

---

# Supervised Graph Contrastive Learning for Gene Regulatory Network

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 Graph representation learning is effective for obtaining a meaningful latent space  
2 utilizing the structure of graph data and is widely applied, including biological  
3 networks. In particular, Graph Contrastive Learning (GCL) has emerged as a  
4 powerful self-supervised method that relies on applying perturbations to graphs for  
5 data augmentation. However, when applying existing GCL methods to biological  
6 networks such as Gene Regulatory Networks (GRNs), they overlooked meaningful  
7 biologically relevant perturbations, e.g., gene knockdowns. In this study, we  
8 introduce SupGCL (Supervised Graph Contrastive Learning), a novel GCL method  
9 for GRNs that directly incorporates biological perturbations derived from gene  
10 knockdown experiments as the supervision. SupGCL mathematically extends  
11 existing GCL methods that utilize non-biological perturbations to probabilistic  
12 models that introduce actual biological gene perturbation utilizing gene knockdown  
13 data. Using the GRN representation obtained by our proposed method, our aim  
14 is to improve the performance of biological downstream tasks such as patient  
15 hazard prediction and disease subtype classification (graph-level task), and gene  
16 function classification (node-level task). We applied SupGCL on real GRN datasets  
17 derived from patients with multiple types of cancer, and in all experiments SupGCL  
18 achieves better performance than state-of-the-art baselines.

## 19 1 Introduction

20 Graph representation learning has recently attracted attention in various fields to learn a meaningful  
21 latent space to represent the connectivity and attributes in given graphs [1]. Applications of graph  
22 representation learning are advancing in numerous areas where network data exists, such as analysis  
23 in social networks, knowledge graphs [2, 3], and biological network analysis in bioinformatics [4, 5].

24 Among these applications, the use of graph representation learning for Gene Regulatory Networks  
25 (GRNs), where each node and edge represents important intracellular functions and/or processes, is  
26 particularly significant in the fields of biology and drug discovery, as it is expected to contribute to  
27 identifying therapeutic targets and understanding disease mechanisms. GRN representation learning  
28 has been applied to tasks such as inferring transcription factors [6] and predicting drug responses in  
29 cancer cell lines [7].

30 With advancements in gene expression measurement and analysis technologies, identification methods  
31 for GRNs from expression data are also evolving. Traditional GRN identification constructs networks  
32 using statistical techniques applied to patient populations. Recently, it has become possible to  
33 construct GRNs specific to individual patients, highlighting distinct gene regulatory patterns compared  
34 to the population as a whole [8]. Hereafter in this paper, we refer to such individualized networks  
35 simply as GRNs. Similarly, in cell-based experiments such as gene knockdowns, it is now possible to  
36 estimate distinct GRNs for each experiment depending on the expression profile.

Graph representation learning applied to GRNs is believed to be effective for a wide range of biological applications. Among such methods, Graph Contrastive Learning (GCL) has gained traction. GCL enhances graph data via artificial perturbations applied to nodes or edges, and learns useful graph embeddings by maximizing the similarity between differently augmented views of the same graph [9]. For example, we can perform contrastive learning by artificial perturbations such as randomly removing nodes to augment the GRN and consequently learn a representation for the GRN.

However, a major challenge arises in that the artificial perturbations employed in conventional GCL methods significantly deviate from genuine biological perturbations, making it difficult to learn effective representations when applied to GRNs. A related issue occurs in heterogeneous node networks, where random perturbations to nodes or edges during augmentation can disrupt topology and attribute integrity, ultimately hindering representation learning [10, 11]. This problem is especially relevant to GRNs, where node heterogeneity exists—for example, in the presence of master regulators [12].

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 expression of specific genes is suppressed, representing biological perturbations that allow for the inference of GRNs. 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.

Our implementation and all experimental codes are available on <http://github.com/xxxxxx>.

## 2 Related Work

Graph Contrastive Learning (GCL) has inspired the development of numerous methods, largely based on the design of data augmentations and the construction of positive/negative sample pairs [10]. GCL methods can be broadly categorized into three types according to how they generate these training pairs: (1) graph-level pairs, (2) node-level pairs, and (3) cross-model pairs.

A representative method using graph-level pairs is GraphCL [13], which applies random data augmentations to graphs and treats the resulting two graph views as a positive pair. This approach enhances the model’s ability to capture graph-level representations. However, it has been pointed out that node-level information can become obscured in the process [14]. This limitation is especially problematic in applications like Gene Regulatory Networks (GRNs), where the semantics of individual nodes are crucial.

87 In contrast, methods such as GRACE, which generate node-level pairs, apply augmentations to graphs  
 88 and treat embeddings of different nodes as negative pairs during training [15]. This enables more  
 89 precise representation learning that captures local structure and attribute information at the node  
 90 level. Although GRACE has been applied to real GRNs [6], the augmentation strategies used do not  
 91 incorporate the specific biological characteristics of GRNs.

92 Beyond direct graph manipulations, methods like BGRL [16] generate positive pairs across mod-  
 93 els—between two instances of the same graph embedding model operating at different learning  
 94 speeds—rather than relying on heuristic graph augmentations. Such cross-model pairing strategies  
 95 have attracted attention for their ability to learn graph representations in a more natural manner.  
 96 Notably, the recently proposed SGRL [17] achieves stable and high-performance self-supervised  
 97 learning by using a pair of models: one that distributes node embeddings uniformly on a hypersphere,  
 98 and another that incorporates graph topology information. However, these methods are constrained  
 99 by their inability to design task-specific perturbations, and their applicability to biological networks  
 100 such as GRNs remains unexplored and unverified.

### 101 3 Preliminaries

#### 102 3.1 Background of Graph Contrastive Learning

103 Although there are various definitions of contrastive learning, it can be expressed using a probabilistic  
 104 model based on KL divergence over pairs of augmentations or node instances [18]. Let  $\mathcal{X}$  denote a  
 105 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  
 106 follows:

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

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

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

$$p(j|i) \triangleq \delta_{ij}, \quad q_{\phi}^{a,b}(j|i) \triangleq \frac{\exp(\text{sim}(z_i^a, z_j^b)/\tau_n)}{\sum_{k \in \mathcal{V}} \exp(\text{sim}(z_i^a, z_k^b)/\tau_n)}. \quad (2)$$

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

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

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

123 This encourages the embeddings at the node level  $z_i^a$  and  $z_i^b$  of the same node under different  
 124 augmentation operations to be close to each other. Typically, augmentation operations  $a, b$  are chosen  
 125 by uniform sampling from a set of candidates  $\mathcal{A}$ . Hence, in practice, the expected value is minimized  
 126 under the uniform distribution  $\mathcal{U}_{\mathcal{A}}$  over  $\mathcal{A}$ :

$$\text{Loss}_{\text{node}} \triangleq \mathbb{E}_{a,b \sim \mathcal{U}_{\mathcal{A}}} [\text{Loss}_{\text{node}}^{a,b}]. \quad (4)$$

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  $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E}, X^{\mathcal{V}}, X^{\mathcal{E}})$  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.  $X_i^{\mathcal{V}}$  is the feature of the  $i$ -th gene, and  $X_i^{\mathcal{E}} \in \mathbb{R}^{|\mathcal{E}|}$  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)  $f_\phi$ . Defining embedded representations through the GNN  $f_\phi$  as

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

where  $z_i^a$  and  $y_i^a$  denote the embedding vectors of the  $i$ -th node in  $Z^a$  and  $Y^a$ , respectively, and  $d$  is the embedding dimension. Note that the same GNN  $f_\phi$  is used to produce both  $Z^a$  and  $Y^a$ .

In this work, we train the neural network  $f_\phi$  using the set of pairs  $\{(Z^a, Y^a)\}_{a \in \mathcal{K}}$ , where  $(Z^a, Y^a)$  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  $\{(Z^a, Y^a)\}_{a \in \mathcal{K}}$ , 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:

$$p_\phi(b|a) \triangleq \frac{\exp(\text{sim}_F(Y^a, Y^b)/\tau_a)}{\sum_{c \in \mathcal{K}} \exp(\text{sim}_F(Y^a, Y^c)/\tau_a)}, \quad q_\phi(b|a) \triangleq \frac{\exp(\text{sim}_F(Z^a, Z^b)/\tau_a)}{\sum_{c \in \mathcal{K}} \exp(\text{sim}_F(Z^a, Z^c)/\tau_a)}, \quad (6)$$

where  $\text{sim}_F(\cdot, \cdot)$  denotes the Frobenius inner product, and  $\tau_a > 0$  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  $\{Y^a\}_{a \in \mathcal{K}}$ . Both probability models  $p_\phi$  and  $q_\phi$  are parameterized by the same GNN  $f_\phi$ .

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

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

Minimizing this loss reduces the discrepancy in embedding distributions between the artificially augmented graphs and the biologically grounded knockdown graphs. However, if both  $p_\phi$  and  $q_\phi$

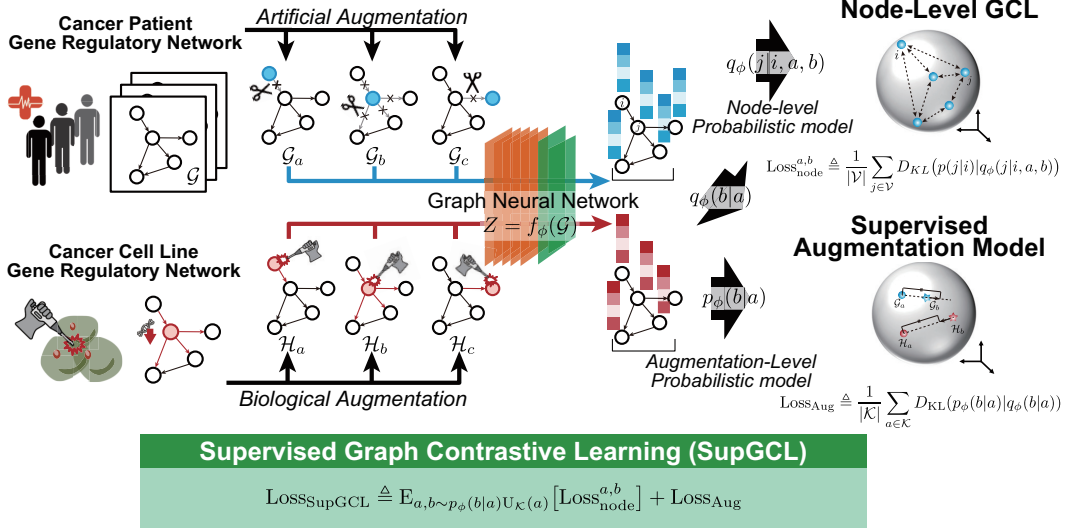


Figure 1: **Schematic overview of SupGCL (Supervised Graph Contrastive Learning).** The proposed method leverages two complementary types of graph augmentations. First, it generates artificially perturbed GRNs by simulating gene knockdowns, where node and edge features are masked based on the targeted gene. Second, it incorporates biologically grounded augmentations derived from real gene knockdown experiments conducted on cancer cell lines, serving as teacher GRNs. Embeddings are extracted using a shared GNN, and both node-level and augmentation-level contrastive losses are computed via KL divergence. This biologically grounded contrastive framework enables more faithful and effective representation learning of GRNs.

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  $\text{Loss}_{\text{Aug}}$  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 target 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:

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

Furthermore, the following theorem shows that by assuming independence between nodes and augmentation operations in the reference distribution  $p_\phi$ , we can avoid the trivial solution:

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

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

**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.  $\square$

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

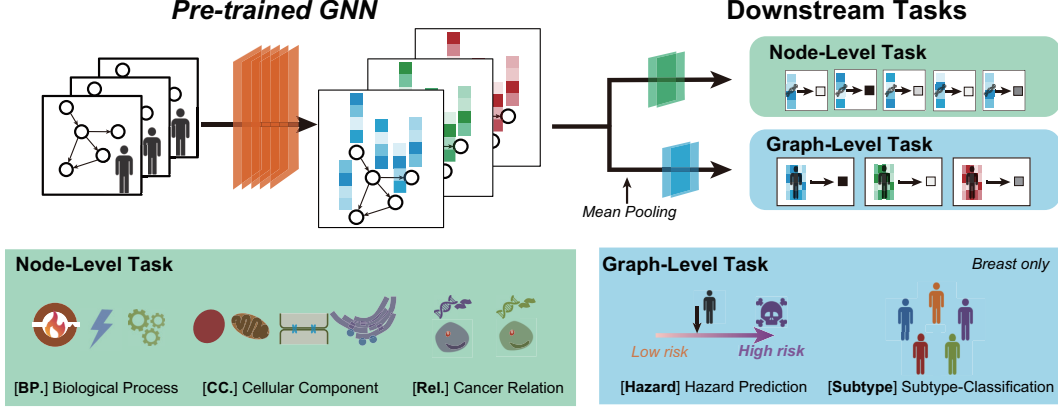


Figure 2: **Overview of Downstream Tasks Used for Benchmark of GRNs Across Three Cancer Types: Breast, Lung, and Colorectal Cancers.** The node-level tasks involve classifying genes into functional categories such as Biological Process [BP.], Cellular Component [CC.], and cancer relevance [Rel.]. The graph-level tasks include patient-level survival risk prediction [Hazard] and disease subtype classification [Subtype], the latter being specific to breast cancer. Mean pooling is applied to obtain graph-level representations.

Table 1: Description of downstream task

Task	Task Type	Metrics
<b>Node-Level Task</b>		
[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
<b>Graph-Level Task</b>		
[Hazard]: Hazard prediction	Survival analysis (1-dim risk score)	C-index
[Subtype]: Disease subtype prediction	Classification (5 groups)	Subtype accuracy

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  $\tau_n, \tau_a$  of each probability model. In particular, when the temperature parameter  $\tau_a$  involved in the augmentation operation is sufficiently large, the augmentation operation becomes independent of the teacher GRNs  $\{Y_a\}_{a \in \mathcal{K}}$ , and coincides with the conventional node-level GCL loss function.

**Corollary 1.**  $\lim_{\tau_a \rightarrow \infty} \text{LOSS}_{\text{SupGCL}} = \text{LOSS}_{\text{node}}$ .

**Proof:** As  $\tau_a \rightarrow \infty$ , we have  $p_\phi(b|a) \rightarrow U_{\mathcal{K}}(b)$  and  $q_\phi(b|a) \rightarrow U_{\mathcal{K}}(b)$ . Therefore, the expectation term becomes:  $\lim_{\tau_a \rightarrow \infty} E_{a,b \sim p_\phi(b|a)U_{\mathcal{K}}(a)}[\text{Loss}_{\text{node}}^{a,b}] = E_{a,b \sim U_{\mathcal{K}}}[\text{Loss}_{\text{node}}^{a,b}]$ ,  $\lim_{\tau_a \rightarrow \infty} D_{\text{KL}}(p_\phi(b|a)|q_\phi(b|a)) = 0$  thus proving the corollary.  $\square$

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

In this study, we formulated SupGCL by naturally extending the loss function of conventional GCL, based on a contrastive learning framework using KL divergence. Furthermore, we clarified the relationship between SupGCL and conventional GCL through the temperature parameter. This chapter verifies the effectiveness of the proposed method using actual gene regulatory networks (GRNs) from cancer patients and augmented GRNs based on gene knockdown experiments.



Table 2: Finetuning result of node-level downstream task.

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

Table 3: Finetuning result of graph-level downstream task.

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

## 5.1 Benchmark of Gene Regulatory Networks

**Evaluation Protocol:** We evaluated the proposed method through the following procedure. First, based on gene expression data from cancer patients, we constructed patient-specific GRNs. Similarly, we constructed teacher GRNs using 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. Finally, we visualized the latent representations at both the node and graph levels extracted from the trained models.

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

**w/o-pretrain** : Directly performs classification or regression for downstream tasks without any pre-training.

**GAE** : [19]: Graph representation learning method based solely on graph reconstruction.

**GraphCL** : [13]: Graph contrastive learning using positive pairs between graphs.

**GRACE** : [15]: Node-level graph contrastive learning method.

**SGRL** : [17]: Node-level GCL that leverages representation scattering in the embedding space.

**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 [20] and the LINCS dataset [21] are both large-scale and widely-used public platforms providing gene expression data from cancer patients and cell lines, respectively.

We used normalized count data from the TCGA TARGET GTEx study [22] provided by UCSC Xena. For LINCS, we used normalized gene expression data from the LINCS L1000 GEO dataset (GSE92742) [23].

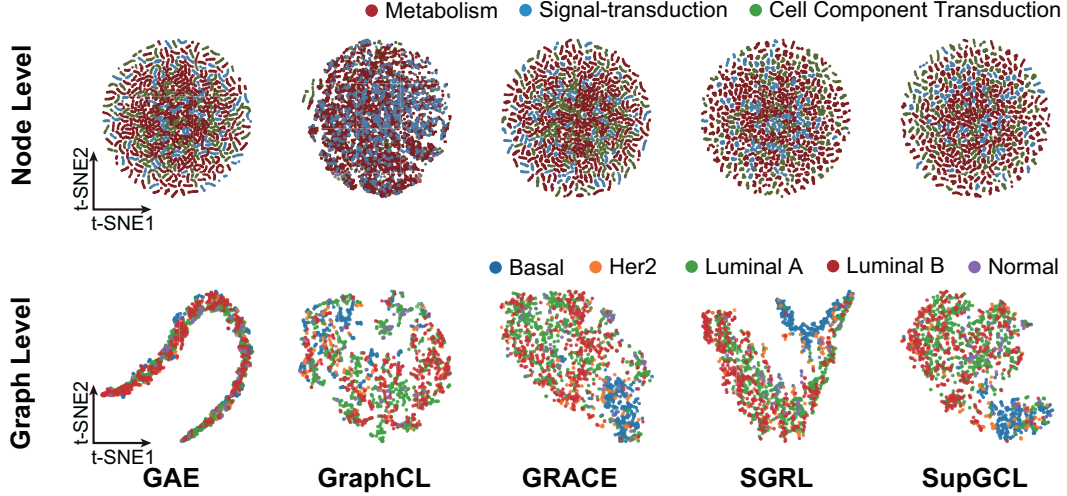


Figure 3: t-SNE visualization of pre-trained embeddings on breast cancer GRNs. The top row shows the node-level embedding space for individual genes, and the bottom row shows the corresponding graph-level readout features for each patient’s network.

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, 15926, and 11843, respectively. The number of unique knockdown target genes / total common genes was 768/975, 948/975, and 948/975.

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 [24] — Biological Process (metabolism, signaling, cell organization; 3 classes), and Cellular Component (nucleus, mitochondria, ER, membrane; 4 classes). We also used the OncoKB [25] 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 nonparametric regression with Gaussian noise [26]. For each experiment, gene expression values were used as node features, while edge features were defined as the product of estimated regression coefficients and the parent node’s gene expression [27]. 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 II.



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  $\pm$  standard deviation.

For all methods including the SupGCL and conventional methods, we used the same 5-layer Graph Transformer architecture [28]. Hyperparameters for pre-training were tuned with Optuna [29], and the model was optimized using the AdamW optimizer [30]. 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. This suggests that SupGCL effectively captures biologically meaningful GRN representations suitable for these tasks. In graph-level tasks like hazard and subtype prediction, while some existing methods showed marginal improvement over without-pretrain, SupGCL achieved significantly higher performance.

### 5.3 Result 2: Latent Space Analysis

Using the pre-trained models, we visualized the embedding spaces derived from the breast cancer dataset (see Figure 3). Other similar visualizations are shown in Appendix F.

The top row of Figure 3 shows the node-level embeddings. Colors represent one of the three single-label Biological Process annotations (metabolism, signaling, or cell organization). Across all models, no clear clustering was observed based on labels. However, GraphCL’s embeddings were notably different from others, with signs of latent space collapse. Detailed observations are provided in Appendix F. This suggests that GraphCL may be more suited to graph-level discrimination than node-level representation.

Compared to GraphCL, the other models showed more dispersed embeddings, though without distinct clustering.

The bottom row of Figure 3 illustrates the graph-level embeddings, colored by disease subtype. From this, it is evident that GAE and GraphCL fail to separate subtypes in the embedding space. In contrast, GRACE and SGRL showed moderate separation between the Basal subtype and others. SupGCL displayed the clearest separation, indicating its ability to better learn subtype-specific network representations.

## 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 limitation of this paper is that the effectiveness of the proposed method has only been confirmed in situations where the teacher GRNs were constructed from knockdown experiments of the same cancer type.

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

- [1] 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.

- [2] 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.
- [3] Xintao Shen and Yulai Zhang. A knowledge graph recommendation approach incorporating contrastive and relationship learning. *IEEE Access*, 11:99628–99637, 2023.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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*, pages 2069–2080, 2021.
- [11] 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*, pages 2397–2405, 2024.
- [12] 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.
- [13] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International conference on machine learning*, pages 12121–12132. PMLR, 2021.
- [14] 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.
- [15] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.
- [16] 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.
- [17] 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.
- [18] Shaden Alshammari, John Hershey, Axel Feldmann, William T Freeman, and Mark Hamilton. I-con: A unifying framework for representation learning. *arXiv preprint arXiv:2504.16929*, 2025.
- [19] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.
- [20] 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.

- 361 [21] Aravind Subramanian, Ramachandran Narayan, Simone M Corsello, and et al. A next generation connec-  
362 tivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6):1437–1452.e17, 2017.
- 363 [22] UCSC Xena. TCGA TARGET GTEx study data. Available from UCSC Xena Platform, 2016. Accessed:  
364 2025-05-15.
- 365 [23] Subramanian. GEO: Gene Expression Omnibus, GSE92742: L1000 phase I landmark gene expression  
366 profiles, 2017. Accessed: 2025-05-15.
- 367 [24] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Helen Butler, J Michael Cherry,  
368 Ana Davis, Kara Dolinski, Sally S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of  
369 biology. *Nature Genetics*, 25(1):25–29, 2000.
- 370 [25] Debyani Chakravarty, J Gao, S Phillips, R Kundra, H Zhang, J Wang, J E Rudolph, R Yaeger,  
371 T Soumerai, M Nissan, et al. Oncokb: A precision oncology knowledge base. *JCO Precision Oncology*,  
372 2017:PO.17.00011, 2017.
- 373 [26] Seiya Imoto, Takao Goto, and Satoru Miyano. Estimation of genetic networks and functional struc-  
374 tures between genes by using Bayesian networks and nonparametric regression. *Pacific Symposium on*  
375 *Biocomputing. Pacific Symposium on Biocomputing*, pages 175–186, 2002.
- 376 [27] Yoshihisa Tanaka, Yoshinori Tamada, Marie Ikeguchi, Fumiyoshi Yamashita, and Yasushi Okuno. System-  
377 based differential gene network analysis for characterizing a sample-specific subnetwork. *Biomolecules*,  
378 10(2), 2020.
- 379 [28] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label predic-  
380 tion: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*,  
381 2020.
- 382 [29] Takuya Akiba, Shotaro Sano, Tetsuo Yanase, Toshihiko Ohta, and Masanori Koyama. Optuna: A next-  
383 generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD Interna-*  
384 *tional Conference on Knowledge Discovery & Data Mining (KDD '19)*, pages 2623–2631, Anchorage,  
385 AK, USA, 2019. ACM.
- 386 [30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on*  
387 *Learning Representations (ICLR)*, 2019.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”.
- Keep the checklist subsection headings, questions/answers and guidelines below.
- Do not modify the questions and only use the provided macros for your answers.

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: Section 1

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Section 6

#### Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: Section 4

#### Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: Section 5

#### Guidelines:

- The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: I will post a link to Section 1 I75 when the camera-ready

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).



- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: Section 5

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: Section 5

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: Section 5 and supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We confirm.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Section 1

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks

Guidelines:

- The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Section 5

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

## 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: In the code to be published

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

## 14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

#### 15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: We did not handle data that required IRB approval.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

#### 16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Only translation and web search assistance.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.