SHINE: SubHypergraph Inductive Neural nEtwork

Anonymous Author(s)
Affiliation
Address
email

Abstract

Hypergraph neural networks can model multi-way connections that are beyond pairwise associations among nodes of the graphs. Multi-way connections are common in many real-world applications and, in particular, genetic medicine. In particular, genetic pathways or broadly speaking gene sets encode relationships among multiple genes that collectively drive a molecular function, which can be naturally modeled as hyperedges connecting all involved nodes (e.g., genes). Thus, hypergraph-guided embedding can capture functional relations in learned representations. Existing hypergraph neural network models often focus on node-level or graph-level inference. There is an unmet need in learning powerful representations of subgraphs of hypergraphs in real-world applications. For example, a cancer patient can be viewed as a subgraph of genes harboring mutations in the patient, while all the genes are connected by hyperedges that correspond to pathways representing specific molecular functions. To achieve accurate inductive subgraph prediction, we propose SubHypergraph Inductive Neural nEtwork (SHINE). SHINE uses informative genetic pathways that encode molecular functions as hyperedges to connect genes as nodes. SHINE jointly optimizes the objectives of end-to-end subgraph classification and hypergraph nodes' similarity regularization. SHINE simultaneously learns representations for both genes and pathways using strongly dual attention message passing. These learned representations are then aggregated via a subgraph attention layer to derive a subgraph representation, which is used in training a multilayer perceptron for subgraph inferencing. We evaluated SHINE against a wide array of state-of-the-art hypergraph neural networks, XGBoost, NMF and polygenic risk score models, using diverse large scale NGS and curated datasets. SHINE outperformed all comparison models significantly, and yielded interpretable models with functional insights on disease molecular mechanisms.

1 Introduction

Hypergraph neural networks have recently emerged as a series of successful methods to model multi-way connections that are beyond pairwise associations among nodes of the graphs. Multi-way connections are common in many real-world applications and, in particular, genetic medicine. From genetic medicine's perspective, pathways or broadly speaking gene sets encode the relationship among multiple genes that collectively correspond to a molecular function [1], which account for the mechanisms of pathogenesis more intuitively and accurately than individual genes. Genetic pathways or gene sets encode functional relations among multiple genes, which can be naturally modeled as hyperedges connecting all involved nodes (e.g., genes). Thus, hypergraph-guided embedding can capture functional relations in learned representations.

Existing hypergraph neural network models often adopt the semi-supervised learning (SSL) paradigm to assign labels to initially unlabeled nodes in a hypergraph [2–4]. Other methods have focused on learning graph representations [5, 6]. Node-level and graph-level representations give either local or

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

overarching views of the graphs, i.e., at the two extremes of hypergraph topological structures. There is an unmet need in learning powerful representations of subgraphs in hypergraphs. Such capabilities are important in genetic medicine. For example, cancer patients can be viewed as subgraphs of genes that harbor mutations, while all the genes are connected by hyperedges that correspond to pathways or gene sets representing specific molecular functions. Powerful subgraph representations will lead to the capability to more accurately account for the subject's pathophysiology. For regular graphs where edges connect node pairs, several subgraph representation learning algorithms were proposed, including methods that can use the learned representations to make predictions for subgraphs with fixed sizes [7] or varying sizes [8]. There are currently few if any work on inductive inference for varying-sized subhypergraphs. In this work, we propose a new framework named SHINE: SubHypergraph Inductive Neural nEtwork. We will share our source code upon publication. Our contributions are as follows:

- To the best of our knowledge, SHINE is the first model to effectively learn subgraph representations for hypergraphs, use the learned representations (for seen subgraphs) and inductively infer representations (for unseen subgraphs) for downstream subgraph predictions.
- Novel applications in the field of genetic medicine on Next Generation Sequencing (NGS) datasets across diverse diseases show significant performance improvements by SHINE over a wide array of state-of-the-art baselines.
- In addition to learning and inductively inferring subgraph representations, SHINE simultaneously learns the representations of nodes and hyperedges. This brings interpretation advantages, allowing assessing pathways (hyperedges) correlations and reasoning multiple molecular functions mutually interacting and collectively contributing to the disease onset.

2 Related Work

Graph Neural Networks. Graph representation learning maps graphs or their components to vector representations and has attracted growing attention over the past decade. Recently, graph neural networks (GNNs), which can learn a distributed representation for a graph or a node in a graph, are widely applied to a variety of areas including social network analysis [9, 10], molecular structure inference [11, 12] and text mining [13, 14]. GNN recursively updates the representation of a node in a graph by aggregating the feature vectors of its neighbors and itself, e.g. [15]. The graph-level representations can then be obtained through set pooling (e.g., [16]) or graph coarsening (e.g., [17]) to aggregate the node representations in the graph. The reader is referred to a comprehensive book [18] on the topic of graph neural networks.

Hypergraph neural network. Hypergraph neural networks [5, 3, 2, 6] have become a popular approach for learning on multiway relations from data. Early work on hypergraph learning, e.g., [19], formulated hypergraph message passing using spectral theory of hypergraphs. This formulation and its variants [2, 3, 20–23] essentially adopted clique expansion to extend graph convolutional network (GCN) for hypergraph learning. Others methods applied attention mechanism to aggregate the information across the hypergraph [5, 6] or directly learned node representations to preserve the proximity of nodes sharing a hyperedge or having similar neighborhoods [24]. In both formulations, messages were passed to the node of interest from its immediate neighbors, and added layers allow propagation of messages to a farther neighborhood.

Subgraph representation learning and prediction. Recent studies on subgraph embeddings and prediction starts with learning representations of small subgraphs. For example, [7] encoded small fixed-sized subgraphs for subgraph evolution prediction. SubGNN [8] learned representations for varying-sized subgraphs through neighborhood, position and structure channels using random patches distributed throughout the graph. [25] and [26] learned subgraph representations by pooling across local structures to aid the predictions of the entire graphs.

The intersection of hypergraph neural network and subgraph representation learning is currently underexplored. While the above methods focus on either hypergraph learning or subgraph learning, none of the methods consider subgraph prediction for hypergraphs. Technically, subgraphs can be viewed as a hyperedge and studies on link prediction could predict the existence of a hyperedge [27]. However, few if any such studies addressed the problems of differentiating the classes of the subgraphs, which is especially important in genetic medicine where subgraphs and hyperedges have different

Table 1: Common notations used throughout the paper.

Symbol	Definition	Symbol	Definition
$\boldsymbol{\mathcal{H}}$	An undirected hypergraph	$\lvert\boldsymbol{\mathcal{E}}\rvert$	Number of hypergraph hyperedges
$\boldsymbol{\mathcal{N}}$	Set of hypergraph nodes	$\lvert\boldsymbol{\mathcal{N}}\rvert$	Number of hypergraph nodes
$\boldsymbol{\mathcal{E}}$	Set of hypergraph hyperedges	d	Hidden layer size
$\mathbf{H}$	Hyperedge incidence matrix	$\circ$	Operation composition
n	Number of subgraphs	$*$	Element-wise multiplication

real-world meanings. For example, a hyperedge corresponds to a genetic pathway from curated knowledge and a subgraph corresponds to a patient with mutated genes as nodes.

To sum, there is a major unmet need regarding varying-sized subgraph inference for hypergraphs, and even more so in the inductive learning setting. Our proposed framework SHINE provides an end-to-end framework that operates on hypergraphs and performs inductive subgraph inferencing.

3 Methods

We first outline the workflow of SHINE, refer to Table 1 for symbols used throughout the paper. This study considers both genetic pathways and genetic variants. For genetic variants, we first filter the called variants from exome sequencing data and keep those variants that pass multiple quality control steps. We then calculate aggregated mutation rate at the gene level. For pathways (gene sets), we select the gene sets from curated databases and remove redundancy. We then construct the hypergraph using gene as nodes and pathways as hyperedges. We develop a strongly dual attention message passing algorithm to propagate information between nodes and hyperedges and across layers. We develop a weighted subgraph attention mechanism to learn the subgraph representation by integrating representations of hypergraph nodes (e.g., genes harboring mutations). We next explain each step.

3.1 Collecting Genetic Pathways

To construct the graph and to establish comparison models, we use the pathways with mutated genes as an approximation of the disrupted molecular functions. We use the Molecular Signatures Database (MSigDB) [1] to obtain a comprehensive collection of curated genetic pathways. We focus on MSigDB's curated gene sets collection, which contains human gene sets that are canonical representations of a biological process compiled by domain experts. Gene sets in this collection are curated from various sources, including the biomedical literature and pathway databases such as Reactome [28] and KEGG [29]. Some pathways may overlap with others and have been filtered by MSigDB to remove interset redundancy. There are 21,587 genes in MSigDB Pathways. Genes with unknown functions are not included in the pathways and not used for classification, as our focus here is on interpretable modeling through inferencing with known molecular functions. Adding genes with unknown functions to study their impact will be our future work. With each pathway as a hyperedge, the incidence matrix entry $\mathbf{H}_{ij}$ is 1 if the hyperedge j contains the node i, and 0 otherwise.

3.2 Hypergraph Learning

We first review the hypergraph analysis theory. Different from a simple graph, a hyperedge in a hypergraph connects two or more vertices. A hypergraph is defined as $\mathcal{H} = (\mathcal{N}, \mathcal{E})$, which includes a set of nodes $\mathcal{N}$, a set of hyperedges $\mathcal{E}$. In the case of genetic medicine, we model genes as hypergraph nodes, i.e., $\mathcal{N} = \{g_1, ..., g_{|\mathcal{N}|}\}$, and pathways as hyperedges, i.e., $\mathcal{E} = \{p_1, ..., p_{|\mathcal{E}|}\}$, where $|\mathcal{N}|, |\mathcal{E}|$ are sizes of the nodes and hyperedges respectively. The hypergraph's topological structure can be denoted by an $|\mathcal{N}| \times |\mathcal{E}|$ incidence matrix $\mathbf{H}$, whose entries are defined as

$$\mathbf{H}_{ij} = \begin{cases} 1 & \text{if node } g_i \in \text{ hyperedge } p_j \\ 0 & \text{if node } g_i \notin \text{ hyperedge } p_j \end{cases} \tag{1}$$

Generally speaking, each node in the hypergraph may be accompanied by a d-dimensional node feature/embedding matrix $\mathbf{N} \in R^{|\mathcal{N}| \times d}$, where each row corresponds to a node's feature/embedding. The hypergraph with its topological structure and node features can be represented succinctly as $\mathcal{H} = (\mathbf{H}, \mathbf{N})$. For our experiments, we choose to use one-hot initialization of node embeddings, i.e.,

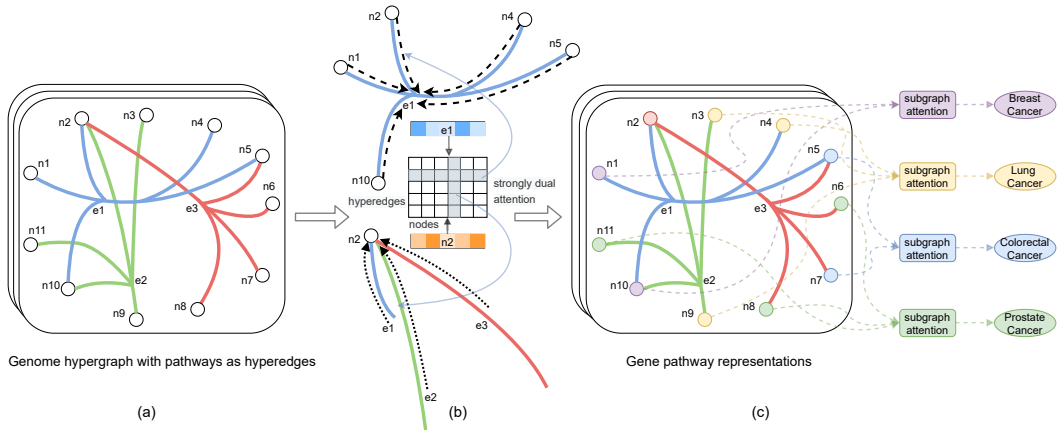

Figure 1: SHINE's strongly dual attention mechanism for message passing for the genome hypergraph, and its use of subgraph attention to integrate gene nodes in the feature learning for patients.

$\mathbf{N} = \mathbf{I}$, and let SHINE learn gene embeddings through message passing. This choice also increases the difficulty for the overall learning algorithm, thus puts SHINE under stress test.

Fig. 1 (a) shows a schematic of the constructed genome hypergraph with nodes denoted by circles and hyperedges denoted by colored arcs. While a pathway can contain multiple genes, a gene can also contribute to multiple pathways. That is, we can have multiple hyperedges incident on the same node (gene), as can be seen in Fig. 1 (a) nodes n_2, n_5, n_{10}.

3.3 Strongly Dual Attention Message Passing

Hyperedge attention over nodes. For a hyperedge edge $p_j \in \mathcal{E}$, in order to update its hidden representation at layer k, we aggregate the information from its incident nodes using the following attention mechanism. We first calculate the hyperedge attention over nodes as in

$$a_E(p_j, g_i) = \frac{\exp\left(\mathbf{c}^T \mathbf{s}(p_j, g_i)\right)}{\sum_{g_{i'} \in p_j} \exp\left(\mathbf{c}^T \mathbf{s}(p_j, g_{i'})\right)} \tag{2}$$

where $\mathbf{c}$ is a trainable context vector and the attention ready state $\mathbf{s}(p_j, g_i)$ for a hyperedge-node pair (p_j, g_i) is calculated by mapping and combining nodes' and hyperedges' states from the $(k-1)$th layer as in

$$\mathbf{s}(p_j, g_i) = \text{LeakyReLU}\Big(\left(\mathbf{W}_N \mathbf{h}_N^{k-1}(g_i) + \mathbf{b}_N\right) * \left(\mathbf{W}_E \mathbf{h}_E^{k-1}(p_j) + \mathbf{b}_E\right)\Big) \tag{3}$$

where $*$ denotes element-wise product, $\mathbf{W}_N$ and $\mathbf{b}_N$ ($\mathbf{W}_E$ and $\mathbf{b}_E$) are the transformation weights and bias of the nodes (the hyperedges) for the attention ready state. This is motivated by the observation that different nodes (genes) contribute differently to the hyperedges (pathways), thus we need proper attentions across the nodes to up- or down-weight their contributions when aggregating their representations to compute the representation of the hyperedge. Once we have the hyperedge attention over nodes, we calculate the hyperedge's representation in layer k from the nodes' representations in layer $k-1$ (equation 4) where σ is the nonlinearity layer (ReLU in our experiment).

$$\mathbf{h}_E^k(p_j) = \sigma\left(\sum\nolimits_{g_i \in p_j} a_E(p_j, g_i)\, \mathbf{h}_N^{k-1}(g_i)\right) \tag{4}$$

Node attention over hyperedges. For a node $g_i \in \mathcal{N}$, in order to update its hidden representation at layer k, we aggregate the information from its incident hyperedges using the following attention mechanism. We first calculate the node attention over hyperedges as in

$$a_N(g_i, p_j) = \frac{\exp\left(\mathbf{c}^T \mathbf{s}(p_j, g_i)\right)}{\sum_{p_{j'} \ni g_i} \exp\left(\mathbf{c}^T \mathbf{s}(p_{j'}, g_i)\right)} \tag{5}$$

where $\mathbf{c}$ is the same trainable context vector as used in hyperedge attention calculation and the attention ready state $\mathbf{s}(p_j, g_i)$ for hyperedge-node pair (p_j, g_i) is calculated as in equation 3. This allows us to

have the node's attention over the hyperedges, i.e., we can weight hyperedges' contributions when aggregating their representations to compute the representation of the node. We calculate the node's representation in layer k from the hyperedges' representations in layer $k-1$ as in

$$\mathbf{h}_N^k(g_i) = \sigma\left(\sum\nolimits_{p_j \ni g_i} a_N(g_i, p_j)\, \mathbf{h}_E^{k-1}(p_j)\right) \tag{6}$$

Note that different from HyperGAT, here the calculation of hyperedge's and node's attentions share the same underlying dual-attention matrix as shown in Fig. 1 (b), which is essentially unstandardized covariance matrix. Such parameter sharing across hyperedges and nodes allows us to cross-regulate the learning of their mutual attentions to prevent overfitting. This difference not only allows for simplification of the model, but also is more consistent with the notion of duality. The dual $\mathcal{H}^*$ of the hypergraph $\mathcal{H}$ is a hypergraph with $\mathcal{H}$'s vertices and edges interchanged, and we should have $(\mathcal{H}^*)^* = \mathcal{H}$. It is easily provable that the dual-attentions for $(\mathcal{H}^*)^*$ is the same as those for $\mathcal{H}$. Such a self-dual statement is generally not true for the attentions proposed in HyperGAT due to their unsymmetrical way of calculating the node-level and the edge-level attentions, despite that the HyperGAT attention was termed as "dual" attention. For this reason, we term our attention message passing scheme as strongly dual attention message passing.

Hypergraph regularization. One important intuition about graph and hypergraph convolutional network is that the learned representations for nodes with similar context of (hyper)edges should be similar. In the case of a simple graph $G = (V, E)$, this is to minimize the summed distance $\sum_{(u,v)\in E} \|h_u - h_v\|^2$ or its weighted variants. Instead of using it as an explicit regularizer, graph or hypergraph convolutional networks leverage an appropriately defined graph or hypergraph Laplacian. As noted in [19], the hypergraph Laplacian is $\mathbf{\Delta} = \mathbf{I} - \mathbf{\Theta}$ where $\mathbf{I}$ is the identity matrix and $\mathbf{\Theta}$ is defined as (let $\mathbf{W}$ be a diagonal matrix with diagonal entries as hyperedge weights)

$$\mathbf{\Theta} = \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2} \tag{7}$$

Here, different from hypergraph convolutional networks, we use explicit regularization on the similarity of representations of nodes with similar hyperedge context. Let $\mathbf{X}$ be the matrix of the learned nodes' representations, where row $\mathbf{X}_i = \mathbf{h}_N^K(g_i)$ and the Kth layer is the last hypergraph message passing layer. We can define the regularizer as

$$\mathcal{L}_{reg} = \sum\nolimits_{i,j} \left((\mathbf{X}_i\mathbf{X}_i^T - 2\mathbf{X}_i\mathbf{X}_j^T + \mathbf{X}_j\mathbf{X}_j^T) * \mathbf{\Theta}_{ij}\right) \tag{8}$$

Intuitively, the more hyperedges are incident on the node pair i, j, the more we should penalize their representational differences. On the other hand, the regularizer down-weights the penalization if a hyperedge connects many nodes or if a node has many incident hyperedges, indicating lack of specificity for hyperedges and nodes, respectively.

3.4 Weighted Subgraph Attention

The multiple layers of strongly dual attention message passing allow learning the nodes' and hyperedges' representations. However, the instance for the classification algorithm is a subgraph (e.g., a patient, who has mutations in multiple genes (nodes)). From the hypergraph perspective, a patient $j(1 \le j \le n)$ can be considered as a subhypergraph $\mathcal{G}_j$ whose nodes (genes) have mutations in j and are a subset of those of $\mathcal{H}$. This is shown in Fig. 1 (c) where different node colors in the hypergraph denote different patients. In order to calculate the subgraph's representation from its component nodes' representations at the Kth layer, we use the following weighted subgraph attention (WSA) mechanism. In fact, none of the previous hypergraph methods support subgraph inferencing, and we had to add our WSA module to those models for subgraph inferencing as well. We first compute the subgraph attention over nodes (e.g., g_i's) as in

$$a(\mathcal{G}_j, g_i) = \frac{\exp\left(\mathbf{M}_{ji}\mathbf{b}^T\mathbf{h}_N^K(g_i)\right)}{\sum_{g_{i'}\in\mathcal{G}_j}\exp\left(\mathbf{M}_{ji'}\mathbf{b}^T\mathbf{h}_N^K(g_{i'})\right)} \tag{9}$$

where $\mathbf{b}$ is a trainable context vector, $\mathbf{M}$ is the mutation rate feature matrix with each row corresponding to a patient and each column corresponding to a gene. Thus, equation 9 is a mutation rate weighted subgraph attention mechanism. This choice conforms to the intuition that the rate of a mutation is more informative than a categorical indicator of the mutation's occurrence. With these

subgraph level attentions, we compute the patient (subgraph) representation from the Kth layer's gene representations as in

$$\mathbf{h}(\mathcal{G}_j) = \sigma\left(\sum\nolimits_{g_i \in \mathcal{G}_j} a(\mathcal{G}_j, g_i)\ \mathbf{h}_N^K(g_i)\right) \tag{10}$$

We then stacked the learned patient representations to form the new patient feature matrix, as in

$$\mathbf{S} = [\mathbf{h}(\mathcal{G}_1)^T \mid \mathbf{h}(\mathcal{G}_2)^T \mid ... \mid \mathbf{h}(\mathcal{G}_n)^T]^T \tag{11}$$

where each row is a patient (subgraph) embedding.

3.5 Inductive Classification on Subgraphs

Let the learned feature matrix be $\mathbf{S}$ and feed it into a softmax classifier

$$\mathbf{Z} = \text{softmax}(\mathbf{W}^{(1)}\ (ReLU \circ FC)^{(2)}(\mathbf{S})\ +\ \mathbf{W}^{(0)}) \tag{12}$$

where (2) in the superscript indicates two MLP layers (FC=Fully Connected layer). The loss function is defined as the cross-entropy error over all subjects in all classes as in

$$\mathcal{L} = -\sum\nolimits_{j \in \mathcal{Y}_D} \sum\nolimits_{f=1}^{F} \mathbf{Y}_{jf} \ln \mathbf{Z}_{jf} + \mathcal{L}_{reg} \tag{13}$$

where $\mathcal{Y}_D$ is the training set of subjects that have labels and F is the dimension of the output labels, which is equal to the number of classes. $\mathbf{Y}$ is the label indicator matrix. Note that the subgraph attention layer allows us to compute any patient's representation, which effectively eliminates the need for access to test set patient features during training, making the model inductive. Existing models such as HyperGCN and HGNN are transductive, we cascade our subgraph attention layer on top of these models and make them inductive to serve as our comparison models. We implement SHINE on PyTorch, and run it on NVIDIA V100 GPUs. We train SHINE for up to 6000 epochs using Adam [30] and stop training if the validation loss does not decrease for 10 consecutive epochs.

4 Experiments

We conducted experiments on real-world datasets in genetic medicine. Both datasets have more than 20 different classes, indicating significant complexity of the prediction tasks. These datasets are different in nature, e.g., curated from literature vs. obtained directly from high-throughput sequencing, and multi-class vs. multi-class multi-label. Our experiments are motivated by the fact that massive genomic data call for novel methods and present unique technical challenges, in this case inductive subgraph inferencing on hypergraph. The summary statistics for each dataset is shown in Table 2, and the description of each dataset is as follows. Most pathways have small to medium sizes, see Table 2 pathway sizes IQR. In fact, even for 95 percentile, the pathway size is just over 200. On the other hand, we observed that the larger the pathway (hyperedge), the more subgraph it is incident on, and the less attention our model will give it as a discriminative feature. The DisGeNet and the TCGA-MC3 datasets are publicly available and this study is approved by Institutional Review Board.

4.1 Disease Type Prediction with DisGeNet Data

In this experiment, we have used the DisGeNet dataset [31] that is a collection of mutated genes involved in human diseases compiled from expert curated repositories, GWAS catalogs, animal models and the scientific literature. In the following text, we abuse terminology to use "gene" to really mean "variants in the gene". We model genes as hypergraph nodes and diseases as hyperedges. Each disease is labeled with one or more of 22 MeSH codes, and the task is a multi-class multi-label classification problem. We used 6:2:2 train:validation:test partition, and the split distribution is shown in the appendix. The DisGeNet dataset has 6226 pathways and 9133 genes involved in 8383 diseases.

4.2 Cancer Type Prediction with NGS Somatic Mutations Data

In this experiment, we have used the consensus somatic mutations for TCGA subjects produced by the Multi-Center Mutation Calling in Multiple Cancers (MC3) project [32]. Aiming to enable robust

Table 2: Real-world hypergraph datasets for subgraph inference used in our work. For hyperedge sizes, shown are medians and interquartile ranges. From the distribution, it is clear that the hyperedge size has a positive skewness.

Dataset	# hypernodes	# hyperedges	Hyperedge size	# classes	# subgraphs
DisGeNet	9133	6226	25 (12 - 57)	22	8383
TCGA-MC3	18059	6229	33 (15 - 77)	25	9012

cross-tumor-type analyses, the MC3 approach applied an ensemble of 7 mutation-calling algorithms and assigned a PASS identifier to a mutation that was called by 2 or more variant callers out of the total 7 callers [33]. The MC3 approach accounted for variance and batch effects introduced by the rapid advancement of DNA extraction, hybridization-capture, and sequencing over time. Following this approach, we restricted our analysis to PASS calls in order to maintain sample sizes and uniformity in mutation calling. Each subject is labeled with one of 25 cancer types, and the task is a multi-class classification problem. We used 6:2:2 train:validation:test partition, stratified by cancer types, and the split distribution is shown in the appendix. The TCGA-MC3 dataset has 6229 pathways and 18059 genes involved in 9012 subjects in total.

4.3 Baselines

We compared SHINE with the following state-of-the-art baselines, we use validation datasets to tune parameters and hyperparameters, please see the appendix for details.

- Hypergraph neural networks (**HGNN**) [2] uses clique expansion to transform the hypergraph to graph and computes the graph Laplacian. HGNN then uses Chebyshev approximation to derive a simplified hypergraph convolution operation.
- **HyperGCN** [3] constructed Hypergraph Laplacian by reducing the hypergraph to graph and representing a hyperedge by a selected pairwise simple edge connecting two most unlike nodes, and adds the remaining nodes in the hyperedge as mediators.
- Hypergraph attention networks (**HyperGAT**) [6] learns node representations by aggregating information from nodes to edges and vice versa. Different from SHINE, HyperGAT's alternating attention mechanism is not strictly dual attention. Moreover, there was no regularization that nodes with similar context of hyperedges should have similar representations.
- Polygenic risk score (**PRS**) is currently a standard practice widely used in genetic medicine and calculates liability to a disease according to their genotype profile using regression. Ridge regression is a frequent choice for PRS, as adopted in our experiments [34].
- Non-negative matrix factorization (**NMF**) is a dimensionality reduction tool with much success in discovering low-dimensional structure from high-dimensional omics data and enabling inference of complex biological processes [35].
- **XGBoost** XGBoost is a scalable end-to-end tree boosting system and a state-of-the-art machine learning method [36] that frequently achieves the top results on many machine learning challenges.

To assess whether performance changes are due to added information (e.g., pathway information) and/or better utilization of the added information, we run PRS, NMF, and XGBoost in the following three settings: gene features only, pathway features only, and both gene and pathway features.

5 Results

The held-out test set micro-averaged F1 scores (micro-F1) for our proposed method SHINE and all the other comparison models are in Table 3. Comparing all the models, we can see that SHINE clearly outperforms a comprehensive array of state-of-the-art baselines in various configurations, with non-overlapping standard deviation intervals. PRS is indeed a competitive baseline, as can be seen from its close performance compared with XGBoost that frequently topped many machine learning challenges' leaderboards. Previously state-of-the-art hypergraph neural network models (HyperGCN, HGNN, HyperGAT) do not always outperform PRS and XGBoost (e.g., on the TCGA-MC3 dataset). On the other hand, pathway as features do improve performance, whether alone or jointly with genes.

Table 3: Held-out test set micro-F1 on real-world datasets. Standard deviations are provided from runs with 10 random seeds. SHINE significantly outperforms all the state-of-the-art comparison models. PRS: polygenic risk score. NMF: non-negative matrix factorization. Best model in bold.

Model *Metrics*	Feature	DisGeNet Dataset Test Micro F1	TCGA-MC3 Dataset Test Micro F1
PRS	gene	0.6303	0.4981
PRS	pathway	0.6461	0.5047
PRS	gene+pathway	0.6512	0.5042
XGBoost	gene	0.6259 ± 0.0012	0.4927 ± 0.0058
XGBoost	pathway	0.6467 ± 0.0035	0.4936 ± 0.0092
XGBoost	gene+pathway	0.6486 ± 0.0036	0.5117 ± 0.0084
NMF	gene	0.6167 ± 0.0040	0.4181 ± 0.0125
NMF	pathway	0.5867 ± 0.0039	0.4842 ± 0.0057
NMF	gene+pathway	0.5847 ± 0.0045	0.4839 ± 0.0032
HyperGCN	gene+pathway	0.6638 ± 0.0028	0.4384 ± 0.0095
HGNN	gene+pathway	0.6809 ± 0.0027	0.4504 ± 0.0042
HyperGAT	gene+pathway	0.6495 ± 0.0050	0.4721 ± 0.0032
SHINE	gene+pathway	$\mathbf{0.6955 \pm 0.0034}$	$\mathbf{0.5319 \pm 0.0049}$

This comparison shows that genetic pathway information is useful to disease type classification, consistent to the intuition that pathways encode molecular functional mechanisms that underlie the disease etiology. However, properly utilizing such information is non-trivial, as evidenced by the difficulty to outperform PRS and XGBoost models by NMF models and hypergraph models including HyperGCN, HGNN and HyperGAT. Given that difficulty, SHINE still attained the best performance on each dataset. Intuitively speaking, HyperGCN and HGNN focus on similarity regularization: hypergraph nodes with similar context of hyperedges should have similar representations. HyperGAT's attention mechanism gears more towards minimizing the classification loss. SHINE, in an attempt to balance the similarity regularization with the end-to-end classification task via strongly dual attention mechanism, achieved better trade-off between the two objectives and effectively integrated the functional pathway's (hyperedge) information with individual gene's (node) information. Also note that in general, we have some performance drop when moving from the DisGeNet dataset to the TCGA-MC3 dataset, likely due to the fact that the former uses genetic features from curated literatures and the latter is from high throughput sequencing intended for data-driven discovery. Both complex classification tasks ($>$20 classes) are uniquely challenging because diseases may have overlapping disrupted molecular functions (genetic pathways, hyperedges), especially for the TCGA-MC3 experiment that is distinguishing subcategories of similar diseases as they are loosely all cancers. In addition, both tasks exhibit class distributional shift between the train and the test datasets, as shown in Appendix Tables 1 and 2, and have been designed to require inductive inference on subgraphs with highly variable hyperedge sizes. Strong performance of SHINE on these tasks thus suggests that our model can leverage its relational inductive biases for more robust generalization.

Model interpretation. SHINE simultaneously learns the representations of nodes and hyperedges, which are then used to learn and inductively infer subgraph representations. This brings model interpretation advantages as it allows assessing pathways (hyperedges) correlations and reasoning multiple molecular functions mutually interacting and collectively contributing to the disease onset. We identify the top pathways that are enriched in different cancers using the attention weights learned for SHINE, as shown in Table 4. From the table, we see that many of the listed pathways reflect innate key events in the development of individual or multiple types of cancers, consistent with genetic and medical knowledge from wet lab (e.g., TNF/Stress Related Signaling [37]). We showcase interpretations for breast cancer and lung cancer here, and refer the reader to the appendix for full interpretation of Table 4. For breast cancer, TNFα is not only closely involved in its onset, progression and in metastasis formation, but also linked to therapy resistance [37]. Regarding the 4-1BB pathway, studies have suggested HER2/4-1BB bispecific molecule as a candidate of alternative therapeutic strategy to patients in HER2-positive breast cancer [38]. VIP/PACAP and their receptors have prominent roles in transactivation of the Epidermal growth factor (EGF) family and growth effects in breast cancer [39]. For lung cancer, the ErbB3 receptor recycling controlled by neuroregulin receptor degradation protein-1 is linked to lung cancer and small inhibitory RNA (siRNA) to ErbB3

Table 4: Top enriched genetic pathways associated with different cancer risks. The text color indicates the source database for pathways that MSigDB integrated: BioCarta, Reactome, WikiPathways, Pathway Interaction Database, KEGG.

BRCA	LUAD	LGG	HNSC
Stress pathway	PTK6 stabilizes HIF1α	Citrate cycle TCA cycle	Apoptotic factor response
4-1BB pathway	ErbB3 pathway	Cytosine methylation	Programmed cell death
VIP pathway	Hypertrophic cardiomyopathy	TCA cycle and deficiency of pyruvate dehydrogenase	MECP2 regulates neuronal receptors and channels
CD40 pathway	Diseases of metabolism	Glutathione metabolism	FRA pathway
TOLL pathway	TFAP2 regulates growth factors transcription	Digestion of dietary carbohydrate	Caspase activation via extrinsic apoptotic signalling

shows promise as a therapeutic approach to treatment of lung adenocarcinoma [40]. Lung cancer is also modulated by multiple miRNAs interacting with the TFAP2 family [41].

6 Discussion, Limitation and Future Work

In addition to being significantly more accurate and interpretable, SHINE uses inductive subgraph inferencing that works well with minibatch, and scales well to large scale problems, as showcased by real-world experiments. On the other hand, our work comes with limitations. We assumed that the hyperedges are known in advance. However, in reality as our domain knowledge increases and evolves, we need to account for unknown hyperedges and, better, simultaneously discover novel hyperedges from data while predicting disease classes. Such a task has important clinical utilities in genetic medicine to discover new genetic pathways that may underlie disease etiology, and will be our future work. Another line of future work is to derive a hypergraph coarsening model on top of SHINE. SHINE currently has flat hypergraph layout and does not learn hierarchical representations of hypergraphs. In genetic medicine, often a group of hotspot mutations occur in a limited set of genes, displaying locality patterns. A flexible hypergraph coarsening model that can effectively learn hierarchical network structure out of the hypergraphs can shed light on the organizations of the hyperedges (e.g., pathways representing synergistic molecular functions). From the application point of view, detecting tumor subtypes is often interesting, and we expect to apply our method to such detections when large shared dataset will become available. To certain extent, the TCGA labels we used reflect subtypes of organ-specific primary tumors, e.g., LUAD vs. LUSC in lung cancer, KIRC vs. KIRP in kidney cancer. On the other hand, identifying drivers genes and pathways for cancer types (as experimented in this paper) continue to be biologically important, as shown in [33].

7 Conclusions

We proposed a novel framework termed SubHypergraph Inductive Neural nEtwork (SHINE) for inductive subgraph inferencing on hypergraphs, designed for jointly optimizing the objectives of end-to-end subgraph classification and similarity regularization for representations of hypergraph nodes with similar context of hyperedges. We showed that SHINE improved the performance (micro-F1) of the learned model for disease type prediction for complex (>20 classes) genetic medicine datasets of different characteristics and under different settings (e.g., multi-class and/or multi-label). Genetic pathways directly correspond to molecular mechanisms and functions, which are more informative than individual genes and are represented as hyperedges in SHINE. The novel formulation of disease classification as a subgraph inferencing problem allows a hypergraph neural network to link correlated pathways, i.e., interacting molecular mechanisms, to disease etiology. This leads to better performance with added interpretability. We compared SHINE with a wide array of state-of-the-art hypergraph neural networks, XGBoost, NMF, and PRS models with different configurations of genes and pathways as features. SHINE consistently outperformed all state-of-the-art baselines significantly in each of the disease classification and cancer classification tasks. Feature analysis of the learned pathway groups that are automatically identified by SHINE in a data-driven fashion offered significant clinical insights about multiple molecular mechanisms that interact and are associated with disease types and status.

References

[1] Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P Mesirov, and Pablo Tamayo. The molecular signatures database hallmark gene set collection. *Cell systems*, 1 (6):417–425, 2015.

[2] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3558–3565, 2019.

[3] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method of training graph convolutional networks on hypergraphs. *arXiv preprint arXiv:1809.02589*, 2018.

[4] Naganand Yadati. Neural message passing for multi-relational ordered and recursive hypergraphs. *Advances in Neural Information Processing Systems*, 33, 2020.

[5] Ruochi Zhang, Yuesong Zou, and Jian Ma. Hyper-sagnn: a self-attention based graph neural network for hypergraphs. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=ryeHuJBtPH`.

[6] Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. Be more with less: Hypergraph attention networks for inductive text classification. *arXiv preprint arXiv:2011.00387*, 2020.

[7] Changping Meng, S Chandra Mouli, Bruno Ribeiro, and Jennifer Neville. Subgraph pattern neural networks for high-order graph evolution prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[8] Emily Alsentzer, Samuel G Finlayson, Michelle M Li, and Marinka Zitnik. Subgraph neural networks. *arXiv preprint arXiv:2006.10538*, 2020.

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, pages 1024–1034, 2017.

[10] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *ACM SIGKDD*, pages 974–983. ACM, 2018.

[11] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NeurIPS*, pages 2224–2232, 2015.

[12] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272. JMLR. org, 2017.

[13] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *AAAI*, 2019.

[14] Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *WWW*, pages 1063–1072, 2018.

[15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[16] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

[17] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *arXiv preprint arXiv:1806.08804*, 2018.

[18] William L Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

[19] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems*, 19: 1601–1608, 2006.

[20] Taisong Jin, Liujuan Cao, Baochang Zhang, Xiaoshuai Sun, Cheng Deng, and Rongrong Ji. Hypergraph induced convolutional manifold networks. In *IJCAI*, pages 2670–2676, 2019.

[21] Jianwen Jiang, Yuxuan Wei, Yifan Feng, Jingxuan Cao, and Yue Gao. Dynamic hypergraph neural networks. In *IJCAI*, pages 2635–2641, 2019.

[22] Sai Nageswar Satchidanand, Harini Ananthapadmanaban, and Balaraman Ravindran. Extended discriminative random walk: A hypergraph approach to multi-view multi-relational transductive learning. In *IJCAI*, pages 3791–3797, 2015.

[23] Fuli Feng, Xiangnan He, Yiqun Liu, Liqiang Nie, and Tat-Seng Chua. Learning on partial-order hypergraphs. In *Proceedings of the 2018 World Wide Web Conference*, pages 1523–1532, 2018.

[24] Ke Tu, Peng Cui, Xiao Wang, Fei Wang, and Wenwu Zhu. Structural deep embedding for hyper-networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[25] Kexin Huang and Marinka Zitnik. Graph meta learning via local subgraphs. *Advances in Neural Information Processing Systems*, 33, 2020.

[26] Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Yuanxing Ning, Phillip S Yu, and Lifang He. Sugar: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism. *arXiv preprint arXiv:2101.08170*, 2021.

[27] Muhan Zhang, Zhicheng Cui, Shali Jiang, and Yixin Chen. Beyond link prediction: Predicting hyperlinks in adjacency space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[28] David Croft, Gavin O'Kelly, Guanming Wu, Robin Haw, Marc Gillespie, Lisa Matthews, Michael Caudy, Phani Garapati, Gopal Gopinath, Bijay Jassal, et al. Reactome: a database of reactions, pathways and biological processes. *Nucleic acids research*, 39(suppl_1):D691–D697, 2010.

[29] Minoru Kanehisa, Miho Furumichi, Mao Tanabe, Yoko Sato, and Kanae Morishima. Kegg: new perspectives on genomes, pathways, diseases and drugs. *Nucleic acids research*, 45(D1): D353–D361, 2017.

[30] DP Kingma and JL Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[31] Janet Piñero, Àlex Bravo, Núria Queralt-Rosinach, Alba Gutiérrez-Sacristán, Jordi Deu-Pons, Emilio Centeno, Javier García-García, Ferran Sanz, and Laura I. Furlong. DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Research*, 45(D1):D833–D839, 10 2016. ISSN 0305-1048. doi: 10.1093/nar/gkw943. URL `https://doi.org/10.1093/nar/gkw943`.

[32] Kyle Ellrott, Matthew H Bailey, Gordon Saksena, Kyle R Covington, Cyriac Kandoth, Chip Stewart, Julian Hess, Singer Ma, Kami E Chiotti, Michael McLellan, et al. Scalable open science approach for mutation calling of tumor exomes using multiple genomic pipelines. *Cell systems*, 6(3):271–281, 2018.

[33] Matthew H Bailey, Collin Tokheim, Eduard Porta-Pardo, Sohini Sengupta, Denis Bertrand, Amila Weerasinghe, Antonio Colaprico, Michael C Wendl, Jaegil Kim, Brendan Reardon, et al. Comprehensive characterization of cancer driver genes and mutations. *Cell*, 173(2):371–385, 2018.

[34] Shing Wan Choi, Timothy Shin-Heng Mak, and Paul F O'Reilly. Tutorial: a guide to performing polygenic risk score analyses. *Nature Protocols*, 15(9):2759–2772, 2020.

[35] Genevieve L Stein-O'Brien, Raman Arora, Aedin C Culhane, Alexander V Favorov, Lana X Garmire, Casey S Greene, Loyal A Goff, Yifeng Li, Aloune Ngom, Michael F Ochs, et al. Enter the matrix: factorization uncovers knowledge from omics. *Trends in Genetics*, 34(10):790–805, 2018.

[36] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[37] María Florencia Mercogliano, Sofía Bruni, Patricia V Elizalde, and Roxana Schillaci. Tumor necrosis factor α blockade: an opportunity to tackle breast cancer. *Frontiers in oncology*, 10, 2020.

[38] Marlon J Hinner, Rachida Siham Bel Aiba, Thomas J Jaquin, Sven Berger, Manuela Carola Dürr, Corinna Schlosser, Andrea Allersdorfer, Alexander Wiedenmann, Gabriele Matschiner, Julia Schüler, et al. Tumor-localized costimulatory t-cell engagement by the 4-1bb/her2 bispecific antibody-anticalin fusion prs-343. *Clinical Cancer Research*, 25(19):5878–5889, 2019.

[39] Terry W Moody, Bernardo Nuche-Berenguer, and Robert T Jensen. Vip/pacap, and their receptors and cancer. *Current opinion in endocrinology, diabetes, and obesity*, 23(1):38, 2016.

[40] G Sithanandam and LM Anderson. The erbb3 receptor in cancer and cancer gene therapy. *Cancer gene therapy*, 15(7):413–448, 2008.

[41] Damian Kołat, Żaneta Kałuzińska, Andrzej K Bednarek, and Elżbieta Płuciennik. The biological characteristics of transcription factors ap-2α and ap-2γ and their importance in various types of cancers. *Bioscience reports*, 39(3), 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the appendixal material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the appendixal material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We discuss the IRB approval process in Section 4.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]