



**HAL**  
open science

# BioHAN: a Knowledge-based Heterogeneous Graph Neural Network for precision medicine on transcriptomic data

Victoria Bourgeois, Farida Zehraoui, Blaise Hanczar

► **To cite this version:**

Victoria Bourgeois, Farida Zehraoui, Blaise Hanczar. BioHAN: a Knowledge-based Heterogeneous Graph Neural Network for precision medicine on transcriptomic data. 2023. hal-04092210

**HAL Id: hal-04092210**

**<https://hal.science/hal-04092210v1>**

Preprint submitted on 9 May 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License


---

# BIOHAN: A KNOWLEDGE-BASED HETEROGENEOUS GRAPH NEURAL NETWORK FOR PRECISION MEDICINE ON TRANSCRIPTOMIC DATA

---

A PREPRINT

 **Victoria Bourgeois\***

 **Farida Zehraoui**

 **Blaise Hanczar**

IBISC, Université Paris-Saclay (Univ. Évry)  
Évry-Courcouronnes, 91020, France

May 8, 2023

## ABSTRACT

Deep-learning models promisingly benefit precision medicine in automatically solving phenotype prediction tasks on high-throughput omic data. However, their lack of interpretability limits their development in healthcare. Some studies are leveraging high-level human comprehensible biological concepts to increase the interpretability of these models, but interpretability is still not direct, and managing different knowledge types is limited. We propose BioHAN, a heterogeneous and self-explaining graph neural network, using a self-attention mechanism. The heterogeneous input graph has a central gene graph and auxiliary graphs that compensate for the sparsity of the central graph. Experiments on a real dataset show that BioHAN has similar accuracy to the non-interpretable state-of-the-art and provides automatic explanations by listing the most relevant genes and identifying the most important concept-based neighbors of these genes. All these features should make BioHAN a functional tool to clinicians.

**Keywords** Interpretability · Deep learning · Graph neural network · Heterogeneous graph · Domain knowledge · Graph classification · Precision medicine

## 1 Introduction

With the rapid improvement of sequencing techniques, the production of patient-specific low-scale characteristics (known as omic profiles) is increasing. The study of these molecular data contributes to the emergence of precision medicine, which proposes solutions at different stages of individual patient care. Among omic data, transcriptomic or gene expression data measure genes' activity through quantifying RNA fragments. These data play a key role in precision medicine as they provide information about the cellular state and allow the study of complex diseases such as cancers. In particular, one of the main applications is the prediction of diagnosis, prognosis, or treatment response from the gene expression profile of a patient. Although the classic shallow models showed interesting results for these supervised classification tasks in the past 20 years<sup>?</sup>, there is now an increasing interest in deep learning approaches. The application of deep learning methods on these data shows that healthcare could benefit from the automatic classification and extraction of reliable medical patterns Zhang et al. [2019a]. However, one of the main disadvantages of these models is their lack of interpretability, preventing their development in omics medicine. They do not provide end users (physicians, clinicians, and biologists) with understandable explanations for their predictions. End users need to understand why a phenotype has been predicted to ensure that it is based on reliable medical features rather than on irrelevant artifacts. Regardless of the model's effectiveness, this will have a major effect on their decisions and

---

\*corresponding author: [victoria.bourgeois@univ-evry.fr](mailto:victoria.bourgeois@univ-evry.fr)

confidence towards the model. Another motivation of interpretability is that the model inspection may contribute to biological discovery by revealing interesting signatures.

The first attempts to interpret neural networks are based on the *a posteriori* use of scoring methods to evaluate the impact of each input variable on the prediction. The result is a list of important genes for the model Ahn et al. [2018], Hanczar et al. [2020]. Although this may provide interesting information, they are too limited to enlighten the complexity of the model. These interpretations explain neither how the genes are used for the prediction computation nor how genes interact with each other. Moreover, an understandable explanation cannot be based only on genes. Indeed, their role is not always known, or they can be a small part of a larger relevant biological process. An understandable explanation must also include higher-level biological concepts such as biological processes or metabolic pathways. To overcome these limitations, knowledge-based approaches, that integrate high-level biological concepts, have been proposed in the literature. A first approach is based on feed-forward networks, where a knowledge database constrains the model architecture. It means that each neuron is associated with a biological concept and each network connection to a relation between biological concepts. Most models are multilayered perceptrons (MLP) so that one or more hidden layers can integrate knowledge. For example, Kang et al. [2017] connect the input genes to a hidden layer whose nodes represent proteins or compounds regulating the genes. In Deep GONet, part of the Gene Ontology hierarchy is represented by several hidden layers of a fully connected MLP Bourgeais et al. [2021]. These works present some limitations. They can represent only a certain type of graph (i.e., directed acyclic graph in case of a MLP) and cannot fully encapsulate all the knowledge semantics. They also fail to explain how genes and biological concepts interact with each other. The second approach uses graph neural network Wu et al. [2020] to integrate gene relations into predictions and interpretations. In these methods, a patient is usually represented by a homogeneous graph based on gene networks such as protein-protein interactions (PPI) or co-expression graphs Rhee et al. [2018], Ramirez et al. [2020] and the prediction is a graph classification task. Note that all the patients share the same graph structure; only the omic signal differs. These models can explain which genes are implied in the predictions and how they interact. However, they are limited to the gene level and do not provide high-level biological concepts for interpretation. Regardless of the approach used, most of the models devised require a post-hoc interpretation to identify the most relevant patterns (gene, concept). Few methods, known as *self-explaining*, have been developed to automatically provide explanations Bourgeais et al. [2022].

In this paper, we propose BioHAN (**B**iological **H**eterogeneous **A**ttention **N**etwork), a novel and self-explaining heterogeneous graph neural network for phenotype prediction. This model makes predictions at least as accurate as the state-of-the-art and provides intelligible interpretation. This interpretation includes multiple-level biological concepts (genes, biological processes, metabolic pathways) and explains how these elements interact for prediction computing. To provide these explanations, our model integrates several biological networks and ontologies (STRING Snel et al. [2000], Gene Ontology Consortium [2004], Reactome Fabregat et al. [2018]) into its architecture based on a heterogeneous graph neural network with a self-attention mechanism. In addition, BioHAN successfully deals with the sparsity of the integrated knowledge through an original architecture centered on a gene interactions graph and an adapted pooling layer. Unlike classical graph classification approaches, BioHAN considers a fixed heterogeneous graph structure. An input patient is an instantiation of this graph, and the output is a patient-specific subgraph. Experiments on cancer diagnosis demonstrate that BioHAN performs well for cancer-type prediction and automatically produces a subgraph to explain the individual outcome of a sample, which biologists and physicians can easily understand.

## 2 Related Work

Our work is closely related to both the interpretation of deep learning models and the graph neural networks.

**Interpretation of deep learning-based methods** Two main approaches emerge for interpreting deep-learning-based methods: post-hoc and self-explaining models Barredo Arrieta et al. [2020]. In the post-hoc approach, an interpretation method is applied on top of the trained deep learning model. Several post-hoc methods are available, including surrogate models and backpropagation attribution methods. Surrogate models leverage interpretable machine learning models, such as linear models in LIME Ribeiro et al. [2016], to approximate the local decision-making process of black box models. Meanwhile, backpropagation attribution methods, such as Layer-wise Relevance Propagation (LRP) Bach et al. [2015] or DeepLIFT Shrikumar et al. [2017], aim to explain a neural network’s prediction by identifying the most influential input variables and neurons for the prediction. They mainly consist in computing a relevance score by backpropagating the output signal to the input layer. As mentioned in the introduction, these methods are not accurate enough to produce a valuable interpretation in the context of gene expression-based models ?. In addition, studies show that surrogate models can produce unfaithful and inconsistent explanations ??, while explanations from backpropagation attribution methods are sensitive to random noises and adversarial attacks ?. Moreover, it is possible to obtain different explanations for the same prediction using various post-hoc methods or by applying the same post-hoc method with

different parameters Elton [2020]. In contrast, the self-explaining models are inherently interpretable models that automatically provide a form of explanation of their decisions. We can cite as examples the concept bottleneck model, which predicts an intermediate set of human-defined concepts Koh et al. [2020] and attention-based models that can provide a kind of local interpretability through the attention scores Vaswani et al. [2017]. A general opinion is that the black boxes are more accurate than the self-explaining models, with interpretation capacity being seen as a potential constraint that can reduce a model’s performance. As a result, there has traditionally been a trade-off between model performance and interpretation. However, a part of the machine learning community now argues that performance and interpretation are not mutually exclusive ?Elton [2020].

**Graph neural network** A Graph Neural Network (GNN) is a neural network designed specifically for graph-structure data Wu et al. [2020]. GNNs can handle various types of graphs, including both directed and undirected ones, and perform a wide range of tasks at the node, edge, or graph level in many domains. Graph-level GNNs comprise graph convolution and pooling layers, followed by fully-connected and output layers. Graph convolution layers update the node features, combining the signal from their neighborhood, whereas the role of pooling layers is to reduce the size of the graph to facilitate the graph classification. Common graph convolution (GCN Kipf and Welling [2017], GAT Velickovic et al. [2018], GIN Xu et al. [2019]) and pooling (DiffPool Ying et al. [2018], SAGPool Lee et al. [2019], GMN ?) generally handle homogeneous graph. Few GNNs have been designed to deal with heterogeneous graphs, and most of them cover node and edge level tasks (RGCN ?, HetGNN Zhang et al. [2019b], HAN Wang et al. [2020], HGT ?). Most GNN models miss interpretation and, as such, require either adaptations of the existing post-hoc methods that do not fit graph-based data (e.g., GNN-LRP ?, GraphLIME ?) or new methods designed specifically for them (e.g., GNNExplainer ? which generate the minimum explanation subgraph while identifying the most influential node features). However, most of these interpretation methods are designed for homogeneous graphs and do not scale to the size of omic graphs. Moreover, all these methods inherit the drawbacks of the post-hoc methods.

### 3 Biological Heterogeneous Attention Network

#### 3.1 Problem Formulation

Let  $\{(X, Y)\}_{i=1}^N$  be a training set containing  $N$  samples, where  $X = [x_1, \dots, x_d]$  is the gene expression profile of a patient with  $d$  the number of genes and  $Y = \{0, 1\}^C$  is the indicator of the class to predict with  $C$  the number of classes.  $y_c = 1$  when the sample belongs to the class  $c$ , and  $y_c = 0$  otherwise. Our model integrates the biological information from gene interactions, biological functions, metabolic pathways or other biological concepts within a large heterogeneous graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  described as follows. The interactions between genes are represented by a sparse graph  $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$ , where each node represents a gene and each edge a known relation between two genes<sup>2</sup>. Biological concepts annotating genes and characterizing their activities (biological functions, pathways...) are each represented by graphs  $\mathcal{G}_{K_i} = (\mathcal{V}_{K_i}, \mathcal{E}_{K_i})_{i=1}^I$ . For each of this knowledge graph, each node represents a biological concept and each edge a relation between two concepts. The heterogeneous graph  $\mathcal{G}$ , on which is based our model, integrates the genes graph  $\mathcal{G}^*$  as a central graph, since it receives the gene expression profile of a patient. The  $I$  auxiliary graphs  $\mathcal{G}_{K_i}$ , ( $1 \leq i \leq I$ ), are connected to  $\mathcal{G}^*$  to compensate the sparsity of the central graph through sets of edges  $\mathcal{E}_{*i}$ , ( $1 \leq i \leq I$ ), between the central graph and the  $i$ -th knowledge graph. Therefore, the set of nodes of  $\mathcal{G}$  is defined as  $\mathcal{V} = \{\mathcal{V}^*, \mathcal{V}_{K_i}(1 \leq i \leq I)\}$  and its set of edges as  $\mathcal{E} = \{\mathcal{E}^*, \mathcal{E}_{K_i}(1 \leq i \leq I), \mathcal{E}_{*i}(1 \leq i \leq I)\}$ .  $\mathcal{E}^*$  and  $\mathcal{E}_{K_i}$  are called *intra*-community edges because they connect two nodes of the same graph and  $\mathcal{E}_{*i}$  are the *inter*-community edges since they connect two different graphs. In order to identify the different types of nodes and edges, we introduce two functions  $t$  and  $r$ .  $t$  (resp.  $r$ ) assigns a type  $t(v) \in \mathcal{T}$  (resp. relation type  $r(v, u) \in \mathcal{R}$ ) to a node  $v \in \mathcal{V}$  (resp. an edge  $(u, v) \in \mathcal{E}$ ). The information of each node is embedded into a vector  $h_v \in \mathbb{R}^{D^{t(v)}}$ . In our context, since the gene expression is a scalar, the embedding dimension is one for all types of nodes:  $h_v \in \mathbb{R}, \forall v \in \mathcal{V}$ . A neighborhood of a node  $v$  is defined as the set of one-hop nodes that are directly connected to it. We can distinguish several neighborhoods of a node  $v$  according to the type of nodes with which it interacts. Depending on its connectivity to other graphs, it may have  $B$  ( $0 \leq B \leq I$ ) *inter*-community neighborhoods, denoted as  $\mathcal{N}_{inter}^j(v)$ , where  $j = 0, \dots, B$ . However, a node always has at most one *intra*-community neighborhood  $\mathcal{N}_{intra}(v)$ . Note that the node  $v$  can itself be included in its neighborhood; we talk about self-loops (SL).

#### 3.2 Model architecture

BioHAN is based on a GNN integrating the four basic modules for graph classification, namely the convolution layer, pooling and readout reduction layers, and finally, the dense layer Wu et al. [2020]. The gene expression profile of each

<sup>2</sup>For example, these relations can be extracted from a gene network based on protein-protein relations.

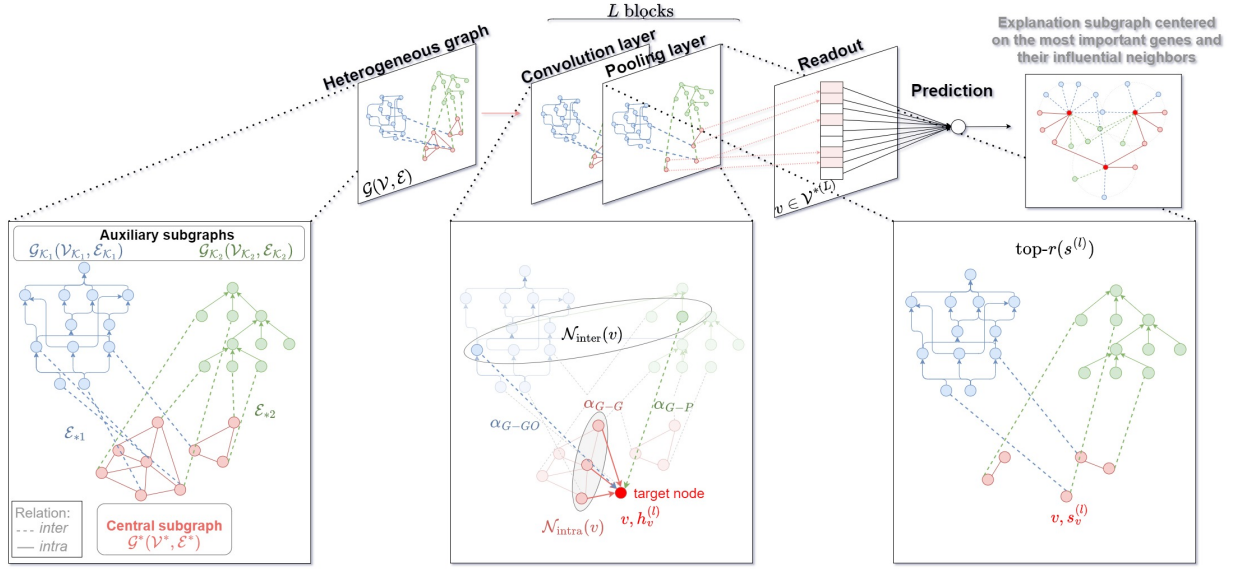


Figure 1: **Overall framework of BioHAN.** In this example, the heterogeneous graph of a sample is composed of a central gene graph  $G^*$  (colored in red) and two auxiliary graphs  $G_{\mathcal{K}_1}$  (in blue) and  $G_{\mathcal{K}_2}$  (in green). The set of genes are denoted  $G$ .  $\mathcal{K}_1$  corresponds to a set of biological functions  $GO$  and  $\mathcal{K}_2$  to a set of pathways  $P$ .  $\mathcal{R} = \{G - G, G - GO, G - P\}$  and  $\mathcal{T} = \{G, GO, P\}$ . The convolution and pooling steps are illustrated in the black boxes. An example of an explanation subgraph is given along with the prediction in the output layer.

patient is inputted into the central graph and is propagated through the knowledge graphs. Note that the structure of the graph is fixed. It is the same one for all the patients, only the data propagated through the model is different. The complete approach is illustrated in Fig.1. Each layer is described in this subsection.

**Convolution layer** The convolution layer (GCONV) is based on a self-attention mechanism to allow nodes to focus only on selected neighbors among the different types of nodes. It is based on the node level attention in HAN Wang et al. [2019]. For each pair of nodes  $(v, u)$ , an attention score  $e_{vu}$  measures the contribution of node  $v$ 's features to node  $u$ . This score depends on the type of node and the type of relationship between two nodes. Note that the score is asymmetric, it means that  $e_{vu} \neq e_{uv}$ . Node representations are initially projected in a common feature space  $D'$  by the type-specific weights  $w_{t(v)} \in \mathbb{R}^{D' \times D^{t(v)}}$ . The nodes from the same graph share the same weight. Computing scores is based on an attention mechanism,  $a : \mathbb{R}^{D'} \times \mathbb{R}^{D'} \rightarrow \mathbb{R}$ , such as  $e_{vu} = a(w_{t(v)}h_v, w_{t(u)}h_u)$ . In our experiments, this function refers to a single-layer feed-forward neural network and is parameterized by attention relation-specific weights  $w_{\alpha, r(v, u)} \in \mathbb{R}^{2D'}$ :

$$e_{vu} = \text{LeakyReLU}\left(w_{\alpha, r(v, u)}^T [w_{t(v)}h_v \parallel w_{t(u)}h_u]\right) \quad (1)$$

where  $\parallel$  represents the concatenation operation and  $^T$  the transposition.

To facilitate the comparison of the scores across all nodes  $u$  targeting the same node  $v$ , the attention coefficients are normalized using the softmax function:

$$\alpha_{vu} = \text{softmax}(e_{vu}) = \frac{\exp(e_{vu}/\tau)}{\sum_{p \in \mathcal{N}(v)} \exp(e_{vp}/\tau)} \quad (2)$$

where  $\tau$  is the temperature hyperparameter used to control the sparsity of the attention coefficients.

The update of node  $v \in \mathcal{V}$  by the convolution layer  $l$  using multi-head attention is the following propagation rule:

$$\begin{aligned} \tilde{h}_v^{(l)} &= \text{GCONV}(h_v^{(l-1)}) \\ &= \sigma \left( \frac{1}{(B+1)K} \sum_k \left( \sum_{u \in \mathcal{N}_{intra}^{(l)}(v)} \alpha_{vu}^{(l,k)} w_{t(u)}^{(l,k)} h_u^{(l-1)} \right. \right. \\ &\quad \left. \left. + \sum_j \sum_{u \in \mathcal{N}_{inter}^{(j,l)}(v)} \alpha_{vu}^{(l,k)} w_{t(u)}^{(l,k)} h_u^{(l-1)} \right) \right) \end{aligned} \quad (3)$$

where  $h_u^{(l-1)}$  is the vector representation of node  $u$  at layer  $(l-1)$ ,  $\mathcal{N}_{(\cdot)}^{(l)}(v)$  denotes the set of neighboring nodes of node  $v$  (inter or intra),  $\alpha_{vu}^{(l,k)}$  and  $w_{t(u)}^{(l,k)}$  are respectively the normalized attention coefficient for the pair of nodes  $(v, u)$  and type-specific weight at layer  $(l-1)$  and head  $k$ ,  $\sigma$  refers to an activation function (i.e., ReLU function), and  $K$  to the number of attention heads. These heads are generally used to stabilize the learning process and obtain various independent coefficients for the same node pair  $(v, u)$ . The new node representation of a node  $v$  is given by adding, without additional cost, a residual connection (RC)  $\tilde{h}_v^{(l)}$  to the vector representation of node  $v$  at the previous layer  $(l-1)$ :

$$h_v^{(l)} = \tilde{h}_v^{(l)} + h_v^{(l-1)}. \quad (4)$$

The residual connections are usually devised to make the network training easier and solve over-smoothing and gradient vanishing problems. All nodes, regardless of their type, are updated in parallel. Note that the same propagation rule applies to all directed or undirected nodes.

**Pooling layer** The pooling layer is based on top- $r$  methods Grattarola et al. [2022] for homogeneous graphs that attribute an importance score to each node and select only the highest scoring  $r$ -nodes. In BioHAN, we adapt top- $r$  strategy to consider multiple types of relations and deal with the sparsity of the central graph  $\mathcal{G}^*$  by not reducing the auxiliary graphs. We reduce the central graph  $\mathcal{G}^*$  by selecting the nodes that concentrate the most information. We look for nodes whose *intra*- and *inter*-community neighbors are also informative. These nodes will continue to interact with each other even if they belong to different connected components thanks to the auxiliary graphs  $\mathcal{G}_{\mathcal{K}_i}$ . This allows to compensate for the loss of connectivity within the central graph. A score  $s_v$  for a gene  $v$  from the central graph to layer  $l$  is calculated as:

$$\begin{aligned} s_v^{(l)} &= \gamma \times h_v^{(l)} + \frac{(1-\gamma)}{(B+1)} \times \left( \max_{u \in \mathcal{G}^{*(l)}} (h_u^{(l)} | u \in \mathcal{N}_{intra}^{(l)}(v)) \right. \\ &\quad \left. + \sum_j \max_{u \in \mathcal{G}_{\mathcal{K}_j}^{(l)}} (h_u^{(l)} | u \in \mathcal{N}_{inter}^{(j,l)}(v)) \right) \end{aligned} \quad (5)$$

where  $\gamma$  is a hyperparameter that determines how much information we consider from the node itself and its neighborhoods. The top function then returns the indices of the  $\lceil |\mathcal{V}^{*(l)}| \times r^{(l)} \rceil$  nodes with the best score  $s_v^{(l)}$ ,  $|\cdot|$  is the cardinality,  $r^{(l)} \in (0, 1]$  is the pooling ratio at the layer  $l$  indicating the number of nodes to keep.

BioHAN is composed of successive blocks of convolution and pooling layers. At the end of each block, we obtain a central graph of a reduced size such that  $\mathcal{G}^{*(l)} = \text{top-}r(s^{(l)})$ ,  $|\mathcal{V}^{*(l)}| < |\mathcal{V}^{*(l-1)}|$ . Note that the nodes in the central graph  $\mathcal{G}^{*(0)}$  initially receive the expression of the genes from the expression profile of a sample. The nodes in the auxiliary graphs are zero value as they are annotations and do not have any proper numerical information, but are initialized by a first propagation, such as:

$$h_v^{(0)} = \begin{cases} \text{GCONV}(\hat{h}_v^{(0)}) & \text{where } \hat{h}_v^{(0)} = 0 \quad \text{if } v \notin \mathcal{V}^{*(0)}, \\ x_v & \text{otherwise.} \end{cases} \quad (6)$$

**Readout layer** The readout layer only transforms the central graph of genes into a vector to perform the graph prediction task. We have chosen to keep only the genes of the central graph, but we can also keep the most important biological concepts. This layer concatenates the information from the last convolution-pooling block  $L$  from the remaining nodes in the central graph  $\mathcal{G}^{*(l)}$ , while keeping track of the original gene indices. A vector  $M$  of dimension

equal to the initial number of genes  $|\mathcal{V}^*|$  is created. The entries of this vector, corresponding to the last selected genes, receive their expression from the  $L^{th}$  block:

$$m_v = \begin{cases} h_v^{(L)} & \text{if } v \in \text{top-}r(s^{(L)}), \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

**Dense layer** A fully connected layer followed by a softmax function is finally applied to complete the classification task.

### 3.3 Model interpretation

Our model is self-explaining, which means that it automatically explains the prediction by selecting the most important genes involved in the prediction. In addition, it performs gene annotation by associating biological concepts with the selected genes. In particular, the readout layer gives the subset of genes directly implied in the final computation of the prediction. For each sample, the model returns a prediction (e.g., cancer/non-cancer) with attached probability and an explanation composed of the subset of genes and their importance score. The importance of a gene  $g$  towards a class  $c$  is measured through the relevance score<sup>3</sup>  $R_g^c \in \mathbb{R}$ . This score depends on the activation value  $m_g$  of the neurons in the readout layer (the last hidden layer of the model):

$$R_g^c = m_g \times w_{g,c} \quad (8)$$

where  $w_{g,c}$  is the weight between the gene  $g$  from the readout layer and class  $c$  from the output layer. Consequently, the genes previously dismissed by the pooling layers have a relevance score equal to zero. Using this score helps to discriminate within the subset the most relevant genes. The size of the subset depends on the pooling ratio. The smaller the subset is, the easier the interpretation will be. Note that it is not the same subset selected for each sample. The attention scores are also explored to enrich the explanations and, notably, to help to generate explanation subgraphs, as shown in Fig. 1. The attention scores from the most reliable genes can be used to identify the most influential neighbors from the different graphs (central and auxiliary) in the heterogeneous graph. Relevance and attention scores can both reveal interesting signatures across patients and models.

## 4 Experiments

### 4.1 Dataset

Most of the transcriptomic datasets contain at most 1,000 samples and not necessarily annotations in terms of labels or domain knowledge. Consequently, we apply BioHAN for cancer diagnosis on a unique, well-annotated pan-cancer dataset with around 6,000 samples. This dataset is from the RNA-Seq repository in The Cancer Genome Atlas (TCGA) platform Tomczak et al. [2015]. It contains 56,602 input genes for 5,982 cancer samples from 11 cancer types and 482 non-cancer samples from multiple tissues. We are interested in the multi-classification task where the number of classes  $C = 12$ . The dataset is standardized and divided into training (64%), validation (16%) and test sets (20%). The validation set is used for fine-tuning the hyperparameters and early stopping. A thorough description of the dataset is available in Suppl. Table S1.

### 4.2 Knowledge-based graph processing

We choose to base our heterogeneous knowledge graph  $\mathcal{G}$  on three different biological networks: gene network (G) based on protein-protein interactions (PPI) from STRING database, biological functions (GO) from Gene Ontology, and pathway networks (P) from Reactome database.  $\mathcal{R} = \{G - G, G - GO, G - P\}$  and  $\mathcal{T} = \{G, GO, P\}$ . PPI is an undirected graph composed of several connected components, and the original nodes represent proteins. They are mapped to their corresponding genes<sup>4</sup> and form the central graph  $\mathcal{G}^*$ . This graph is sparse with a connectivity of about 0.096%. There are 86 connected components, including a main one containing more than 97.99% nodes.

GO and Reactome are both directed acyclic graphs (DAG) and annotate the genes biologically. The GO annotations are functions describing the biological activity of the genes. In particular, we are interested in one of the sub-ontology of GO, GO-BP, that gathers biological processes. Reactome is a hierarchy of metabolic pathways (i.e., sequence of chemical reactions) in which the genes are implied. GO and Reactome correspond to the auxiliary graphs in  $\mathcal{G}$ . Both graphs support the propagation of the gene signal within the parsimonious central graph  $\mathcal{G}^*$  and enrich the explanations

<sup>3</sup>It is equivalent to the relevance score computed by the backpropagation attribution method *Gradients* $\times$ *Inputs* ?.

<sup>4</sup>This correspondence is admitted by the bioinformatics community.

biologically. We preliminarily extract and filter the knowledge relations to retain only the annotations with the genes contained in the dataset. Respectively, 92.63% and 61.78% of the genes are annotated by GO and Reactome graphs.  $\mathcal{G}$  is described as  $(\{\mathcal{V}^* = 10,947, \mathcal{V}_{K_1} = 16,062, \mathcal{V}_{K_2} = 2,502\}, \{\mathcal{E}^* = 115,745, \mathcal{E}_{K_1} = 37,406, \mathcal{E}_{K_2} = 2,521, \mathcal{E}_{*1} = 73,580, \mathcal{E}_{*2} = 29,305\})$  with  $K_1 = GO$  and  $K_2 = P$ . We remind that all the samples share the same heterogeneous graph structure, only the graph signal changes. Each gene in the central graph receives its corresponding signal from TCGA gene profiles. In contrast, the nodes in the auxiliary graphs do not have any value. We apply a first convolution as described by the Eq. (6) to initialize them. Regardless of the type of graph to which they belong, all nodes are subject to the same propagation rule. More details about the heterogeneous graph’s composition and its connectivity can be found in Suppl. Table S2.

### 4.3 Experimental setup

In the following, we discuss how the optimal hyperparameters are obtained. In Eq. 3, the information from the node itself is not considered explicitly. Solutions are to consider the node in its own neighborhood via self-loops Kipf and Welling [2017] or to use residual connections He et al. [2016] as described in Eq. 4. Residual connections could also help to increase the depth of the GNN and avoid gradient vanishing in the backpropagation. In our experiments, BioHAN is more accurate with residual connections than with self-loops or nothing. It costs less compared to increasing the adjacency matrix with self-loops. Concerning the temperature hyperparameter  $\tau$ , we test values from 0.1 to 1.0. Experiments show that whatever the value, the models perform the same. In the following experiments, we keep evaluating the model with two different values ( $\tau \in \{0.1, 1.0\}$ ) to analyze the differences in the model interpretation. The number of attention heads  $K$  is set to 1. Experiments show that using one head performs better than with a higher number. Besides, adding more heads has the consequence of increasing the number of parameters to learn. Fine-tuning results of the convolution hyperparameters are available in Suppl. Table S4 and Fig. S5. About the pooling convolution, the hyperparameter  $\gamma$  is fixed to 0.5 to balance the information from the node itself and its neighboring nodes. Meanwhile, isolated nodes are disadvantaged as half of their information is considered. The pooling ratio  $r$  depends highly on the number of convolution-pooling blocks used. The objective of having consecutive convolution-pooling blocks is to decrease the size of the central graph and automatically capture the smallest subset of the most important genes, which is used as support of the explanation. Experiments (available in Suppl. Fig. S6) show that a configuration of two blocks with pooling ratios  $r^{(1)} = 0.5, r^{(2)} = 0.1$  achieves the best threshold between interpretation and performances. The reduced size obtained at the pooling layer ( $L = 2$ ) is 548 on the TCGA dataset. Suppl. Table S3 lists the different hyperparameters with their optimal and tested values.

Models are trained using Adam optimizer with an initial learning rate of 0.001 and a batch size of 16. We use the early stopping strategy on the validation set with a patience of 5 epochs. We perform a multi-classification on the TCGA dataset with the softmax function in the output layer. All the experiments were executed on GPUs RTX 2080Ti and A6000 using PyTorch v1.11.0 and PyTorch Geometric v2.0.4.

### 4.4 Benchmarking

We compare BioHAN with  $\tau = 1$  against state-of-the-art machine learning models<sup>5</sup>: decision tree (DT) with the Gini criterion, random forest (RF) with the Gini criterion and a number of trees of 100, support vector machine (SVM) with the linear kernel and  $C = 1.0$ , and multi-layered perceptrons with three hidden layers (1000-500-200 neurons). Only the decision tree is an inherently interpretable model among the four models. In this benchmark, we also consider ablations of the heterogeneous graphs to assess the auxiliary knowledge contribution in BioHAN. In a first ablation, we consider only one auxiliary graph such as  $\mathcal{G} = (\mathcal{G}^*, \mathcal{G}_{K_1})$  with  $K_1 \in \{GO, P\}$ . They are reported in the results as "BioHAN-ppi-go" and "BioHAN-ppi-reactome". In a second ablation, no auxiliary graphs are considered. Currently, the input graph results in an homogeneous graph of genes such as  $\mathcal{G} = \mathcal{G}^*$ . This ablation is referred as "BioHAN-ppi-only". In that case, BioHAN works similarly as state-of-the-art GNNs composed of GAT convolution Velickovic et al. [2018] with pooling layers. In the related work, there are no comparative heterogeneous GNNs using multiple biological graphs for graph classification task (e.g., phenotype prediction), while MLP are not designed for handling this type of data. We vary the number of training samples  $N$  from 25 to 4136 (full size of the training set). For each point size, ten models are learned with a different random seed.



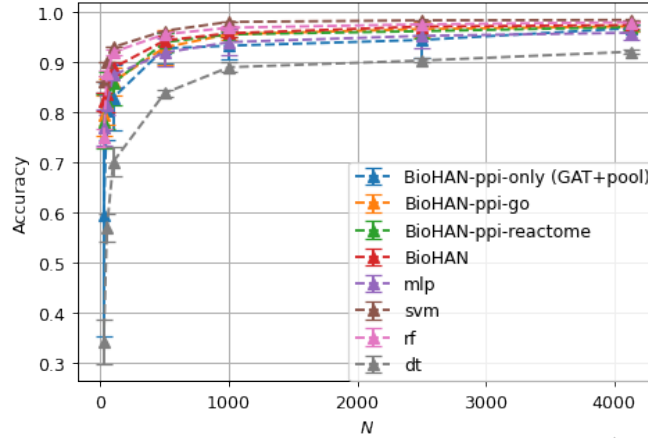


Figure 2: Classification accuracy curves according to the number of training samples  $N$ .

## 4.5 Results

### 4.5.1 Sensitivity Analysis

We compare BioHAN with  $\tau = 1$  against state-of-the-art machine learning models<sup>6</sup>: decision tree (DT) with the Gini criterion, random forest (RF) with the Gini criterion and a number of trees of 100, support vector machine (SVM) with the linear kernel and  $C = 1.0$ , and multi-layered perceptrons with three hidden layers (1000-500-200 neurons). Only the decision tree is an inherently interpretable model among the four models. In this benchmark, we also consider ablations of the heterogeneous graphs to assess the auxiliary knowledge contribution in BioHAN. In a first ablation, we consider only one auxiliary graph such as  $\mathcal{G} = (\mathcal{G}^*, \mathcal{G}_{\mathcal{K}_1})$  with  $\mathcal{K}_1 \in \{GO, P\}$ . They are reported in the results as BioHAN-ppi-go and BioHAN-ppi-reactome. In a second ablation, no auxiliary graphs are considered. Currently, the input graph results in an homogeneous graph of genes such as  $\mathcal{G} = \mathcal{G}^*$ . This ablation is referred as BioHAN-ppi-only (GAT+pool). In that case, BioHAN works similarly as state-of-the-art GNNs composed of GAT convolution Velickovic et al. [2018] with pooling layers. In the related work, there are no comparative heterogeneous GNNs for graph classification task, while MLP are not designed for handling this type of data. We vary the number of training samples  $N$  from 25 to 4136 (full size of the training set). For each point size, ten models are learned with a different random seed.

### 4.5.2 Biological analysis and interpretation

### 4.5.3 Sensitivity Analysis

Fig. 2 shows the mean and standard deviation of the accuracy evaluated on TCGA test samples. The models learn globally well in the presence of a sufficient number of training samples (from 1000). The performance of all models gradually deteriorates as the number of training samples decreases. Our approach remains as competitive as the state-of-the-art. With the entire set of training samples at  $N = 4136$ , BioHAN, SVM, and RF models have equivalent accuracies. With fewer training samples, the SVM and RF black box models perform slightly better than the deep learning models. The MLP, another black box model, generally has lower accuracy and more parameters than BioHAN (11, 551, 112 against 131, 421). BioHAN is also more competitive than the different ablations conducted, especially the BioHAN-ppi-only variation, which has a similar accuracy trend as the MLP. We can conclude that the more auxiliary knowledge we have in BioHAN, the better they make remote genes communicate and the better the performance. In addition, deep learning is expected to significantly address phenotype prediction problems in the coming years, given the increasing production of transcriptomic data. In particular, a study Hanczar et al. [2022] shows that deep learning models outperform standard machine learning models in the presence of larger training sets ( $> 5000$  samples). In all cases, the set of BioHAN variations outperforms the decision tree, the only inherently interpretable method. In Suppl. Fig. S7, F1-score and AUC metrics are displayed.

<sup>5</sup>The models are learned with the Python package Scikit-learn.

<sup>6</sup>The models are learned with the Python package Scikit-learn.

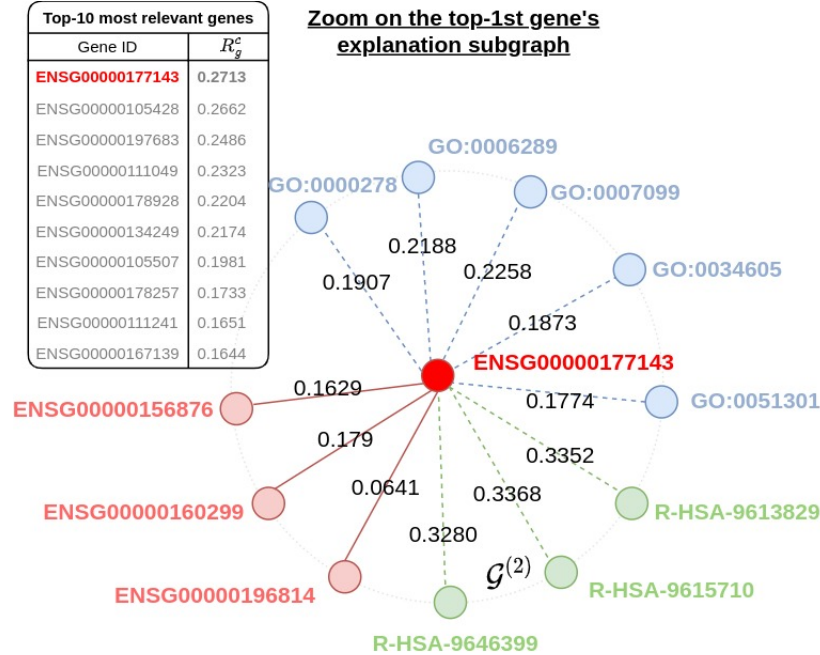


Figure 3: **Explanation subgraph on a patient predicted HNSC with a probability of 1.** The table on the left corner shows the top-10 most relevant genes with their corresponding relevance score  $R_g^c$ . A subgraph is generated around the most relevant gene ENSG00000177143. The weights on the edges represent attention scores between each pair of nodes.

#### 4.5.4 Biological analysis and interpretation

In the following, all experiments have been conducted with the maximum number of training samples. BioHAN is trained without performance loss with a temperature  $\tau = 0.1$ . We choose this value as Suppl. Fig. S8 show that the attention maps produced with  $\tau = 0.1$  reveal type-specific biological signatures contrary to the ones produced with  $\tau = 1.0$ . We first propose to analyze an explanation of a prediction returned by a BioHAN model on a patient. The explanation takes the form of a personalized subgraph centered on gene nodes. For each patient, a different explanation subgraph is generated. The final gene subset contains 548 genes  $v$  for which  $m_v \neq 0$ . For a given patient, we rank the genes according to their relevance score. The higher the score towards a class is, the more positive impact the gene has. Our analyses showed that the distribution of these scores follows a Gaussian distribution centered at zero. Assuming that these scores follow a distribution  $\mathcal{N}(0, \sigma)$  where  $\sigma$  is the empirical variance, we can apply a two-tailed t-test with a  $p$ -value = 0.05 to determine which scores are far from the upper bound and thus identify the most important associated genes. We present in Fig. 3 an example of the explanation of a patient outcome correctly predicted HNSC with a probability of 1.0. An explanation subgraph is designed around the most important gene, ENSG00000177143<sup>7</sup>. The surrounding nodes represent its *intra*- and *inter*-community 1-hop neighbors in  $\mathcal{G}^{(2)}$ . The weights on the edges represent the attention score  $\alpha^{(2,1)}$  obtained in the last convolution-pooling block ( $L = 2$ ). We can leverage the sparsity of the attention coefficients to determine which neighbors are the most influential. This explanation offers a broader view of the different interactions between the biological concepts (gene, biological function, metabolic pathway). It can be easily understood by end users familiar with this knowledge (e.g., biologists and clinicians). According to the explanation, domain-experts can then judge the predicted phenotype’s relevancy. The full list of top-relevant genes that passed the two-sided t-test and the corresponding description of all biological concepts are given in Suppl. Tables S6,S5. An explanation subgraph enriched is provided in Suppl. Fig. S9. An interpretation by cancer type can be obtained by averaging the explanations from the same cancer type predictions.

In this last analysis, we evaluate the consistency of the gene signatures to measure the sensibility of the gene subsets across models. Consistency is one evaluation criterion in XAI that expects that different models learned on the same dataset should output similar explanations Robnik-Šikonja and Bohanec [2018]. To this purpose, we train one hundred BioHAN and BioHAN-ppi-only models with  $\tau = 0.1$  and different random seeds. We evaluate the occurrences of the genes across models and test samples. We count how many times a gene  $v$  is in the subset ( $m_v \neq 0$ ). Then, we sum up each gene’s occurrences across models and samples from the same cancer type. We finally compute the maximal

<sup>7</sup>corresponding to *centrin 1*

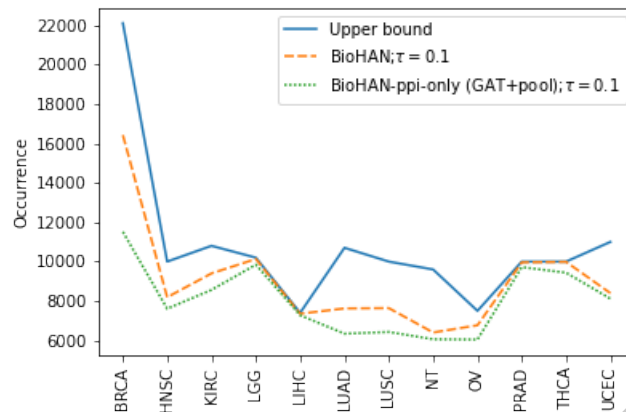


Figure 4: Maximal occurrence (y-axis) obtained across genes for each cancer type (x-axis) for two configurations of BioHAN. The upper bound represents for each cancer type the maximal frequency reachable by any genes across models and samples.

frequency among all genes for each cancer type. We compare the results of the two configurations against the upper bound (i.e., the maximal frequency reachable by any genes corresponding to the number of cancer type-specific samples times the number of models). Fig. 4 shows that using auxiliary knowledge and lower attention temperature increases the consistency of the signatures across models. We also notice that consistency occurs on the attention maps (available in Suppl. Fig. S10). By averaging the attention scores across models for one particular node target, this node captures more attention from one single neighbor specific to the cancer type.

## 5 Conclusion

We propose BioHAN, a new graph neural network adapted to heterogeneous graphs that combine different biological concepts based on STRING, GO, and Reactome databases. Note that other knowledge graphs can be integrated into BioHAN. Our model performs similarly to state-of-the-art black box machine learning methods. Our experiments show that enriching BioHAN with different knowledge graphs allows it to outperform MLP and ablations of BioHAN. This enrichment compensates for the parsimony of the central gene graph that increases along the model. We guarantee model interpretation through a self-attention mechanism, reduction layers (pooling and readout), and high-level biological concepts. A patient’s prediction explanation has been proposed as a graph centered on the most important genes while highlighting the most influential neighbors (gene, biological function, pathway). The auxiliary knowledge has another advantage: making the explanations more intelligible by characterizing gene activity and gaining domain-experts’ trust. Statistical tests for biological enrichment are, therefore, no longer necessary. In contrast, in black box methods like SVM, a two-step approach is necessary: firstly, a post-hoc interpretation is required to identify the relevant genes, and secondly, a biological enrichment analysis must be performed on the obtained list to determine overrepresented biological concepts. As discussed in the state-of-the-art, the explanations produced could be unfaithful and inconsistent. In addition, post-hoc methods specific to GNNs, like GNNExplainer, are not adapted to heterogeneous graphs. In future works, we plan to specify the heterogeneous graph depending on the specificities of individual cancer types. We finally wish to estimate the model performance and interpretation on more complicated tasks such as survival.

## References

- Zhiqiang Zhang, Yi Zhao, Xiangke Liao, Wenqiang Shi, Kenli Li, Quan Zou, and Shaoliang Peng. Deep learning in omics: a survey and guideline. *Briefings in functional genomics*, 18(1):41–57, 2019a. doi:10.1093/bfpg/ely030.
- TaeJin Ahn, Taewan Goo, Chan-hee Lee, SungMin Kim, Kyullhee Han, Sangick Park, and Taesung Park. Deep Learning-based Identification of Cancer or Normal Tissue using Gene Expression Data. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1748–1752, Madrid, Spain, 2018. IEEE. doi:10.1109/BIBM.2018.8621108.
- Blaise Hanczar, Farida Zehraoui, Tina Issa, and Mathieu Arles. Biological interpretation of deep neural network for phenotype prediction based on gene expression. *BMC Bioinformatics*, 21(1), 2020. doi:10.1186/s12859-020-03836-4.

- Tianyu Kang, Wei Ding, Luoyan Zhang, Daniel Ziemek, and Kourosh Zarringhalam. A biological network-based regularized artificial neural network model for robust phenotype prediction from gene expression data. *BMC Bioinformatics*, 18(1):565, 2017. doi:10.1186/s12859-017-1984-2.
- Victoria Bourgeais, Farida Zehraoui, Mohamed Ben Hamdoune, and Blaise Hanczar. Deep GONet: self-explainable deep neural network based on Gene Ontology for phenotype prediction from gene expression data. *BMC Bioinformatics*, 22(10):455, 2021. doi:10.1186/s12859-021-04370-7.
- Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2020. doi:10.1109/TNNLS.2020.2978386.
- SungMin Rhee, Seokjun Seo, and Sun Kim. Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 3527–3534. ijcai.org, 2018. doi:10.24963/ijcai.2018/490.
- Ricardo Ramirez, Yu-Chiao Chiu, Allen Herrera, Milad Mostavi, Joshua Ramirez, Yidong Chen, Yufei Huang, and Yu-Fang Jin. Classification of cancer types using graph convolutional neural networks. *Frontiers in Physics*, 8(203): 203, 2020. doi:10.3389/fphy.2020.00203.
- Victoria Bourgeais, Farida Zehraoui, and Blaise Hanczar. GraphGONet: a self-explaining neural network encapsulating the Gene Ontology graph for phenotype prediction on gene expression. *Bioinformatics*, 2022. doi:10.1093/bioinformatics/btac147.
- Berend Snel, Gerrit Lehmann, Peer Bork, and Martijn A Huynen. String: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. *Nucleic Acids Research*, 28(18):3442–3444, 2000. doi:10.1093/nar/28.18.3442.
- Gene Ontology Consortium. The gene ontology (go) database and informatics resource. *Nucleic Acids Research*, 32 (suppl\_1):D258–D261, 2004. doi:10.1093/nar/gkh036.
- Antonio Fabregat, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, et al. The reactome pathway knowledgebase. *Nucleic Acids Research*, 46(D1):D649–D655, 2018. doi:10.1093/nar/gkz1031.
- Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bannetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020. doi:10.1016/j.inffus.2019.12.012.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Why Should I Trust You?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1135–1144, 2016. doi:10.1145/2939672.2939778.
- Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):e0130140, 2015. doi:10.1371/journal.pone.0130140.
- Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70, pages 3145–3153. PMLR, 2017.
- Daniel C. Elton. Self-explaining AI as an alternative to interpretable AI. In Ben Goertzel, Aleksandr I. Panov, Alexey Potapov, and Roman Yampolskiy, editors, *Artificial General Intelligence*, volume 12177, pages 95–106. Springer, 2020.
- Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 2020.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

- Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.
- Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.
- Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.
- Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical Graph Representation Learning with Differentiable Pooling. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4800–4810. Curran Associates, Inc., 2018.
- Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3734–3743. PMLR, 2019.
- Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous Graph Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19*, pages 793–803, New York, NY, USA, 2019b. Association for Computing Machinery. doi:10.1145/3292500.3330961.
- Wei Wang, Xi Yang, Chengkun Wu, and Canqun Yang. CGINet: graph convolutional network-based model for identifying chemical-gene interaction in an integrated multi-relational graph. *BMC Bioinformatics*, 21(1):544, 2020. doi:10.1186/s12859-020-03899-3.
- Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous Graph Attention Network. In *The World Wide Web Conference, WWW '19*, pages 2022–2032, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3308558.3313562.
- Daniele Grattarola, Daniele Zambon, Filippo Maria Bianchi, and Cesare Alippi. Understanding pooling in graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–11, 2022. doi:10.1109/TNNLS.2022.3190922.
- Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. The cancer genome atlas (TCGA): an immeasurable source of knowledge. *Contemp Oncol*, 19(1A):68–77, 2015. doi:10.5114/wo.2014.47136.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi:10.1109/CVPR.2016.90.
- Blaise Hanczar, Victoria Bourgeais, and Farida Zehraoui. Assessment of deep learning and transfer learning for cancer prediction based on gene expression data. *BMC Bioinformatics*, 23(1):262, 2022. doi:10.1186/s12859-022-04807-7.
- Marko Robnik-Šikonja and Marko Bohanec. Perturbation-Based Explanations of Prediction Models. In Jianlong Zhou and Fang Chen, editors, *Human and Machine Learning*, pages 159–175. Springer International Publishing, Cham, 2018. doi:10.1007/978-3-319-90403-0\_9.

## A Dataset

Class	BRCA	HNSC	KIRC	LGG	LIHC	LUAD	LUSC	OV	PRAD	THCA	UCEC	NT	Total
#train	705	320	344	327	238	341	321	239	318	321	353	309	4136
#validation	176	80	86	82	59	85	81	60	80	81	88	77	1035
#test	221	100	108	102	74	107	100	75	100	100	110	96	1293
Total	1102	500	538	511	371	533	502	374	498	502	551	482	6464
Class frequency (%)	17.05	7.74	8.32	7.91	5.74	8.25	7.77	5.79	7.71	7.77	8.53	7.46	100

Table S1: **Description of the TCGA dataset.** # indicates the number of samples in each set {train,validation,test}. Meaning of the abbreviations: BRCA (Breast invasive carcinoma), HNSC (Head and Neck squamous cell carcinoma), KIRC (Kidney renal clear cell carcinoma), LGG (Brain Lower Grade Glioma), LIHC (Liver hepatocellular carcinoma), LUAD (Lung adenocarcinoma), LUSC (Lung squamous cell carcinoma), OV (Ovarian serous cystadenocarcinoma), PRAD (Prostate adenocarcinoma), THCA (Thyroid carcinoma), UCEC (Uterine Corpus Endometrial Carcinoma), NT (Normal samples). The data are pre-normalized with FPKM (fragments per kilobase per million mapped reads) and transformed using  $\log_2$ .

Database	Size	$ \mathcal{V}_{\{*,K_i\}} $	$ \mathcal{E}_{\{*,K_i\}} $	$\text{avg}(d_{\text{intra}})$	$ \mathcal{E}_{*i} $	#annotated_genes (avg $\pm$ std/node)
GO-BP	H=20	14,685	34,047	4.64 $\pm$ 17.51	73,580	10,141 (5.01 $\pm$ 15.24)
Reactome (P)	H=12	2,399	2,416	2.01 $\pm$ 5.66	29,305	6,763 (12.22 $\pm$ 19.86)
IPP (G)	CC=86	10,947	115,745	21.15 $\pm$ 978.59	102,885	10,199 (10.09 $\pm$ 13.29)

Table S2: **Details about the heterogeneous graph’s composition (GO-BP, Reactome, PPI).** Column one indicates, depending on the type of graph, either the number of related components (CC), or the graph height (H) corresponding to the maximum distance between a node and the root. Columns two to four give information about the number of nodes, *intra*-community edges and the associated average degree. Columns five and six describe the number of *inter*-community edges and the number of genes annotated by auxiliary graphs (the mean and standard deviation per node).

## B Hyperparameters tuning

Hyperparameter	Optimal values	Range of tested values
$D'$	1	-
$L$	2	{1, 2, 3}
$K$	1	[1, 6]
$ns$	0.2	-
$\tau$	0.1	$[10^{-5}, 1.0]$
RC/SL	RC&NO_SL	(NO)_RC&(NO)_SL
$\gamma$	0.5	[0, 1]
$r$	$0.5(l = 1) - 0.1(l = 2)$	$[10^{-3}, 0.5]$

Table S3: List of the hyperparameters fine-tuned during the learning step.

a)				
	NO_RC & NO_SL	<b>RC &amp; NO_SL</b>	NO_RC & SL	RC & SL
ACC (avg±std)	0.596 ± 0.389	0.964 ± 0.002	0.609 ± 0.401	0.961 ± 0.008

b)					
$K$	<b>1</b>	2	3	4	6
ACC (avg±std)	0.964 ± 0.004	0.964 ± 0.002	0.962 ± 0.004	0.960 ± 0.010	0.962 ± 0.009

Table S4: Performance comparison of the different tested values of the convolution layer’s hyperparameters: a) combination of residual connections (RC) and self-loops (SL), b) variation of the number of heads  $K$ . Ten models are learned for each configuration. The selected value that achieves the best accuracy is in bold.

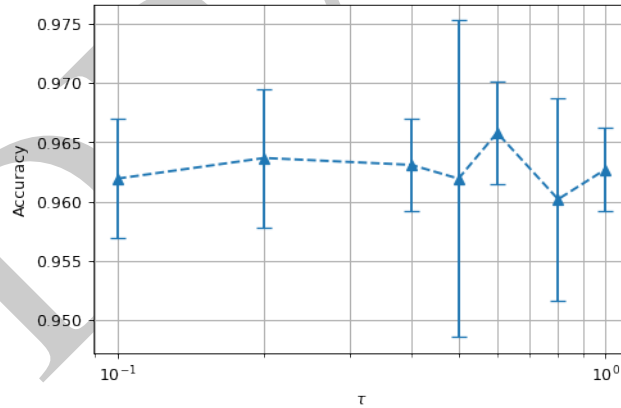


Figure S5: Evaluation of the model accuracy according to the temperature hyperparameter  $\tau$ . Ten models are learned for each configuration.

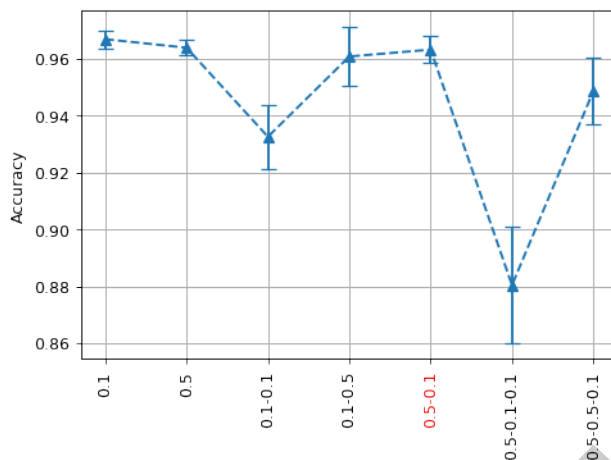


Figure S6: **Evaluation of the model accuracy according to the number of blocks  $L$  and pooling ratios  $r^{(l)}$ .** Depending on the number of blocks tested, the different configurations are named according to the following patterns:  $r^{(1)}$ ,  $r^{(1)}-r^{(2)}$  and  $r^{(1)}-r^{(2)}-r^{(3)}$ . Ten models are learned for each configuration. The optimal configuration is a good trade-off between accuracy and interpretability, in that case it corresponds to the configuration in red.

### C Sensitivity analysis

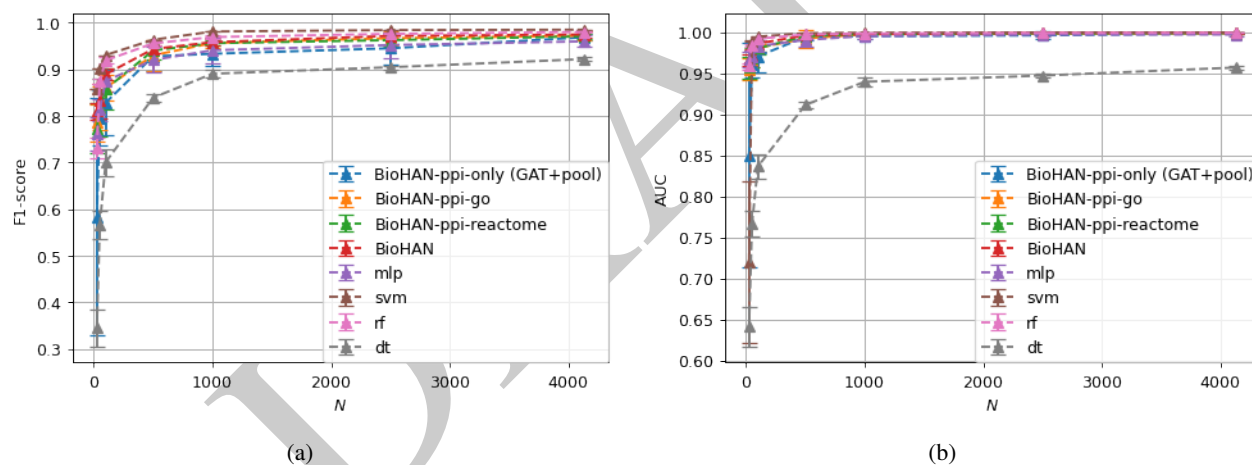


Figure S7: **Evaluation of the model performances according to the number of training samples  $N$ .** Metrics used: (a) F-1 Score (b) AUC.



## D Biological analysis and interpretation

We evaluate the attention maps obtained with one BioHAN model between one node and its neighborhood with two different values of the temperature (i.e.,  $\tau \in \{0.1, 1.0\}$ ). The pooling layer can select a different subset of genes across samples. To make the comparison across samples easier, we study the attention maps from the first convolution layer. In the central graph, the size of the neighborhood is in  $[1, 503]$  (mean = 21.15). We randomly pick a gene  $v$  with a neighborhood size close to the mean, i.e., the gene ENSG00000128655<sup>8</sup> where  $|\mathcal{N}_{intra}(v)| = 24$ . We compare the coefficients  $\alpha^{(1,1)}$  across neighbors and samples from the same cancer type. Both results on LGG (Figures S8a-b) and HNSC (Figures S8c-d) show that with a lower temperature value, the model can discriminate which neighbors are more informative to a target node. Attention coefficients learned with  $\tau = 0.1$  also reveal type-specific biological signatures. For most patients with the same cancer type, it is generally the same neighbor that is important for the target gene. It is not the same neighbor across the cancer types. For instance, it is the neighbor numbered 19 (ENSG00000198931<sup>9</sup>) for LGG, whereas it is the neighbor numbered 7 (ENSG00000174233<sup>10</sup>) for HNSC. These type-specific attentions could further be investigated to understand the biological significance of these differences.

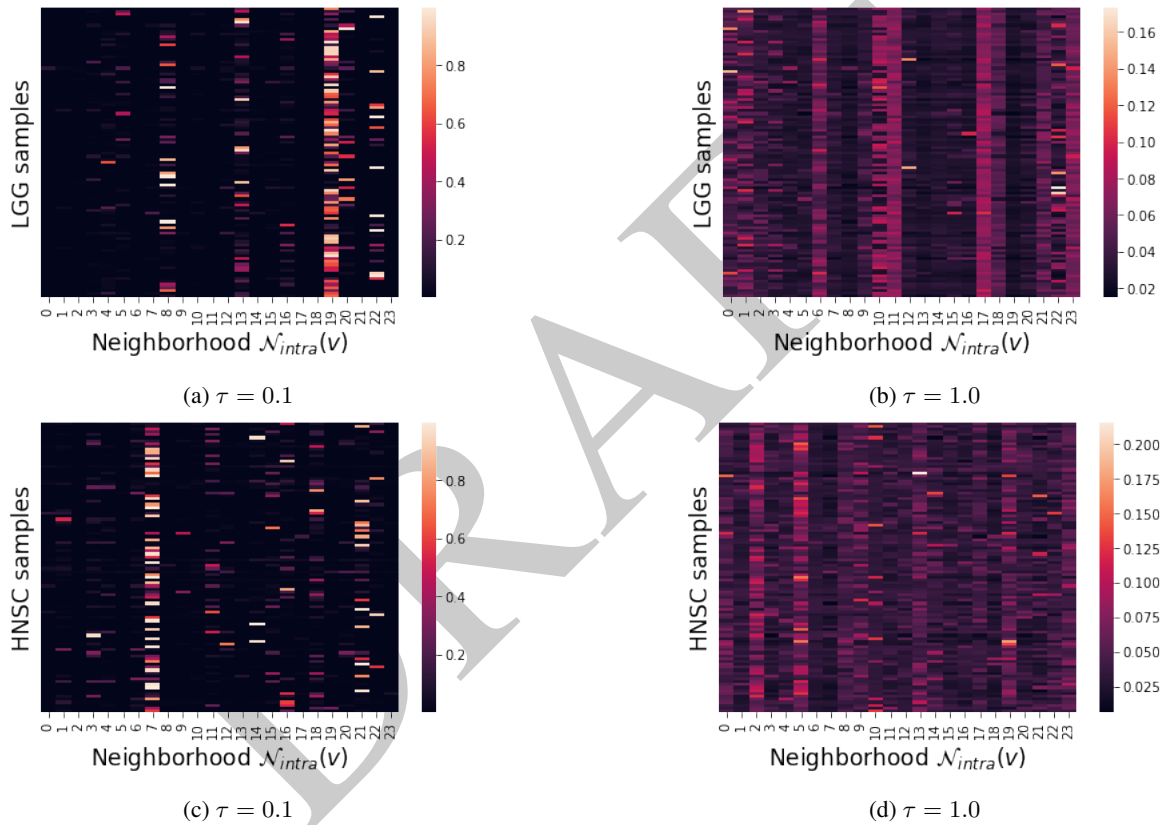


Figure S8: Attention maps on gene ENSG00000128655 for LGG and HNSC samples.

<sup>8</sup>corresponding to *phosphodiesterase 11A*

<sup>9</sup>corresponding to *adenine phosphoribosyltransferase*

<sup>10</sup>corresponding to *adenylate cyclase 6*

Gene ID	Relevance score
<b>ENSG00000177143</b>	<b>0.2713</b>
ENSG00000105428	0.2662
ENSG00000197683	0.2486
ENSG00000111049	0.2323
ENSG00000178928	0.2204
ENSG00000134249	0.2174
ENSG00000105507	0.1981
ENSG00000178257	0.1978
ENSG00000111241	0.1808
ENSG00000167139	0.1733
ENSG00000183310	0.1651
ENSG00000063601	0.1644
ENSG00000204548	0.1624
ENSG00000213218	0.1614
ENSG00000233816	0.1585
ENSG00000278057	0.1536
ENSG00000160882	0.1499

Table S5: **Top-15 relevant genes returned by the two-sided t-test (p-value=0.05).**

Concept ID	Description
ENSG00000177143	centrin 1
ENSG00000160299	pericentrin
ENSG00000196814	multivesicular body subunit 12B
ENSG00000156876	SAS-6 centriolar assembly protein
GO:0000278	mitotic cell cycle
GO:0006289	nucleotide-excision repair
GO:0007099	centriole replication
GO:0034605	cellular response to heat
GO:0051301	cell division
R-HSA-9613829	chaperone mediated autophagy
R-HSA-9615710	late endosomal microautophagy
R-HSA-9646399	aggrephagy

Table S6: **Description of the biological concepts in the explanation subgraph.**

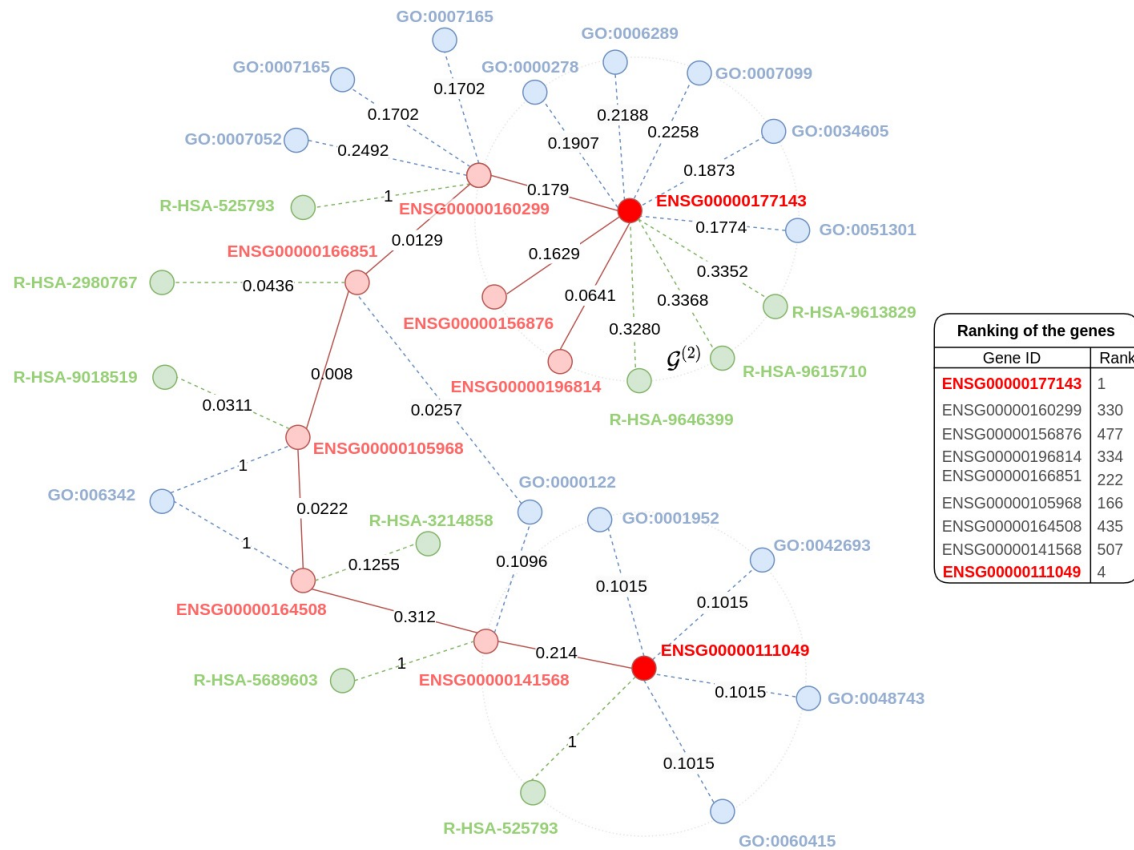


Figure S9: Example of a larger explanation subgraph based on the top-relevant genes "ENSG00000177143" and "ENSG00000111049" (respectively ranked 1 and 4). A path from gene "ENSG00000177143" to gene "ENSG00000111049" is presented. The intermediate nodes that connect these two genes are within the explanation subset returned by the readout layer. Their rank is indicated in the table on the right of the figure. GO functions and Reactome pathways with the highest attention are also displayed. The weights on the edges represent attention scores between each pair of nodes.

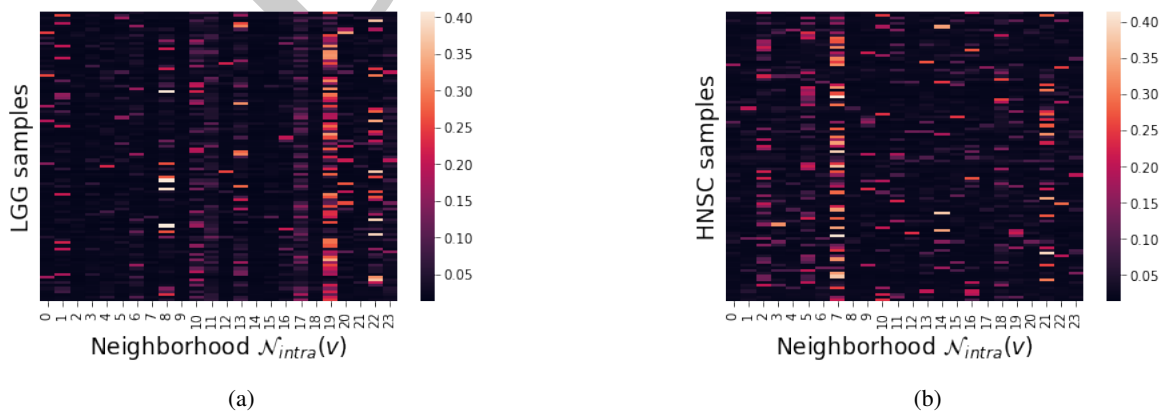


Figure S10: Attention maps averaged across models on gene 'ENSG00000128655' for LGG and HNSC samples obtained with  $\tau = 0.1$ .