The effective rank of the within-type covariance matrices is between 1.4 and 12.7, well below the embedding dimension (32 to 128). A Gaussian copula with kernel-density marginals has a higher held-out log-density than a single Gaussian for 6 of the 16 types, including all three analysed types of QS-Muscle. A cell type need not correspond to a single mixture component, so these statistics describe the embeddings and do not evaluate the fitted GMM or GMCM.

J.4 Synthetic experiment: marginal misspecification

To isolate the effect of non-Gaussian marginals, we generated ten synthetic datasets with five correlated Gaussian clusters (200 cells each, 10 dimensions) and applied a strictly increasing transforma-

Table 13: Within-type statistics of the standardized embeddings, median over the analysed cell types. Skewness and excess kurtosis are averaged over latent dimensions (absolute values). Rank–linear discrepancy: mean $|\mathrm{Spearman} - \mathrm{Pearson}|$ correlation over pairs of latent dimensions. Gain: held-out log-density per dimension of a Gaussian copula with kernel-density marginals minus that of a single Gaussian.

Dataset	Dims	Types analysed	Skewness	Excess kurtosis	Rank–linear	Effective rank	Gain
QS-Diaphragm	64	2 of 5	1.08	8.62	0.113	5.35	−0.91
QS-Muscle	32	3 of 6	1.42	6.15	0.159	6.48	2.62
Qx-Muscle	128	3 of 6	1.35	8.67	0.083	2.78	−1.30
Young	64	8 of 11	0.88	2.11	0.080	3.14	−0.46

tion of increasing strength a to every dimension: $x \mapsto s\,(e^{asx} - 1)/a$ with $s = \pm 1$ per dimension, which makes the marginals skewed, or $x \mapsto \sinh(ax)/a$, which makes them heavy-tailed. The dependence structure within each cluster remains Gaussian, so the experiment tests robustness to marginal misspecification only. Each transformed dataset is standardized before clustering.

Figure 11 shows the results. Without transformation, the three methods obtain a mean ARI of at least 0.994. At the largest strength ($a = 2$), the mean ARI of the GMM falls to 0.882 (exponential) and 0.800 (sinh), and that of k-means to 0.425 and 0.748, whereas the GMCM keeps a mean ARI of 0.999. Because strictly increasing transformations leave the ranks of every dimension unchanged, the GMCM receives identical input at every strength; its labels were identical across strengths in all 20 synthetic datasets.

J.5 Summary and limitations

On synthetic data with non-Gaussian marginals and Gaussian within-cluster dependence, the GMCM is unaffected by the marginal transformation while the GMM and k-means degrade. On the four real embeddings, the within-type marginals are skewed and heavy-tailed, but the standalone GMCM does not obtain a higher ARI than the GMM on three of the four datasets and has a lower held-out log-density on all four. The real-data results therefore do not show an advantage of the copula cluster model applied to a fixed embedding; they do not test the joint training of scCGNet, in which the embedding and the cluster model are optimized together.

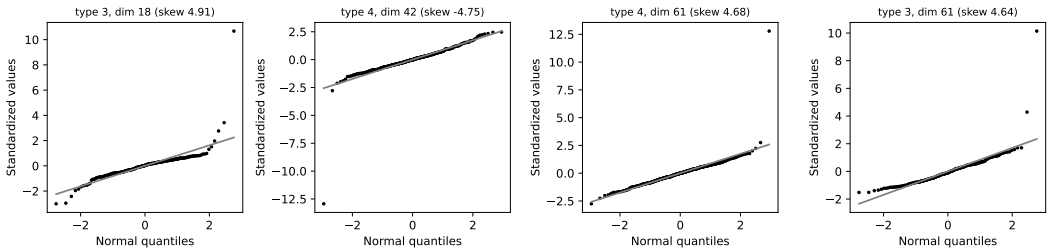
K Implementation details

scCGNet is implemented in PyTorch (Paszke et al., 2019), with graph layers from PyTorch Geometric (Fey & Lenssen, 2019).

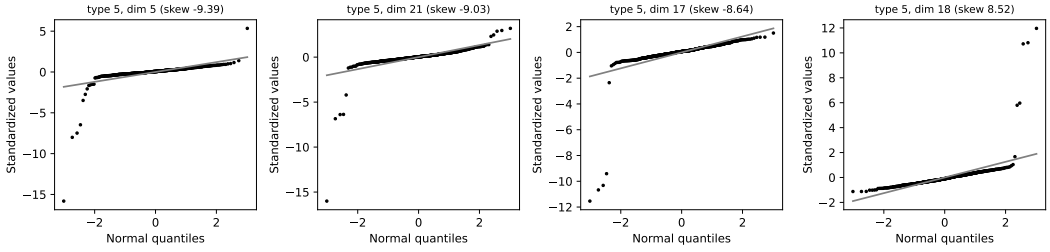
We remove cells with no detected genes and genes detected in no cell. Counts are normalized to the median library size and log-transformed, and the 2,000 most variable genes are selected with Scanpy (Wolf et al., 2018). The encoder input is $\log(1 + x)$ of the raw counts of the selected genes, and the ZINB decoder reconstructs these raw counts. The size factor of each cell is its library size over the selected genes divided by the median of these library sizes.

We build a k-nearest-neighbor graph with $k = 15$ and cosine distance using Scanpy (Wolf et al., 2018) with the Gaussian-kernel method. Distances are computed on the log-normalized selected genes, or on their first 50 principal components for datasets with at least 10,000 cells. All non-zero connectivities are set to 1, which gives a binary, symmetric adjacency matrix.

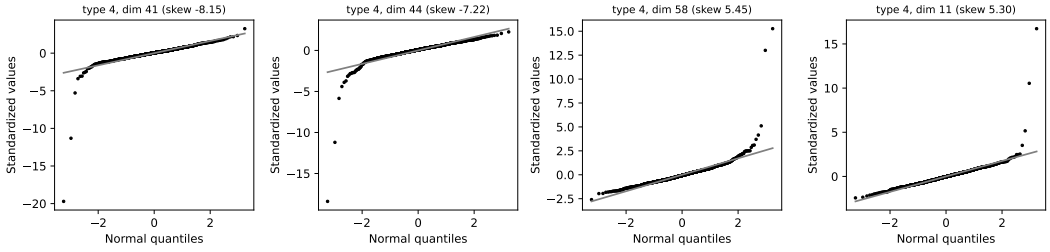
Training is full-batch and uses a single Adam optimizer (Kingma & Ba, 2014). It has three phases. (1) Pretraining: the model is trained on the graph reconstruction, ZINB, and KL losses. (2) Initialization: the component means of the Gaussian mixture copula are set to the centroids of k-means (20 initializations, with k equal to the number of annotated cell types) applied to the latent means of all cells. (3) Joint training: all parameters are trained on the unweighted sum of the graph reconstruction, ZINB, clustering, and KL losses. At inference, each cell is represented by its latent mean and assigned to the component with the highest posterior probability.



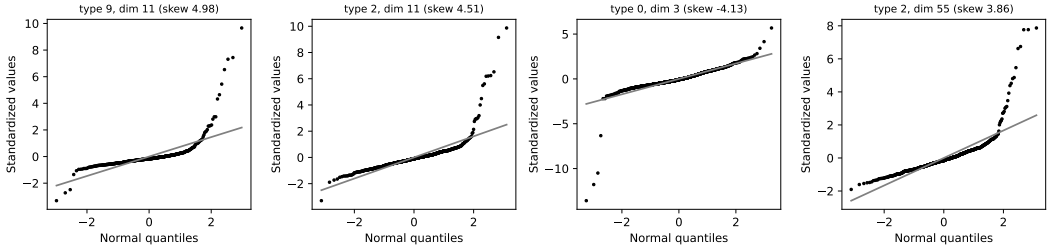
(a) QS-Diaphragm.



(b) QS-Muscle.



(c) Qx-Muscle.



(d) Young.

Figure 10: Normal Q–Q plots of the four most skewed combinations of cell type and latent dimension in each dataset.

Hyperparameters were selected by random search. The selected values cover the following ranges: graph layer type GCN (Kipf & Welling, 2016b) or GraphSAGE (Hamilton et al., 2017); hidden width 64 to 512; latent dimension 32 to 512; learning rate 10^{-4} to 5×10^{-3}; 200 to 500 pretraining epochs; and 300 to 800 joint-training epochs. The model is evaluated every 10 epochs, and we report the evaluated epoch with the highest ARI with respect to the ground-truth labels.

ARI and NMI are computed with scikit-learn (Pedregosa et al., 2011). All random number generators are seeded. UMAP projections (McInnes et al., 2018) use the default parameters of umap-learn with a fixed random state.

Experiments on smaller datasets were run on an Ubuntu server with an NVIDIA GTX 1080 Ti GPU and 64 GB of system memory. Hyperparameter tuning and experiments on the larger datasets were

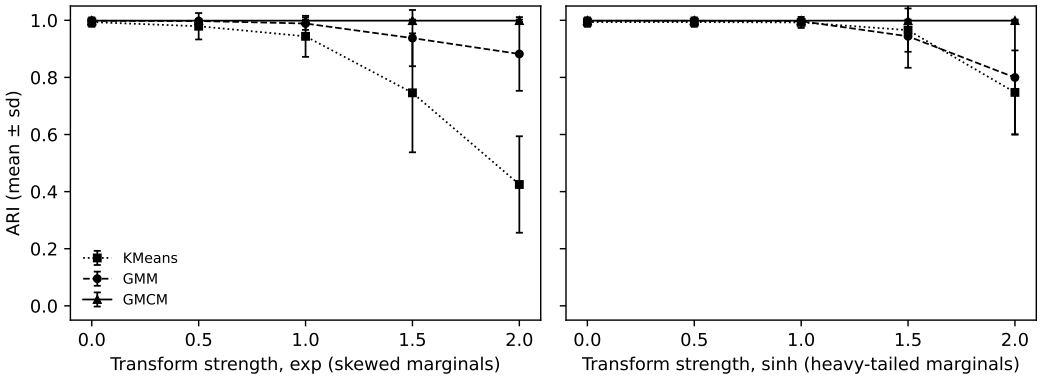


Figure 11: ARI (mean $\pm$ standard deviation over ten synthetic datasets) of k-means, GMM, and GMCM as a function of the strength of a strictly increasing marginal transformation: exponential (skewed marginals, left) and sinh (heavy-tailed marginals, right).

run on a high-performance computing cluster, on a Multi-Instance GPU partition of an NVIDIA H100 (80 GB) with 40 GB of GPU memory.

L Limitations and future work

scCGNet delivers comparable clustering across diverse scRNA-seq datasets. Two limitations nonetheless point to natural extensions. First, scCGNet requires the number of clusters to be specified in advance; inferring it directly from the mixture model would make the framework fully unsupervised. Second, its diagonal-covariance components capture dependence induced by the latent mixture rather than within each cluster; richer covariance structures could model finer intra-cluster dependence, provided training stability is preserved. More broadly, because scCGNet couples a statistical dependence model with a neural architecture under a shared objective, the same principle could extend beyond clustering to tasks such as multi-omic integration and spatial transcriptomics.

M LLM Usage

Large language models were used for editorial and drafting assistance, including wording, structure, and LaTeX formatting of the manuscript. All research ideation, experimental design, implementation, data analysis, and result interpretation were carried out by the authors, who verified every claim, figure, and number against the underlying code and data. The authors take full responsibility for the paper's contents.