Disentangling Hyperedges through the Lens of Category Theory

Yoonho Lee
KAIST
sml0399benbm@kaist.ac.kr

Junseok Lee
KAIST
junseoklee@kaist.ac.kr

Sangwoo Seo
KAIST
sangwooseo@kaist.ac.kr

Sungwon Kim
KAIST
swkim@kaist.ac.kr

Yeongmin Kim
KAIST
cytotoxicity8@kaist.ac.kr

Chanyoung Park*
KAIST
cy.park@kaist.ac.kr

Abstract

Despite the promising results of disentangled representation learning in discovering latent patterns in graph-structured data, few studies have explored disentanglement for hypergraph-structured data. Integrating hyperedge disentanglement into hypergraph neural networks enables models to leverage hidden hyperedge semantics, such as unannotated relations between nodes, that are associated with labels. This paper presents an analysis of hyperedge disentanglement from a category-theoretical perspective and proposes a novel criterion for disentanglement derived from the naturality condition. Our proof-of-concept model experimentally showed the potential of the proposed criterion by successfully capturing functional relations of genes (nodes) in genetic pathways (hyperedges). Our implementation is available at https://github.com/Yoonho-Lee-AI4Science/Natural-HNN.

1 Introduction

Disentangled representation learning, which aims to identify underlying factors behind observed data, has been applied to graph neural networks (GNNs) to capture hidden semantics or mechanisms in graph-structured data. In molecular graphs, for example, molecular properties are determined by underlying graph-level mechanisms, where specific substructures play distinct roles in shaping these properties. To reflect such graph-level mechanisms, graph-level disentanglement can be designed to capture multiple substructures, each corresponding to different molecular properties. As another example, in opinion dynamics [54, 29, 28], which studies how individuals' opinions evolve through interactions within a social network, an individual's opinion can change after engaging in discussions with neighbors. These discussions act as edge-level mechanisms that influence opinion updates. To reflect edge-level mechanisms, edge-level disentanglement can be designed to capture multiple topics underlying discussions, each affecting different aspects of individual opinions. Depending on the type of mechanism, several types of disentanglement, including node-level [47], edge-level [85], and graph-structure-level [77] approaches, have been proposed.

A fundamental challenge lies in designing a criterion for disentanglement, which determines how relevant each factor is to each mechanism (e.g., each node, edge, or subgraph). Since the representation reflects each factor in proportion to its relevance determined by the criterion, the criterion should be designed in accordance with the type of disentanglement to ensure that the intended type of mechanism is properly captured in the representation. Thus, many disentanglement models strive to identify fundamental characteristics associated with the type of disentanglement and incorporate them into the design of the criterion.

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

https://doi.org/10.52202/085713-1232

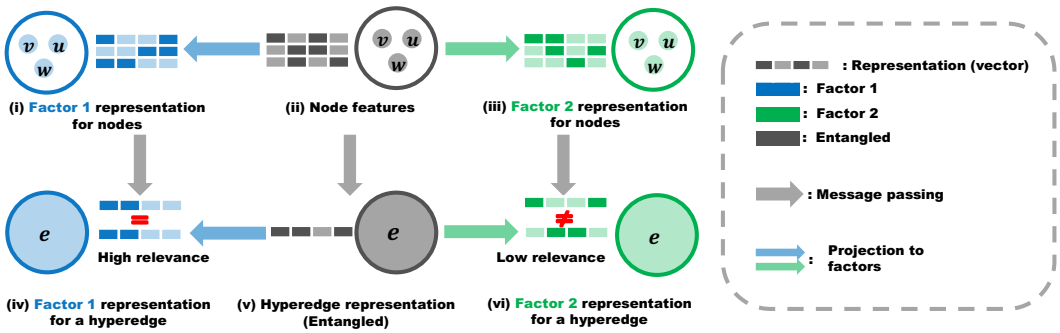


Figure 1: The factor representation consistency criterion assigns a high relevance score when the factor representation learned by two different routes is similar (i.e., consistent factor representation).

Despite numerous studies on disentanglement conducted so far [47, 85, 77, 31, 51], hyperedge disentanglement remains largely unexplored. Hyperedge disentanglement assumes that group interactions (i.e., hyperedges) have mechanisms that determine the labels, and aims to capture the factors (i.e., context or condition) that influence these group interactions. A representative example is the genetic pathway, which is a set of genes (i.e., a hyperedge) that interact to perform a specific biological function. When a pathway becomes dysregulated, its associated biological function can be impaired, potentially leading to diseases such as cancer. The functional context of a pathway can thus be regarded as an underlying factor that governs how group interaction among genes influences high-level labels, such as disease types [65, 74]. Therefore, we aim to design a criterion for hyperedge-level disentanglement that enables a model to capture hyperedge-level factors, such as the functional context of genetic pathways.

To the best of our knowledge, we are the first to propose a criterion designed for hyperedge disentanglement. To identify characteristics that can be derived from the definition of hyperedge disentanglement, rather than from any data-specific assumptions, we analyzed the hypergraph message passing neural network (MPNN) and hyperedge disentanglement from a category-theoretical perspective, as this viewpoint provides a global structural understanding of 'how the system works.' We discovered that the naturality condition holds between entangled and disentangled representations, and we used this as a characteristic associated with hyperedge disentanglement. Based on this characteristic, we defined factor representation consistency as the criterion. Figure 1 briefly illustrates our criterion for hyperedge disentanglement. As shown in Figure 1, there are two ways to obtain hyperedge factor representations: one by disentangling first and then performing message passing (i.e., $(ii) \to (i) \to (iv)$), and the other by performing message passing first and then disentangling (i.e., $(ii) \to (v) \to (iv)$). Our criterion suggests that the hyperedge factor representation learned by both methods should be similar (i.e., consistent representation) when the factor is relevant to hyperedge disentanglement. To validate whether our novel criterion can disentangle hyperedges, we created a *proof-of-concept* model, **Natural-HNN** (Naturality-guided disentangled Hypergraph Neural Network), and performed a cancer subtype classification task with hypergraphs of genetic pathways. Our model outperformed the baselines by successfully capturing the functional context of pathways, which are the underlying factors influencing group interactions.

Our main contributions are summarized as follows:

- This paper, for the first time, provides an analysis of hypergraph message passing neural networks and hyperedge disentanglement through the lens of category theory.
- Based on the analysis, we derive a novel criterion for hyperedge disentanglement. To the best of our knowledge, this is the first paper to propose a criterion for hyperedge disentanglement.
- We create a simple yet effective *proof-of-concept* model, Natural-HNN, and performed a cancer subtype classification task. Experimental results showed that the model could capture the functional context of pathways, which are factors associated with hyperedge disentanglement.

2 Related Work

In Section 2.1, we briefly describe a criterion widely used in disentangled representation learning and discuss why it may not be suitable for hyperedge disentanglement. In Section 2.2, we discuss how category theory has been applied in the field of deep learning and explain how we adopt the theory to our problem at hand. In Section 2.3, we briefly summarize several hypergraph neural networks.

2.1 Disentangled Representation Learning (DRL)

Disentangled representation learning consists of three components: factor encoder, factor discrimination loss, and criterion. A factor encoder projects the entangled representation into factor-specific representations. These factor encoders are implemented as K MLPs, where K denotes the number of factors, given as a hyperparameter. To encourage each factor representation to contain distinct information, factor discrimination losses, including factor classifier-based loss [85], factor-wise contrastive learning loss [41], and the Hilbert-Schmidt Independence Criterion (HSIC) [47], are used.

The disentanglement criterion is the most crucial component of DRL, as it determines the relevance of each factor and consequently how much it is reflected in the representation. This criterion is designed based on the characteristics that the intended disentangled factors should ideally possess. Although defining the criterion based on such ideal properties does not theoretically guarantee successful disentanglement, numerous studies have empirically confirmed that these criteria indeed enable effective disentanglement. For example, in the early image generative models that pioneered DRL, disentanglement was guided by adopting the equivariant property as the disentanglement criterion [30], since an ideal generative factor should cause the image to vary equivariantly with changes in the generative factor. Therefore, many studies have strived to identify suitable ideal characteristics for the type of disentanglement they pursue, under the assumption that such ideal properties of factors facilitate effective disentanglement.

In the field of graph and hypergraph representation learning, disentanglement has been used to exploit hidden semantics behind subgraphs or interactions with neighbors. Since such hidden semantics are highly abstract concepts, it is difficult to identify generally applicable properties. Consequently, as these semantics exhibit different characteristics depending on the data, the design of criteria has often relied heavily on assumptions about the data. The most widely used criterion for disentanglement is the factor representation similarity-based approach. For example, DisenGCN [49] assumes that the k-th factor is likely the reason behind the existence of an edge in a graph if the k-th factor representations of the two connected nodes are similar. A similar criterion is also used by HSDN [31], which performs hypergraph-structure-level disentanglement aimed at identifying substructures that contribute to hypergraph properties. The authors of the paper assume that important hyperedges would share commonalities and therefore need to have similarity in the factor representations of nodes.

However, factor representation similarity-based criterion may not be suitable for hyperedge disentanglement because the way group interactions influence labels are not necessarily related to the similarity or commonalities between participants. For instance, consider the case of opinion dynamics involving a group engaged in a discussion; the topic of such discourse need not necessarily pertain to the commonalities shared among its participants. One can easily imagine a situation where researchers from diverse fields gather to discuss and solve complex and challenging problems. As another example, in genetic pathways, the similarity of gene expression values (i.e., gene features) of the constituent genes bears no relation to the functional context. Therefore, since the existing criteria based on data-specific assumptions are not suitable for hyperedge disentanglement, we aim to develop a broadly applicable hyperedge disentanglement criterion that does not rely on heuristics.

To develop a universally applicable criterion that does not rely on heuristics or data-specific assumptions, we first need to analyze how hidden semantics are involved in the mechanisms through which group interactions contribute to labels, and to derive the corresponding characteristics or properties from this analysis. However, since the hidden semantics underlying group interactions are highly abstract concepts, conducting such an analysis is inherently challenging. To address this challenge, we employ category theory, which is well-suited for representing and analyzing systems as compositional structures. By formulating hyperedge disentanglement and investigating how factors contribute to the label-mapping mechanism through the lens of category theory, we discovered that a naturality condition must hold between the entangled and disentangled representations of nodes and hyperedges. Based on this observation, we derive a novel criterion based on hyperedge representation consistency. Finally, we conclude this section with a formal definition of hyperedge disentanglement.

Hyperedge Disentanglement. A hypergraph with N nodes and M hyperedges can be represented by incidence matrix $\mathcal{I} \in \{0,1\}^{N\times M}$, which indicates whether a node belongs to a hyperedge or not. Hyperedge disentanglement assumes the existence of multiple hidden factors underlying group interactions, which influence how labels are determined, and aims to capture these factors while predicting the labels. In other words, it is assumed that there exists a set of disentangled incidence

 https://doi.org/10.52202/085713-1232

matrices $\mathcal{I}^{dis} = \{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$ which are not explicitly provided in the data, where $\mathcal{I}_i$ denotes incidence matrix of subhypergraph for factor i. The objective of hyperedge disentanglement is to learn a hypergraph neural network $f_{HNN}(\mathcal{I}, X)$ that approximates the ground-truth label mapping function $f_{data}(\mathcal{I}^{dis}, X)$ by learning an approximation of $\mathcal{I}^{dis}$. Note that hyperedge-level disentanglement differs from hypergraph structure-level disentanglement, as the latter assumes that the presence of certain substructures determines the labels (i.e., $f_{data}(\mathcal{I}^{dis})$).

2.2 Category Theory for Deep Learning

Category theory is an abstract language of mathematics that focuses on the compositional structure of a system. One of the applications in the field of deep learning that uses category theory is neural algorithmic reasoning [68] which aims to train a neural network that can execute algorithmic computation in latent space. Several studies [14, 15] have attempted to align the computational structure of an algorithm with that of the model to effectively approximate computer algorithms. The motivation for aligning the structures comes from the theoretical conclusion [75] that structurally aligned models generalize better due to lower sample complexity (i.e., they require fewer samples in training to ensure low test error). Motivated from the works above, we analyze a hyperedge disentanglement model using category theory from the perspective that the computational structure of the model should be structurally aligned with the factor-related mechanism. Through this formulation, we identify a characteristic that can serve as a criterion. Note that the basic concepts in category theory we used are described in Appendix A.

2.3 Hypergraph Neural Networks (HNNs)

Several HNN models have been recently proposed to leverage information contained in multiway interaction. HGNN [20] and HCHA [3] use a normalized hypergraph Laplacian, which is mathematically equivalent to clique expansion (CE) [67], and apply the traditional graph convolution mechanism. HNHN [12] additionally adopts nonlinearity when calculating hyperedge representations to differentiate a hypergraph from a clique expanded graph, while UniGNN [32] unifies HNNs and GNNs into the same framework. Moreover, HyperGAT [11] adopts the attention mechanism to HNN for text classification, and SHINE [48] proposes dual attention mechanism for the disease classification task. ED-HNN [70] proposes equivariant message passing HNN, which allows hyperedges to propagate different messages to its incident nodes. AllDeepSets and AllSetTransformer [6] consider a hyperedge as a set and apply DeepSets [83] and Set Transformer [37], respectively, to increase expressive power of HNN.

Efforts to apply disentanglement to hypergraph-structured data have been relatively limited. HIDE [42] and DisenHCN [43] applied hypergraph disentanglement in the context of recommender systems. However, in these works, the hyperedge semantics were explicitly provided as hyperedge types in the data, and their approaches focused on disentangling node features corresponding to each hyperedge type, rather than capturing the underlying hyperedge semantics. HSDN [31] proposed hypergraph structure-level disentanglement, rather than hyperedge-level disentanglement.

3 Categorical Interpretation of Message Passing HNN and disentanglement

Before addressing hyperedge disentanglement, we first analyze the relationship between the mechanism by which group interactions influence labels and hypergraph MPNNs from the perspective of category theory, which will be discussed in Section 3.1. In Section 3.2, we further concretize this analysis by examining how factors relate to the mechanism and describe the process of deriving the characteristic (i.e., naturality condition) from it.

Notation. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a hypergraph, where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ indicates a set of nodes and $\mathcal{E} = \{e_1, e_2, ..., e_M\}$ indicates a set of hyperedges, where $N = |\mathcal{V}|$ and $M = |\mathcal{E}|$ are the number of nodes and the number of hyperedges in a hypergraph $\mathcal{G}$, respectively. A set of node features given as input to each layer of the model is denoted as $X = \{x_{v_1}, ..., x_{v_N}\}$, a set of hyperedge representations (calculated in each layer of the model) is denoted as $H = \{h_{e_1}, ..., h_{e_M}\}$, and a set of representations obtained after message passsing is denoted as $Y = \{y_{v_1}, ..., y_{v_N}\}$. '*en*' denotes an entangled object

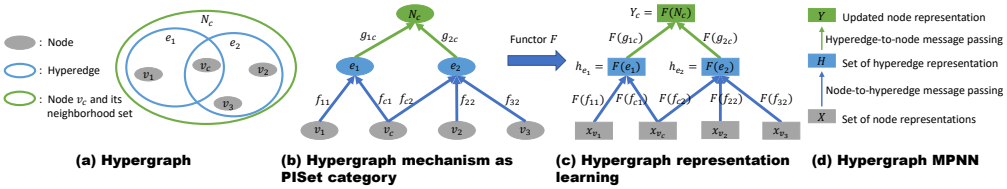


Figure 2: Compositional structure in hypergraph representation learning.

or morphism and is written in superscript or subscript, while '*dis*' denotes a disentangled object or morphism. The symbol '$\fatsemi$' is used to denote the composition of morphisms.[1]

3.1 Compositionality in Hypergraph Representation Learning

Most hypergraph representation learning methods produce the representation of a node by integrating its own representation and its neighbors' representations defined by a hypergraph topology. The fundamental assumption underlying these models is that group interactions with neighbors contribute, in some manner, to the labels. To further elucidate this assumption, consider the hypergraph example depicted in Figure 2 (a). Although not given by the hypergraph topology, we introduced the set N_c, which includes the center node v_c and its neighbors, in order to represent the information that v_c possess after message passing. Then, the assumption can be illustrated as in Figure 2 (b). Each group interaction, given as a hyperedge, can produce new meanings or information (e.g., a new meaning for e_1) through some interaction mechanisms (e.g., f_{11}, f_{c1}). Subsequently, the assumption posits that this newly generated information influences the participants (e.g., v_c) of the group interaction via some mechanism (e.g., g_{1c}, g_{2c}), thereby resulting in new information (e.g., N_c) for the participants (e.g., v_c) that may be associated with the label.

The abstract description above can be formalized through the lens of category theory. Specifically, if we consider each node as a set, since a hyperedge contains nodes, there are morphisms (inclusion) between nodes and hyperedges induced by the poset structure. We defined this as **PISet**, the category with **p**oset structure where morphisms are **i**nclusions and objects are **set**s. Thus, we can see nodes (v_1, v_c, v_2, v_3) and hyperedges (e_1, e_2) constitute **PISet** as shown in Figure 2 (b), where gray-colored nodes and hyperedges are set objects, and inclusions are morphisms (blue arrow) between sets. The same mechanism holds between hyperedges (e_1, e_2) and a set N_c that includes node v_c and its neighbors. In Figure 2 (b), for instance, we can see hyperedges (e_1, e_2) and N_c constitute **PISet** as they have morphisms (green arrow) induced by the poset structure.

In order to learn and predict with computers, such objects and morphisms must be expressed in numerical values and their transformations. Hence, we define a category of deep learning representations, **DLRep**, where objects are vector representations and morphisms are transformations between them. Figure 2 (c) shows the result of applying a functor F : **PISet** $\rightarrow$ **DLRep**, which can be simplified to a diagram in Figure 2 (d). Thus, any kind of hypergraph MPNNs can be seen as a way of learning representations and their transformations respecting compositional structure of entities. In other words, hypergraph MPNNs can be seen as structurally aligned, to some extent, with the mechanisms by which group interactions present in hypergraph data influence the labels.

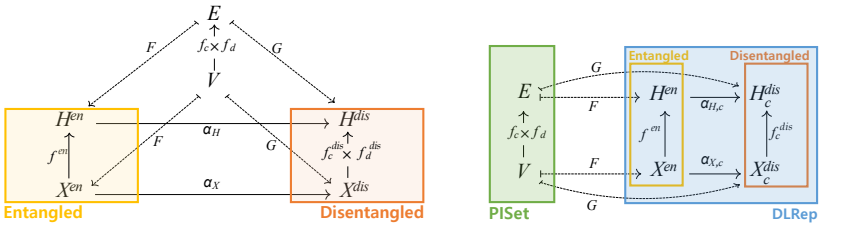
However, the degree to which a model is structurally aligned depends on implementation details. For example, convolution-based models are structurally well-aligned with mechanisms in which all nodes contribute equally during group interactions. Conversely, when node contributions vary within the group interaction, attention-based methods are more structurally aligned with the mechanism than convolution-based ones. Therefore, to perform hyperedge disentanglement, we structurally analyze how factors are involved in the mechanism and, in Section 3.2, investigate the characteristics of a hyperedge disentanglement model that is well structurally aligned with the mechanism.

3.2 Guiding Disentanglement with Naturality Condition

Since entangled and disentangled representations are different ways of representing the same compositional structure, we can regard them as the result of applying two different functors F : **PISet** $\rightarrow$ **DLRep** (for entangled representations) and G : **PISet** $\rightarrow$ **DLRep** (for disentangled representations) as shown in Figure 3 (a). Thus, we have the naturality condition between

[1]Two notations $f \fatsemi g$ and $g \circ f$ have the same meaning : "applying f first, and then applying g". We use the notation '$\fatsemi$' following [23].

https://doi.org/10.52202/085713-1232



(a) Natural Transformation in HNN (b) Natural Transformation in HNN, factor perspective

Figure 3: Naturality condition in disentangled representation learning to capture group interaction mechanism related factors. X denotes a set of node representations and H denotes hyperedge representation. V and E denote nodes and hyperedge in **PISet**. 'c' and 'd' denotes factors.

entangled and disentangled representations. Figure 3 (b) is equivalent to Figure 3 (a), but only the components related to the factor 'c' are shown (explanations are in Appendix A.6). Note that $\alpha_{X,c} = \alpha_X \fatsemi p_c$ where $p_c : X^{dis} \rightarrow X_c^{dis}$. If factor '$c$' is relevant to the morphism between node set V and hyperedge E, the naturality condition must hold for the perspective of factor 'c'. Thus, factor 'c' representation of a hyperedge (i.e., H_c^{dis}) must be the same (or similar) regardless of applying $f^{en} \fatsemi \alpha_{H,c}$ (i.e., message passing on entangled representation first, and then disentangling factors) or $\alpha_{X,c} \fatsemi f_c^{dis}$ (i.e., disentangling factors first, and then message passing on disentangled representation). In other words, the factor representation must be consistent regardless of the sequence of operations if that factor is relevant to the interaction context of a hyperedge. We use this property as a guidance for disentanglement, since it must hold for any kind of hypergraph message passing neural networks, and must work regardless of data characteristics.

4 Proof-of-concept model : Natural-HNN

To validate whether our criterion can effectively capture factors relevant to hyperedge disentanglement, we implemented a simple yet effective model, Natural-HNN. Each layer of the model is consisted with 3 components as shown in Figure 4: **1)** Node-to-hyperedge propagation step that learns hyperedge factor representations and relevance scores, which is calculated by our criterion. **2)** Hyperedge-to-Node propagation step that propagates factor representations of hyperedges to nodes with weights proportional to relevance scores. **3)** The last component concatenates factor representations and produces final outputs by interpolating with the node representations given as input to the layer. Note that each layer of Natural-HNN has K factors where K is a hyperparamter.

4.1 Node-to-Hyperedge Factor Propagation

Obtaining Two Disentangled Hyperedge Representations. To validate whether the naturality condition (Figure 4 (a)) holds, we need to get two disentangled hyperedge factor representations for every factor (i.e., H_k^{dis} for every factor $k \in [1, K]$). The two disentangled representations are obtained through 1) Aggregation-first Branch and 2) Disentalgle-first Branch. In the following, we describe how morphisms in Figure 4 (a) are implemented as operations in the two branches shown in Figure 4 (b).

- **Aggregation-first Branch.** The first disentangled representation is obtained from the aggregation-first branch performing $f^{en} \fatsemi \alpha_{H,k}$ for each factor k. This process is implemented as performing aggregation agg_{n2e} (i.e., f^{en} in Figure 4 (a)) first, and then disentangling into hyperedge factor representations using a factor encoder $\alpha_{H,k}$. The factor representations of hyperedge e_i obtained from this branch are denoted as $\tilde{h}_{e_i}^1, \ldots, \tilde{h}_{e_i}^K$.
- **Disentangle-first Branch.** The other one is obtained from the disentangle-first branch performing $\alpha_{X,k} \fatsemi f_k^{dis}$ for each factor k. This process is implemented as disentangling into node factor representations with factor encoder $\alpha_{X,k}$ first, and then performing aggregation agg_{n2e} (i.e., f_c^{dis} in Figure 4 (a)). Factor representations of hyperedge e_i obtained from this branch are denoted as $h_{e_i}^1, \ldots, h_{e_i}^K$.

For both branches, we used mean aggregation as agg_{n2e} and K MLPs as factor encoders for disentangling factors. Factor representations are vectors with size d/K (i.e., $h_{e_i}^k, \tilde{h}_{e_i}^k \in \mathbb{R}^{\frac{d}{K}}$), when the desired size for node representations after message passing is d. In summary, operations of the two branches regarding factor k can be written as follows:

$$\tilde{h}_{e_j}^k = \text{MLP}_k(\text{mean}(\{x_{v_i} | v_i \in e_j\})), \quad h_{e_j}^k = \text{mean}(\{\text{MLP}_k(x_{v_i}) | v_i \in e_j\}) \tag{1}$$

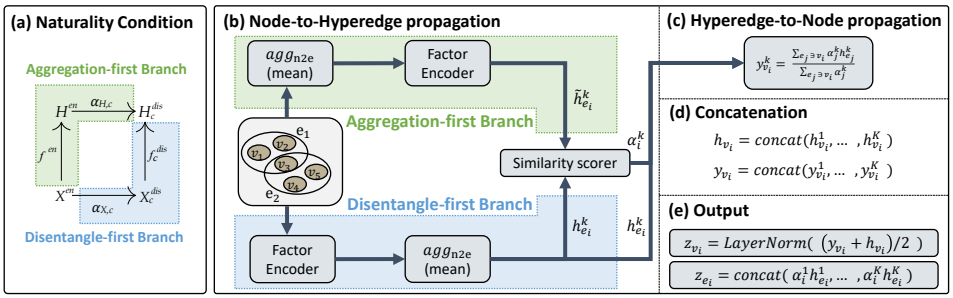


Figure 4: Architecture of proof-of-concept model Natural-HNN. It calculates the relevance of factor k (α_i^k) and performs weighted message passing for each factor.

Deciding Factors with Consistency. The extent to which the naturality condition is satisfied can be measured by calculating the similarity between the two disentangled hyperedge factor representations $\tilde{h}_{e_j}^k$ and $h_{e_j}^k$. In other words, we can consider that the naturality condition holds when the two representations are similar (i.e., consistent), and does not hold when the two representations are largely different. We introduce a similarity scorer that calculates the similarity of two L_2-normalized vectors. Specifically, we calcualte the relevance or importance of factor k for a hyperedge e_i as $\alpha_i^k = \sigma(\frac{h_{e_i}^k}{\|h_{e_i}^k\|_2} W_k \frac{\tilde{h}_{e_i}^{k^T}}{\|\tilde{h}_{e_i}^k\|_2})$, where $W_k \in \mathbb{R}^{\frac{d}{K} \times \frac{d}{K}}$ is a learnable parameter matrix for factor k, and σ is the sigmoid function. Lastly, we obtain the final hyperedge factor representations by multiplying α_i^k to the corresponding hyperedge factor representations obtained from the disentangle-first branch[2], i.e., $\alpha_i^k h_{e_i}^k$, that reflects the relevance of the factor k for the hyperedge e_i.

4.2 Hyperedge-to-Node Factor Propagation

When aggregating hyperedge representations (i.e., $\alpha_i^k h_{e_i}^k$) to update node representations, the sum of neighboring hyperedge representations with respect to factor k must be divided by the sum of α_i^k so that hyperedge relevance scores (i.e., α_i^k) are normalized during aggregation. Thus, the updated factor k representation of node v_i, i.e., $y_{v_i}^k$, can be written as $y_{v_i}^k = \frac{1}{\sum_{e_j \ni v_i} \alpha_j^k} \sum_{e_j \ni v_i} \alpha_j^k h_{e_j}^k$.

4.3 Final Output of each Layer of Natural-HNN

To allow a model to determine its focus between information from neighbors (i.e., y_{v_i}) and information from the node itself (i.e., x_{v_i}), one can introduce a hyperparameter β that determines the interpolation ratio between them (i.e., interpolate in $\beta : 1-\beta$ ratio). However, for simplicity, we set $\beta = 0.5$, so that the two pieces of information are interpolated in a 1:1 ratio. To make sure that interpolation is performed on disentangled representations, we used the factor encoder used in the message passing step (i.e., $h_{v_i}^k = \text{MLP}_k(x_{v_i})$). Specifically, $z_{v_i} = \text{LayerNorm}(0.5y_{v_i} + 0.5h_{v_i})$, where $y_{v_i} = \text{Concat}(y_{v_i}^1, \ldots, y_{v_i}^K)$, $h_{v_i} = \text{Concat}(h_{v_i}^1, \ldots, h_{v_i}^K)$.

4.4 Optional: Factor Discrimination Loss

Existing disentangled representation learning methods [47, 77] have widely adopted a factor discrimination loss aiming at promoting factors to contain different information. Following [85], we added a factor discrimination loss $\mathcal{L}_{dis}$ to the final loss, i.e., $\mathcal{L} = \mathcal{L}_{task} + \lambda \mathcal{L}_{dis}$[3], where λ is a factor discrimination loss weight given as a hyperparameter. Details can be found in the Appendix C.2. Using the factor discrimination loss increases the performance of our model (Table 7) and helps each factor to contain different information (Figure 6). However, introducing this loss requires additional hyperparameter tuning for λ, which often involves a large search space and increases experimental runtime. Considering that this loss is not closely related to our primary experimental objective—validating whether our proposed criterion captures factors relevant to hyperedge disentanglement—**we consider it an optional component** of the *proof-of-concept* model.

[2]Although we choose the disentangle-first branch here, we can instead use the output of the aggregation-first branch. Both choices give similar results. Please refer to Appendix E.1.

[3]$\mathcal{L}_{task}$ denotes the task related loss calculated from cross-entropy loss with labels and predictions. Details are available at Appendix C.3

https://doi.org/10.52202/085713-1232

Table 1: Model performance on cancer subtype classification task (Macro F1). Top two models are colored by **First**, **Second**. † : the variant of the model using multihead attention. ⋆ : $\mathcal{L}_{dis}$ is not used.

Method	BRCA	STAD	SARC	LGG	HNSC	CESC
HGNN	0.726 ± 0.053	0.563 ± 0.040	0.684 ± 0.067	0.694 ± 0.033	0.799 ± 0.053	0.835 ± 0.052
HCHA	0.704 ± 0.051	0.558 ± 0.044	0.675 ± 0.068	0.682 ± 0.041	0.783 ± 0.055	0.844 ± 0.054
HNHN	0.697 ± 0.046	0.573 ± 0.072	0.688 ± 0.075	0.674 ± 0.038	0.791 ± 0.035	0.837 ± 0.059
UniGCNII	0.697 ± 0.052	0.617 ± 0.059	**0.728** ± 0.066	0.663 ± 0.039	0.830 ± 0.030	0.841 ± 0.046
AllDeepSets	0.716 ± 0.058	0.557 ± 0.044	0.599 ± 0.058	0.665 ± 0.046	0.801 ± 0.058	0.870 ± 0.044
AllSetTransformer	0.743 ± 0.057	0.553 ± 0.046	0.719 ± 0.052	0.653 ± 0.038	0.814 ± 0.036	0.847 ± 0.046
HyperGAT	0.637 ± 0.121	0.534 ± 0.063	0.574 ± 0.153	0.665 ± 0.054	0.789 ± 0.061	0.832 ± 0.046
HyperGAT†	0.641 ± 0.115	0.502 ± 0.087	0.584 ± 0.150	0.646 ± 0.043	0.791 ± 0.079	0.827 ± 0.041
SHINE	0.446 ± 0.155	0.371 ± 0.135	0.529 ± 0.160	0.628 ± 0.104	0.718 ± 0.055	0.745 ± 0.159
SHINE†	0.651 ± 0.053	0.532 ± 0.064	0.673 ± 0.059	0.650 ± 0.046	0.770 ± 0.040	0.837 ± 0.061
HSDN	**0.757** ± 0.044	**0.629** ± 0.045	0.726 ± 0.063	0.692 ± 0.038	0.811 ± 0.044	0.867 ± 0.033
ED-HNN	0.735 ± 0.047	0.615 ± 0.050	0.718 ± 0.071	**0.700** ± 0.030	0.835 ± 0.047	0.875 ± 0.053
ED-HNNII	0.722 ± 0.045	0.536 ± 0.057	0.650 ± 0.087	0.695 ± 0.039	**0.845** ± 0.025	**0.895** ± 0.044
Natural-HNN⋆ (Ours)	**0.804** ± 0.036	**0.659** ± 0.049	**0.745** ± 0.045	**0.707** ± 0.035	**0.862** ± 0.045	**0.881** ± 0.042

5 Experiment

To evaluate our criterion, we performed a cancer subtype classification task from genetic pathways using our *proof-of-concept* model, Natural-HNN. Genetic pathways possess unannotated or hidden functional contexts (i.e., factors) underlying group interactions. Since these are closely linked to cancer and disease, they serve as appropriate data for validating the criterion. Through experiments, we aim to answer the following questions:

- **RQ1** Does Natural-HNN perform well on data where factors, such as functional context, underlying the mechanism are present? (Section 5.2)
- **RQ2** Are the factors captured by Natural-HNN related to hyperedge disentanglement? In other words, are they related to the functional context? (Section 5.3)
- **RQ3** Can Natural-HNN generalize well? And how much is Natural-HNN affected by hyperparameters? (Section 5.4)

5.1 Experimental Setup

Dataset. For the cancer subtype classification task, we downloaded clinical data for 6 cancer types (BRCA, STAD, SARC, LGG, CESC, HNSC) and preprocessed data following Pathformer [46] (Details in Appendix B.2). Every patient (i.e., a hypergraph) has the same genes (i.e., nodes) and pathways (i.e., hyperedges), but the clinical data (i.e., gene features) are different. The data statistic of each cancer data is provided in Appendix B.1.

Compared Methods. We compared Natural-HNN with HNNs introduced in Section 2.3. Specifically, HGNN[20], HCHA [3], HNHN [12], UniGCNII [32], AllDeepSets [6], AllSetTransformer [6], HyperGAT [11], SHINE [48], ED-HNN [70], ED-HNNII [70] and a hypergraph disentangling method HSDN [31] are used as baselines. Implementation details of some baselines and their variants are described in Appendix C.1.

Evaluation. We randomly split the data into 50%/25%/25% for training/validation/test set. We measured average and standard deviation of the performances for 10 different data splits. The hyperparameter search space is provided in Appendix C.5.

5.2 Results for Cancer Subtype Classification (RQ1)

The cancer subtype classification task can be considered as a hypergraph classification task, since every patient (i.e., a hypergraph) has the same genes (i.e., nodes) and pathways (i.e., hyperedges). Specifically, we generated the representation of a hyperedge by simply concatenating representations of hyperedges in a hypergraph following Pathformer [46], due to the lack of an effective pooling method reflecting the hypergraph topology developed to date. Then, we applied one layer MLP as the classifier. We have the following observations in Table 1. **1)** Natural-HNN shows superior performance in most of the cancers with large performance gap compared with most of the models. Especially in the case of BRCA, we achieve approximately a 5% performance improvement compared to the second-best model. It can be concluded that incorporating the functional context (i.e., factors) of pathways has contributed to improved performance. **2)** Natural-HNN outperforms the hypergraph-structure-level disentanglement model, HSDN, with a significant performance gap. HSDN uses a factor similarity-based criterion to determine the relevance of factors. However, the superior performance of Natural-HNN validates that naturality-guided disentanglement is more effective at integrating the context behind group interactions.

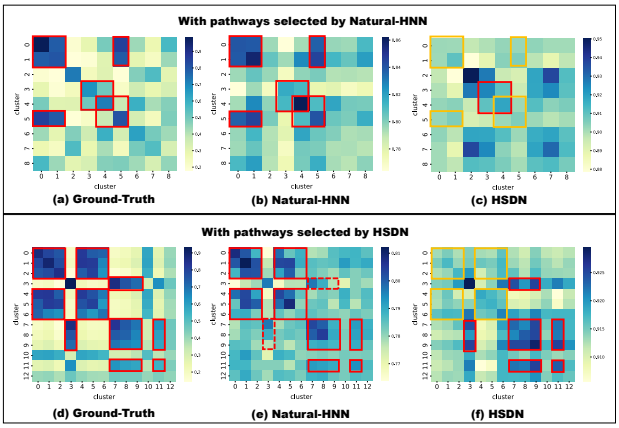


Figure 5: Captured interaction context. Captured patterns are shown in red boxes and not captured patterns are shown with orange boxes. Weakly captured cases are marked as dotted red block.

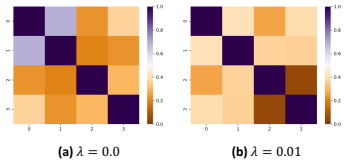


Figure 6: Pearson correlation between hyperedge factors.

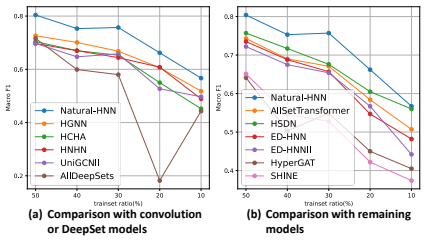


Figure 7: Marcro F1 scores with different training set ratio.

5.3 Capturing the Interaction Context of Hyperedges (RQ2)

To validate that Natural-HNN captured factors relevant to hyperedge disentanglement, we checked whether our model captures the functional semantics of genetic pathways. Because the models rely solely on cancer subtype labels during training[4], we expect the interaction contexts of informative hyperedges (such as cancer-related pathways) to be captured by the models, while non-informative hyperedges (such as pathways not relevant to cancer) are not. For this experiment, we first selected top-15 pathways[5] based on the SHAP value for each model (Natural-HNN in Figure 5 top and HSDN in Figure 5 bottom). Note that we rely on the SHAP value since information regarding which pathways are relevant to cancers is not given. Then, after clustering these 15 pathways with CliXO algorithm [34], we calculate the similarity between clusters based on the average similarity of pathways that belong to each cluster. Our goal is to check how well Natural-HNN preserves the functional semantic similarity between pathway clusters compared with the cluster similarity calculated with Lin's method [44] (BMA), which we consider as the ground-truth. For HSDN and Natural-HNN, cluster similarity is calculated based on the relevance score vector of each hyperedge e_i across all factors, i.e., $\alpha_i = [\alpha_i^1, ..., \alpha_i^K]$, which can be calculated as $1/(1 + \|\alpha_i - \alpha_j\|_2)$. As the experiment setting is somewhat complicated, we described the detailed procedure in Appendix B.3.

The result on the BRCA datset is shown in Figure 5. The row and column of each heatmap is the index of the pathway clusters and color represents similarity between clusters. Figure 5 (a), (b) and (c) shows the measured similarity between clusters with pathways selected by Natural-HNN. Comparing (b) and (c) with (a), we observe that Natural-HNN preserves the functional similarity (red box) better than HSDN, which fails to do so (orange box). Moreover, Figure 5 (d), (e) and (f) shows the measured similarity between clusters with pathways selected by HSDN. An interesting observation is that even with the pathways that were informative to the HSDN, HSDN fails (orange box) to preserve the functional similarity between clusters while Natural-HNN could capture them. The results imply that the naturality condition in category theory is effective in capturing the interaction context of a hyperedge.

Finally, we checked whether each factor captures a different context by calculating Pearson correlation coefficients among hyperedges captured by each factor, following [85]. As shown in Figure 6 (b), factors tend to exhibit only weak correlations. Note that even when factors are completely disentangled, a small degree of correlation can naturally exist between factors, as described in [59]. We observe that the factor discrimination loss decreases correlation between factors when comparing Figures 6 (b) and (a).

[4]This means that models do not use external data related to pathway types or pre-trained models.

[5]Only a few pathways are related to each type of cancer. We can also observe this with the SHAP value distribution in Figure 15 of Appendix B.4.

https://doi.org/10.52202/085713-1232

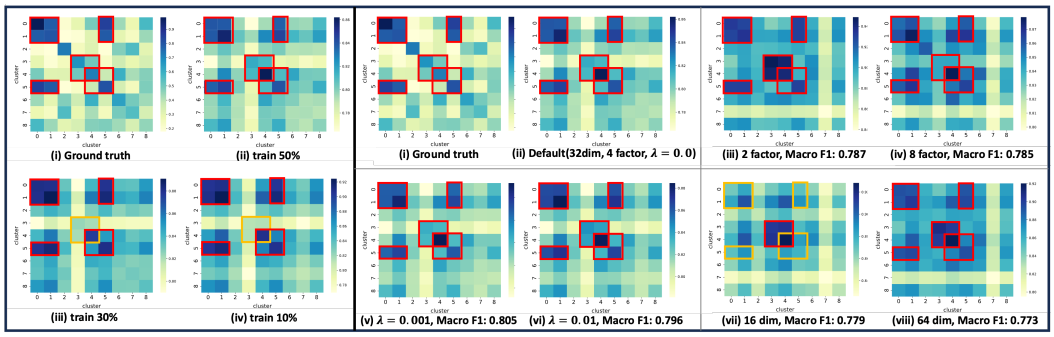


Figure 8: Captured functional context with different (a) Training set ratio and (b) Hyperparameters. Patterns that are well-captured are shown in red and those that are not captured are shown in orange.

5.4 Generalizability and hyperparameter sensitivity of Natural-HNN (RQ3)

Generalizability. To validate the generalizability of Natural-HNN, we measured performance while gradually reducing the training set ratio from 50% to 10% in 10% decrements. Figure 7 shows the performance of our model (blue) and baselines. Figure 7 (a) shows a comparison with convolution- or DeepSet-based models. These baselines rely on a strong inductive bias that nodes contribute equally to hyperedges during the group interaction process. Models with such strong inductive biases typically exhibit strong generalizability. Observing the extent of performance degradation as the training ratio decreases, we can see that Natural-HNN also demonstrates good generalizability. Figure 7 (b) compares Natural-HNN with attention-based models, which are known for their strong expressivity. As shown in the figure, Natural-HNN consistently outperforms these models. This indicates that Natural-HNN possesses both strong generalizability and sufficient expressivity. Figure 8 (a) presents experimental results evaluating whether the functional context (i.e., factors) is well captured even as the training ratio decreases. As can be seen, a significant portion of the functional context is well captured despite the reduced training data, demonstrating that our proposed criterion effectively captures the factors.

Hyperparameter Sensitivity. We conducted experiments to evaluate the impact of hyperparameters, such as the number of factors, on Natural-HNN's ability to capture factors. Figure 8 (b) reveals the following insights: **1)** When the number of factors is 2 or 8, the overall similarity tends to be slightly higher than the ground truth; however, the core strong similarities are still well captured. **2)** Regardless of the value of the factor discrimination loss weight λ, the functional context (factors) is consistently well captured. **3)** When the dimensionality is too large, the core strong similarities are well captured, but the overall similarity tends to be slightly higher than the ground truth. Conversely, when the dimensionality is too small, some functional similarities are missed. These observations suggest that, except when the dimensionality is too small, Natural-HNN can generally capture the functional context well, regardless of hyperparameter settings.

Additional Experiments. In the Appendix, we provide ablation studies (Appendix E), time complexity analysis (Appendix F.1) and results on hypergraph benchmark datasets (Appendix D).

6 Conclusion

In this work, we propose a criterion for hyperedge disentanglement by discovering a characteristic called factor representation consistency. To uncover this characteristic, we analyzed the compositional structure in hypergraph message passing and focused on the naturality condition that is satisfied between entangled and disentangled representations. The characteristic derived from a hyperedge disentanglement model that structurally aligns with the underlying mechanism demonstrated effectiveness in capturing the functional context (i.e., factors) of genetic pathways (i.e., group interactions). Experiments showed that this simple criterion generalizes well and consistently captures factors regardless of hyperparameter choices.

Acknowledgments and Disclosure of Funding

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2025-02304967, AI Star Fellowship(KAIST)). Additionally, this work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-00004) Finally, this work was supported by National Research Foundation of Korea(NRF) funded by Ministry of Science and ICT (RS-2022-NR068758). The results shown here are in whole or part based upon data generated by the TCGA Research Network: https://www.cancer.gov/tcga.

References

[1] S. A. Aleksander, J. Balhoff, S. Carbon, J. M. Cherry, H. J. Drabkin, D. Ebert, M. Feuermann, P. Gaudet, N. L. Harris, et al. The gene ontology knowledgebase in 2023. *Genetics*, 224(1): iyad031, 2023.

[2] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[3] S. Bai, F. Zhang, and P. H. Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110:107637, 2021.

[4] P. Barbiero, S. Fioravanti, F. Giannini, A. Tonda, P. Lio, and E. Di Lavore. Categorical foundations of explainable ai: A unifying formalism of structures and semantics. *arXiv preprint arXiv:2304.14094*, 2023.

[5] M. G. Bergomi and P. Vertechi. Neural network layers as parametric spans. *arXiv preprint arXiv:2208.00809*, 2022.

[6] E. Chien, C. Pan, J. Peng, and O. Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. *arXiv preprint arXiv:2106.13264*, 2021.

[7] A. Colaprico, T. C. Silva, C. Olsen, L. Garofano, C. Cava, D. Garolini, T. S. Sabedot, T. M. Malta, S. M. Pagnotta, I. Castiglioni, et al. Tcgabiolinks: an r/bioconductor package for integrative analysis of tcga data. *Nucleic acids research*, 44(8):e71–e71, 2016.

[8] D. Croft, G. O'kelly, G. Wu, R. Haw, M. Gillespie, L. Matthews, M. Caudy, P. Garapati, G. Gopinath, B. Jassal, et al. Reactome: a database of reactions, pathways and biological processes. *Nucleic acids research*, 39(suppl_1):D691–D697, 2010.

[9] G. S. Cruttwell, B. Gavranović, N. Ghani, P. Wilson, and F. Zanasi. Categorical foundations of gradient-based learning. In *European Symposium on Programming*, pages 1–28. Springer International Publishing Cham, 2022.

[10] P. de Haan, T. S. Cohen, and M. Welling. Natural graph networks. *Advances in neural information processing systems*, 33:3636–3646, 2020.

[11] K. Ding, J. Wang, J. Li, D. Li, and H. Liu. Be more with less: Hypergraph attention networks for inductive text classification. *arXiv preprint arXiv:2011.00387*, 2020.

[12] Y. Dong, W. Sawin, and Y. Bengio. Hnhn: Hypergraph networks with hyperedge neurons. *arXiv preprint arXiv:2006.12278*, 2020.

[13] A. Dudzik, T. von Glehn, R. Pascanu, and P. Veličković. Asynchronous algorithmic alignment with cocycles. *arXiv preprint arXiv:2306.15632*, 2023.

[14] A. J. Dudzik and P. Veličković. Graph neural networks are dynamic programmers. *Advances in neural information processing systems*, 35:20635–20647, 2022.

[15] A. J. Dudzik, T. von Glehn, R. Pascanu, and P. Veličković. Asynchronous algorithmic alignment with cocycles. In *Learning on Graphs Conference*, pages 3–1. PMLR, 2024.

 https://doi.org/10.52202/085713-1232

[16] S. Durinck, Y. Moreau, A. Kasprzyk, S. Davis, B. De Moor, A. Brazma, and W. Huber. Biomart and bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics*, 21(16):3439–3440, 2005.

[17] S. Durinck, P. T. Spellman, E. Birney, and W. Huber. Mapping identifiers for the integration of genomic datasets with the r/bioconductor package biomart. *Nature protocols*, 4(8):1184–1191, 2009.

[18] I. Duta, G. Cassarà, F. Silvestri, and P. Liò. Sheaf hypergraph networks. *Advances in Neural Information Processing Systems*, 36, 2024.

[19] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 264–272, 2018.

[20] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565, 2019.

[21] S. Fiorini, G. M. Bovolenta, S. Coniglio, M. Ciavotta, P. Morerio, M. Parrinello, and A. Del Bue. Dlgnet: Hyperedge classification through directed line graphs for chemical reactions. *arXiv preprint arXiv:2410.06969*, 2024.

[22] B. Fong and M. Johnson. Lenses and learners. *arXiv preprint arXiv:1903.03671*, 2019.

[23] B. Fong and D. I. Spivak. Seven sketches in compositionality: An invitation to applied category theory. *arXiv preprint arXiv:1803.05316*, 2018.

[24] B. Fong, D. Spivak, and R. Tuyéras. Backprop as functor: A compositional perspective on supervised learning. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13. IEEE, 2019.

[25] B. Gavranović. Compositional deep learning. *arXiv preprint arXiv:1907.08292*, 2019.

[26] J. Hansen and T. Gebhart. Sheaf neural networks. *arXiv preprint arXiv:2012.06333*, 2020.

[27] J. Hansen and R. Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3:315–358, 2019.

[28] J. Hansen and R. Ghrist. Opinion dynamics on discourse sheaves. *SIAM Journal on Applied Mathematics*, 81(5):2033–2060, 2021.

[29] A. Hickok, Y. Kureh, H. Z. Brooks, M. Feng, and M. A. Porter. A bounded-confidence model of opinion dynamics on hypergraphs. *SIAM Journal on Applied Dynamical Systems*, 21(1): 1–32, 2022.

[30] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner. Towards a definition of disentangled representations. arxiv. *arXiv preprint arXiv:1812.02230*, 2018.

[31] B. Hu, X. Wang, Z. Feng, J. Song, J. Zhao, M. Song, and X. Wang. Hsdn: A high-order structural semantic disentangled neural network. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8742–8756, 2022.

[32] J. Huang and J. Yang. Unignn: a unified framework for graph and hypergraph neural networks. *arXiv preprint arXiv:2105.00956*, 2021.

[33] M. Kanehisa and S. Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

[34] M. Kramer, J. Dutkowski, M. Yu, V. Bafna, and T. Ideker. Inferring gene ontologies from pairwise similarity data. *Bioinformatics*, 30(12):i34–i42, 2014.

[35] A. Kratz, M. Kim, M. R. Kelly, F. Zheng, C. A. Koczor, J. Li, K. Ono, Y. Qin, C. Churas, J. Chen, et al. A multi-scale map of protein assemblies in the dna damage response. *Cell Systems*, 14(6):447–463, 2023.

[36] H. Kvinge, B. Jefferson, C. Joslyn, and E. Purvine. Sheaves as a framework for understanding and interpreting model fit. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4222–4230, 2021.

[37] J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. W. Teh. Set transformer. 2018.

[38] Y. Lee, J. Lee, S. Seo, S. Kim, Y. Kim, and C. Park. Capturing functional context of genetic pathways through hyperedge disentanglement. In *ICLR 2025 Workshop on Machine Learning for Genomics Explorations*, 2025.

[39] T. Leinster. Basic category theory. *arXiv preprint arXiv:1612.09375*, 2016.

[40] M. Lewis. Compositionality for recursive neural networks. *arXiv preprint arXiv:1901.10723*, 2019.

[41] H. Li, X. Wang, Z. Zhang, Z. Yuan, H. Li, and W. Zhu. Disentangled contrastive learning on graphs. *Advances in Neural Information Processing Systems*, 34:21872–21884, 2021.

[42] Y. Li, C. Gao, H. Luo, D. Jin, and Y. Li. Enhancing hypergraph neural networks with intent disentanglement for session-based recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1997–2002, 2022.

[43] Y. Li, C. Gao, Q. Yao, T. Li, D. Jin, and Y. Li. Disenhcn: Disentangled hypergraph convolutional networks for spatiotemporal activity prediction. *arXiv preprint arXiv:2208.06794*, 2022.

[44] D. Lin et al. An information-theoretic definition of similarity. In *Icml*, volume 98, pages 296–304, 1998.

[45] M. Liu and P. D. Thomas. Go functional similarity clustering depends on similarity measure, clustering method, and annotation completeness. *BMC bioinformatics*, 20(1):1–15, 2019.

[46] X. Liu, Y. Tao, Z. Cai, P. Bao, H. Ma, K. Li, M. Li, Y. Zhu, and Z. J. Lu. Pathformer: a biological pathway informed transformer integrating multi-omics data for disease diagnosis and prognosis. *bioRxiv*, pages 2023–05, 2023.

[47] Y. Liu, X. Wang, S. Wu, and Z. Xiao. Independence promoted graph disentangled networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4916–4923, 2020.

[48] Y. Luo. Shine: Subhypergraph inductive neural network. *Advances in Neural Information Processing Systems*, 35:18779–18792, 2022.

[49] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu. Disentangled graph convolutional networks. In *International conference on machine learning*, pages 4212–4221. PMLR, 2019.

[50] S. M. Mansourbeigi. *Sheaf Theory as a Foundation for Heterogeneous Data Fusion*. PhD thesis, Utah State University, 2018.

[51] G. Mercatali, A. Freitas, and V. Garg. Symmetry-induced disentanglement on graphs. *Advances in neural information processing systems*, 35:31497–31511, 2022.

[52] C. H. Mermel, S. E. Schumacher, B. Hill, M. L. Meyerson, R. Beroukhim, and G. Getz. Gistic2. 0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. *Genome biology*, 12:1–14, 2011.

[53] M. Mounir, M. Lucchetta, T. C. Silva, C. Olsen, G. Bontempi, X. Chen, H. Noushmehr, A. Colaprico, and E. Papaleo. New functionalities in the tcgabiolinks package for the study and integration of cancer data from gdc and gtex. *PLoS computational biology*, 15(3):e1006701, 2019.

[54] L. Neuhäuser, M. T. Schaub, A. Mellor, and R. Lambiotte. Opinion dynamics with multi-body interactions. In *International Conference on Network Games, Control and Optimization*, pages 261–271. Springer, 2021.

 https://doi.org/10.52202/085713-1232

[55] D. Nishimura. Biocarta. *Biotech Software & Internet Report: The Computer Software Journal for Scient*, 2(3):117–120, 2001.

[56] J. H. Oh, W. Choi, E. Ko, M. Kang, A. Tannenbaum, and J. O. Deasy. Pathcnn: interpretable convolutional neural networks for survival prediction and pathway analysis applied to glioblastoma. *Bioinformatics*, 37(Supplement_1):i443–i450, 2021.

[57] Y. Qin, C. F. Winsnes, E. L. Huttlin, F. Zheng, W. Ouyang, J. Park, A. Pitea, J. F. Kreisberg, S. P. Gygi, J. W. Harper, et al. Mapping cell structure across scales by fusing protein images and interactions. *bioRxiv*, pages 2020–06, 2020.

[58] J. Reimand, R. Isserlin, V. Voisin, M. Kucera, C. Tannus-Lopes, A. Rostamianfar, L. Wadi, M. Meyer, J. Wong, C. Xu, et al. Pathway enrichment analysis and visualization of omics data using g: Profiler, gsea, cytoscape and enrichmentmap. *Nature protocols*, 14(2):482–517, 2019.

[59] K. Roth, M. Ibrahim, Z. Akata, P. Vincent, and D. Bouchacourt. Disentanglement of correlated factors via hausdorff factorized support. *arXiv preprint arXiv:2210.07347*, 2022.

[60] F. Sanchez-Vega, M. Mina, J. Armenia, W. K. Chatila, A. Luna, K. C. La, S. Dimitriadoy, D. L. Liu, H. S. Kantheti, S. Saghafinia, et al. Oncogenic signaling pathways in the cancer genome atlas. *Cell*, 173(2):321–337, 2018.

[61] C. F. Schaefer, K. Anthony, S. Krupa, J. Buchoff, M. Day, T. Hannay, and K. H. Buetow. Pid: the pathway interaction database. *Nucleic acids research*, 37(suppl_1):D674–D679, 2009.

[62] A. Sheshmani and Y.-Z. You. Categorical representation learning: morphism is all you need. *Machine Learning: Science and Technology*, 3(1):015016, 2021.

[63] D. Shiebler, B. Gavranović, and P. Wilson. Category theory in machine learning. *arXiv preprint arXiv:2106.07032*, 2021.

[64] T. C. Silva, A. Colaprico, C. Olsen, F. D'Angelo, G. Bontempi, M. Ceccarelli, and H. Noushmehr. Tcga workflow: Analyze cancer genomics and epigenomics data using bioconductor packages. *F1000Research*, 5, 2016.

[65] R. Stoney, D. L. Robertson, G. Nenadic, and J.-M. Schwartz. Mapping biological process relationships and disease perturbations within a pathway network. *NPJ systems biology and applications*, 4(1):22, 2018.

[66] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[67] L. Sun, S. Ji, and J. Ye. Hypergraph spectral learning for multi-label classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 668–676, 2008.

[68] P. Veličković and C. Blundell. Neural algorithmic reasoning. *Patterns*, 2(7), 2021.

[69] L. Vepstas. Sheaves: a topological approach to big data. *arXiv preprint arXiv:1901.01341*, 2019.

[70] P. Wang, S. Yang, Y. Liu, Z. Wang, and P. Li. Equivariant hypergraph diffusion neural operators. *arXiv preprint arXiv:2207.06680*, 2022.

[71] T. Wang, W. Shao, Z. Huang, H. Tang, J. Zhang, Z. Ding, and K. Huang. Mogonet integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification. *Nature Communications*, 12(1):3445, 2021.

[72] Y. Wang, Q. Gan, X. Qiu, X. Huang, and D. Wipf. From hypergraph energy functions to hypergraph neural networks. In *International Conference on Machine Learning*, pages 35605–35623. PMLR, 2023.

[73] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, and J. M. Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.

[74] S. F. Windels, N. Malod-Dognin, and N. Pržulj. Identifying cellular cancer mechanisms through pathway-driven data integration. *Bioinformatics*, 38(18):4344–4351, 2022.

[75] K. Xu, J. Li, M. Zhang, S. S. Du, K.-i. Kawarabayashi, and S. Jegelka. What can neural networks reason about? *arXiv preprint arXiv:1905.13211*, 2019.

[76] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. *Advances in neural information processing systems*, 32, 2019.

[77] Y. Yang, Z. Feng, M. Song, and X. Wang. Factorizable graph convolutional networks. *Advances in Neural Information Processing Systems*, 33:20286–20296, 2020.

[78] G. Yu. Gene ontology semantic similarity analysis using gosemsim. *Stem Cell Transcriptional Networks: Methods and Protocols*, pages 207–215, 2020.

[79] G. Yu, F. Li, Y. Qin, X. Bo, Y. Wu, and S. Wang. Gosemsim: an r package for measuring semantic similarity among go terms and gene products. *Bioinformatics*, 26(7):976–978, 2010.

[80] G. Yu, L.-G. Wang, Y. Han, and Q.-Y. He. clusterprofiler: an r package for comparing biological themes among gene clusters. *Omics: a journal of integrative biology*, 16(5):284–287, 2012.

[81] Y. Yuan. A categorical framework of general intelligence. *arXiv preprint arXiv:2303.04571*, 2023.

[82] Y. Yuan. On the power of foundation models. In *International Conference on Machine Learning*, pages 40519–40530. PMLR, 2023.

[83] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

[84] Y. Zhang and M. Sugiyama. A category-theoretical meta-analysis of definitions of disentanglement. In *International Conference on Machine Learning*, pages 41596–41612. PMLR, 2023.

[85] T. Zhao, X. Zhang, and S. Wang. Exploring edge disentanglement for node classification. In *Proceedings of the ACM Web Conference 2022*, pages 1028–1036, 2022.

[86] F. Zheng, M. R. Kelly, D. J. Ramms, M. L. Heintschel, K. Tao, B. Tutuncuoglu, J. J. Lee, K. Ono, H. Foussard, M. Chen, et al. Interpretation of cancer mutations using a multiscale map of protein systems. *Science*, 374(6563):eabf3067, 2021.

 https://doi.org/10.52202/085713-1232

NeurIPS Paper Checklist

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

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

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

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

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

IMPORTANT, please:

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

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: We have clearly mentioned our scope(disentangle, hypergraph, interaction context) and summarized contributions
   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: Appendix G
   Guidelines:
   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification,

asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
   Answer: [NA]
   Justification: Although our paper uses concepts in category theory, it is not about theoretical result. We used existing concepts to create our model.
   Guidelines:
   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.
4. **Experimental result reproducibility**
   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?
   Answer: [Yes]
   Justification: Appendix B
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

 https://doi.org/10.52202/085713-1232

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**
   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
   Answer: [Yes]
   Justification: We provide instructions for downloading original data, preprocessing codes at https://anonymous.4open.science/r/Natural_HNN well as preprocessed data at
   Guidelines:
   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.
6. **Experimental setting/details**
   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?
   Answer: [Yes]
   Justification: Appendix C.5
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.
7. **Experiment statistical significance**
   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
   Answer: [Yes]
   Justification: For the most of the tables, we provide standarad deviations.
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**
   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
   Answer: [Yes]
   Justification: Appendix C.6
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).
9. **Code of ethics**
   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?
   Answer: [Yes]
   Justification: We have read Code of Ethics. Our work does not violate the contents in the Code of Ethics.
   Guidelines:
   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).
10. **Broader impacts**
    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?
    Answer: [Yes]
    Justification: Appendix G
    Guidelines:
    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used

https://doi.org/10.52202/085713-1232

as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**
    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?
    Answer: [NA]
    Justification: This paper does not contain such risks
    Guidelines:
    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.
12. **Licenses for existing assets**
    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?
    Answer: [Yes]
    Justification: For biological datasets, we provided links(URL), their paper in Appendix B.2 and acknowledgement in the main text.
    Guidelines:
    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.
13. **New assets**
    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?
    Answer: [Yes]
    Justification: Yes, we have well described how to use our code in the repository.
    Guidelines:
    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.
14. **Crowdsourcing and research with human subjects**
    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
    Answer: [NA]

Justification: This paper does not contain crowdsourcing or human subject research.

Guidelines:

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

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

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

Answer: [NA]

Justification: This paper only uses publicly available datasets.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: Our model does not use LLM.

Guidelines:

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

https://doi.org/10.52202/085713-1232

Appendix

A Category Theory **24**

A.1 Category Theory 24

A.2 Category 24

A.3 Functor 25

A.4 Natural Transformation 25

A.5 Product 26

A.6 Derivation of Figure 3 (b) from Figure 3 (a). 26

B Dataset and Experiment Details **27**

B.1 Statistics : Cancer Subtype Classification Dataset 27

B.2 Preprocessing : Cancer Subtype Classification Dataset 27

B.3 Experiment Details of Capturing Context Types 28

B.4 Selecting Pathways with SHAP values 29

B.5 Calculating Functional Similarity between Pathways 30

B.6 Assigning Pathway Type with CliXO 30

B.7 Calculating Functional Similarity between clusters 31

C Implementation Details **32**

C.1 Baselines and their variants 32

C.2 Factor Discrimination Loss 32

C.3 Loss used for training $\mathcal{L}_{task}$ 32

C.4 Factor Encoder 32

C.5 Hyperparameter search space 33

C.6 Environment for experiment 34

D Standard Hypergraph Benchmark dataset **35**

D.1 Statistics : Standard Hypergraph Benchmark Dataset 35

D.2 Node Classification on Benchmark Datasets 36

D.3 Training with only 5% of data 36

E Ablation studies and Hyperparameter sensitivity **37**

E.1 Selecting Alternative Branch 37

E.2 Natural-HNN without naturality constraint 37

E.3 Hyperparameter Analysis 38

F Additional Experiment Result **39**

F.1 Computational Complexity 39

F.2 Scalability Analysis (training time) 39

https://doi.org/10.52202/085713-1232

F.3 Generalization power of Natural-HNN 39
F.4 Captured Context in CESC 41
F.5 Cancer Subtype Classification (Micro F1) 41
F.6 Chemical Reaction Classification (Hyperedge Classification) 42
F.7 Reliability of Natural-HNN in Biology 42

G Limitations, impacts and Future Work **45**
G.1 Broader Impacts 45
G.2 Limitation 45
G.3 Future Work 1 : Model for Graph Neural Network 45
G.4 Future Work 2 : Hyperedge-Node co-disentanglement 45

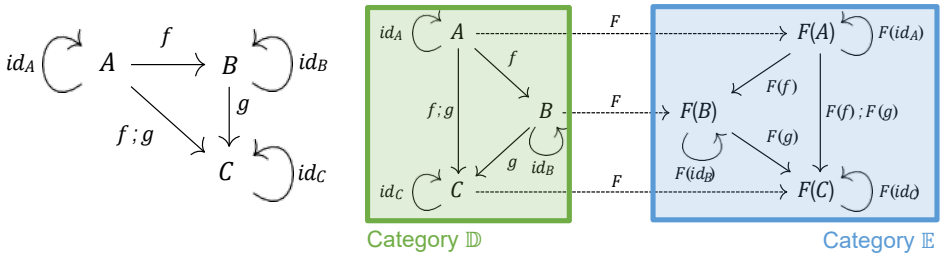
 https://doi.org/10.52202/085713-1232

A Category Theory

A.1 Category Theory

Category theory [23, 39] is widely used to represent and analyze the structure or relation of a system. Instead of focusing on the details, category theory takes bird's eye view to see global structure and patterns. Recently, category theory is used to explain learning mechanism of machine learning methods [5, 40, 25, 22, 24, 9, 63, 10, 4, 82, 13, 14, 81]. In this paper, we only use simple, fundamental concepts of category theory: category, functor, natural transformation and product.

A.2 Category



(a) Category **(b) Functor**

Figure 9: Category and Functor

A category $\mathbb{C}$ is contains four components: collection of objects, morphisms, composition rule and identities.

- Collection of objects : $Ob(\mathbb{C})$ (ex : $\{A, B, C\}$ in Figure 9 (a))
- For every pair of objects $A, B \in Ob(\mathbb{C})$, there exists a set $Hom_{\mathbb{C}}(A, B)$. Element of the set is morphism and is denoted as: $f : A \to B$.
- For every three objects $A, B, C \in Ob(\mathbb{C})$, morphisms $f \in Hom_{\mathbb{C}}(A, B)$ (i.e. $f : A \to B$) and $g \in Hom_{\mathbb{C}}(B, C)$ (i.e. $g : B \to C$), **composition rule** holds : $f \fatsemi g = g \circ f \in Hom_{\mathbb{C}}(A, C)$[6].
- For every object $A \in Ob(\mathbb{C})$, there exists an identity morphism $id_A \in Hom_{\mathbb{C}}(A, A)$ satisfying the following : $id_A \fatsemi f = f = f \fatsemi id_B$ for morphism $f : A \to B$.

Fig. 9 (a) shows an example of a category with three objects (A, B, C). For each object, there is an identity morphism (id_A, id_B, id_C). For every object pair, there is morphism $(f, g, f \fatsemi g)$ with composition rules.

One of the most important categories is **Set**. In **Set**, the objects are sets and morphisms are functions mapping two sets. The composition rule is satisfied since a composition of two functions becomes a function. Another important category is category of relations, which is denoted as **Rel**. The objects of **Rel** are sets and relations $R \subseteq A \times B$ are morphisms between objects A and B. Partially ordered set or poset can be considered as a category where objects are sets and morphisms are partial orders $\leqslant$. Since partial order is a kind of a relation, we can consider this category is a kind of **Rel**.

In Section 3, we analyzed hypergraph message passing framework, and found that, as nodes (considering node as set) are included in hyperedges, hypergraph message passing framework has poset structure with inclusion maps between them. We will define it **PISet**, a category for poset with inclusion morphisms (object is a set, morphisms are inclusions). Since inclusions are partial orders, which is also a relation, we can consider **PISet** as a kind of **Rel** category.

We can define our own category, similar to the one in a prior work [62], such that objects are vector representations and their (linear or non-linear) transformations are morphisms. We will call this a 'category of Deep Learning Representations' and denote **DLRep**.

[6]Two notations $f \fatsemi g$ and $g \circ f$ have the same meaning : "applying f first, and then applying g"

https://doi.org/10.52202/085713-1232

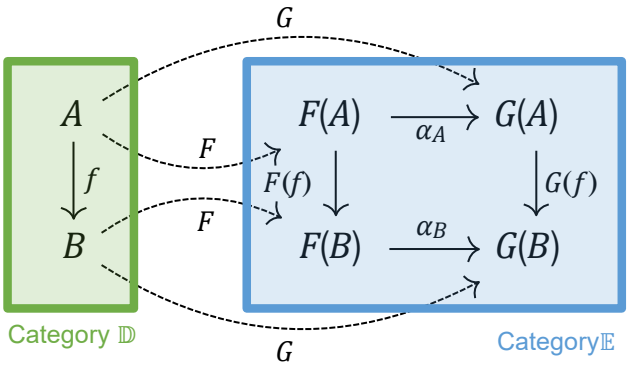


Figure 10: Natural transformation. Identity morphisms are omitted in the figure for simplicity.

A.3 Functor

Functor is a structure preserving map between categories. Objects and morphisms in one category are mapped to objects and morphisms in different category, respectively. Figure 9 (b) shows an example of a functor mapping from category $\mathbb{D}$ to category $\mathbb{E}$. Each object in category $\mathbb{D}$ (i.e., A, B, C) is mapped to objects in category $\mathbb{E}$ (i.e., $F(A), F(B), F(C)$). The morphisms, including identity morphism, and their compositions in category $\mathbb{D}$ (i.e., $id_A, id_B, id_C, f, g, f \fatsemi g$) are also mapped to morphisms in category $\mathbb{E}$ (i.e., $F(id_A), F(id_B), F(id_C), F(f), F(g), F(f) \fatsemi F(g)$). In a metaphorical sense, functors serve as bridges that connect two distinct realms while maintaining an identical compositional structure[7].

One example can be a functor mapping from **Set** to **DLRep**. Each set (object) in **Set** is mapped to a vector representation (object) in **DLRep**. Functions (morphisms) in **Set** are mapped to transformations (morphism) between vector representations in **DLRep**. This functor is related to representation learning, since entities (i.e. concept or set) are mapped to their vector representations preserving their compositional structure (relation).

A.4 Natural Transformation

Given two functors mapping from one category to another category, i.e., F and $G : \mathbb{D} \rightarrow \mathbb{E}$, natural transformation is a way of relating these two functors using morphisms in target category $\mathbb{E}$. Specifically, for each object $A \in \mathbb{D}$, there exists a morphism $\alpha_A : F(A) \rightarrow G(A)$ in $\mathbb{E}$. The natural transformation must satisfy the following condition. For every morphism $f : A \rightarrow B$ in $\mathbb{D}$,

$$F(f) \fatsemi \alpha_B = \alpha_A \fatsemi G(f) \tag{2}$$

must hold. This condition is called the ***naturality condition***. Figure 10 shows an example of natural transformation. Functors F and G map objects and morphisms in category $\mathbb{D}$ to category $\mathbb{E}$. Natural transformation $\alpha : F \Rightarrow G$ maps $F(A)$ and $F(B)$ with α_A and maps $G(A)$ and $G(B)$ with α_B. The objects and morphisms mapped by two functors as well as natural transformation α all belong to the category $\mathbb{E}$. Thus, natural transformation can be seen as a way of relating different views using morphisms in $\mathbb{E}$[8].

[7]The typical example of deep learning method using this concept is sheaf neural network [26], motivated from cellular sheaf [27]. There are also numerous studies in data science with a similar perspective [50, 69, 36].

[8]One typical example of deep learning method using this concept is Natural Graph Networks [10].

https://doi.org/10.52202/085713-1232

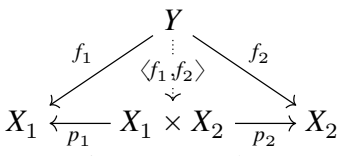


Figure 11: Product

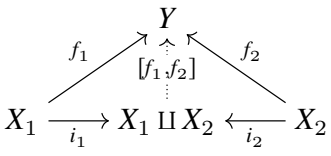


Figure 12: Coproduct

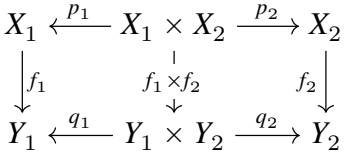


Figure 13: Product of morphisms.

A.5 Product

Product (Objects) Let $\mathbb{C}$ be a category. For two objects $X_1, X_2 \in Ob(\mathbb{C})$, one can define product of two objects $X_1 \times X_2$ with morphisms $p_1 : X_1 \times X_2 \to X_1$ and $p_2 : X_1 \times X_2 \to X_2$ which are called **projections**. Then, the composition of objects in Figure 11 must be satisfied. Given object $Y \in Ob(\mathbb{C})$ with two morphisms $f_1 : Y \to X_1$ and $f_2 : Y \to X_2$, there exists a unique morphism called 'paring' [84] $\langle f_1, f_2 \rangle : Y \to X_1 \times X_2$ that satisfies the composition : $f_1 = \langle f_1, f_2 \rangle \fatsemi p_1$ and $f_2 = \langle f_1, f_2 \rangle \fatsemi p_2$.

Coproduct (Objects)

A coproduct is the dual of a product, which can be obtained by reversing the direction of the arrows. Let $\mathbb{C}$ be a category. For two objects $X_1, X_2 \in Ob(\mathbb{C})$, one can define coproduct of two objects $X_1 \amalg X_2$ with morphisms $i_1 : X_1 \to X_1 \times X_2$ and $i_2 : X_2 \to X_1 \times X_2$ which are called **injections**. Then, the composition of objects in Figure 12 must be satisfied. Given object $Y \in Ob(\mathbb{C})$ with two morphisms $f_1 : X_1 \to Y$ and $f_2 : X_2 \to Y$, there exists a unique morphism $[f_1, f_2] : X_1 \amalg X_2 \to Y$ that satisfies the composition : $f_1 = i_1 \fatsemi [f_1, f_2]$ and $f_2 = i_2 \fatsemi [f_1, f_2]$.

Product of Morphisms

Let $\mathbb{C}$ be a category. For objects $X_1, X_2, Y_1, Y_2 \in ob(\mathbb{C})$ and morphisms $f_1 : X_1 \to Y_1$ and $f_2 : X_2 \to Y_2$, we can define **product of morphisms** $f_1 \times f_2 : X_1 \times X_2 \to Y_1 \times Y_2 := \langle p_1 \fatsemi f_1, p_2 \fatsemi f_2 \rangle$ satisfying the compositional structure shown in Figure 13.

A.6 Derivation of Figure 3 (b) from Figure 3 (a).

Since we are dealing with the commutative diagram between entangled and disentangled representations, we focus on the morphisms between $X^{en}, X^{dis}, H^{en}, H^{dis}$ in Figure 3 (a). Since the morphism (i.e., $f_c^{dis} \times f_d^{dis}$) in the disentangled representation is the product of factor-specific morphisms f_c^{dis} and f_d^{dis}, we apply the diagram at Figure 13 to $f_c^{dis} \times f_d^{dis}$. Then we can get the morphisms between $H_c^{dis}, H^{dis}, H_d^{dis}, X_c^{dis}, X^{dis}, X_d^{dis}$ in the Figure 14 where $H^{dis} = H_c^{dis} \times H_d^{dis}$ and $X^{dis} = X_c^{dis} \times X_d^{dis}$. Note that morphisms between $H^{en}, H_c^{dis}, H^{dis}, H_d^{dis}$, are products shown in the Figure 11. If we extract components in Figure 14 that are related to factor 'c' and entangled representation, we have the diagram in Figure 3 (b).

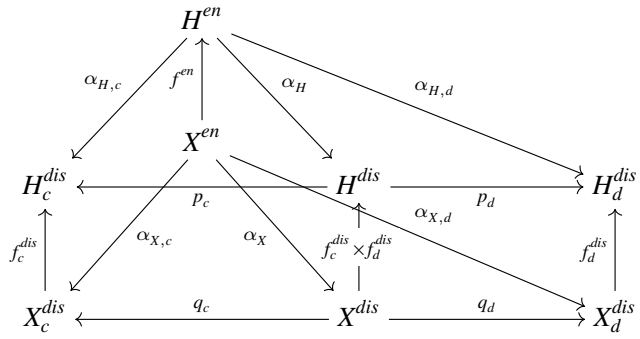


Figure 14: Derivation of Figure 3 (b) from Figure 3 (a).

https://doi.org/10.52202/085713-1232

B Dataset and Experiment Details

Note that KIPAN and NSCLC are known to be cancers where subtypes can be easily classified based on features alone [71, 56]. Because these datasets offer limited value for evaluating model performance, we excluded them in this study. The results of Natural-HNN and baselines for these datasets (i.e., KIPAN and NSCLC) are reported in [38].

B.1 Statistics : Cancer Subtype Classification Dataset

The statistics of cancer datasets are shown in the Table 2. Note that every hypergraphs in all 6 cancers have 1497 pathways (hyperedges) and 11552 genes (nodes) with 9 feature dimension. The degree statistics of cancer dataset is shown in the Table 3. When converted to a graph with star-expansion, the graph contains 98013 edges. When converted to a graph with clique-expansion, the graph contains 10114890 edges. Thus, converting the hypergraph into a graph with clique-expansion requires large computation during message passing. The downloading and preprocessing details are provided in Appendix B.2.

Table 2: Statistics of 6 cancer datasets used for cancer subtype classification task.

dataset	summary	class distribution(counts)
BRCA	5 class, 769 hypergraphs	Normal-like 33, Her2 44, Basal-like 134, LumB 143, LumA 415
STAD	5 class, 341 hypergraphs	CIN 200, EBV 29, GS 46, MSI 59, HM-SNV 7
SARC	4 class, 257 hypergraphs	LMS 104, MFS/UPS 75, DDLPS 57, Other 21
LGG	2 class, 503 hypergraphs	G2 242, G3 261
HNSC	2 class, 507 hypergraphs	HPV- 411, HPV+ 96
CESC	2 class, 280 hypergraphs	AdenoCarcinoma 46, SquamousCarcinoma 234

Table 3: statistics of hypergraphs in cancer subtype classification task

	min	median	mean	max	std
node degree	2	5	8.485	239	13.301
hyperedge degree	13	35	57	1371	84.720

B.2 Preprocessing : Cancer Subtype Classification Dataset

The overall procedure was adopted from Pathformer [46]. However, statistics of the data can be slightly different due to the difference of time at which the data was downloaded.

Creating Hypergraph

We downloaded pathways from several pathway databases including KEGG [33], PID [61], Reactome [8] and Biocarta.[55]. The pathways were selected based on their size and overlap ratio with other pathways. These two conditions must be considered as 1) extremely large pathways do not represent specific functions but rather general functions, 2) small pathways complicate interpretations 3) overlapping pathways cause redundancies. The more detailed explanations can be found in [58]. Pathways with too small or too big size or large overlaps are excluded. A specific threshold was chosen following the Pathformer.

Generating Hypergraph Labels

For BRCA and STAD, we gathered cancer subtypes from TCGA [73] using TCGAbiolinks [7, 64, 53] R library. For the rest of 4 cancer datasets we downloaded cancer subtypes from Broad GDAC Firehose (https://gdac.broadinstitute.org/)[9].

[9]Pathformer used labels from pan-cancer atlas study [60] for HNSC, CESC and SARC. However, we decided to use the one in Broad GDAC Firehose since it was easier to process the same data

https://doi.org/10.52202/085713-1232

Generating Node Features

We gathered mRNA/miRNA expression, DNA methylation[10], DNA copy number variation (CNV)[11] using TCGAbiolinks. Gene lengths were acquired from biomaRt R package [17, 16]. The procedure of processing each data with Gistic2 [52], normalization by TPM are adopted from Pathformer. At the end of the processing step, we calculate statistics (mean, min, max, count) of modalities as values for each feature dimension.

B.3 Experiment Details of Capturing Context Types

To check whether HNNs could capture functional semantics of pathways (i.e, interaction context of hyperedges), we need functional context annotations for each hyperedge. However, there is no data that annotates the functional semantics of genetic pathways. Instead, to assign function-related hyperedge types or labels, we clustered pathways based on the functional similarity between pathways, which can be calculated with computational biology method.

Now that we have obtained the hyperedge types, one might think we can simply check whether there is a one-to-one correspondence between hyperedge types and factors. However, there is another issue: hyperedge types themselves can be similar to each other. In other words, due to functional correlations between hyperedge types, a single factor may appear not in just one hyperedge type but across multiple hyperedge types. Therefore, examining the relationship between factors and hyperedge types alone makes it difficult to determine whether disentanglement has captured the functional context. Instead, we can indirectly verify that factors are related to the functional context by checking whether the functional similarity between hyperedge types aligns with the functional similarity inferred from the model's factor relevance. Thus, we evaluated whether the model effectively captured the functional context by comparing the ground truth functional similarity between hyperedge types (i.e., clusters) with the similarity inferred from the model. If the functional similarity predicted by the model shows some correlation with the functional similarity defined as ground truth, we can say that the model has captured the functional context. We do not directly compare the exact values of prediction and the ground truth since the way of calculating the value is different in prediction (calculation based on relevance scores $\alpha^k_{e_i}$) and ground truth (algorithm used in computational biology). Therefore, instead of comparing exact values, we assessed it based on the similarity of patterns observable in a heatmap, as shown in Figure 5.

In summary, our experiment involves selecting pathways to be analyzed, collecting function-related information for each pathway, measuring the functional similarity between pathways based on the collected information, and performing clustering based on this similarity. Afterward, we compute the similarity between clusters to derive the ground truth similarity, which is then compared with the model's predictions. Thus, in order to perform the experiment, we need to consider the followings: **1)** Which pathways need to be analyzed? **2)** How to get ground truth pathway functions? (i.e. How to get function related information?) **3)** How to calculate ground truth functional similarity between pathways **4)** How to cluster functionally similar pathways in a reliable manner **5)** How to measure ground truth cluster similarity and how to predict cluster similarity with model outputs.

Which pathways need to be analyzed? There are two reasons behind selecting pathways : 1) Since CliXO algorithm (Appendix B.6) used for clustering pathways takes a lot of time, the number of pathways to be analyzed must be reduced. 2) The ground truth functional similarity (Appendix B.5) contains vast biological context derived from biological domain knowledge or researches, which might not be present in our dataset. Since our dataset contains only cancer-specific information, there is no way to capture non-existing context (contexts that are not related to cancer) without external supervision. Thus direct comparison between the ground truth and our result is impossible. The most ideal way for fair comparison would be selecting the ground truth that is only relevant to our dataset or task. However, it is impossible since there are no databases with annotated context (cancer or environment) specific pathway functionalities. An alternative way was selecting the pathways that were informative or important in the decision of the model. If a model can correctly capture functional context of pathways, since pathway functions are highly related to the cancers [74, 65], informative pathways (for the model prediction) are the pathways that contain cancer-specific contexts. Since we only need to check whether functional context are correctly captured under the cancer specific

[10] but we do not use promoter methylation

[11] but we do not use gene level CNV

circumstances or condition, by selecting those pathways, we can compare functional similarities that are specific to our data or cancer[12]. The details for selecting pathways are described in Appendix B.4.

How to get ground truth pathway functions. Since there is no database that annotates functional similarity scores between pathways, we rely on methods used in computational biology. Hence, we need to get pathway function information. Similarity calculations and clusterings are based on the annotation of pathway functions. The details are described in Appendix B.5.

How to calculate ground truth functional similarity between pathways. Based on the functions of pathways, pathway functional similarity can be calculated. The calculated similarity will be used in clustering and generating ground truth functional similarity between clusters. The details are dealt in Appendix B.5.

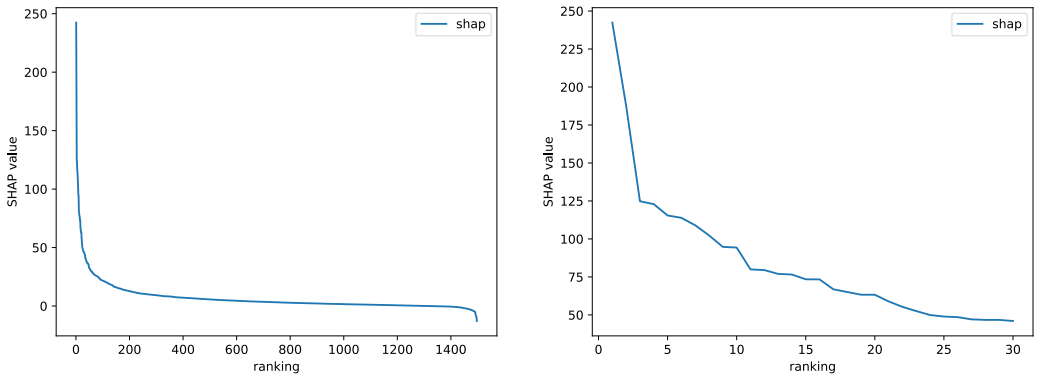
How to cluster functionally similar pathways in a reliable manner. With functional similarity between pathways, we can cluster functionally similar pathways with CliXO algorithm. The details and example results are shown in Appendix B.6.

How to measure ground truth cluster similarity and how to predict cluster similarity with model outputs. Finally, we need to devise a way to measure the similarity between clusters based on the model outputs. Also, we need to measure ground truth functional similarity between clusters. The details are described in Appendix B.7.

In summary, the procedure of experiments can be described as follows. First, we get functional annotation of pathways (hyperedges). Second, we calculate functional similarity between pathways based on annotations. Third, we select pathways to be analyzed based on the model output. Fourth, we cluster the selected pathways with pathway similarity. Finally, we calculate the predicted functional similarity between clusters from model prediction and compare that with the ground truth cluster similarity.

B.4 Selecting Pathways with SHAP values

To select pathways that were the most informative for prediction, we provide the final representation of pathways generated by a model, 1 layer classifier (MLP) that predicts labels from final representation as well as labels to the DeepExplainer to get SHAP values. Then we select top-k pathways based on the SHAP value. Note that only small number of pathways are relevant to the task as shown in Figure 15. This is due to the fact that not all pathways are related to very specific type of cancer. Although Natural-HNN and HSDN both use the same number of pathways (top-k), the pathways selected by each model can be different. This also leads to different number of clusters in Figure 5 and 18.



(a) SHAP value for all pathways **(b) SHAP value for top-30 pathways**

Figure 15: SHAP value distribution of Natural-HNN on BRCA dataset. We sorted pathways with SHAP value. X axis represents ranking of pathways and Y axis represents SHAP value for pathways with corresponding ranking.

[12]On the other hand, if the model could not correctly capture pathway functionalities, cancer irrelevant pathways will be selected and will have different result from the ground truth in section 5.3

https://doi.org/10.52202/085713-1232

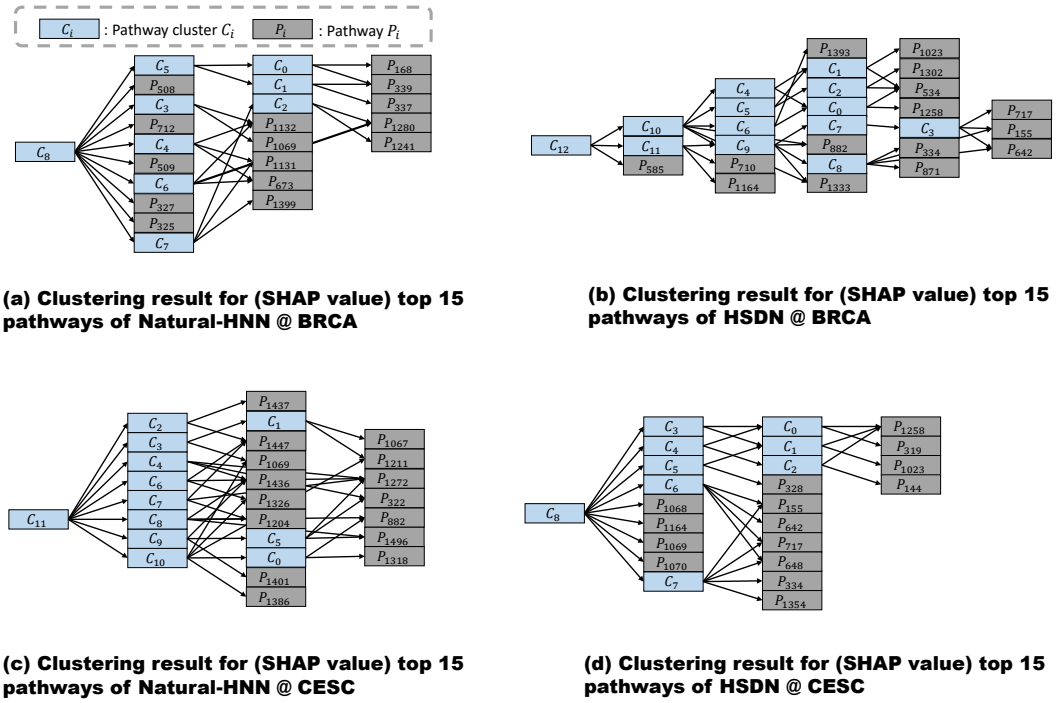


Figure 16: The result of applying CliXO algorithm to top-15 pathways of Natural-HNN and HSDN on BRCA and CESC. The pathway number denotes the index of pathway in our dataset (hyperedge index).

B.5 Calculating Functional Similarity between Pathways

This process consists of two steps: 1) assigning pathway level function to pathways and 2) calculating functional semantic similarities between pathways. For both two steps, we adopted the most frequently used and verified methods through several studies. For the assignment of pathway functions, we use GO enrichment analysis. Gene ontology (GO) [2, 1] is a functional annotation of genes that has a hierarchical structure. Note that, however, the hierarchical structure of functional annotations is close to a directed acyclic graph (DAG) rather than a tree-like hierarchical structure. As an example, we can see DAG structure in the result of CliXO algorithm in the Figure 16. We can computationally annotate pathway functions with GO terms using GO enrichment analysis. We use 'enrichGO' function provided by R package clusterProfiler [80], with pvalue of 0.01 followig the paper [65]. Then we selected the most specific GO terms with set cover algorithm proposed in [65] to assign pathways precise representation of their functions.

The next step is calculating functional semantic similarities between pathways. We used Lin's method [44] with best matching average (BMA) as the combination was proven to perform well with CliXO and was proven to be robust in incomplete annotation cases in [45]. We used mgoSim function in R package GOSemSim [79, 78] for the calculation of Lin's method.

B.6 Assigning Pathway Type with CliXO

To cluster functionally similar pathways, we adopted CliXO [34]. It was originally designed to cluster gene function annotations (GO) and has been used in multiple biological studies[35, 57]. However, it can also be effectively applied to higher functional semantics such as pathways as in [86]. We used official implementation of CliXO 1.0 for our research. We used the following 4 values as hyperparameter of CliXO : a = 0.1, b = 0.6, m = 0.005, s = 0.2.

Since CliXO can cluster functionally similar pathways, we can assign interaction types to pathways by assigning them to the cluster. Figure 16 shows the result of applying CliXO for top-15 pathways selected by Natural-HNN or HSDN for BRCA as well as CESC. Unlike other hierarchical clustering based methods, CliXO created clusters having DAG structure. Considering that GO also has DAG structure, CliXO can be seen as a natural way of reflecting complex structure or relations in biology.

B.7 Calculating Functional Similarity between clusters

Ground Truth Given a pair of clusters, calculating functional similarity between them is simple. We average the similarity of all possible pathway pairs belonging to different clusters to get functional similarity between clusters.

Model's prediction If a model correctly captures functional context of pathways, then the relevance scores (α_i^k) of two similar pathways must be similar for all factors. Thus we define the similarity between pathways as $\frac{1}{1+\|\alpha_i-\alpha_j\|_2}$, where $\alpha_i = [\alpha_i^1, ..., \alpha_i^K]$ is a factor vector of pathway (hyperedge) e_i. The cluster similarity can be calculated in the same way as in the ground truth case. We average the similarity of all possible pathway pairs belonging to different clusters to get functional similarity between clusters.

https://doi.org/10.52202/085713-1232

C Implementation Details

In Appendix C.1, we describe some implementation details of baselines and their variants, which can be different from official implementations. From Appendix C.2 to C.5, we describe implementation details for the components of Natural-HNN.

C.1 Baselines and their variants

We implemented HyperGAT based on the paper as its official implementation is different from what is explained in the paper. Moreover, as the original version of SHINE and HyperGAT do not involve multihead attention, we implement it for fair comparisons. For SHINE, we also implemented two versions, one without using $\mathcal{L}_{reg}$ and the other with $\mathcal{L}_{reg}$ which is a loss introduced by the paper for the purpose of making node representations to be similar if the nodes are included in the same hyperedge. However, we did not use the version with $\mathcal{L}_{reg}$ in cancer subtype classification task since the loss converts a hypergraph to a graph using clique expansion, which causes tremendous computational cost.

C.2 Factor Discrimination Loss

We defined a factor discrimination loss $\mathcal{L}_{dis}$ similar to the one used in [85]. In order to promote factors to contain different information, we use a factor classifier implemented with one layer MLP. Each factor representation of every hyperedge will be given as input to the factor classifier. The classifier needs to identify to which factor the factor representation belongs. If the classifier can correctly identify the factor with factor representation, i.e. if factor representations of two different factors of a hyperedge are distinguishable, it is highly likely that factors contain different information.

Specifically, we can calculate the loss by creating pseudo labels. For each factor representation of each hyperedge ($h^k_{e_i}$), we assign a pseudo label $Y^k_{e_i} = k$. Then the loss can be defined as follows:

$$\mathcal{L}_{dis} = -\sum_{e_i \in \mathcal{E}} \sum_{k=1}^{K} \sum_{c=1}^{K} \mathbf{1}(Y^k_{e_i} = c) log(softmax(MLP(h^k_{e_i}))) \tag{3}$$

This loss is applied to each layer of Natural-HNN. As described in Section 4.4, the final loss would be $\mathcal{L} = \mathcal{L}_{task} + \lambda \mathcal{L}_{dis}$. As mentioned before, $\mathcal{L}_{dis}$ is an optional part of our model. The hyperparameter search space for λ is provided in Appendix C.5

C.3 Loss used for training $\mathcal{L}_{task}$

After the final message passing layer of Natural-HNN, we get the final node embeddings z_{v_i}. The classifier of Natural-HNN will predict labels $p_{v_i} \in \mathbb{R}^C$ where C denotes the number of classes. In other words, $p_{v_i,c}$ denotes the probability that node v_i has class c as answer. If we denote l_{v_i} as the label (one-hot vector) for node v_i, the task loss can be calculated with cross-entropy loss.

$$\mathcal{L}_{task} = -\sum_{i=1}^{|V|} \sum_{c=1}^{C} l_{v_i,c} \log(p_{v_i,c}) \tag{4}$$

Note that, we use hyperedge embedding of the final layer instead of node embeddings for cancer subtype classification task.

C.4 Factor Encoder

In Section 4, we explained that we use K number of MLPs to get K factor representations. The resulting factor representation is a vector with size d/K when desired output representation size of a layer is given as d. When implementing the factor encoder as a code, we use single MLP that outputs vector with size d. Note that applying K different MLPs (with output vector size d/K) is the same as applying one MLP (with output vector size d) and chunking the vector to smaller ones with size d/K. (i.e. First d/K values corresponds to the 1^{st} factor representation, and following d/K values

Table 4: Hyperparameter search space in standard benchmark dataset. † : MLP layers used in AllDeepSets, AllSetTransforer, ED-HNN, ED-HNNII

models	♯ cl	classifier dim	head (factor)	♯ MLP layer †	λ for $\mathcal{L}_{dis}$	♯ Total
HGNN	1	-	1	-	-	32
HCHA	1	-	1	-	-	32
HNHN	1	-	1	-	-	32
UniGCNII	1	-	1	-	-	32
AllDeepSets	1,2	64,128,256,512	1	1,2	-	320
AllSetTransformer	1,2	64,128,256,512	1,2,4,8	1,2	-	1280
HyperGAT	1	-	1,2,4,8	-	-	128
SHINE	1	-	1,2,4,8	-	-	128
HSDN	1	-	1,2,4,8	-	0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1	896
ED-HNN	1,2	64,128,256,512	1	[0,1,2] × [1,2] × [0,1,2]	-	2880
ED-HNNII	1,2	64,128,256,512	1	[0,1,2] × [1,2] × [0,1,2]	-	2880
Natural-HNN	1	-	2,4,8	1	-	96
Natural-HNN+$\mathcal{L}_{dis}$	1	-	2,4,8	1	0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1	672

Table 5: Hyperparameter search space in cancer subtype classification task. † : MLP layers used in AllDeepSets, AllSetTransforer, ED-HNN, ED-HNNII

models	head (factor)	♯ MLP layer †	λ for $\mathcal{L}_{dis}$	♯ Total
HGNN	1	-	-	24
HCHA	1	-	-	24
HNHN	1	-	-	24
UniGCNII	1	-	-	24
AllDeepSets	1	1,2	-	48
AllSetTransformer	1,2,4,8	1,2	-	192
HyperGAT	1,2,4,8	-	-	96
SHINE	1,2,4,8	-	-	96
HSDN	1,2,4,8	-	0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1	672
ED-HNN	1	[0,1] × [1] × [0,1]	-	96
ED-HNNII	1	[0,1] × [1] × [0,1]	-	96
Natural-HNN	2,4,8	-	-	72

corresponds to the 2^{nd} factor representation and so on.) The nonlinear activation function we used for factor encoder is hyperbolic tangent (tanh).

C.5 Hyperparameter search space

We report the hyperparameter search space of each model in standard benchmark dataset as well as cancer subtype classification task. We used Adam optimizer for Natural-HNN. For the baselines, we closely followed optimizers or schedulers they used in their paper. Table 4 and Table 5 shows the hyperparameter search space in the standard benchmark dataset and cancer subtype datasets respectively. '♯ Total' denotes the number of all possible hyperparameter combinations that each model needs to search. 'cl' denotes the number of classifier layers. When the number of classifiers is larger than 1, those models have an additional hyperparameter that decides the hidden dimension of the classifier. ♯ MLP layer denotes the number of layers in MLP that was used in AllDeepSets, AllSetTransformer, ED-HNN, ED-HNNII. In the case of ED-HNN and ED-HNNII, there were three types of MLPs and each MLP could have different number of layers. λ for $\mathcal{L}_{dis}$ is hyperparameter that changes the reflection ratio of the factor discrimination loss.

For standard hypergraph benchmark datasets, we used [64, 128, 256, 512] as hidden dimension and [0.1, 0.01, 0.001, 0.0001] as learing rate. For weight decay, we used [0, 1e-5]. We fixed the number of layers to 2, except for HSDN, because HSDN uses only a single layer. Generally, we used 0.5 as dropout. (If the paper of a model specified dropout to a specific value, we used the value following the paper.) As we can see, our model generally has a small hyperparameter search space comparable to GAT (when not using $\mathcal{L}_{dis}$). Although ED-HNN and ED-HNNII had good performance on standard hypergraph benchmark datasets, they had to rely on very large hyperparameter search space.

For cancer subtype classification tasks, we used [16, 32, 64] as the hidden dimension and [0.1, 0.01, 0.001, 0.0001] as learning rate. For weight decay, we used [0, 1e-5]. We fixed the number of layers to 2, except for HSDN, because HSDN uses only a single layer. During training, we set 50 as the batch size. Generally, we used 0.5 as dropout. (If the paper of a model specified dropout to a specific value, we used the value following the paper.) Since we fixed the number of classifiers to 1, the hyperparameter search space of some models are largely reduced when compared to the node

https://doi.org/10.52202/085713-1232

classification task. For ED-HNN and ED-HNNII, we reduced the search space of the number of MLPs since it took too much time to get the results.

C.6 Environment for experiment

We used 48GB NVIDIA RTX A6000 GPU. We created a anaconda environment with python 3.7.16, pytorch 1.11.0 and pytorch geometric with version 2.0.4. Details can also be found at https://github.com/Yoonho-Lee-AI4Science/Natural-HNN.

Table 6: Dataset statistics of standard hypergraph benchmark dataset

	Cora	Citeseer	Pubmed	Cora-CA	DBLP-CA	NTU2012	ModelNet40	20Newsgroups
# nodes	2708	3312	19717	2708	41302	2012	12311	16242
# edge	1579	1079	7963	1072	22363	2012	12311	16242
# feature	1433	3703	500	1433	1425	100	100	100
# classes	7	6	3	7	6	67	40	4
avg. |e|	3.03	3.200	4.349	4.277	4.452	5	5	654.51
CE Homophily	0.897	0.893	0.952	0.803	0.869	0.753	0.853	0.461

Table 7: Model performance on standard hypergraph benchmark datasets (Accuracy). The last row is the result with extreme hyperparameter search space that includes hyperparmeter searching for dropout and interpolation ratio β (introduced in Section 4.3). Top three models (excluding the last row) are colored by **First**, **Second**, Third. † : the variant of the model using multihead attention. ⋆ : the variant of the model using $\mathcal{L}_{reg}$ defined in SHINE[48].

Method	Cora	Citeseer	Pubmed	Cora-CA	DBLP-CA	NTU2012	ModelNet40	20Newsgroups
HGNN	79.453 ± 1.003	73.092 ± 1.582	87.336 ± 0.443	83.383 ± 1.028	91.410 ± 0.365	88.350 ± 1.082	95.567 ± 0.411	81.246 ± 0.435
HCHA	79.276 ± 1.158	**73.693** ± 1.687	87.230 ± 0.511	83.191 ± 0.868	91.358 ± 0.374	88.270 ± 1.304	94.703 ± 0.283	81.189 ± 0.397
HNHN	76.765 ± 1.560	72.524 ± 1.570	87.237 ± 0.523	77.480 ± 0.932	86.927 ± 0.346	88.489 ± 0.878	97.811 ± 0.231	81.059 ± 0.485
UniGCNII	79.498 ± 1.508	73.514 ± 2.107	88.124 ± 0.376	83.840 ± 0.693	**91.728** ± 0.225	89.245 ± 0.882	97.243 ± 0.334	81.687 ± 0.452
AllDeepSets	79.306 ± 1.627	72.959 ± 1.795	**89.418** ± 0.360	84.594 ± 0.793	91.594 ± 0.308	88.847 ± 0.984	97.532 ± 0.185	81.721 ± 0.653
AllSetTransformer	79.749 ± 1.620	73.140 ± 1.804	**88.667** ± 0.388	84.786 ± 0.690	91.593 ± 0.309	89.404 ± 1.074	**98.217** ± 0.138	**81.783** ± 0.569
HyperGAT	55.908 ± 4.128	41.751 ± 1.814	48.191 ± 0.443	73.560 ± 1.829	90.292 ± 0.468	83.857 ± 1.490	92.465 ± 0.387	80.997 ± 0.390
HyperGAT†	58.183 ± 2.079	42.246 ± 1.874	48.389 ± 0.426	73.752 ± 1.508	90.394 ± 0.362	85.467 ± 1.876	92.481 ± 0.463	81.083 ± 0.374
SHINE	57.755 ± 3.198	41.413 ± 0.680	48.576 ± 0.455	75.037 ± 1.912	90.759 ± 0.292	87.256 ± 1.393	93.803 ± 0.395	81.061 ± 0.632
SHINE†	56.307 ± 4.452	41.763 ± 0.693	48.576 ± 0.433	75.613 ± 1.508	90.697 ± 0.329	87.157 ± 1.426	93.878 ± 0.332	81.239 ± 0.459
SHINE⋆	58.818 ± 1.591	41.413 ± 1.563	46.682 ± 1.177	74.623 ± 1.444	61.507 ± 12.169	81.451 ± 2.399	89.406 ± 0.775	61.492 ± 12.666
SHINE†⋆	58.065 ± 1.616	41.123 ± 1.707	43.619 ± 1.402	73.087 ± 1.077	36.215 ± 17.676	70.835 ± 23.388	75.956 ± 23.688	56.452 ± 13.043
HSDN	76.632 ± 1.509	71.824 ± 1.779	87.193 ± 0.323	81.595 ± 1.011	90.229 ± 0.242	**89.722** ± 1.196	83.439 ± 1.204	81.372 ± 0.435
ED-HNN	**80.635** ± 1.670	**73.696** ± 1.992	**88.911** ± 0.410	**85.480** ± 0.828	**92.151** ± 0.291	87.594 ± 0.811	97.999 ± 0.199	81.608 ± 0.695
ED-HNNII	78.951 ± 1.445	72.524 ± 1.682	79.355 ± 0.953	83.693 ± 0.839	**91.702** ± 0.325	86.223 ± 0.958	95.749 ± 0.335	80.150 ± 0.753
Natural-HNN (ours)	**80.709** ± 1.635	73.285 ± 1.742	87.136 ± 0.450	**84.993** ± 0.491	90.961 ± 0.137	**89.900** ± 1.017	**98.558** ± 0.295	**81.734** ± 0.745
Natural-HNN (ours + $\mathcal{L}_{dis}$)	**80.739** ± 1.570	**73.551** ± 1.964	88.475 ± 0.466	**85.081** ± 0.583	91.032 ± 0.179	**90.060** ± 1.565	**98.584** ± 0.254	**81.827** ± 0.695
Natural-HNN (ours, extreme)	81.300 ± 1.323	74.058 ± 1.335	88.746 ± 0.511	85.583 ± 0.774	91.910 ± 0.192	90.417 ± 0.919	98.629 ± 0.229	82.083 ± 0.742

D Standard Hypergraph Benchmark dataset

We performed experiments with standard hypergraph benchmark dataset to check whether Natural-HNN can be applied to the **datasets that are not verified to have multiple factors behind group interactions. Considering how hyperedges were created for benchmark datasets, it is not likely that those datasets contain meaningful or task related interaction contexts.** In co-citation and co-authorship networks, for example, hyperedges are created by simply connecting all documents cited by a paper or written by an author. Citations between a pair of papers might have context that is related to a reason for citation, however, it is hard to expect that a group of documents (papers) cited by a paper creates a special meaning or have a special context. Even if we assume that hyperedges in co-citation networks contain interaction context, it is still not clear how these interaction contexts are related to the labels of nodes. It is also hard to expect interaction context in co-authorship networks for a similar reason. Thus, **the benchmark dataset experiment will verify whether Natural-HNN can be applied to the datasets where the existence of factors behind group interactions is not known.**

For the node classification task with standard hypergraph benchmark datasets, we randomly split the data into 50%/25%/25% for training/validation/test set. We measured average and standard deviation of the performances for 10 different data splits. The hyperparameter search space is provided in Appendix C.5.

D.1 Statistics : Standard Hypergraph Benchmark Dataset

Cocitaion networks and coauthor networks are adopted from [76]. The node features are bag-of-words representation of each documents. NTU2012 and ModelNet40 dataset is computer vision and graphics datasets where features are generated by applying GVCNN[19] and MVCNN[66]. Node feature of 20Newsgroups are generated by TF-IDF representations of news. The statistics of standard benchmark dataset is given in Table 6. Homophily ratio was calculated after converting hypergraph into a graph with clique expansion (CE)[67] following the method described in the other work [70].

https://doi.org/10.52202/085713-1232

Table 8: Model performance on standard hypergraph benchmark datasets (Accuracy) trained with only 5% of data

Method	Cora	Citeseer	Pubmed	Cora-CA	DBLP-CA	NTU2012	ModelNet40	20Newsgroups
HGNN	66.773 ± 2.806	61.445 ± 2.465	81.161 ± 0.531	71.548 ± 2.652	89.689 ± 0.384	58.884 ± 5.045	94.795 ± 0.381	79.690 ± 0.675
HCHA	67.403 ± 2.865	61.600 ± 2.279	81.135 ± 0.549	71.379 ± 2.465	89.689 ± 0.274	59.032 ± 5.083	93.939 ± 0.448	79.596 ± 0.652
HNHN	58.272 ± 1.970	58.473 ± 5.296	79.793 ± 0.804	58.831 ± 2.399	82.855 ± 0.499	58.737 ± 5.344	96.845 ± 0.382	78.456 ± 0.602
UniGCNII	68.212 ± 2.559	63.600 ± 1.203	83.024 ± 0.820	70.799 ± 2.606	88.751 ± 0.281	60.255 ± 5.022	96.584 ± 0.248	79.061 ± 0.506
AllDeepSets	65.694 ± 2.306	61.388 ± 4.012	84.485 ± 0.647	71.319 ± 2.964	59.689 ± 0.296	59.892 ± 4.833	96.055 ± 0.286	78.868 ± 0.534
AllSetTransformer	65.914 ± 2.155	62.506 ± 1.720	82.942 ± 0.491	71.249 ± 2.796	89.665 ± 0.216	60.444 ± 5.204	96.608 ± 0.291	79.409 ± 0.590
HSDN	58.332 ± 2.882	57.812 ± 1.808	80.195 ± 0.45	64.845 ± 4.025	87.636 ± 0.243	51.949 ± 17.016	97.159 ± 0.179	79.406 ± 0.594
ED-HNN	66.433 ± 2.824	61.759 ± 2.296	82.348 ± 0.559	69.809 ± 2.569	90.039 ± 0.342	57.984 ± 6.477	96.698 ± 0.265	78.386 ± 0.542
Natural-HNN (ours)	67.343 ± 1.837	62.620 ± 2.277	82.393 ± 0.467	70.809 ± 2.789	88.700 ± 0.251	60.511 ± 5.338	98.031 ± 0.196	79.329 ± 0.666
Natural-HNN (ours + $\mathcal{L}_{dis}$)	67.393 ± 1.938	62.694 ± 2.218	82.838 ± 0.609	70.909 ± 3.439	88.906 ± 0.204	61.384 ± 4.570	98.141 ± 0.116	79.431 ± 0.552

D.2 Node Classification on Benchmark Datasets

Table 7 summarizes the node classification performance in standard hypergraph benchmark datasets. We have the following observations: **1)** Our model generally performs well on various datasets by taking the first or second place in terms of accuracy. In the case of Citeseer and Cora-CA, the performance of our model is comparable to the best performing model. The results indicate that our model can be applied to various circumstances, even when the context variety of hyperedges is not guaranteed. **2)** Attention-based models (i.e., AllSetTransformer, SHINE, and HyperGAT) and disentangle-based model (i.e., HSDN) generally perform similar to or worse than convolution-based models (i.e., HGNN, HCHA, HNHN, UniGCNII) and AllDeepSets (which also does not have heads or factors) on Citeseer, Pubmed and DBLP-CA. Through the results, we can guess that those datasets do not contain various interaction contexts that is helpful for the model performance. This can also be a reason why our model does not perform well on those datasets as much as on other datasets.

We consider a model that achieves sufficiently good performance without relying excessively on hyperparameter tuning to be reliable. However, there has been an increasing number of papers, such as Sheaf Hypergraph Networks [18], PhenomNN [72], and ED-HNN [70], that report performance obtained through an extreme level of hyperparameter tuning. Therefore, to enable a fair comparison with these works, we also included dropout and the interpolation ratio β (introduced in Section 4.3) in the hyperparameter tuning and conducted additional experiments. For both dropout and the interpolation ratio β, we set the hyperparameter search space from 0.1 to 0.9 with an interval of 0.1. The results are reported in the last row of Table 7. Comparing the results of Natural-HNN with the official performance results of the papers mentioned earlier that rely on extreme hyperparameter tuning, we can see that Natural-HNN outperforms them despite having a much simpler model architecture.

D.3 Training with only 5% of data

To check the generalization power of our model, we performed an experiment of training with only 5% of data. Following the split ratio of HGNN for Cora dataset, we trained with 5% of data, validated with 18.5% and tested with 37% of data. Table 8 shows the result. We have the following observations: **1)** The performance of Natural-HNN tends to be similar or slightly better than convolution-based models. This shows that Natural-HNN has good generalization power that is comparable to convolution-based methods. **2)** Our model performs better than recently introduced model, ED-HNN. Even if ED-HNN has much larger hyperparameter search space, Natural-HNN performs better due to generalization power.

E Ablation studies and Hyperparameter sensitivity

E.1 Selecting Alternative Branch

In Section 4, we used the representation earned from 'Disentangle-first Branch' ($h^k_{e_i}$) when creating final hyperedge factor representations ($\alpha^k_i h^k_{e_i}$). The experiment results below shows the result when using the other branch, 'Aggregation-first Branch' for creating final hyperedge factor representations ($\alpha^k_i \tilde{h}^k_{e_i}$). Table 9 shows the result for standard hypergraph benchmark dataset and Table 10 shows the result for cancer subtype classification task.

Table 9: Comparison of our model (first two rows) with alternative model that uses the other type of hyperedge factor representation (last two rows)

Method	Cora	Citeseer	Pubmed	Cora-CA	DBLP-CA	NTU2012	ModelNet40	20Newsgroups
Natural-HNN	80.709 ± 1.635	73.285 ± 1.742	87.163 ± 0.450	84.993 ± 0.491	90.961 ± 0.137	89.900 ± 1.017	98.558 ± 0.295	81.734 ± 0.745
Natural-HNN (+$\mathcal{L}_{dis}$)	80.739 ± 1.570	73.551 ± 1.964	88.475 ± 0.466	85.081 ± 0.583	91.032 ± 0.179	90.060 ± 1.565	98.584 ± 0.254	81.827 ± 0.695
Natural-HNN (other branch)	80.650 ± 1.684	73.237 ± 1.678	87.137 ± 0.408	84.993 ± 0.434	90.968 ± 0.137	89.821 ± 0.847	98.557 ± 0.232	81.729 ± 0.701
Natural-HNN (other branch + $\mathcal{L}_{dis}$)	80.827 ± 1.157	73.575 ± 1.790	88.521 ± 0.424	85.081 ± 0.503	91.030 ± 0.178	90.060 ± 0.795	98.577 ± 0.227	81.837 ± 0.534

As we can see in Table 9, there is no big difference in the performance between using 'Disentangle-first Branch' and 'Aggregation-first Branch'.

Table 10: Comparison of our model (first row) with alternative model that uses the other type of hyperedge factor representation (last row).

Method	BRCA	STAD	SARC	LGG	HNSC	CESC
Natural-HNN	0.804 ± 0.036	0.659 ± 0.049	0.745 ± 0.045	0.707 ± 0.035	0.860 ± 0.042	0.881 ± 0.042
Natural-HNN (other branch)	0.797 ± 0.028	0.654 ± 0.041	0.747 ± 0.063	0.707 ± 0.033	0.863 ± 0.022	0.875 ± 0.051

As we can see in Table 10, there is no big difference in the performance between using 'Disentangle-first Branch' and 'Aggregation-first Branch'. The reason for this phenomenon is quite simple. We can consider the two cases: 1) when $h^k_{e_i}$ and $\tilde{h}^k_{e_i}$ are similar and 2) when they are largely different. **1)** When $h^k_{e_i}$ and $\tilde{h}^k_{e_i}$ are similar, the result will not differ a lot between using $h^k_{e_i}$ or $\tilde{h}^k_{e_i}$ as similar representations will be used. **2)** When $h^k_{e_i}$ and $\tilde{h}^k_{e_i}$ are largely different, the result will not be different a lot since relevance score α^k_i will be very small. In other words, $\alpha^k_i h^k_{e_i} - \alpha^k_i \tilde{h}^k_{e_i} = \alpha^k_i (h^k_{e_i} - \tilde{h}^k_{e_i})$ will be very small for very small α^k_i. This case means that the factor representation will not be reflected a lot during message passing since the representation is inconsistent (different result for two branches).

E.2 Natural-HNN without naturality constraint

We performed another ablation study to check whether naturality condition proposed in the paper is important part that contributes to the model. We created an ablation model that do not satisfies naturality condition by not reflecting relevance score α^k_i during message passing. The results for standard hypergraph benchmark dataset is provided in Table 11. The results for the cancer subtype classification task are provided in Table 12.

Table 11: Model performance on standard hypergraph benchmark datasets (Accuracy). The ablation model does not satisfy the naturality condition.

Method	Cora	Citeseer	Pubmed	Cora-CA	DBLP-CA	NTU2012	ModelNet40	20Newsgroups
Natural-HNN (ours)	80.709 ± 1.635	73.285 ± 1.742	87.136 ± 0.450	84.993 ± 0.491	90.961 ± 0.137	89.900 ± 1.017	98.558 ± 0.295	81.734 ± 0.745
Natural-HNN (ours + $\mathcal{L}_{dis}$)	80.739 ± 1.570	73.551 ± 1.964	88.475 ± 0.466	85.081 ± 0.583	91.032 ± 0.179	90.060 ± 1.565	98.584 ± 0.254	81.827 ± 0.695
Natural-HNN (ablation)	80.220 ± 1.573	73.237 ± 1.745	87.121 ± 0.170	84.874 ± 0.424	90.896 ± 0.165	89.281 ± 0.718	98.144 ± 0.226	81.685 ± 0.675
Natural-HNN (ablation + $\mathcal{L}_{dis}$)	80.250 ± 1.555	73.392 ± 1.832	88.448 ± 0.407	85.022 ± 0.508	90.968 ± 0.169	89.679 ± 1.129	98.177 ± 0.216	81.783 ± 0.771

In Table 11, we can see that there is a slight to moderate level of performance gap between Natural-HNN and its ablation model. It is not a surprising result that there is not big difference between them since standard benchmark datasets do not seem to have informative interaction contexts related to the task (Appendix D).

In Table 12, we can observe that there is a big difference between Natural-HNN and its ablation model. Since interaction context matters in cancer subtype classification task, naturality condition seems to boost the performance by capturing interaction context.

https://doi.org/10.52202/085713-1232

Table 12: Model performance on cancer subtype classification task (Macro F1). The ablation model does not satisfy the naturality condition.

Method	BRCA	STAD	SARC	LGG	HNSC	CESC
Natural-HNN* (ours)	0.804 ± 0.036	0.659 ± 0.049	0.745 ± 0.045	0.707 ± 0.035	0.862 ± 0.045	0.881 ± 0.042
Natural-HNN* (ablation)	0.756 ± 0.031	0.605 ± 0.039	0.713 ± 0.071	0.692 ± 0.034	0.814 ± 0.037	0.852 ± 0.032

E.3 Hyperparameter Analysis

Since Natural-HNN does not have many hyperparameters, we analyzed how performance changes by the number of factors. Table 13 shows the result for the standard hypergraph benchmark dataset. Table 14 shows the result for cancer subtype classification task. Note that the tables below show the result of Natural-HNN without $\mathcal{L}_{dis}$.

Table 13: Performance of Natural-HNN with a different number of factors. The best performances (reported in Table 7) are marked in red.

number of factors	Cora	Citeseer	Pubmed	Cora-CA	DBLP-CA	NTU2012	ModelNet40	20Newsgroups
1	80.384 ± 1.820	73.133 ± 1.767	87.063 ± 0.373	84.934 ± 0.418	90.951 ± 0.139	89.622 ± 0.953	98.480 ± 0.310	81.684 ± 0.725
2	80.532 ± 1.638	73.285 ± 1.742	87.055 ± 0.401	84.904 ± 0.432	90.961 ± 0.137	89.622 ± 0.759	98.513 ± 0.272	81.734 ± 0.745
4	80.709 ± 1.652	73.188 ± 1.967	87.083 ± 0.450	84.993 ± 0.491	90.939 ± 0.151	89.821 ± 1.070	98.558 ± 0.295	81.635 ± 0.716
8	80.591 ± 1.673	73.237 ± 1.783	87.136 ± 0.450	84.934 ± 0.385	90.955 ± 0.131	89.900 ± 1.017	98.513 ± 0.286	81.660 ± 0.714

We have interesting observations when we analyze the result in Table 7 with Table 13. **1)** In Table 7, we observe that Natural-HNN does not perform well on the Citeseer, Pubmed, and DBLP-CA datasets. Except for Pubmed, Table 13 shows that Natural-HNN used two or fewer factors on these datasets.

2) Natural-HNN demonstrated good performance on the remaining five datasets in Table 7. Except for the 20Newsgroups dataset, Natural-HNN used four or more factors to achieve its best performance, as shown in Table 13. These observations suggest that Natural-HNN generally performs well when capturing multiple factors. Furthermore, since the model did not benefit from using more than two factors on Citeseer and DBLP-CA, we suspect that these datasets lack diverse interaction contexts that would enhance performance. A similar trend is observed for other attention-based (AllSetTransformer) and disentanglement-based (HSDN) models in Table 7. Although these models are capable of capturing relational information, they showed poor performance—sometimes even worse than some convolution-based models.

Table 14: Performance of Natural-HNN with different number of factors. The best performance (reported in Table 1) are marked in red.

number of factors	BRCA	STAD	SARC	LGG	HNSC	CESC
1	0.789 ± 0.036	0.630 ± 0.046	0.729 ± 0.055	0.695 ± 0.030	0.853 ± 0.047	0.869 ± 0.048
2	0.787 ± 0.038	0.642 ± 0.043	0.745 ± 0.045	0.707 ± 0.035	0.858 ± 0.031	0.867 ± 0.043
4	0.804 ± 0.036	0.659 ± 0.049	0.725 ± 0.048	0.689 ± 0.047	0.858 ± 0.036	0.881 ± 0.042
8	0.785 ± 0.027	0.637 ± 0.032	0.729 ± 0.058	0.691 ± 0.044	0.860 ± 0.042	0.878 ± 0.034

We have similar observations when comparing the result in Table 1 and Table 14. **1)** For the SARC and LGG datasets in Table 14, Natural-HNN achieved its best performance when using two factors. **2)** For the remaining datasets, Natural-HNN achieved its best performance with four or more factors. Except for CESC, these cases showed a meaningful increase in performance. Therefore, we can draw a similar conclusion to the one derived from the comparison of Table 7 and Table 13.

F Additional Experiment Result

F.1 Computational Complexity

Let d_i be the input embedding dimension, d_o be the output embedding dimension, K be number of factors. N denotes number of nodes and M denotes number of hyperedges, E denotes the number of node(v)-hyperedge(e) pair (v, e) satisfying $v \in e$. We will assume that $d_i \geqslant d_o$, $d_o \geqslant K$, $E \geqslant M$ and $E \geqslant N$.

The computational complexity of one layer of Natural-HNN can be calculated by the following:

- Aggregation-first Branch (aggregation + MLP): $O(Ed_i) + O(Md_id_o)$
- Disentangle-first Branch (MLP + aggregation): $O(Nd_id_o) + O(Ed_o)$
- Similarity (α) calculation : $O(K(\frac{d_o^2}{K^2} + \frac{d_o}{K})) = O(\frac{d_o^2}{K})$
- propagation back to nodes : $O(KE + Ed_o) = O(Ed_o)$
- other calculations (concat, interpolation by β) : $O(Nd_o)$ Thus, total computational complexity becomes $O((M+N)d_id_o + E(d_i + d_o + 1) + Nd_o + \frac{d_o^2}{K}) = O((M+N)d_id_o + E(d_i + d_o))$

For HGNN with dimension $d_i \geqslant d_e \geqslant d_o$ (d_e denotes dimension of hyperedge embedding), computational complexity becomes $O(E(d_i + d_e) + (Md_i + Nd_o)d_e)$. The computational complexity of HGNN and Natural-HNN differs only by constant times. It is not surprising since Natural-HNN is quite similar to HGNN but instead use two branches (only) during Node-to-Hyperedge propagation and use factor similarity calculation. Thus, Natural-HNN is as scalable as HGNN.

F.2 Scalability Analysis (training time)

We measured the time it takes for the model to train for 10 epochs. We averaged the values after measuring 5 times each. Also, we conducted the experiment in two settings: one with 2 heads and 16-dimensional vector as hidden representation and the other with 8 heads and 64-dimensional vector as hidden representation. Note that convolution-based models, AllDeepSets and ED-HNN (II) use 1 head as they do not have a multi-head attention mechanism. The table 15 shows the result of our model's scalability. We have the following observations: **1)** Our model is slower than convolution-based models and HSDN. Since convolution-based models use strong inductive bias with simple computations, they are naturally scalable than our model. HSDN took less time since they use only one message passing layer. **2)** Our model is much faster than all attention-based models. Thus, we can conclude that our model scales well with hypergraph and parameter size next to the convolution-based models.

Table 15: Time took for training 10 epochs for BRCA. We tested with two cases by differing hidden dimension size and number of heads†: multihead attention version

(dimension , head)	(16, 2)	(64, 8))
HGNN	2.171 ± 0.003	8.492 ± 0.010
HCHA	2.130 ± 0.003	8.322 ± 0.011
HNHN	1.169 ± 0.005	4.362 ± 0.005
UniGCNII	2.384 ± 0.004	9.166 ± 0.009
AllDeepSets	7.870 ± 0.026	18.679 ± 0.040
AllSetTransformer	11.213 ± 0.030	27.004 ± 0.024
HyperGAT†	7.191 ± 0.024	24.579 ± 0.047
SHINE†	9.099 ± 1.419	22.253 ± 0.162
HSDN	2.944 ± 0.003	10.130 ± 0.006
ED-HNN	11.937 ± 0.026	22.738 ± 0.026
ED-HNNII	21.621 ± 0.029	36.418 ± 0.026
Natural-HNN (ours)	5.479 ± 0.006	18.924 ± 0.070

F.3 Generalization power of Natural-HNN

To check the generalization power of our model, we experimented with different training set split ratio, while maintaining the validation and test set ratio to 25%. From 50%, we gradually reduced training set proportion to 10% as shown in Figure 17. Figure 17(a) and (b) are the result of measuring performance with accuracy or Macro-F1 scores and (c) and (d) are the result of measuring relative degradation of performance to the performance when trained with 50% For example, for BRCA dataset experiment, which is measured with Macro-F1 score, the relative degradation of performance is caculated by $(F_{50} - F_x)/F_{50} \times 100\%$ where F_x denotes the Macro-F1 score when trained with x%. The same applies to Cora-CA, which is measured with accuracy. Figure 17 (a) and (c) are the result in Cora-CA dataset, which is standard hypergraph benchmark, (b) and (d) are the result

https://doi.org/10.52202/085713-1232

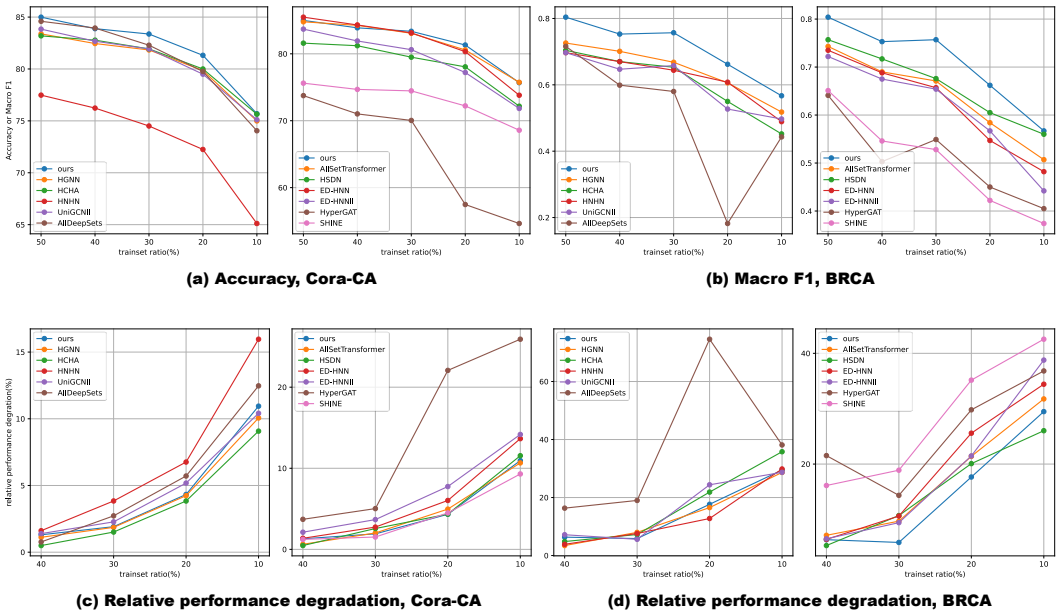


Figure 17: The performance of models when reducing training set proportion. First row shows Macro F1 score and the second row shows relative performance degradation compared to the performance when using 50% of dataset as training set. Natural-HNN (ours, colored in blue) maintains best Macro F1 score and small relative performance degradation on both Cora-CA and BRCA dataset.

for BRCA dataset, which is dataset used for cancer subtype classification task. The left figure in each Figure 17 (a,b,c,d) is the result of comparing ours (blue) and convolution of deepset based models. These baselines cannot perform context-dependent message passing. The right figure in each Figure 17 (a,b,c,d) is the result of comparing ours (blue) and other baselines that have potential for context-dependent message passing

We have the following observations : **1)** The degradation of performance for Natural-HNN was smaller when compared with most of the baselines in both Cora-CA and BRCA. Specifically, we can see that Natural-HNN has comparable result with convoluation based models in left figures of Figure 17 (c) and (d). Considering that convolutions based models have strong generalization performance due to their strong inductive bias, we can say that our model has good generalization power comparable to convolution based models. When compared with other baselinese in Figure 17 (b) and (d), we can observe that Natural-HNN had very small degradation in performance. In other words, Natural-HNN had nearly the smallest degradation when compared with models that have more expressive power than convolution based methods. We can consider our model had good generalization among baselines with more expressive powers. Specifically, in Figure 17 (d), Natural-HNN showed outstanding result in cancer dataset which has various context of interactions. This might be due to the fact that the inductive bias (context of interaction) that Natural-HNN used matched the actual data characteristics.

2) Natural-HNN had the best Macro-F1 score for all different training ratio. Our model always had the best performance compared to convolution or deepset based models in left figures of Figure 17 (a) and (b). Specifically, we can see that Natural-HNN had outstanding performance in BRCA cancer dataset in the left figure of Figure 17 (b). Thus, we can conclude that Natural-HNN is more expressive compared to convolution based models. Also, when inductive bias (interaction context) matches the data characteristics (BRCA), Natural-HNN provides outstanding performances. From the result, we could verify that Natural-HNN can utilize context information to get good performance. When compared with other baselines, in the right figures of Figure 17 (a) and (b), we can see that our model could achieve better, or at least comparable performance when compared with baselines. We can conclude that our model has expressive power comparable to other attention (including Set Transformer) or equivariance based models. Again, we can observe that Natural-HNN achieved outstanding performance in BRCA dataset by capturing context types. Considering that Natural-HNN had good generalization and expressivity, we argue that our model made a proper trade-off between expressive power and generalization.

F.4 Captured Context in CESC

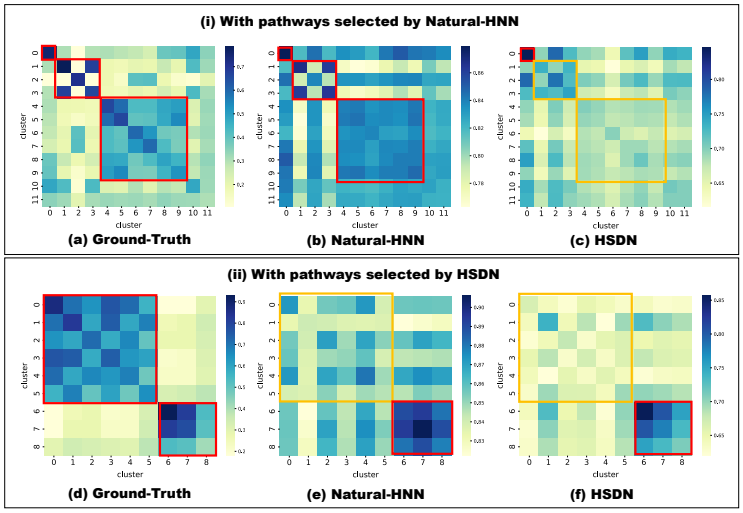


Figure 18: Captured interaction context. Pathways are selected by SHAP value. Captured patterns are shown in red box and not captured patterns are shown with orange box. Weakly captured case is marked as dotted red block.

Figure 18 shows the captured context result in CESC. The evaluation and interpretation method is identical to that of Section 5.3. As we can see in the figure, for pathways selected by Natural-HNN, Natural-HNN correctly captures context similarities between clusters (red box) while HSDN does not (orange box). For the pathways selected by HSDN, Natural-HNN and HSDN partially captures cluster similarity. However, when comparing orange box in (d) and (f), we can observe that Natural-HNN captures interaction context slightly better than HSDN even with the pathways selected by HSDN.

F.5 Cancer Subtype Classification (Micro F1)

We briefly provide Micro F1 scores of each model in cancer subtype classification task. The Table 16 also shows that our model generally performs well on most of cancer datasets.

Table 16: Micro F1 score of each model with parameter and hyperparameter of the best Macro F1 score. Top two models are colored by **First**, **Second**. †: the variant of the model using multihead attention. ⋆ : we did not use $\mathcal{L}_{dis}$.

Method	BRCA	STAD	SARC	LGG	HNSC	CESC
HGNN	0.817 ± 0.027	0.727 ± 0.026	0.739 ± 0.057	0.696 ± 0.034	0.888 ± 0.031	0.903 ± 0.034
HCHA	0.808 ± 0.024	0.725 ± 0.036	0.731 ± 0.058	0.685 ± 0.039	0.876 ± 0.034	0.911 ± 0.034
HNHN	0.806 ± 0.027	0.729 ± 0.067	0.733 ± 0.046	0.676 ± 0.037	0.884 ± 0.018	0.910 ± 0.033
UniGCNII	0.791 ± 0.027	0.797 ± 0.038	**0.761** ± 0.046	0.665 ± 0.038	0.910 ± 0.013	0.911 ± 0.018
AllDeepSets	0.823 ± 0.025	0.748 ± 0.039	0.657 ± 0.035	0.669 ± 0.045	0.895 ± 0.025	0.927 ± 0.024
AllSetTransformer	0.827 ± 0.031	0.710 ± 0.047	0.749 ± 0.047	0.656 ± 0.037	0.898 ± 0.016	0.908 ± 0.025
HyperGAT	0.754 ± 0.116	0.725 ± 0.050	0.645 ± 0.106	0.669 ± 0.051	0.889 ± 0.030	0.900 ± 0.025
HyperGAT†	0.753 ± 0.072	0.676 ± 0.108	0.643 ± 0.098	0.665 ± 0.042	0.883 ± 0.053	0.896 ± 0.021
SHINE	0.659 ± 0.090	0.590 ± 0.127	0.618 ± 0.106	0.649 ± 0.058	0.846 ± 0.032	0.890 ± 0.044
SHINE†	0.783 ± 0.027	0.711 ± 0.061	0.709 ± 0.045	0.654 ± 0.044	0.873 ± 0.027	0.907 ± 0.031
HSDN	**0.838** ± 0.022	**0.801** ± 0.033	0.758 ± 0.047	0.694 ± 0.036	0.892 ± 0.025	0.925 ± 0.024
ED-HNN	0.826 ± 0.024	0.793 ± 0.047	**0.761** ± 0.039	**0.703** ± 0.028	0.913 ± 0.021	0.925 ± 0.035
ED-HNNII	0.815 ± 0.027	0.748 ± 0.024	0.694 ± 0.050	0.696 ± 0.038	**0.916** ± 0.013	**0.942** ± 0.024
Natural-HNN⋆ (ours)	**0.869** ± 0.024	**0.824** ± 0.027	**0.770** ± 0.040	**0.709** ± 0.033	**0.923** ± 0.020	**0.932** ± 0.024

https://doi.org/10.52202/085713-1232

Table 17: Hyperedge classification result (accuracy). Top two models are colored by **First**, **Second**.

Dataset	HGNN	HCHA	HNHN	UniGCNII	AllDeepSets	AllSetTransformer	HSDN	Natural-HNN (ours)
Chemical Reaction	0.449 ± 0.005	0.482 ± 0.010	0.257 ± 0.008	0.672 ± 0.004	0.493 ± 0.023	**0.727** ± 0.026	0.491 ± 0.023	**0.773** ± 0.008

F.6 Chemical Reaction Classification (Hyperedge Classification)

To validate whether Natural-HNN performs well not only on cancer subtype classification but also on other hypergraph datasets that contains meaningful hyperedge semantics, we performed hyperedge classification task on a chemical reaction dataset [21]. Among the three datasets proposed in that paper, we used the first dataset for validation, as the other two datasets have relatively small number of samples and the prediction tasks are too easy to serve as a meaningful evaluation of the model. The hyperparameter search space was kept the same as that used for the standard hypergraph benchmark datasets, and Natural-HNN was evaluated without $\mathcal{L}_{dis}$. As shown in Table 17, Natural-HNN demonstrates overwhelmingly superior performance compared to other models, including HSDN. Therefore, Natural-HNN proves to be highly effective not only for cancer subtype classification but also for datasets in which hyperedges contain hidden semantics related to labels.

F.7 Reliability of Natural-HNN in Biology

In order for a model to be reliable, the model should provide consistent output regardless of the choice of hyperparameters. So we conducted an experiment to check whether models consistently rely on the same pathways. If a model consistently rely on the same pathways for prediction regardless of the hyperparameter, biologists might consider the model to be reliable since it potentially captured and used what can be explained with biological domain knowledge. On the other hand, if the model relies on different pathways for different hyperparameters, biologists might not trust the model.

To check whether model relies on the same pathways, we ranked the pathways with SHAP value and selected top-k pathways. These pathways are the ones that models relied most for their prediction. Then, we calculated Jaccard similarity of top-k pathways for different hyperparameters. If top-k pathways earned from each hyperparameter combination is similar, then we can conclude that model always rely on the same pathways regardless of the hyperparameters.

Figure 19 and Figure 20 are the result of calculating Jaccard similarity between different hyperparameter combinations on BRCA dataset. The hyperparameters we changed was the hidden dimension size and the number of factors. Values in each tick of row and column is the pair of the two hyperparameters (i.e., the value in the ticks represent (hidden dimesion, number of factors) pair). Each heatmap shows Jaccard similarity when selecting top 10, 15, 20, 50, 100 and 500 pathways. Figure 19 is the results for Natural-HNN and Figure 20 is the result for HSDN. We also calculated average Jaccard similarity for each heatmap.

The ideal result would show dark blue colors (high similarity) to all cells in the heatmap. It means that top-k pathways that a model relied on are always the same regardless of the hyperparameter. When comparing Figures 19 and 20, we can see that Natural-HNN tends to rely on the same pathway regardless of the hyperparameter while HSDN does not. When comparing average Jaccard similarity scores, we can quantitatively observe that Natural-HNN has better consistency when compared to HSDN. For example, Jaccard similarity with top 15 pathways of Natural-HNN (19 (b)) has average similarity of 0.759 while that of HSDN (20 (b)) has average similarity of 0.555.

From this experiment, we can conclude that Natural-HNN is reliable since it consistently focuses on the same pathways regardless of the choice of hyperparameters. Also, we could again verify that our model captures the functionality of pathways (interaction context of hyperedge) and expect that our model will work reliably in different dataset or different biological applications. Note that similar analysis for Figure 21 and Figure 22 provides similar conclusion.

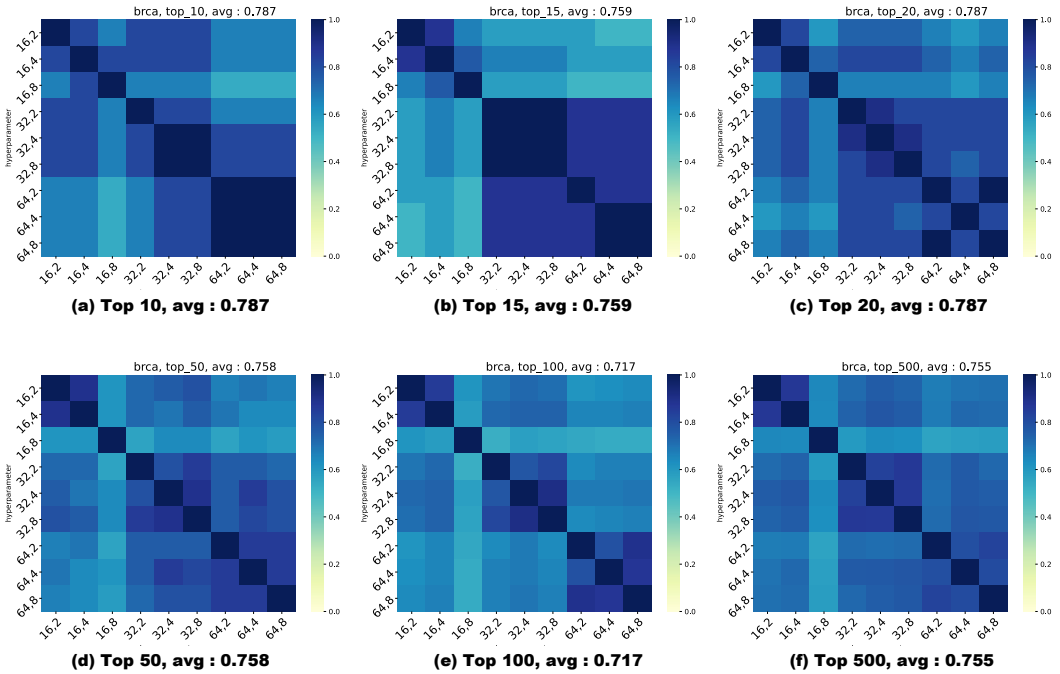


Figure 19: Jaccard similarity calculation result for Natural-HNN on BRCA. We can observe that Natural-HNN generally relies on similar pathways regardless of hyperparameters by showing high Jaccard similarity value.

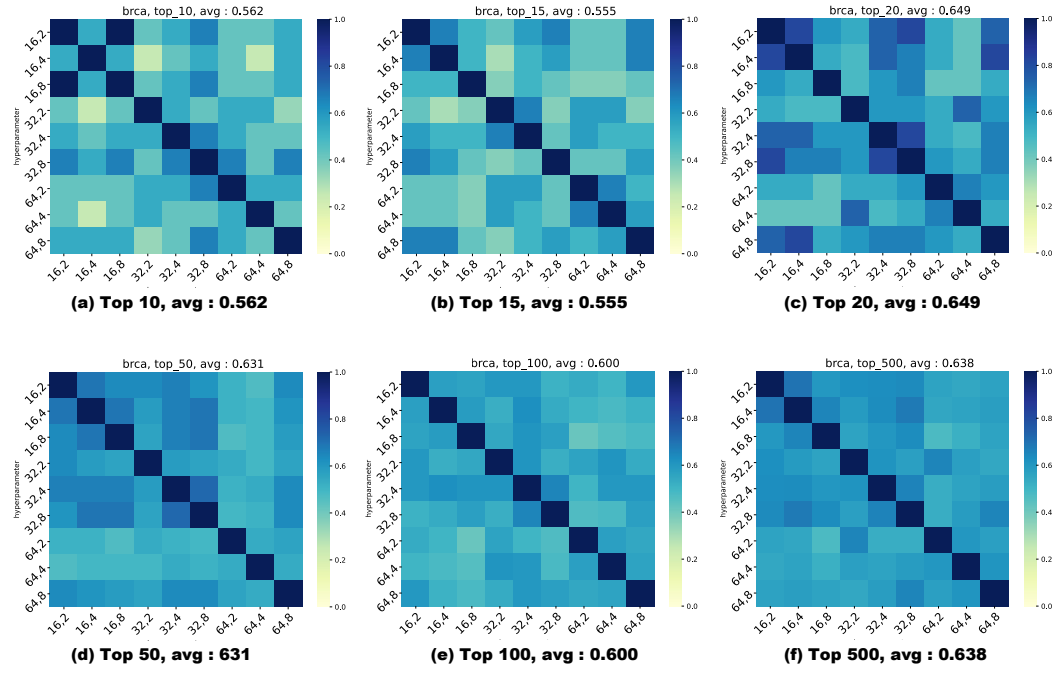


Figure 20: Jaccard similarity calculation result for HSDN on BRCA. We can observe that HSDN relies on different pathways for different hyperparameters by showing strong diagonal pattern. This inconsistency makes HSDN an unreliable model for biology.

https://doi.org/10.52202/085713-1232

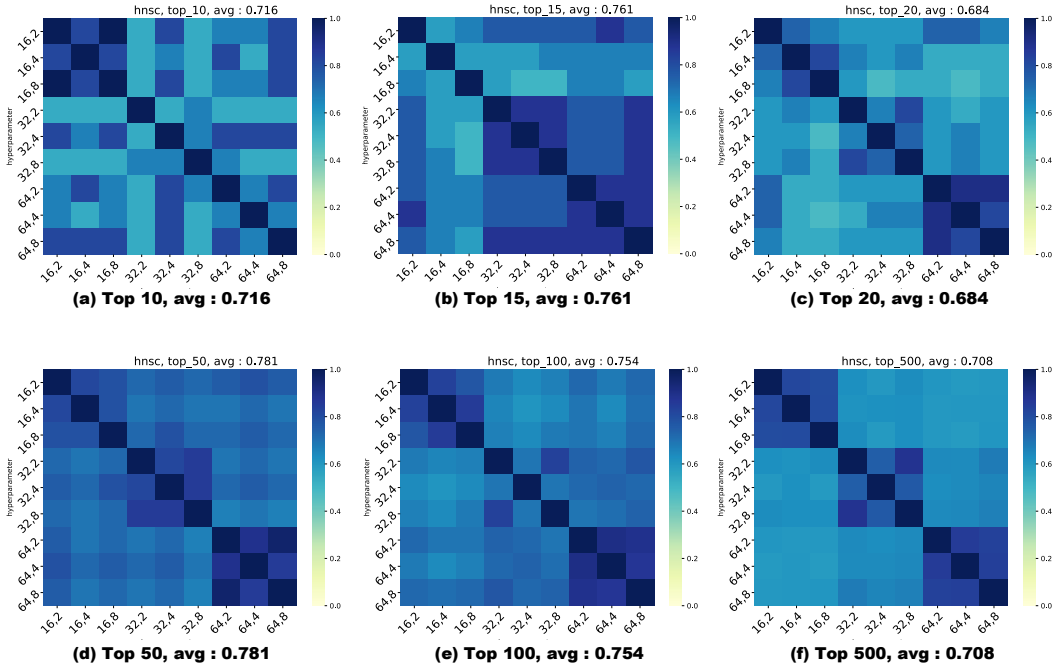


Figure 21: Jaccard similarity calculation result for Natural-HNN on HNSC. We can observe that Natural-HNN generally relies on similar pathways regardless of hyperparameters by showing high Jaccard similarity value.

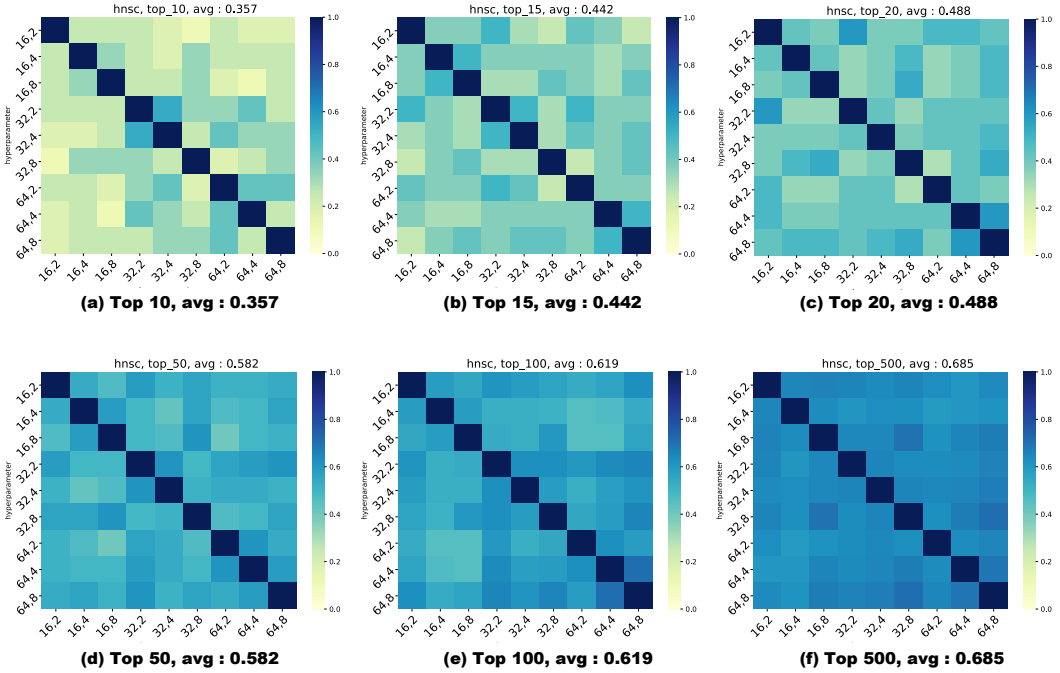


Figure 22: Jaccard similarity calculation result for HSDN on HNSC. We can observe that HSDN relies on different pathways for different hyperparameters by showing strong diagonal pattern. This inconsistency makes HSDN an unreliable model for biology.

G Limitations, impacts and Future Work

G.1 Broader Impacts

Potential Positive Societal Impacts. As demonstrated through various experiments, Natural-HNN has the potential to capture the inherent heterogeneity of interactions and diverse interaction contexts. In complex systems such as biological organisms, many interactions have unknown functionalities. Natural-HNN's ability to capture these latent interaction contexts can contribute to the development of more reliable models for a wide range of real-world problems.

Potential Negative Societal Impacts. Our proposed method is designed to automatically identify and incorporate the factors underlying interactions. The relevance scores indicate which factors are most relevant to each interaction. However, if this method is applied to data where privacy is critical, it could potentially lead to indirect leakage of sensitive information through those relevance scores.

G.2 Limitation

Natural-HNN uses hyperparameter K to decide number of factors instead of automatically discovering the number of factors within data. In real world problems, it might require a lot of time to get optimal number of factors. This is a kind of a problem that all disentangle-based methods need to solve in the future.

G.3 Future Work 1 : Model for Graph Neural Network

Since Natural-HNN is designed for hypergraph neural network, we can apply our model to graphs. However, it is computationally inefficient since Natural-HNN performs two step message passing (node-to-hyperedge, hyperedge-to-node) while most of the gnns perform one step message passing. Thus, we need to devise a novel criterion for disentangling edge types in graphs without using edge representations. Since there are many interaction types in graphs, developing reliable edge disentangling model in the perspective of category theory will be useful for many real world applications.

G.4 Future Work 2 : Hyperedge-Node co-disentanglement

Our goal was to disentangle the factors behind group interactions, and thus we assumed that the nodes participating in an interaction share the same context (factor). However, it is also possible that individual nodes have their own distinct contexts (factors) when participating in an interaction. For example, consider a group discussion involving multiple individuals. In Natural-HNN, the disentanglement focused on hyperedge-level factors, such as the discussion topic. However, node-level disentanglement could also be applied in this scenario. Each participant might have a specific role in the discussion. Separately from the discussion topic, factors such as the context or role of each participant in the discussion could also be disentangled. Performing a hyperedge-node co-disentanglement, which is disentangling both hyperedge-level and node-level factors, would allow for a more nuanced approximation of diverse underlying mechanisms.

https://doi.org/10.52202/085713-1232