Appendix

Table of Contents

A GIANT's selection of datasets

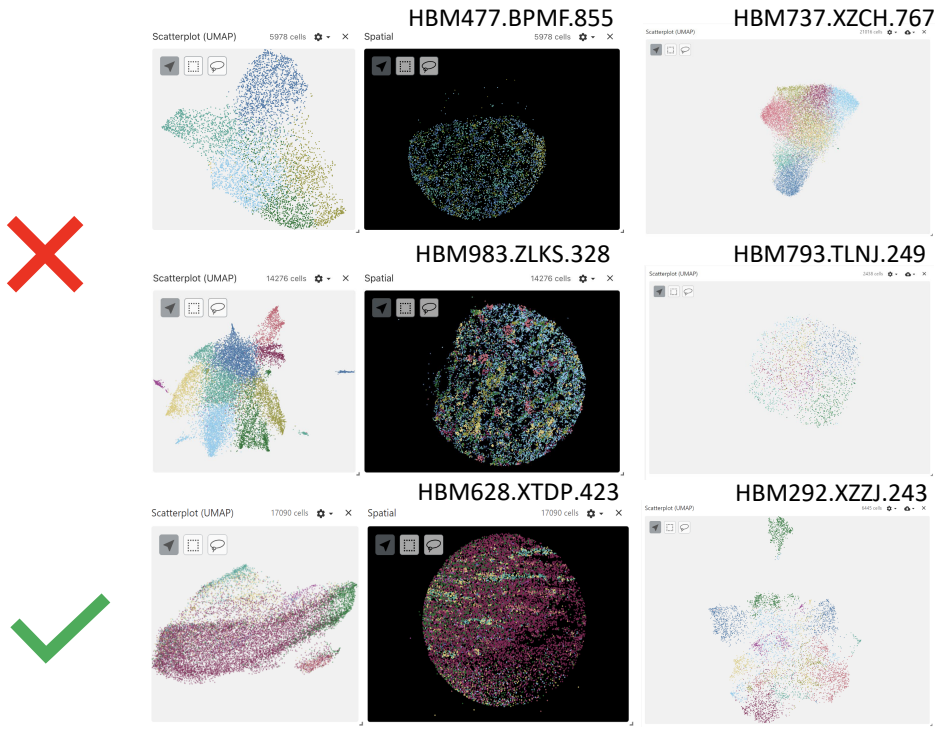

Figure 5: Problematic datasets selected by GIANT.

The quality of datasets, as depicted in Figure 5, impacts the conclusions derived from gene embedding learning methods. GIANT employed low-quality spatial transcriptomic and scATAC-seq data, which ideally should be circumvented at the research inception. The spatial data, HBM477.BPMF.855 and HBM983.ZLKS.328, exhibit considerable barcode expression measurement loss. Under normal circumstances, gene reads distribution in sectioned samples should form complete circles instead of truncated patterns. The scATAC-seq datasets, HBM737.XZCH.767 and HBM793.TLNJ.249, present challenges in discerning differences in gene expression assays across distinct cells, as evidenced by the UMAP visualization results. In an ideal scenario, cells marked with varying colors should occupy different positions in the low-dimensional representation. HBM628.XTDP.423 and HBM292.XZZJ.243 serve as examples of high-quality datasets.

B Constructing co-expression network and training datasets

In this section, we outline the algorithmic details of preprocessing steps and network construction using CS-CORE, scTransform, and SPARK-X. Numerous databases exist for atlas-scale transcriptomic and epigenomic data analysis, including the Human BioMolecular Atlas Program (HuBMAP) [28], the Human Cell Atlas (HCA) [77], the 20 Cellular Senescence Network (SenNet) [53], and more. We gather Single-cell RNA sequencing (scRNA-seq) from HCA datasets, [14] and [76], Single-cell sequencing Assay for Transposase-accessible Chromatin (scATAC-seq) from HuBMAP datasets and [14], and high-quality spatial data from [3, 112]. For distinct omics data, we implement one common and one specific process.

In the common process, we filter barcodes with gene expression counts below 200 and genes with expression counts below three in each barcode. We also filter Mitochondrial (MT) genes. For scRNA-seq datasets, we employ scTransform [35] to select HVGs and generate Pearson residuals, replacing raw expression with residuals. scTransform is the first model to incorporate sequencing depth as a covariate rather than directly applying the size factor to normalization. It eliminates confounding

caused by sequencing depth in raw single-cell or spatial expression data, generating corrected gene expression profiles. These advantages make it a widely used normalization method [11].

To construct scRNA-seq dataset co-expression networks based on Unique Molecular Identifier (UMI), we use CS-CORE [88], a state-of-the-art tool for co-expression inference based on UMI count data. CS-CORE demonstrates increased robustness and a lower false positive rate compared to other tools. For scATAC-seq datasets, we use Seurat [39] to convert the original cells-peaks matrix into the cell-gene activity matrix, incorporating prior information. The cell-gene activity matrix can be processed similarly to the scRNA-seq data matrix, so subsequent preprocessing steps remain the same.

To construct spatial data co-expression networks, we consider spatial expression patterns (SE genes or spatially HVGs) and treat each barcode as a sample. We identify SE genes using SPARK-X [116], then generate corrected gene expression profiles based on scTransform and co-expression networks based on CS-CORE.

B.1 CS-CORE

Since our data are in UMI type, CS-CORE can be used to perform the co-expression network construction. Now considering we have n cells and for one cell i, its expression profile can be denoted as a vector $(x_{i1}, ..., x_{ip})$, where p is the number of genes. We can also use s_i to represent the sequencing depth of cell i. Given the underlying expression levels from cell i with p genes as $(z_{i1}, \ldots, z_{ip})$, the assumption of CS-CORE for the expression profile follows:

$$(z_{i1}, \ldots, z_{ip}) \sim F_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad x_{ij} \mid z_{ij} \sim \text{Poisson}\left(s_i z_{ij}\right) \tag{8}$$

where F_p is an unknown non-negative p-variate distribution with $\boldsymbol{\mu}$ as mean and $\boldsymbol{\Sigma}$ as covariance matrix. Here the observed expression x_{ij} follows the Poisson measurement model depending on the underlying expression z_{ij} and the sequencing depth $s_i = \sum_{j=1}^{n} x_{ij}$. CS-CORE applies a moment-based iteratively reweighted least squares (IRLS) estimation procedure to estimate the covariance matrix. Once we have the covariance matrix $\boldsymbol{\Sigma}_{p\times p} = [\sigma_{ij}]_{i=1\ldots p}^{j=1\ldots p}$, we can use the correlation $\rho_{jj'} = \frac{\sigma_{jj'}}{\sqrt{\sigma_{jj}\sigma_{j'j'}}}$ to estimate the co-expression relation between gene j and gene j'.

With such an assumption, CS-CORE can model the relationship between gene j and gene j' based on a statistical test:

$$T_{jj'} = \frac{\sum_i s_i^2 \left(x_{ij} - s_i\mu_j\right)\left(x_{ij'} - s_i\mu_{j'}\right) g_{ijj'}}{\sqrt{\sum_i s_i^4 \left(s_i\mu_j + s_i^2\sigma_{jj}\right)\left(s_i\mu_{j'} + s_i^2\sigma_{j'j'}\right) g_{ijj'}^2}} \tag{9}$$

The statistic $T_{jj'}$ under the null hypothesis assume gene j and gene j' are independent, which means there is no edge between these two genes. We chose to use p-value ($p < 0.005$) to construct the edges in the co-expression network.

B.2 scTransform

To remove the confounding effect of sequencing depth from the expression level, we first process the single-cell data using scTransform to get the Pearson residuals and then use the Pearson residuals as the initial embeddings of different genes. By assuming that the UMI count data follow the negative binomial distribution, for a given gene g in cell c, we have:

$$\begin{aligned} x_{gc} &\sim \text{NB}\left(\mu_{gc}, \theta_g\right) \\ \ln \mu_{gc} &= \beta_{g0} + \ln s_c, \end{aligned} \tag{10}$$

scTransform regularizes θ as a function of gene means μ, by utilizing the Generalized Linear Model (GLM) with a log link function provided by the above equation. Furthermore, we can estimate the unknown parameters and calculate the Pearson residuals Z_{gc} based on:

$$\begin{aligned} Z_{gc} &= \frac{x_{gc} - \mu_{gc}}{\sigma_{gc}} \\ \mu_{gc} &= \exp(\beta_{g0} + \ln s_c) \\ \sigma_{gc} &= \sqrt{\mu_{gc} + \frac{\mu_{gc}^2}{\theta_{gc}}} \end{aligned} \tag{11}$$

We can finally replace the original expression matrix with the residual matrix Z generated by GLM, and store the expression matrix and the corresponding graph in Scanpy files.

B.3 SPARK-X

The primary distinctions between single-cell transcriptomic data and spatial transcriptomic data involve two aspects: 1. In most spatially resolved data, each barcode represents a mixture of different cells. 2. The additional spatial information introduces spatial gene expression patterns (SE genes) for spatially resolved data. To identify SE genes in spatial transcriptomic data, SPARK-X employs a statistical test comparing the distance covariance matrix, which is constructed using barcode positions, and the expression covariance matrix, which is built from the gene expression profiles.

More specifically, for a spatial transcriptomic gene expression matrix with size $n \times d$, we can donate the matrix for coordinates of samples as $S = (s_1^T, ..., s_n^T), s_i = (s_{i1}, s_{i2})$. Therefore, the whole expression matrix can be represented as : $y = (y_1(s_1), ..., y_n(s_n))^T$. Our target is to test whether y is independent from S, so we construct the expression covariance matrix based on $E = y(y^T y)^{-1} y^T$, and the distance covariance matrix based on $\Sigma = S(S^T S)^{-1} S^T$. For the distance covariance matrix S, SPARK-X considers different kernels to describe different spatial expression patterns, including 1. Gaussian kernel $(s'_{i1}, s'_{i2}) = (exp(\frac{-s_{i1}^2}{2\sigma_1^2}), exp(\frac{-s_{i2}^2}{2\sigma_2^2}))$ and 2. Cosine kernel $(s'_{i1}, s'_{i2}) = (cos(\frac{2\pi s_{i1}}{\Phi_1}), cos(\frac{2\pi s_{i2}}{\Phi_2}))$. After centering these two matrices, we have E_C and Σ_C. Therefore, the test statistic is:

$$T = \frac{\text{trace}(E_C \Sigma_C)}{n} \tag{12}$$

This statistic follows a χ_1^2 distribution and we record the p-value for each gene. The null hypothesis is that the gene expression is irrelevant to the position of barcodes. After finding the p-value list, we rank the p-value in ascending and select the top 1000 genes to perform normalization and co-expression graph construction.

The whole process of the graph construction is shown in Figure 6.

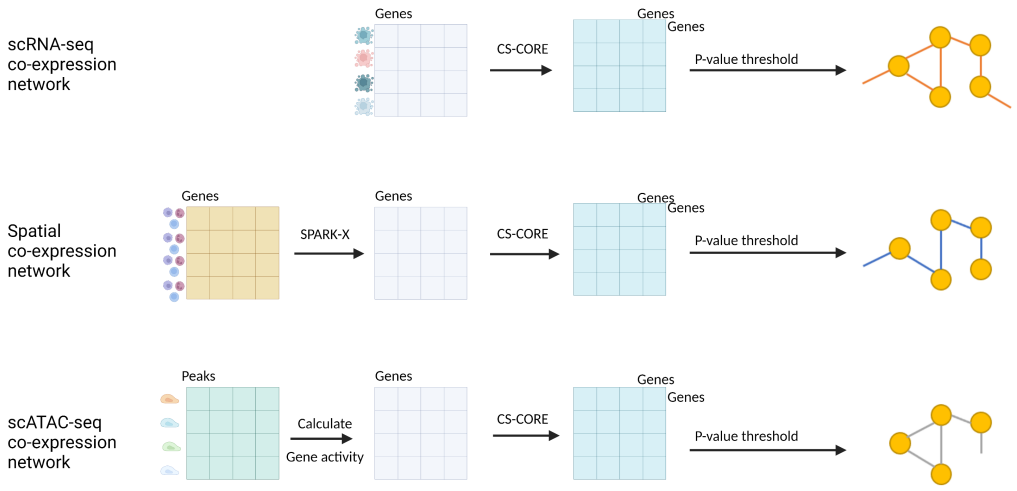

Figure 6: Schematic diagram of the graph data construction methods.

B.4 Transformconv

If we consider c as the index of attention heads, the multi-head attention for GNN from node j to node i can be expressed as:

$$\begin{aligned} q_{c,i}^{(l)} &= W_{c,q}^{(l)} h_i^{(l)} + b_{c,q}^{(l)}; k_{c,j}^{(l)} = W_{c,k}^{(l)} h_j^{(l)} + b_{c,k}^{(l)}, \\ e_{c,ij} &= W_{c,e} e_{ij} + b_{c,e}; \alpha_{c,ij}^{(l)} = \frac{\left\langle q_{c,i}^{(l)}, k_{c,j}^{(l)} + e_{c,ij} \right\rangle}{\sum_{u \in \mathcal{N}(i)} \left\langle q_{c,i}^{(l)}, k_{c,u}^{(l)} + e_{c,iu} \right\rangle}, \end{aligned} \tag{13}$$

where $< q, k >= e^{\frac{q^T k}{\sqrt{d}}}$. $\sqrt{d}$ is a scalar used to reduce gradient vanishing [95] as introduced in the original Transformer paper. The different vectors q, k correspond to the query vector and the key vector, while e represents the edge features. The attention $\alpha_{c,ij}^{(l)}$ denotes the c_{th} attention value from node j to i for layer l. h represents the node embedding, $W_{c,q}^{(l)}, W_{c,k}^{(l)}, W_{c,e}$ are weight matrices, and $b_{c,q}^{(l)}, b_{c,k}^{(l)}, b_{c,e}$ are bias terms.

We define the embedding of node i in layer $l+1$ as h_i^{l+1} and update the node embedding by:

$$\begin{aligned} v_{c,j}^{(l)} &= W_{c,v}^{(l)} h_j^{(l)} + b_{c,v}^{(l)}, \\ h_i^{(l+1)} &= \|_{c=1}^{C} \left[\sum_{j \in \mathcal{N}(i)} \alpha_{c,ij}^{(l)} \left(v_{c,j}^{(l)} + e_{c,ij} \right) \right], \end{aligned} \tag{14}$$

where the vector v represents the value vector. The operation $\|_{c=1}^{C}$ denotes the concatenation of C total attention heads. $\mathcal{N}(i)$ represents the neighbors of node i. Additionally, we construct the residual link [40] based on h_i^{l-1} for h_i^{l+1}. The activation function employed to connect different layers is Mish (Mish$(x) = \tanh(\ln(1 + \exp(x)))$) [65]. We also incorporated a GraphNorm layer [13] to the connection path between hidden layers to enhance convergence.

C Differential co-expression analysis

In this section, we prove that using CS-CORE, we could obtain the co-expression relationships from multimodal biological data without the negative impact of confounding factors brought by the batch effect.

Theorem C.1 *Given the true observed gene expression level x and gene expression affected by batch effect x', the construction of the co-expression network is the same based on the statistical test.*

Proof. We consider the point $x'_{ij} = ax_{ij} + b$, where a, b represent the effect caused by batch effect, and x_{ij} represents the true biological data. Moreover, based on the probabilistic model provided by CS-CORE, we have $\mathbb{E}(x_{ij}) = s_i \mu_j, Var(x_{ij}) = s_i \mu_j + s_i^2 \sigma_{jj}$. Therefore, for the two points x_{ij} and x'_{ij} we have:

$$\begin{aligned} & x_{ij} = s_i \mu_j + \epsilon_{ij} \\ & \left(x_{ij} - s_i \mu_j\right)^2 = s_i \mu_j + s_i^2 \sigma_{jj} + \eta_{ij} \\ & \left(x_{ij} - s_i \mu_j\right)\left(x_{ij'} - s_i \mu_{j'}\right) = s_i^2 \sigma_{jj'} + \xi_{ijj'} \end{aligned} \tag{15}$$

and

$$\begin{aligned} & x'_{ij} = a s_i \mu_j + b + a\epsilon_{ij} \\ & \left(x'_{ij} - (a s_i \mu_j + b)\right)^2 = a^2 s_i \mu_i + s_i^2 a^2 \sigma_{jj} + \eta_{ij} \\ & \left(x'_{ij} - (a s_i \mu_j + b)\right)\left(x'_{ij'} - (a s_i \mu_{j'} + b)\right) = s_i^2 a^2 \sigma_{jj'} + \xi_{ijj'} \end{aligned} \tag{16}$$

Based on this assumption, we can calculate the test statistic $T'_{jj'}$ with batch effect, which is:

$$\begin{aligned} T'_{jj'} &= \frac{\sum_i s_i^2 \left(x'_{ij} - as_i\mu_j - b\right)\left(x'_{ij'} - as_i\mu_{j'} - b\right) g_{ijj'}}{\sqrt{\sum_i s_i^4 \left(a^2 s_i\mu_j + a^2 s_i^2\sigma_{jj}\right)\left(a^2 s_i\mu_{j'} + a^2 s_i^2\sigma_{j'j'}\right) g_{ijj'}^2}} \\ &= \frac{a^2 \sum_i s_i^2 \left(x_{ij} - s_i\mu_j\right)\left(x_{ij'} - s_i\mu_{j'}\right) g_{ijj'}}{a^2\sqrt{\sum_i s_i^4 \left(s_i\mu_j + s_i^2\sigma_{jj}\right)\left(s_i\mu_{j'} + s_i^2\sigma_{j'j'}\right) g_{ijj'}^2}} \\ &= \frac{\sum_i s_i^2 \left(x_{ij} - s_i\mu_j\right)\left(x_{ij'} - s_i\mu_{j'}\right) g_{ijj'}}{\sqrt{\sum_i s_i^4 \left(s_i\mu_j + s_i^2\sigma_{jj}\right)\left(s_i\mu_{j'} + s_i^2\sigma_{j'j'}\right) g_{ijj'}^2}} \\ &= T_{jj'} \end{aligned} \tag{17}$$

This statistic is irrelevant to a, b. Therefore, the batch effect cannot affect our co-expression calculation, and our correlation defined by CS-CORE can reflect the correlation for true biological content.

D Theory analysis for multimodal machine learning

Here we present a series of explanation for why using more modalities is more suitable in our task based on arguing that MMML method can generate more accurate estimate of the latent space representation for genes as the number of modalities increases.

Here we consider our datasets set $\mathcal{D}$ and different types of modalities $\mathcal{M}$ and $\mathcal{N}$, and their corresponding gene embeddings as $e_{\mathcal{M}}$ and $e_{\mathcal{N}}$, where $\textsc{Card}(\mathcal{N}) < \textsc{Card}(\mathcal{M})$. We also have the true representation e^*. Based on our model, we can estimate $\hat{e}_{\mathcal{M}}, \hat{e}_{\mathcal{N}}$. Here we intend to prove the theorem:

Theorem D.1 *$\hat{e}_{\mathcal{M}}$ is a better estimation for e^* comparing to $\hat{e}_{\mathcal{N}}$ with $\textsc{Card}(\mathcal{N}) < \textsc{Card}(\mathcal{M})$.*

Here we consider dataset $\mathcal{S} = \{(X_i, Y_i)\}_{i=1}^m$ as a general training dataset. In our specific cases, we can treat X_i carries information with graph structure and Y_i contains the edges information of X_i. Therefore, we have formalized the dataset we used in our unsupervised learning task. Our target is to minimize the empirical risk:

$$\begin{aligned} \min \quad & \hat{r}\left(h \circ g_{\mathcal{M}}\right) \triangleq \frac{1}{m}\sum_{i=1}^{m} \ell\left(h \circ g_{\mathcal{M}}\left(\mathbf{X}_i\right), Y_i\right) \\ \text{s.t.} \quad & h \in \mathcal{H}, g_{\mathcal{M}} \in \mathcal{G}_{\mathcal{M}} \end{aligned}, \tag{18}$$

where $l(\cdot,\cdot)$ represents the loss function, $h \circ g_{\mathcal{M}}\left(\mathbf{X}_i\right) = h(g_{\mathcal{M}}(\mathbf{X}_i))$ represents the composite function of h and $g_{\mathcal{M}}$. Here function g maps data from X_i to the latent space, while function h maps data from latent space to the space of Y_i. We have $\mathcal{H}$ as the function class of h and $G_{\mathcal{M}}$ is the function class of $g_{\mathcal{M}}$. We define the true map functions as h^* and g^*. We donate $r(\cdot,\cdot)$ as risk.

To finish our proof, we need three assumptions.

Assumption D.1. *The loss function is L-smooth with respect to the first ordinate, and is bounded by a constant C [67, 92, 93].*

In our case, we have three components. BCELoss and Cosine Similarity follow this property [105, 47]. InfoNCE loss also follows this property proved by [61].

Assumption D.2. The true latent representation $g^\star$ is contained in $\mathcal{G}$, and the task mapping $h^\star$ is contained in $\mathcal{H}$ [25, 92, 93].

Assumption D.3. For any $g' \in \mathcal{G}'$ and $\mathcal{M} \subset [K], g' \circ p'_{\mathcal{M}} \in \mathcal{G}'$ [45].

Here the set $[K]$ represents modality set with K modalities. $p'_{\mathcal{M}}$ is a diagonal matrix using 1 for ii-th entry $\in \mathcal{M}$ and 0 otherwise.

We also need one metric to evaluate the quality of our generated embeddings, so here we introduce our definition of *latent representation quality*.

Definition D.1. For any latent space generated by g, the **latent representation quality** is defined as [45]

$$\eta(g) = \inf_{h \in \mathcal{H}} \left[r(h \circ g) - r\left(h^* \circ g^*\right) \right] \tag{19}$$

One approach we can use to prove **Theorem D.1** is to explore the order relationship between the upper bound of $\eta(g_{\mathcal{M}})$ and the upper bound of $\eta(g_{\mathcal{N}})$. That is, we intend to prove:

$$\sup_{g_{\mathcal{M}} \in G_{\mathcal{M}}} \eta(g_{\mathcal{M}}) \leq \sup_{g_{\mathcal{N}} \in G_{\mathcal{N}}} \eta(g_{\mathcal{N}})$$

To prove the above relation, we rely on a new theorem to compute the upper bound of $\eta(\cdot)$.

Theorem D.2. [45] Let $\mathcal{S} = \{(\mathbf{X}_i, Y_i)\}_{i=1}^m$ be a dataset of m examples drawn i.i.d. according to $\mathcal{D}$. Let $\mathcal{M}$ be a subset of $[K]$. Assuming we have produced the empirical risk minimizers $\left(\hat{h}_{\mathcal{M}}, \hat{g}_{\mathcal{M}}\right)$ training with the $\mathcal{M}$ modalities. Then, for all $1 > \delta > 0$ with probability at least $1 - \delta$:

$$\eta\left(\hat{g}_{\mathcal{M}}\right) \leq 4L\Re_m\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{M}}\right) + 4L\Re_m(\mathcal{H} \circ \mathcal{G}) + 6C\sqrt{\frac{2\ln(2/\delta)}{m}} + \hat{L}\left(\hat{h}_{\mathcal{M}} \circ \hat{g}_{\mathcal{M}}, \mathcal{S}\right),$$

where $\hat{L}\left(\hat{h}_{\mathcal{M}} \circ \hat{g}_{\mathcal{M}}, \mathcal{S}\right) \triangleq \hat{r}\left(\hat{h}_{\mathcal{M}} \circ \hat{g}_{\mathcal{M}}\right) - \hat{r}\left(h^\star \circ g^\star\right)$ is the centered empirical loss and L represents the smoothness coefficient.

Remark. Now we consider sets $\mathcal{N} \subset \mathcal{M} \subset [K]$, and based on Assumption D.3, we have $\mathcal{G}_{\mathcal{N}} \subset \mathcal{G}_{\mathcal{M}} \subset \mathcal{G}$. We can interpret this relation as the size of parameter space PARAM follows the relation: $\text{PARAM}_{\mathcal{N}} \subset \text{PARAM}_{\mathcal{M}} \subset \text{PARAM}_{\mathcal{G}}$. Therefore, larger function class has a smaller empirical risk, so we have

$$\hat{L}\left(\hat{h}_{\mathcal{M}} \circ \hat{g}_{\mathcal{M}}, \mathcal{S}\right) \leq \hat{L}\left(\hat{h}_{\mathcal{N}} \circ \hat{g}_{\mathcal{N}}, \mathcal{S}\right) \tag{20}$$

Based on Theorem D.2, we can derive the upper bound of $\eta(\hat{g}_{\mathcal{N}})$ as:

$$\eta\left(\hat{g}_{\mathcal{N}}\right) \leq 4L\Re_m\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{N}}\right) + 4L\Re_m(\mathcal{H} \circ \mathcal{G}) + 6C\sqrt{\frac{2\ln(2/\delta)}{m}} + \hat{L}\left(\hat{h}_{\mathcal{N}} \circ \hat{g}_{\mathcal{N}}, \mathcal{S}\right),$$

where $\Re_m(\cdot)$ represents the Rademacher complexity [10]. Here we can use the Right-hand Side (RHS) to approximate term $4L\Re_m\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{M}}\right) \sim \sqrt{C\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{M}}\right)/m}$ and term $4L\Re_m\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{N}}\right) \sim \sqrt{C\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{N}}\right)/m}$. Based on the basic structural property of Rademacher complexity [10], we have

$$C\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{N}}\right) \leq C\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{M}}\right)$$

Therefore, we can compute:

$$\begin{aligned}
\eta\left(\hat{g}_{\mathcal{M}}\right) - \eta\left(\hat{g}_{\mathcal{N}}\right) &= 4L\Re_m\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{M}}\right) + \hat{L}\left(\hat{h}_{\mathcal{M}} \circ \hat{g}_{\mathcal{M}}, \mathcal{S}\right) - 4L\Re_m\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{N}}\right) - \hat{L}\left(\hat{h}_{\mathcal{N}} \circ \hat{g}_{\mathcal{N}}, \mathcal{S}\right) \\
&= \sqrt{\frac{C\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{M}}\right)}{m}} - \sqrt{\frac{C\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{N}}\right)}{m}} + \hat{L}\left(\hat{h}_{\mathcal{M}} \circ \hat{g}_{\mathcal{M}}, \mathcal{S}\right) - \hat{L}\left(\hat{h}_{\mathcal{N}} \circ \hat{g}_{\mathcal{N}}, \mathcal{S}\right) \\
&= \frac{\sqrt{C\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{M}}\right)} - \sqrt{C\left(\mathcal{H} \circ \mathcal{G}_{\mathcal{N}}\right)}}{\sqrt{m}} + \hat{L}\left(\hat{h}_{\mathcal{M}} \circ \hat{g}_{\mathcal{M}}, \mathcal{S}\right) - \hat{L}\left(\hat{h}_{\mathcal{N}} \circ \hat{g}_{\mathcal{N}}, \mathcal{S}\right) \\
&\to \hat{L}\left(\hat{h}_{\mathcal{M}} \circ \hat{g}_{\mathcal{M}}, \mathcal{S}\right) - \hat{L}\left(\hat{h}_{\mathcal{N}} \circ \hat{g}_{\mathcal{N}}, \mathcal{S}\right) |_{m \to \infty} \\
&\leq 0
\end{aligned}$$

Remark. Therefore, as the number of modalities increases, the representation quality of more modalities will be better than the representation quality of fewer modalities. Hence we proved Theorem D.1 under large number of modalities cases. In the real biological application, genes can be active in many different scenarios, including different cells, different tissues and even different species. Therefore, our proof fits the context of MuSe-GNN's real application.

E Extra experiments, ablation test and model selection

E.1 Ablation tests

We further investigate the contributions of various loss functions and GNN models to MuSe-GNN's performance. Our final loss function comprises three components: the graph reconstruction loss $\mathcal{L}_{\text{reconst}}$, the similarity maximization loss $\mathcal{L}_{\text{CosSim}}$, and the contrastive learning loss $\mathcal{L}_{\text{InfoNCE}}$. The reconstruction loss serves as the basic loss, and we assess our model's performance with different loss function components across all scRNA-seq datasets. The results are presented in Table 3 and 4.

Table 3: The overall benchmarking average rank for ablation tests based on different tissues.

$\mathcal{L}_{\text{CosSim}}$	$\mathcal{L}_{\text{InfoNCE}}$	Heart	Lung	Liver	Kidney	Thymus	Spleen	Pancreas	Cerebrum	Cerebellum	PBMC	Avg Rank
		2.83	3.50	3.17	3.33	2.67	3.17	3.00	3.67	3.00	**3.33**	3.17
✓		2.17	**1.67**	2.00	2.17	2.50	2.00	**1.50**	2.00	2.17	5.67	2.38
	✓	3.00	3.00	3.17	2.83	3.17	3.17	3.33	3.00	2.83	5.00	3.25
✓	✓	**2.00**	1.83	**1.67**	**1.67**	**1.67**	**1.67**	2.17	**1.33**	**2.00**	3.83	**1.98**

Table 4: The overall benchmarking average score for ablation tests based on different tissues.

$\mathcal{L}_{\text{CosSim}}$	$\mathcal{L}_{\text{InfoNCE}}$	Heart	Lung	Liver	Kidney	Thymus	Spleen	Pancreas	Cerebrum	Cerebellum	PBMC	Avg Score
		0.42	0.13	0.21	0.24	0.36	0.19	0.28	0.09	0.37	0.23	0.25
✓		0.65	**0.81**	0.77	0.75	0.57	0.70	**0.83**	0.71	0.68	0.66	0.71
	✓	0.34	0.27	0.25	0.23	0.23	0.20	0.18	0.26	0.35	0.23	0.25
✓	✓	**0.78**	**0.81**	**0.94**	**0.80**	**0.70**	**0.96**	0.53	**0.91**	**0.74**	**0.90**	**0.81**

Based on the results of this ablation test, we conclude that combining all three components to construct our loss function is the optimal choice, achieving a performance that is 224.0% higher than the version without any additional regularization. Furthermore, using only contrastive learning may reduce the performance of our model, implying that learning similarity is more crucial than learning differences for the gene embedding generation task. When both components are included, the final average rank is 14.1% higher than the version with only the similarity learning part. Thus, we need to incorporate both weighted similarity learning and contrastive learning in our final design to learn unified gene embeddings with improved performance.

Additionally, we compare the TransformConv framework with other GNN models, including GCN, Graph Attention Network (GAT) [96], SUGRL (GCN+MLP+Contrastive Learning) [66], GPS [75], GRACE (GCN+Contrastive Learning) [118], GraphSAGE [36], Graph Isomorphism Network (GIN) [106], GraphMAE [43] and Graphormer [110]. We set the number of epochs to 1000 for this comparison. The results are displayed in Table 5.

Table 5: The overall benchmarking score and rank for model selection.

Methods	ASW	AUC	iLISI	GC	CGR	NO	Avg Rank	Avg Score
GCN	0.74±0.02	0.82±0.01	0.32±0.02	0.44±0.02	0.32±0.04	0.31±0.01	3.5	0.51
GAT	0.74±0.09	**0.83±0.01**	0.14±0.03	0.42±0.03	0.00±0.00	0.15±0.03	5	0.17
SUGRL	**0.88±0.01**	0.62±0.00	0.45±0.01	0.43±0.03	0.38±0.02	0.29±0.00	3.5	0.56
GPS	0.77±0.02	0.63±0.00	0.50±0.01	**0.72±0.03**	0.57±0.02	0.30±0.01	2.67	0.68
GRACE	0.82±0.02	0.81±0.00	0.35±0.01	0.42±0.04	0.11±0.01	0.25±0.01	4.17	0.48
GraphSAGE	OOM	OOM	OOM	OOM	OOM	OOM	7.00	0
GIN	OOM	OOM	OOM	OOM	OOM	OOM	7.00	0
GraphMAE	OOM	OOM	OOM	OOM	OOM	OOM	7.00	0
Graphormer	OOM	OOM	OOM	OOM	OOM	OOM	7.00	0
Transformconv	0.75±0.01	0.78±0.02	**0.51±0.02**	0.68±0.04	**0.61±0.02**	**0.31±0.00**	**2.17**	**0.80**

In this table, OOM means out of memory. We can conclude that using TransformConv as the basic graph neural network framework is the best choice, which is 17.6% higher than the version based on GPS (the top2 model). Moreover, based on Figure 7, we can also discover that methods based on GCN or GAT failed to learn the gene function similarity across different datasets. Therefore, choosing TransformConv is reasonable, and the major contribution of Transformer model towards gene embedding learning task is the multi-head attention design.

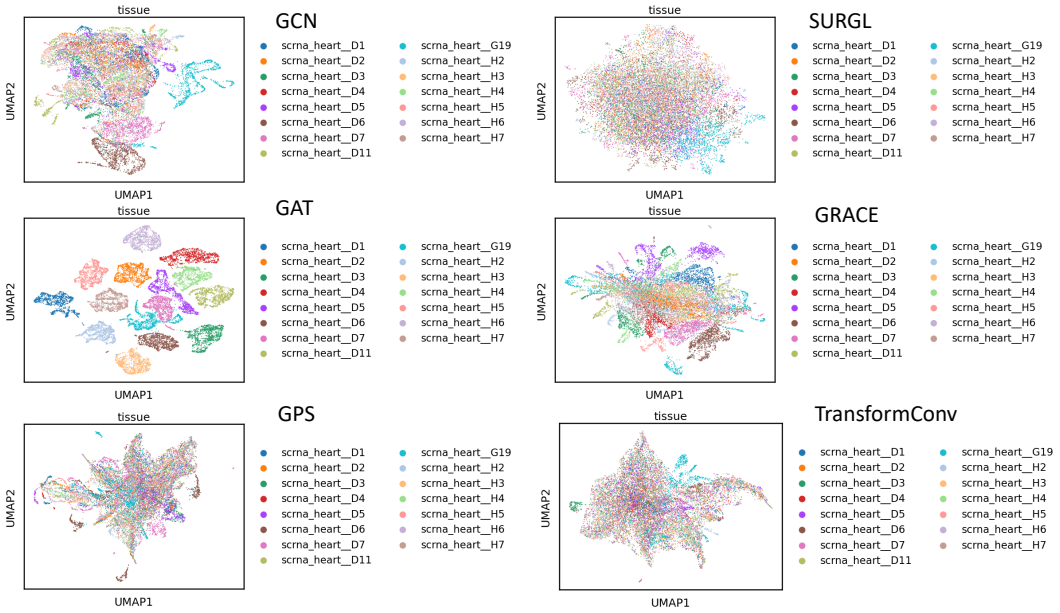

Figure 7: Gene embeddings based on different GNNs.

We also conduct experiments to justify the need for weight-sharing design. In this comparison, we evaluate the performance and model statistics of two models (with weight-sharing design (ws) and without weight-sharing design (w/o ws)). As shown in Table 6, the combined high or equivalent rank of weight-sharing designs in comparison to w/o weight-sharing designs is nearly consistent across all tissue types. Furthermore, the weight-sharing design can also reduce the model's training cost by decreasing the model size, the number of trainable parameters, and the total training time, as confirmed in Table 7.

Table 6: Comparison between with weight-sharing design and without weight-sharing design.

Methods	Heart	Lung	Liver	Kidney	Thymus	Spleen	Pancreas	Cerebrum	Cerebellum	PBMC	Avg Rank
ws	1.50	**1.33**	**1.33**	**1.33**	**1.33**	**1.33**	1.67	1.50	**1.17**	**1.17**	**1.37**
w/o ws	1.50	1.67	1.67	1.67	1.67	1.67	**1.33**	1.50	1.83	1.83	1.63

Table 7: Model statistics.

Methods	model size (MB)	# of parameters (M)	training time per epoch (s)
ws	**1333**	**349**	**3.64**
w/o ws	1346.8	353	4.85

To demonstrate the necessity of both expression profiles and graph structure for learning gene embeddings with biological information, we design an experiment in which all nodes in different graphs are replaced with the same features while maintaining the original graph structure. Based on the results shown in Figure 8 and Table 8, we can conclude that without the information from expression profiles, it becomes difficult for our model to learn unified gene embeddings because of the decline in scores of different metrics.

To demonstrate the necessity of Weighted Similarity Learning (WSL) design, we design an experiment to analyze the performance of different model structures with WSL or w/o WSL. According to Table 9, we found that MuSe-GNN with WSL performed better than MuSe-GNN without WSL across all of the metrics in heart tissue. Therefore, we need to utilize WSL manchism to effectively learn the gene-gene relationships across different datasets.

Table 8: Comparison between with feature design and without feature design.

Methods	ASW	AUC	iLISI	GC	CGR	NO
with feature	0.75±0.01	**0.78±0.02**	**0.53±0.01**	**0.73±0.04**	**0.65±0.02**	**0.31±0.00**
w/o feature	0.75±0.05	0.64±0.02	0.24±0.04	0.43±0.03	0.02±0.01	0.10±0.03

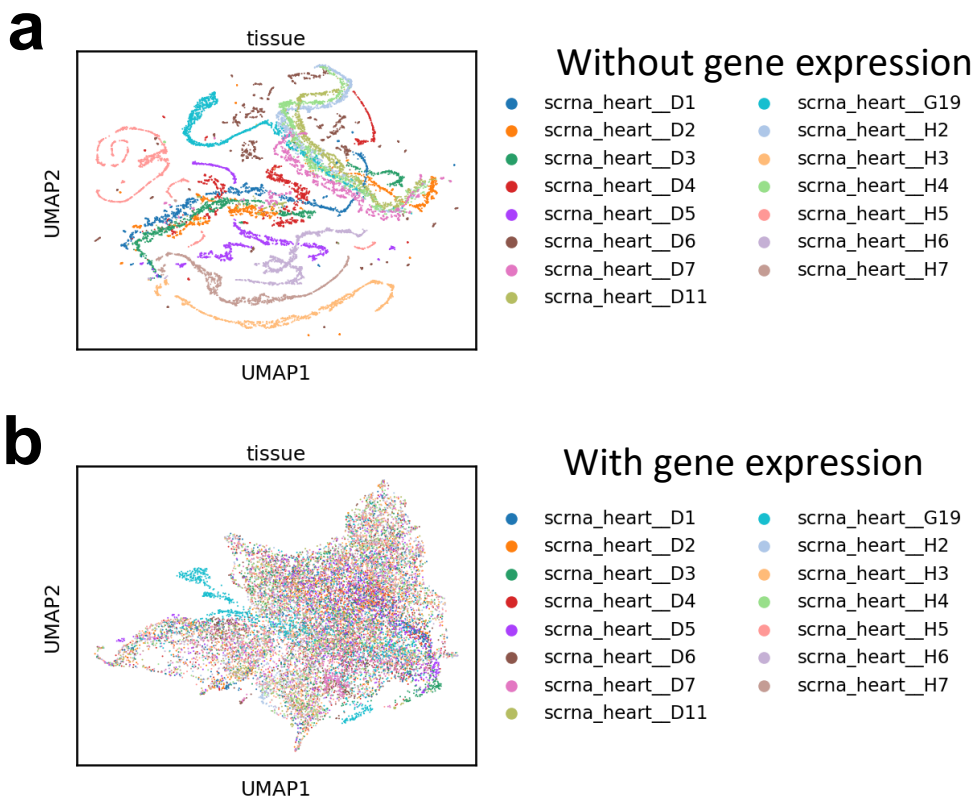

Figure 8: Gene embeddings based on different input information. **(a)** represents the UMAPs of gene embeddings without expression profiles and **(b)** represents the UMAPs of gene emebddings with expression profiles.

E.2 Hyper-parameter tuning

Hyper-parameter settings for various models can be divided into two groups: 1. Methods such as PCA, Gene2vec, GIANT, WSMAE, GAE, VGAE, and MAE, which are free of pre-training, are configured with the best hyper-parameters based on parameter searching to ensure fairness. Details are included in Table 10. 2. Method such as scBERT uses the pre-trained model. The optimal hyper-parameters for MuSe-GNN are provided in Table 11. Number of trainable parameters for different models is summarized in Table 12.

Table 9: Comparison between with weighted similarity learning design and without weighted similarity learning design.

Methods	ASW	AUC	iLISI	GC	CGR	NO
with WSL	**0.75±0.01**	**0.78±0.02**	**0.53±0.01**	**0.73±0.04**	**0.65±0.02**	**0.31±0.00**
w/o WSL	0.69±0.02	0.66±0.03	0.46±0.03	0.57±0.07	0.48±0.05	0.29±0.00

Table 10: Hyper-parameter candidates for benchmarking methods.

Models	Hyper-parameters
PCA	Dim:[32,128]
Gene2vec	Dim:[32,128]
GIANT	Dim:[32,128]
WSMAE	LR:[1e-4,1e-2]; Batch size:[1000,3000]; Dim:[32,128]
GAE	LR:[1e-4,1e-2]; Batch size:[1000,3000]; Dim:[32,128]
VGAE	LR:[1e-4,1e-2]; Batch size:[1000,3000]; Dim:[32,128]
MAE	LR:[1e-4,1e-2]; Batch size:[1000,3000]; Dim:[32,128]

Table 11: Hyper-parameters list for MuSe-GNN.

Parameter	Value
Epoch	2000
LR of encoder	1e-4
LR of decoder	1e-3
λ_c	1e-2
Dim of embeddings	32
Sample size	100

The average rank results for various hyper-parameters are presented in Table 13. We set the initial parameter settings as follows: Epoch = 2000, LR of encoder = 1e-2, LR of decoder = 1e-3, λ_c = 1e-2, Dim = 32, and Sample size (for contrastive learning) = 100. Notably, for the value choice of λ_c, we have two candidates with the same average rank result. To further compare the gene embeddings generated by models based on these two choices, we plot the UMAPs for these two models. According to Figure 9, we observe that choosing $\lambda_c = 0.1$ fails to integrate genes from dataset G19. Therefore, we ultimately set λ_c = 1e-2.

Additionally, the dimensions of gene embeddings should not be set to a very large value, not only due to the results in our table but also because of theoretical constraints and practical feasibility. Generally, an excessively large number of embedding settings may cause the number of hidden layer neurons to exceed the number of features in the input partial dataset, increasing the probability of overfitting by introducing more noise [57]. Furthermore, a larger number of dimensions will consume more computational resources, potentially leading to out-of-memory issues, such as when setting the Dim of embeddings = 256.

E.3 Details about benchmarking experiments

The average ranks result for all the scRNA-seq datasets is displayed in Table 14. The experimental results for all the scRNA-seq datasets are displayed in Tables 15 to 24 (unscaled). Based on the benchmark results for different tissues, we can conclude that MuSe-GNN consistently outperforms other methods according to specific metrics, such as iLISI and NO. Furthermore, the gene embeddings learned by MuSe-GNN are relatively more stable than those from other methods, as evidenced by the analysis of standard deviation. Considering the overall performance, MuSe-GNN, as the top-performing method, also surpasses the second-best methods from 20.1% to 97.5% and surpasses the third-best methods from 26.9% to 99.7%.

Table 12: Number of trainable parameters for different models

Models	# of parameters (M)
MuSe-GNN	349
GIANT	260
VGAE	210
WSMAE	105
GAE	52.5
MAE	52.5

Table 13: Hyper-parameter searching results.

LR for encoder	Avg Rank	InfoNCE penalty	Avg Rank
0.01	3.17	1	3.5
0.001	2.50	0.1	**2.13**
0.0005	2.33	0.01	**2.13**
0.0001	**1.83**	0.001	2.25
LR for decoder	Avg Rank	Dim	Avg Rank
0.01	3.43	32	**1.86**
0.001	**1.86**	64	2.00
0.0005	2.29	128	2.14
0.0001	2.43	256	4.00
batch size	Avg Rank	sample size	Avg Rank
1000	3	50	2.38
1500	3	100	**1.75**
2000	**1.5**	150	2.88
3000	2.5	200	2.5

Table 14: The average ranks for gene embeddings benchmarking across different tissues.

Methods	Heart	Lung	Liver	Kidney	Thymus	Spleen	Pancreas	Cerebrum	Cerebellum	PBMC
PCA	4.33	4.33	4.67	5.50	4.00	4.17	4.83	4.50	5.50	4.00
Gene2vec	5.67	6.67	6.50	6.67	7.33	7.17	6.83	6.67	6.00	7.33
GIANT	6.00	4.67	5.33	4.67	3.67	5.67	3.83	5.50	6.00	6.17
WSMAE	5.83	5.33	4.83	4.83	3.83	5.17	5.00	5.33	4.33	5.00
GAE	4.33	5.83	5.33	5.50	5.17	5.33	4.50	6.00	5.00	5.33
VGAE	3.50	6.33	6.33	6.67	4.83	6.50	6.17	6.67	6.83	5.83
MAE	7.00	5.17	5.17	5.17	5.50	5.17	5.67	5.33	5.33	5.33
scBERT	5.67	4.50	4.33	3.17	7.83	3.83	4.83	3.17	3.17	3.67
MuSeGNN	**2.67**	**2.17**	**2.50**	**2.83**	**2.67**	**2.00**	**2.83**	**1.67**	**2.67**	**2.33**

Table 15: Benchmark score table for Heart.

Methods	ASW	AUC	iLISI	GC	CGR	NO
PCA	0.77±0.01	0.64±0.00	0.43±0.00	0.49±0.02	0.30±0.01	0.27±0.00
Gene2vec	0.82±0.03	0.76±0.00	0.09±0.01	0.65±0.03	0.00±0.00	0.10±0.02
GIANT	**0.84±0.01**	0.55±0.00	0.26±0.02	0.48±0.03	0.10±0.01	0.01±0.00
WSMAE	0.79±0.01	0.72±0.00	0.34±0.01	0.43±0.04	0.11±0.02	0.26±0.01
GAE	0.79±0.01	**0.98±0.00**	0.38±0.01	0.45±0.02	0.12±0.02	0.28±0.01
VGAE	0.80±0.01	0.97±0.00	0.39±0.01	0.44±0.02	0.18±0.03	0.28±0.00
MAE	0.82±0.01	0.52±0.00	0.30±0.02	0.24±0.05	0.03±0.00	0.22±0.01
scBERT	0.77±0.01	0.47±0.00	0.39±0.01	0.37±0.03	0.26±0.01	0.28±0.00
MuSe-GNN	0.75±0.01	0.78±0.02	**0.53±0.01**	**0.73±0.04**	**0.65±0.01**	**0.31±0.00**

Table 16: Benchmark score table for Lung.

Methods	ASW	AUC	iLISI	GC	CGR	NO
PCA	0.79±0.00	0.69±0.00	0.58±0.00	0.70±0.01	0.07±0.01	0.10±0.00
Gene2vec	0.71±0.40	0.76±0.00	0.00±0.00	**0.89±0.01**	0.00±0.00	0.01±0.00
GIANT	**0.90±0.01**	0.49±0.00	0.12±0.03	0.88±0.03	0.07±0.01	0.01±0.00
WSMAE	0.76±0.01	0.69±0.01	0.45±0.03	0.77±0.03	0.05±0.01	0.09±0.01
GAE	0.73±0.03	0.86±0.00	0.23±0.04	0.78±0.03	0.02±0.00	0.07±0.01
VGAE	0.51±0.10	**0.87±0.01**	0.00±0.00	0.87±0.03	0.00±0.00	0.01±0.01
MAE	0.85±0.02	0.62±0.01	0.35±0.05	0.81±0.03	0.02±0.01	0.08±0.01
scBERT	0.87±0.01	0.48±0.00	0.29±0.02	0.86±0.05	0.06±0.01	0.10±0.00
MuSe-GNN	0.84±0.02	0.86±0.01	**0.64±0.04**	0.87±0.03	**0.37±0.03**	**0.19±0.00**

Table 17: Benchmark score table for Liver.

Methods	ASW	AUC	iLISI	GC	CGR	NO
PCA	0.77±0.00	0.58±0.00	0.39±0.00	0.69±0.01	0.21±0.00	0.15±0.00
Gene2vec	0.63±0.34	0.69±0.00	0.00±0.00	0.88±0.01	0.00±0.00	0.03±0.01
GIANT	**0.90±0.01**	0.55±0.00	0.40±0.02	0.44±0.03	0.11±0.01	0.00±0.00
WSMAE	0.80±0.01	0.62±0.01	0.26±0.03	0.83±0.03	0.09±0.01	0.13±0.01
GAE	0.74±0.02	0.84±0.00	0.17±0.02	0.80±0.01	0.05±0.01	0.14±0.01
VGAE	0.49±0.05	**0.87±0.00**	0.00±0.00	0.87±0.02	0.01±0.00	0.01±0.00
MAE	0.83±0.01	0.58±0.01	0.25±0.04	0.88±0.04	0.04±0.01	0.09±0.01
scBERT	0.87±0.01	0.50±0.00	0.23±0.05	**0.90±0.03**	0.11±0.02	0.14±0.00
MuSe-GNN	0.86±0.01	0.78±0.03	**0.65±0.02**	0.83±0.04	**0.46±0.02**	**0.18±0.00**

Table 18: Benchmark score table for Kidney.

Methods	ASW	AUC	iLISI	GC	CGR	NO
PCA	0.69±0.01	0.55±0.00	0.39±0.01	0.98±0.00	0.00±0.00	0.21±0.00
Gene2vec	0.33±0.42	0.74±0.00	0.00±0.00	1.00±0.01	0.00±0.00	0.02±0.03
GIANT	**0.92±0.00**	0.55±0.00	**0.89±0.01**	0.78±0.03	0.02±0.00	0.01±0.00
WSMAE	0.74±0.05	0.68±0.01	0.08±0.04	0.99±0.00	0.01±0.00	0.15±0.04
GAE	0.52±0.10	0.86±0.00	0.01±0.01	0.98±0.01	0.00±0.00	0.05±0.02
VGAE	0.51±0.16	**0.88±0.00**	0.00±0.00	0.98±0.01	0.00±0.00	0.00±0.01
MAE	0.84±0.04	0.52±0.01	0.17±0.07	**1.00±0.00**	0.00±0.00	0.16±0.04
scBERT	0.89±0.02	0.47±0.00	0.59±0.18	**1.00±0.00**	0.05±0.01	0.30±0.00
MuSe-GNN	0.88±0.03	0.71±0.02	0.84±0.04	0.98±0.02	**0.31±0.04**	**0.31±0.01**

Table 19: Benchmark score table for Thymus.

Methods	ASW	AUC	iLISI	GC	CGR	NO
PCA	0.76±0.01	0.79±0.00	0.93±0.00	0.44±0.05	0.00±0.00	0.00±0.00
Gene2vec	0.00±0.00	0.75±0.00	0.00±0.00	0.58±0.03	0.00±0.00	0.00±0.00
GIANT	**0.88±0.02**	0.55±0.00	**0.98±0.01**	**0.80±0.02**	0.00±0.00	0.00±0.00
WSMAE	0.81±0.03	0.74±0.01	0.93±0.06	0.48±0.06	0.00±0.00	0.00±0.00
GAE	0.72±0.12	0.81±0.00	0.89±0.10	0.69±0.05	0.00±0.00	0.00±0.00
VGAE	0.77±0.05	**0.84±0.01**	0.89±0.05	0.44±0.10	0.00±0.00	0.00±0.00
MAE	0.77±0.28	0.59±0.01	0.94±0.05	0.33±0.05	0.00±0.00	0.00±0.00
scBERT	0.09±0.27	0.50±0.00	0.41±0.18	0.42±0.04	0.00±0.00	0.00±0.00
MuSe-GNN	0.71±0.05	0.80±0.02	**0.98±0.01**	0.65±0.07	**0.01±0.00**	0.00±0.00

Table 20: Benchmark score table for Spleen.

Methods	ASW	AUC	iLISI	GC	CGR	NO
PCA	0.81±0.01	0.56±0.00	0.47±0.00	0.73±0.01	0.23±0.01	0.17±0.00
Gene2vec	0.69±0.36	0.68±0.01	0.00±0.00	0.85±0.01	0.00±0.00	0.01±0.01
GIANT	**0.88±0.01**	0.55±0.00	0.42±0.02	0.45±0.02	0.10±0.00	0.01±0.00
WSMAE	0.79±0.01	0.61±0.01	0.26±0.03	0.86±0.02	0.08±0.01	0.15±0.01
GAE	0.75±0.02	0.84±0.00	0.18±0.02	0.81±0.01	0.05±0.01	0.16±0.01
VGAE	0.52±0.05	**0.86±0.00**	0.00±0.00	0.85±0.03	0.00±0.00	0.01±0.00
MAE	0.84±0.02	0.56±0.01	0.30±0.04	0.92±0.02	0.03±0.00	0.12±0.00
scBERT	0.87±0.01	0.50±0.00	0.34±0.06	0.94±0.02	0.11±0.01	0.16±0.00
MuSe-GNN	0.86±0.01	0.79±0.02	**0.70±0.02**	**0.89±0.03**	**0.47±0.01**	**0.19±0.00**

Table 21: Benchmark score table for Pancreas.

Methods	ASW	AUC	iLISI	GC	CGR	NO
PCA	0.61±0.01	0.62±0.00	0.64±0.00	0.57±0.01	0.00±0.00	0.01±0.00
Gene2vec	0.17±0.36	0.73±0.00	0.00±0.00	0.51±0.05	0.00±0.00	0.00±0.00
GIANT	**0.91±0.01**	0.55±0.00	**0.97±0.03**	0.65±0.05	0.00±0.00	0.00±0.00
WSMAE	0.67±0.09	0.70±0.01	0.66±0.10	0.49±0.06	0.00±0.00	0.00±0.00
GAE	0.09±0.27	0.82±0.00	0.90±0.07	**0.65±0.03**	0.00±0.00	0.00±0.00
VGAE	0.37±0.40	**0.83±0.01**	0.13±0.10	0.51±0.04	0.00±0.00	0.00±0.00
MAE	0.71±0.11	0.56±0.01	0.83±0.07	0.37±0.05	0.00±0.00	0.00±0.00
scBERT	0.76±0.05	0.47±0.00	0.54±0.15	0.39±0.04	0.00±0.00	**0.01±0.00**
MuSe-GNN	0.72±0.08	0.65±0.01	0.95±0.13	0.62±0.06	**0.01±0.00**	0.00±0.00

Table 22: Benchmark score table for Cerebrum.

Methods	ASW	AUC	iLISI	GC	CGR	NO
PCA	0.82±0.01	0.63±0.00	0.34±0.00	0.95±0.00	0.12±0.00	0.18±0.00
Gene2vec	0.07±0.23	0.69±0.00	0.00±0.00	**0.99±0.00**	0.00±0.00	0.01±0.01
GIANT	0.89±0.00	0.55±0.00	0.40±0.02	0.51±0.02	0.03±0.00	0.01±0.00
WSMAE	0.80±0.02	0.67±0.01	0.21±0.04	0.97±0.01	0.03±0.01	0.13±0.02
GAE	0.68±0.04	0.83±0.00	0.03±0.01	0.95±0.01	0.02±0.00	0.13±0.01
VGAE	0.47±0.09	**0.84±0.01**	0.00±0.00	0.98±0.01	0.00±0.00	0.01±0.00
MAE	0.83±0.02	0.53±0.01	0.20±0.05	0.99±0.01	0.02±0.01	0.14±0.01
scBERT	0.89±0.01	0.48±0.00	0.50±0.07	**0.99±0.00**	0.10±0.01	0.17±0.00
MuSe-GNN	**0.90±0.01**	0.73±0.02	**0.79±0.03**	0.99±0.01	**0.54±0.01**	**0.21±0.00**

Table 23: Benchmark score table for Cerebellum.

Methods	ASW	AUC	iLISI	GC	CGR	NO
PCA	0.82±0.00	0.63±0.00	0.34±0.00	0.95±0.00	0.12±0.01	0.18±0.00
Gene2vec	0.31±0.40	0.71±0.00	0.00±0.00	**1.00±0.00**	0.00±0.00	0.02±0.02
GIANT	0.89±0.00	0.55±0.00	0.34±0.03	0.73±0.02	0.03±0.00	0.01±0.00
WSMAE	0.83±0.02	0.66±0.01	0.26±0.04	**1.00±0.00**	0.06±0.01	0.29±0.02
GAE	0.74±0.02	**0.83±0.00**	0.11±0.08	0.99±0.01	0.07±0.04	0.25±0.04
VGAE	0.19±0.26	0.83±0.01	0.00±0.00	0.99±0.01	0.00±0.00	0.00±0.00
MAE	0.86±0.01	0.51±0.00	0.16±0.03	**1.00±0.00**	0.02±0.00	0.26±0.03
scBERT	0.90±0.01	0.47±0.00	0.54±0.06	**1.00±0.00**	0.15±0.02	0.31±0.01
MuSe-GNN	**0.92±0.01**	0.64±0.02	**0.86±0.02**	0.98±0.02	**0.65±0.00**	**0.38±0.00**

Table 24: Benchmark score table for PBMC.

Methods	ASW	AUC	iLISI	GC	CGR	NO
PCA	0.86±0.01	0.53±0.00	0.46±0.00	0.66±0.01	0.22±0.01	0.13±0.00
GENE2VEC	0.00±0.00	0.59±0.00	0.00±0.00	0.92±0.01	0.00±0.00	0.00±0.00
GIANT	0.85±0.01	0.55±0.00	0.38±0.02	0.55±0.03	0.04±0.00	0.00±0.00
WSMAE	0.81±0.01	0.58±0.01	0.29±0.02	0.81±0.05	0.10±0.02	0.12±0.01
GAE	0.74±0.01	0.82±0.00	0.15±0.03	0.79±0.03	0.06±0.02	0.12±0.01
VGAE	0.67±0.04	**0.89±0.00**	0.00±0.00	0.93±0.02	0.02±0.00	0.01±0.00
MAE	0.85±0.02	0.53±0.01	0.30±0.05	**0.94±0.03**	0.02±0.00	0.10±0.01
scBERT	**0.88±0.01**	0.52±0.00	0.48±0.05	0.93±0.03	0.15±0.02	0.11±0.00
MuSe-GNN	0.86±0.01	0.79±0.02	**0.75±0.02**	0.88±0.02	**0.54±0.01**	**0.18±0.00**

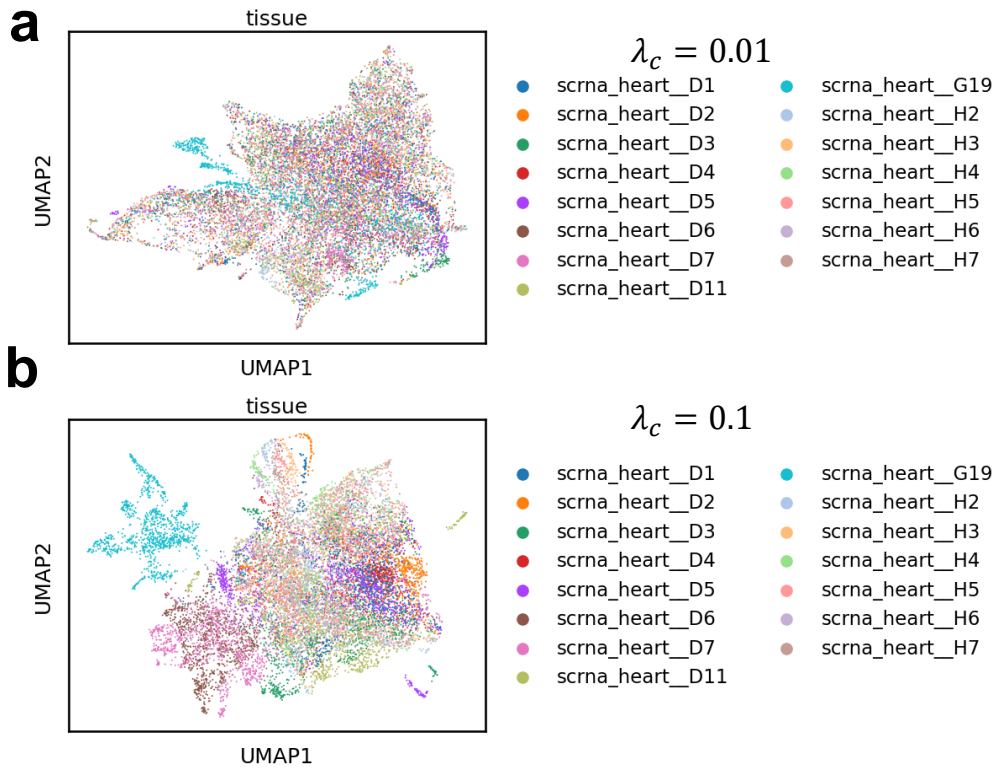

Figure 9: Gene embeddings on different λ_c.

E.4 Details about function cluster identification

In this paper, we utilized Leiden cluster method [91] to identify genes with common functions based on the value of their embeddings. The genes with distance smaller than a community-discovering based threshold (known as co-embedded genes) will be identified into a group. Here we sample 100 genes in our final gene embedding result from heart tissue to illustrate the distribution of such distance. Based on Figure 10, we can conclude that our method has the ability to identify genes with similar embeddings, and further, it can be used to derive genes with similar functions.

All of the experiments of this paper are finished based on Intel Xeon Gold 6240 Processor and one NVIDIA A100 GPU. The RAM setting is 30 GB.

F Extra Discussion about Model Design

This section aims to demonstrate the reasoning behind our model design in greater details. As a novel graph neural network model, MuSe-GNN offers three significant advantages.

Firstly, our method's graph construction process is robust and reliable. Graph construction is a critical step for GNN models [117], and we place considerable emphasis on both the graph construction method and data quality throughout our work. In single-cell data analysis research, some researchers model cell-cell similarity based on the nearest neighbor graph [99, 94]. This approach may be affected by batch effects resulting from different sequencing profiles, leading to incorrect conclusions. For instance, in scRNA-seq data imputation tasks, such a design may result in a high false positive rate [5]. A gene co-expression network derived from a robust statistical framework is more trustworthy, and biological experiments can validate the co-expression performance of various genes. Moreover, to understand the difference of co-expression networks from different tissues or omics data, we calculate the Edge Signal-Noise Ratio (ESNR) [22] for each network, and figure out that the co-expression

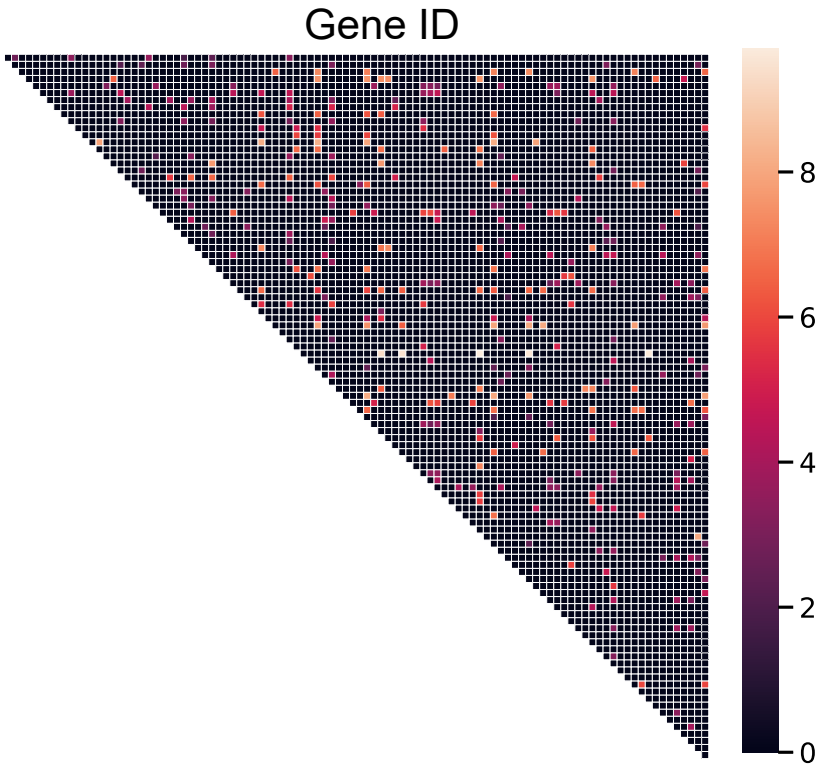

Figure 10: Heatmap of gene embedding distances. The heatmap color represents the distance between genes, with brighter colors indicating greater distances and more distinct functions.

network based on scRNA-seq has the best quality, while the co-expression networks from other omics have very small ESNR. Such conclusions are also consistent with the model performance.

Secondly, our model utilizes a sophisticated and stable training strategy. By employing the cosine similarity penalty, we ensure that genes with the same functions from different tissues or techniques can be co-embedded in similar positions. Additionally, we use graph auto-encoder and contrastive learning strategies to retain the original biological information of various datasets. Each of our training strategies serves a specific purpose.

Finally, our method showcases the high effectiveness of MMML in addressing heterogeneous data learning challenges. MMML can successfully learn similar information across multimodal data and represent it in a unified space through model fusion [55]. This approach can be applied to more complex data studies in the future, such as integrating data from Magnetic Resonance Imaging (MRI) results, spatial resolution image information, and current expression profiles into a single space. Furthermore, GNN remains the primary model for handling tasks related to graph-structured data. By combining these methods and uncovering biological insights, our research contributes to a deeper understanding of efficient human genome representation.

G Selecting anchors of common functional genes

In this paper, we employ explicit common functional genes as anchors across different datasets to implement regularization in our training process. The common functional genes we select can potentially recover a portion of the functional overlap of the genes, while the remaining implicit genes can be recovered during the training process.

In this paper, we choose to use common highly variable genes (HVGs) combined from different datasets as anchors, and employ their co-expression network overlap as weights. An example is illustrated in Figure 12 case 1. In addition to the proof provided by the ablation test results, there are four reasons for this choice: 1. HVGs represent the active genes of a expression profile (dataset),

which are correlated to gene functions. 2. For datasets from the same tissue, the overlap of HVGs and the overlap of neighbor genes (known as the co-expressed genes in the same graph) of HVGs pair are similar. This demonstrates a type of consistency in biological function inference. 3. For datasets from different modalities (tissues, techniques, etc.), only using the overlap of HVGs is not reliable. The overlap of neighbor genes of HVGs pair from different tissues shows functional similarity based on hierarchical clustering. We can observe the clear difference of HVGs overlap score and neighbors overlap score of HVGs by comparing Figure 11 (a) to (b). Furthermore, we can consider an extreme case, illustrated in Figure 12 case 2. If two HVGs from different tissues do not contain shared neighbors, the training process will be processed for such genes without weighted constraints. 4. Since it is quite challenging to select all the true genes with similar functions, we can perform a trade-off between learning the similarity of potential functionally similar genes and learning the difference of potential functionally distinct genes. The neighbor genes overlap in common HVGs is significantly different from the overlap in different HVGs for graphs from the same tissue, as shown in Figure 13. Such statistical results also support our choice of using common HVGs as anchors.

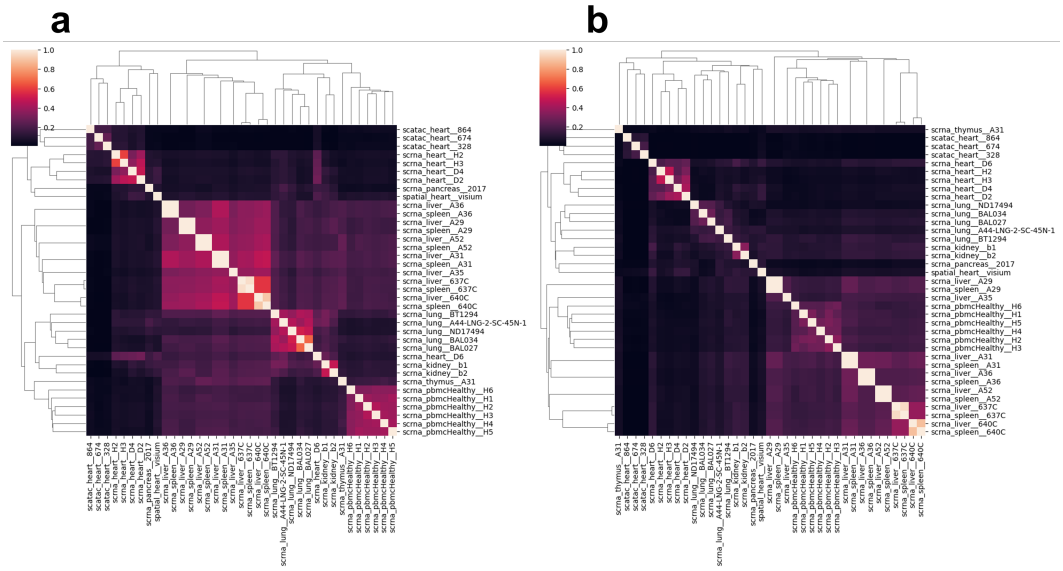

Figure 11: Cluster heatmaps of anchor genes for multimodal data. **(a)** represents the cluster heatmap based on HVGs overlap, **(b)** represents the cluster heatmap based on the neighbor genes overlap of HVGs.

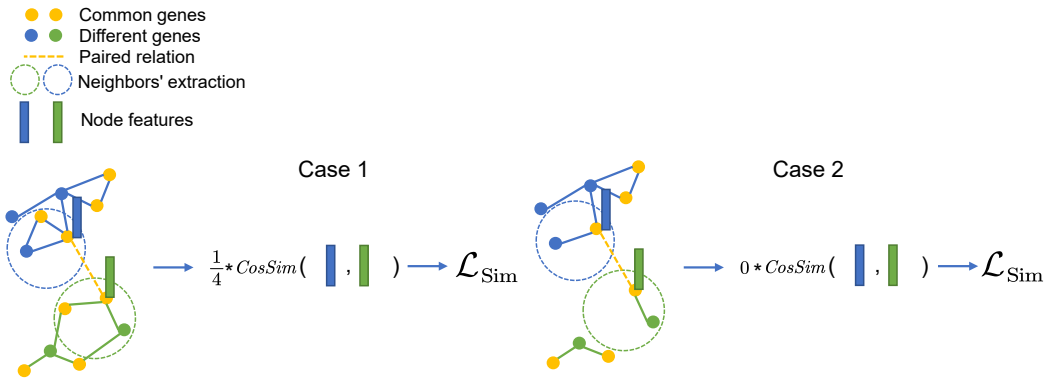

Figure 12: The need for weight similarity learning illustrated by two cases.

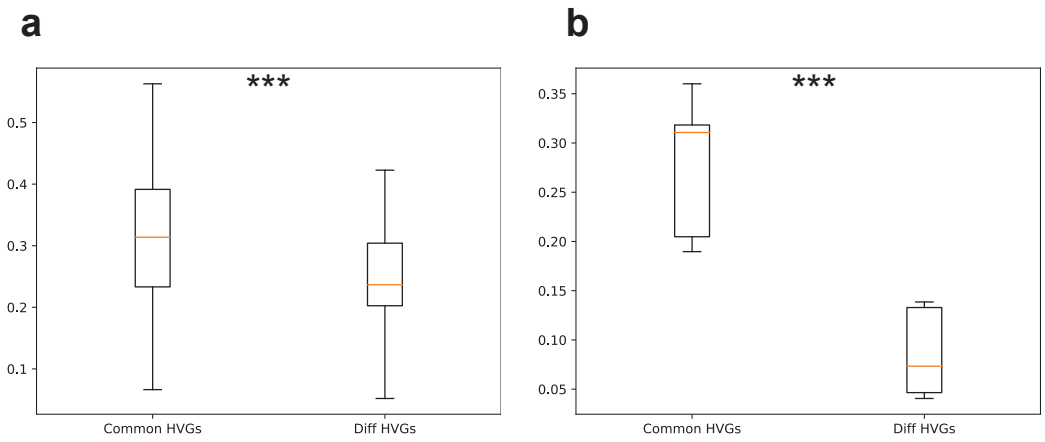

Figure 13: Boxplots for the neighbor overlap score of HVGs from different tissues. We used Kolmogorov–Smirnov test [97] to compare the difference between the two groups of genes. The significance level of group difference is shown by the stars ($***$: P-value ≤ 0.01). **(a)** represent the distribution of overlap scores for heart tissue, grouped by common HVGs and different HVGs. **(b)** represent the distribution of overlap score for cerebrum tissue, grouped by common HVGs and different HVGs.

In conclusion, we aim to generate similar embeddings for functionally related genes, incorporating common HVGs and their neighbors in co-expression networks, by utilizing the regularization term associated with common HVGs. Conversely, embeddings for functionally distinct genes should differ, enabling their identification through community detection-based clustering algorithms.

H Metrics details

Here we assume that HVGs from the same tissues with the same name have similar functions across different datasets, and we also consider evaluating the preservation of the known gene co-expression relation by using node similarity. These metrics include:

- edge AUC: Edge Area under the ROC Curve (AUC) is a metric widely used in the edge prediction task. We calculate the AUC score between the multiplication of normalized embeddings with Sigmoid threshold ($Sigmoid(zz^T)$) and the true edge relation. This metric can reflect the co-expression information we have in our embedding space.
- common genes ASW: Common genes ASW represents the ASW score we calculated based on common genes across different tissues. Since the overlap genes of different datasets are not very large, based on the assumption that different highly variable genes have similar functions in different datasets, we can utilize common genes ASW as a metric to evaluate our integration performance.
- common genes Graph connectivity: Graph connectivity is a metric to evaluate the connectivity based on the ratio between the number of nodes in the largest connected component of the graph constructed by the gene cluster and the number of nodes belonging to the graph. We calculate the graph connectivity score for each gene cluster and take the average.
- common genes iLISI: The inverse Simpson's index (iLISI) determines the number of common genes that can be drawn from a neighbor list before one dataset (or datasets from the same tissue) is observed twice. Therefore, this metric can be used to evaluate the integration level of the common genes across different datasets or tissues based on the choice of our index type.
- common genes ratio: The common genes ratio is defined based on the weighted average of one minus the ratio of unique genes from one cluster to the total genes in the cluster across all the clusters. We utilize the Leiden method to generate the clusters. This metric can be used to evaluate the level of integration of the same functional genes by a given method.

- neighbors overlap: The neighbor overlap value is defined based on the neighbors of one gene in the co-expression network. For each cluster, we calculate the similarity of genes' neighbors among different datasets and perform a weighted average of the similarity. A larger value means that the method is able to integrate similar genes.

I Shared transcription factors and pathways analysis

Transcription factors (TFs) are crucial in transcriptional regulation [37]. By examining the overlap of TFs across diverse datasets, we can assess the similarities between distinct modal data in terms of transcriptional regulation. We employed the TRRUST v2 database [37] to identify regulators for each gene within the gene embedding space. Subsequently, for every cluster, we extracted the transcription factors associated with the genes and computed the transcription factor overlap for the various multimodal data linked to the genes. We iteratively repeated this process for each cluster and documented the number of shared transcription factors based on different combinations of biological multimodal data. The results are illustrated in Figure 14.

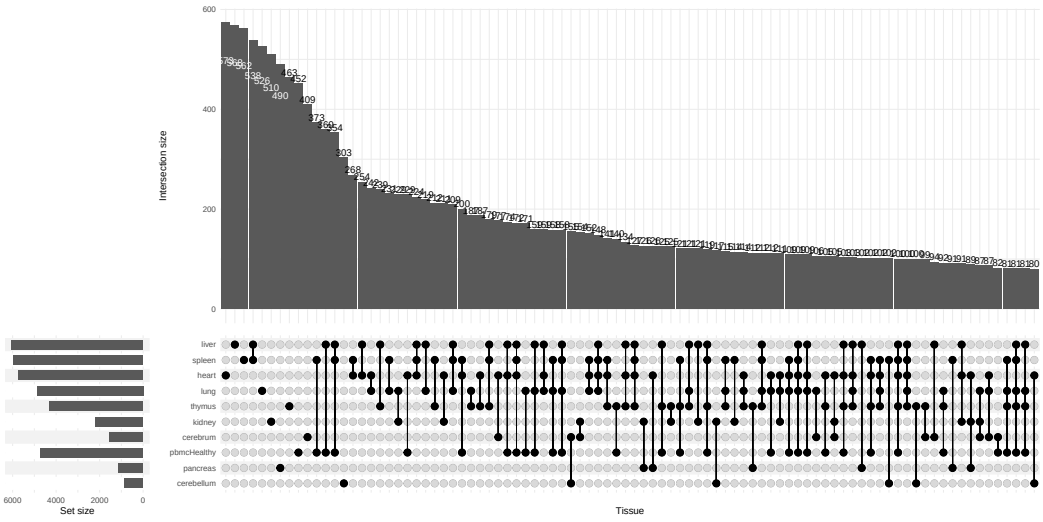

Figure 14: Results of common TFs discovery.

Figure 14 reveals that tissues with similar functions exhibit greater TF overlap, such as the heart and lung group, and the spleen and liver group. These findings demonstrate that our generated embeddings effectively capture the similarities between various data types. ComplexUpset [51] was utilized to create this figure.

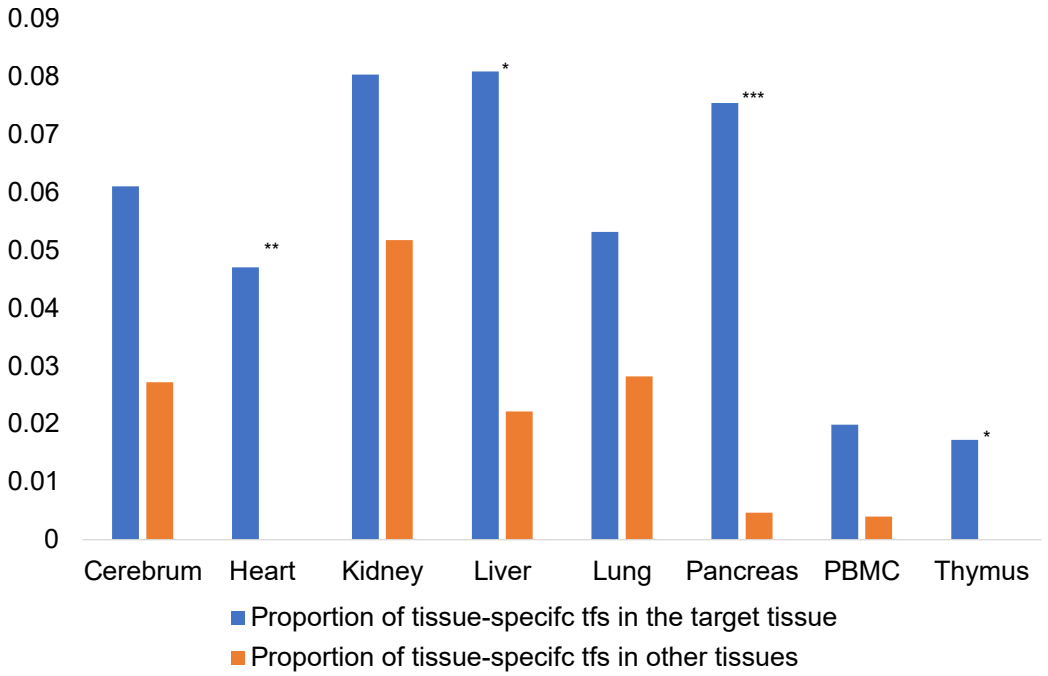

Figure 15: The proportion of tissue-specific TFs enrichment in the target tissue (blue) and in other tissues (orange). Here the significant enrichment in blue groups are marked by stars ($*$: P-value ≤ 0.1; $**$: P-value ≤ 0.05; $***$: P-value ≤ 0.01;).

Additionally, we can emphasize the enrichment status of tissue-specific TFs, as depicted in Figure 15. We utilized the TF information from [107] to compute the enrichment proportion for tissue-specific TFs enriched in the given tissue and other tissues. A one-tailed Fisher's exact test was also conducted to assess the significance of enrichment differences. The figure suggests that our gene embeddings consistently enrich tissue-specific TFs across all evaluated tissues. Notably, for some tissues, the enrichment is significant, which aligns with the conclusions from [16]. However, in other tissues, the difference is not significant. A potential explanation is that such tissues may play a critical role in the interaction between different tissues for specific biological processes.

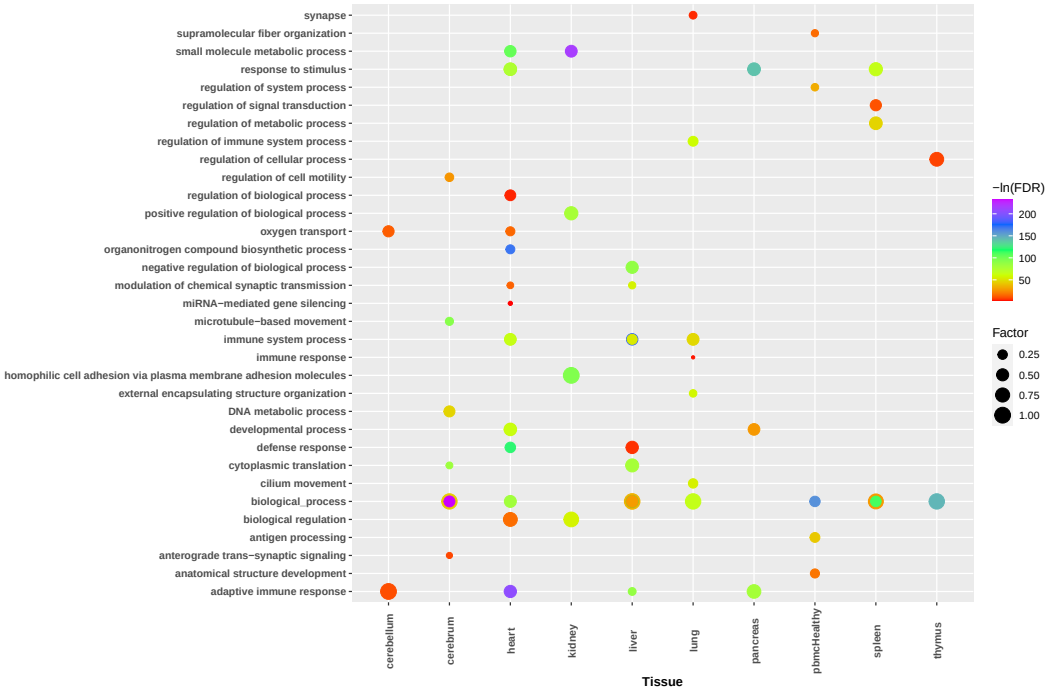

Figure 16: Top1 enriched pathways across different gene function clusters grouped by major tissue in each cluster. Here the color of bubbles represents the $-ln(FDR)$ value and the size of bubbles represents the value of rich factor for different pathways.

In Figure 16, we determined the most enriched pathway for each cluster and classified it into the most representative tissue within that cluster by GOEA. The representative tissue was chosen based on the highest proportion of the source tissue for genes. This figure allows us to conclude that the heart tissue is the most versatile in the human body, followed by liver and lung tissues, which are also crucial tissues in humans.

J Multi-species gene embeddings

Integrating datasets from different species based on cells is challenging because we always need extra information, either orthology genes (genes from different species whose evaluation is only related to speciation events) information or protein embeddings information [76]. However, using genes as anchors simplifies this task. We examined gene embeddings generated by MuSe-GNN across three mammalian species, including humans, lemurs, and mice. As shown in Figure 17 (a) and (b), our method effectively integrated information from different species into a shared space, preserving gene similarity across various techniques, tissues, and species. We also observed the co-embedding of genes from different species but the same tissue, supporting the similarity of evaluation direction. Since the enrichment of orthology genes can also be used as proof to evaluate whether genes with similar functions are grouped [32], we collected genes from different clusters with three species pairs and computed the proportion of these genes in the orthology genes database [4]. Furthermore, we compared the results based on gene embeddings to random selection as a null approach. From Figure 18, we could conclude that gene embeddings from our model can effectively enrich genes from different species but with similar functions.

Furthermore, Figure 17 (c) reveals that genes from different species are distributed across multiple common function clusters. To explore shared gene functions across species, we selected one common function cluster containing genes from all three species for GOEA analysis. Since most genes in this cluster originated from heart tissue, we anticipated pathway enrichment related to multicellular systems and circulatory systems, as displayed in Figure 3 (d). Thus, we demonstrated that MuSe-

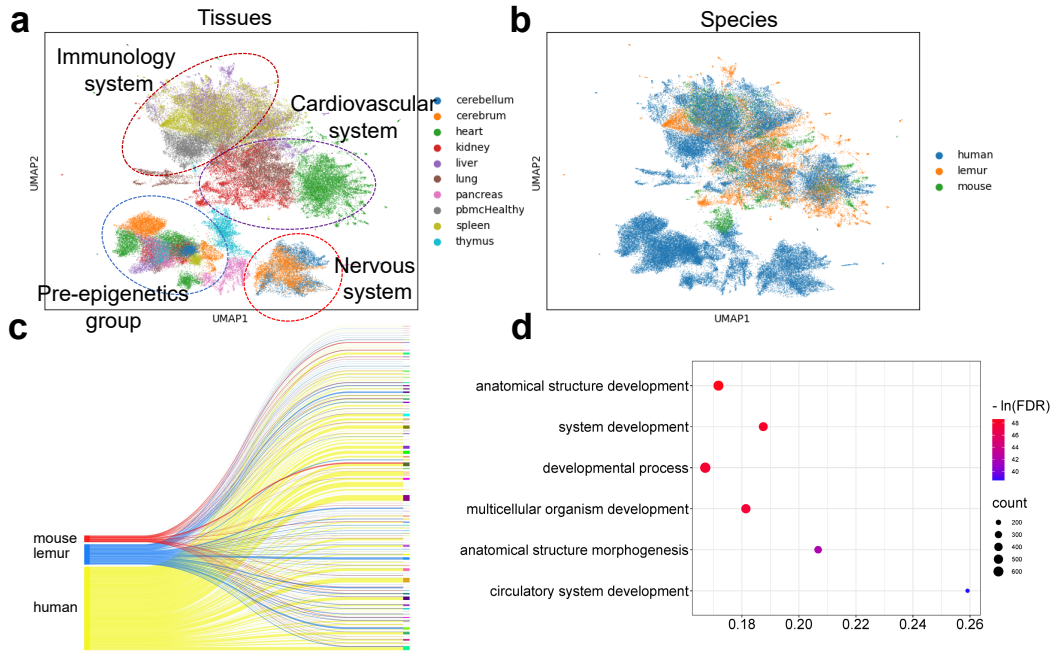

Figure 17: Gene representation learning results for multimodal biological data with different species. **(a)** represents the UMAPs of gene embeddings colored by tissue type. **(b)** represents the UMAPs of gene embeddings colored by common function groups. **(c)** is a Sankey plot [80] to show the genes overlap of different species in the same clusters. **(d)** shows the top5 pathways related to the genes in the special cluster discovered by GOEA. The bubble plots in this paper were created based on ggplot2 [102].

GNN can learn gene embeddings from multiple species and identify genes with similar functions across various species.

K Lung cancer analysis

In this section, we applied MuSe-GNN to analyze differentially co-expressed genes in two types of datasets, comprising lung cancer samples and healthy samples. We converted the scRNA-seq results into distinct graphs according to our graph construction strategy and trained MuSe-GNN to generate gene embeddings. We then employed the Leiden algorithm to identify common functional gene groups. After collecting essential genes with cancer-related special functions, we used GOEA and IPA to uncover more biological information.

As shown in Figure 19 (a)-(c), we observed that for lung tissue, the co-expression relationships of genes in lung tumor cells and normal cells differ significantly. Figure 19 (d) displays the top pathways identified by GOEA for genes recognized as specifically co-expressed in cancer cells, with most pathways related to tissue development and cell migration. These pathways align with the fact that cancer cells often undergo multiple divisions and frequently relocate. Furthermore, using IPA, we analyzed causal networks and disease functions. Figure 19 (e) illustrates an example of a causal network discovered in a specific gene group, primarily regulated by ethyl protocatechuate, indicating a chemical-dominated regulatory network. Thus, with the genes identified by MuSe-GNN, we can uncover various specific regulatory networks present in cancer and investigate strategies to inhibit cancer cell growth and metastasis. Lastly, Figure K (f) reveals that functions with low FDR rates include tissue development and cancer disease.

Our analysis demonstrates that MuSe-GNN can be utilized to discover functional gene clusters exhibiting specificity in cancer, assisting researchers in gaining a deeper understanding of gene expression and regulatory relationships in cancer.

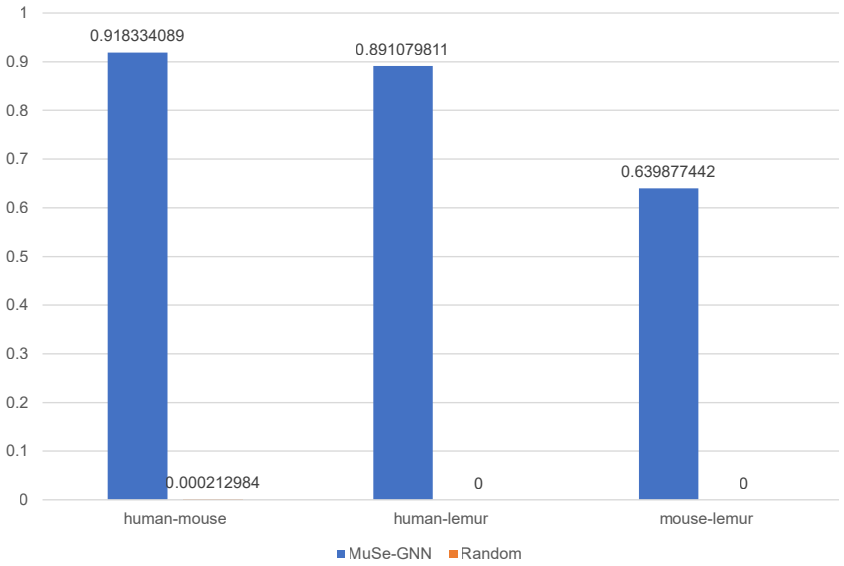

Figure 18: Compare the proportion of orthology genes discovered by MuSe-GNN and random selection.

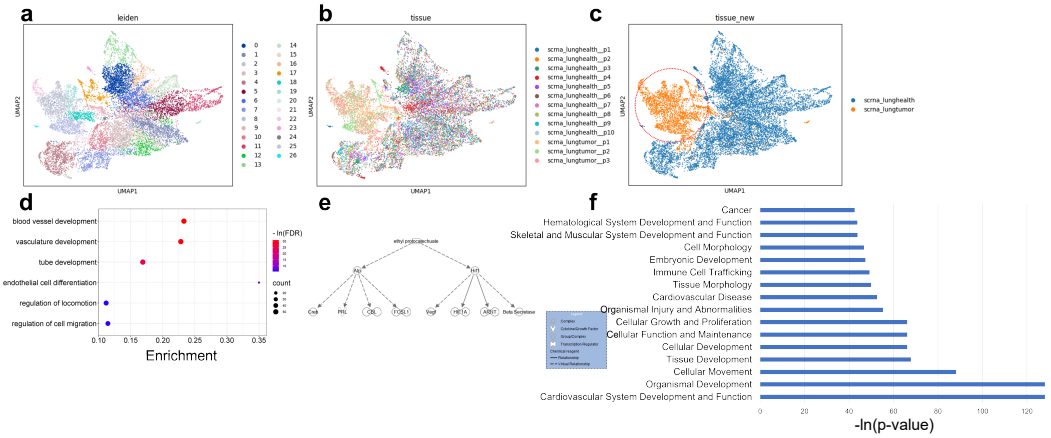

Figure 19: Gene representation learning results for samples with cancer and healthy samples. **(a)** represents the UMAPs of gene embeddings colored by functional groups. **(b)** represents the UMAPs of gene embeddings colored by datasets. **(c)** represents the gene embeddings colored by the conditions, and the red circle reflects the differential co-expression genes. **(d)** shows the top6 pathways related to the genes in the special cluster discovered by GOEA. **(e)** represents the causal network existing in the special cluster discovered by IPA. **(f)** represents the top diseases & biological functions discovered by IPA.

L Gene Function Analysis

Here we continued discussing the application of gene embeddings in the gene function prediction task. We further intended to predict the characters of genes in the regulation process (as transcript factor or not) [37]. We used MuSe-GNN to generate gene embeddings for different datasets based on an unsupervised learning framework and utilized the gene embeddings as training dataset to predict the function of genes based on k-NN classifier.

In this task, we evaluated the performance of MuSe-GNN based on scRNA-seq datasets from different tissues, comparing it to the prediction results based on raw data or PCA. On average, the gene embeddings from MuSe-GNN are the most powerful embeddings, yielding highest accuracy in

the validation set. Moreover, the accuracy based on gene embeddings from MuSe-GNN also surpass the score based on raw data or output of PCA in different tissues: Heart, Lung, Liver, Thymus, PBMC, Cerebellum and Pancreas. Such result is shown in Figure 20.

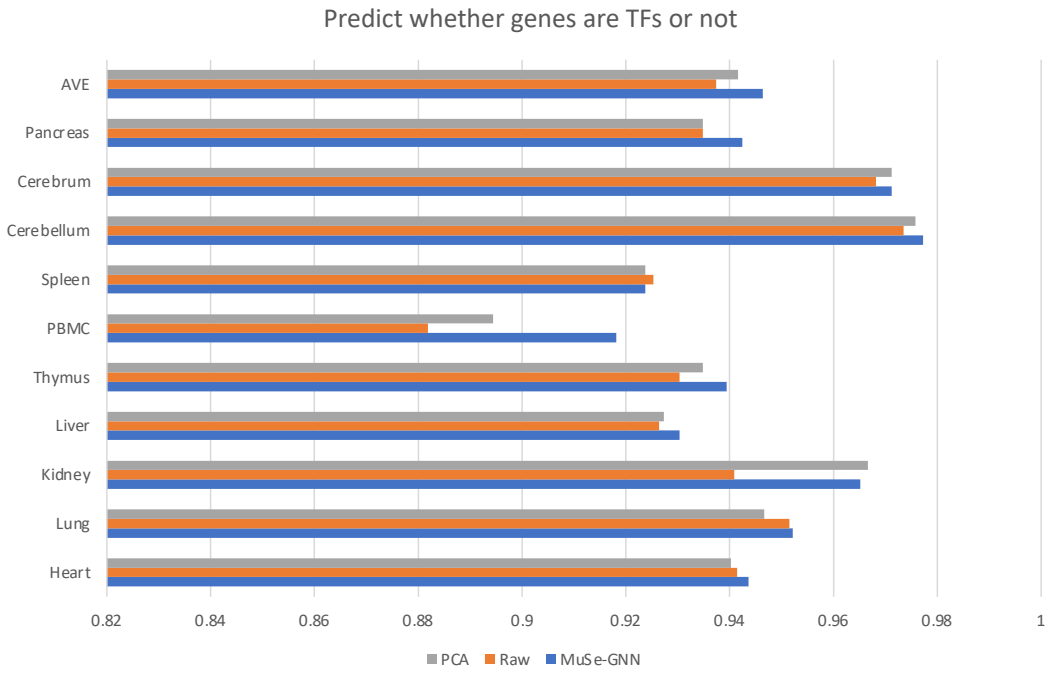

Figure 20: Accuracy for Gene-TF prediction across different tissues.

M Datasets information

Tissue	Datatype	Link
Heart	scRNA-seq	https://www.heartcellatlas.org/ https://data.mendeley.com/datasets/mbvhhf8m62/2
Lung	scRNA-seq	https://beta.fastgenomics.org/datasets/detail-dataset-427f1eee6dd44f50bae1ab13f0f3c6a9
Kidney	scRNA-seq	https://www.kidneycellatlas.org/
Liver, Thymus, Spleen	scRNA-seq	https://singlecell.broadinstitute.org/single_cell/study/SCP1845/cross-tissue-immune-cell-analysis-reveals-tissue-specific-features-in-humans?scpbr=human-cell-atlas-main-collection#study-download https://descartes.brotmanbaty.org/bbi/human-gene-expression-during-development/
Cerebrum, Cerebellum	scRNA-seq	https://descartes.brotmanbaty.org/bbi/human-gene-expression-during-development
PBMC (with COVID)	scRNA-seq	https://www.covid19cellatlas.org/index.patient.html PBMCs
Lung (with Cancer)	scRNA-seq	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE196303
Pancreas	scRNA-seq	https://azimuth.hubmapconsortium.org/references/#Human%20-%20Pancreas
Heart	scATAC-seq	https://portal.hubmapconsortium.org/browse/dataset/ba41e71358136f6a202114681a217a95 https://portal.hubmapconsortium.org/browse/dataset/d180976b42a9ed8b5fb12a508f79a238 https://portal.hubmapconsortium.org/browse/dataset/d39a1b1f40bbab8990078d343b332cdc https://descartes.brotmanbaty.org/bbi/human-gene-expression-during-development/
Lung, Kidney, Thymus, Spleen, Cerebrum, Cerebellum, Pancreas	scATAC-seq	https://descartes.brotmanbaty.org/bbi/human-gene-expression-during-development/
Heart	Spatial Transcriptomics	https://support.10xgenomics.com/spatial-gene-expression/datasets/1.1.0/V1_Human_Heart
Cerebrum	Spatial Transcriptomics	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE205055

Figure 21: Information about multimodal biological datasets from Human.

Tissue	Datatype	Link
Heart, Lung, Kidney, Liver, Thymus, Spleen	scRNA-seq	https://figshare.com/articles/dataset/Tabula_Microcebus_v1_0/14468196?file=31777475 https://figshare.com/articles/dataset/Single-cell_RNA-seq_data_from_microfluidic_emulsion_v2_/5968960/2

Figure 22: Information about multimodal biological datasets from Lemur and Mouse.

The statistics of different graphs can be found in Supplementary File 1.