# Rethinking Cancer Gene Identification through Graph Anomaly Analysis

**Anonymous submission**

## Abstract

Graph neural networks (GNNs) have shown promise in integrating protein–protein interaction (PPI) networks for identifying cancer genes in recent studies. However, due to the insufficient modeling of the biological information in PPI networks, more faithfully depiction of complex protein interaction patterns for cancer genes within the graph structure remains largely unexplored. This study takes a pioneering step toward bridging biological anomalies in protein interactions caused by cancer genes to statistical graph anomaly. We find a unique graph anomaly exhibited by cancer genes, namely weight heterogeneity, which manifests as significantly higher variance in edge weights of cancer gene nodes within the graph. Additionally, from the spectral perspective, we demonstrate that the weight heterogeneity could lead to the "flattening out" of spectral energy, with a concentration towards the extremes of the spectrum. Building on these insights, we propose the HIerarchical-Perspective Graph Neural Network (HIPGNN) that not only determines spectral energy distribution variations on the spectral perspective, but also perceives detailed protein interaction context on the spatial perspective. Extensive experiments are conducted on two reprocessed datasets STRINGdb and CPDB, and the experimental results demonstrate the superiority of HIPGNN. Our code and data are released at https://anonymous.4open.science/r/HIPGNN.

## Introduction

Identifying cancer genes is a crucial endeavor in both research and clinical practice (Beroukhim et al. 2010; Martínez-Jiménez et al. 2020; Bailey et al. 2018). Cancer genes are closely related with protein interactions (Leiserson et al. 2015), motivating solutions that integrate the protein–protein interaction (PPI) network for efficient identification (Yang et al. 2021; Levi, Elkon, and Shamir 2021; Chitra, Park, and Raphael 2022; Yang et al. 2023). Such approaches exploit, for example, multi-omics data and protein interaction information to extract and derive features that distinguish cancer genes.

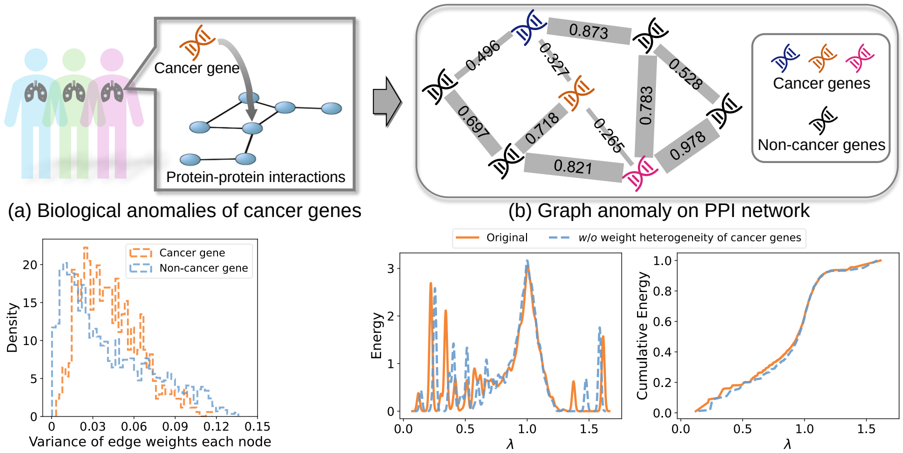
The aggregation capabilities of graph neural networks (GNNs) (Wu et al. 2020) have led to notable success in methods for cancer gene identification, based on graph convolutional networks (Schulte-Sasse et al. 2021), Chebyshev graph convolutional works (Peng et al. 2022) and masked graph autoencoders (Cui et al. 2023). However, these methods only use the PPI network to update the node features by referring to neighbor representations, which do not model the complete biological information within the network. Therefore there exists a gap: more faithfully depicting complex protein interaction patterns within the graph structure.

Our motivation is as follows: cancer genes induce significant biological anomalies in protein interactions, such as mutations, changes in expression levels, or alterations in protein modifications, as illustrated in Figure 1 (a). These biological anomalies can be interpreted as graph anomalies in the PPI network, as shown in Figure 1 (b). By identifying and analyzing these graph anomalies, we aim to develop a more comprehensive understanding of cancer gene behavior on PPI networks for cancer gene identification.

Based on this vision, our statistical experiments in this paper reveal a distinctive graph anomaly of cancer in the PPI network, which we term *weight heterogeneity*. As shown in Figure 1 (c), we compute the variance distribution of all edge weights (protein interaction confidence) for each node in a widely used PPI dataset, STRINGdb. It reveals that cancer genes exhibit greater weight variance compared to non-cancer genes.

Additionally, from the spectral perspective, we demonstrate that weight heterogeneity leads to the "flattening out" of the spectral energy, theoretically and experimentally. Figure 1 (d) illustrates the spectral energy distribution with and without weight heterogeneity in cancer nodes using the graph Fourier transform of node attributes. The spectral energy of the original graph (with weight heterogeneity) tends to concentrate more towards the extremes of the spectrum. We describe this phenomenon as the "flattening out" of spectral energy and provide rigorous proof through theoretical analysis. To further illustrate this phenomenon, we also validate it on two synthetic graphs. Based on the above observations, we recognize that both spatial and spectral perspectives offer information about graph anomaly of cancer. This motivates us to design a cancer gene identification model that integrates both perspectives.

Therefore, we propose an innovative HIerarchical-Perspective Graph Neural Network, termed HIPGNN, to identify cancer genes on the PPI network. HIPGNN can not only discern spectral energy distribution variations to tackle the "flattening out" on the spectral perspective, but also perceive detailed protein interaction context for handling weight heterogeneity on the spatial perspective. Specifically,



(a) Biological anomalies of cancer genes

(b) Graph anomaly on PPI network

(c) Weight heterogeneity

(d) "Flattening out" of spectral energy

Figure 1: Overview of our motivation. (a) Cancer genes induce significant biological anomalies in protein interactions. (b) We interpret these anomalies as graph anomaly in the PPI network. (c) Then, we calculate the variance distribution of the edge weights for each gene and investigate the weight heterogeneity of cancer genes from the spatial perspective. (d) Furthermore, we compute and compare the spectral energy distribution with and without weight heterogeneity in cancer nodes and explore the "flattening out" of spectral energy from the spectral perspective. We remove the weight heterogeneity of all cancer gene nodes by setting their edge weights to 0.5.

after obtaining the Laplace matrix eigenvalues, we encode the position and proximity of the eigenvalues to integrate the spectral energy distribution information. Following this, we design the proximity-aware spectral graph representation using spectral eigenvalue encoding to update node representations. Finally, we decode the spatial context for node representation by perceiving protein interaction information.

Building on previous works (Schulte-Sasse et al. 2021; Cui et al. 2023), we reprocessed two datasets, STRINGdb and CPDB, which contain real-world PPIs and cancer gene data, to extract more comprehensive protein interaction information. Extensive experiments on these datasets demonstrate the superior performance of the proposed HIPGNN compared to state-of-the-art methods.

# Preliminaries

## Theoretical analysis

we first provide several necessary definitions and notation.

**Weighted graph** We define a weighted graph as $\mathcal{G}_w = \{\mathcal{V}, \mathcal{E}, \mathcal{W}, \mathcal{X}, \mathcal{Y}\}$, where $v_i \in \mathcal{V}$ represents the node and $N = |\mathcal{V}|$. The node features and labels are denoted as $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, respectively. The edge $e_{ij} \in \mathcal{E}$ connects nodes $v_i$ and $v_j$, and $w_{ij} \in \mathcal{W}$ is the edge weight of $e_{ij}$. Let $A$ be the corresponding adjacency matrix, where $A_{ij} = w_{ij}$ if there exists a weighted edge. It is worth mentioning that all graphs studied in this paper are undirected graphs, i.e., $A_{ij} = A_{ji}$.

**Unweighted graph** An unweighted graph $\mathcal{G}_{uw} = \{\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{Y}\}$ is defined similarly to a weighted graph, except that in its adjacency matrix $A$, $A_{ij} = 1$ if there exists an edge.

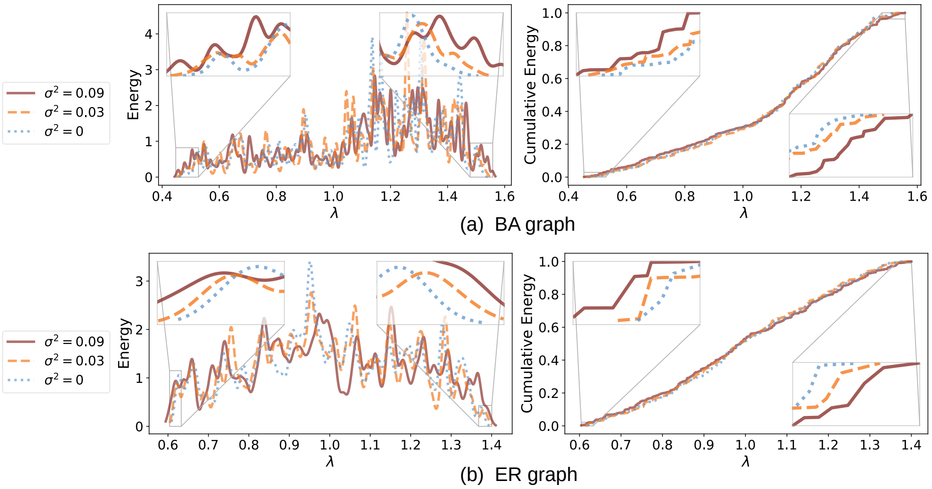
Given the weight heterogeneity exhibited by cancer genes on the PPI network, we model this phenomenon using a random weighted graph (Khorunzhy, Shcherbina, and Vengerovsky 2004; Ding and Jiang 2010). Specifically, for an unweighted graph $\mathcal{G}$, we define a set of variables $\{w_{ij}; 1 \leq i < j \leq N\}$ that are independently and identically Gaussian distributed, while assigning the same weight to the symmetric edge weights, making $\mathcal{G}$ a weighted graph. For all $i, j$, $w_{ij} = w_{ji}$, $\mathbb{E}(w_{ij}) = \mu$, and $\mathrm{Var}(w_{ij}) = \sigma^2$. Based on this, we use $\sigma^2$ to measure the degree of weight heterogeneity. Holding $\mu$ constant, we argue that the larger the $\sigma^2$, the higher the weight heterogeneity on the graph.

On the weighted graph $\mathcal{G}$, let $D$ be the diagonal degree matrix. The Laplacian matrix $L$ is defined $L = D - A$ (regular) or $L = I - D^{-1/2}AD^{-1/2}$ (normalized), where $I$ is the identity matrix. $L$ is a symmetric matrix with eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ and a corresponding orthonormal basis of eigenvectors $U = (u_1, u_2, \cdots, u_N)$. Assume that $x = (x_1, x_2, \cdots, x_N)$ is a random signal whose graph Fourier transform is $\hat{x} = U^T x = (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_N)$. The spectral energy distribution at $\lambda_k$ is denoted as $f_k(x, L) = \hat{x}_k^2 / \sum_{i=1}^{N} \hat{x}_i^2$. We summarize the following finding from the theory: ***The weight heterogeneity observed among cancer genes results in "flattening out" of spectral energy****, which means that spectral energy is elevated at extremes and lowered in the middle.*

To verify the finding theoretically, we first provide some definitions of the spectral energy distribution:

**Definition 1.** Expectation of spectral energy. *For $\lambda \in \lambda_1, \lambda_2, \cdots, \lambda_N$, we define the expectation of the spectral energy on $\lambda$ as:*

$$\mathbb{E}_\lambda(f(x, L)) = \frac{\sum_{k=1}^{N} \lambda_k \hat{x}_k^2}{\sum_{k=1}^{N} \hat{x}_k^2}.$$



(a) BA graph

(b) ER graph

Figure 2: The distributions of spectral energy and cumulative energy on two synthetic graphs: Barabasi–Albert (BA) graph and Erdős–Rényi (ER) graph. We measure the effect of weight heterogeneity with different weight variances $\sigma^2$. Red solid line means $\sigma^2 = 0.09$, orange dashed line means $\sigma^2 = 0.03$, and blue dotted line means $\sigma^2 = 0$.

And Definition 1 can also be converted into the form of Rayleigh quotient (Dong, Zhang, and Wang 2023; Gao et al. 2023):

$$\mathbb{E}_\lambda(f(x, L)) = \frac{\sum_{k=1}^{N} \lambda_k \hat{x}_k^2}{\sum_{k=1}^{N} \hat{x}_k^2} = \frac{x^T L x}{x^T x} \tag{1}$$

$$= \frac{1}{2} \frac{\sum_{i,j=1}^{N} (x_i - x_j)^2 w_{ij}}{\sum_{i=1}^{N} x_i^2}. \tag{2}$$

Equation (2) bridges the energy distribution in the spectral domain with the smoothness of the signal on the graph structure in the spatial domain. It can be seen that if the signal is less smooth, the spectral energy moves to higher points. Further, we define the variance of the spectral energy distribution.

**Definition 2.** Variance of spectral energy. *For* $\lambda \in \lambda_1, \lambda_2, \cdots, \lambda_N$*, The variance of spectral energy on* $\lambda$ *is defined as:*

$$\mathrm{Var}_\lambda(f(x, L)) = \frac{\sum_{k=1}^{N} \lambda_k^2 \hat{x}_k^2}{\sum_{k=1}^{N} \hat{x}_k^2} - \left( \frac{\sum_{k=1}^{N} \lambda_k \hat{x}_k^2}{\sum_{k=1}^{N} \hat{x}_k^2} \right)^2.$$

A larger variance indicates that the spectral energy disperses more to both sides of the spectrum. Up to this point, we will now state how the weight heterogeneity on the graph structure affects the variance of spectral energy.

**Proposition 3.** *Give* $L = D - A$ *and* $\{w_{ij} \sim \mathcal{N}(\mu, \sigma^2), w_{ij} = w_{ji}; 1 \leq i < j \leq N\}$*, the expectation of variance of spectral energy with respect to* $w$*,* $\mathbb{E}_w(\mathrm{Var}_\lambda(f(x, L)))$*, monotonically increases with the variance of edge weights* $\sigma^2$*.*

The details of the proof process we put in the technical appendix. Proposition 3 illustrates that a larger variance in the edge weights (weight heterogeneity) on the graph leads to a broader dispersion ("flattening out") of the spectral energy. Intuitively, disrupting edge weights affects functional connectivity metrics such as effective resistance (Ghosh, Boyd, and Saberi 2008), which in turn affects the upper and lower bounds of spectral energy distribution (Barooah and Hespanha 2006).

## Validation on synthetic graphs

To illustrate our theoretical findings more intuitively, we investigate the "flattening out" of spectral energy on Barabasi–Albert (BA) (Albert and Barabási 2002) and Erdős–Rényi (ER) (Erdős et al. 2012) graphs, each with 500 nodes. The BA graph models real-world network properties, while the ER graph has a uniform degree distribution.

To add weight heterogeneity to original graphs, we assign Gaussian independently distributed weights $w_{ij} \sim \mathcal{N}(\mu, \sigma^2)$ to all edges, with $w_{ij} = w_{ji}$. The $\sigma^2$ is set to 0.09, 0.03, and 0 (unweighted graph). Given that even with the largest variance, there is only a probability of less than $0.001\%$ for a weight to fall outside the range of $[0, 2]$, we confine the weight values to this interval to ensure realistic weight values. The graph signal is uniformly distributed between 0 and 1.

As shown in Figure 2, we compute and plot the spectral energy distribution (KDE) and the spectral cumulative energy $\eta_k(x, L) = \sum_{i=1}^{k} \hat{x}_i^2 / \sum_{i=1}^{N} \hat{x}_i^2$ on the two synthetic graphs. For clarity, we omit the small energy at $\lambda_1 = 0$ and provide a magnified view of the spectrum's extremes. A clear "flattening out" of spectral energy is observed on both graphs. We summarize the following observations: (1) For the spectral energy distribution, a larger $\sigma^2$ causes the spectral energy to deviate from the middle and to have lower wave peaks, with a regular arrangement according to $\sigma^2$ size at both ends of the spectrum. (2) For the spectral cumulative energy distribution, an increase in $\sigma^2$ leads to elevated energy in the low-frequency range and reduced energy in the high-frequency range. These trends are particularly pronounced at both ends of the spectrum. Overall, these observations

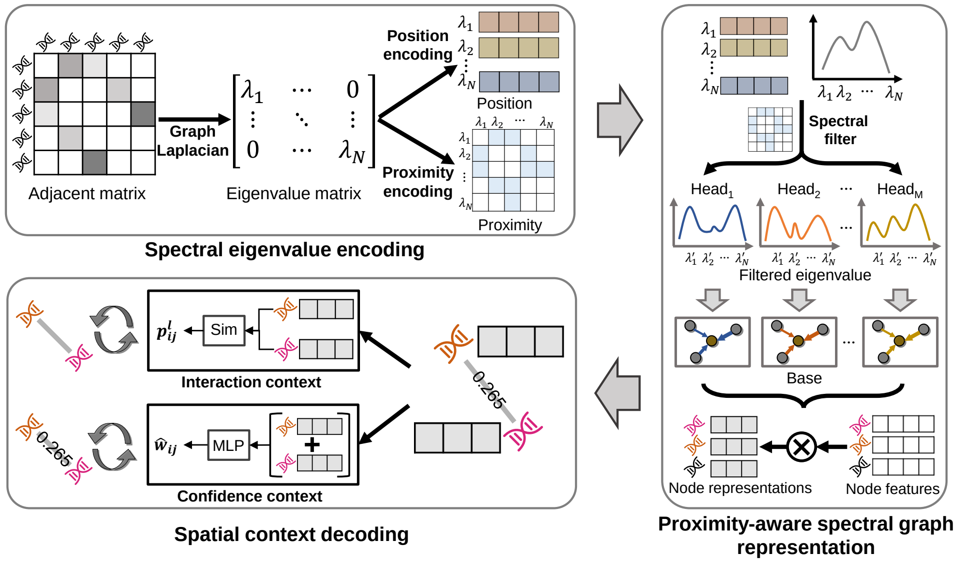


Figure 3: The overview of the HIPGNN framework. It comprises three modules: the spectral eigenvalue encoding to encode the position and proximity of eigenvalues; the proximity-aware spectral graph representation to fuse eigenvalue position and proximity encoding with spectral filters and get the node representation; the spatial context decoding for perceiving protein interaction information.

empirically substantiate the Proposition 3 and provide intuitive insights into the effects of varying $\sigma^2$ on spectral energy distribution.

## Problem formulation

So far, we have formulated the PPI network-based cancer gene identification problem. Existing methods (Schulte-Sasse et al. 2021; Peng et al. 2022; Cui et al. 2023) treat protein interactions above a certain confidence threshold as unweighted edges, constructing an unweighted graph. In contrast, we use confidence as edge weights to construct a weighted graph, capturing variations in confidence levels and their correlation with cancer genes in the PPI network.

**PPI network based cancer gene identification** This task is regarded as a semi-supervised node classification task. Given a weighted graph $\mathcal{G}_w$ based on a PPI network and some nodes with known labels, our goal is to infer the labels of the remaining nodes, determining whether they are cancer genes.

# Method

Based on the analysis in Preliminaries, cancer genes exhibit a unique graph anomaly, i.e. weight heterogeneity, in the PPI network and show a "flattening out" phenomenon in the spectral energy distribution. To address the anomaly from both spectral and spatial perspectives simultaneously, we introduce a hierarchical-perspective graph neural network, termed HIPGNN, for cancer gene identification as shown in Figure 3.

## Spectral eigenvalue encoding

Most polynomial filter-based spectral GNNs (Defferrard, Bresson, and Vandergheynst 2016; He, Wei, and Wen 2022; He et al. 2021; Wang and Zhang 2022) use a fixed polynomial basis for all eigenvalues to approximate arbitrary filters. Nevertheless, these scalar eigenvalue computation methods fall short of expressive capability and cannot capture the "flattening out" of the spectral energy well. To tackle this issue, we intend to design a more powerful eigenvalue encoding rule to directly reflect the distribution of eigenvalues, such as the spectral gap (Hoffman, Kahle, and Paquette 2021).

**Eigenvalue position encoding** Given a normalized Laplace matrix $L = U\Lambda U^T$ of a weighted graph $\mathcal{G}_w$, we encode each eigenvalue of the matrix $\lambda \in \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ from a scalar to a meaningful vector: $\mathbb{R}^1 \rightarrow \mathbb{R}^d$, by using a position encoding function as follows:

$$\begin{aligned}\rho_{2i}(\lambda) &= \sin(100\lambda/10000^{2i/d}),\\ \rho_{2i+1}(\lambda) &= \cos(100\lambda/10000^{2i/d}),\end{aligned} \tag{3}$$

where $i$ is an integer and its value domain ranges from 0 to $d/2-1$. This function forms a multiscale representation of the eigenvalues and has the advantage of filtering arbitrary multivariate continuous functions(Bo et al. 2022).

**Eigenvalue proximity encoding** Furthermore, to intuitively perceive the spectral energy distribution, we propose to encode the proximity between eigenvalues. A proximity matrix is computed by eigenvalue position encodings: $\mathbb{R}^{N\times d} \rightarrow \mathbb{R}^{N\times N}$. Each element of this matrix obtained is as follows:

$$R_{ij} = \rho(\lambda_i)^T \rho(\lambda_j), \tag{4}$$

where there exist the following two theoretical properties of $R_{ij}$ that can be proved.

**Proposition 4.** *The proximity between $\lambda_i$ to $\lambda_j$, $R_{ij}$, is determined by $\lambda_i - \lambda_j$.*

**Proposition 5.** *The $R_{ij}$ is undirected.*

The two propositions (proof in the technical appendix) illustrate that proximity matrix $R_{ij}$ can effectively capture and represent the spectral energy distribution variations, which further enables the GNN to process the spectral energy "flattening out".

## Proximity-aware spectral graph representation

After generating the valuable eigenvalue encoding, we utilize it in the spectral graph representation. Then we propose a Transformer-based(Vaswani et al. 2017) proximity-aware spectral graph representation to fuse eigenvalue position and proximity encoding with spectral filters and get the node representation.

**Proximity-aware spectral filter** Unlike using a regular transformer to design spectral filters (Bo et al. 2022), we introduce the eigenvalue proximity information to the attention computation process for designing trainable spectral filters. Given the initial representation which concatenates eigenvalues with their encodings:$Z = [\lambda_1 \| p(\lambda_1), \cdots, \lambda_n \| p(\lambda_n)]^T \in \mathbb{R}^{N \times (d+1)}$, an innovative attention computation function is proposed as follows:

$$\begin{aligned} &Q = ZW_m^Q, \quad K = ZW_m^K, \quad V = ZW_m^V, \\ &Z'_m = \text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T + R}{\sqrt{d_q}})V, \end{aligned} \tag{5}$$

where $d_q$ is the dimension of each head, and $m$ represent the $m$-th head. We include the proximity matrix $R$ as part of attention for learning the global distribution of eigenvalues. Afterward, the representation $Z'_m$ of each head is used as a spectral filter to compute new eigenvalues as $\lambda'_m = \phi(Z'_m W_\lambda)$, where $\lambda'_m \in \mathbb{R}^{N \times 1}$ is the $m$-th eigenvalue vector after the spectral filtering.

**Learnable bases** After obtaining $M$ vectors of filtered eigenvalues, we use a feed-forward network (FFN) in the standard Transformer layer to create the learnable bases for graph convolution. The reconstruction and concatenating processes can be formulated as follows:

$$S_m = U\text{diag}(\lambda'_m)U^T, \quad \hat{S} = \text{FFN}([I_N \| S_1 \| \cdots \| S_M]), \tag{6}$$

where $I_N \in \mathbb{R}^{N \times N}$ denotes the unit matrix.

**Graph convolution** Eventually, we regard each dimension of the node features as a graph signal and multiply it with the combined Laplace matrix base $\hat{S}$:

$$\hat{X}^{l-1}_{:,i} = \hat{S}_{:,:,i} X^{l-1}_{:,i}, \quad X^l = \sigma(\hat{X}^{l-1} W_x^{l-1}) + X^{l-1}, \tag{7}$$

where $\sigma()$ is activation and $X^l$ is the node representation in the $l$-th layer.

## Spatial context decoding

Recalling our findings, in addition to the "flattening out" of the spectral energy, we also observe weight heterogeneity within the weighted graph. This indicates that the information about the protein interaction context over the spatial domain is also helpful in distinguishing such anomalies. Motivated by this hypothesis, we decode the protein interaction context to correlate different nodes on graph data. Therefore, after obtaining the node representations, we design the spatial context decoding module to perceive the protein interaction and confidence information in the spatial domain.

**Interaction context perception** Given node representations $X^l$, we compute the interaction probability between $x_i^l$ and $x_j^l$ by cosine similarity and then leverage cross entropy to approximate the interactions on the graph:

$$\begin{aligned} p_{ij}^l &= \cos(x_i^l, x_j^l), \\ \mathcal{L}_l &= \sum_{(i,j) \in \hat{\mathcal{E}}} (y_{ij}^l \log(p_{ij}^l) + (1 - y_{ij}^l)\log(1 - p_{ij}^l)), \end{aligned} \tag{8}$$

where the set $\hat{\mathcal{E}}$ contains the edges $\mathcal{E}$ on the graph and the negatively sampling edges from the original dataset. if $(i, j)$ is negatively sampling edge, $y_{ij}^l = 0$.

**Confidence context perception** More importantly, the model needs to perceive protein interaction confidence to tackle weight heterogeneity. Given the node representations $X^w$, the MLP model and MSE are utilized to predict confidence scores between $x_i^w$ and $x_j^w$ as well as to compute losses, respectively:

$$\begin{aligned} \hat{w}_{ij} &= \text{MLP}((x_i^w + x_j^w)/2), \\ \mathcal{L}_w &= \sum_{(i,j) \in \hat{\mathcal{E}}_{\text{train}}} \text{MSE}(\hat{w}_{ij}, w_{ij}). \end{aligned} \tag{9}$$

In $(i, j) \in \hat{\mathcal{E}}$, if $(i, j)$ is negatively sampling edge, $w_{ij} = 0$.

It is worth mentioning that we set up node representing channels independent of cancer gene identification for the above two perception modules. We use multiple standard transformer models to obtain separate node representations for each channel: $X^n$, $X^l$, and $X^w$.

**Cancer gene identification** Here, we proceed with the loss function for cancer gene identification. We feed $X^n$ to the MLP with sigmoid function to get the cancer gene node probability $p^n$. The weighted cross-entropy loss is used to alleviate the challenge from label imbalance as follows:

$$\mathcal{L}_n = \sum_{i \in \mathcal{V}_{train}} (\gamma y_i \log p_i^n + (1 - y_i)\log(1 - p_i^n)), \tag{10}$$

where $\mathcal{V}_{train}$ is the training set of nodes $\mathcal{V}$, and $\gamma$ is the ratio of cancer gene nodes ($y_i = 1$) to non-cancer gene nodes ($y_i = 0$) in the training set. At last, we sum all the losses with weights to get the total loss:$\mathcal{L} = \alpha\mathcal{L}_n + \beta\mathcal{L}_l + \gamma\mathcal{L}_w$.

## Complexity analysis

Considering the large size of the graph, we intend to use only a few important eigenvalues as inputs to the model in

| Graph | Method | STRINGdb | | | | | | CPDB | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 20% | | | 80% | | | 20% | | | 80% | | |
| | | AUC | F1 | AP | AUC | F1 | AP | AUC | F1 | AP | AUC | F1 | AP |
| Unweighted | GCN | 81.68 | 72.09 | 64.16 | 87.99 | 77.43 | 75.61 | 81.84 | 73.54 | 67.55 | 82.48 | 73.67 | 69.16 |
| | GAT | 81.67 | 71.62 | 58.88 | 85.25 | 73.35 | 68.52 | 80.16 | 70.66 | 61.02 | 84.50 | 76.26 | 68.70 |
| | GraphSAGE | 84.37 | 73.75 | 65.48 | 87.09 | 78.98 | 72.13 | 78.02 | 69.32 | 62.99 | 85.96 | 78.36 | 76.31 |
| | Chebnet | 83.35 | 73.20 | 66.47 | 86.44 | 78.33 | 73.39 | 75.70 | 66.77 | 59.29 | 82.00 | 71.72 | 68.32 |
| | EMOGI | 79.06 | 64.27 | 59.74 | 86.88 | 70.18 | 73.22 | 80.84 | 67.10 | 66.94 | 80.84 | 68.25 | 64.00 |
| | MTGCN | 84.30 | 73.97 | 66.82 | 86.90 | 76.05 | 73.89 | 77.25 | 69.62 | 60.95 | 83.71 | 68.84 | 70.22 |
| | SMG | **89.81** | 78.62 | 75.69 | 90.80 | 79.74 | 77.43 | 84.75 | 72.34 | 70.83 | 86.57 | 77.77 | 77.83 |
| | HIPGNN | 89.08 | 78.17 | 75.56 | 90.81 | 81.71 | 79.66 | **87.99** | 79.13 | 78.07 | 87.80 | 79.87 | 77.40 |
| Weighted | GCN | 81.65 | 72.46 | 63.91 | 87.17 | 76.25 | 74.22 | 81.79 | 73.91 | 67.04 | 82.97 | 73.70 | 68.93 |
| | GAT | 77.99 | 69.93 | 54.49 | 85.85 | 74.50 | 72.67 | 74.48 | 42.19 | 57.71 | 85.51 | 75.69 | 68.55 |
| | Chebnet | 83.64 | 73.91 | 66.58 | 87.17 | 78.03 | 74.72 | 76.00 | 66.32 | 59.81 | 83.35 | 74.34 | 69.82 |
| | HIPGNN | 88.39 | **79.60** | **76.13** | **91.18** | **83.33** | **81.21** | 87.88 | **79.38** | **78.13** | **89.66** | **80.91** | **79.71** |

Table 1: Performance on the two datasets under different percentages of the training data. (%)

order to greatly reduce the computational complexity. By analyzing the spectral energy distribution, we believe that the eigenvalues at both extremes are more effective in encoding the “flattening out” of the spectral energy.

Therefore, we introduce a hyperparameter $q$ to adjust for only considering the first $q$ small and the last $q$ large eigenvalues. Ultimately, the complexity of HIPGNN is $\mathcal{O}(2q^2d_1 + 2q^2M + Nd_1L + Nd_1^2 + N^2d_2 + 2E^2d_2 + Nd_2^2 + 2Ed_2^2)$, where $N$ and $E$ are the nodes and edges of the weighted graph, $M$ and $L$ denote the number of filters and layers, and $d_1$ and $d_2$ represent the hidden dimensions of graph layer and node representation.

# Experiments

## Experimental setup

**Datasets** Based on previous works (Schulte-Sasse et al. 2021; Cui et al. 2023), we extract richer protein interaction information on two widely used PPI datasets with confidence (Szklarczyk et al. 2021; Kamburov et al. 2009), and integrate cancer gene data to construct two datasets. We name these two datasets directly after the PPI databases: STRINGdb and CPDB. Unlike previous works that used fixed threshold confidence to construct unweighted graphs, HIPGNN directly leverages protein confidence as edge weights to construct weighted graph.

**Metrics** We choose AUC, F1 (macro), and AP for model performance evaluation. AUC measures the area under the ROC curve, providing a global assessment across all classification thresholds. F1 (macro) is the unweighted average of F1 scores for both categories, suitable for imbalanced datasets. AP is the area under the precision-recall curve, and is considered the most important metric for cancer gene identification (Schulte-Sasse et al. 2021).

**Baselines** The baseline methods can be categorized into two groups: firstly, general GNN-based models including GCN (Kipf and Welling 2016), GAT (Veličković et al. 2018), GraphSAGE (Hamilton, Ying, and Leskovec 2017), and Chebnet (Defferrard, Bresson, and Vandergheynst 2016); and secondly, state-of-the-art cancer gene identification methods including EMOGI (Schulte-Sasse et al. 2021), MTGCN (Peng et al. 2022) and SMG (Cui et al. 2023). We also implement GCN, GAT, and Chebnet on weighted graph and HIPGNN on unweighted graph.

## Performance comparison

Table 1 presents the results of HIPGNN and other baseline methods with training ratios of 20% and 80%. From the table, we draw the following conclusions.

**Importance of spectral graph representation** Only Chebnet and HIPGNN outperform on weighted graphs compared to unweighted ones, highlighting that edge weights can negatively impact models like GCN, which function as low-pass filters. This demonstrates the effectiveness of appropriate spectral filters in addressing weight heterogeneity.

**Importance of spatial context** HIPGNN shows a significant performance boost at a 20% training ratio, particularly on the CPDB dataset, outperforming SMG by 7.30% in AP. This indicates that spatial context in protein interactions aids in identifying unknown cancer genes, especially when labels are sparse.

**Superiority of HIPGNN** HIPGNN consistently outperforms other models across most metrics, effectively handling weight heterogeneity to distinguish cancer genes. Notably, HIPGNN improves AP by 0.44% on STRINGdb and 7.30% on CPDB at a 20% training ratio, and by 3.78% on STRINGdb and 1.88% on CPDB at an 80% training ratio, compared to SMG.

Due to space constraints, subsequent experiments focus on the STRINGdb dataset, with CPDB results provided in the technical appendix.

## Ablation analysis

**Proximity-aware spectral graph representation** We examine the impact of the spectral graph representation module with spectral eigenvalue encoding in HIPGNN. We compare EGOGI, MTGCN, SMG, and HIPGNN (without spatial context decoding) using five-fold cross-validation at an 80% training ratio. The left subfigure of Figure 4 shows box plots of the results, where HIPGNN with only spectral graph representation still outperforms, highlighting the effectiveness of spectral eigenvalue encoding.

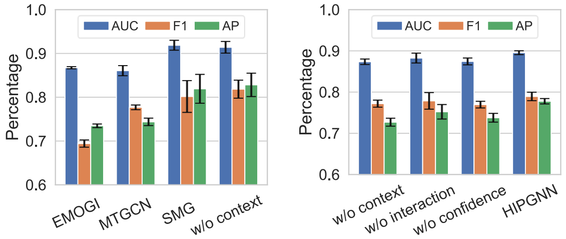


Figure 4: Ablation analysis of spectral graph representation (left) and spatial context decoding (right) on the STRINGdb dataset.

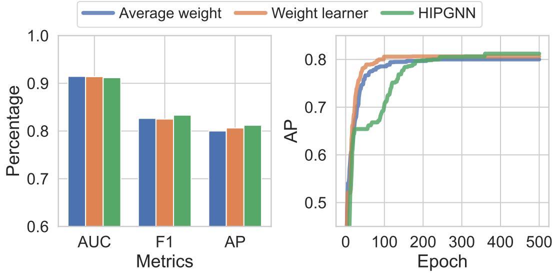


Figure 5: Comparison of the three model metrics as well as the variation of the best AP metric in the test set under three different loss weight schemes.

**Spatial context decoding** For decoding protein interaction contexts, we consider both interaction and confidence contexts for node representations. We evaluate the contribution of these contexts to HIPGNN using four variants: (1) Without context: removes both interaction and confidence contexts; (2) Without interaction: removes interaction context; (3) Without confidence: removes confidence context; (4) HIPGNN: the original model. The right subfigure of Figure 4, using a 20% training ratio and five-fold cross-validation, shows that both contexts improve HIPGNN's performance, with confidence context being particularly impactful.

## Parameter analysis

**Loss weights** For the final loss computation, we used $\alpha$, $\beta$, and $\gamma$ to weight the protein interaction context, interaction confidence context, and cancer gene label loss, respectively. We empirically set $\alpha = 0.01$ on STRINGdb and $\alpha = 0.02$ on CPDB, with $\beta = 2/3(1-\alpha)$ and $\gamma = 1/3(1-\alpha)$. To validate this, we compared two other schemes: Average weight (uniformly setting all weights to $1/3$) and Weight learner, which uses a Bayesian learnable loss function (Li et al. 2020; Peng et al. 2021): $\mathcal{L} = \frac{1}{\alpha_l^2}\mathcal{L}_n + \frac{1}{\beta_l^2}\mathcal{L}_l + \frac{1}{\gamma_l^2}\mathcal{L}_w + 2\log(\alpha_l\beta_l\gamma_l)$, where $\alpha_l$, $\beta_l$, and $\gamma_l$ are learnable parameters. Figure 5 shows the three model performance metrics and the best AP metric variation over 500 epochs under the three schemes. Empirical weights achieved the best results. Additionally, setting $\alpha$ smaller aids in the convergence of cancer gene labeling loss. We observed that Average weight and Weight learner fall into local optima early, while our scheme continues optimizing, with metrics possibly improving beyond 500 epochs.

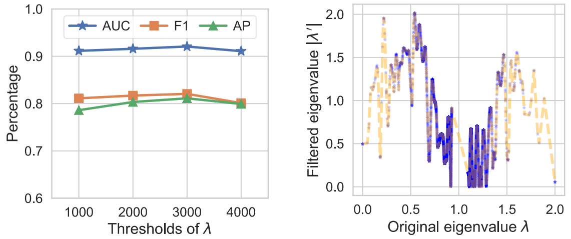


Figure 6: The eigenvalue thresholds analysis (left) and the spectral filter visualization (right).

**Thresholds of eigenvalue** We introduced the parameter $q$ to control the selection of the first $q$-small and last $q$-large eigenvalues input into the model, thereby regulating its complexity. We evaluated HIPGNN's performance with $q$ values ranging from 1,000 to 4,000. As shown in the left subfigure of Figure 6, setting $q$ to 3,000 yielded the best performance. Increasing $q$ beyond this point resulted in decreased performance, aligning with our initial observation that the spectral energy tends to ”flatten out” more at the spectrum's ends.

## Visualization of spectral filter

We applied a spectral filter to obtain the filtered eigenvalues in the proximity-aware spectral graph representation. In the right subfigure of Figure 6, we visualize the relationship between the filtered and original eigenvalues. The blue stars depict the distribution of eigenvalues, connected by a yellow dashed line. The figure shows that the spectral filter prioritizes eigenvalues near the spectrum's ends over those in the middle. This suggests that spectral eigenvalue coding effectively addresses the key phenomenon in PPI networks caused by cancer genes: the ”flattening out” of spectral energy.

## Conclusion

This work takes a pioneering step toward bridging significant biological anomalies in protein interactions caused by cancer genes to the statistical graph anomaly. We identify a unique graph anomaly in cancer genes, termed weight heterogeneity, which leads to the “flattening out” of spectral energy. In response, we propose a novel model, HIPGNN, for the identification of cancer genes.

**Broader impact.** This work has the potential to benefit both the bioinformatics and network science fields. It not only lays a new theoretical foundation for cancer gene identification but also offers a fresh perspective and direction for research in graph anomaly detection.

**Limitations.** The phenomenon of weight heterogeneity was observed only in cancer genes on two PPI networks. Further validation on other PPI networks is necessary to refine this observation. Additionally, exploring other real-world scenarios where weight heterogeneity occurs could provide more validation datasets for graph anomaly detection.

## References

Albert, R.; and Barabási, A.-L. 2002. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1): 47.

Bailey, M. H.; Tokheim, C.; Porta-Pardo, E.; Sengupta, S.; Bertrand, D.; Weerasinghe, A.; Colaprico, A.; Wendl, M. C.; Kim, J.; Reardon, B.; et al. 2018. Comprehensive characterization of cancer driver genes and mutations. *Cell*, 173(2): 371–385.

Barooah, P.; and Hespanha, J. P. 2006. Graph effective resistance and distributed control: Spectral properties and applications. In *Proceedings of the 45th IEEE Conference on Decision and Control*, 3479–3485. IEEE.

Beroukhim, R.; Mermel, C. H.; Porter, D.; Wei, G.; Raychaudhuri, S.; Donovan, J.; Barretina, J.; Boehm, J. S.; Dobson, J.; Urashima, M.; et al. 2010. The landscape of somatic copy-number alteration across human cancers. *Nature*, 463(7283): 899–905.

Bo, D.; Shi, C.; Wang, L.; and Liao, R. 2022. Specformer: Spectral Graph Neural Networks Meet Transformers. In *The Eleventh International Conference on Learning Representations*.

Chitra, U.; Park, T. Y.; and Raphael, B. J. 2022. NetMix2: Unifying network propagation and altered subnetworks. In *International Conference on Research in Computational Molecular Biology*, 193–208. Springer.

Cui, Y.; Wang, Z.; Wang, X.; Zhang, Y.; Zhang, Y.; Pan, T.; Zhang, Z.; Li, S.; Guo, Y.; Akutsu, T.; et al. 2023. SMG: self-supervised masked graph learning for cancer gene identification. *Briefings in Bioinformatics*, 24(6): bbad406.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29.

Ding, X.; and Jiang, T. 2010. Spectral distributions of adjacency and Laplacian matrices of random graphs. *The annals of applied probability*, 2086–2117.

Dong, X.; Zhang, X.; and Wang, S. 2023. Rayleigh Quotient Graph Neural Networks for Graph-level Anomaly Detection. In *The Twelfth International Conference on Learning Representations*.

Erdős, L.; Knowles, A.; Yau, H.-T.; and Yin, J. 2012. Spectral statistics of Erdős-Rényi graphs II: Eigenvalue spacing and the extreme eigenvalues. *Communications in Mathematical Physics*, 314(3): 587–640.

Gao, Y.; Wang, X.; He, X.; Liu, Z.; Feng, H.; and Zhang, Y. 2023. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM Web Conference 2023*, 1528–1538.

Ghosh, A.; Boyd, S.; and Saberi, A. 2008. Minimizing effective resistance of a graph. *SIAM review*, 50(1): 37–66.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

He, M.; Wei, Z.; and Wen, J.-R. 2022. Convolutional neural networks on graphs with chebyshev approximation, revisited. *Advances in neural information processing systems*, 35: 7264–7276.

He, M.; Wei, Z.; Xu, H.; et al. 2021. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, 34: 14239–14251.

Hoffman, C.; Kahle, M.; and Paquette, E. 2021. Spectral gaps of random graphs and applications. *International Mathematics Research Notices*, 2021(11): 8353–8404.

Kamburov, A.; Wierling, C.; Lehrach, H.; and Herwig, R. 2009. ConsensusPathDB—a database for integrating human functional interaction networks. *Nucleic acids research*, 37(suppl_1): D623–D628.

Khorunzhy, O.; Shcherbina, M.; and Vengerovsky, V. 2004. Eigenvalue distribution of large weighted random graphs. *Journal of Mathematical Physics*, 45(4): 1648–1672.

Kipf, T. N.; and Welling, M. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Leiserson, M. D.; Vandin, F.; Wu, H.-T.; Dobson, J. R.; Eldridge, J. V.; Thomas, J. L.; Papoutsaki, A.; Kim, Y.; Niu, B.; McLellan, M.; et al. 2015. Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nature genetics*, 47(2): 106–114.

Levi, H.; Elkon, R.; and Shamir, R. 2021. DOMINO: a network-based active module identification algorithm with reduced rate of false calls. *Molecular systems biology*, 17(1): e9593.

Li, H.; Wang, Y.; Lyu, Z.; and Shi, J. 2020. Multi-task learning for recommendation over heterogeneous information network. *IEEE Transactions on Knowledge and Data Engineering*, 34(2): 789–802.

Martínez-Jiménez, F.; Muiños, F.; Sentís, I.; Deu-Pons, J.; Reyes-Salazar, I.; Arnedo-Pac, C.; Mularoni, L.; Pich, O.; Bonet, J.; Kranas, H.; et al. 2020. A compendium of mutational cancer driver genes. *Nature Reviews Cancer*, 20(10): 555–572.

Peng, J.; Guan, J.; Hui, W.; and Shang, X. 2021. A novel subnetwork representation learning method for uncovering disease-disease relationships. *Methods*, 192: 77–84.

Peng, W.; Tang, Q.; Dai, W.; and Chen, T. 2022. Improving cancer driver gene identification using multi-task learning on graph convolutional network. *Briefings in Bioinformatics*, 23(1): bbab432.

Schulte-Sasse, R.; Budach, S.; Hnisz, D.; and Marsico, A. 2021. Integration of multiomics data with graph convolutional networks to identify new cancer genes and their associated molecular mechanisms. *Nature Machine Intelligence*, 3(6): 513–526.

Szklarczyk, D.; Gable, A. L.; Nastou, K. C.; Lyon, D.; Kirsch, R.; Pyysalo, S.; Doncheva, N. T.; Legeay, M.; Fang, T.; Bork, P.; et al. 2021. The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic acids research*, 49(D1): D605–D612.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, X.; and Zhang, M. 2022. How powerful are spectral graph neural networks. In *International Conference on Machine Learning*, 23341–23362. PMLR.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24.

Yang, L.; Chen, R.; Goodison, S.; and Sun, Y. 2021. An efficient and effective method to identify significantly perturbed subnetworks in cancer. *Nature computational science*, 1(1): 79–88.

Yang, L.; Chen, R.; Melendy, T.; Goodison, S.; and Sun, Y. 2023. Identifying Significantly Perturbed Subnetworks in Cancer Using Multiple Protein–Protein Interaction Networks. *Cancers*, 15(16): 4090.

## Reproducibility Checklist

Unless specified otherwise, please answer "yes" to each question if the relevant information is described either in the paper itself or in a technical appendix with an explicit reference from the main paper. If you wish to explain an answer further, please do so in a section titled "Reproducibility Checklist" at the end of the technical appendix.

This paper:

- Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes)
- Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes)
- Provides well marked pedagogical references for less-familiare readers to gain background necessary to replicate the paper (yes)
- Does this paper make theoretical contributions? (yes)

If yes, please complete the list below.

- All assumptions and restrictions are stated clearly and formally. (yes)
- All novel claims are stated formally (e.g., in theorem statements). (yes)
- Proofs of all novel claims are included. (yes)
- Proof sketches or intuitions are given for complex and/or novel results. (yes)
- Appropriate citations to theoretical tools used are given. (yes)
- All theoretical claims are demonstrated empirically to hold. (yes)
- All experimental code used to eliminate or disprove claims is included. (yes)
- Does this paper rely on one or more datasets? (yes)

If yes, please complete the list below.

- A motivation is given for why the experiments are conducted on the selected datasets (yes)
- All novel datasets introduced in this paper are included in a data appendix. (yes)
- All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (yes)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. (yes)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. (yes)
- All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing. (yes)
- Does this paper include computational experiments? (yes)

If yes, please complete the list below.

- Any code required for pre-processing data is included in the appendix. (yes).
- All source code required for conducting and analyzing the experiments is included in a code appendix. (yes)
- All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (yes)
- All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes)
- If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. (yes)
- This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. (yes)
- This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. (yes)
- This paper states the number of algorithm runs used to compute each reported result. (yes)
- Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. (yes)
- The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). (yes)
- This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. (yes)
- This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. (yes)