Spring 5-15-2023

PathFormer: Interpretable and Powerful Graph Transformer for Gene Network Analysis

Qihang Zhao

Zehao Dong

Muhan Zhang

Philip Payne

Michael Province

See next page for additional authors

WASHINGTON UNIVERSITY IN ST. LOUIS

McKelvey School of Engineering
Department of Computer Science & Engineering

Thesis Examination Committee:
Yixin Chen, Chair
Cynthia Ma
Netanel Raviv

PathFormer: Interpretable and Powerful Graph Transformer for Gene Network Analysis

by
Qihang Zhao

A thesis presented to
the McKelvey School of Engineering
of Washington University in
partial fulfillment of the
requirements for the degree
of Master of Science

May 2023
St. Louis, Missouri

Table of Contents

List of Figures iiii
List of Tables iiv
Acknowledgments v
Abstract vii
Chapter 1: Introduction 1
Chapter 2: Background and Preliminary Analysis 5
2.1 Backgrounds 5
2.2 Gene Network Analysis 6
Chapter 3: Pathformer Model 9
3.1 Notations 9
3.2 Overveiw of Pathformer Model 9
3.3 Pathformer Encoder Layer 10
3.4 Interpretation Machines 13
3.4.1 Interpretation Machines Based on Attention Mechanism 13
3.4.2 Interpretation Machines Based on Trainable Parameter Strategy 15
3.5 Discussion 15
Chapter 4: Experiment 17
4.1 Datasets: Mayo and Rosmap 17
4.2 Baselines and Experiment Setup 17
4.3 Predictive Performance 19
4.4 Interpretation Results 19
4.4.1 Interpretation Results Based on Attention Mechanism 20
4.4.2 Interpretation Results Based on Trainable Pameter Strategy 24
Chapter 5: Conclusion 28
References 29
Appendices 35

List of Figures

Figure 1: The Overview of Pipeline 3
Figure 2: Graph Property Comparison 8
Figure 3: Architecture Overiew 9
Figure 4: Experimental Result 19
Figure 5: Population-based explanation 21
Figure 6: Personalized explanation 21
Figure 7: Summary of Detected Biomarkers and Pathways 22
Figure 8: Comparison of Instance-level Interpretatio 23
Figure 9: Detected Core Gene Networks Using Population-based Interpretation 25
Figure 10: Part of Common Genes from Two Different Datasets 26
Figure 11: Gene Ontology Term Analysis 29
Figure 12: Detected Core Gene Networks Using Pathformer-v2 36

List of Tables

Table 1: Performance of Top-K Models Using Different K Values on Two Datasets 38
Table 2: Common Genes Limited to Same Genes on Two Datasets 39

Acknowledgments

I would like to express my sincere gratitude to my professors, lab team members who have provided me with invaluable support and guidance throughout my academic journey. Their encouragement and assistance have been instrumental in helping me overcome the challenges and difficulties that I faced during my research. Without their constant help and support, I would not have been able to complete this thesis. I am particularly grateful to Prof. Yixin Chen, Prof. Fuhai Li, and Dr. Zehao Dong for their patience, dedication, and unwavering support. Their insightful feedback and constructive criticism have greatly contributed to the quality of this work. I am truly fortunate to have such amazing people in my life who have played a significant role in my academic and personal growth. Thank you all for your selfless and unwavering support.

Qihang Zhao

Washington University in St. Louis

May 2023

Dedicated to my parents and my girlfriend.

ABSTRACT OF THE THESIS

PathFormer: Interpretable and Powerful Graph Transformer

for Gene Network Analysis

by

Qihang Zhao

Master of Science in Computer Science

Washington University in St. Louis, 2023

Professor Yixin Chen, Chair

Understanding which gene/pathway expression profiles are related to specific disease phenotypes has been a critical active research area in Bioinformatics. Although graph neural networks (GNNs) have achieved impressive performance on various graph-based real-world applications such as recommendation systems and social network analysis, applying GNNs in gene-network-based Bioinformatical tasks is still challenging due to the effectiveness issue and lack of interpretation method. In this paper, we propose PathFormer, an interpretable graph Transformer (i.e. GNN), to effectively analyze gene networks and discover meaningful biomarkers/pathways. PathFormer is composed of a stack of PathFormer encoder layers and two subsequent interpretation machines. The PathFormer encoder layer is constructed upon the global attention mechanism, where a novel positional encoding scheme is proposed to enhance the model expressivity and the pathway message is incorporated in the attention matrix computation. On the other hand, the proposed interpretation machines leverage topological information and pathway message to identify core sub-gene networks of significant biomarkers and pathways through the top-K selection strategy. We apply the PathFormer model on the notorious Alzheimer disease (AD) classification task. Experiments are performed on two independent AD datasets: Mayo and Rosmap, and empirical results show that our proposed PathFormer model

significantly outperforms strong baselines, including state-of-the-art GNNs and graph Transformers. On average, Pathformer model successfully increases the prediction accuracy of 33% and 55% over best existing GNN and interpretable GNN. Furthermore, the interpretation machines in PathFormer can provide instance-level explanation (i.e. personalized explanation) as well as the group-level explanation (i.e. population-based explanation), and experiments show that PathFormer can identify meaningful core gene sub-networks that consist of multiple reported AD-related genes and rational pathways.

Chapter 1. Introduction

Gene networks are ubiquitous in various bioinformatical applications, including drug synergy prediction [1, 2], Alzheimer's disease (AD) detection [59,60], cancer subtype classification [3,4,5], etc. The increasing availability of omics data in genomic era provides huge potential in gene network analysis to reveal informative molecule structures in the bioinformatical tasks. However, the diversity of gene networks and corresponding omics data makes the testing space massive and it's impractical to manually analyze all possible situations. Thus, computational AI models are developed to analyze the functionality of genes based on gene expression profile to reveal causal processes that contribute to disease onset and progression. These computational models [6,7,8] have revolutionized the field of bioinformatics, yet they still share limitations such as the lack of transparency and ignorance of gene connectivity/topology information.

Graph neural networks (GNNs) [9,10,11,12,13,14] are dominant architectures for modeling relational structured data such as social networks or molecules. Numerous efforts are put into the direction of implementing GNNs in real-world applications [15,16,17]. Most GNNs follow the neighborhood aggregation scheme that iteratively passes messages between each node and its neighbors to learn a node embedding that encodes the local substructure. The message passing scheme shows impressive graph representation learning ability to extract highly expressive features representing the local structure information in a graph. Nowadays, GNNs have become a widely applied graph analysis method and achieved impressive performance on various graph-based tasks. Thus, GNNs show superior representation learning ability and interpretability than previous computational AI models in analyzing gene networks.

Though GNNs are theoretically ideal deep learning tools for analytical tasks on gene networks, some drawbacks limit their practical potentials in real-world Bioinformatics. First, recent works [19, 21] reveal that the dominant GNNs suffer the over-squashing problem when aggregating information from a long path or in graphs with large average node degrees. Compared to well-studied graphs like protein networks and social networks, gene networks usually have much larger average node degree and more long-range information (i.e. pathways) to encode. Thus, effectiveness of existing GNNs on gene networks is degraded due to the over-squashing problem and the prediction accuracy is usually very low. Furthermore, the interpretability of AI models [24,25,26] is usually of great importance to relieve the distrust in real-world applications, especially where high-state decisions are made based on decisions of AI models. Currently, dominant GNNs are not interpretable, and their output predictions are not transparent. Though some interpretation methods [27] are developed for GNNs, they can only provide the instance-level explanations, while population-level explanations are usually required in Bioinformatics.

To tackle the limitations of effectiveness and transparency in previous GNNs, we propose our PathFormer model. In analog to a traditional Transformer model [65], PathFormer is composed of several PathFormer encoder layers. The PathFormer encoder layer resort to a pathway-and-topology based global attention mechanism to address the over-squashing problem. Unlike the self-attention mechanism used in general Transformer encoder, the proposed attention mechanism utilizes GNNs to incorporate topology information in the key and query matrix, while computing a pathway-based attention matrix as the bias term. We also design a novel domain-knowledge-specific positional encoding scheme that takes gene index as the gene canonical label in the attention mechanism to improve the expressive power and to facilitate the pathway encoding. On the other hand, PathFormer is also equipped with two interpretation machines to provide users

with instance-level and population-level explanations. Both interpretation machines take the top-K selection strategy: (1) The first interpretation machine provides instance-level (personalized) explanation which uses the attention matrices in PathFormer encoder layers to characterize the nodes' overall impact in the prediction task, then detected core gene sub-network is composed of top-K genes and pathways connect them. (2) Inspired by SAGpool [23] and DGCNN [22], the second interpretation machine proposes a (trainable-parameter) strategy to implement the top-K selection that detects the most important genes in the prediction task. Specifically, we assign to each gene a trainable parameter and then sort the trainable parameters of all genes. Henceforth, the second interpretation machine provides group-level (population-level) explanation.

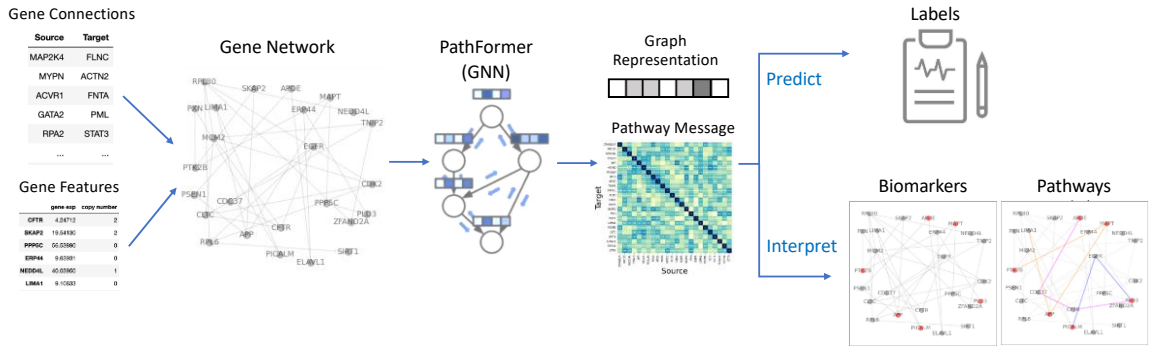


Figure 1: The overview of the pipeline. Gene meta data are transferred to graphs (i.e. gene networks). Then, the gene networks are sent to the proposed PathFormer model, which provides predictions of gene network analytical tasks and generate corresponding interpretations.

In this work, we focus on a specific gene-network based bioinformatical task: Alzheimer's disease (AD) prediction. As the most common type of dementia, Alzheimer's disease is a neurodegenerative disease that leads to cognitive deterioration of the brain, affecting the memory, thinking, and daily activities of patients. Alzheimer's disease is generally approached from diagnosis and treatment. To date, there have been only a few symptomatic treatments, including cholinesterase inhibitors [55], NMDA receptor antagonists [56], and memantine [57],

but none have been effective in stopping the progression of Alzheimer's disease. As a result, researchers have focused more on early diagnosis, as this allows patients to keep their level of function longer. From a genetic perspective, researchers are investigating the potential association of certain genes with the progression of Alzheimer's disease. For example, three genes (APP, PSEN1 and PSEN2) and one genetic risk factor (APOEε4 allele) are associated with autosomal dominant familial Alzheimer's disease [58]. Following the inspiration, two AD datasets (Mayo and Rosmap) are constructed where gene networks are formulated with gene expression of patients and known gene interactions, then the objective is to classify Alzheimer's disease versus healthy controls.

To effectively implement the early detection of AD, we introduce a GNN-based pipeline, which can also be applied to other gene network analysis tasks, such as cancer subtype classification and longevity prediction. Figure 1 illustrates the overview of the pipeline. Based on the pipeline, we evaluate our proposed PathFormer model against previous GNNs. Experimental results indicate that PathFormer significantly outperforms strong baselines, including state-of-the-art graph Transformers and GNN baselines. Furthermore, numerous visualizations show that PathFormer can extract biologically meaningful core gene sub-networks of biomarkers and pathways for future research in Alzheimer's disease.

Chapter 2. Background and Preliminary Analysis

2.1. Backgrounds

Transformer: The Transformer model solves the language modeling problem [28,29,30,31] using self-attention mechanism, and improves the performance over RNN-based or convolution-based deep learning models in both accuracy and efficiency. The Transformer encoder consists of a stack of Transformer encoder layers, where each layer is composed of two sub-networks: a (multi-head) self-attention network and a feed-forward network (FFN).

let $H = \{h_1^T, h_2^T, \dots, h_n^T\}$ be the input to a Transformer encoder layer. In the self-attention network, the attention mechanism takes H as input and implements different linear projections to get the query matrix Q, key matrix K and value matrix V, Then the attention matrix A is computed as following to measure the similarities, which is then used to update the representation in parallel.

$$A = \frac{Q\,K^T}{\sqrt{d^k}} \qquad Z = \text{softmax}(A)V \qquad (2.1)$$

After the self-attention network, the feed-forward network consists of two linear transformations with a Rectified Linear Unit (ReLU) activation in between to generate the output. i.e., $O = FFN(Z)$. The FFN is composed of a standard Dropout Layer → Layer Norm → FC (fully connected) Layer → Activation Layer → Dropout Layer → FC Layer → LayerNorm sequence, with residual connections from Z to after the first dropout, and from before the first FC layer to after the dropout immediately following the second FC layer.

Transformer on graphs: Recently, there has been a trend to generalize transformer to graph representation learning tasks. Recent works [34, 35] propose message passing layers that update node representation from nodes in surrounding neighborhood via Transformer-style attention. On the other hand, as graph data do not have the canonical grid to embed the position of nodes, different techniques are developed to embed the graph structural information and nodes spatial information in following works: SAN [32], Graphormer [35], GraphiT [33]

2.2. Gene Network Analysis

The effectiveness of GNNs for a specific graph learning task is usually affected by graph properties. For instance, Morris et al. [18] and Xu et al. [19] show that message passing GNNs cannot be more powerful than 1-dimensional Weisfeiler-Lehman (1-WL) algorithm [20] in distinguishing non-isomorphic graphs, thus these GNNs cannot predict certain graph properties such as cycle counts and will fail in corresponding graph prediction tasks. As such, comprehensive analysis of graph properties always plays an important role when applying GNNs in real world. Based on our analysis, gene networks have two domain-specific properties critical to the success of GNNs. (1) gene networks always contain numerous high-centrality nodes than general graphs. (2) Any gene at most appears once in each graph, and the existence of the edge between a certain gene pair is invariant among different graphs. Then we will introduce the impact of these properties.

The **high-centrality property** (i.e. first property) indicates that the average node degree in gene networks can be extremely larger than graphs in other real-world applications. Hence, GNNs on gene-networks suffer more severe over-squashing problem [21]. Basically, the receptive field of a

node in GNN is the size of rooted subtree to encode, and the over-squashing problem states that the receptive field of nodes will grow exponentially with the number of GNN layers, where the base of the exponential function can be approximated by the average node degree. Then, GNNs are susceptible to a bottleneck as they aggregate too much information to a single node and the exponentially growing information are squeezed into fixed-size vectors (i.e. node representations). As such, when developing GNNs for gene networks, the over-squashing problem should be taken into consideration, so that the generate representations can capture meaningful gene structure information.

Here, we also compare the graph property of gene networks in AD datasets (i.e. Mayo and Rosmap) with graphs in other benchmark datasets (i.e. (i.e. QM9 [40,41], molhiv [42], molpcba [42], and NA [43]). It has been shown that the large average node degree in graphs usually leads to the over-squashing problem in GNNs. In the study, we select 4 well-adopted graph benchmark datasets: QM9 [40,41], molhiv [42], molpcba [42], and NA [43] as baselines, and count the average number of nodes and edges in graphs. The ratio of average number of nodes and edges measures the average node degrees, reflecting the expansion speed of the receptive field. Figure 2 illustrates the empirical evaluation results. We find that the ratio in gene networks is close to 20, while that in general graphs is usually smaller than 2.

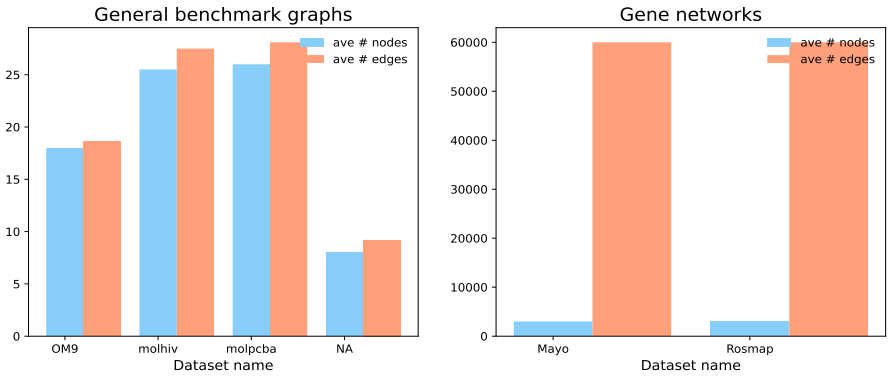


Figure 2: Graph property comparison

The **canonical labeling property** (i.e., second property) provides finer inductive bias than general graphs such that gene index (i.e., gene name) itself encodes the 'position' of a node and can be used to distinguish genes. The gene indexes itself can serve as canonical labels, which implicitly solve the challenging graph canonization problem in the situation. That is, using gene index as gene label will provides the canonical form of input gene network. Formally, we can formulate the graph classification/regression problems on gene networks as following: Given a set of graphs $\mathbb{G} = \{G_n = (V_n, E_n) | n = 1,2, \dots N)\}$, where V_n and E_n are the node set and edge set of graph G_n, respectively. The set of node types $\mathbb{T} = \{1, 2, \dots T\}$ consists of node types in $\mathbb{G}$. For a graph G_n, each node $i \in G_n$ has a d-dimensional feature x(i) and a node type p(i), while the graph G_n has a label y_n. Then, in the studied gene-network learning problem, (1) $\forall\, G_n \in \mathbb{G}$ and $\forall i, j \in V_n$, we have $p(i) \neq p(j)$; (2) For $\forall\, G_n, G_m \in \mathbb{G}$ and $i, j \in V_n$ $s, k \in V_m$, if we have $p(i) = p(s)$ & $p(j) = p(k)$, then we have $(i, j) \in E_n \iff (s, k) \in E_m$; The objective is to learn graph-representations to predict graph labels. The problem formulation aligns with the fact that the gene correlations are shared among different gene networks. That is, we can infer the connectivity of genes in unseen gene networks from the observed gene networks one their gene indexes are identical.

Chapter 3. PathFormer Model

3.1 Notations

In this section, we introduce the proposed PathFormer model. For notation convenience, we first define some basic concepts that will be used in the graph analysis and graph signal processing. We consider a graph $G = (V, E, X)$ where $V = \{1,2, \dots, n\}$ is the node set of size $N = |V|$, $E \in V \times V$ is the edge set, and $X \in \mathbb{R}^{n \times d}$ is the d-dimensional node feature tensor with a feature dimension of d. For each node i in a graph G, we denote by $\mathcal{N}(i) = \{j \in V | (i,j) \in E\}$ the neighboring node set of node i, and we denote by $\mathcal{D}(j)$ the dependent node set of node j such that there exists a path from node i to node j iff $j \in \mathcal{D}(j)$.

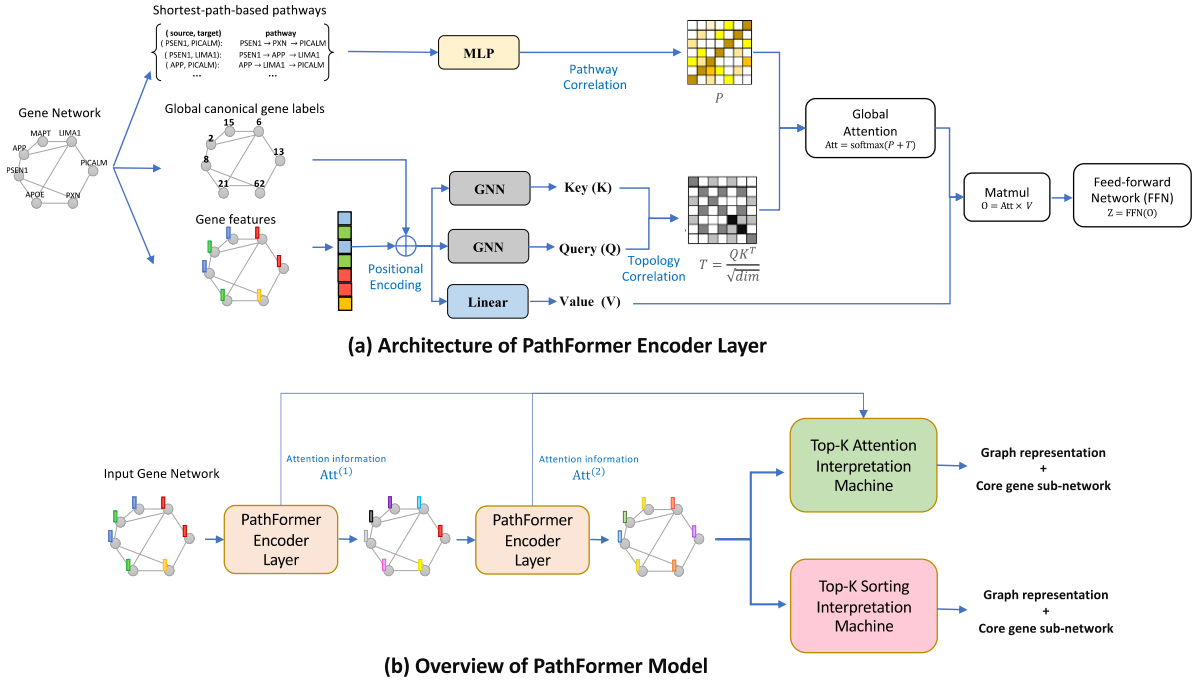


Figure 3: Architecture overview. (a) introduces the basic component, PathFormer encoder layer. (b) illustrates that the PathFormer model consists of a stack of PathFormer encoder layers and two subsequent interpretation mechanisms.

3.2 Overview of PathFormer Model

The AD early detection problem is essentially a graph classification task. Learning based graph classification is usually achieved by three steps: (1) iteratively updating node feature vectors through message passing (graph convolution) layers; (2) summarizing final node feature vectors as a graph vector/representation through a readout layer; (3) predicting the graph label based on graph representation through a standard MLP (multiple layer perceptron). Following this scheme, our PathFormer model introduces a novel graph convolution layer, PathFormer encoder layer, to tackle the effective limitation and over-squashing problem. In addition, PathFormer also proposes two interpretation machines which inject the interpretability into the readout layer to generate the graph representation/vector. Figure 3 (b) illustrates the overall architecture of proposed PathFormer model, where the PathFormer encoder layer, interpretation machines, and MLP can be trained together in an end-to-end fashion.

3.3 PathFormer Encoder Layer

The proposed PathFormer encoder layer (graph convolution layer) resorts to the attention mechanism to address the over-squashing problem. An Intuitive choice for implementing the attention-based graph convolution operation is to use the self-attention mechanism in the standard Transformer model, As (self)-attention based information propagation models (such as Transformer) have been proven to be efficient for over-squashing problem [21]. However, self-attention mechanism equivalently treats genes with same profile expression, which ignores the topology and pathway information in the attention scores and makes it difficult to design downstream interpretation method for extracting biological meaningful information. In addition, the self-attention mechanism also fails to incorporate the inductive bias that each gene only

appears at most once in each gene network and the same pair of nodes' connectivity is consistent across gene networks.

To address these drawbacks, our PathFormer encoder layer introduces three innovative design aspects to improve the standard self-attention mechanism (SAM) in gene network analysis. (1) To incorporate topology information in the attention matrix of attention mechanism, GNNs are used to replace the linear layer in SAM when computing the key matrix K, and query matrix Q. (2) To encode pathway information in the input gene network, we assume that genes interact with each other through the shortest path between them. Then PathFormer encoder layer extracts pathways between genes and uses MLP to learn the pathway correlation. (3) Gene indexes are used as the positional encodings. As each gene at most appears once in each gene network, the gene index can be used to uniformly distinguish genes in the same graph, indicating that using gene indexes as the additional features can provably find the canonical form of input gene network, which inherently solves the graph canonization problem and maximize the expressive power.

Figure 3 (a) illustrates the architecture of PathFormer encoder layer. We denote by the input to the k-th PathFormer encoder layer $H^k = (h_1^k, h_2^k, \dots, h_n^k)$, where vector h_v^k is obtained by concatenating the gene feature $z^{k-1}(v)$ from the last layer and the one-hot encoding of gene index (i.e. canonical label) i(v). In the first layer, $z^0(v)$ is the initial gene expression profile. Then, in the proposed global attention mechanism, the query matrix Q^k and key matrix K^k are computed with GNNs, while value matrix V^k is computed with a linear projection layer (i.e. LP),

$$Q^k = GNN(H^k, A) \qquad (3.1)$$

$$K^k = GNN(H^k, A) \qquad (3.2)$$

$$V^k = LP(H^k) \qquad (3.3)$$

As the strategy of using gene indexes as positional encoding provides equal expressivity as directly distinguishing the canonical form of input gene network, adding i(v) in the input h_v^k will enable GNNs to distinguish all different gene network architectures. As the gene index of the same gene and the connections of same gene pair are shared among different gene networks, the proposed positional encoding method also has good generalization ability to be applied to unseen gene networks.

Some recent research in Bioinformatics [] indicate that the personalized biomarkers may varies across different patients/samples, which makes the population-based analysis fail to detect these biomarkers. However, many personalized biomarkers from different patients are observed to formulate signaling pathways, thus pathway analysis shows huge potential in revealing the mechanism of disease phenotypes. Following this inspiration, we also propose to incorporate pathway information in the global attention mechanism. Specifically, for each pair of gene (i.e. gene i and gene j), we extract the shortest path ($SP_{i,j}$) between them. Each shortest path is represented as the sequence of gene features and the gene index. Then the pathway correlation matrix T is computed through a MLP and is used as a bias term to attention score computed from query and key:

$$T_{i,j}^k = \begin{cases} \mathrm{MLP}(SP_{i,j}) & \text{there exists a path from gene i to gene j} \\ -\infty & \text{otherwise} \end{cases} \qquad (3.4)$$

$$Att^k = \frac{Q^k {K^k}^T}{\sqrt{d_k}} + T^k \qquad (3.5)$$

Above equations also illustrates that our global attention mechanism can manipulate all components in the graph Fourier space, which helps to address the low-path concern (see Appendix B) in graph learning problem. After that, the PathFormer encoder layer updates the node features as the standard Transformer encoder:

$$O^k = \text{softmax}(\exp(Att^k))V^k \qquad (3.6)$$

$$Z^{k+1} = \text{FFN}(O^k) \qquad (3.7)$$

3.4 Interpretation Machines

In this section, we introduce our proposed interpretation machines in the PathFormer model. Specifically, the first interpretation machine is based on the proposed global attention mechanism to provide instance-level (personalized) explanation and group-level (population-based) explanation, while revealing their correlations. On the other hand, the second interpretation machine introduces trainable parameters to characterize the significance of each gene and can generate a population-based explanation. The PathFormer model equipped with the first/second interpretation machine is named as PathFormer-V1/PathFormer-V2, accordingly.

3.4.1 Interpretation machine based on attention mechanism

The first interpretation machine resorts to the framework of additive feature attribution methods, which provide explanation through a linear function of binary variables. Numerous interpretable AI models in image processing are unified in this framework and provide meaningful explanations on images. To specifically interpret graph neural networks on graph-structured data, we adapt the definition of the additive property for graph data attribution as follows:

$$\epsilon(A) = \phi_0 + \sum \phi_{i,j} A_{i,j} \qquad (3.8)$$

where A is the adjacency matrix of the explanation, $A_{i,j}$ is a binary variable representing the existence of an edge.

Personalized explanation Unlike a standard message passing GNN model, the attention matrices $Att^1 \ldots Att^K$ from our PathFormer model (where K is the number of PathFormer encoder layers) encode the pathway message information, gene profile expressions and topology information. Thus, we can interpret the K-layer PathFomer of each sample/patient as following:

$$\epsilon_{i,j} = \sum_{k=1}^{K} Att_{i,j}^{k} \qquad (3.9)$$

Where Att^k is the learnt attention matrix in the k-th PathFormer encoder layer. Here we define the biomarkers as the genes most relevant to the prediction of the PathFormer model, which can be quantitatively measured as $e_i = \sum_j \epsilon_{i,j}$. Then, to detect the core gene sub-network, we can sort $\{e_i, i \in V\}$ to detect the most relevant genes. Furthermore, as each attention matrix Att^k is adjusted by the pathway correlation matrix, the core gene sub-network also contains shortest pathways between selected genes to allow the pathway information aggregation.

Population-based explanation Above gene 'importance' e_i is personalized. We can also compute the average e_i across different patients/samples to reflect the population-based significance. Then the core gene sub-network for the whole population is constructed following the same sorting strategy and shortest path connection framework.

3.4.2 Interpretation machine based on trainable-parameter strategy

Though attention mechanism based interpretation can provide the population-based interpretation, it is not population oriented and is originally designed for personalized explanation. Hence, here we also introduce a population-oriented interpretation mechanism

based on a trainable parameter strategy. Based on our previous analysis, gene indexes in the gene networks provides meaningful additional information for structure representation learning. Hence, we propose to use a gene-index-related trainable parameter to explicitly perform the sorting and top-K selection operation, instead of using $\{e_i, i \in V\}$ computed based on learnt attention matrices. Concretely, we denote by ϵ the trainable parameter of size T, where T is the total number of genes in the graph dataset $\mathbb{G}$. Then, for each input graph $G = (V, E)$, let H^k be the output node representations from the last PathFormer encoder layer. Let index set $idx = \{i(v)|v \in V\}$, where i(v) is the gene index. Then, the trainable-parameter strategy first extracts the corresponding trainable weights by index selection:

$$\epsilon_G = \epsilon[idx] \qquad (3.10)$$

Then nodes in graph G is sorted according to ϵ_G, and we only use top-K node representations in H^k for prediction. Let idx_G be the returned index set in the sorting operation on ϵ_G, then the output node representations of the trainable parameter strategy can be represented as $H^k\epsilon_G[idx_G]$. We also provide examples in Appendix A to explain this process more.

3.5 Discussions

PathFormer has multiple exclusive advantages. First, the gene-index-based attention matrix in Pathformer encoder layers enables the learnt attention matrix to embed path-based information between nodes from the additional gene index information. Henceforth, it inherently avoids the complexity of finding all possible paths between nodes. In addition, as the PathFormer encoder layer is formulated upon the attention mechanism, it contains rich higher-order connectivity patterns and avoids the exponentially increased receptive field in message passing GNNs, which address the low-path challenges [36,37,38, 39] and over-squashing challenges in gene networks. We demonstrate the effectiveness of the PathFormer model on the designed gene-network

datasets. Experimental results indicate that PathFormer significantly outperforms strong baselines, including state-of-the-art graph Transformers and GNN baselines. Furthermore, visualizations show that PathFormer can identify more meaningful substructures in gene networks than general GNN explanation models like GNNExplainer [27] in impactful real-world bioinformatical tasks.

Chapter 4. Experiment

4.1 Datasets: Mayo and Rosmap

Two datasets of Alzheimer's disease, Mayo and Rosmap, were used for our model. These two datasets are designed for the challenging Alzheimer's disease (AD) classification problem in bioinformatics [50,51], such as distinguishing Alzheimer's disease (AD) samples from normal elderly controls [52]. The node features of the graphs in Mayo and RosMap were first mapped to the reference genome using STAR (v.2.7.1a), and then the transcriptomic (TPM) values of 16,132 common protein-coding genes were obtained in both datasets by applying the Salmon quantification tool in alignment-based RNA-seq data. The aim is to distinguish the AD samples from the control Samples. The Mayo dataset contains 158 graphs, each including 16,132 genes, while the Rosmap dataset contains 357 graphs, each also including the same 16,132 genes. The only difference is that the feature values are different. Also, according to the Biological General Repository for Interaction Datasets (BioGRID: https://thebiogrid.org/), any two interrelated genes are undirected.

4.2 Baselines and Experiment Setup

Baselines In the experiment, we select two types of well adopted baselines in graph representation learning tasks: (1) The first type of baselines are state-of-art graph Transformers on graph property predictions, including position-aware models (Graphormer), structure-aware models (graphTrans [44]) and SAT [45]. (2) The second type of our baselines are powerful GNNs that have achieved leader positions in various graph learning problems: such as flat GNNs (i.e., GAT, DGCNN, GIN, and GCN), Subgraph-based GNNs (i.e., ID- GNN [46] and NGNN [47]) and hierarchical graph pooling GNNs (i.e., Diffpool and SAGpool).

Experiment setup. To provide robust evaluation, we perform 5-fold cross validation due to the size of the dataset and the number of baselines for comparison, and report the accuracy averaged over 5 folds and the standard deviation of validation accuracies across the 5 folds. All baselines are implemented in PyTorch on NVIDIA Tesla P100 12GB GPUs.

Next, we provide the (hyper-)parameter setting of the proposed Pathformer models and baselines. The training protocols is composed of the selection of the evaluation rates and training stop rules. Specifically, the learning rate of optimizer picks the best from the set {1e 4, 1e 3, 1e 2}; the training process is stopped when the validation metric does not improve further under a patience of 10 epochs.

Similar to the standard Transformer model, the PathFormer encoder is composed of a stack of PathFormer encoder layers. In the experiments, we use 2 PathFormer encoder layers. In each layer, the number of head is set to be 2; dimensions of key, query, and value vectors are set to be 32; the dimension of linear layers in the feed-forward network is set as 64. In each graph convolution layer of GNN baselines, the embedding dimension is set to be 128. The number of graph convolution layer is selected from the set {2, 3, 4}. In graph pooling models (SAGpool and Diffpool), the proportion of nodes to keep in each graph pooling layer is set to be 10%. In NGNN, we use height-2 rooted subgraphs due to the memory consideration. The graph-level readout function is selected from the set {*mean*, *sum*}. In GNNExplainer, we use GCN as the base model. Graph Transformers in the experiments are based on standard Transformers. When a standard Transformer model is used, the number of Transformer encoder layer is 4, the dimension d_k is set to be 6, the number of heads is set to be 4. Due to the property of designed dataset, Graphormer does not perform the pre-training as on OGB datasets. In graphTrans, we

use GIN with 2 layers to extract the node embeddings. In SAT, we use height-2 rooted subgraphs.

4.3 Predictive Performance

In the section, we evaluate the predictive performance of PathFormer on graph-level learning tasks. Figure 4 summarizes the predictive performance of PathFormer relative to strong baselines. Experimental results indicate that PathFormer consistently improve the performance over all baselines.

	Mayo		RosMap	
Methods	Accuracy ↑	F1 score ↑	Accuracy ↑	F1 score ↑
GIN	0.547 ± 0.042	0.531 ± 0.036	0.519 ± 0.039	0.522 ± 0.041
GCN	0.514 ± 0.049	0.482 ± 0.021	0.501 ± 0.036	0.459 ± 0.032
GAT	0.502 ± 0.036	0.471 ± 0.028	0.509 ± 0.031	0.477 ± 0.030
DAGNN	0.521 ± 0.034	0.501 ± 0.021	0.522 ± 0.037	0.508 ± 0.042
NGNN	0.564 ± 0.018	0.533 ± 0.021	0.551 ± 0.024	0.516 ± 0.020
SAGpool	0.521 ± 0.033	0.517 ± 0.031	0.509 ± 0.030	0.471 ± 0.042
Diffpool	0.539 ± 0.031	0.542 ± 0.021	0.517 ± 0.038	0.492 ± 0.026
PNA	0.554 ± 0.037	0.539 ± 0.046	0.560 ± 0.035	0.528 ± 0.041
Graphformer	0.594 ± 0.041	0.601 ± 0.038	0.602 ± 0.050	0.613 ± 0.046
graphTrans	0.513 ± 0.027	0.500 ± 0.031	0.503 ± 0.041	0.487 ± 0.037
SAT	0.622 ± 0.021	0.5641 ± 0.037	0.619 ± 0.033	0.627 ± 0.031
PathFormer-V1	**0.835± 0.036**	**0.825 ± 0.022**	**0.821± 0.025**	**0.781± 0.019**
PathFormer-V2	**0.865± 0.035**	**0.862± 0.034**	**0.783± 0.011**	**0.747± 0.028**

Figure 4: Experiential results

Specifically, our PathFormer significantly outperforms all GNNs by **33%** on average in terms of classification accuracy. Furthermore, when we compare PathFormer against other interpretable GNNs (i.e., SAGpool, Diffpool) that provide interpretability by detecting core subgraphs, the performance improvement increases to about **55%.** Furthermore, we also find that previous GNNs' classification accuracy of distinguishing AD samples from health controls is only slightly higher than 0.5. This observation indicates that the applicability of previous GNNs are limited in real-world bioinformatics as these AI models are slightly better than random guess. From this perspective, our PathFormer successfully increases the classification accuracy to a reasonable

level (around 0.8) for applications. Hence, PathFormer show significant practical advantage over previous GNNs in the early detection of AD.

4.4 Interpretation results

The interpretability of deep learning models is of vital importance for real-world applications in the field of bioinformatics, as the interpretability can illustrate the chain of reasoning to aid in trust and to increase the testability. In the section, we present the interpretation results from two proposed interpretation mechanism in our PathFormer model.

4.4.1 Interpretation results based on attention mechanism

Population-based and personalized explanation results. In the first part, we provide and analyze the population-based/personalized explanation results. Figure 5 presents visualizations of population-based core gene sub-networks on dataset Mayo and Rosmap, while Figure 6 provides visualizations of personalized core gene sub-networks for 4 random-selected patients from the testing samples in Mayo. In these visualizations, we highlight the provable risky Alzheimer's genes as red color, genes on the shortest pathways between provable risky genes as yellow color, genes connected to provable risky genes as green color, and rest genes in the core sub-network as blue color. While shortest pathways are highlighted with purple lines.

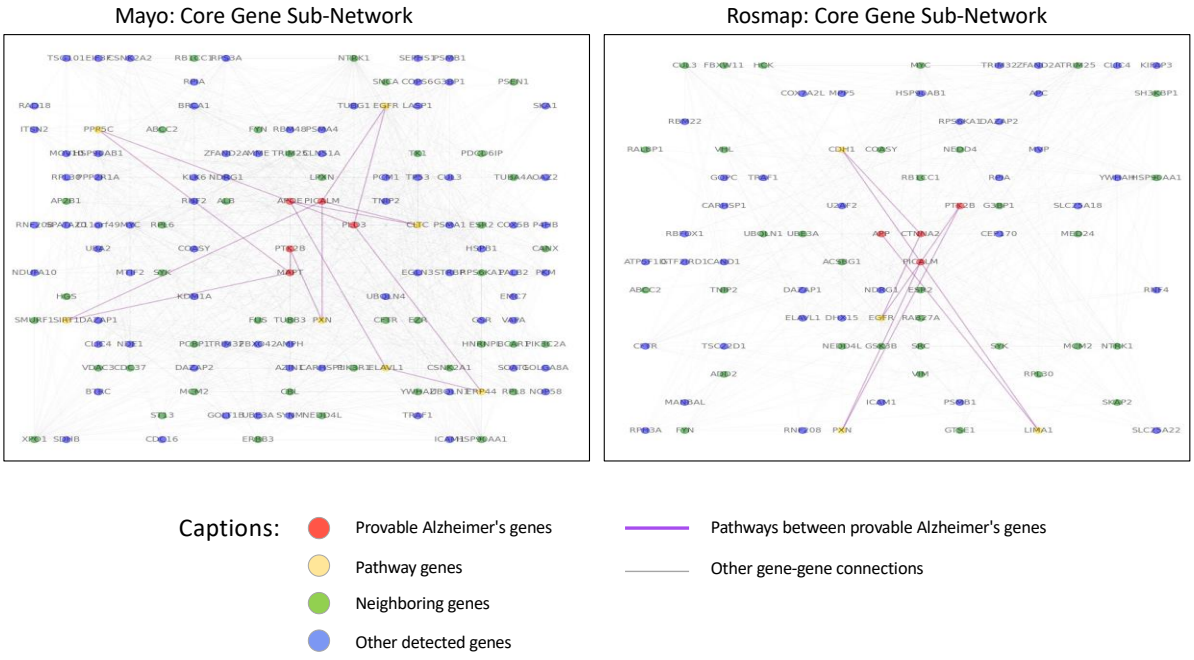


Figure 5: Population-based explanation: Visualization of population-based core gene sub-networks

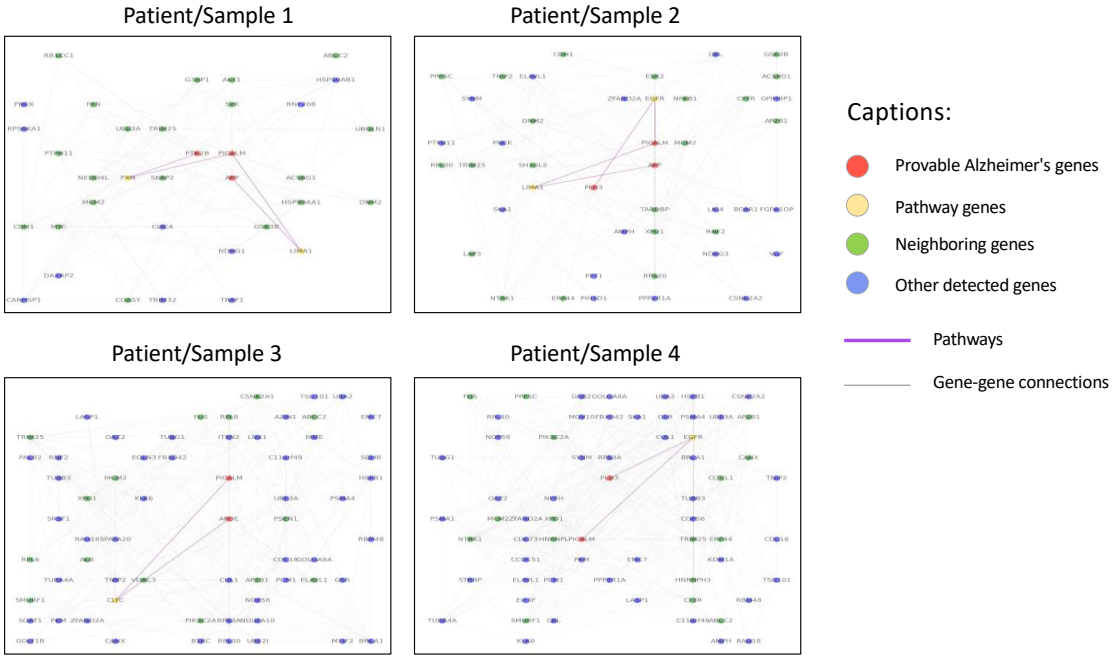


Figure 6: Personalized explanation: visualization of detected core gene sub-network across samples/patients

We find out several intriguing observations from these visualizations. First, Figure 6 illustrates that the detected personalized gene sub-networks across patients are different. This observation

aligns with the bioinformatical research findings that biomarkers vary across patients. Second, by comparing detected biomarkers and pathways of population-based gene sub-networks in Figure 5 and personalized gene sub-networks in Figure 6, we find that personalized biomarkers usually target on specific population-based pathways. Hence, we summarize the detected target biomarkers and pathways in Figure 7. Specifically, target biomarkers consist of some provable risky Alzheimer's genes as well as their corresponding pathway genes and common 1-hop genes (from Mayo and Rosmap). While the connections between biomarker genes contains all detected pathways of these genes.

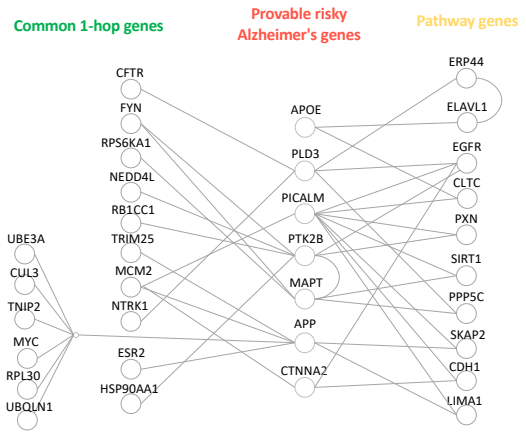


Figure 7: Summary of detected biomarkers and pathways.

By comparing Figure 7 with recent genomic research findings in AD, we find that the detected biomarkers contain many reported common Alzheimer's genes, including late-onset Alzheimer's gene: APOE and young-onset Alzheimer's gene: APP. Furthermore, research on the genetics of Alzheimer's progresses reveal some biological important links between late-onset Alzheimer's gene (APOE) and other genes such as PICALM, PLD3, CLU, etc. Some of these link can be tracked in Figure 7, and Figure 7 provides how these genes are linked to each other through

pathways. For example, we think that APOE and PLD3 are linked through the pathway APOE↔ ELAVL1↔ ERP44↔PLD3.

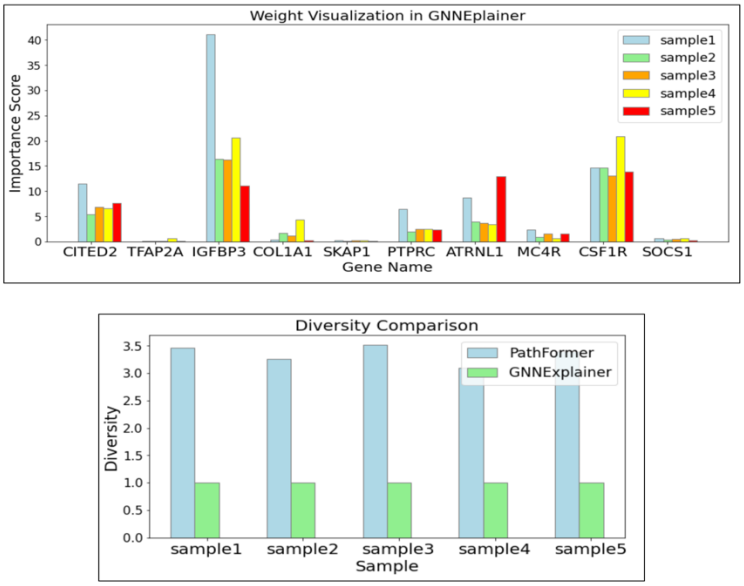


Figure 8: Comparison of instance-level interpretation between PathFormer and GNNExplainer

Comparison to previous GNN explanation models. In the experiment, we also PathFormer against the current state-of-art GNN explanation model: GNNexplainer [27]. GNNExplainer provides a general solution to introduce the interpretability in graph neural networks without modification of the underlying model architecture, and it can identify subgraphs consists of important graph pathways in the prediction process. In the experiment, we use GCN as base GNN model. We find that personalized core sub-networks detected by GNNexplainer is always same across different patients, and the visualization result can be found in Appendix C. The result indicates that base GCN model in GNNexplainer consistently focus on the same localized pattern (a gene sub-network shared among patients), while assigning genes in the localized pattern different importance scores (weights) as the top in Figure 8. We also use the distance of

shortest path of all gene pairs in the gene sub-networks deteted by GNNexplainer and PathFomer to measure the diversity of the sub-networks. Thus, high diversity indicates that long-range interactions between genes are tractable and explainable. The bottom in Figure 8 shows the comparison, and we can see that PathFormer can capture and interpret longer interactions between genes than usually GNN models.

4.4.2 Interpretation results based on trainable parameter strategy

Next, we present the population-based explanation results of our second interpretation machine. Figure 9 illustrate the detected core gene sub-networks, where K is set to be 1000, to reveal the mechanism of Alzheimer's progress. The green path represents the shortest path connecting each known Alzheimer's gene and risk gene variant, and the dark blue edge links 1-hop neighbors of the known Alzheimer's genes and risk gene variants above. The red nodes indicate the known Alzheimer's genes and risk variants, and the orange nodes indicate their 1-hop neighbors and the points passed on the shortest path. The remaining light blue nodes indicate genes that may have little association with Alzheimer's disease. Furthermore, we remove the singleton nodes, i.e., the known Alzheimer's genes and risk gene variants whose neighboring genes are not part of the list of top 1000 genes. Compared to known and probable Alzheimer's genes [49,61,62,63], Pathformer-v2 identified known genes APP and PSEN, as well as the risk genetic variants PICALM and CTNNA2 from Mayo dataset, while it recognized known gene APOE and risk genetic variants PTK2B, MAPT, PLD3, CTNNA2 and CNTNAP2 from Rosmap dataset. In addition, we also provide strategy to sparser the detected core gene sub-networks in Appendix A by incorporating the attention mechanism as we did in last section.

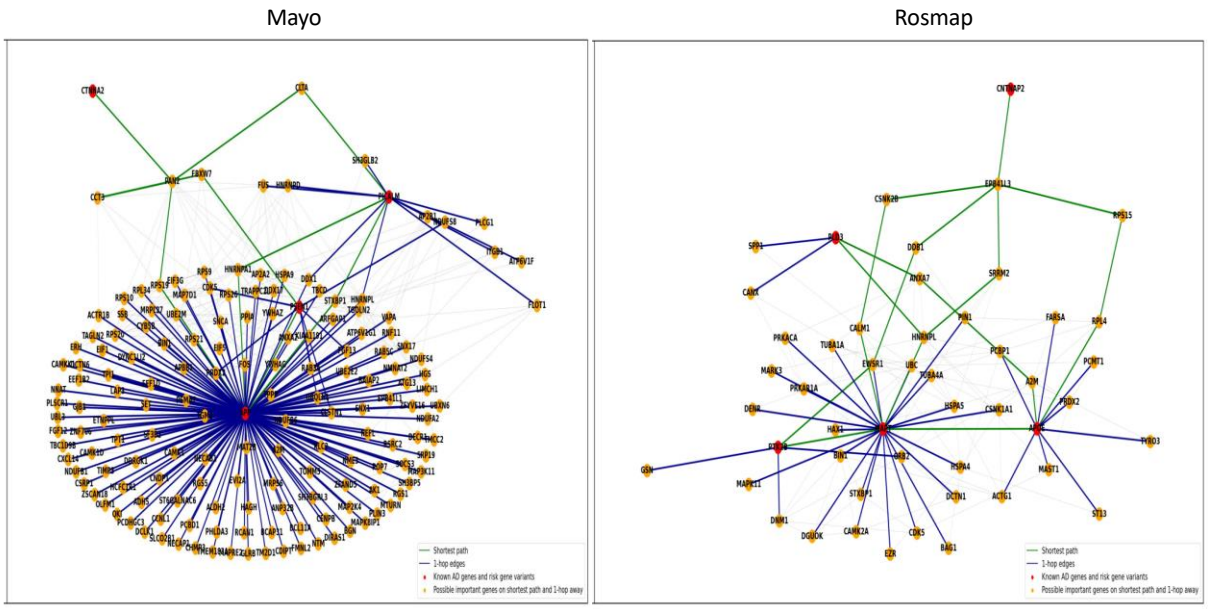
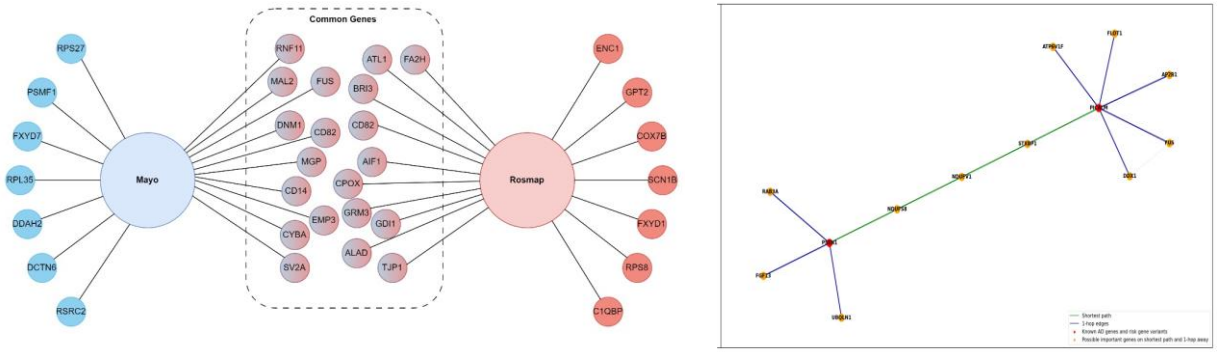


Figure 9: Detected core gene networks using population-based interpretation machine.

Since the interpretation machine is population oriented, then we can extract the common subgraph from dataset Mayo and Rosmap to reveal how Alzheimer's genes and other risky genes are interact with each other, which provides insights into detecting more potential genes for Alzheimer's prevention and treatment. Figure 10 (a) presents some reported/potential common genes detected in Mayo and Rosmap, which illustrates our population-based interpretation machine successfully recognizes known genes APOE and PSEN1 [49,61,62,63] and the risk genetic variants PICALM, CTNNA2, VPS35, MAPT, and MEF2C. Furthermore, Figure 10 (b) provides an example explanation of how these functionally interact with each other.



(a) (b)

Figure 10: Part of common genes from two different datasets

We also perform Gene Ontology (GO) term analysis on the common genes subset to determine whether this subset of genes is significantly enriched in particular functional categories or biological pathways. This information serves to narrow down and prioritize the genes that require further experimental validation, potentially uncovering the underlying mechanisms of Alzheimer's disease, and identifying promising biomarkers and novel target genes that could be useful for future diagnostics. In figure 11, we present three subontology analyses including BP for Biological Process on top left, MF for Molecular Function on top right, and CC for Cellular Component on bottom. In the output graphs, the y-axis represents the enriched GO terms obtained from the GO enrichment analysis, while the x-axis represents the percentage of genes in the study set that are associated with each GO term. Each bar in the graph corresponds to a GO term, and the color of the bar represents the corrected p-value for the enrichment of the term, with blue indicating higher p-values (less significant enrichments) and red indicating lower p-values (more significant enrichments).

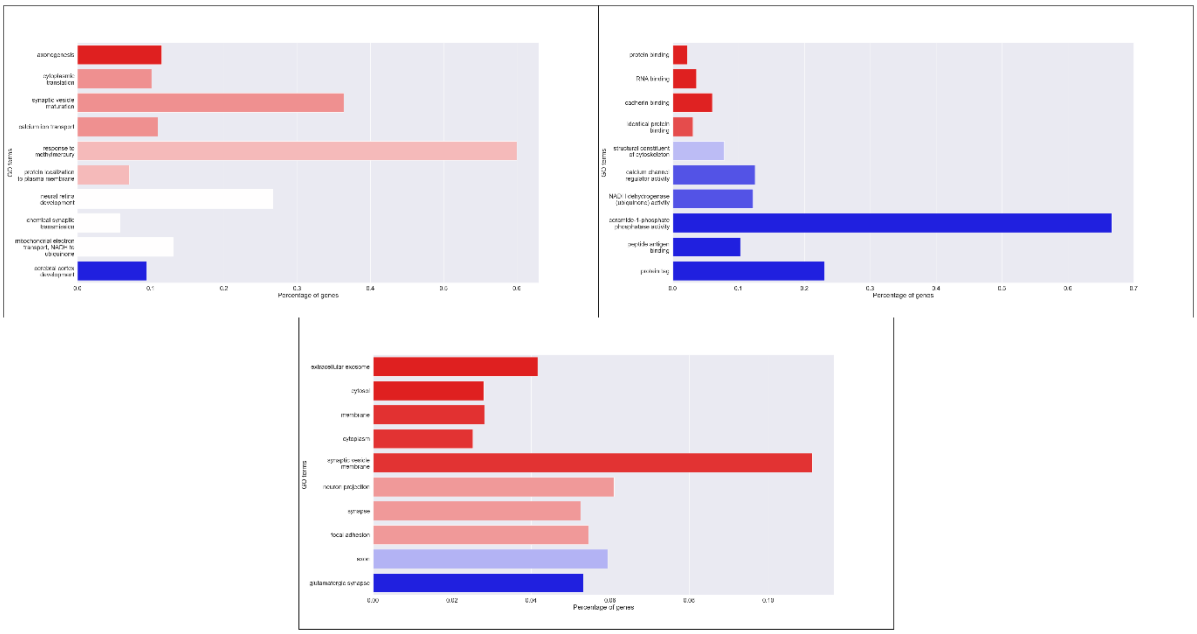


Figure 11: Gene Ontology term analysis on three subontologies, (BP = Biological Process(top left); CC = Cellular Component(top right); MF = Molecular Function(bottom))

Chapter 5. Conclusion

In this paper, we introduce a novel graph Transformer model, PathFormer, to support effective and interpretable analysis on gene networks. We test our PathFormer model on the Alzheimer's disease (AD) prediction task. PathFormer significantly improves of the prediction accuracy over GNN baselines, where the improvement is about 33% over existing best GNN and 55% over interpretable GNNs. Furthermore, PathFormer can provide different level explanations by detecting the personalized/population-based core gene sub-networks, which contain biomarkers of reported risky Alzheimer's genes and probable pathways that reveal the mechanism of Alzheimer's progresses. Overall, the PathFormer model is the current state-of-art deep learning (GNN) method for early detection of AD. Furthermore, the proposed interpretation machines can systematically identify genes that may affect the risk of Alzheimer's disease, which provides huge potentials in developing new therapies to treat and prevent Alzheimer's disease in the future.

References

1. Andrew L Hopkins. 2008. Network pharmacology: the next paradigm in drug discovery. Nature chemical biology 4, 11 (2008), 682–690.

2. Scott H Podolsky and Jeremy A Greene. 2011. Combination drugs—hype, harm, and hope. New England Journal of Medicine 365, 6 (2011), 488–491

3. Lu, Ying, and Jiawei Han. "Cancer classification using gene expression data." *Information Systems* 28.4 (2003): 243-268.

4. Viale, Giuseppe. "The current state of breast cancer classification." *Annals of oncology* 23 (2012): x207-x210.

5. Amrane, Meriem, et al. "Breast cancer classification using machine learning." *2018 electric electronics, computer science, biomedical engineerings' meeting (EBBT)*. IEEE, 2018

6. Yang, Y., Han, L., Yuan, Y. *et al.* Gene co-expression network analysis reveals common system-level properties of prognostic genes across cancer types. *Nat Commun* **5**, 3231 (2014).

7. Horvath, Steve, and Jun Dong. "Geometric interpretation of gene coexpression network analysis." *PLoS computational biology* 4.8 (2008): e1000117

8. Song, Won-Min, and Bin Zhang. "Multiscale embedded gene co-expression network analysis." *PLoS computational biology* 11.11 (2015): e1004574

9. Justin Gilmer et al. “Neural message passing for quantum chemistry”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1263–1272.

10. William L. Hamilton, Rex Ying, and Jure Leskovec. “Inductive Representation Learning on Large Graphs”. In: (June 2017). URL: https://arxiv.org/abs/1706.02216.

11. Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).

12. Franco Scarselli et al. “The graph neural network model”. In: *IEEE transactions on neural networks* 20.1 (2008), pp. 61–80.

13. Petar Velickovic et al. “Graph Attention Networks”. In: *ArXiv* abs/1710.10903 (2018).

14. Jiaxuan You et al. “Graphrnn: Generating realistic graphs with deep auto-regressive models”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5708–5717.

15. Rianne van den Berg, Thomas Kipf, and Max Welling. “Graph Convolutional Matrix Completion”. In: *ArXiv* abs/1706.02263 (2017).

16. Tian Bian et al. “Rumor Detection on Social Media with Bi-Directional Graph Convolutional Net- works”. In: *ArXiv* abs/2001.06362 (2020).

17. Zehao Dong et al. “Interpretable Drug Synergy Prediction with Graph Neural Networks for Human-AI Collaboration in Healthcare”. In: *arXiv preprint arXiv:2105.07082* (2021).

18. Christopher Morris et al. “Weisfeiler and leman go neural: Higher-order graph neural networks”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 4602–4609.

19. Keyulu Xu et al. “How powerful are graph neural networks?” In: *arXiv preprint arXiv:1810.00826* (2018).

20. AA Leman and Boris Weisfeiler. “A reduction of a graph to a canonical form and an algebra arising during this reduction”. In: *Nauchno-Technicheskaya Informatsiya* 2.9 (1968), pp. 12–16.

21. Uri Alon and Eran Yahav. “On the bottleneck of graph neural networks and its practical implications”. In: *arXiv preprint arXiv:2006.05205* (2020).

22. Muhan Zhang et al. “An end-to-end deep learning architecture for graph classification”. In: *Proceed- ings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 2018.

23. Lee, J.; Lee, I.; and Kang, J. 2019. Self-attention graph pooling. In *International conference on machine learning*, 3734–3743. PMLR.

24. Alex A Freitas. “Comprehensible classification models: a position paper”. In: *ACM SIGKDD explo- rations newsletter* 15.1 (2014), pp. 1–10.

25. 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*. 2016, pp. 1135–1144.

26. Dong Z, Chen Y, Payne P, et al. Interpreting mechanism of Synergism of drug combinations using attention based hierarchical graph pooling[J]. arXiv preprint arXiv:2209.09245, 2022.

27. Ying, Z.; Bourgeois, D.; You, J.; Zitnik, M.; and Leskovec, J. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32.

28. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J. G.; Le, Q.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive Lan- guage Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Com- putational Linguistics*, 2978–2988.

29. Al-Rfou, R.; Choe, D.; Constant, N.; Guo, M.; and Jones, L. 2019. Character-Level Language Modeling with Deeper Self-Attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3159–3166.

30. Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

31. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880.

32. Kreuzer, D.; Beaini, D.; Hamilton, W. L.; Le´tourneau, V.; and Tossou, P. 2021. Rethinking Graph Transformers with Spectral Attention. *arXiv preprint arXiv:2106.03893*.

33. Mialon, G.; Chen, D.; Selosse, M.; and Mairal, J. 2021. GraphiT: Encoding Graph Structure in Transformers. *arXiv preprint arXiv:2106.05667*.

34. Dong, Z.; Zhang, M.; Li, F.; and Chen, Y. 2022. PACE: A Parallelizable Computation Encoder for Directed Acyclic Graphs. *arXiv preprint arXiv:2203.10304*.

35. Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; and Liu, T.-Y. 2021. Do Transformers Really Perform Bad for Graph Representation? *arXiv preprint arXiv:2106.05234*.

36. Hoang, N.; Maehara, T.; and Murata, T. 2021. Revisit- ing graph neural networks: Graph filtering perspective. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 8376–8383. IEEE.

37. Zhu, M.; Wang, X.; Shi, C.; Ji, H.; and Cui, P. 2021. Interpreting and unifying graph neural networks with an optimization framework. In *Proceedings of the Web Conference 2021*, 1215–1226.

38. Pan, X.; Song, S.; and Huang, G. 2020. A unified framework for convolution-based graph neural networks.

39. Ortega, A.; Frossard, P.; Kovac˘evic´, J.; Moura, J. M.; and Van- dergheynst, P. 2018. Graph signal processing: Overview, chal- lenges, and applications. *Proceedings of the IEEE*, 106(5): 808–828.

40. Raghunathan Ramakrishnan et al. “Quantum chemistry structures and properties of 134 kilo molecules”. In: *Scientific data* 1.1 (2014), pp. 1–7.

41. Zhenqin Wu et al. “MoleculeNet: a benchmark for molecular machine learning”. In: *Chemical science* 9.2 (2018), pp. 513–530.

42. Weihua Hu et al. “Open graph benchmark: Datasets for machine learning on graphs”. In: *arXiv preprint arXiv:2005.00687* (2020).

43. Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: (2009).

44. Wu, Z.; Jain, P.; Wright, M.; Mirhoseini, A.; Gonzalez, J. E.; and Stoica, I. 2021. Representing long-range context for graph neural networks with global attention. *Advances in Neural Information Processing Systems*, 34: 13266–13279.

45. Chen, D.; O’Bray, L.; and Borgwardt, K. 2022. Structure- aware transformer for graph representation learning. In *In- ternational Conference on Machine Learning*, 3469–3489. PMLR.

46. You, J.; Gomes-Selman, J.; Ying, R.; and Leskovec, J. 2021. Identity-aware graph neural networks. *arXiv preprint arXiv:2101.10320*.

47. Zhang, M.; and Li, P. 2021. Nested Graph Neural Networks. *Advances in Neural Information Processing Systems*, 34.

48. Kovalerchuk, B., Ahmad, M.A., Science, A.T., University, C.W., Science, U.U., Systems, Tacoma, U.O., Inc., U.K., & Usa 2020. Survey of explainable machine learning with visual and granular methods beyond quasi-explanations. *arXiv, abs/2009.10221*.

49. Agrawal, Sapeck. 2022. Alzheimer’s Disease: Genes. Validated Antibody Database and Reagents.

50. Allen, M., Carrasquillo, M. M., Funk, C., Heavner, B. D., Zou, F., Younkin, C. S., Burgess, J. D., Chai, H. S., Crook, J., Eddy, J. A., Li, H., Logsdon, B., Peters, M. A., Dang, K. K., Wang, X., Serie, D., Wang, C., Nguyen, T., Lincoln, S., … Ertekin-Taner, N. (2016). Human whole genome genotype and transcriptome data for Alzheimer’s and other neurodegenerative diseases. Scientific Data, 3.

51. de Jager, P. L., Ma, Y., McCabe, C., Xu, J., Vardarajan, B. N., Felsky, D., Klein, H. U., White, C. C., Peters, M. A., Lodgson, B., Nejad, P., Tang, A., Mangravite, L. M., Yu, L., Gaiteri, C., Mostafavi, S., Schneider, J. A., & Bennett, D. A. (2018). Data descriptor: A multi-omic atlas of the human frontal cortex for aging and Alzheimer’s disease research. Scientific Data, 5.

52. Custodio, N., Montesinos, R., Chambergo-Michilot, D., Herrera-Perez, E., Pintado-Caipa, M., Seminario G, W., Cuenca, J., Mesía, L., Failoc-Rojas, V. E., & Diaz, M. M. (2022). A Functional Assessment Tool to Distinguish Controls From Alzheimer’s Disease in Lima, Peru. American Journal of Alzheimer’s Disease and Other Dementias, 37.

53. Terry, A V Jr, and J J Buccafusco. “The cholinergic hypothesis of age and Alzheimer's disease-related cognitive deficits: recent challenges and their implications for novel drug

development.” The Journal of pharmacology and experimental therapeutics vol. 306,3 (2003): 821-7. doi:10.1124/jpet.102.041616.

54. Hardy, John, and Dennis J Selkoe. “The amyloid hypothesis of Alzheimer's disease: progress and problems on the road to therapeutics.” Science (New York, N.Y.) vol. 297,5580 (2002): 353-6. doi:10.1126/science.1072994.

55. Grossberg, George T. “Cholinesterase inhibitors for the treatment of Alzheimer's disease:: getting on and staying on.” Current therapeutic research, clinical and experimental vol. 64,4 (2003): 216-35. doi:10.1016/S0011-393X(03)00059-6.

56. Danysz, Wojciech, and Chris G Parsons. “Alzheimer's disease, β-amyloid, glutamate, NMDA receptors and memantine--searching for the connections.” British journal of pharmacology vol. 167,2 (2012): 324-52. doi:10.1111/j.1476-5381.2012.02057.x.

57. Reisberg, Barry et al. “Memantine in moderate-to-severe Alzheimer's disease.” The New England journal of medicine vol. 348,14 (2003): 1333-41. doi:10.1056/NEJMoa013128.

58. Bekris, Lynn M et al. “Genetics of Alzheimer disease.” Journal of geriatric psychiatry and neurology vol. 23,4 (2010): 213-27. doi:10.1177/0891988710383571.

59. T. -A. Song et al., "Graph Convolutional Neural Networks For Alzheimer’s Disease Classification," 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 2019, pp. 414-417, doi: 10.1109/ISBI.2019.8759531.

60. Z. Qin, Z. Liu and P. Zhu, "Aiding Alzheimer's Disease Diagnosis Using Graph Convolutional Networks Based on rs-fMRI Data," 2022 15th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Beijing, China, 2022, pp. 1-7, doi: 10.1109/CISP-BMEI56279.2022.9980159.

61. Giri, Mohan et al. “Genes associated with Alzheimer's disease: an overview and current status.” *Clinical interventions in aging* vol. 11 665-81. 17 May. 2016, doi:10.2147/CIA.S105769

62. Cuyvers, Elise, and Kristel Sleegers. “Genetic variations underlying Alzheimer's disease: evidence from genome-wide association studies and beyond.” *The Lancet. Neurology* vol. 15,8 (2016): 857-868. doi:10.1016/S1474-4422(16)00127-7

63. Naj, Adam C et al. “Genomic variants, genes, and pathways of Alzheimer's disease: An overview.” *American journal of medical genetics. Part B, Neuropsychiatric genetics : the official publication of the International Society of Psychiatric Genetics* vol. 174,1 (2017): 5-26. doi:10.1002/ajmg.b.32499

64. Karch, Celeste M et al. “Alzheimer's disease genetics: from the bench to the clinic.” *Neuron* vol. 83,1 (2014): 11-26. doi:10.1016/j.neuron.2014.05.041

65. Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Appendices

Appendix A. Incorporating Attention Mechanism to Get Sparser Core Gene Sub-Network in Population-based Interpretation Machine

As visualizations in the main paper indicates, some detected core gene sub-network for revealing the mechanism of Alzheimer's progress may contain too many edges, which makes it difficult to interpret. Hence, as the summation/mean of learnt attention matrices $\sum Att^k$ reflects how much message is passed between genes, we can use $\sum Att^k \times A$ to measure the effects of each edge, so that we can decide whether keep them in the core gene sub-network.

Figure 12 illustrates the visualizations of the Pathformer-v2 (population-based explanations), where left column equipped with above edge filtering machine and right column is the original detected sub-networks. The graphs on the left column show the incorporation of the edge filtering machine and top-K method, where we selected the top 1000 genes out of 16132 genes and sorted the edges in the subgraph formed by these 1000 genes based on the edge filtering machine, which records the connection strength between genes. Depending on our needs, we can decide the number of edges required for the final core gene network, which is another reason why our proposed explanations are more flexible and easier to understand. In our experiments, we only keep top 50 edges in the core gene sub-networks. Then, each subgraph on the left column contains the top 50 edges that connect 71 genes, with the upper left subgraph consisting of one set of Mayo genes, while the lower left subgraph is made up of a separate set of Rosmap genes. Still, the green path represents the shortest path connecting each known Alzheimer's gene and risk gene variant, and the dark blue edge links 1-hop neighbors of the known Alzheimer's genes and risk gene variants above. The red nodes indicate the known Alzheimer's genes and risk variants, and the orange nodes indicate their 1-hop neighbors and the points passed on the

shortest path. The remaining light blue nodes indicate genes that may have little association with Alzheimer's disease. In the right column, we utilize the top-K method, without edge filtering machine. Experiment indicates that the proposed method can successfully make the core gene sub-networks easier to understand.

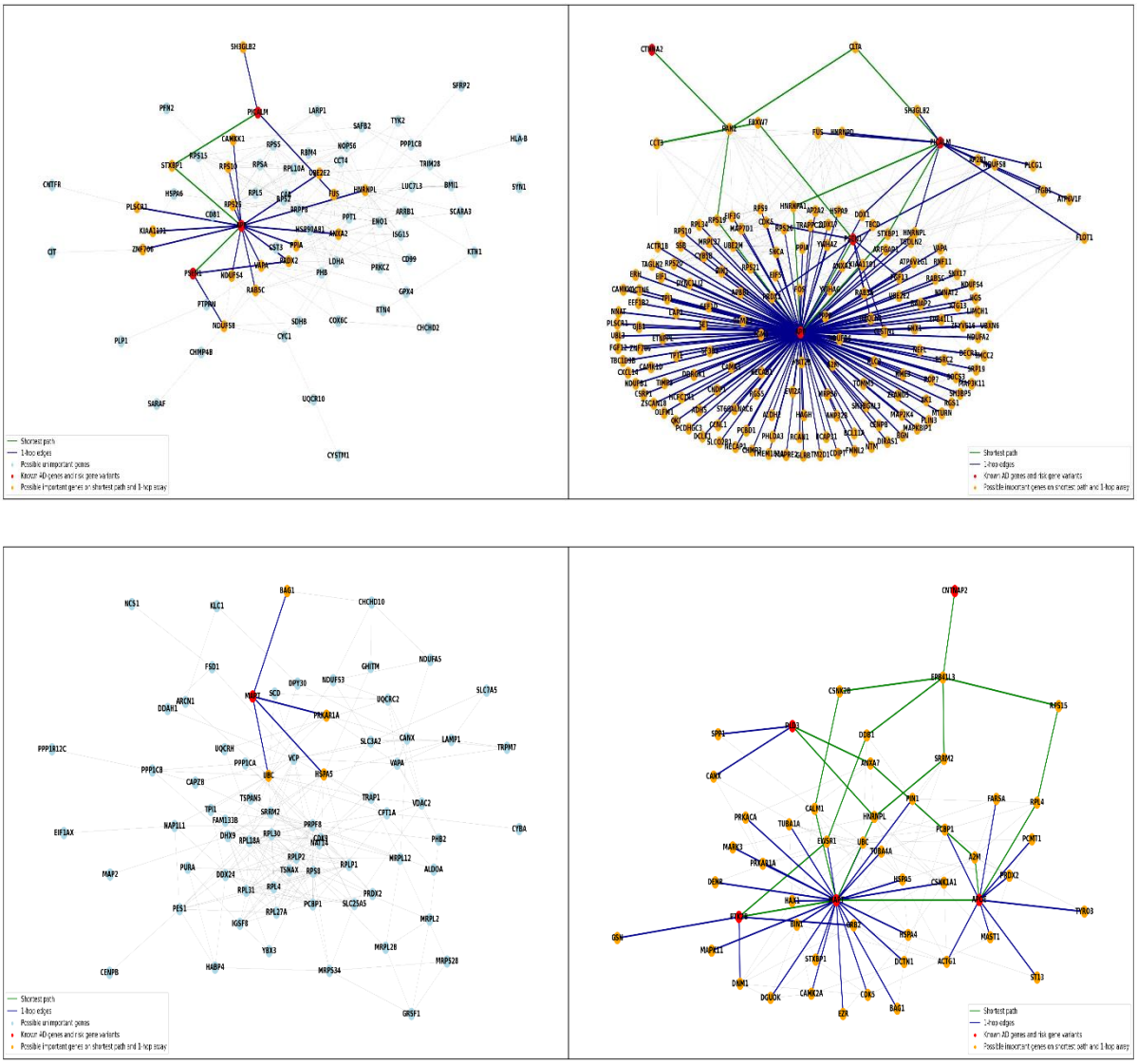


Figure 12: Visualization of detected core gene networks using Pathformer-v2 on Mayo(top) and Rosmap(bottom), where graphs on the left column show the result using incorporation of global attention matrix and top-K method, while graphs on the right column show the result only applied with top-K method.

Appendix B. Other Backgrounds

Interpretable GNN: The interpretable GNN aims to show a transparent and understandable prediction process to humans. In other words, which parts of the input have a significant impact on the prediction? For a gene network, this could be genes, relationships between correlated genes, or a combination of both, i.e., motifs [27]. Most interpretable GNNs, such as GNNExplainer [27] take a local interpretable mechanism to explain the key subgraphs of each graph. One of the assumptions behind this type of interpretation is that there are input components that contribute significantly to the prediction, while the insignificant components have less impact. Such an assumption can lead to the fact that these interpretable GNNs do a poor job of discovering the important subgraphs when the genetic features or the relationships among these genes cannot be clearly distinguished. Furthermore, local interpretability treats GNNs as black boxes [48] thus limiting human trust in the given interpretation.

Low-path Nature of GNN: In general graph learning problems like semi-supervised node classification, node features x(i) are often regarded as signals on nodes, and techniques in graph signal processing [39] are then leveraged to understand the signal characteristics. Various prior works [36,37,38] assume or observe that node features x(i) consist of low-frequency true features and noises. Based on the assumption, numerous GNNs are designed to decrease the high-frequency components in node features, thus essentially acting as a low-pass filter on graph signals. However, the assumption is not verified on gene networks. Figure 1 shows that gene networks do not benefit from omitting high-frequency components in signals. This means that the low-pass nature might not exist in the studied problem, and only keeping low-frequency signals might degrade the performance of GNNs due to the information loss.

Appendix C. Ablation Study

Effect of K in population-based interpretation machine: Here we test the effect of K in the population-based interpretation machine. To provide robust analysis, we also equip the interpretation machine with base GNNs (GIN and GCN), where the node representations are learnt by GNNs and then the interpretation results and the predictions are generated by the population-based interpretation machine. Table 1 illustrate our results. Our finding suggests that although the accuracy of the population-based interpretation machine does not exhibit significant changes on the two datasets when K increase, the number of genes in common was lower at comparable ratios than those found with K=1000, which should be reasonable because as the sample size increases, the model is able to learn more information from diverse gene expressions. As a result, the model can better identify patterns in gene expression and find more shared genes that are significant to both datasets.

Methods	K	Mayo_accuracy	Rosmap_accuracy
GIN-v2	500	0.567 ± 0.015	0.626 ± 0.024
	1000	0.551 ± 0.049	0.63 ± 0.043
GCN-v2	500	0.543 ± 0.024	0.632 ± 0.025
	1000	0.53 ± 0.025	0.629 ± 0.028
PathFormer-v2	500	0.874 ± 0.025	0.795 ± 0.016
	1000	0.865 ± 0.035	0.802 ± 0.016

Table 1: Performance of top-K models using different K values on two datasets

Table 2 also illustrates the then number of common genes in the detected core gene sub-networks of Mayo and Rosmap are not significantly different when using different graph convolution layers (PathFormer, GIN, GCN). However, as PathFormer can sparse the core sub-network as

Appendix A states, it provides users more flexibility to using larger K to detect longer dependency between Alzheimer's genes and risky genes.

Methods	Common Genes of Top 500 Genes from Same 3000 Genes
PathFormer-v2	83
GIN-v2	88
GCN-v2	90

Table 2: Common genes limited to same genes on two datasets