SUPERVISED GRAPH CONTRASTIVE LEARNING FOR GENE REGULATORY NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Contrastive Learning (GCL) is a powerful self-supervised learning framework that performs data augmentation through graph perturbations, with growing applications in the analysis of biological networks such as Gene Regulatory Networks (GRNs). The artificial perturbations commonly used in GCL, such as node dropping, induce structural changes that can diverge from biological reality. This concern has contributed to a broader trend in graph representation learning toward augmentation-free methods, which view such structural changes as problematic and to be avoided. However, this trend overlooks the fundamental insight that structural changes from biologically meaningful perturbations are not a problem to be avoided but a rich source of information, thereby ignoring the valuable opportunity to leverage data from real biological experiments. Motivated by this insight, we propose SupGCL (Supervised Graph Contrastive Learning), a new GCL method for GRNs that directly incorporates biological perturbations from gene knockdown experiments as supervision. SupGCL is a probabilistic formulation that continuously generalizes conventional GCL, linking artificial augmentations with real perturbations measured in knockdown experiments and using the latter as explicit supervisory signals. To assess effectiveness, we train GRN representations with SupGCL and evaluate their performance on downstream tasks. The evaluation includes both node-level tasks, such as gene function classification, and graph-level tasks on patient-specific GRNs, such as patient survival hazard prediction. Across 13 tasks built from GRN datasets derived from patients with three cancer types, SupGCL consistently outperforms state-of-the-art baselines.

1 INTRODUCTION

Graph representation learning has recently attracted attention in various fields to learn a meaningful latent space to represent the connectivity and attributes in given graphs (Ju et al., 2024). Applications of graph representation learning are advancing in numerous areas where network data exists, such as analysis in social networks, knowledge graphs (Hu et al., 2023; Shen & Zhang, 2023), and biological network analysis in bioinformatics (Liu et al., 2023; Wu et al., 2021).

The application of graph representation learning to Gene Regulatory Networks (GRNs), which contain information about intracellular functions and processes, is particularly important in the fields of biology and drug discovery. It is expected to contribute to the identification of therapeutic targets and the elucidation of disease mechanisms. Representation learning for GRNs has been applied to tasks such as transcription factor inference (Yu et al., 2025) and predicting drug responses in cancer cell lines (Liu et al., 2022). Advances in gene expression measurement and analysis technologies have enabled the construction of patient-specific GRNs, highlighting gene regulation patterns that differ from the population as a whole (Nakazawa et al., 2021). Hereafter, this paper will refer to such individualized networks simply as GRNs.

Among the various graph representation learning methods, Graph Contrastive Learning (GCL) has emerged as a powerful self-supervised learning framework (You et al., 2020). GCL learns by maximizing the similarity between node representations across different views of the same graph generated via data augmentation, a process that assumes the preservation of essential features like graph topology (Zhu et al., 2021; Wang et al., 2024). While augmentation methods have been refined in the application of GCL to GRNs, there have been concerns that conventional artificial perturbations

like node dropping can cause structural changes so significant—disrupting the function of networks including critical nodes like master regulators—that they hinder learning (Paull et al., 2021). In response to these challenges, Augmentation-Free approaches have been developed, which perturb model parameters instead of the graph structure itself, and have shown high performance (Thakoor et al., 2021; He et al., 2024).

However, this trend overlooks a fundamental insight: that the representational shifts caused by graph augmentation are not an obstacle to overcome, but rather a rich source of information to be exploited. Consequently, it ignores the valuable opportunity to leverage data obtained from actual biological experiments, such as knockdown experiments.

To address these issues, we propose a novel supervised GCL method (SupGCL) that leverages gene knockdown perturbations within GRNs. Our method uses experimental data from actual gene knockdowns as supervision, enabling biologically faithful representation learning. In gene knockdown experiments, the suppression of specific genes causes biological perturbations, leading to the observation of a new, altered GRN. By using these perturbations as supervision signals for GCL, we can perform data augmentation that retains biological characteristics. Moreover, since our method naturally extends traditional GCL models in the direction of supervised augmentation within a probabilistic framework, conventional GCL approaches emerge as special cases of our proposed model.

To evaluate the effectiveness of the proposed SupGCL method, we apply it to GRN datasets from cancer patients across three cancer types and conduct multiple downstream tasks. For gene-level downstream tasks, we perform classification into biological process, cellular component, and cancer-related gene categories. For patient-level tasks, we conduct hazard prediction and disease subtype classification. The performance of our method is compared against existing graph representation learning techniques, including conventional GCL methods.

The main contributions of this study are as follows:

- **Proposal of a novel GRN representation learning method utilizing gene knockdown experiments:** We develop a new GCL method tailored for GRNs that incorporates gene knockdown data as supervision to enhance biological plausibility.
- **Theoretical extension of GCL:** We formulate supervised GCL, incorporating augmentation selection into a unified probabilistic modeling framework, and theoretically demonstrate that existing GCL methods are special cases of our proposed approach.
- **Empirical validation of the proposed method:** We apply the method to 13 downstream tasks on GRNs derived from real cancer patients and consistently outperform conventional approaches across all tasks.

2 Related Works

2.1 Augmentation-Free Graph Contrastive Learning (GCL)

GCL is a powerful framework for learning representations by maximizing the similarity between multiple "views" of a graph generated through data augmentation. While early node-level methods like GRACE achieved high performance (Zhu et al., 2020), graph-level approaches like GraphCL struggled with the loss of key node features (You et al., 2021; Sun et al., 2025).

A common challenge for these methods was their heavy reliance on the choice of data augmentations. This led to a trend in augmentation-free GCL, pioneered by BGRL (Thakoor et al., 2021), which used a bootstrapping mechanism to circumvent heuristic augmentations that could damage the graph's structure, influencing subsequent research such as simGRACE (Xia et al., 2022) and AF-GRL (Lee et al., 2022). SGRL (He et al., 2024) further advanced this trend by integrating concepts like feature uniformity from early GCL to prevent the representation collapse problem often faced by methods like BGRL, achieving stable and high-performance self-supervised learning. However, these methods view perturbations to the graph as a problem to be avoided, thereby overlooking the opportunity to utilize real-world supervisory signals that arise from structural changes.

2.2 Representation Learning for Gene Regulatory Networks(GRNs)

Applying Graph Neural Networks (GNNs) to known GRNs is a growing area of research for downstream tasks such as gene function classification and patient survival prediction(Zohari & Chehreghani, 2025). The prevailing approach in this field has been supervised learning without pre-training (Liu et al., 2022; Yu et al., 2025). While link prediction addresses the important task of inferring the unknown structure of GRNs (Huynh-Thu et al., 2010; Yu et al., 2025), our work focuses on learning representations from known GRN structures for downstream applications.

Recently, self-supervised methods have been introduced to this domain. Initial efforts focused on applying general augmentation-free methods, such as the Graph Autoencoder (GAE), which learns representations by reconstructing the graph's structure (Jung et al., 2024). More recently, the focus has begun to shift towards specialized GCL frameworks tailored for GRNs, although such approaches are still uncommon. A notable example is MuSe-GNN (Liu et al., 2023), which leverages different data modalities (e.g., scRNA-seq and scATAC-seq) as natural "views," avoiding the need for artificial augmentations by using biologically diverse information. In contrast to these methods, our work, SupGCL, uniquely incorporates real-world perturbations from within a single modality as supervisory signals for GCL.

3 Preliminaries

3.1 Background of Graph Contrastive Learning

Although there are various definitions of contrastive learning, it can be expressed using a probabilistic model based on KL divergence over pairs of augmentations or node instances (Alshammari et al., 2025). Let $\mathcal{X}$ denote a set of entities and let $(x, y) \in \mathcal{X} \times \mathcal{X}$ be a pair from that set. The contrastive loss is formulated as follows:

$$\text{Loss}_{\text{I-CON}} \triangleq \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} D_{\text{KL}}(p_\theta(y|x) | q_\phi(y|x)). \tag{1}$$

Here, $q_\phi(y|x)$ is the probability distribution of the learned model with parameter ϕ, and $p_\theta(y|x)$ is a reference distribution. To avoid trivial solutions when training both p_θ and q_ϕ simultaneously, the reference distribution p_θ is almost fixed. The reference model $p_\theta(y|x)$ is often designed as a probability that assigns a non-zero constant to positive pairs (x, y) and zero to negative pairs (x, y).

Graph Contrastive Learning (GCL) handles the learned model $q_\phi(j|i)$ corresponding to a pair of nodes (i, j). Consider graph operations for augmentation, order them, and represent the index of these operations by a. Let $\boldsymbol{z}_i^a \in \mathbb{R}^d$ be the graph embedding of the i-th node obtained from the Graph Neural Network under the a-th augmentation operation. For two augmentation operations (a, b), the pair of probability models $(p, q_\phi^{a,b})$ used in GCL is defined by

$$p(j|i) \triangleq \delta_{ij}, \quad q_\phi^{a,b}(j|i) \triangleq \frac{\exp\big(\text{sim}(\boldsymbol{z}_i^a, \boldsymbol{z}_j^b)/\tau_{\text{n}}\big)}{\sum_{k \in \mathcal{V}} \exp\big(\text{sim}(\boldsymbol{z}_i^a, \boldsymbol{z}_k^b)/\tau_{\text{n}}\big)}. \tag{2}$$

Here, $\mathcal{V}$ is the set of nodes in the given graph, δ_{ij} is the Dirac delta, $\tau_{\text{n}} > 0$ is a temperature parameter and $\text{sim}(\cdot, \cdot)$ denotes cosine similarity. This setting is often extended so that the definitions of $(p, q_\phi^{a,b})$ vary according to how positive and negative pairs are sampled. Note that the target model q_ϕ depends on the sampling method of augmentation operators, so the probability model also depends on (a, b).

GCL trains the model using the following loss function on the pair of probability models $(p, q_\phi^{a,b})$ induced by augmentation operations (a, b), according to the formulation of contrastive learning loss (1).

$$\text{Loss}_{\text{node}}^{a,b} \triangleq \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} D_{\text{KL}}(p(j|i) | q_\phi^{a,b}(j|i)). \tag{3}$$

This encourages the embeddings at the node level $\boldsymbol{z}_i^a$ and $\boldsymbol{z}_i^b$ of the same node under different augmentation operations to be close to each other. Typically, augmentation operations a, b are chosen by

uniform sampling from a set of candidates $\mathcal{A}$. Hence, in practice, the expected value is minimized under the uniform distribution $\mathrm{U}_{\mathcal{A}}$ over $\mathcal{A}$:

$$\mathrm{Loss}_{\mathrm{node}} \triangleq \mathbb{E}_{a,b\sim \mathrm{U}_{\mathcal{A}}}[\mathrm{Loss}^{a,b}_{\mathrm{node}}]. \tag{4}$$

While GCL achieves node-level representation learning via the procedure described above, in many cases the augmentation operations themselves rely on artificial perturbations such as randomly adding and/or deleting nodes and/or edges. In this study, we introduce gene knockdown—a biological perturbation—as supervision for these augmentation operations.

3.2 Notation and Problem Definition

In this study, we describe a GRN as a directed graph $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E}, \boldsymbol{X}^{\mathcal{V}}, \boldsymbol{X}^{\mathcal{E}})$ that contains information on nodes and edges. Here, $\mathcal{V}$, and $\mathcal{E}$ are the sets of nodes and edges, respectively, and each node represents a gene. $\boldsymbol{X}^{\mathcal{V}}_{:,i}$ is the feature of the i-th gene, and $\boldsymbol{X}^{\mathcal{E}}_{:,i}$ is the feature of the i-th edge in the network. The augmentation operation corresponding to the knockdown of the i-th gene is modeled by setting the feature of the i-th gene to zero and also setting the features of all edges connected to the i-th gene to zero.

We associate the a-th augmentation operation with the knockdown of the a-th gene. In what follows, we denote by $\mathcal{G}_a$ the graph obtained by applying the a-th augmentation operation to $\mathcal{G}$. Moreover, in this study, let $\mathcal{H}_a$ be the teacher GRN for the knockdown of the a-th gene , and let $\mathcal{K}$ be the set of all augmentation operations for which such teacher GRNs exist. In other words, $\mathcal{H}_a$ is a GRN that serves as a teacher for artificial augmentation for the a-th gene.

Our goal is to use the original GRN $\mathcal{G}$ and its teacher GRNs $\{\mathcal{H}_a\}_{a\in\mathcal{K}}$ to train a Graph Neural Network (GNN) f_ϕ. Defining embedded representations through the GNN f_ϕ as

$$\boldsymbol{Z}^a \triangleq f_\phi(\mathcal{G}_a) \in \mathbb{R}^{|\mathcal{V}|\times d}, \quad \boldsymbol{Y}^a \triangleq f_\phi(\mathcal{H}_a) \in \mathbb{R}^{|\mathcal{V}|\times d}, \tag{5}$$

where $\boldsymbol{z}^a_i$ and $\boldsymbol{y}^a_i$ denote the embedding vectors of the i-th node in $\boldsymbol{Z}^a$ and $\boldsymbol{Y}^a$, respectively, and d is the embedding dimension. Note that the same GNN f_ϕ is used to produce both $\boldsymbol{Z}^a$ and $\boldsymbol{Y}^a$.

In this work, we train the neural network f_ϕ using the set of pairs $\{(\boldsymbol{Z}^a, \boldsymbol{Y}^a)\}_{a\in\mathcal{K}}$, where $(\boldsymbol{Z}^a, \boldsymbol{Y}^a)$ corresponds to the graph embedding obtained by the GRN augmentation operation and the embedding of the teacher GRN for the corresponding gene knockdown.

4 Method

For the set of embedded representations $\{(\boldsymbol{Z}^a, \boldsymbol{Y}^a)\}_{a\in\mathcal{K}}$, we consider the pair of augmentation operations (a, b) and the pair of nodes (i, j) according to the contractive learning scheme. First, for the pair of augmentation operations (a, b), we clarify the supervised learning problem for augmentation operations using KL divergence and then propose SupGCL using a distribution over pairs of combinations of nodes and extension operations. A sketch of the proposed method is shown in Figure 1.

The probability distribution of augmentation operations is naturally introduced by using similarities in the entire graph embedding space $\mathbb{R}^{|\mathcal{V}|\times d}$ (rather than per node). By introducing the Frobenius inner product as the similarity in the matrix space, we define the probability models for the augmentation operations as:

$$p_\phi(b|a) \triangleq \frac{\exp\left(\mathrm{sim}_F(\boldsymbol{Y}^a, \boldsymbol{Y}^b)/\tau_{\mathrm{a}}\right)}{\sum_{c\in\mathcal{K}} \exp\left(\mathrm{sim}_F(\boldsymbol{Y}^a, \boldsymbol{Y}^c)/\tau_{\mathrm{a}}\right)}, \quad q_\phi(b|a) \triangleq \frac{\exp\left(\mathrm{sim}_F(\boldsymbol{Z}^a, \boldsymbol{Z}^b)/\tau_{\mathrm{a}}\right)}{\sum_{c\in\mathcal{K}} \exp\left(\mathrm{sim}_F(\boldsymbol{Z}^a, \boldsymbol{Z}^c)/\tau_{\mathrm{a}}\right)}, \tag{6}$$

where $\mathrm{sim}_F(\cdot,\cdot)$ denotes the cosine similarity via the Frobenius inner product, and $\tau_{\mathrm{a}} > 0$ is a temperature parameter. Unlike node-level learning, $p_\phi(b|a)$ is not a fixed constant but rather a reference distribution based on the supervised embeddings $\{\boldsymbol{Y}^a\}_{a\in\mathcal{K}}$. Both probability models p_ϕ and q_ϕ are parameterized by the same GNN f_ϕ.

Using these probability distributions, substituting the reference model $p_\phi(b|a)$ and the learned model $q_\phi(b|a)$ into the formulation of contrastive learning in (1) yields the loss function for augmentation

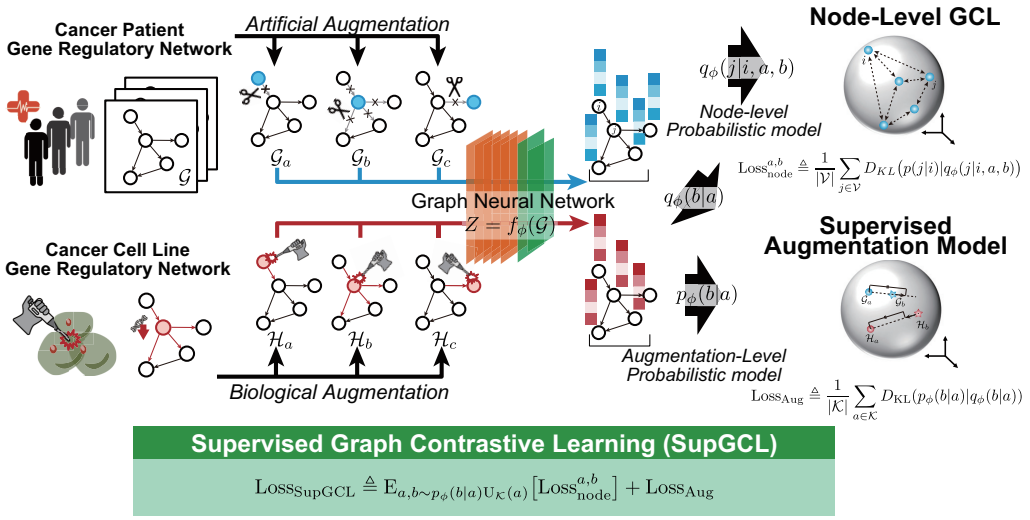

Figure 1: **Schematic overview of SupGCL.** Artificial augmentations are generated by simulating gene knockdowns in a patient GRN, while the teacher GRNs for supervision are derived from real-world knockdown experiments. Embeddings are extracted using a shared GNN, and both node-level and augmentation-level contrastive losses are computed via KL divergence.

operations:

$$\text{Loss}_{\text{Aug}} \triangleq \frac{1}{|\mathcal{K}|}\sum_{a\in\mathcal{K}} D_{\text{KL}}(p_\phi(b|a)|q_\phi(b|a)). \tag{7}$$

Minimizing this loss reduces the discrepancy in embedding distributions between the artificially augmented graphs and the biologically grounded knockdown graphs. However, if both p_ϕ and q_ϕ are optimized simultaneously, the model may converge to a trivial solution. For instance, if the GNN outputs constant embeddings, both distributions become uniform and $\text{Loss}_{\text{Aug}} = 0$. Thus, minimizing Loss_{Aug} alone is insufficient for learning meaningful graph representations.

To address this issue, here we first introduce a reference model $p_\phi(j, b|i, a)$ and a learned model $q_\phi(j, b|i, a)$ that use conditional probabilities for each pair of node and augmentation $(i, a), (j, b) \in \mathcal{V} \times \mathcal{K}$. By substituting these into the contrastive learning formulation in (1), we derive the loss function of Supervised Graph Contrastive Learning:

$$\text{Loss}_{\text{SupGCL}} \triangleq \frac{1}{|\mathcal{V}||\mathcal{K}|}\sum_{i\in\mathcal{V},a\in\mathcal{K}} D_{\text{KL}}(p_\phi(j, b|i, a)|q_\phi(j, b|i, a)). \tag{8}$$

Since data augmentation affects the entire graph, we assume that the graph-level teacher distribution $p_\phi(b|a)$, which evaluates the similarity between augmentation operations, and the node-level teacher distribution $p(j|i)$, which evaluates node identity, are independent. This leads to the following theorem.

Theorem 1. *Assuming* $p_\phi(i, j, a, b) = p(i, j)p_\phi(a, b)$*, then*

$$\text{Loss}_{\text{SupGCL}} = \mathbb{E}_{a,b\sim p_\phi(b|a)\mathrm{U}_\mathcal{K}(a)}\left[\text{Loss}^{a,b}_{\text{node}}\right] + \text{Loss}_{\text{Aug}}. \tag{9}$$

Proof: This follows directly from the standard decomposition of KL divergence: $D_{\text{KL}}(p(x, y)|q(x, y)) = \mathbb{E}_{x\sim p(x)}[D_{\text{KL}}(p(y|x)|q(y|x))] + D_{\text{KL}}(p(x)|q(x))$. See Appendix A for details. □

The first term in Theorem 1 corresponds to the expectation of the node-level GCL loss $\text{Loss}^{a,b}_{\text{node}}$ (as defined in (3)) with respect to the supervised augmentation distribution $p_\phi(b|a)$. This allows

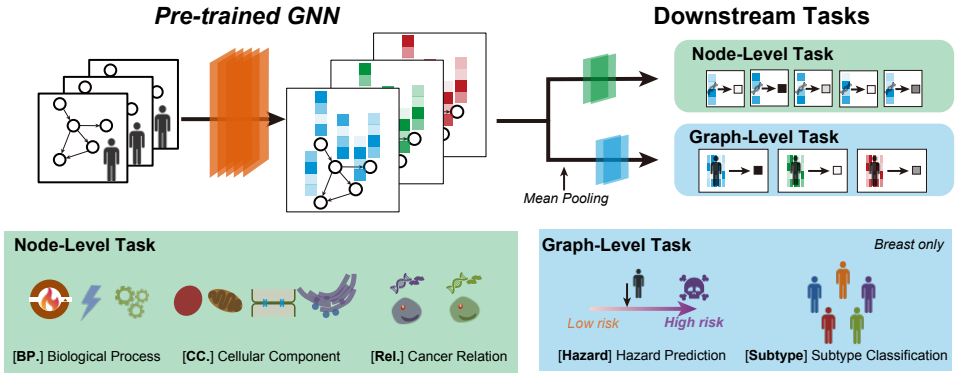

Figure 2: **Overview of downstream tasks.** Node-level tasks involve gene classification into Biological Process [**BP.**], Cellular Component [**CC.**], and cancer relevance [**Rel.**]. Graph-level tasks include patient survival prediction [**Hazard**] and breast cancer subtyping [**Subtype**]. Mean pooling provides graph-level representations.

Table 1: Description of the downstream tasks

Task	Task Type	Metrics
Node-Level Task		
[**BP.**]: Biological process classification	Multi-label binary classification (with 3 labels)	Subset accuracy
[**CC.**]: Cellular component classification	Multi-label binary classification (with 4 labels)	Subset accuracy
[**Rel.**]: Cancer relation	Classification (binary)	Accuracy
Graph-Level Task		
[**Hazard**]: Hazard prediction	Survival analysis (1-dim risk score)	C-index
[**Subtype**]: Disease subtype prediction	Classification (5 groups)	Accuracy

node-level contrastive learning to reflect biological similarity between knockdown operations. Importantly, since the theorem is independent of the specific choice of the node-level model $(p, q_\phi^{a,b})$, any contrastive loss described by KL divergence can be used in practice. Meanwhile, the second term reduces the distributional difference between the artificially generated augmentation-based GRN and the teacher GRN. Together, these two components ensure both expressive node representations and biologically meaningful augmentations.

Moreover, the performance of node-level representation learning and the biological validity following the teacher data for augmentation operations can be controlled by the temperature parameters τ_{n} and τ_{a} of each probability model. In particular, when the temperature parameter τ_{a} involved in the augmentation operation is sufficiently large, the augmentation operation becomes independent of the teacher GRNs $\{\boldsymbol{Y}_a\}_{a\in\mathcal{K}}$, and coincides with the conventional node-level GCL loss function.

Corollary 1. $\lim_{\tau_{\mathrm{a}}\to\infty} \mathrm{Loss}_{\mathrm{SupGCL}} = \mathrm{Loss}_{\mathrm{node}}$.

Proof: As $\tau_{\mathrm{a}} \to \infty$, we have $p_\phi(b|a) \to \mathrm{U}_{\mathcal{K}}(b)$ and $q_\phi(b|a) \to \mathrm{U}_{\mathcal{K}}(b)$. Therefore, the expectation term becomes: $\lim_{\tau_{\mathrm{a}}\to\infty} \mathbb{E}_{a,b\sim p_\phi(b|a)\mathrm{U}_{\mathcal{K}}(a)}\left[\mathrm{Loss}_{\mathrm{node}}^{a,b}\right] = \mathbb{E}_{a,b\sim \mathrm{U}_{\mathcal{K}}}[\mathrm{Loss}_{\mathrm{node}}^{a,b}]$, $\lim_{\tau_{\mathrm{a}}\to\infty} D_{\mathrm{KL}}(p_\phi(b|a)|q_\phi(b|a)) = 0$ thus proving the corollary. □

In this study, we train the GNN using standard gradient-based optimization applied to the loss function defined in Theorem 1. The corresponding pseudocode is provided in Appendix B.

5 Experiments

This chapter verifies the effectiveness of the proposed method using actual GRNs from cancer patients and biologically augmented GRNs based on gene knockdown experiments.

Table 2: Fine-tuning results of node-level downstream tasks

Task	w/o-pretrain	GAE	GraphCL	GRACE	SGRL	SupGCL
BP.						
Breast	0.232±0.031	0.230±0.029	0.167±0.042	0.230±0.051	0.220±0.052	**0.243±0.052**
Lung	0.259±0.056	0.247±0.038	0.115±0.024	0.259±0.063	0.233±0.027	**0.282±0.037**
Colorectal	0.231±0.062	0.245±0.023	0.207±0.058	0.249±0.050	0.146±0.029	**0.262±0.030**
CC.						
Breast	0.264±0.042	0.250±0.034	0.131±0.050	0.236±0.026	0.249±0.030	**0.291±0.026**
Lung	0.267±0.041	0.245±0.033	0.069±0.041	0.255±0.043	0.248±0.037	**0.274±0.044**
Colorectal	0.278±0.098	0.256±0.042	0.190±0.062	0.265±0.030	0.133±0.081	**0.279±0.052**
Rel.						
Breast	0.573±0.033	0.561±0.059	0.553±0.051	0.575±0.035	0.580±0.055	**0.600±0.057**
Lung	0.575±0.053	0.568±0.029	0.555±0.036	0.592±0.038	0.593±0.034	**0.604±0.053**
Colorectal	0.563±0.071	0.574±0.049	0.535±0.056	0.576±0.071	0.580±0.042	**0.594±0.039**

Table 3: Fine-tuning results of graph-level downstream tasks

Task	w/o-pretrain	GAE	GraphCL	GRACE	SGRL	SupGCL
Hazard						
Breast	0.601±0.035	0.625±0.035	0.638±0.049	0.642±0.064	0.640±0.077	**0.650±0.059**
Lung	0.611±0.052	0.619±0.062	0.616±0.049	0.609±0.055	0.611±0.060	**0.627±0.051**
Colorectal	0.621±0.070	0.631±0.091	0.657±0.071	0.647±0.059	0.616±0.123	**0.698±0.085**
Subtype						
Breast	0.804±0.031	0.834±0.028	0.719±0.077	0.841±0.026	0.829±0.030	**0.847±0.036**

5.1 Benchmark of Gene Regulatory Networks

Evaluation Protocol: We evaluated the proposed method through the following procedure. First, we constructed patient-specific GRNs from cancer patient gene expression data and teacher GRNs from gene knockdown experiment data. Then, pre-training was performed on the proposed method using both the patient-specific and teacher GRNs. Subsequently, the performance of the downstream tasks, such as classification accuracy and regression performance, was evaluated using the pre-trained models and compared against comparative methods.

We compared the proposed method with the following five comparative models:

w/o-pretrain : Directly performs classification or regression for downstream tasks.

GAE (Kipf & Welling, 2016): Reconstruction based graph representation learning method.

GraphCL (You et al., 2021): Graph contrastive learning using positive pairs between graphs.

GRACE (Zhu et al., 2020): Node-level graph contrastive learning method.

SGRL (He et al., 2024): State-of-the-art augmentation-free GCL.

Datasets: To evaluate the performance of SupGCL, we conducted benchmark evaluations using real-world datasets. For constructing patient-specific GRNs, we used cancer cell sample data from The Cancer Genome Atlas (TCGA). For constructing teacher GRNs, we used gene knockdown experiment data from cancer cell lines in the Library of Integrated Network-based Cellular Signatures (LINCS). The TCGA dataset (Weinstein et al., 2013) and the LINCS dataset (Subramanian et al., 2017) are both large-scale and widely-used public platforms providing gene expression data from cancer patients and cell lines, respectively.

Experiments were conducted for three cancer types across both datasets: breast cancer, lung cancer, and colorectal cancer. Furthermore, the set of genes constituting each network was restricted to the 975 genes common to the TCGA gene set and the 978 LINCS landmark genes. The number of patient samples for each cancer type was N=1092 (breast), 1011 (lung), and 288 (colorectal), and the total number of knockdown experiments was 8793, 11843, and 15926, respectively. The number of unique knockdown target genes / total common genes was 768/975, 948/975, and 948/975.

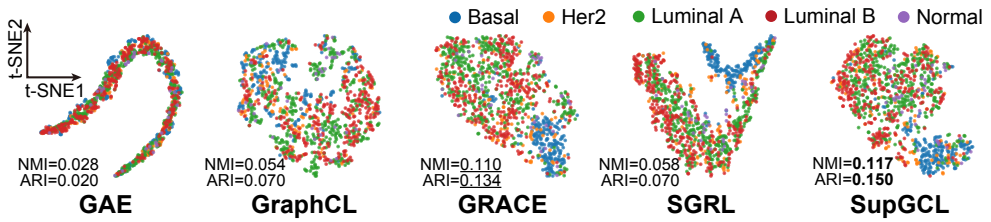

Figure 3: t-SNE visualization of pre-trained graph-level embeddings on breast cancer GRNs. Each point represents the readout feature of an individual patient's network. NMI and ARI scores indicate quantitative clustering metrics of the embeddings.

The TCGA dataset also includes survival status and disease subtype labels associated with each gene expression profile. Additionally, each gene was annotated with multi-labels based on Gene Ontology (Ashburner et al., 2000) — Biological Process (metabolism, signaling, cell organization; 3 classes), and Cellular Component (nucleus, mitochondria, ER, membrane; 4 classes). We also used the OncoKB (Chakravarty et al., 2017) cancer-related gene list to assign binary relevance labels. These labels were used for downstream tasks. Details are provided in Appendix C.

Pre-processing: To estimate the network structure of each GRN from gene expression data, we used a Bayesian network structure learning algorithm based on B-spline regression (Imoto et al., 2002). For each experiment, gene expression values were used as node features, while edge features were defined as the linear sum of estimated regression coefficients and the parent node's gene expression (Tanaka et al., 2020). This structure estimation was performed per cancer type per dataset using the above algorithm. Further details are provided in Appendix D.

5.2 Result 1: Evaluation by Downstream Task

In this experiment, pre-training of the proposed and conventional methods was conducted using patient-specific GRNs from TCGA and teacher GRNs from LINCS. Subsequently, fine-tuning was performed on the pre-trained models using patient GRNs, and downstream task performance was evaluated (see Figure 2). During fine-tuning, two additional fully connected layers were appended to the node-level representations and graph-level representations (obtained via mean pooling), and downstream tasks were performed.

Graph-level tasks (hazard prediction, subtype classification) used survival and subtype labels from TCGA. Note that subtype classification was conducted only for breast cancer. Node-level tasks (Biological Process - BP., Cellular Component - CC., and cancer relevance - Rel.) used gene-level annotations from Gene Ontology and OncoKB. Details of these downstream tasks are summarized in Table 1.

Each downstream task — hazard prediction, subtype classification, BP, and CC classification — was evaluated using 10-fold cross-validation. For cancer gene classification, due to label imbalance, we performed undersampling over 10 random seeds. Results are reported as mean ± standard deviation.

For all methods including the SupGCL and conventional methods, we used the same 5-layer Graph Transformer architecture (Shi et al., 2021). Hyperparameters for pre-training were tuned with Optuna (Akiba et al., 2019), and the model was optimized using the AdamW optimizer (Loshchilov & Hutter, 2019). All training runs were performed on a single NVIDIA H100 SXM5 GPU. Additional experimental details can be found in Appendix E.

Tables 2 and 3 show the results for node-level and graph-level tasks, respectively. The best performance is indicated in bold, and the second-best is underlined. Although SupGCL did not achieve statistically significant superiority in every single task, it consistently outperformed other pre-training methods across all datasets and tasks.

For node-level tasks, many existing methods did not show much improvement over without-pretrain, whereas SupGCL consistently demonstrated strong performance. In graph-level tasks like hazard

and subtype prediction, while some existing methods showed marginal improvement over without-pretrain, SupGCL achieved significantly higher performance.

In addition, we evaluated the generalization performance across different cancer types. While the pre-training was robust even when the cancer types of the teacher and patient GRNs differed, the fine-tuning was not (see Appendix G for details).

5.3 Result 2: Effectiveness of Supervised Learning

In this section, to verify the effectiveness of the reference model p_ϕ, we conducted additional experiments by varying the augmentation-level temperature parameter τ_{a}, which controls its degree of influence. This experiment also serves to empirically validate the theoretical relationship (Corollary 1) that as τ_{a} increases, the SupGCL loss function asymptotically approaches that of unsupervised node-level GCL like GRACE. The experimental results for breast cancer in Table 4 show that SupGCL outperforms GRACE in all settings, demonstrating the effectiveness of incorporating biologically plausible information. Furthermore, as τ_{a} increases and the influence of supervised information weakens, the performance of SupGCL tends to approach that of the unsupervised method GRACE, which supports Corollary 1. It should be noted that due to the different definitions of the denominator in their loss functions, this experiment and GRACE are not strictly identical.

Table 4: Ablation study of SupGCL on the augmentation temperature τ_{a} for breast cancer.

τ_{a}	**Hazard**	**Subtype**	**BP.**	**CC.**	**Rel.**
0.10	$\mathbf{0.670 \pm 0.078}$	0.837 ± 0.029	$\mathbf{0.262 \pm 0.035}$	0.289 ± 0.049	0.586 ± 0.047
0.25 (default)	0.650 ± 0.059	$\mathbf{0.847 \pm 0.036}$	0.243 ± 0.052	$\mathbf{0.291 \pm 0.026}$	0.600 ± 0.057
0.50	0.648 ± 0.053	0.846 ± 0.032	0.261 ± 0.034	0.284 ± 0.061	$\mathbf{0.606 \pm 0.039}$
1.00	0.640 ± 0.056	0.835 ± 0.031	0.244 ± 0.042	0.280 ± 0.032	0.596 ± 0.048
2.00	0.656 ± 0.060	0.842 ± 0.031	0.237 ± 0.024	0.277 ± 0.058	0.590 ± 0.044
Reference: GRACE	0.642 ± 0.064	0.841 ± 0.026	0.230 ± 0.051	0.236 ± 0.026	0.575 ± 0.035

5.4 Result 3: Latent Space Analysis

We visualized the graph-level embedding spaces from the breast cancer dataset (Figure 3; additional visualizations including node-level embeddings are in Appendix F). The embeddings, colored by disease subtype, show that GAE and GraphCL fail to separate subtypes, while GRACE and SGRL exhibit moderate separation. SupGCL achieves the clearest separation, indicating a superior ability to learn subtype-specific representations. This qualitative observation was confirmed by quantitative clustering metrics. SupGCL consistently outperformed other methods on both the Normalized Mutual Information (NMI) and the Adjusted Rand Index (ARI), affirming that its embeddings more effectively capture subtype-specific structure.

6 Conclusion

In this study, we proposed a supervised graph contrastive learning method, SupGCL, for representation learning of gene regulatory networks (GRNs), which incorporates real-world genetic perturbation data as supervision during training. We formulated GCL with supervision-guided augmentation selection within a unified probabilistic framework, and theoretically demonstrated that conventional GCL methods are special cases of our proposed formulation. Through benchmark evaluations using downstream tasks based on both node-level and graph-level embeddings of GRNs from cancer patients, SupGCL consistently outperformed existing GCL methods.

A key limitation of our study is that SupGCL, when pre-trained on a specific cancer type, fails to improve performance on downstream tasks for other cancer types. As future work, we plan to expand the target cancer types and develop a large-scale, general-purpose SupGCL model that can operate across multiple cancer types.

REFERENCES

Takuya Akiba, Shotaro Sano, Tetsuo Yanase, Toshihiko Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*, pp. 2623–2631, Anchorage, AK, USA, 2019. ACM.

Shaden Naif Alshammari, Mark Hamilton, Axel Feldmann, John R. Hershey, and William T. Freeman. A unifying framework for representation learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Helen Butler, J Michael Cherry, Ana Davis, Kara Dolinski, Sally S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

Debyani Chakravarty, J Gao, S Phillips, R Kundra, H Zhang, J Wang, J E Rudolph, R Yaeger, T Soumerai, M Nissan, et al. Oncokb: A precision oncology knowledge base. *JCO Precision Oncology*, 2017:PO.17.00011, 2017.

Dongxiao He, Lianze Shan, Jitao Zhao, Hengrui Zhang, Zhen Wang, and Weixiong Zhang. Exploitation of a latent mechanism in graph contrastive learning: Representation scattering. *Advances in Neural Information Processing Systems*, 37:115351–115376, 2024.

Xinxin Hu, Haotian Chen, Hongchang Chen, Shuxin Liu, Xing Li, Shibo Zhang, Yahui Wang, and Xiangyang Xue. Cost-sensitive gnn-based imbalanced learning for mobile social network fraud detection. *IEEE Transactions on Computational Social Systems*, 11(2):2675–2690, 2023.

Vân Anh Huynh-Thu, Alexandre Irrthum, Louis Wehenkel, and Pierre Geurts. Inferring regulatory networks from expression data using tree-based methods. *PloS one*, 5(9):e12776, 2010.

Seiya Imoto, Takao Goto, and Satoru Miyano. Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pp. 175–186, 2002. ISSN 2335-6928.

Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, Junwei Yang, Jingyang Yuan, Yusheng Zhao, Yifan Wang, Xiao Luo, and Ming Zhang. A comprehensive survey on deep graph representation learning. *Neural Networks*, 173:106207, May 2024. ISSN 0893-6080.

Seunghwan Jung, Seunghyun Wang, and Doheon Lee. Cancergate: Prediction of cancer-driver genes using graph attention autoencoders. *Computers in Biology and Medicine*, 176:108568, 2024.

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

Namkyeong Lee, Junseok Lee, and Chanyoung Park. Augmentation-free self-supervised learning on graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7372–7380, Jun. 2022.

Tianyu Liu, Yuge Wang, Rex Ying, and Hongyu Zhao. Muse-gnn: Learning unified gene representation from multimodal biological graph data. *Advances in neural information processing systems*, 36:24661–24677, 2023.

Xuan Liu, Congzhi Song, Feng Huang, Haitao Fu, Wenjie Xiao, and Wen Zhang. Graphcdr: a graph neural network method with contrastive learning for cancer drug response prediction. *Briefings in Bioinformatics*, 23(1):bbab457, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Mai Adachi Nakazawa, Yoshinori Tamada, Yoshihisa Tanaka, Marie Ikeguchi, Kako Higashihara, and Yasushi Okuno. Novel cancer subtyping method based on patient-specific gene regulatory network. *Scientific reports*, 11(1):23653, 2021.

Evan O Paull, Alvaro Aytes, Sunny J Jones, Prem S Subramaniam, Federico M Giorgi, Eugene F Douglass, Somnath Tagore, Brennan Chu, Alessandro Vasciaveo, Siyuan Zheng, et al. A modular master regulator landscape controls cancer transcriptional identity. *Cell*, 184(2):334–351, 2021.

Xintao Shen and Yulai Zhang. A knowledge graph recommendation approach incorporating contrastive and relationship learning. *IEEE Access*, 11:99628–99637, 2023.

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 1548–1554. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

Aravind Subramanian, Ramachandran Narayan, Simone M Corsello, and et al. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6):1437–1452.e17, 2017.

Jiawei Sun, Ruoxin Chen, Jie Li, Yue Ding, Chentao Wu, Zhi Liu, and Junchi Yan. Understanding and mitigating dimensional collapse of graph contrastive learning: A non-maximum removal approach. *Neural Networks*, 181:106652, 2025.

Yoshihisa Tanaka, Yoshinori Tamada, Marie Ikeguchi, Fumiyoshi Yamashita, and Yasushi Okuno. System-based differential gene network analysis for characterizing a sample-specific subnetwork. *Biomolecules*, 10(2), 2020.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 workshop on geometrical and topological representation learning*, 2021.

Chenhao Wang, Yong Liu, Yan Yang, and Wei Li. Hetergcl: graph contrastive learning framework on heterophilic graph. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 2397–2405, 2024.

John N Weinstein, Eric A Collisson, Gordon B Mills, Karthik R M Shaw, Bradley A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M Stuart, et al. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113–1120, 2013.

Lirong Wu, Haitao Lin, Cheng Tan, Zhangyang Gao, and Stan Z Li. Self-supervised learning on graphs: Contrastive, generative, or predictive. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):4216–4235, 2021.

Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM web conference 2022*, pp. 1070–1079, 2022.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823, 2020.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International conference on machine learning*, pp. 12121–12132. PMLR, 2021.

Weiming Yu, Zerun Lin, Miaofang Lan, and Le Ou-Yang. Gclink: a graph contrastive link prediction framework for gene regulatory network inference. *Bioinformatics*, 41(3):btaf074, 2025.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the web conference 2021*, pp. 2069–2080, 2021.

Payam Zohari and Mostafa Haghir Chehreghani. Graph neural networks in multi-omics cancer research: A structured survey. *arXiv preprint arXiv:2506.17234*, 2025.

APPENDIX CONTENTS

A Detail Proof of Theorem 1

Since the loss function of contrastive learning (1) is formulated 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 $\mathrm{Loss}_{\mathrm{I-CON}}$:

$$\begin{aligned}
\mathrm{Loss}_{\mathrm{SupGCL}} &= \frac{1}{|V||\mathcal{K}|} \sum_{i \in V, a \in \mathcal{K}} D_{\mathrm{KL}}(p_\phi(j, b|i, a)|q_\phi(j, b|i, a)). \\
&= D_{\mathrm{KL}}(p_\phi(i, j, a, b)|q_\phi(i, j, a, b)) \\
&= \mathbb{E}_{(a,b)\sim p_\phi(a,b)} \Big[D_{\mathrm{KL}}(p_\phi(i, j|a, b)|q_\phi(i, j|a, b)) \Big] + D_{\mathrm{KL}}(p_\phi(a, b)|q_\phi(a, b)) \\
&= \mathbb{E}_{(a,b)\sim p_\phi(a,b)} \Big[D_{\mathrm{KL}}(p(i, j)|q_\phi(i, j|a, b)) \Big] + D_{\mathrm{KL}}(p_\phi(a, b)|q_\phi(a, b)) \\
&= \mathbb{E}_{a,b\sim p_\phi(b|a)\mathrm{U}_{\mathcal{K}}(a)} \big[\mathrm{Loss}^{a,b}_{\mathrm{node}} \big] + \mathrm{Loss}_{\mathrm{Aug}}
\end{aligned}$$

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

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, $\mathcal{G}_{\mathrm{all}} = \{\mathcal{G}^{(i)}\}_{i=1}^N$, and the teacher GRN dataset, $\mathcal{H}_{\mathrm{all}} = \{\mathcal{H}_a^{(i)}\}_{i\in\mathcal{I}_a, a\in\mathcal{K}}$. 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 $\mathrm{Loss}_{\mathrm{SupGCL}}$ 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 $\mathrm{Loss}^{a,b}_{\mathrm{node}}$ and $\mathrm{Loss}_{\mathrm{Aug}}$.

Algorithm 1 Training loop of SupGCL:

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

for $\mathcal{G} \subset \mathcal{G}_{\mathrm{all}}$ **do**
  $a, b \sim \mathrm{U}_{\mathcal{K}}$
  $\mathcal{G}_a, \mathcal{G}_b \leftarrow$ The a-th and b-th artificial augmentation of $\mathcal{G}$
  $\mathcal{H}_a, \mathcal{H}_b \leftarrow$ Pick up the a-th and b-th knockdown teacher GRN from $\mathcal{H}_{\mathrm{all}}$
  $\boldsymbol{Z}^a, \boldsymbol{Z}^b \leftarrow f_\phi(\mathcal{G}_a),\ f_\phi(\mathcal{G}_b)$ ▷ Embedding target GRNs
  $\boldsymbol{Y}^a, \boldsymbol{Y}^b \leftarrow f_\phi(\mathcal{H}_a),\ f_\phi(\mathcal{H}_b)$ ▷ Embedding teacher GRNs
  $q_\phi(b|a) \leftarrow \mathrm{softmax}\left(\left[\frac{\langle \boldsymbol{Z}^a, \boldsymbol{Z}^* \rangle_F}{\tau_{\mathrm{a}}}\right]\right)[b]$ ▷ Calculate augmentation-level target model
  $p_\phi(b|a) \leftarrow \mathrm{softmax}\left(\left[\frac{\langle \boldsymbol{Y}^a, \boldsymbol{Y}^* \rangle_F}{\tau_{\mathrm{a}}}\right]\right)[b]$ ▷ Calculate augmentation-level teacher model
  $\mathrm{Loss}_{\mathrm{Aug}} \leftarrow |\mathcal{K}| p_\phi(b|a)\big(\log p_\phi(b|a) - \log q_\phi(b|a)\big)$ ▷ Importance sampling of $\mathrm{Loss}^{a,b}_{\mathrm{Aug}}$
  $q_\phi(j|i, a, b) \leftarrow \mathrm{softmax}\left(\left[\frac{\langle \boldsymbol{z}_i^a, \boldsymbol{z}_*^b \rangle}{\tau_n}\right]\right)[j]$ ▷ Calculate node-level target model
  $\mathrm{Loss}^{a,b}_{\mathrm{node}} \leftarrow \frac{1}{|\mathcal{V}|} \sum_{i\in\mathcal{V}} \log q_\phi(i|i, a, b)$
  $\mathrm{Loss}_{\mathrm{SupGCL}} \leftarrow |\mathcal{K}| p_\phi(b|a) \mathrm{Loss}^{a,b}_{\mathrm{node}} + \mathrm{Loss}_{\mathrm{Aug}}$ ▷ Importance sampling of $\mathrm{Loss}^{a,b}_{\mathrm{node}}$
  Update f_ϕ using $\mathrm{Loss}_{\mathrm{SupGCL}}$ and AdamW optimizer
end for
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, for gene expression data, we used normalized count data from the TCGA TARGET GTEx study provided by UCSC Xena (UCSC Xena (2016)) for the TCGA dataset. For the LINCS dataset, we used normalized gene expression data from the LINCS L1000 GEO dataset (GSE92742) (Subramanian (2017)).

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 5, 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 5.

Table 5: 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' (Paolacci et al. (2009)). Similarly, for CC labels, four categories were used: 'Nucleus', 'Mitochondrion', 'Endoplasmic Reticulum', and 'Plasma Membrane' (Costa et al. (2010)). 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 (Xin et al. (2015)) and the OBO file (available at https://geneontology.org/docs/download-ontology/).

2. Aggregate multi-labels for the target genes using the GOATOOLS library (Ashburner et al. (2000)).

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 6.

Table 6: 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.4 Annotation by OncoKB

For cancer-related gene classification in downstream tasks, cancer-related genes were obtained from the OncoKB™ Cancer Gene List provided by OncoKB (Oncology Knowledge Base) (Chakravarty et al., 2017). 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 (Langfelder & Horvath (2008)), mutual information-based (Margolin et al. (2006)), probabilistic graphical models-based such as Bayesian networks (Friedman et al. (2000)), and deep learning-based approaches (Shu et al. (2021)). 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 (Imoto et al., 2002) to identify patient- or sample-specific GRNs (Nakazawa et al., 2021).

D.2 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 $x_1, x_2, ..., x_n$ be random variables for n nodes, and let $\mathrm{pa}(i)$ be the set of parent nodes of the i-th node. In this case, a Bayesian network using B-spline curves (Imoto et al., 2002) is defined as a

probabilistic model decomposed into conditional distributions with parent nodes:

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_{\mathrm{pa}(i)}) = \prod_{i=1}^{n} \mathcal{N}\left(x_i \middle| \sum_{j \in \mathrm{pa}(i)} m_{ij}(x_j), \sigma^2 \right).$$

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

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

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 (Imoto et al., 2002). 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 (Imoto et al. (2003)). 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 (Tanaka et al., 2020). 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 $x_1, \ldots, x_n$. Furthermore, the feature of the i-th node $X_i^{\mathcal{V}}$ 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 $X_k^{\mathcal{E}} = m_{ij}(x_j)$. Patient-specific networks using such edge features have led to the discovery of subtypes that correlate more strongly with prognosis than existing subtypes (Nakazawa et al., 2021). 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 (Tomoto et al. (2024)) and chronic nonbacterial osteomyelitis (Yahara et al. (2022))

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 (Tamada et al. (2011)), 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 LINCS 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 7. In addition, the number of parallel threads was set to 4 for estimations using TCGA data and 2 for LINCS data.

Table 7: Estimated network times in supercomputer Fugaku

	Breast cancer		**Lung cancer**		**Colorectal cancer**	
	TCGA	LINCS	TCGA	LINCS	TCGA	LINCS
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.3 RESULTS

The statistics of the estimated networks are shown in Table 8. 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 8: 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 VALIDITY OF GNN ENCODER

In this study, we consistently employed the Graph Transformer as the GNN encoder architecture. The choice of encoder is a critical factor that directly affects model performance, and its validity must be carefully assessed for our method. Our decision to adopt the Graph Transformer was based on prior work on GRNs (Liu et al., 2023) (Appendix E.1), where it achieved the best performance among various architectures. That study evaluated multiple GNN encoders—including GCN, GAT, TransformConv, SURGL, GPS, and GRACE—in the context of learning gene representations by integrating heterogeneous modalities such as scRNA-seq, scATAC-seq, and spatial transcriptomics. Their results showed that the Graph Transformer consistently outperformed other models, whereas GCN- and GAT-based models failed to sufficiently capture gene functional similarity across datasets. It should be noted, however, that this prior work focused on heterogeneous modalities, and is therefore not directly comparable to our setting, which relies exclusively on GRNs.

Motivated by these findings, we chose the Graph Transformer over GCN or GAT in our framework. Both GAT and Graph Transformer share the ability to incorporate real-valued edge features into node updates. However, in our experiments (using GRACE as an example), replacing the Graph Transformer with GAT sometimes led to degraded performance. The results are reported in Table 9.

Table 9: Comparison of downstream task performance with different encoders in GRACE (Breast Cancer GRN).

Model Setting	**Hazard (c-index)**	**BP (Subset Accuracy)**
GRACE (Graph Transformer, in paper)	0.642 ± 0.064	0.230 ± 0.051
GRACE (GAT)	0.632 ± 0.038	0.241 ± 0.040

E.2 HYPERPARAMETERS SETTINGS

In this study, we used Optuna (Akiba et al., 2019), a Bayesian optimization tool, for hyperparameter search in pre-training. The search space for all models included the AdamW learning rate $\mathrm{lr} \in [10^{-5}, 10^{-3}]$, batch size $\mathrm{batch\_size} \in \{4, 8\}$, 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 10.

Table 10: 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.3 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 11. 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.

Table 11: 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

E.4 Fine-tuning Settings

E.4.1 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.

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

$$h(t \mid \boldsymbol{x}) = h_0(t) \exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}\right)$$

Here, $h_0(t)$ is the baseline hazard, $\boldsymbol{x}$ is the input variable, and $\boldsymbol{\beta}$ is the regression coefficient to be learned. The prognosis estimation is performed by connecting this input variable $\boldsymbol{x}$ 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.4.2 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 six 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.

4. Statistical significance testing for the performance metrics across downstream tasks.
5. Comparison with Augmentation-Adjustment Methods and Augmentation-Free.
6. Evaluation on Link Prediction.

Furthermore, an analysis of the robustness of SupGCL against dataset changes is provided in Appendix H.

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 12 and 13 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 14 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 12: 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	0.558±0.022	0.543±0.022	**0.571±0.025**
Lung	0.538±0.039	0.546±0.021	**0.584±0.065**	0.555±0.026	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**	0.560±0.040	0.547±0.038
CC.						
Breast	0.404±0.036	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	0.393±0.026	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	0.401±0.049	0.397±0.030
Rel.						
Breast	0.523±0.094	0.571±0.048	0.593±0.072	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	0.575±0.061	**0.592±0.067**
Colorectal	0.474±0.242	0.582±0.081	0.556±0.124	0.547±0.197	0.569±0.145	**0.596±0.060**

Table 13: Node-level downstream task: Jaccard index

Task	w/o-pretrain	GAE	GraphCL	GRACE	SGRL	SupGCL
BP.						
Breast	0.490±0.017	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	0.518±0.027
Colorectal	**0.537±0.031**	0.506±0.019	0.500±0.036	0.514±0.024	0.469±0.030	0.502±0.022
CC.						
Breast	0.402±0.052	0.378±0.021	0.303±0.028	0.359±0.028	0.377±0.029	**0.422±0.028**
Lung	0.387±0.040	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	0.388±0.036	0.360±0.053	**0.395±0.033**

Table 14: 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	0.761 ± 0.063	0.715 ± 0.064	**0.785 ± 0.056**

F.2 Additional Latent Space Analysis

Node-level latent Space: Previously, in Result 3: Latent Space Analysis, we visualized the graph-level embedding space generated by pre-trained models(Figure 3). Node-level results for the breast,

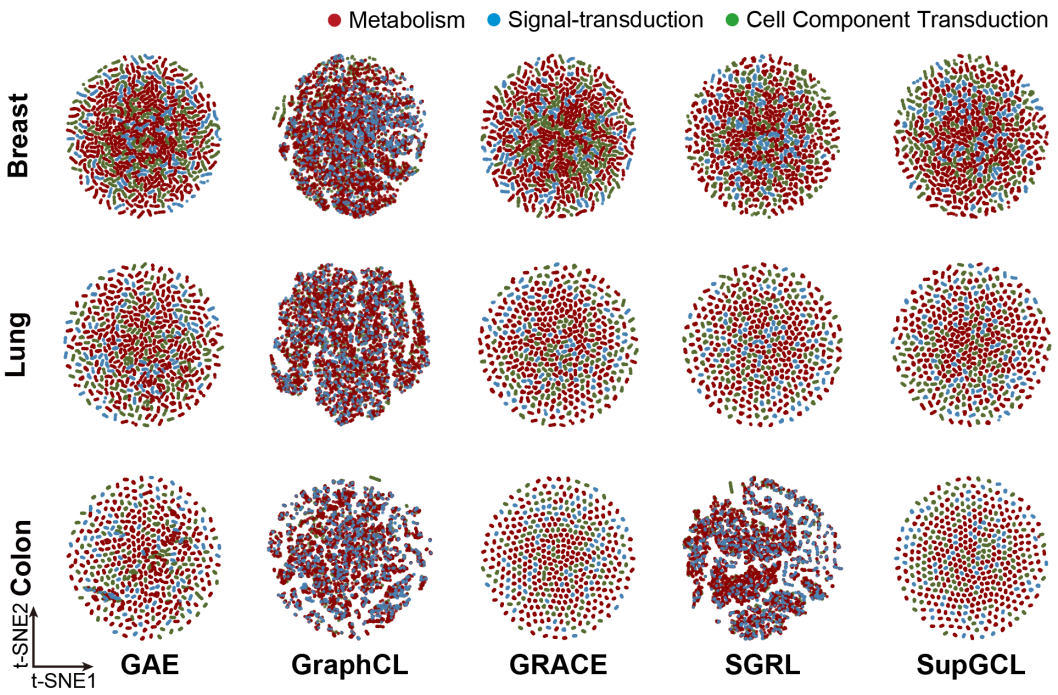

Figure 4: t-SNE visualization of pre-trained embeddings on breast, lung, and colorectal cancer GRNs.

lung, 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. A more detailed analysis on latent space collapse is provided in the next section. Furthermore, a more detailed analysis of the latent space from a biological perspective is provided in Appendix I

Analysis of Latent Space Collapse: To further investigate the characteristics of the node-level latent spaces presented in Result 3: 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.

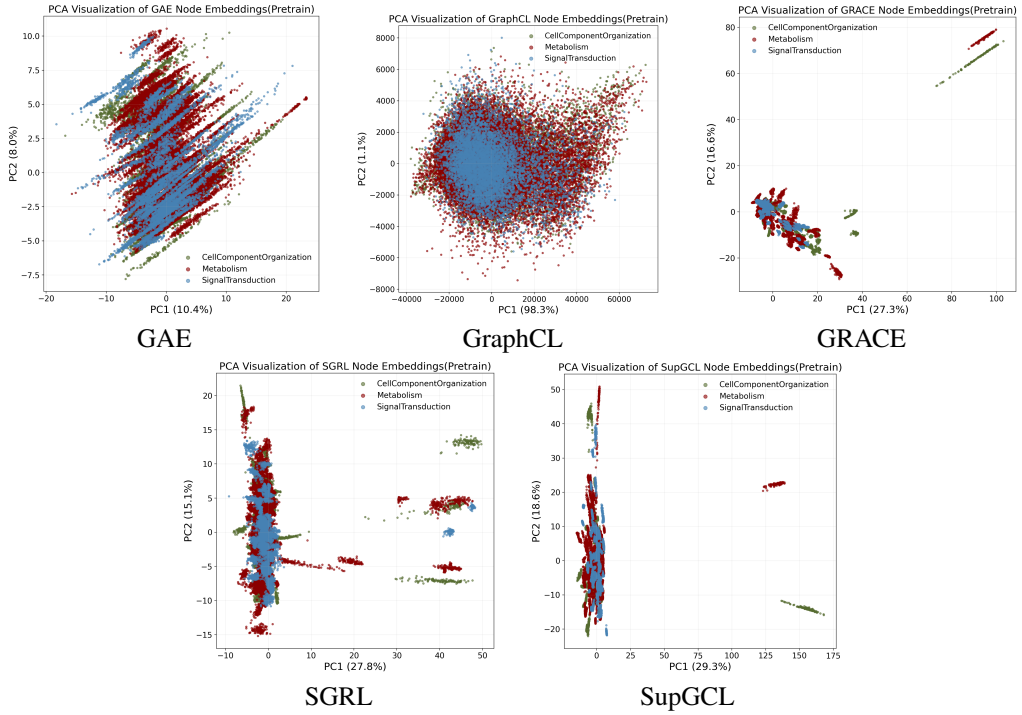

Figure 5: PCA analysis of the latent spaces of pre-trained models.

F.4 Statistical Significance Testing

We further conducted statistical significance testing for the performance metrics across downstream tasks. Specifically, we computed Bonferroni-corrected p-values for pairwise comparisons between our proposed method SupGCL and five baseline methods, using Student's t-test with a significance level of 5%. While most comparisons did not reveal statistically significant differences, we observed significant improvements over GraphCL and SGRL in a subset of node-level tasks. Table 15 reports the tasks where significant differences were found, along with the corresponding p-values.

Table 15: Comparison of SupGCL with other methods using Bonferroni-corrected p-values.

Task	Cancer Subtype	Compared Method	p-value
BP	Lung	GraphCL	1.88×10^{-3}
BP	Colorectal	SGRL	1.65×10^{-2}
CC	Breast	GraphCL	1.43×10^{-2}
CC	Lung	GraphCL	4.01×10^{-3}

F.5 Comparison with Augmentation-Adjustment Methods and Augmentation-Free

In graph contrastive learning (GCL), topological changes introduced by augmentations such as node or edge dropping are known to degrade performance. To address this issue, recent works have proposed frameworks that either correct or avoid the effects of augmentation. For instance, SGRL (He et al., 2024) (NeurIPS 2024), the successor to BGRL (Thakoor et al. (2021)), has established a state-of-the-art augmentation-free paradigm. Other representative methods include GCA (Zhu et al. (2021)), which automatically adjusts and corrects augmentation effects; AFGRL (Lee et al., 2022), which integrates reconstruction and bootstrap learning; and SimGRACE (Xia et al., 2022), which perturbs model parameters with simple Gaussian noise.

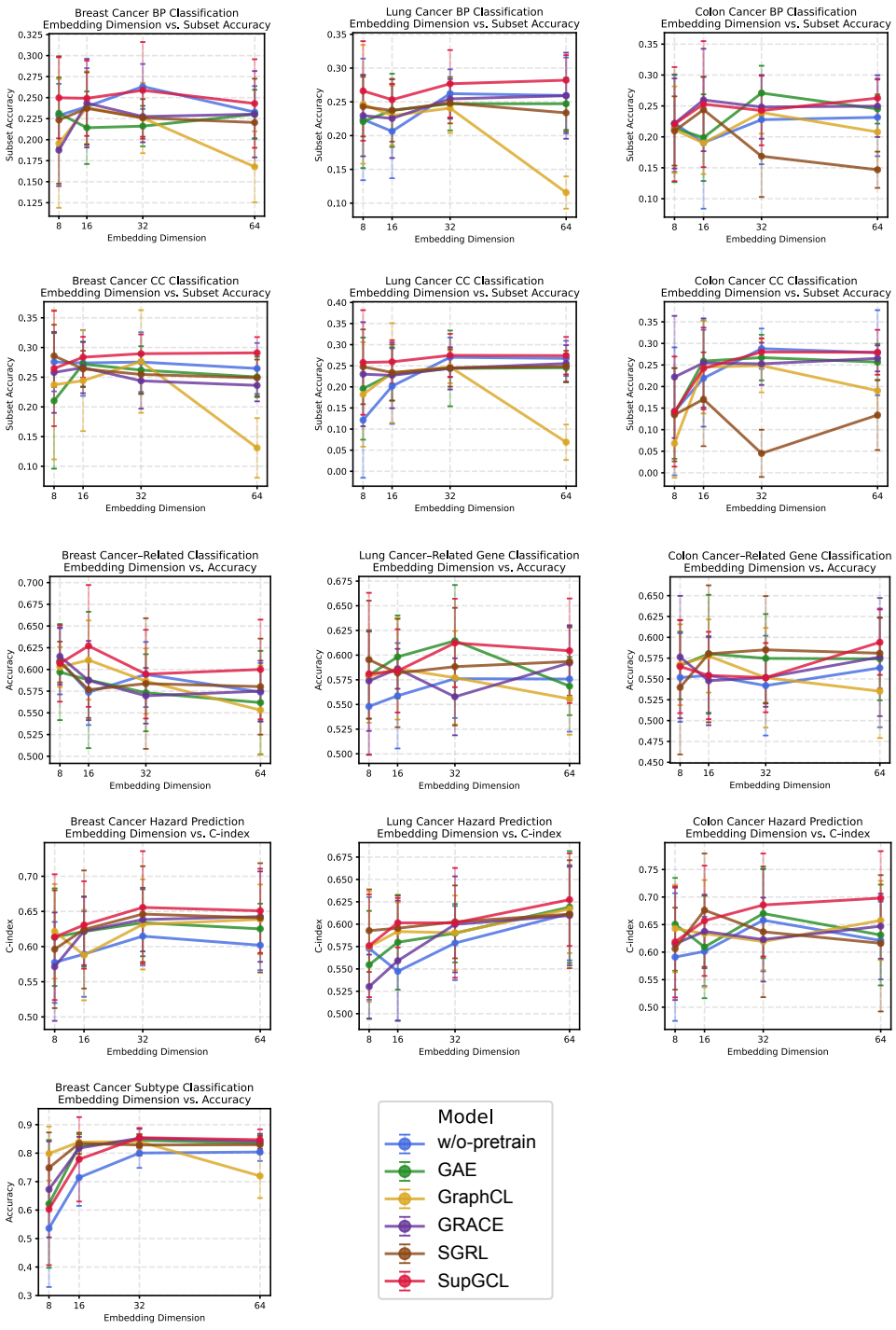

Figure 6: Embedding dimension analysis. This figure shows the performance changes across 13 tasks as the embedding dimension varies. The left column shows the results for breast cancer, the center column for lung cancer, and the right column for colorectal cancer. The first row presents the subset accuracy 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.

In contrast, SupGCL differs fundamentally from augmentation-free approaches. Rather than avoiding topological changes, SupGCL leverages them as informative signals by supervising augmentations with knockdown-derived GRNs. That is, while conventional approaches aim to circumvent augmentation-induced topology shifts, SupGCL explicitly exploits them as positive supervision, transforming a traditionally negative factor into a beneficial learning signal.

Table 16 reports the comparison on breast cancer GRNs for hazard prediction (graph-level) and BP classification (node-level).

As shown in the results, SupGCL consistently outperformed representative augmentation-correction and augmentation-free models (GCA, AFGRL, SimGRACE). This indicates that actively utilizing topology changes as supervisory signals leads to improved representation learning performance.

Table 16: Comparison with augmentation-correction and augmentation-free methods (Breast Cancer GRN).

Method	**Hazard (c-index)**	**BP (Subset Accuracy)**
GCA	0.620 ± 0.039	0.241 ± 0.021
AFGRL	0.616 ± 0.037	0.240 ± 0.026
SimGRACE	0.638 ± 0.032	0.228 ± 0.044
SupGCL (Ours)	$\mathbf{0.650 \pm 0.059}$	$\mathbf{0.243 \pm 0.052}$

F.6 Evaluation on Link Prediction

In addition to node-level and graph-level downstream tasks, we also evaluated performance on an edge-level task, namely link prediction. While the primary goal of SupGCL is to perform representation learning on already constructed GRNs rather than GRN inference itself, it is nevertheless possible to blind edges during training and use the learned representations to predict the existence of unseen edges. This provides a useful auxiliary evaluation of the learned embeddings.

We conducted experiments on the breast cancer GRN. Edges present in the GRN were treated as positive examples, and absent edges were treated as negatives. From these, 500 positive and 500 negative samples were drawn to form 1000 test cases, which were then split into train:test = 8:2 with a fixed seed. This procedure was repeated across 10 different seeds (0–9). Node embeddings were fed into a two-layer MLP head, and the inner product between resulting vectors was passed through a sigmoid function to predict edge existence. Binary cross-entropy (BCE) was used as the loss function, with target edges masked during encoding to simulate inference of unseen edges. Optimization settings were aligned with those of other tasks and kept consistent across all methods. We report Accuracy and F1 score as evaluation metrics, averaged over the 10 seeds with standard deviations.

Recent studies have also proposed supervised GNN-based approaches specifically designed for GRN inference. A representative example is GCLink (Yu et al., 2025), which directly learns from ground-truth transcriptional networks. In contrast, our study focuses on GRNs estimated from gene expression via Bayesian network inference, which encode statistical causal relationships and are not restricted to transcription factor–regulon interactions. For reference, we additionally report results of applying a GCLink-style model to our setting. Note that the GCLink results in Table 17 correspond not to the original setup (supervised by ground-truth transcriptional networks), but to a modified variant adapted to our estimated GRNs.

The results (Table 17) show that SupGCL outperforms existing baselines on edge-level GRN inference tasks as well. In particular, SupGCL achieved the highest Accuracy and F1 score compared to all other models, including GRACE. Performance was also comparable to or slightly better than GCLink. These findings suggest that although SupGCL is primarily designed for representation learning, it remains competitive for GRN inference tasks such as link prediction.

Table 17: Performance on link prediction (Breast Cancer GRN).

Method	Accuracy	F1 score
w/o pretrain	0.730 ± 0.021	0.843 ± 0.014
GAE	0.743 ± 0.024	0.846 ± 0.014
GraphCL	0.714 ± 0.031	0.824 ± 0.022
GRACE	0.757 ± 0.031	0.848 ± 0.016
SGRL	0.741 ± 0.032	0.835 ± 0.023
GCLink	0.756 ± 0.031	0.853 ± 0.018
SupGCL (Ours)	$\mathbf{0.763 \pm 0.028}$	$\mathbf{0.855 \pm 0.018}$

G CROSS-DOMAIN TRAINING

Our proposed method ultimately aims to enable cancer-type–agnostic representation learning of GRNs. Since SupGCL is a scale-free graph representation learning framework with respect to the set of genes and the knockdown perturbations applied, it can in principle be applied to diverse GRNs derived from different cancer types. Accordingly, pre-training is feasible even when the patient graphs $\mathcal{G}$ and supervision graphs $\mathcal{H}_a$ come from different cancer types. Moreover, downstream tasks can be learned even if the cancer types used for pre-training and fine-tuning differ. It is also possible to take a model fine-tuned on one cancer type and directly apply it to predict outcomes for another. Nevertheless, the extent to which SupGCL retains predictive accuracy under such cross-domain conditions requires careful investigation.

To examine cancer-type dependency, we conducted three cross-domain experiments. (i) **Pre-training phase:** using knockdown GRNs from a different cancer type than the patient GRNs. (ii) **Fine-tuning phase:** applying a pre-trained model from one cancer type to fine-tune on another. (iii) **OOD setting:** applying a model pre-trained and fine-tuned on one cancer type directly to patients of a different type. The results are summarized in Table 18.

From experiment (i), we found that using breast knockdown data to supervise learning on lung or colorectal patient GRNs yielded performance comparable to the in-domain setting. In experiment (ii), performance was preserved when the model was pre-trained on lung data and fine-tuned on breast, but performance degraded when using colorectal as the source. A similar trend was observed in experiment (iii). Overall, these results suggest that SupGCL can generalize across cancer types at the pre-training stage, where the supervision GRN comes from a different cancer type. However, differences in cancer type during the fine-tuning phase cannot be ignored: models fine-tuned on a single cancer type do not exhibit robustness when directly applied to unseen cancer types.

Table 18: Cross-domain performance of SupGCL across three evaluation settings. (i) **Pre-training phase**: "original" indicates settings where patient GRNs and knockdown GRNs come from the same cancer type (e.g., Lung/Lung KD, Colorectal/Colorectal KD). "cross domain 1, 2" use breast knockdown GRNs as supervision for lung or colorectal patient GRNs, respectively. (ii) **Fine-tuning phase**: "original" indicates pre-training and fine-tuning on the same cancer type (Breast→Breast). "cross domain 1, 2" denote models pre-trained on lung or colorectal GRNs and fine-tuned on breast patient GRNs. (iii) **OOD setting**: "original" indicates training and testing on the same cancer type (e.g., Lung/Lung, Colorectal/Colorectal). "cross domain 1, 2" denote directly applying a breast-trained model to lung or colorectal patient GRNs.

Setting	Configuration	Hazard (c-index)	BP (Subset Acc.)
(i) Cross Domain at Pre-training Phase: patient / cell line (teacher graph)			
	original: Lung / Lung KD	0.627 ± 0.051	0.282 ± 0.037
	cross domain 1: Lung / Breast KD	0.633 ± 0.069	0.270 ± 0.060
	original: Colorectal / Colorectal KD	0.698 ± 0.085	0.262 ± 0.030
	cross domain 2: Colorectal / Breast KD	0.687 ± 0.092	0.257 ± 0.044
(ii) Cross Domain at Fine-tuning Phase: pre-trained model → downstream task			
	original: model `Breast`→Breast	0.650 ± 0.059	0.243 ± 0.052
	cross domain 1: model `Lung` → Breast	0.654 ± 0.075	0.248 ± 0.043
	cross domain 2: model `Colorectal` → Breast	0.632 ± 0.072	0.249 ± 0.030
(iii) OOD: cancer type of fine-tuning / prediction			
	original: Lung / Lung	0.627 ± 0.051	0.282 ± 0.037
	cross domain 1: Breast / Lung	0.603	0.163
	original: Colorectal / Colorectal	0.698 ± 0.085	0.262 ± 0.030
	cross domain 2: Breast / Colorectal	0.583	0.162

H ROBUSTNESS ANALYSIS OF SUPGCL

In this section, we evaluate the robustness of the proposed SupGCL method. We analyze three aspects: (a) scaling with respect to the number of pretraining samples, (b) the impact of the sample size of supervision graphs, and (c) robustness to noise in the estimated GRNs.

A. SCALING WITH RESPECT TO PRETRAINING DATA SIZE

To assess the effect of pretraining sample size, we progressively reduced the number of breast cancer patient GRNs and evaluated performance on downstream tasks. Specifically, we compared the original dataset with conditions using $1/2$ and $1/4$ of the samples, evaluating hazard prediction (graph-level task) and BP classification (node-level task) (Table 19).

The results show a consistent performance improvement in hazard prediction with increasing data size. In contrast, the impact of sample size was marginal for node-level BP classification, suggesting that node-level GCL methods, including SupGCL, can effectively learn node representations even from a limited number of graphs (in some cases a single graph).

Table 19: Performance as a function of patient sample size (Breast Cancer GRN).

Sample Setting	**Hazard (c-index)**	**BP (Subset Accuracy)**
original (N=1092)	0.650 ± 0.059	0.243 ± 0.052
1/2 sample (N=546)	0.640 ± 0.040	0.243 ± 0.026
1/4 sample (N=273)	0.631 ± 0.045	0.247 ± 0.038

B. EFFECT OF SUPERVISION GRAPH SAMPLE SIZE

Next, we examined the effect of reducing the number of knockdown samples used to construct supervision graphs. For breast cancer cell lines, we randomly subsampled the knockdown dataset from 8793 (original) $\rightarrow$ 4397 ($\simeq 1/2$ original) $\rightarrow$ 2199 ($\simeq 1/4$ original) and evaluated downstream task performance (Table 20).

Even with a reduction to one quarter of the original size, performance degradation was minimal. SupGCL targets GRNs with 975 genes, meaning that the 1/4 setting (2199 samples) corresponds to approximately two experiments per gene. This indicates that the method is robust to limited supervision and does not easily overfit, making it suitable for realistic data conditions.

Table 20: Performance under reduced supervision graph sample size (Breast Cancer GRN).

Supervision Sample Setting	**Hazard (c-index)**	**BP (Subset Accuracy)**
original (N=8793)	0.650 ± 0.059	0.243 ± 0.052
1/2 sample (N=4397)	0.656 ± 0.062	0.239 ± 0.034
1/4 sample (N=2199)	0.644 ± 0.053	0.245 ± 0.043

C. ROBUSTNESS TO NOISE IN ESTIMATED GRNS

In our framework, edge contributions from estimated GRNs are used as input features to the GNN. To evaluate robustness against noise in estimated GRNs, we performed additional experiments.

We employed Bayesian network inference for GRN estimation (see Appendix D for details), which is known to be sensitive to initialization and may introduce errors. To mitigate this, we estimated GRNs 1000 times and applied frequency-based filtering of edges. The current setting applies a cutoff of 5%, but we additionally evaluated thresholds of 3% and 10%. The results (Table 21) show only minor differences across thresholds, confirming that SupGCL is robust to uncertainty inherent in GRN estimation.

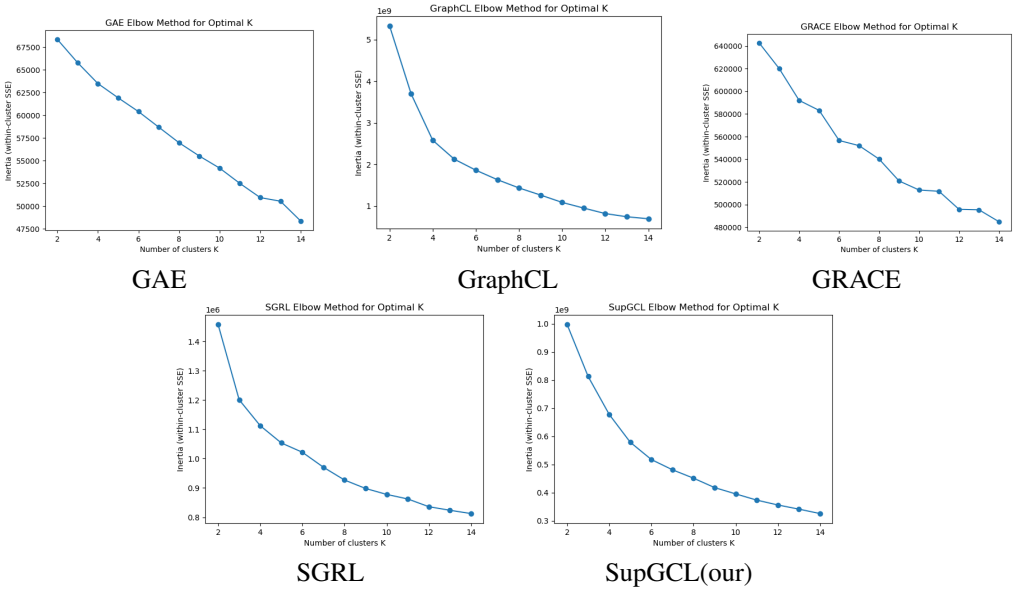

Figure 7: Elbows of k-means clustering

Table 21: Performance under different filtering thresholds in GRN estimation (Breast Cancer GRN).

Setting	**Hazard (c-index)**	**BP (Subset Accuracy)**
SupGCL (3% threshold)	0.665 ± 0.059	0.238 ± 0.029
SupGCL (5% threshold, original)	0.650 ± 0.059	0.243 ± 0.052
SupGCL (10% threshold)	0.646 ± 0.039	0.244 ± 0.029

I Biological Enrichment Analysis Using Latent Representations

In this section, we investigate the biological interpretability and validity of the gene embeddings learned by GCL models through enrichment analyses based on Gene Ontology (GO) and KEGG. We focus on breast cancer, whose molecular mechanisms are well characterized, and examine the embeddings obtained by our proposed method and baseline approaches. The analysis consists of three steps: (a) clustering in the latent space, (b) GO enrichment analysis for each cluster, and (c) KEGG pathway enrichment analysis.

A. Clustering in the Latent Space

We applied k-means clustering ($K = 7$) to the gene embeddings obtained from each method and performed GO:BP (Biological Process) and KEGG enrichment analyses using g:Profiler for each cluster. Since embeddings are computed for each node in each patient graph, we used patient-averaged gene embeddings for clustering. The number of clusters ($K = 7$) was determined based on the elbow criterion (see Fig. 7).

A.1 Clustering Results

The clustering results are shown in Table 22 and Fig. 8. SupGCL and GAE achieved lower Gini Index values compared to other methods, indicating more balanced cluster formation.

B. Gene Ontology Enrichment Analysis

For enrichment analysis, we primarily focus on GAE, a reconstruction-based baseline, and GRACE, which corresponds to an ablation of our proposed method. Clusters containing fewer than five genes were excluded from the analysis.

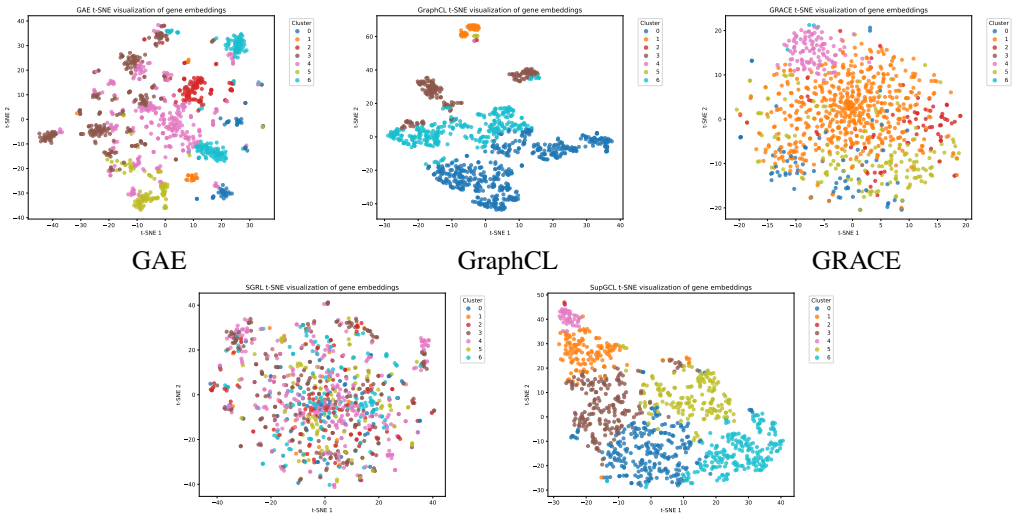

Figure 8: Gene-level embedding and the k-means clustering

Table 22: Clustering results (number of genes per cluster) for each method.

Cluster	GAE	GraphCL	GRACE	SGRL	SupGCL
Cl 0	53	482	73	658	251
Cl 1	22	44	556	11	130
Cl 2	96	1	70	129	2
Cl 3	262	151	1	1	192
Cl 4	293	3	102	25	37
Cl 5	105	6	169	1	160
Cl 6	144	288	4	150	203
Gini Index	0.375	0.632	0.594	0.699	**0.334**

B.1 GO Results

Table 23 reports the top 5 significantly enriched GO terms for each cluster and method.

Overall, SupGCL and GAE recovered significant functional modules in 6 out of 7 clusters, whereas GRACE succeeded in only 4 clusters. Notably, SupGCL captured distinct separation of autophagy-related processes: regulatory terms (Cl 3) versus execution processes (Cl 6). Cl 0 represented a composite cluster including organelle organization, macromolecule localization, and stress/catabolism. GAE revealed clear boundaries between ROS response, mitophagy, and other biological processes. In contrast, GRACE detected only a limited number of significant terms in most clusters, indicating lower resolution.

C. KEGG Pathway Enrichment Analysis

C.1 KEGG Results

The KEGG enrichment results are shown in Table 24. SupGCL identified breast cancer–related modules, capturing endocrine resistance, ER stress, and estrogen signaling in Cl 6, and separating FoxO/p53/metabolic pathways into Cl 5. GAE distinguished DNA damage response, cell cycle, and metabolism, reflecting clear functional boundaries. In contrast, GRACE grouped infection-related and metabolic pathways into broad clusters, showing lower resolution.

Table 23: GO enrichment results per cluster (Top 5 terms). "**skip**" indicates clusters with fewer than five genes (excluded), and "—" indicates no significant enrichment.

Cluster	GAE	GraphCL	GRACE	SGRL	SupGCL
Cl 0	Negative regulation of reactive oxygen species metabolic process; Negative regulation of intrinsic apoptotic signaling pathway; Intrinsic apoptotic signaling pathway; Regulation of reactive oxygen species metabolic process; Lymphoid progenitor cell differentiation	Mitotic cell cycle process; Cell cycle process; Cell cycle; Cellular response to stress; Mitotic cell cycle	Intrinsic apoptotic signaling pathway; Regulation of programmed cell death; Intrinsic apoptotic signaling pathway in response to DNA damage; Apoptotic process; Programmed cell death	Positive regulation of cellular process; Positive regulation of biological process; Cellular response to stress; Regulation of cellular component organization; Catabolic process	Organelle organization; Macromolecule localization; Chromosome organization; Catabolic process; Cellular response to stress
Cl 1	—	Antigen processing and presentation of exogenous peptide antigen via MHC class II; Catabolic process	Catabolic process; Cellular response to stress; Positive regulation of cellular process; Protein metabolic process; Regulation of metabolic process	Mitotic cell cycle process; Mitotic cell cycle; Cell cycle phase transition; Mitotic cell cycle phase transition; Cell cycle process	Cell cycle; Positive regulation of hydrolase activity; Cell cycle process; Regulation of cell cycle
Cl 2	Catabolic process; Autophagy of mitochondrion	**skip**	—	Cell cycle process; Cellular response to stress; Chromosome organization; Mitotic cell cycle process; DNA metabolic process	**skip**
Cl 3	Regulation of cellular component organization; Positive regulation of biological process; Cellular response to stress; Organelle organization; Positive regulation of cellular process	Programmed cell death; Cell death; Regulation of programmed cell death; Negative regulation of programmed cell death; Apoptotic process	**skip**	**skip**	Regulation of macroautophagy; Positive regulation of macroautophagy; Positive regulation of autophagy; Positive regulation of organelle assembly; Regulation of autophagy
Cl 4	Positive regulation of cellular process; Cell cycle; Positive regulation of biological process; Regulation of cell cycle; Cell cycle process	**skip**	Mitotic cell cycle process; Mitotic cell cycle; Cell cycle process; Cell cycle; Sister chromatid segregation	Mitotic sister chromatid segregation; Sister chromatid segregation; Mitotic nuclear division; Nuclear chromosome segregation; Mitotic cell cycle	Positive regulation of biological process; System development; Positive regulation of cellular process; Nervous system development; Regulation of multicellular organismal development
Cl 5	Cell migration; Organelle organization; Cellular response to stress; Positive regulation of cellular process	Regulation of calcium ion transport	Catabolic process; Establishment of protein localization; Protein transport	**skip**	Small molecule metabolic process; Small molecule biosynthetic process; Response to endogenous stimulus; Carboxylic acid metabolic process; Oxoacid metabolic process
Cl 6	Macromolecule localization; Establishment of protein localization; Protein transport; Protein localization; Protein localization to organelle	Positive regulation of cellular process; Positive regulation of biological process; Positive regulation of metabolic process; Catabolic process; Regulation of metabolic process	**skip**	Amino acid metabolic process; Catabolic process; Small molecule metabolic process; Apoptotic mitochondrial changes; Sulfur compound metabolic process	Catabolic process; Process utilizing autophagic mechanism; Autophagy; Macroautophagy; Regulation of programmed cell death

D. SUMMARY

In breast cancer samples, SupGCL demonstrated superior biological interpretability by: (i) separating autophagy into regulatory (Cl 3) and execution (Cl 6) processes in GO analysis, (ii) distinguishing endocrine resistance/ER stress (Cl 6) and FoxO/p53/metabolic modules (Cl 5) in KEGG analysis, and (iii) isolating DNA repair (Cl 0) as an independent cluster.

While GAE also delineated major biological functions, it failed to detect breast cancer–specific endocrine pathways as distinct clusters. GRACE exhibited limited resolution, with insufficient reflection of cancer-specific pathway differentiation. These results suggest that SupGCL provides more biologically meaningful and interpretable latent representations in the context of breast cancer.

Table 24: KEGG enrichment results per cluster (Top 5 terms). "**skip**" indicates clusters with fewer than five genes (excluded), and "—" indicates no significant enrichment.

Cluster	GAE	GraphCL	GRACE	SGRL	SupGCL
Cl 0	Amino sugar and nucleotide sugar metabolism	Cell cycle; Mismatch repair; Nucleotide excision repair; DNA replication; Terpenoid backbone biosynthesis	Kaposi sarcoma-associated herpesvirus infection; Epstein-Barr virus infection; Hepatitis B; Colorectal cancer; Human immunodeficiency virus 1 infection	MAPK signaling pathway; Epstein-Barr virus infection; Biosynthesis of nucleotide sugars; Valine, leucine and isoleucine degradation; Endocrine resistance	Mismatch repair
Cl 1	—	Lysosome	Protein processing in endoplasmic reticulum; Lysosome; Vibrio cholerae infection	Cell cycle; Motor proteins	—
Cl 2	—	**skip**	Protein processing in ER; Lysosome	—	**skip**
Cl 3	Mismatch repair; Endometrial cancer; Nucleotide excision repair; Platinum drug resistance; DNA replication	Metabolic pathways	—	**skip**	—
Cl 4	Cell cycle; p53 signaling pathway	**skip**	—	Cell cycle	—
Cl 5	—	—	Fructose and mannose metabolism	**skip**	FoxO signaling pathway; p53 signaling pathway; Insulin resistance; Metabolic pathways
Cl 6	Valine, leucine and isoleucine degradation; Arginine and proline metabolism	Colorectal cancer; Endometrial cancer; Pancreatic cancer; Breast cancer; Pathways in cancer	**skip**	Nucleotide excision repair	Protein processing in endoplasmic reticulum; Endocrine resistance; Estrogen signaling pathway; Vibrio cholerae infection

APPENDIX REFERENCES

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.

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, pp. 1–15. Springer, 2010.

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, pp. 127–135, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 978-1-58113-186-4.

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.

Peter Langfelder and Steve Horvath. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics*, 9(1):559, December 2008. ISSN 1471-2105.

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. ISSN 1471-2105.

Anna R Paolacci, Oronzo 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.

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. ISSN 2662-8457. Publisher: Nature Publishing Group.

Subramanian. GEO: Gene Expression Omnibus, GSE92742: L1000 phase I landmark gene expression profiles, 2017. Accessed: 2025-05-15.

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.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer, Remi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. *arXiv preprint arXiv:2102.06514*, 2021.

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. ISSN 2045-2322. Publisher: Nature Publishing Group.

UCSC Xena. TCGA TARGET GTEx study data. Available from UCSC Xena Platform, 2016. Accessed: 2025-05-15.

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*, pp. 035667, 2015.

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. ISSN 2405-5808.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, WWW '21, pp. 2069–2080. ACM, April 2021.