

A Detail Proof of Theorem 1

Since the loss function of contrastive learning (1) is fomulated as the KL divergence between the joint distributions $p(x, y)$ and $q(x, y)$, under the assumption that $p(x)$ and $q(x)$ follow uniform distributions. Then, the following derived for $\text{Loss}_{\text{I-CON}}$:

$$\begin{aligned}
\text{Loss}_{\text{SupGCL}} &= \frac{1}{|V||\mathcal{K}|} \sum_{i \in V, j \in \mathcal{K}} D_{\text{KL}}(p_{\phi}(j, b|i, a)|q_{\phi}(j, b|i, a)). \\
&= D_{\text{KL}}(p_{\phi}(i, j, a, b)|q_{\phi}(i, j, a, b)) \\
&= E_{(a,b) \sim p_{\phi}(a,b)} \left[D_{\text{KL}}(p_{\phi}(i, j|a, b)|q_{\phi}(i, j|a, b)) \right] + D_{\text{KL}}(p_{\phi}(a, b)|q_{\phi}(a, b)) \\
&= E_{(a,b) \sim p_{\phi}(a,b)} \left[D_{\text{KL}}(p(i, j)|q_{\phi}(i, j|a, b)) \right] + D_{\text{KL}}(p_{\phi}(a, b)|q_{\phi}(a, b)) \\
&= E_{a,b \sim p_{\phi}(b|a) \cup_{\mathcal{K}}(a)} [\text{Loss}_{\text{node}}^{a,b}] + \text{Loss}_{\text{Aug}}
\end{aligned}$$

The derivation from the second to the third line utilizes the basic decomposition of KL divergence: $D_{\text{KL}}(p(x, y)|q(x, y)) = E_{x \sim p(x)} [D_{\text{KL}}(p(y|x)|q(y|x))] + D_{\text{KL}}(p(x)|q(x))$.

B Algorithm

The learning algorithm of this research is presented in Algorithm 1. We train the Graph Neural Network (GNN) f_{ϕ} using the target GRN dataset for all patients, $\mathcal{G}_{\text{all}} = \{\mathcal{G}^{(i)}\}_{i=1}^N$, and the teacher GRN dataset, $\mathcal{H}_{\text{all}} = \{\mathcal{H}_a^{(i)}\}_{i \in \mathcal{I}_a, a \in \mathcal{K}}$. Here, \mathcal{I}_a is the set of indices for teacher GRNs, $\{\mathcal{H}_a^{(i)}\}_{i \in \mathcal{I}_a}$, which correspond to data augmentations for the a -th node.

Our algorithm follows a standard training loop, consisting of the calculation of $\text{Loss}_{\text{SupGCL}}$ and the optimization of f_{ϕ} using AdamW. Furthermore, to reduce computational costs, we employ sampling-based estimation of the normalization constant for the calculation of softmax functions $p_{\phi}(b|a)$ and $q_{\phi}(b|a)$, and use importance sampling for the calculation of $\text{Loss}_{\text{node}}^{a,b}$ and Loss_{Aug} .

Algorithm 1 Training loop of SupGCL:

Require: Graph Neural Net f_{ϕ} , all patient GRNs $\mathcal{G}_{\text{all}} = \{\mathcal{G}^{(i)}\}_{i=1}^N$, all teacher GRNs $\mathcal{H}_{\text{all}} = \{\mathcal{H}_a^{(i)}\}_{i \in \mathcal{I}_a, a \in \mathcal{K}}$

- 1: **for** $\mathcal{G} \subset \mathcal{G}_{\text{all}}$ **do**
- 2: $a, b \sim \mathcal{U}_{\mathcal{K}}$
- 3: $\mathcal{G}_a, \mathcal{G}_b \leftarrow$ The a -th and b -th artificial augmentation of \mathcal{G}
- 4: $\mathcal{H}_a, \mathcal{H}_b \leftarrow$ Pick up the a -th and b -th knockdown teacher GRN from \mathcal{H}_{all}
- 5: $Z^a, Z^b \leftarrow f_{\phi}(\mathcal{G}_a), f_{\phi}(\mathcal{G}_b)$ ▷ Embedding target GRNs
- 6: $Y^a, Y^b \leftarrow f_{\phi}(\mathcal{H}_a), f_{\phi}(\mathcal{H}_b)$ ▷ Embedding teacher GRNs
- 7: $q_{\phi}(b|a) \leftarrow \text{softmax} \left(\left[\frac{\langle Z^a, Z^* \rangle_F}{\tau_a} \right] \right) [b]$ ▷ Calculate augmentation-level target model
- 8: $p_{\phi}(b|a) \leftarrow \text{softmax} \left(\left[\frac{\langle Y^a, Y^* \rangle_F}{\tau_a} \right] \right) [b]$ ▷ Calculate augmentation-level teacher model
- 9: $\text{Loss}_{\text{Aug}} \leftarrow |\mathcal{K}| p_{\phi}(b|a) (\log p_{\phi}(b|a) - \log q_{\phi}(b|a))$ ▷ Importance sampling of $\text{Loss}_{\text{Aug}}^{a,b}$
- 10: $q_{\phi}(j|i, a, b) \leftarrow \text{softmax} \left(\left[\frac{\langle z_i^a, z_*^b \rangle}{\tau_n} \right] \right) [j]$ ▷ Calculate node-level target model
- 11: $\text{Loss}_{\text{node}}^{a,b} \leftarrow \frac{1}{|V|} \sum_{i \in V} \log q_{\phi}(i|i, a, b)$
- 12: $\text{Loss}_{\text{SupGCL}} \leftarrow |\mathcal{K}| p_{\phi}(b|a) \text{Loss}_{\text{node}}^{a,b} + \text{Loss}_{\text{Aug}}$ ▷ Importance sampling of $\text{Loss}_{\text{node}}^{a,b}$
- 13: Update f_{ϕ} using $\text{Loss}_{\text{SupGCL}}$ and AdamW optimizer
- 14: **end for**
- 15: **return** Trained Graph NN: f_{ϕ}

C Details of Experimental Datasets

This section provides details on data acquisition and preprocessing procedures.

C.1 Details on the TCGA Dataset

In this study, we utilized the TCGA TARGET GTEx platform provided by UCSC Xena to extract patient-specific expression data and clinical labels (survival time data, disease subtype labels) for downstream tasks.

The TCGA platform contains four datasets: the gene expression dataset "dataset: gene expression RNAseq – RSEM norm_count," the dataset for cancer type attributes of individual patients "dataset: phenotype – TCGA TARGET GTEx selected phenotypes," the patient prognosis dataset "dataset: phenotype – TCGA survival data," and the patient phenotype data "dataset: phenotype - Phenotypes."

In this study, based on the dataset of patient cancer type attributes, we extracted patient IDs corresponding to the TCGA cohorts listed in Table 4, and subsequently obtained the associated gene expression data. Notably, the study population was limited to patients whose target cancer was a primary tumor. Additionally, overall survival time (OS.time) and survival status (OS: alive = 0 / deceased = 1) were retrieved from the patient prognosis dataset. For breast cancer specifically, disease subtype labels based on the PAM50 classification were acquired from the patient phenotype dataset. PAM50 classification using RNA expression data was feasible for all samples, assigning all 1,092 breast cancer patients to one of five subtypes: Luminal A (N = 438), Luminal B (N = 311), Basal (N = 196), Her2 (N = 111), and Normal (N = 36). Patient samples with missing values were excluded during the data extraction process. The final sample sizes and mortality rates for each cancer type after processing are summarized in Table 4.

Table 4: Sample extraction conditions and survival data statistics for each cancer type

Cancer Type	TCGA Cohort Name	Number of samples with valid survival time data	Number of deaths	Mortality rate (%)
Breast cancer	Breast Invasive Carcinoma	1,090	151	13.9
Lung cancer	Lung Adenocarcinoma + Lung Squamous Cell Carcinoma	996	394	39.6
Colon cancer	Colon Adenocarcinoma	286	69	24.1

C.2 Details on the LINCS Dataset

In this study, we used the Level 3 normalized gene expression data (filename on LINCS datasets: "GSE92742_Broad_LINCS_Level3_INF_mlr12k_n1319138x12328.gctx.gz") provided from the GEO dataset (GSE92742) of the LINCS L1000 project. Additionally, by referring to the concurrently provided experimental metadata ("GSE92742_Broad_LINCS_inst_info.txt.gz" indicating cell lines and treatment conditions, and "GSE92742_Broad_LINCS_pert_info.txt.gz" indicating drug and gene knockdown information), we extracted only the shRNA-mediated knockdown experiment groups. The cell lines and treatment durations were limited to samples from: MCF7 breast cancer cell line treated for 96 hours, A549 lung cancer cell line treated for 96 hours, and HT29 colon cancer cell line treated for 96 hours. The expression data was limited to 978 landmark genes.

C.3 Extraction of Gene Label Data for BP/CC Tasks using Gene Ontology

In this study, for Biological Process (BP) classification and Cellular Component (CC) classification in downstream tasks, we performed multi-label annotation for each of the 975 genes constituting the GRN, based on terms obtained from the GO database. For BP labels, three categories whose importance is known were used: 'Metabolism', 'Signal Transduction', and 'Cellular Organization' [31]. Similarly, for CC labels, four categories were used: 'Nucleus', 'Mitochondrion', 'Endoplasmic Reticulum', and 'Plasma Membrane' [32]. These categories were selected to cover the major functions of the GRN.

To extract these multi-labels, the following steps were performed:

1. Batch retrieve terms associated with each gene using the MyGene.info API [33] and the OBO file (available at <https://geneontology.org/docs/download-ontology/>).
2. Aggregate multi-labels for the target genes using the GOOtools library [34].

Genes that could not be labeled into any of the BP or CC categories were excluded from the downstream tasks in this study. The number and percentage of genes included in each category are shown in Table 5.

Table 5: Gene label distribution for high-level BP/CC categories (n = 975)

Category	Number of genes	Percentage of category (%)
BP: Metabolism	533	54.67
BP: Signal Transduction	261	26.77
BP: Cellular Organization	369	37.85
BP: Not Applicable	204	20.92
CC: Nucleus	388	39.80
CC: Mitochondrion	151	12.49
CC: Endoplasmic Reticulum	148	15.18
CC: Plasma Membrane	231	23.69
CC: Not Applicable	257	26.36

C.3.1 Annotation by OncoKB

For cancer-related gene classification in downstream tasks, cancer-related genes were obtained from the OncoKBTM Cancer Gene List provided by OncoKB (Oncology Knowledge Base) [25]. Using the list of 1188 cancer-related genes provided as of April 30, 2025, positive labels were assigned to the 975 genes used in this study. As a result, 106 genes were labeled as cancer-related genes.

D Estimating Gene Regulatory Networks

D.1 Taxonomy of Gene Regulatory Network Estimation

Estimating accurate gene regulatory networks (GRNs) is crucial for elucidating cellular processes and disease mechanisms. Methods for computationally estimating GRNs from gene expression data can be categorized into correlation-based [35], mutual information-based [36], probabilistic graphical models-based such as Bayesian networks [37], and deep learning-based approaches [38]. The number of experimentally validated GRNs is limited. Therefore, GRN estimation methods need the ability to account for measurement errors and appropriately capture non-linear and multimodal interactions between genes while mitigating overfitting. Thus, we adopted a method combining Bayesian network estimation using multiple sampling and non-parametric regression [26] to identify patient- or sample-specific GRNs [8].

D.1.1 Details of Estimating GRNs

To construct patient-specific GRNs, we estimate Bayesian Networks using B-spline regression. First, we estimate the conditional probability density functions between genes using the entire gene expression dataset. Then, using these learned parameters, we construct patient- or sample-specific GRNs.

Let x_1, x_2, \dots, x_n be random variables for n nodes, and let $\text{pa}(i)$ be the set of parent nodes of the i -th node. In this case, a Bayesian network using B-spline curves [26] is defined as a probabilistic model decomposed into conditional distributions with parent nodes:

$$\begin{aligned}
 p(x_1, \dots, x_n) &= \prod_{i=1}^n p(x_i | x_{\text{pa}(i)}) \\
 &= \prod_{i=1}^n \mathcal{N}\left(x_i \mid \sum_{j \in \text{pa}(i)} m_{ij}(x_j), \sigma^2\right).
 \end{aligned}$$

Here, \mathcal{N} is a Gaussian distribution, and m_{ij} is a B-spline curve defined by B-spline basis functions $b_s : \mathbb{R} \rightarrow \mathbb{R}$ as

$$m_{ij}(x_j) = \sum_{s=1}^M w_{i,j}^s b_s(x_j) \quad (10)$$

The Bayesian network is estimated by learning the relationships with parent nodes based on the model described above and the parameters of the conditional distributions. In this study, the score function used for searching the structure of the Bayesian Network can be analytically derived using Laplace approximation [26]. Since the problem of finding a Directed Acyclic Graph (DAG) that maximizes this score is NP-hard, we performed structure search using a heuristic structure estimation algorithm, the greedy hill-climbing (HC) algorithm [39]. Furthermore, to ensure the reliability of the estimation results by the HC algorithm, we performed multiple sampling runs.

We extracted edges that appeared more frequently than a predefined threshold relative to the number of sampling runs in the estimated networks. Finally, for each edge in the obtained network structure, the conditional probabilities were relearned using all input data.

Using the network and conditional probabilities learned here, we derive patient-specific GRNs [27]. The node set and edge set of the graph for a patient-specific GRN are defined by the network learned with gene expression levels as random variables x_1, \dots, x_n . Furthermore, the feature of the i -th node X_i^y is the gene expression level of each sample, and the feature of an edge from i to j (designated as the k -th edge) is the realization of the learned B-spline curve $X_k^e = m_{ij}(x_j)$. Patient-specific networks using such edge features have led to the discovery of subtypes that correlate more strongly with prognosis than existing subtypes [8]. Additionally, using differences in edge features between patients to extract patient-specific networks has been reported to contribute to the identification of novel diagnostic and therapeutic marker candidates in diseases such as idiopathic pulmonary fibrosis [40] and chronic nonbacterial osteomyelitis [41].

For GRN estimation, we used INGOR (version 0.19.0), a software that estimates Bayesian Networks based on B-spline regression, and executed it on the supercomputer Fugaku. INGOR is based on SiGN-BN [42], a software that similarly estimates Bayesian Networks using B-spline regression, and achieves faster estimation by optimizing parallel computation on Fugaku. In all GRN estimations, the number of sampling runs was set to 1000, and the threshold for adopting edges was set to 0.05. Since network estimation with a large amount of sample data can lead to Out of Memory errors, the upper limit of gene expression data used for network estimation was set to 3000 (especially for LINC data). All other hyperparameters related to network estimation used the default settings of SiGN-BN. The number of Fugaku nodes and the required execution times are shown in Table 6. In addition, the number of parallel threads was set to 4 for estimations using TCGA data and 2 for LINC data.

Table 6: Estimated network times in supercomputer Fugaku

	Breast cancer		Lung cancer		Colorectal cancer	
	TCGA	LINC	TCGA	LINC	TCGA	LINC
Number of Fugaku Nodes	288	288	288	528	288	528
Estimated Time [hh:mm:ss]	00:54:49	08:35:12	00:38:24	13:28:07	00:05:40	21:01:38

D.2 Results

The statistics of the estimated networks are shown in Table 7. It should be noted that network estimation methods using sampling can extract highly reliable edges, but they may occasionally extract structures containing cyclic edges. However, all networks estimated in this study maintained a DAG structure.

Table 7: Estimated network statistics

	Breast cancer		Lung cancer		Colorectal cancer	
	TCGA	LINCS	TCGA	LINCS	TCGA	LINCS
Number of Nodes	975	975	975	975	975	975
Number of Edges	13170	10498	13322	13968	13686	12541
Average Degree	13.5077	10.7671	13.6636	14.3262	14.0369	12.8626

E Experimental Setting

E.1 Hyperparameters Settings

In this study, we used Optuna [29], a Bayesian optimization tool, for hyperparameter search in pre-training. The search space for all models included the AdamW learning rate $lr \in [10^{-5}, 10^{-3}]$, batch size $batch_size \in \{4, 8\}$, and model-specific hyperparameters. Model-specific hyperparameters were the temperature parameter $\tau \in \{0.25, 0.5, 0.75, 1.0\}$ for GraphCL, GRACE, and SupGCL, and the global-hop parameter $k \in \{1, 2, 3\}$ for SGRL. The graph embedding dimension was unified to 64 for all models. See Appendix F for a discussion on embedding dimension. For training the pre-trained models, the data was split into training and validation sets at an 8 : 2 ratio, and the validation loss was used as the metric for various decisions. The optimal hyperparameters determined by actual hyperparameter tuning are shown in Table 8.

Table 8: Hyperparameter settings for each cancer type

Cancer Type	Model	learning rate	batch size	temperature	global hop
Breast cancer	GAE	5.74×10^{-4}	4	—	—
	GRACE	9.71×10^{-4}	4	0.25	—
	GraphCL	1.23×10^{-4}	4	0.25	—
	SGRL	2.39×10^{-4}	4	—	3
	SupGCL	2.37×10^{-4}	4	0.25	—
Lung cancer	GAE	2.16×10^{-4}	4	—	—
	GRACE	4.44×10^{-5}	8	0.25	—
	GraphCL	6.24×10^{-5}	4	0.25	—
	SGRL	8.18×10^{-5}	8	—	2
	SupGCL	1.89×10^{-4}	4	0.25	—
Colorectal cancer	GAE	2.26×10^{-4}	4	—	—
	GRACE	4.03×10^{-5}	8	0.25	—
	GraphCL	8.83×10^{-5}	4	0.25	—
	SGRL	7.10×10^{-4}	4	—	3
	SupGCL	3.32×10^{-4}	4	0.25	—

E.2 Computational Environment and Computation Time

All experiments were conducted on an NVIDIA H100 SXM5 (95.83 GiB), and the computation time for each model is shown in Table 9. Please note that although the number of training steps is based on 3000 epochs, the actual training time varies due to early stopping using the validation data.

E.3 Fine-tuning Settings

E.3.1 Graph-Level Task

For fine-tuning graph-level tasks (Hazard Prediction, Subtype Classification), training was performed on a per-patient basis. The latent states embedded by the Graph Neural Network were transformed into graph-level embeddings using mean-pooling, and then fed through a 2-layer MLP to train task-specific models.

Table 9: Pre-training computation time for each cancer type

Cancer Type	Model	Computation Time	Epochs
Breast cancer	GAE	3.354 hr	3000
	GRACE	10.77 hr	3000
	GraphCL	8.306 hr	2000
	SGRL	5.859 hr	2000
	SupGCL	21.40 hr	1500
Lung cancer	GAE	1.875 hr	1800
	GRACE	9.960 hr	3000
	GraphCL	3.303 hr	1300
	SGRL	7.485 hr	3000
	SupGCL	19.67 hr	1500
Colorectal cancer	GAE	45.23 min	2500
	GRACE	2.839 hr	3000
	GraphCL	1.476 hr	2000
	SGRL	53.14 min	1100
	SupGCL	9.551 hr	2500

We employed 10-fold cross-validation across patients to generate training/test datasets. Fine-tuning was performed using AdamW with a learning rate of 1×10^{-3} . For evaluation, we reported the mean and standard deviation of the scores across all folds.

Hazard Prediction : For the hazard prediction task, we adopted the classic Cox proportional hazards model. In the Cox model, the hazard function for a patient at time t is defined as

$$h(t | x) = h_0(t) \exp(\beta^\top x)$$

Here, $h_0(t)$ is the baseline hazard, x is the input variable, and β is the regression coefficient to be learned. The prognosis estimation is performed by connecting this input variable x to the graph NN and its head.

This study performed training using partial likelihood maximization based on patient prognosis information and evaluated performance using the C-index.

Subtype Classification: For the subtype classification task, we created a classification model using a 5-class softmax function and trained it using multi-class cross-entropy. Furthermore, performance was evaluated using Accuracy and Macro F1-score. The F1-score results are shown in Appendix [F](#)

E.3.2 Node-Level Task

For fine-tuning node-level tasks (BP/CC Classification, Cancer Rel. Classification), tasks were solved on a per-gene basis. For the latent state of each node embedded by the Graph Neural Network, task-specific models were learned through a 2-layer MLP.

In BP/CC Classification, performance was evaluated using gene-wise 10-fold cross-validation. For Cancer Rel. Classification, it is necessary to mitigate class imbalance in positive and negative label data. To achieve this, we prepared a dataset by undersampling the negative label data, split it into training and test data at an 8:2 ratio, and performed fine-tuning and accuracy evaluation. This undersampling and data splitting process was repeated 10 times with different seeds to evaluate the performance on this task.

For optimization, AdamW was used with a batch size of 8 and a learning rate of 1×10^{-3} . For evaluation, we reported the mean and standard deviation of the scores for each fold.

BP/CC. Classification : In Biological Process (BP) classification, three categories for each gene—"metabolism," "signal transduction," and "cellular organization"—are predicted as a multi-hot vector. In Cellular Component (CC) classification, four categories—"nucleus," "mitochondria," "endoplasmic reticulum," and "plasma membrane"—are predicted as a multi-hot vector. The model was structured using a sigmoid function for each category, and training was performed using binary

cross-entropy for each respective category. Performance was evaluated using Subset Accuracy, Macro F1-score, and Jaccard Index as evaluation metrics. The Macro F1-score and Jaccard Index results are shown in Appendix F.

Cancer Rel. Classification In cancer-related gene classification, 106 genes defined as positive by OncoKB were labeled as "positive," and all other genes were labeled as "negative" for binary classification. A model was created to estimate negative and positive cases using a sigmoid function, and training was performed using binary cross-entropy. Performance was evaluated using Accuracy and F1-score as evaluation metrics. The F1-score results are shown in Appendix F.

F Additional Results

As additional experimental results, we performed the following three analyses:

1. Performance evaluation of the proposed and existing methods using various evaluation metrics.
2. Visualization of the latent states of pre-trained models.
3. Performance comparison with varying embedding dimensions.

F.1 Additional Evaluation Metrics

In the main paper, we presented results using only Accuracy (or subset accuracy). Below, we report the results using other metrics for the same tasks.

Tables 10 and 11 show the Macro F1-score and Jaccard index results for node-level tasks. Please note that for the cancer-related classification task, we evaluate only the F1-score because it involves binary data. Additionally, Table 12 shows the Macro F1-score for subtype classification in breast cancer. While our proposed method, SupGCL, did not individually achieve state-of-the-art results across all tasks and metrics, it demonstrated the most balanced performance overall.

Table 10: Node-level downstream task: macro F1-score

Task	w/o-pretrain	GAE	GraphCL	GRACE	SGRL	SupGCL
BP.						
Breast	0.553±0.024	0.551±0.034	0.540±0.045	<u>0.558±0.022</u>	0.543±0.022	0.571±0.025
Lung	0.538±0.039	0.546±0.021	0.584±0.065	<u>0.555±0.026</u>	0.549±0.023	0.546±0.031
Colorectal	0.514±0.053	0.550±0.025	0.516±0.033	0.560±0.042	<u>0.560±0.040</u>	0.547±0.038
CC.						
Breast	<u>0.404±0.036</u>	0.378±0.021	0.336±0.018	0.362±0.040	0.384±0.037	0.418±0.024
Lung	0.349±0.086	0.395±0.023	0.376±0.072	<u>0.393±0.026</u>	0.385±0.026	0.387±0.028
Colorectal	0.288±0.060	0.403±0.032	0.265±0.029	0.372±0.047	<u>0.401±0.049</u>	0.397±0.030
Rel.						
Breast	0.523±0.094	0.571±0.048	<u>0.593±0.072</u>	0.591±0.038	0.578±0.067	0.610±0.070
Lung	0.507±0.117	0.559±0.045	0.538±0.236	0.535±0.139	<u>0.575±0.061</u>	0.592±0.067
Colorectal	0.474±0.242	<u>0.582±0.081</u>	0.556±0.124	0.547±0.197	0.569±0.145	0.596±0.060

F.2 Additional Latent Space Analysis

Node-level embedding of other cancers: Previously, in Result 2: Latent Space Analysis, we visualized the embedding space generated by pre-trained models on the breast cancer dataset (Figure 3). Analogous results for the lung cancer and colorectal cancer datasets are presented in Figure 4. Since subtype data were unavailable for the lung and colorectal cancer datasets, only node-level latent space visualizations are presented for these cancers. These results confirm that both GRACE and our proposed method, SupGCL, yield stable latent representations for these cancers, with no observed latent space collapse.

Table 11: Node-level downstream task: Jaccard index

Task	w/o-pretrain	GAE	GraphCL	GRACE	SGRL	SupGCL
BP.						
Breast	<u>0.490±0.017</u>	0.487±0.028	0.454±0.046	0.478±0.037	0.468±0.028	0.500±0.035
Lung	0.539±0.030	0.494±0.034	0.484±0.030	0.510±0.051	0.479±0.019	<u>0.518±0.027</u>
Colorectal	0.537±0.031	0.506±0.019	0.500±0.036	<u>0.514±0.024</u>	0.469±0.030	0.502±0.022
CC.						
Breast	<u>0.402±0.052</u>	0.378±0.021	0.303±0.028	0.359±0.028	0.377±0.029	0.422±0.028
Lung	<u>0.387±0.040</u>	0.382±0.035	0.321±0.062	0.384±0.036	0.376±0.031	0.392±0.034
Colorectal	0.377±0.055	0.379±0.036	0.308±0.067	<u>0.388±0.036</u>	0.360±0.053	0.395±0.033

Table 12: Macro F1-score for subtype classification

Task	w/o-pretrain	GAE	GraphCL	GRACE	SGRL	SupGCL
Subtype						
Breast	0.626 ± 0.070	0.720 ± 0.057	0.552 ± 0.089	<u>0.761 ± 0.063</u>	0.715 ± 0.064	0.785 ± 0.056

Analysis of Latent Space Collapse: To further investigate the characteristics of the node-level latent spaces presented in [Result 2: Latent Space Analysis](#), we employed Principal Component Analysis (PCA). Figure [5](#) displays the PCA-projected latent spaces from pre-trained models on the breast cancer dataset, along with their corresponding explained variance ratios. For GraphCL, which previously exhibited tendencies towards latent space collapse, this analysis confirmed that its PCA explained variance ratio was overwhelmingly concentrated in the first principal component (PC1), accounting for 98.3%.

F.3 Performance Evaluation across Different Embedding Dimensions

Finally, we investigated the effect of varying embedding dimensions on performance. Figure [6](#) presents the performance metrics and their corresponding standard deviations across 13 tasks for embedding dimensions of {8, 16, 32, 64}. Excluding GraphCL, which exhibited instability in generating stable latent spaces, the other five methods showed only marginal performance gains when the embedding dimension was increased from 32 to 64. Furthermore, the proposed method consistently achieved high performance across all tasks and embedding dimensions, experimentally demonstrating its superiority over existing representation learning approaches for biological downstream tasks.

Appendix References

- [31] Anna R Paolacci, Oronzio A Tanzarella, Enrico Porceddu, and Mario Ciaffi. Identification and validation of reference genes for quantitative rt-pcr normalization in wheat. *BMC molecular biology*, 10:1–27, 2009.
- [32] Pedro R Costa, Marcio L Acencio, and Ney Lemke. A machine learning approach for genome-wide prediction of morbid and druggable human genes based on systems-level data. In *BMC genomics*, volume 11, pages 1–15. Springer, 2010.
- [33] Jiwen Xin, Adam Mark, Cyrus Afrasiabi, Ginger Tsueng, Moritz Juchler, Nikhil Gopal, Gregory S Stupp, Timothy E Putman, Benjamin J Ainscough, Obi L Griffith, et al. Mygene. info and myvariant. info: gene and variant annotation query services. *bioRxiv*, page 035667, 2015.
- [34] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.
- [35] Peter Langfelder and Steve Horvath. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics*, 9(1):559, December 2008.
- [36] Adam A. Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics*, 7(1):S7, March 2006.

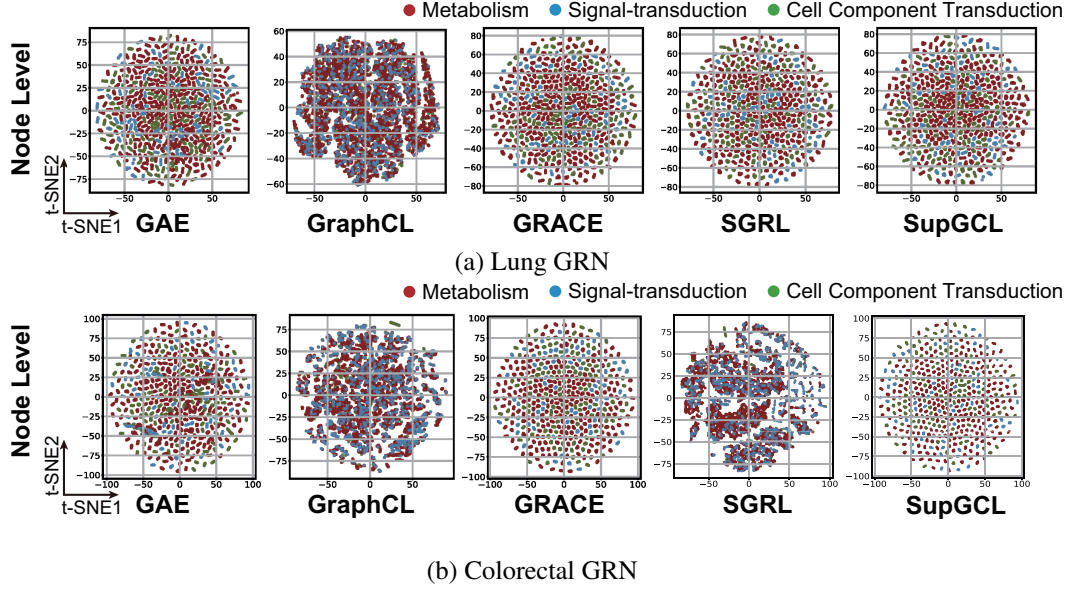


Figure 4: t-SNE visualization of pre-trained embeddings on lung and colorectal cancer GRNs.

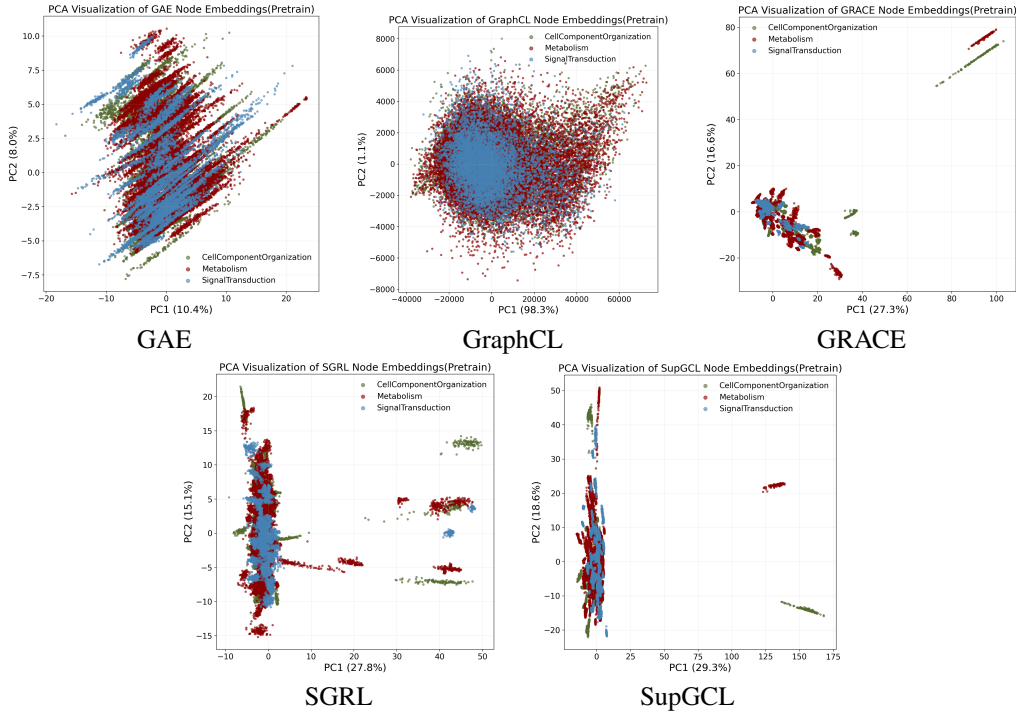


Figure 5: PCA analysis of the latent spaces of pre-trained models.

- [37] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using Bayesian networks to analyze expression data. In *Proceedings of the fourth annual international conference on Computational molecular biology*, RECOMB '00, pages 127–135, New York, NY, USA, 2000. Association for Computing Machinery.
- [38] Hantao Shu, Jingtian Zhou, Qiuyu Lian, Han Li, Dan Zhao, Jianyang Zeng, and Jianzhu Ma. Modeling gene regulatory networks using neural network architectures. *Nature Computational Science*, 1(7):491–501, July 2021. Publisher: Nature Publishing Group.
- [39] Seiya Imoto, Sunyong Kim, Takao Goto, Sachiyo Aburatani, Kousuke Tashiro, Satoru Kuhara, and Satoru Miyano. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic

- network. *Journal of bioinformatics and computational biology*, 1(02):231–252, 2003.
- [40] Mei Tomoto, Yohei Mineharu, Noriaki Sato, Yoshinori Tamada, Mari Nogami-Itoh, Masataka Kuroda, Jun Adachi, Yoshito Takeda, Kenji Mizuguchi, Atsushi Kumanogoh, Yayoi Natsume-Kitatani, and Yasushi Okuno. Idiopathic pulmonary fibrosis-specific Bayesian network integrating extracellular vesicle proteome and clinical information. *Scientific Reports*, 14(1):1315, January 2024. Publisher: Nature Publishing Group.
 - [41] Hiroko Yahara, Souichi Yanamoto, Miho Takahashi, Yuji Hamada, Haruo Sakamoto, Takuya Asaka, Yoshimasa Kitagawa, Kuniyasu Moridera, Kazuma Noguchi, Masaya Sugiyama, Yutaka Maruoka, and Koji Yahara. Whole blood transcriptome profiling identifies gene expression subnetworks and a key gene characteristic of the rare type of osteomyelitis. *Biochemistry and Biophysics Reports*, 32:101328, December 2022.
 - [42] Yoshinori Tamada, Teppei Shimamura, Rui Yamaguchi, Seiya Imoto, Masao Nagasaki, and Satoru Miyano. Sign: Large-Scale Gene Network Estimation Environment for High Performance Computing. *Genome Informatics*, 25(1):40–52, 2011.

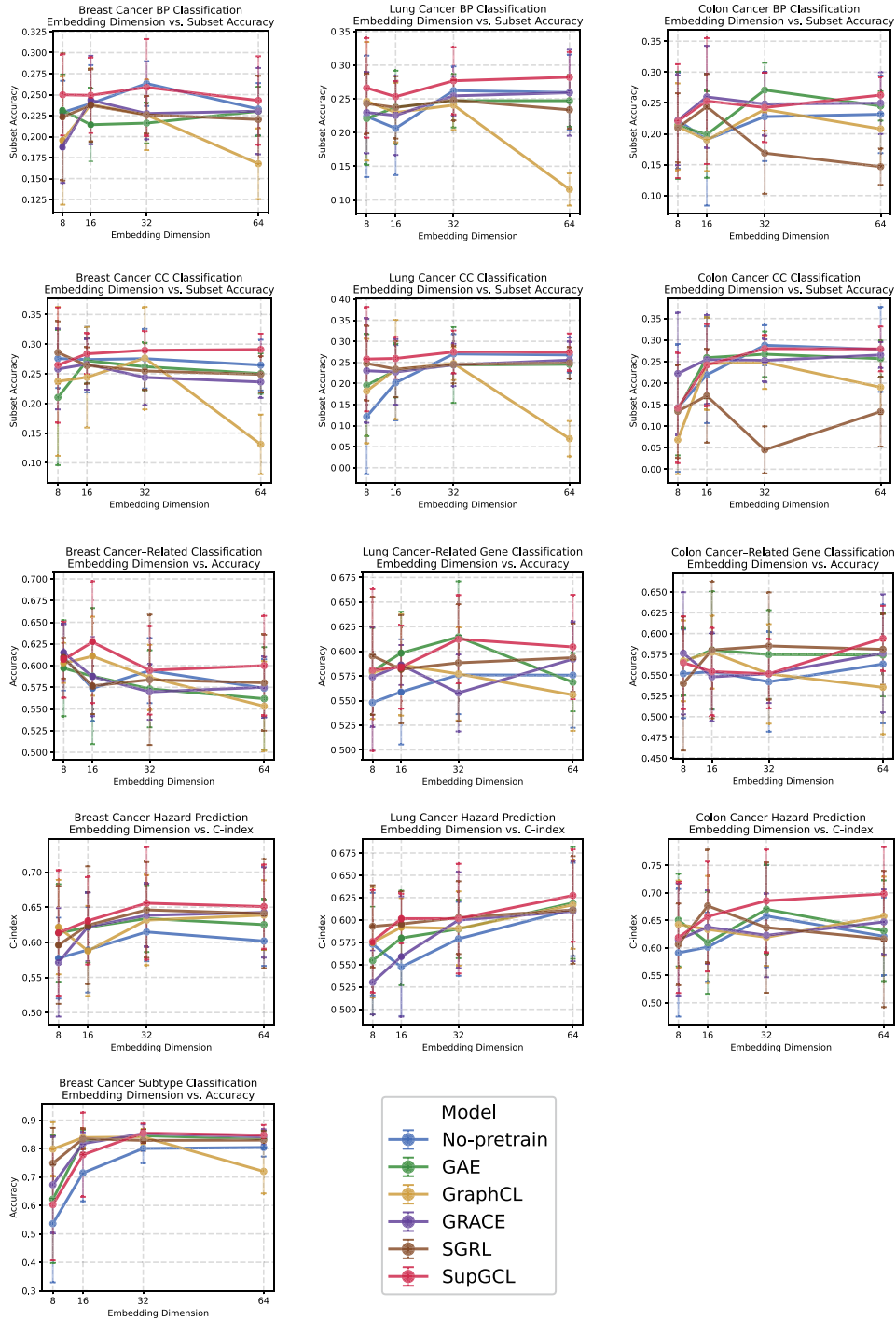


Figure 6: Embedding dimension analysis. This figure shows the performance changes across 13 tasks as the embedding dimension varies. The left column shows the results for breast cancer, the center column for lung cancer, and the right column for colorectal cancer. The first row presents the subset accuracy of BP classification, the second row shows the subset accuracy of CC classification, the third row displays the accuracy of cancer-related gene classification, the fourth row indicates the C-index for hazard prediction, and the fifth row shows the results of subtype classification.