CAPTURING FUNCTIONAL CONTEXT OF GENETIC PATHWAYS THROUGH HYPEREDGE DISENTANGLEMENT

Yoonho Lee, Junseok Lee, Sangwoo Seo, Sungwon Kim, Yeongmin Kim, Chanyoung Park*
KAIST
Daejeon, 34141, Republic of Korea

ABSTRACT

The hypergraph data structure has been used to represent the multiway interactions of a set of genes of a genetic pathway. Since genes within each genetic pathway collaboratively perform a biological function, the functional context of a pathway (i.e., the interaction context of a hyperedge), which is often unannotated, needs to be captured. However, most existing hypergraph neural networks fail to reflect the interaction context of each hyperedge due to their limited ability to capture important or relevant factors. In this paper, we propose a simple yet effective hyperedge disentangling method, **Natural-HNN**, which captures the interaction context of a hyperedge. We introduce a novel guidance mechanism for hyperedge disentanglement based on the naturality condition in category theory. In our experiments, we applied our model to hypergraphs of genetic pathways for the cancer subtype classification task and demonstrated that our model outperforms baseline approaches by capturing the functional semantic similarity of genetic pathways.

1 INTRODUCTION

A genetic pathway is a set of genes that collaborate to perform a specific function in a biological process. As gene interactions within each pathway contribute to biological functions, there have been several attempts to leverage genetic pathways for predicting labels that are associated with biological functions. For example, genetic pathways were used to predict cancer (Liu et al., 2024) or other diseases (Sharma & Xu, 2023) that are often the result of dysregulation of pathways with specific biological function. To reflect multiway interactions among a set of genes, several hypergraph neural networks (HNNs) (Luo, 2022; Tang et al., 2024), in which each hyperedge connects multiple nodes, have been proposed.

Since genes within a genetic pathway interact to perform biological functions, HNNs need to reflect the functional context in which the genetic interaction occurs. However, such functional contexts of pathways are often unannotated (Liu & Thomas, 2019), making heterogeneous hypergraph models inapplicable. Most homogeneous hypergraph models, on the other hand, cannot perform context-dependent message passing, as nodes generate the same message to their neighboring hyperedges. Thus, capturing unannotated interaction contexts (i.e., functional contexts) is necessary.

To this end, we propose a novel Naturality-guided disentangled Hypergraph Neural Network (**Natural-HNN**) that can inherently reflect the interaction context of an hyperedge. We approach the task with the category theoretical perspective (Fong & Spivak, 2018), and determine the criterion for disentangling factors as the factor representation consistency based on the naturality condition that must be satisfied between entangled and disentangled representations. Figure 1 shows the naturality condition applied to our genetic pathway example. Let's suppose that genes in a pathway interacts under the context 2 and does not interact under context 1. The result of interaction under context 2 must be consistent, regardless of whether interaction was performed only on context 2 (i.e., factor specific message passing, Figure 1 $(ii) \rightarrow (iii) \rightarrow (vi)$) or the interaction was performed for both contexts but only context 2 related result was selected (i.e., factor information extraction after entangled message passing, Figure 1 $(ii) \rightarrow (v) \rightarrow (vi)$). On the other hand, this commutativity does not hold for context 1 (i.e., the result of $(ii) \rightarrow (i) \rightarrow (iv)$ and $(ii) \rightarrow (v) \rightarrow (iv)$ is different) as the pathway is not related to context 1. The adoption of consistency constraint derived from category theory allows us to capture context related factors without relying on any assumption on the data.

*denotes the corresponding author.

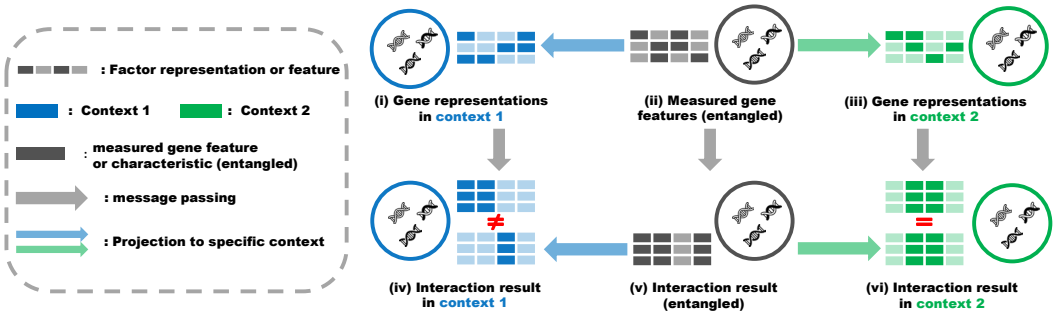


Figure 1: Naturality condition (commutativity) guides interaction context disentanglement.

Our main contributions are summarized as follows:

- To the best of our knowledge, we are the first to propose a hyperedge disentanglement-based method that is systematically designed to capture the context related to the background or condition of multiway interaction .
- We proposed a novel way to guide the hyperedge disentanglement, by focusing on the compositional structure of entities in hypergraph message passing framework. Through a new criterion derived from the category theory, we created a simple but effective model, showing outstanding performance even with a small hyperparameter search space.
- We applied our model to the cancer subtype classification task, and showed our model can actually capture functional semantics of pathways.

2 Related work

Hypergraph Neural Network. Several HNN models have been recently proposed to leverage information contained in multiway interaction. HGNN (Feng et al., 2019) and HCHA (Bai et al., 2021) use a normalized hypergraph Laplacian, which is mathematically equivalent to clique expansion (CE) (Sun et al., 2008), and apply the traditional graph convolution mechanism. HNHN (Dong et al., 2020) additionally adopts nonlinearity when calculating hyperedge representations to differentiate a hypergraph from a clique expanded graph, while UniGNN (Huang & Yang, 2021) unifies HNNs and GNNs into the same framework. Moreover, HyperGAT (Ding et al., 2020) adopts the attention mechanism to HNN for text classification, and SHINE (Luo, 2022) proposes dual attention mechanism for the disease classification task. ED-HNN (Wang et al., 2022) proposes equivariant message passing HNN, which allows hyperedges to propagate different messages to its incident nodes. AllDeepSets and AllSetTransformer (Chien et al., 2021) consider a hyperedge as a set and apply DeepSets (Zaheer et al., 2017) and Set Transformer (Lee et al., 2018), respectively, to increase expressive power of HNN. All of theses methods, however, cannot give different weights to different heads or factors, limiting their capability of capturing the interaction context of an hyperedge, which is crucial in practice.

Disentangled Representation Learning. Disentangled representation learning (DRL) (Roth et al., 2022; Fumero et al., 2021; Higgins et al., 2018) aims to disentangle the factor of variation of observed data. The effectiveness of DRL has garnered attention of researchers, leading to its expansion into the field of GNN. DisenGCN (Ma et al., 2019) disentangles the factor of variations in nodes to find the factor behind connections, while FactorGCN (Yang et al., 2020) disentangles graphs into several factor graphs. DisGNN (Zhao et al., 2022) recently proposes to disentangle edge types with the self-supervision from label conformity.

Since graph-based disentangling methods cannot model multiway interactions, DRL is also being applied to hypergraphs. HSDN (Hu et al., 2022) attempts to capture structural semantics by disentangling a hypergraph into several factor hypergraphs. Although this method is advantageous when capturing the functional structure in molecules or finding communities in a social network, it is not suitable for capturing the interaction context as this approach captures semantics derived from different connectivity or substructure. DisenHCN (Li et al., 2022) disentangles user embeddings for recommender systems, but is only applicable to hypergraphs with known hyperedge types.

3 CATEGORICAL INTERPRETATION OF MESSAGE PASSING HNN AND DISENTANGLEMENT

Prior to the discussion of the naturality condition for hyperedge disentanglement, it is essential to analyze the compositional structure in the hypergraph representation learning. In Section 3.1, we describe the compositional structure of hypergraph message passing neural networks. In Section 3.2, we propose the naturality condition as a guidance for hyperedge disentanglement. The basic concepts in category theory we used are described in Appendix D, and the basic explanation of disentangled representation learning is described in Appendix E.1.

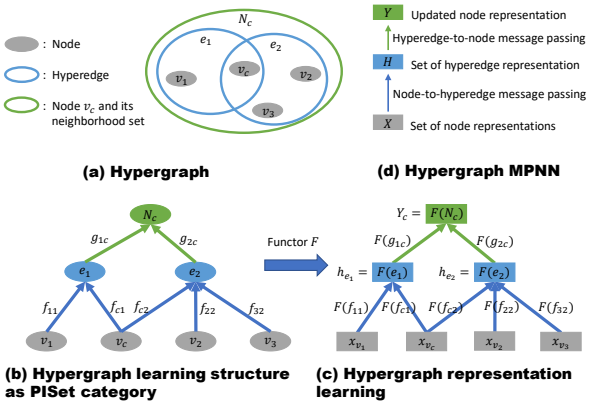


Figure 2: Compositional structure in hypergraph representation learning.

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 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

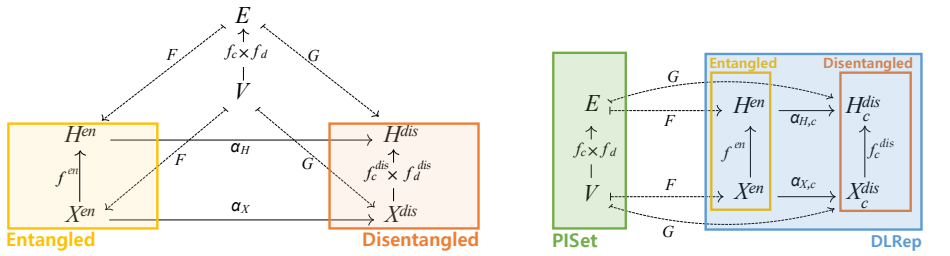
Most hypergraph representation learning methods produce the representation of a node by integrating its own representation and its neighbors' representations defined by a hypergraph topology. As an example, in Figure 2 (a), the representation of a center node v_c is updated to the representation that can express the meaning produced by a set of nodes N_c, the set whose elements are the node v_c and its one-hop neighbors (v_1, v_2, v_3). During the process, the hypergraph topology created by hyperedges are considered.

In this paper, for the first time, we describe the above process of hypergraph representation learning through the lens of the 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 message passing neural networks[2] can be seen as a way of learning representations and their transformations respecting compositional structure of entities.

[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 (Fong & Spivak, 2018).

[2]The message passing types are not only limited to traditional convolution-based or attention-based methods, but also can include complex methods such as general message passing (Papillon et al., 2023).



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

Figure 3: Naturality condition in disentangled representation learning to capture context related factors. X denotes a set of node representations and H denotes hyperedge representation. V and E denote node and hyperedge in **PISet**, respectively.

The most expressive way for a model to accommodate various morphisms would be to assign different learnable parameters to every morphism, which, however, would likely fail in generalizability and scalability perspectives. In this case, providing proper inductive bias is the key to balancing the trade-off between expressive power and generalizability of the model. However, convolution-based methods have a strong assumption that all neighbors can be considered equally regardless of the interaction context of an hyperedge, limiting expressive power of the model. On the other hand, disentangled representation learning can be used as an adequate trade-off by categorizing morphisms into a small number of morphism types, which can be considered as context-dependent message passing. Therefore, we propose a hyperedge disentangling method for context-dependent message passing, which will be introduced in Section 4.

3.2 Guiding Disentanglement with Naturality Condition

Since entangled representations 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 entangled representations and disentangled representations. Figure 3 (b) is equivalent to Figure 3 (a), but only the components related to the factor 'c' are shown. Note that $\alpha_{X,c} = \alpha_X \circ p_c$ where $p_c : X^{dis} \rightarrow X_c^{dis}$ (refer to Appendix E.3). 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 $f^{en} \circ \alpha_{H,c}$ (i.e., message passing on entangled representation first, and then disentangling factors) or $\alpha_{X,c} \circ f_c^{dis}$ (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 an hyperedge[3]. 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. More precise and detailed explanations are provided in Appendix E.3

4 Proposed Method: Natural-HNN

Each layer of Natural-HNN is composed of a message passing lane (left column of Figure 4 (c)), and a non-message passing lane (right column of Figure 4 (c)) as well as their integration with layer normalization (Section 4.3, bottom of Figure 4 (c)). The key component of our model is the message passing lane (Figure 4 (b)) that consists of a Node-to-Hyperedge factor propagation module (Section 4.1), and a Hyperedge-to-Node factor propagation module (Section 4.2). 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

[3]The group discussion example in Figure 1 shows this property.

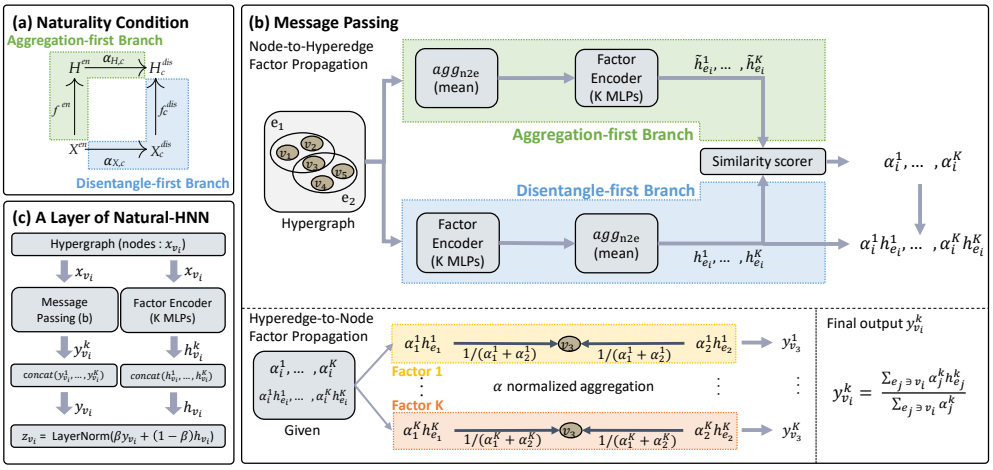


Figure 4: An overview of Natural-HNN. (a) illustrates the naturality condition shown in Figure 3 (b). (b) shows the message passing block of Natural-HNN that consists of a Node-to-Hyperedge and Hyperedge-to-Node factor propagation modules. The Final output of the message passing block is shown at the right bottom corner of (b). (c) shows the composition of each layer of Natural-HNN.

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 $f^{en} \circ \alpha_{H,k}$ 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 $\alpha_{X,k} \circ f_k^{dis}$ 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)). The 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:

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

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[4], i.e., $\alpha_i^k h_{e_i}^k$, that reflects the relevance of the factor k for the hyperedge e_i.

[4]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 C.1.

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.

Method	BRCA	STAD	SARC	LGG	HNSC	CESC	KIPAN	NSCLC
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	0.921 ± 0.016	0.959 ± 0.016
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	0.920 ± 0.015	0.954 ± 0.009
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	0.920 ± 0.021	0.958 ± 0.016
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	**0.935** ± 0.012	0.949 ± 0.017
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	0.912 ± 0.015	0.953 ± 0.010
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	0.925 ± 0.013	0.953 ± 0.014
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	0.899 ± 0.037	0.927 ± 0.020
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	0.896 ± 0.025	0.939 ± 0.009
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	0.837 ± 0.197	0.866 ± 0.128
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	0.925 ± 0.017	0.954 ± 0.013
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	**0.937** ± 0.005	**0.961** ± 0.013
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	0.931 ± 0.013	0.955 ± 0.012
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	0.930 ± 0.015	0.953 ± 0.012
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	**0.934** ± 0.010	**0.962** ± 0.013

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

We allowed our model to determine its focus between information from neighbors (i.e., y_{v_i}) and information of the node itself (i.e., x_{v_i}) by introducing hyperparameter β that decides interpolation ratio between them. 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, $z_{v_i} = \text{LayerNorm}(\beta y_{v_i} + (1-\beta) h_{v_i})$, where $y_{v_i} = \text{Concat}(y_{v_i}^1, \ldots, y_{v_i}^K)$, $h_{v_i} = \text{Concat}(h_{v_i}^1, \ldots, h_{v_i}^K)$. Note that to reduce the burden of hyperparameter tuning, we fix $\beta = 0.5$.

5 Experiments

5.1 Experimental Setup

Dataset. For the cancer subtype classification task, we downloaded clinical data for 8 cancer types (BRCA, STAD, SARC, LGG, CESC, HNSC, KIPAN and NSCLC) and preprocessed data following Pathformer (Liu et al., 2023) (Details in Appendix A.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 representations) are different. The data statistic of each cancer data is provided in Appendix A.1.

Compared Methods. We compared Natural-HNN with HNNs introduced in Section 2. Specifically, HGNN(Feng et al., 2019), HCHA (Bai et al., 2021), HNHN (Dong et al., 2020), UniGCNII (Huang & Yang, 2021), AllDeepSets (Chien et al., 2021), AllSetTransformer (Chien et al., 2021), HyperGAT (Ding et al., 2020), SHINE (Luo, 2022), ED-HNN (Wang et al., 2022), ED-HNNII (Wang et al., 2022) and a hypergraph disentangling method HSDN (Hu et al., 2022) are used as baselines.

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 B.2.

5.2 Results for Cancer Subtype Classification

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 (Liu et al., 2023), 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, we achieve large performance improvements compared with the convolution-based methods as well as AllDeepSets, which cannot leverage the interaction contexts. In the case of BRCA, we achieve about 5% performance improvement compared with the second best model.

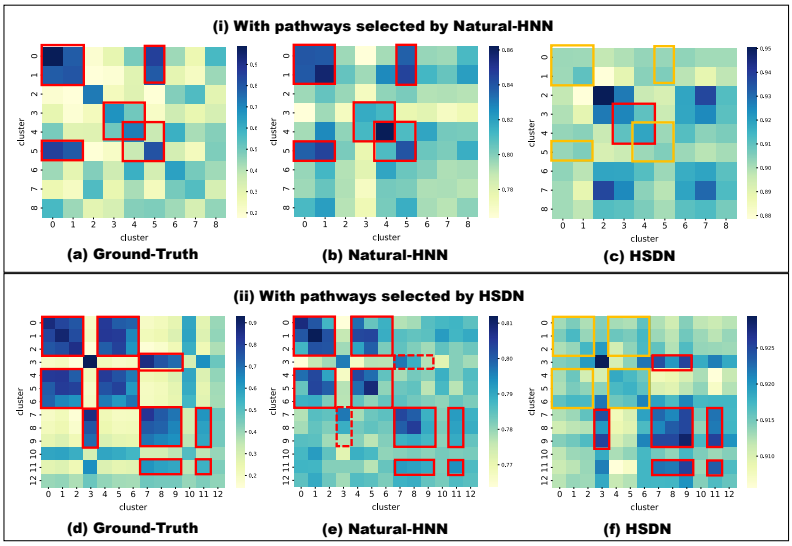


Figure 5: Captured interaction context. Captured patterns are shown in red boxes and not captured patterns are shown with orange boxes. Weakly captured cases are marked as dotted red block.

This result can be attributed to the following two facts: *First*, pathways contain **"context-dependent interaction"**[5] that reflect various functional semantics (Stoney et al., 2018; 2015). *Second*, cancers are directly related to the functions of multiple pathways (Windels et al., 2022; Stoney et al., 2018). Thus, we can conclude that reflecting various functional context of pathways is important in cancer related tasks and our model benefited by effectively capturing such interaction contexts. **2)** Natural-HNN does not show impressive performance on KIPAN and NSCLC compared to other datasets. This is due to the fact that those cancers are relatively easy to be classified with only the gene features (Wang et al., 2021; Oh et al., 2021). **3)** Natural-HNN outperforms the disentangle-based model, HSDN, with a large performance gap. Although HSDN mainly aimed to capture the structural semantics, it is similar to ours in that it can potentially capture interaction types by giving different factor importance for each hyperedge. They also used similarity-based criterion for disentanglement by comparing similarity between factor representations of a hyperedge and nodes. However, the superior performance of Natural-HNN validates that the naturality-guided disentanglement can better integrate contextual information of interaction.

5.3 Natural-HNN captures functional context of pathways

To validate that Natural-HNN can capture the interaction context, we checked whether our model captures functional semantics of genetic pathways. Because the models rely solely on cancer subtype labels during training[6], 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[7] 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 (Kramer et al., 2014), 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 (Lin et al., 1998) (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 A.3.

[5] **A direct quote from (Stoney et al., 2018)**

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

[7] Only a few pathways are related to each type of cancer. We can also observe this with the SHAP value distribution in Figure 7

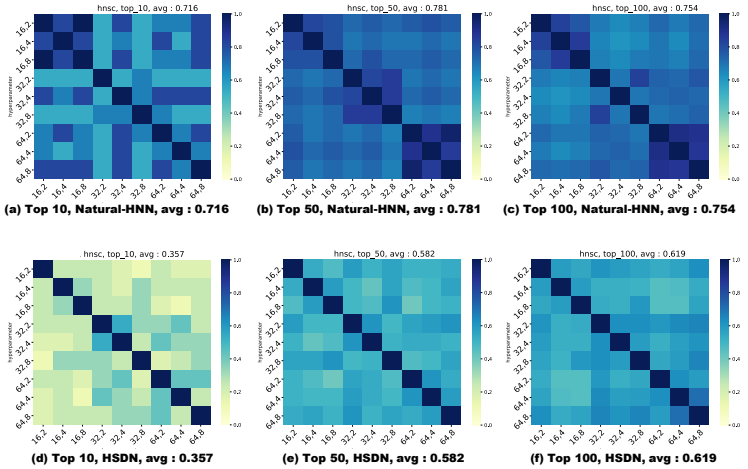


Figure 6: Jaccard similarity of (SHAP) top-k pathways across different hyperparameters.

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 an hyperedge. Additional analyses are described in Appendix E.5

5.4 Natural-HNN shows consistency regardless of the hyperparameter

Since cancers are directly related to the functions and dysregulation of multiple pathways (Sharma & Xu, 2023; Windels et al., 2022; Stoney et al., 2018), models should rely on specific pathways for cancer subtype classification regardless of the choice of hyperparameters. To check whether models rely on the same pathways, we ranked the pathways with SHAP value and selected top-k pathways. These pathways are the ones that models relied on 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 the model rely on the same pathways regardless of the hyperparameters.

Figure 6 is the result of calculating Jaccard similarity between different hyperparameter combinations on HNSC 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. Heatmap (a), (b) and (c) corresponds to the result of Natural-HNN for top 10, 50 and 100 pathways respectively. Heatmap (d), (e) and (f) corresponds to the result of HSDN for top 10, 50 and 100 pathways respectively. We can observe that Natural-HNN tends to rely on the same pathway (i.e. high Jaccard similarity) regardless of the hyperparameter while HSDN does not. This consistency makes Natural-HNN more reliable.

6 Conclusion

In this work, we propose Natural-HNN, which captures the interaction context of nodes within a hyperedge during the message passing process. We analyzed compositional structure in hypergraph message passing through the lens of category theory and focused on the naturality condition that must be satisfied between entangled and disentangled representations. Through several experiments with cancer subtype classification dataset, we validated that our novel hyperedge disentangle-based model successfully captures functional contexts of genetic pathways without the help of external knowledge or a complex objective function.

REFERENCES

Suzi A Aleksander, James Balhoff, Seth Carbon, J Michael Cherry, Harold J Drabkin, Dustin Ebert, Marc Feuermann, Pascale Gaudet, Nomi L Harris, et al. The gene ontology knowledgebase in 2023. *Genetics*, 224(1):iyad031, 2023.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

Song Bai, Feihu Zhang, and Philip HS Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110:107637, 2021.

Pietro Barbiero, Stefano Fioravanti, Francesco Giannini, Alberto Tonda, Pietro Lio, and Elena Di Lavore. Categorical foundations of explainable ai: A unifying formalism of structures and semantics. *arXiv preprint arXiv:2304.14094*, 2023.

Mattia G Bergomi and Pietro Vertechi. Neural network layers as parametric spans. *arXiv preprint arXiv:2208.00809*, 2022.

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. *arXiv preprint arXiv:2106.13264*, 2021.

Antonio Colaprico, Tiago C Silva, Catharina Olsen, Luciano Garofano, Claudia Cava, Davide Garolini, Thais S Sabedot, Tathiane M Malta, Stefano M Pagnotta, Isabella Castiglioni, et al. Tcgabiolinks: an r/bioconductor package for integrative analysis of tcga data. *Nucleic acids research*, 44(8):e71–e71, 2016.

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

Geoffrey SH Cruttwell, Bruno Gavranović, Neil Ghani, Paul Wilson, and Fabio Zanasi. Categorical foundations of gradient-based learning. In *European Symposium on Programming*, pp. 1–28. Springer International Publishing Cham, 2022.

Pim de Haan, Taco S Cohen, and Max Welling. Natural graph networks. *Advances in neural information processing systems*, 33:3636–3646, 2020.

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

Yihe Dong, Will Sawin, and Yoshua Bengio. Hnhn: Hypergraph networks with hyperedge neurons. *arXiv preprint arXiv:2006.12278*, 2020.

Andrew Dudzik, Tamara von Glehn, Razvan Pascanu, and Petar Veličković. Asynchronous algorithmic alignment with cocycles. *arXiv preprint arXiv:2306.15632*, 2023.

Andrew J Dudzik and Petar Veličković. Graph neural networks are dynamic programmers. *Advances in Neural Information Processing Systems*, 35:20635–20647, 2022.

Steffen Durinck, Yves Moreau, Arek Kasprzyk, Sean Davis, Bart De Moor, Alvis Brazma, and Wolfgang Huber. Biomart and bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics*, 21(16):3439–3440, 2005.

Steffen Durinck, Paul T Spellman, Ewan Birney, and Wolfgang Huber. Mapping identifiers for the integration of genomic datasets with the r/bioconductor package biomart. *Nature protocols*, 4(8): 1184–1191, 2009.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3558–3565, 2019.

Brendan Fong and Michael Johnson. Lenses and learners. *arXiv preprint arXiv:1903.03671*, 2019.

Brendan Fong and David I Spivak. Seven sketches in compositionality: An invitation to applied category theory. *arXiv preprint arXiv:1803.05316*, 2018.

Brendan Fong, David Spivak, and Rémy Tuyéras. Backprop as functor: A compositional perspective on supervised learning. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 1–13. IEEE, 2019.

Marco Fumero, Luca Cosmo, Simone Melzi, and Emanuele Rodolà. Learning disentangled representations via product manifold projection. In *International conference on machine learning*, pp. 3530–3540. PMLR, 2021.

Bruno Gavranović. Compositional deep learning. *arXiv preprint arXiv:1907.08292*, 2019.

Jakob Hansen and Thomas Gebhart. Sheaf neural networks. *arXiv preprint arXiv:2012.06333*, 2020.

Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3:315–358, 2019.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. arxiv. *arXiv preprint arXiv:1812.02230*, 2018.

Bingde Hu, Xingen Wang, Zunlei Feng, Jie Song, Ji Zhao, Mingli Song, and Xinyu Wang. Hsdn: A high-order structural semantic disentangled neural network. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Jing Huang and Jie Yang. Unignn: a unified framework for graph and hypergraph neural networks. *arXiv preprint arXiv:2105.00956*, 2021.

Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

Michael Kramer, Janusz Dutkowski, Michael Yu, Vineet Bafna, and Trey Ideker. Inferring gene ontologies from pairwise similarity data. *Bioinformatics*, 30(12):i34–i42, 2014.

Anton Kratz, Minkyu Kim, Marcus R Kelly, Fan Zheng, Christopher A Koczor, Jianfeng Li, Keiichiro Ono, Yue Qin, Christopher Churas, Jing Chen, et al. A multi-scale map of protein assemblies in the dna damage response. *Cell Systems*, 14(6):447–463, 2023.

Henry Kvinge, Brett Jefferson, Cliff Joslyn, and Emilie Purvine. Sheaves as a framework for understanding and interpreting model fit. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4222–4230, 2021.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer. 2018.

Tom Leinster. Basic category theory. *arXiv preprint arXiv:1612.09375*, 2016.

Martha Lewis. Compositionality for recursive neural networks. *arXiv preprint arXiv:1901.10723*, 2019.

Yinfeng Li, Chen Gao, Quanming Yao, Tong Li, Depeng Jin, and Yong Li. Disenhcn: Disentangled hypergraph convolutional networks for spatiotemporal activity prediction. *arXiv preprint arXiv:2208.06794*, 2022.

Dekang Lin et al. An information-theoretic definition of similarity. In *Icml*, volume 98, pp. 296–304, 1998.

Meng Liu and Paul D Thomas. Go functional similarity clustering depends on similarity measure, clustering method, and annotation completeness. *BMC bioinformatics*, 20(1):1–15, 2019.

Xiaofan Liu, Yuhuan Tao, Zilin Cai, Pengfei Bao, Hongli Ma, Kexing Li, Mengtao Li, Yunping Zhu, and Zhi John Lu. Pathformer: a biological pathway informed transformer integrating multi-omics data for disease diagnosis and prognosis. *bioRxiv*, pp. 2023–05, 2023.

Xiaofan Liu, Yuhuan Tao, Zilin Cai, Pengfei Bao, Hongli Ma, Kexing Li, Mengtao Li, Yunping Zhu, and Zhi John Lu. Pathformer: a biological pathway informed transformer for disease diagnosis and prognosis using multi-omics data. *Bioinformatics*, 40(5):btae316, 2024.

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

Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. Disentangled graph convolutional networks. In *International conference on machine learning*, pp. 4212–4221. PMLR, 2019.

Seyed MH Mansourbeigi. *Sheaf Theory as a Foundation for Heterogeneous Data Fusion*. PhD thesis, Utah State University, 2018.

Craig H Mermel, Steven E Schumacher, Barbara Hill, Matthew L Meyerson, Rameen Beroukhim, and Gad 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.

Mohamed Mounir, Marta Lucchetta, Tiago C Silva, Catharina Olsen, Gianluca Bontempi, Xi Chen, Houtan Noushmehr, Antonio Colaprico, and Elena 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.

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

Jung Hun Oh, Wookjin Choi, Euiseong Ko, Mingon Kang, Allen Tannenbaum, and Joseph O Deasy. Pathcnn: interpretable convolutional neural networks for survival prediction and pathway analysis applied to glioblastoma. *Bioinformatics*, 37(Supplement_1):i443–i450, 2021.

Mathilde Papillon, Sophia Sanborn, Mustafa Hajij, and Nina Miolane. Architectures of topological deep learning: A survey on topological neural networks. *arXiv preprint arXiv:2304.10031*, 2023.

Yue Qin, Casper F Winsnes, Edward L Huttlin, Fan Zheng, Wei Ouyang, Jisoo Park, Adriana Pitea, Jason F Kreisberg, Steven P Gygi, J Wade Harper, et al. Mapping cell structure across scales by fusing protein images and interactions. *bioRxiv*, pp. 2020–06, 2020.

Jüri Reimand, Ruth Isserlin, Veronique Voisin, Mike Kucera, Christian Tannus-Lopes, Asha Rostamianfar, Lina Wadi, Mona Meyer, Jeff Wong, Changjiang 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.

Karsten Roth, Mark Ibrahim, Zeynep Akata, Pascal Vincent, and Diane Bouchacourt. Disentanglement of correlated factors via hausdorff factorized support. *arXiv preprint arXiv:2210.07347*, 2022.

Francisco Sanchez-Vega, Marco Mina, Joshua Armenia, Walid K Chatila, Augustin Luna, Konnor C La, Sofia Dimitriadoy, David L Liu, Havish S Kantheti, Sadegh Saghafinia, et al. Oncogenic signaling pathways in the cancer genome atlas. *Cell*, 173(2):321–337, 2018.

Carl F Schaefer, Kira Anthony, Shiva Krupa, Jeffrey Buchoff, Matthew Day, Timo Hannay, and Kenneth H Buetow. Pid: the pathway interaction database. *Nucleic acids research*, 37(suppl_1): D674–D679, 2009.

Divya Sharma and Wei Xu. Regenne: genetic pathway-based deep neural network using canonical correlation regularizer for disease prediction. *Bioinformatics*, 39(11):btad679, 2023.

Artan Sheshmani and Yi-Zhuang You. Categorical representation learning: morphism is all you need. *Machine Learning: Science and Technology*, 3(1):015016, 2021.

Dan Shiebler, Bruno Gavranović, and Paul Wilson. Category theory in machine learning. *arXiv preprint arXiv:2106.07032*, 2021.

Tiago C Silva, Antonio Colaprico, Catharina Olsen, Fulvio D'Angelo, Gianluca Bontempi, Michele Ceccarelli, and Houtan Noushmehr. Tcga workflow: Analyze cancer genomics and epigenomics data using bioconductor packages. *F1000Research*, 5, 2016.

Ruth Stoney, David L Robertson, Goran Nenadic, and Jean-Marc Schwartz. Mapping biological process relationships and disease perturbations within a pathway network. *NPJ systems biology and applications*, 4(1):22, 2018.

Ruth A Stoney, Ryan M Ames, Goran Nenadic, David L Robertson, and Jean-Marc Schwartz. Disentangling the multigenic and pleiotropic nature of molecular function. *BMC systems biology*, 9 (6):1–15, 2015.

Liang Sun, Shuiwang Ji, and Jieping Ye. Hypergraph spectral learning for multi-label classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 668–676, 2008.

Yi-Ching Tang, Rongbin Li, Jing Tang, W Jim Zheng, and Xiaoqian Jiang. Safer: sub-hypergraph attention-based neural network for predicting effective responses to dose combinations. *Research Square*, 2024.

Linas Vepstas. Sheaves: a topological approach to big data. *arXiv preprint arXiv:1901.01341*, 2019.

Peihao Wang, Shenghao Yang, Yunyu Liu, Zhangyang Wang, and Pan Li. Equivariant hypergraph diffusion neural operators. *arXiv preprint arXiv:2207.06680*, 2022.

Tongxin Wang, Wei Shao, Zhi Huang, Haixu Tang, Jie Zhang, Zhengming Ding, and Kun Huang. Mogonet integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification. *Nature Communications*, 12(1):3445, 2021.

John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.

Sam FL Windels, Noël Malod-Dognin, and Nataša Pržulj. Identifying cellular cancer mechanisms through pathway-driven data integration. *Bioinformatics*, 38(18):4344–4351, 2022.

Yiding Yang, Zunlei Feng, Mingli Song, and Xinchao Wang. Factorizable graph convolutional networks. *Advances in Neural Information Processing Systems*, 33:20286–20296, 2020.

Guangchuang Yu. Gene ontology semantic similarity analysis using gosemsim. *Stem Cell Transcriptional Networks: Methods and Protocols*, pp. 207–215, 2020.

Guangchuang Yu, Fei Li, Yide Qin, Xiaochen Bo, Yibo Wu, and Shengqi Wang. Gosemsim: an r package for measuring semantic similarity among go terms and gene products. *Bioinformatics*, 26(7):976–978, 2010.

Guangchuang Yu, Li-Gen Wang, Yanyan Han, and Qing-Yu He. clusterprofiler: an r package for comparing biological themes among gene clusters. *Omics: a journal of integrative biology*, 16 (5):284–287, 2012.

Yang Yuan. A categorical framework of general intelligence. *arXiv preprint arXiv:2303.04571*, 2023a.

Yang Yuan. On the power of foundation models. In *International Conference on Machine Learning*, pp. 40519–40530. PMLR, 2023b.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Yivan Zhang and Masashi Sugiyama. A category-theoretical meta-analysis of definitions of disentanglement. In *International Conference on Machine Learning*, pp. 41596–41612. PMLR, 2023.

Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Exploring edge disentanglement for node classification. In *Proceedings of the ACM Web Conference 2022*, pp. 1028–1036, 2022.

Fan Zheng, Marcus R Kelly, Dana J Ramms, Marissa L Heintschel, Kai Tao, Beril Tutuncuoglu, John J Lee, Keiichiro Ono, Helene Foussard, Michael Chen, et al. Interpretation of cancer mutations using a multiscale map of protein systems. *Science*, 374(6563):eabf3067, 2021.

Appendix

A Dataset and Experiment Details

A.1 Statistics : Cancer Subtype Classification Dataset

The statistics of cancer datasets are shown in the Table 2. Note that every hypergraphs in all 8 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 A.2

Table 2: Statistics of 8 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
KIPAN	3 class, 649 hypergraphs	KICH 65, KIRC 313, KIRP 271
NSCLC	2 class, 813 hypergraphs	LUAD 451, LUSC 362

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

A.2 Preprocessing : Cancer Subtype Classification Dataset

The overall procedure was adopted from Pathformer (Liu et al., 2023). 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 (Kanehisa & Goto, 2000), PID (Schaefer et al., 2009), Reactome (Croft et al., 2010) and Biocarta.(Nishimura, 2001). 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 (Reimand et al., 2019). 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 (Weinstein et al., 2013) using TCGAbiolinks (Colaprico et al., 2016; Silva et al., 2016; Mounir et al., 2019) R library. For the rest of 6 cancer datasets we downloaded cancer subtypes from Broad GDAC Firehose (`https://gdac.broadinstitute.org/`)[8]. KIPAN and NSCLC, specifically, was created by integrating KIRC, KICH, KIRP and LUAD, LUSC each as shown in Table 2. This is the reason why it is easy to classify cancer subtypes in KIPAN dataset.

[8]Pathformer used labels from pan-cancer atlas study (Sanchez-Vega et al., 2018) 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

GENERATING NODE FEATURES

We gathered mRNA/miRNA expression, DNA methylation[9], DNA copy number variation (CNV)[10] using TCGAbiolinks. Gene lengths were acquired from biomaRt R package (Durinck et al., 2009; 2005). The procedure of processing each data with Gistic2 (Mermel et al., 2011), 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.

A.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 context labels for each hyperedge. However, there is no data that annotates the functional semantics of genetic pathways. Instead, we rely on the methods in computational biology to measure and create ground truth.

We clustered functionally similar pathways and measured functional similarity between clusters. Since each cluster is consisted of functionally similar pathways, we can consider each cluster index as a kind of a label that indicates a functional context type. By comparing the functional similarity between clusters earned from model and ground truth, we can check whether the model effectively captured functional semantics of pathways. If the similarity patterns between clusters (i.e., relative similarity scores that are shown as color in heatmap) of predicted result and the ground truth are similar, we can conclude that model could capture functional semantics. 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 $\alpha_{e_i}^k$) and ground truth (algorithm used in computational biology).

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 **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 A.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 A.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 (Windels et al., 2022; Stoney et al., 2018), 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 circumstances or condition, by selecting those pathways, we can compare functional similarities that are specific to our data or cancer[11]. The details for selecting pathways are described in Appendix A.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 ground truth pathway functions. Similarity calculations and clusterings are based on the annotation of pathway functions. The details are described in Appendix A.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

[9] but we do not use promoter methylation

[10] but we do not use gene level CNV

[11] 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

in clustering and generating ground truth functional similarity between clusters. The details are dealt in Appendix A.5.

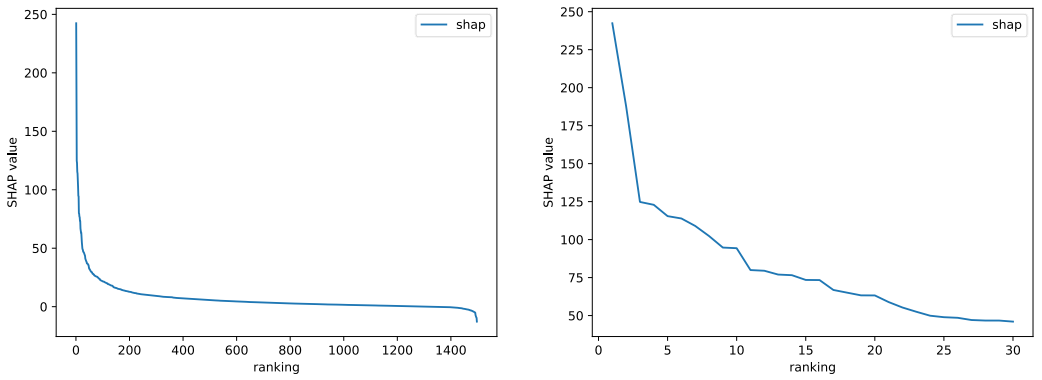
How to cluster functionally similar pathways in a reliable manner. With functional similarity between pathways, we can cluster functionally similar pathways with CliXO algorithm. The details and example results are shown in Appendix A.6.

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 A.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. The detailed explanation for the result is provided in Appendix E.5.

A.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) 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 7. 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 9.



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

Figure 7: SHAP value distribution of Natural-HNN on BRCA dataset. X axis represents ranking and Y axis represents SHAP value.

A.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) (Ashburner et al., 2000; Aleksander et al., 2023) 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 8. We can computationally annotate pathway functions with GO terms using GO enrichment analysis. We use 'enrichGO' function provided by R package clusterProfiler (Yu et al., 2012), with pvalue of 0.01 followig the paper (Stoney et al., 2018). Then we

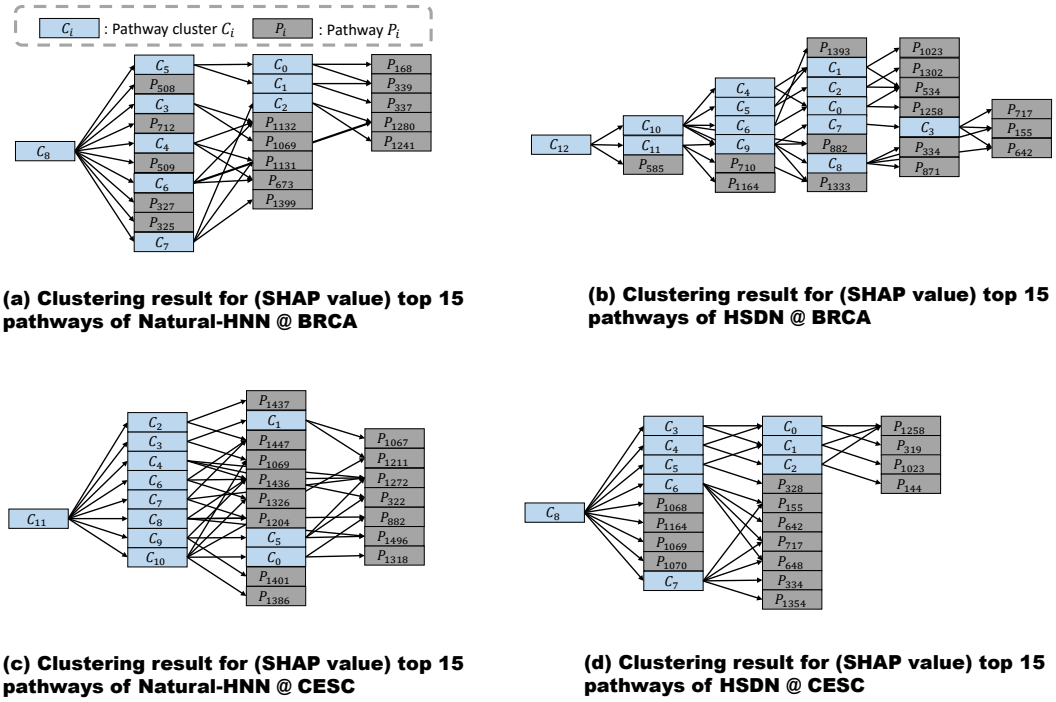


Figure 8: The result of applying CliXO algorithm to top-15 pathways of Natural-HNN and HSDN on BRCA and CESC. The pathway number denotes the index of pathway in our dataset (hyperedge index).

selected the most specific GO terms with set cover algorithm proposed in (Stoney et al., 2018) to assign pathways precise representation of their functions.

The next step is calculating functional semantic similarities between pathways. We used Lin's method (Lin et al., 1998) 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 (Liu & Thomas, 2019). We used mgoSim function in R package GOSemSim (Yu et al., 2010; Yu, 2020) for the calculation of Lin's method.

A.6 Assigning Pathway Type with CliXO

To cluster functionally similar pathways, we adopted CliXO (Kramer et al., 2014). It was originally designed to cluster gene function annotations (GO) and has been used in multiple biological studies(Kratz et al., 2023; Qin et al., 2020). However, it can also be effectively applied to higher functional semantics such as pathways as in (Zheng et al., 2021). 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 8 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.

A.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.

B Implementation Details

B.1 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. As described in E.1, 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 corresponds to the 2^{nd} factor representation and so on.) Hence, in the right lane of Figure 4, the concatenation operation is not performed as the output of a single MLP is equivalent to a concatenated vector. The nonlinear activation function we used for factor encoder is hyperbolic tangent (tanh).

B.2 Hyperparameter search space

We report the hyperparameter search space of each model in 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 shows the hyperparameter search space in the cancer subtype datasets. '♯ 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.

Table 4: 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	1,2,4,8	-	-	96

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 unless the paper of a model fixed the number of layers to a specific number. 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 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 Ablation studies and Additional Experiments

C.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 5 shows the result for cancer subtype classification task.

Table 5: 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	KIPAN	NSCLC
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	0.934 ± 0.010	0.962 ± 0.013
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	0.934 ± 0.011	0.962 ± 0.012

As we can see in Table 5, 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).

C.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 the cancer subtype classification task are provided in Table 6.

Table 6: 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	KIPAN	NSCLC
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	0.934 ± 0.010	0.962 ± 0.013
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	0.929 ± 0.016	0.958 ± 0.016

In Table 6, 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.

C.3 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): $O(Ed_i) + O(Md_i d_o)$
- Disentangle-first Branch (MLP + aggregation): $O(Nd_i d_o) + O(Ed_o)$
- 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 $O((M+N)d_i d_o + E(d_i + d_o + 1) + Nd_o + \frac{d_o^2}{K}) = O((M+N)d_i d_o + E(d_i + d_o))$

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.

C.4 Scalability Analysis (Training Time)

We measured the time took for training 1 epoch in BRCA dataset. 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 an attention mechanism. The Table 7 and Table 8 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 7: Time took for training 1 epoch on BRCA, measured in seconds. d_c denotes hidden dimension. ♯ denotes 'number of'.

(d_c, ♯ heads)	HGNN	HCHA	HNHN	UniGCNII	AllDeepSets	Natural-HNN
(16,2)	0.217 ± 0.000	0.212 ± 0.000	0.117 ± 0.000	0.237 ± 0.000	1.195 ± 0.002	0.544 ± 0.001
(64,8)	0.831 ± 0.001	0.813 ± 0.000	0.426 ± 0.001	0.906 ± 0.001	2.463 ± 0.005	1.853 ± 0.002

Table 8: Time took for training 1 epoch on BRCA, measured in seconds. d_c denotes hidden dimension. ♯ denotes 'number of'.

(d_c, ♯ heads)	AllSetTransformer	HyperGAT	SHINE	HSDN	ED-HNN	ED-HNNII	Natural-HNN
(16,2)	1.108 ± 0.002	0.711 ± 0.001	0.675 ± 0.001	0.289 ± 0.000	2.042 ± 0.003	3.852 ± 0.006	0.544 ± 0.001
(64,8)	2.671 ± 0.002	2.415 ± 0.003	2.204 ± 0.002	0.996 ± 0.000	3.558 ± 0.005	6.169 ± 0.014	1.853 ± 0.002

C.5 Cancer Subtype Classification (Micro F1)

We briefly provide Micro F1 scores of each model in cancer subtype classification task. The Table 9 also shows that our model generally performs well on most of cancer datasets.

Table 9: 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	KIPAN	NSCLC
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	0.935 ± 0.010	**0.960** ± 0.016
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	0.939 ± 0.014	0.954 ± 0.009
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	0.931 ± 0.020	0.958 ± 0.016
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	**0.947** ± 0.010	0.950 ± 0.017
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	0.923 ± 0.016	0.954 ± 0.010
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	0.938 ± 0.011	0.954 ± 0.014
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	0.913 ± 0.036	0.928 ± 0.019
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	0.907 ± 0.256	0.940 ± 0.009
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	0.866 ± 0.149	0.879 ± 0.098
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	0.936 ± 0.012	0.954 ± 0.013
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	**0.950** ± 0.008	**0.962** ± 0.013
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	0.942 ± 0.012	0.955 ± 0.012
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	0.942 ± 0.010	0.953 ± 0.012
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	**0.944** ± 0.009	**0.962** ± 0.013

C.6 Captured Context in CESC

Figure 9 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.

C.7 Factor Discrimination Analysis

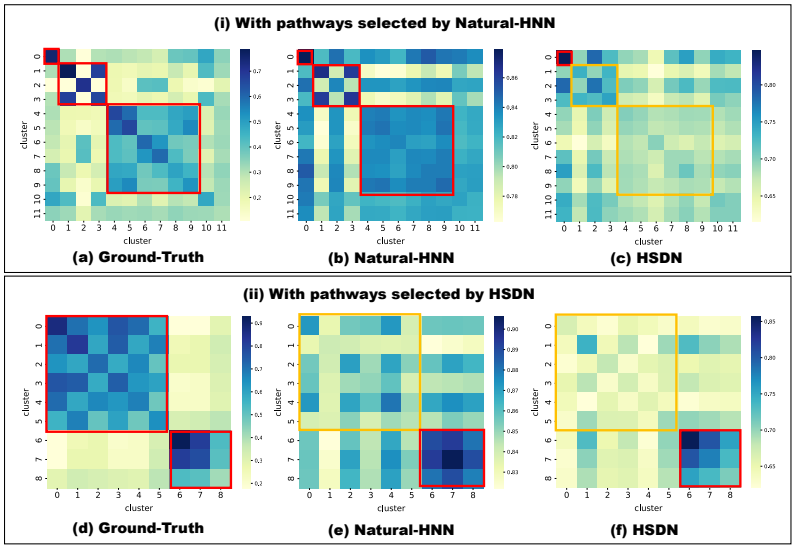


Figure 9: Captured interaction context. Pathways are selected by SHAP value. Captured patterns are shown in red box and not captured patterns are shown with orange box. Weakly captured case is marked as dotted red block.

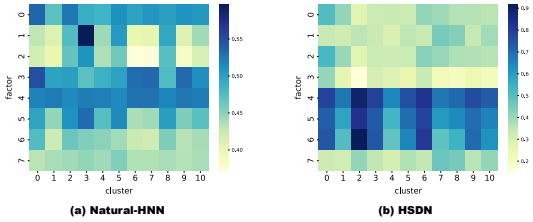


Figure 10: Factor-Cluster Relevance. For the pathways that belongs to the same cluster, we averaged their factor relevance score for each factors. (a) Natural-HNN case shows that each factor contributes to clusters differently. (b) HSDN case shows that some factors have similar contribution over all clusters.

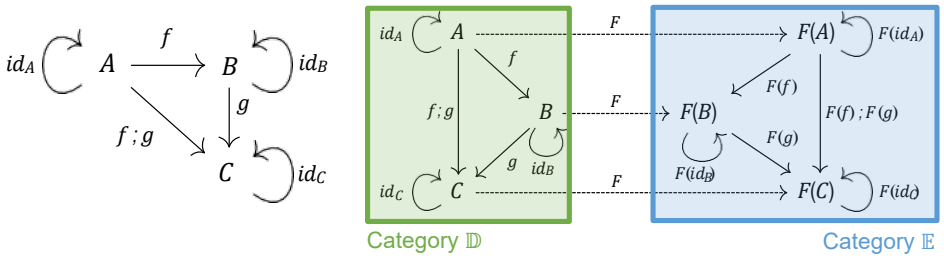
Finally, we perform an experiment to clarify that factors captured by Natural-HNN potentially have different contexts. Since each factor encodes different context and since clusters generated by CliXO algorithm assigns functionally (i.e., context) related hyperedge types, each factor is likely to be related to different clusters. Thus, for each factor and for each cluster, we averaged relevance scores α_i^k of hyperedges that belong to the same cluster. The cluster that is relevant to a specific factor would have high value while irrelevant factors would have small value for that cluster. Figure 10 shows the result of Natural-HNN and HSDN. We have the following observations: **1)** In Natural-HNN, each factor has a different score distribution over clusters. This implies that the factors are contributing to different clusters since they encode different functions. **2)** In HSDN, some factors have similar distribution over clusters. For example, factor 0 and factor 2 of HSDN are similar in every factor. Also, factor 1 and factor 7 have highly similar score distribution over clusters. This implies that some factors of HSDN are correlated. **3)** While scores in (a) are distributed to various clusters and factors, scores in (b) are concentrated on factor 4,5 and 6. Since only few factors are actively reflected while others do not, HSDN fails to utilize different factors effectively. This experiment result is notable since Natural-HNN could capture different context per factor even without factor discrimination loss $\mathcal{L}_{dis}$ while HSDN failed to capture different factors and failed to use them properly even if it adopted factor discrimination loss. Thus, we can consider naturality guidance as an effective criterion for disentanglement.

D BASIC CONCEPTS IN CATEGORY THEORY

D.1 CATEGORY THEORY

Category theory (Fong & Spivak, 2018; Leinster, 2016) 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 (Bergomi & Vertechi, 2022; Lewis, 2019; Gavranović, 2019; Fong & Johnson, 2019; Fong et al., 2019; Cruttwell et al., 2022; Shiebler et al., 2021; de Haan et al., 2020; Barbiero et al., 2023; Yuan, 2023b; Dudzik et al., 2023; Dudzik & Veličković, 2022; Yuan, 2023a). In this paper, we only use simple, fundamental concepts of category theory: category, functor, natural transformation and product.

D.2 CATEGORY



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

Figure 11: 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 11 (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 : $f \fatsemi g = g \circ f \in Hom_{\mathbb{C}}(A, C)$[12].
- 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. 11 (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 (Sheshmani & You, 2021), 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**.

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

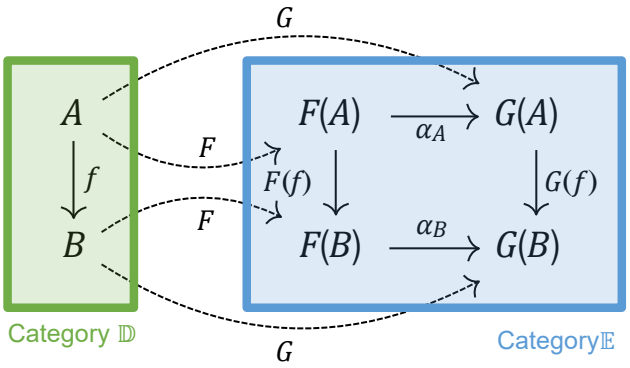


Figure 12: Natural transformation. Identity morphisms are omitted in the figure for simplicity.

D.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 11 (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[13].

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).

D.4 Natural Transformation

Given two functors mapping from one category to another category, i.e., F and $G : \mathbb{D} \to \mathbb{E}$, 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 $\alpha_A : F(A) \to G(A)$ in $\mathbb{E}$. The natural transformation must satisfy the following condition. For every morphism $f : A \to B$ 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 12 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 $\mathbb{E}$[14].

D.5 Product

Product of 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 13 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 **pairing** $\langle f_1, f_2 \rangle : Y \to X_1 \times X_2$

[13] The typical example of deep learning method using this concept is sheaf neural network (Hansen & Gebhart, 2020), motivated from cellular sheaf (Hansen & Ghrist, 2019). There are also numerous studies in data science with a similar perspective (Mansourbeigi, 2018; Vepstas, 2019; Kvinge et al., 2021).

[14] One typical example of deep learning method using this concept is Natural Graph Networks (de Haan et al., 2020).

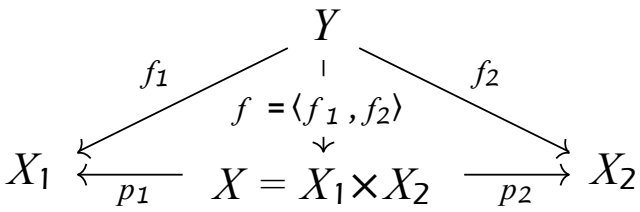


Figure 13: Product of objects.

that satisfies the composition : $f_1 = \langle f_1, f_2 \rangle \fatsemi p_1$ and $f_2 = \langle f_1, f_2 \rangle \fatsemi p_2$. Note that pairing $\langle f_1, f_2 \rangle$ is often called as product of morphisms. However to differentiate the concept we introduce below, we will call it pairng, following the recent work (Zhang & Sugiyama, 2023).

Product of Morphisms

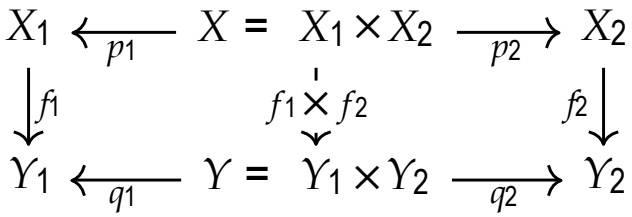


Figure 14: Product morphisms.

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 14.

E ADDITIONAL EXPLANATION IN DETAILS

Note that the basic concepts in category theory are described in Appendix D.

E.1 DISENTANGLED REPRESENTATION LEARNING

Entangled and Disentangled Representation Disentangled representation learning aims to separate the factor that is related to the variations of the data. For example, some might try to discover the factor that affects the color of an object or the factor that affects the background of an image. In graph neural networks, interactions between entities are usually entangled. In other words, interactions usually contain various factor behind connections but are not explicitly separated. Previous works like DisenGCN (Ma et al., 2019) tried to disentangle the factor behind the connections.

Recently, DisGNN (Zhao et al., 2022) tried to disentangle edge types during message passing process of GNNs. The paper considered interaction types (colleague or neighbors as an example) as factors of edges and tried to integrate disentanglement during message passing process. This kind of disentanglement for message passing is the goal of Natural-HNN.

Disentangling as product in category theory

Disentangling methods try to separate an entity into the factors that consists the entity. Thus, it can be analyzed with concept with product in category theory, which was explained in Appendix D. Although recent work (Zhang & Sugiyama, 2023) analyzed the concept of disentanglement, we are going to analyze it in our way, since the paper (Zhang & Sugiyama, 2023) covers disentanglement of generative factors, which does not suit for message passing framework. The difference comes from the fact that, generative factor disentanglement is based on equivariance property, whose morphisms maps an object to itself. Since message passing maps one object to the other object, we need our own way of analyzing disentanglement[15].

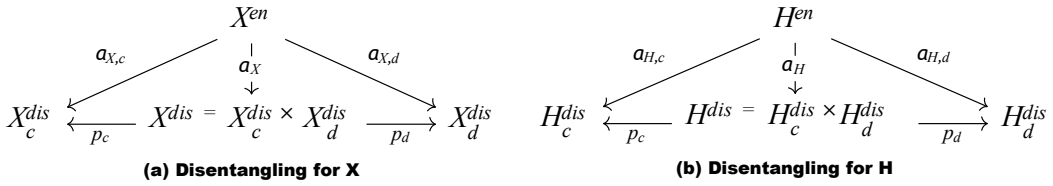


Figure 15: Disentangling as product of objects.

In section 3, we have seen that disentangling the entangled representation can be seen as a natural transformation between two representations. The Figure 15 shows the disentanglement as product of objects. The entangled representation for X (X^{en}) can be converted to disentangled representation X^{dis} through natural transformation $\alpha_X = \langle \alpha_{X,c}, \alpha_{X,d} \rangle$. Since disentangled representation is a collection of factor representations, it can be represented as a product of factor representations $X_c^{dis} \times X_d^{dis}$. The projections p_c, p_d can extract factor representations X_c^{dis}, X_d^{dis}. This process is the same as applying $\alpha_{X,c}, \alpha_{X,d}$ respectively. This is the same for disentangling H.

Figure 16 shows how morphisms between disentangled node representations and disentangled hyperedge representations are separated. Disentangling morphisms can be explained with the concept of product of morphisms. In the Figure 16, f_c^{dis},f_d^{dis} represents factor specific morphisms or factor specific message passing. The product of two morphisms, $f_c^{dis} \times f_d^{dis}$, corresponds to message passing for entire factors. What is different from Figure 14 is that we use the same projections p_c instead of using two different projections p_1, q_1. This is due to the fact that X^{dis} and H^{dis} both are disentangled representation, meaning that the same projection can extract the same factor for both X, H.

Implementation as MLP

[15] Actually, the biggest difference is that, in generative factor, factor specific morphisms can be independently mapped to itself. However, in message passing, we need to map all factor related morphisms from one object (X) to the other (H). If only some of them are used independently, it will be mapped to the another object (not H).

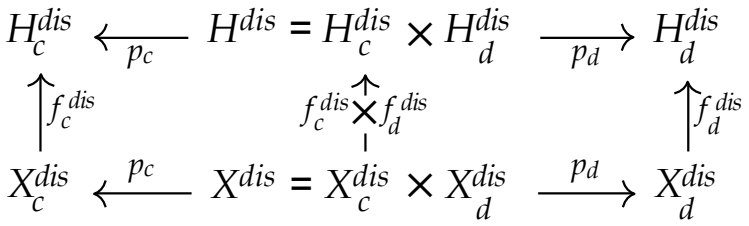


Figure 16: Morphism of products in disentanglement.

Usually, disentangling entangled representation is implemented with MLP. Let's suppose the desired output size of disentangled representation (i.e., output size of a vector that concatenated every factor representations) is d. Usually, K number of factor-specific MLPs (which outputs vector with size $\frac{d}{K}$) are used to extract factor representations. This corresponds to X_c^{dis}, X_d^{dis} in Figure 15. As we have seen above, it is same as applying $\alpha_X \fatsemi p_c, \alpha_X \fatsemi p_d$. This can be implemented as using one MLP (which outputs vector with size d), which corresponds to α_X and then chunking the disentangled representations into factor representations. Chunking operation can be considered as projections (p_c, p_d). Thus, although we explained as using K factor specific MLPs in Section 4, we actually use one MLP (which outputs vector with size d) in actual implementation. Thus, the concatenation operation for h_v is not used in the implementation as applying a single MLP equals to the operation of applying K separate MLPs and then concatenating them.

E.2 Capturing Inherent Heterogeneity

Actually, capturing context of interaction has potential of capturing heterogeneous edge types. Let's consider the case of heterogeneous graph with heterogeneous edges as an example. GNNs reflecting the edge types can be said as considering the context of interactions between entities. Thus, capturing interaction context in hyperedges has potential of capturing heterogenous edge types by considering edge types as categorized result of interaction contexts.

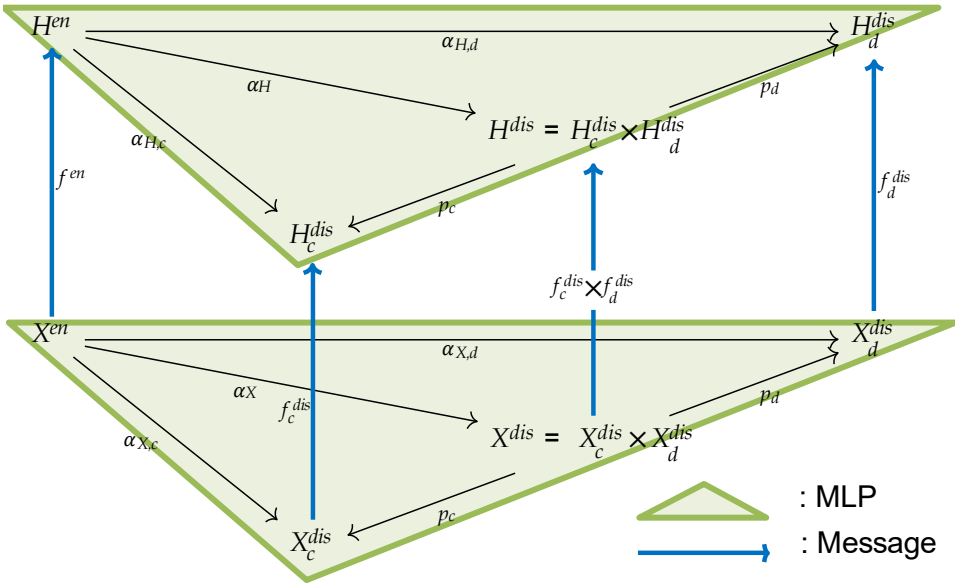


Figure 17: Entire compositional structure. Operations in the implementation are marked with color.

E.3 INTERPRETATION FOR HYPERGRAPH MPNN

In Appendix E.1, we have seen how we can analyze disentanglement with concepts of product in the category theory. Applying Figure 15 and Figure 16 to Figure 3 (a) gives the following result (Figure 17). Since this diagram is too complicated, we simplified the figure by extracting factor c related components which resulted Figure 3 (b). The resulting figure is also a natural transformation as it can be seen as a result of applying two different functors. The actual implementation (operation) are marked as the Figure 17.

E.4 METHODOLOGY (HOW IT WORKS)

Since K MLPs are applied to nodes in a hyperedge, it extracts information related to the factors through projection. However, simple projection does not mean that the factor is related to the interaction context. In this section, we will explain how naturality condition guides, although not guarantees, each factors to be related to interaction context. The parameters of factor encoders (K MLPs) are guided to extract interaction context related information during training process. When a specific factor is helpful for performance (predicting labels), the model would try to update parameters of the factor encoder so that the factor information is reflected a lot in hyperedge representation. Since relevance score α_i^k is multiplied to factor representation to get hyperedge represenation ($\alpha_i^k h_{e_i}^k$), the parameters will be updated to increase relevance score α_i^k. Considering that relevance score α_i^k is calculated by measuring consistency of factor representation (similarity of hyperedge factor representation learned from two different branches), high relevance score means that the representations are similar. Represenations learned from two branches being similar means that it is highly likely that the naturality condition holds, implying that there exists a morphism between nodes in a hyperedge and the hyperedge under specific projection (type) which means the factor is related to the interaction context. In summary, if a specific context (factor) is informative, the parameter of a factor encoder will be updated to the direction of satisfying naturality. Thus, the factor encoder will eventually encode context-related information. When a specific factor is harmful for performance, the opposite would happen. Since naturality condition guides in which direction to update parameters for each factor, although not guaranteed, it is highly likely that each factor contains different context information.

E.5 RESULT ANALYSIS OF CAPTURING CONTEXT

Actually, Figure 8 (a) and (c) can explain the experiment result shown in Figure 5 (a,b) and Figure 9 (a,b). For example, in Figure 8 (a), we can see that cluster C_0 and C_1 both have common parent (C_5) and common child (P_{339}). That's the reason why Figure 5 (a) and (b) both detected high similarity between those clusters. Also, in Figure 8 (a), C_3 and C_4 has common child. This can explain why Figure 5 (a) and (b) both detected high similarity between two clusters. When applying these analysis with Figure 5 (c) and Figure 9 (c), we can clearly see that HSDN failed to capture functional similarity or hierarchy of pathways.

On the other hand, when comparing Figure 8 (b) and Figure 5 (f), we can see some similarities are not captured. For example, in Figure 8 (b), clusters C_0, C_1, C_2 need to have functional similarity since they contain common children or have common parent. However, in Figure 5 (f), we can see that HSDN failed to capture the functional similarities of those clusters. Through this result, we can again conclude that HSDN failed to capture functional context while Natural-HNN could capture it.

Additionally, we can explain why some diagonals of heatmap do not have high value. For example, C_8 in Figure 5 (a) and (b) cannot have high similarity between pathways within cluster C_8 as C_8 contains all pathways. Note that performing the same analysis with Figure 8 (c), (d), Figure 9 gives the similar result.