Learning to Explain Hypergraph Neural Networks

Anonymous Authors[1]

Abstract

Hypergraphs are expressive structures for describing higher-order relationships among entities, with widespread applications across biology and drug discovery. Hypergraph neural networks (HGNNs) have recently emerged as a promising representation learning approach on these structures for clustering, classification, and more. However, despite their promising performance, HGNNs remain a black box, and explaining how they make predictions remains an open challenge. To address this problem, we propose HyperEX, a post-hoc explainability framework for hypergraphs that can be applied to any trained HGNN. HyperEX computes node-hyperedge pair importance to identify sub-hypergraphs as explanations. Our experiments demonstrate how HyperEX learns important sub-hypergraphs responsible for driving node classification to give useful insight into HGNNs.

1. Introduction

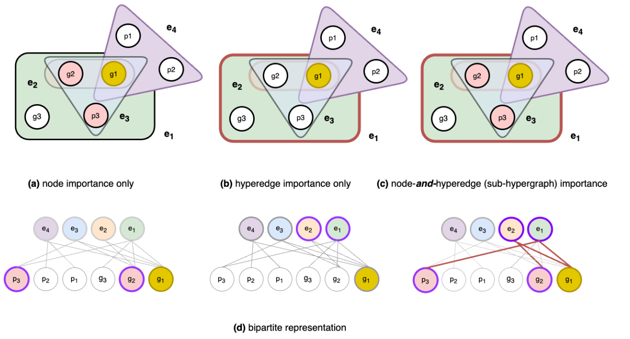
Hypergraphs are a powerful tool for modeling complex relational data, particularly when there are higher-order interactions that simple graphs fail to capture (Benson et al., 2016; Wenping et al., 2022). Whereas standard graphs encode binary relationships, hypergraphs generalize this idea to sets, where a single hyperedge can connect any number of entities (Wenping et al., 2022). As a result, hypergraphs have proven to be a useful representational structure across domains, including social networks, biological networks of genes and proteins, and more (Estrada & Rodríguez-Velázquez, 2006). Recent studies have demonstrated how hypergraph neural networks (HGNNs) can expressively encode information in hypergraph-structured data and achieve excellent predictive performance across many learning tasks (Chien et al., 2022; Wei et al., 2022; Gao et al., 2022), especially in many biological settings where multi-way interactions are a more natural representation (Zhang et al., 2020; Zhang & Ma, 2020; Chan et al., 2022). However, despite their success, HGNNs are not easily interpretable by humans due to their black-box nature. As in similar domains, explainablility methods can help us understand these models and their predictions (Adadi & Berrada, 2018), and provide a basis for further improvement (Ying et al., 2019; Yuan et al., 2023). To the best of our knowledge, there are no *general* methods specifically designed for HGNN explainability that account for the complex nature of hypergraphs (Gao et al., 2022).

Recently, a number of methods have been developed for explaining graph neural networks (GNNs) based on computed gradients, perturbation, and training of surrogate models and provide explanations in the form of scored nodes or edges (Yuan et al., 2023). Although effective for graphs, node-based approaches fail to elucidate key relationships, whereas edge-based approaches are inherently pairwise in nature. Collectively, these approaches do not capture the higher-order set relationships, *i.e.* node-hyperedge association, that are the essence of hypergraph learning. Unfortunately, understanding the impact of a single hyperedge requires considering all the nodes it connects, and existing methods based on combinatorial optimization become intractable for hypergraphs due to the exponential increase in sub-hypergraph candidates. Furthermore, the multi-set nature of hypergraphs and hyperedges introduces inherent heterogeneity in learning models that may be challenging to explain with established methods. New methods that can account for these challenges are critical for explainability.

Given the lack of established methods for HGNNs, we propose a novel approach for explaining the decisions made by hypergraph neural networks by finding important sub-hypergraphs with respect to both nodes and hyperedges. Specifically, given a hypergraph dataset and a trained HGNN model, *we aim to identify the sub-hypergraphs that are most important for the model's predictions, and show how removing these sub-hypergraphs significantly changes the model's predictions*. Our framework, HyperEX, leverages a simple attention mechanism to enable post-hoc explainability for HGNNs, and provides insights into the specific substructures and node-hyperedge associations in the hypergraph that the model relies on (Figure 1).

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by *the 2023 ICML Workshop on Computational Biology*. Do not distribute.



(a) node importance only

(b) hyperedge importance only

(c) node-***and***-hyperedge (sub-hypergraph) importance

(d) bipartite representation

Figure 1. Illustration of various explainability techniques applied to a hypergraph representation of a gene-protein interaction network. Approaches that focus solely on identifying important nodes (a) or hyperedges (b) *alone* may still be ambiguous in their explanations. In contrast, identifying sub-hypergraph that incorporate both key nodes and hyperedges can capture the full spectrum of interactions and relationships within the complex system (c). The node of interest is highlighted in yellow with important neighbors in red.

2. Related Work

Both HGNNs and explainability in machine learning are active areas of research. Numerous approaches for HGNNs have recently been reported (Feng et al., 2019; Yadati et al., 2019; Arya et al., 2020; Zhang et al., 2020; Chien et al., 2022; Wei et al., 2022) due to their ability to learn complex relationships among entities (Gao et al., 2022). Our work seeks to explain these powerful yet complex models, as no methods have been developed for their post hoc explainability. Adjacently, GNNs have proven to be powerful for learning on graph-structured data (Bronstein et al., 2021; Wu et al., 2022). With their application has also come the need for explainability methods, as they play a crucial role in understanding GNN models and their decisions (Pope et al., 2019; Yuan et al., 2023). These methods do not naturally extend to the challenges presented when generalizing beyond binary edges and are unable to account for the heterogeneous nature of hypergraphs. A more comprehensive overview of related work can be found in Appendix A.

3. Problem Formulation

Hypergraphs Explanability Overview. A hypergraph $\mathcal{H}$ is defined as a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents a set of nodes and $\mathcal{E}$ represents a set of hyperedges. A hyperedge $e \in \mathcal{E}$ is a subset of $\mathcal{V}$, indicating the nodes it connects. In our setting, each node $v_i \in \mathcal{V}$ is associated with a feature vector $x_i \in \mathbb{R}^d$, where d denotes the feature dimension. In the context of hypergraphs and HGNNs, post-hoc explainability refers to the process of interpreting the decisions made by the HGNN model after training. Hence, given a trained HGNN, $f(\mathcal{H})$, the goal of post-hoc explainability methods is to offer insight into the *model* $f(\mathcal{H})$.

Generally, there are three ways to provide hypergraph explanations based on their nodes, their hyperedges, or the relationships between them. We further illustrate this using a gene-protein interaction network. Figure 1 shows a hypothetical gene-protein interaction using a hypergraph model where genes and proteins are the nodes of this hypergraph, and hyperedges are the relations between them. An explanation for the gene-protein hypergraph could consist of genes and proteins (node-based), gene-protein relations (hyperedge-based), or provide a subset of genes, proteins, and their interactions (sub-hypergraphs).

Node-based Explanations. One approach is to find important nodes responsible for the model prediction (Figure 1). However, focusing on key entities alone overlooks the *relationships* between them, which are critical for many applications. Furthermore, if a node's importance is largely due to its role within certain hyperedges, focusing on individual nodes could misrepresent the true dynamics. For example, in Figure 1 (a), an explanation focused only on finding key genes and proteins in a network may be confounded by the many possible interactions.

Hyperedges-based Explanations. Hyperedge-based approaches instead focus on the relationship between entities. However, when hyperedges relate many nodes, it can be difficult to interpret the exact nature of the relationship. For example, in Figure 1 (b), an explanation only finds important interactions, while it is not clear if all or only some genes/proteins are important in these interactions.

Sub-hypergraph-based Explanations. Finally, one can find important nodes with respect to hyperedges of a hypergraph. In other words, the sub-hypergraph could be edges in a bipartite representation of a hypergraph that connects nodes to hyperedges. This way of explainability is more general compared to the other approaches as it balances the focus on nodes and hyperedges and offers a more comprehensive view of the network dynamics that can be adapted to different tasks. For example, the model in Figure 1 (c), show significant edges in the bipartite representation which creates a more comprehensive explanation by highlighting which both key genes and proteins are critical connectors and their hyperedges. These could be genes and proteins that might not have the most interactions (high degree nodes) or be part of the largest clusters of interactions (large hyperedges) but serve as key links within the system.

Identifying Sub-hypergraphs with Star-Expansion. Although hypergraphs can be readily represented as multisets based on the formalism above, reformulating them can provide a further basis for interpretation. For example, two commonly-used approaches include clique-expansion (Sun et al., 2008), where a graph is constructed by replacing every hyperedge by fully-connecting its vertices, and star-expansion, where hyperedges are replaced by new

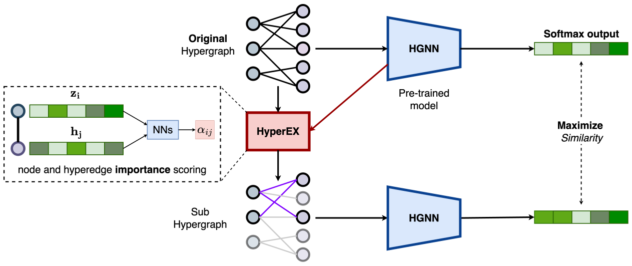


Figure 2. Overview of HyperEX framework. HyperEX learns an importance score α_{ij} between nodes and their hyperedges (left) which defines an explanatory sub-hypergraph. The learning objective maximizes the mutual information between the outputs of the original hypergraph and learned sub-hypergraph.

nodes (Zien et al., 1999). Importantly, the star-expansion formalism provides a convenient basis for interpretability as it establishes a new set of node-hyperedge relationships. Formally, in star-expansion, a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ is transformed into a bipartite graph $\mathcal{H}^* = (\mathcal{V}, \mathcal{E}, \mathcal{B})$. In this bipartite representation, $\mathcal{V}$ and $\mathcal{E}$ are the original sets of nodes and hyperedges, respectively, and $\mathcal{B}$ represents set of node-hyperedge connections in the bipartite graph. We denote these new connections as $b_{ij} \in \mathcal{B}$. Each b_{ij} is indicating that node $v_i \in \mathcal{V}$ is connected to hyperedge $e_j \in \mathcal{E}$ in the bipartite graph.

4. HyperEX: Framework and Method

Here, we describe the framework for HyperEX for post-hoc hypergraph explainability. Given a hypergraph $\mathcal{H}$ and a trained HGNN, $f(\mathcal{H})$, HyperEX defines a framework to identify an explanatory, induced sub-hypergraph that explains the HGNN predictions. Figure 2 summarizes our framework.

Scoring the Importance of Node-Hyperedge Pairs. Given hypergraph $\mathcal{H}$, we first construct its star expansion representation that transforms the hypergraph into its bipartite representation, $\mathcal{H}^* = (\mathcal{V}, \mathcal{E}, \mathcal{B})$, where $\mathcal{B}$ corresponds to the new set of node-hyperedge connections in the bipartite graph. Our procedure begins by running the HGNN model, $f(\cdot)$ to obtain the model's output and using it to generate the embedding matrix $\mathbf{Z} \in \mathbb{R}^{N \times C}$, where $N = |\mathcal{V}|$ and C is the number of the classes in our training dataset. We then generate hyperedge embeddings by taking the average of their respective node embeddings:

$$\mathbf{Z} = f(\mathcal{H}), \quad \text{and} \quad \mathbf{h}_j = \frac{1}{|\mathcal{N}(j)|} \sum_{l \in \mathcal{N}(j)} \mathbf{z}_l, \tag{1}$$

Here, $\mathbf{h}_j$ denotes the embedding vector of hyperedge j, $\mathcal{N}(j)$ represents the set of nodes connected to hyperedge j, and $\mathbf{z}_l$ is the embedding vector of node l. For each node, we then define a scoring function score : $\mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}$ which is used to calculate the attention coefficients between nodes and hyperedges that serve as node-hyperedge association weights ω_{ij} in the bipartite expansion:

$$\begin{aligned} \alpha_{ij} =& \frac{\exp(\omega_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(\omega_{ik})}, \\ &\text{where} \quad \omega_{ij} = (W_Q \mathbf{z}_i)^\top \cdot (W_K \mathbf{h}_j) \cdot s_i \end{aligned} \tag{2}$$

Here W_Q and W_K are linear transformations that capture the importance of nodes and hyperedges respectively, $\mathbf{z}_i$ is the embedding of node i, $\mathbf{h}_j$ is the embedding of hyperedge j, and s_i is a learnable scalar that modulates the score based on the distance to its $\mathcal{K}$-hop neighborhood. In essence, this operation provides a normalized attention score over an expanded receptive field that classifies the relative impact of adjacent hyperedges with a $\mathcal{K}$-hop neighborhood.

Identifying Important Sub-hypergraphs. To identify important sub-hypergraphs, we first find the $\mathcal{K}$-hop neighborhood of a given node. Then, we calculate the node-hyperedge pair weights in this neighborhood using Equation (2). Finally, we pick the top-k pairs with the highest weights as the important sub-hypergraph.

Learning Objective. The learning objective of our framework is to maximize the restricted mutual information (MI) between the node embeddings obtained from the original hypergraph and the embeddings obtained from the induced sub-hypergraph. Formally, let $\mathbf{Z} = f(\mathcal{H})$ denote the node embeddings of the original hypergraph and $F_\theta(\mathcal{H})$ identify the sub-hypergraph induced by important nodes and hyperedges of the original hypergraph learned through HyperEX and $\mathbf{Z}_\theta = f(F_\theta(\mathcal{H}))$ is the node embedding of that sub-hypergraph. Our objective based on the restricted MI can be written as $\max_\theta \left(MI\left(f(\mathcal{H}), f(F_\theta(\mathcal{H})\right)\right)$.

This objective encourages the model to capture the most important structures and relationships within the hypergraph, as reflected in the sub-hypergraph. In our framework, we optimize for the noise-contrastive estimation (NCE) (Gutmann & Hyvärinen, 2010), specifically InfoNCE (van den Oord et al., 2019), as opposed to the direct computation of mutual information, due to the improved computational efficiency offered by InfoNCE.

$$\text{loss} = -\frac{1}{N} \sum_{i=1}^{N} \left(\log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_{i,\theta})}{\sum_{j \neq i} \exp(\mathbf{z}_i \cdot \mathbf{z}_{j,\theta})} \right) \tag{3}$$

where $\mathbf{z}_i$ is the embedding of node v_i in the original hypergraph, $\mathbf{z}_{i,\theta}$ is the embeddings of node v_i in sub-hypergraph, and N denotes the number of samples in a mini-batch respectively.

Table 1. Fidelity$^+$ scores with controlled sparsity using HyperGCL with generated (G), and fabricated (F) augmentations. Higher Fidelity$^+$ indicates better explanation.

HGNN	EXPLAINER	CORA	CITESEER	CORA-CA	ZOO	PUBMED
HYPERGCL (G)	HYPEREX	**0.25± 0.06**	**0.25± 0.06**	**0.34± 0.05**	**0.88 ± 0.16**	**0.11 ± 0.00**
	SALIENCY	0.05± 0.00	0.04± 0.00	0.19± 0.00	0.41± 0.04	0.04 ± 0.00
	IG	0.06± 0.01	0.14± 0.06	0.33 ± 0.01	0.41 ± 0.02	0.09 ± 0.00
HYPERGCL (F)	HYPEREX	**0.18± 0.07**	**0.18± 0.07**	**0.37± 0.05**	**0.73± 0.23**	**0.11 ± 0.01**
	SALIENCY	0.05± 0.00	0.01± 0.00	0.23± 0.36	0.41± 0.02	0.08 ± 0.01
	IG	0.06± 0.00	0.13± 0.01	0.35± 0.00	0.52 ± 0.02	**0.11± 0.01**

5. Experiments and Results

In this section, we evaluate our framework on various datasets and baselines. Complete experimental setup and details are in the Appendix.

Evaluation Metrics. We adapt the standard **fidelity**$^+$ score (Pope et al., 2019) which concentrates on whether removal of the key sub-hypergraph alters the model's prediction. In contrast, we also define *fidelity*$^-$, which corresponds to keeping *only* the important sub-hypergraph. The **sparsity** score quantifies the fraction of structures deemed significant by the explanation method (Pope et al., 2019).

HGNN models. We focus our current work on the AllSet model as our primary HGNN model (Chien et al., 2022). To augment this model, we also apply fabricated and generated augmentations from recent self-supervised pretraining literature on hypergraphs (Wei et al., 2022).

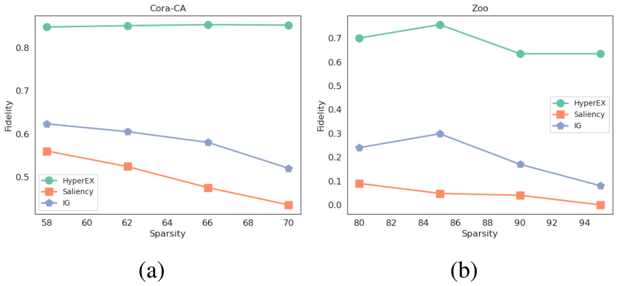
Baselines. Since currently there are no explainable models for HGNNs, we adopt two common gradient-based approaches, Saliency (Simonyan et al., 2013) and Integrated Gradients (IG) (Sundararajan et al., 2017), as our baselines to evaluate against HyperEX.

5.1. Quantitative Evaluation with Fidelity and Sparsity

Table 1 shows *fidelity*+ scores across a range of datasets, with HyperEX outperforming gradient-based methods on all tasks. These results indicate that gradient-based approaches to identify critical nodes are insufficient for hypergraph explainability. In contrast, HyperEX results in significantly higher *fidelity*. We further evaluate our framework using *fidelity*$^-$ metric on Zoo, and Cora-CA, which directly measures the quality of the learned sub-hypergraph (Figure 3). Again, our framework outperforms gradient-based approaches.

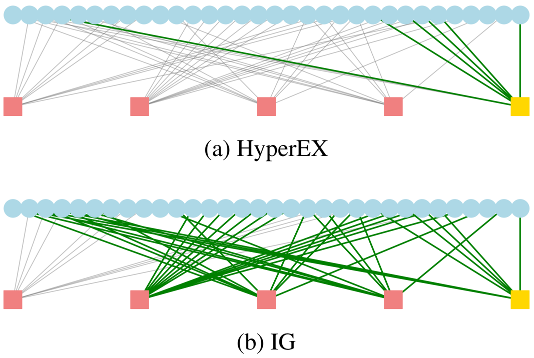
5.2. Visualization of Sub-hypergraphs Explanations

To demonstrate the ability of HyperEX to generate sparse explanations, we next generated visualizations of the resulting sub-hypergraphs and plotted them alongside results from IG (Figure 4). HyperEX (top) directly identifies node-hyperedge pairs that provide a sparse hypothesis relative to a few neighbors. In contrast, IG can only identify relevant nodes, rather than the relationships between nodes and hyperedges. As a result the generated explanations remain dense, as any possible connecting hyperedge must also be considered (bottom).



(a) (b)

Figure 3. Fidelity$^-$ on two datasets Zoo, and Cora-CA. Higher Fidelity$^-$ indicates better explanation.



(a) HyperEX

(b) IG

Figure 4. HyperEX vs Integrated Gradients analysis on Cora-CA. Explanations are generated with respect to the yellow node (far right). The green edges are sub-hypergraphs found by explainable models.

6. Conclusions and Future Directions

In conclusion, we report HyperEX, the first post hoc explainability method for HGNNs. We demonstrate how this approach can provide sparse hypergraph explanations for a range of node classification tasks across diverse datasets. The extension of this work for explaining complex biological networks is the focus of our ongoing work.

References

Adadi, A. and Berrada, M. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018. doi: 10.1109/ACCESS.2018.2870052.

Arya, D., Gupta, D. K., Rudinac, S., and Worring, M. Hypersage: Generalizing inductive representation learning on hypergraphs, 2020.

Benson, A. R., Gleich, D. F., and Leskovec, J. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016. doi: 10.1126/science.aad9029. URL https://www.science.org/doi/abs/10.1126/science.aad9029.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

Chan, L., Kumar, R., Verdonk, M., and Poelking, C. A multilevel generative framework with hierarchical self-contrasting for bias control and transparency in structure-based ligand design. *Nature Machine Intelligence*, 4(12): 1130–1142, December 2022.

Chien, E., Pan, C., Peng, J., and Milenkovic, O. You are allset: A multiset function framework for hypergraph neural networks, 2022.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/04df4d434d481c5bb723be1b6df1ee65-Paper.pdf.

Estrada, E. and Rodríguez-Velázquez, J. A. Subgraph centrality and clustering in complex hypernetworks. *Physica A: Statistical Mechanics and its Applications*, 364:581–594, 2006. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2005.12.002. URL https://www.sciencedirect.com/science/article/pii/S0378437105012550.

Feng, Y., You, H., Zhang, Z., Ji, R., and Gao, Y. Hypergraph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 3558–3565, Jul. 2019. doi: 10.1609/aaai.v33i01.33013558. URL https://ojs.aaai.org/index.php/AAAI/article/view/4235.

Gao, Y., Zhang, Z., Lin, H., Zhao, X., Du, S., and Zou, C. Hypergraph learning: Methods and practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2548–2566, 2022. doi: 10.1109/TPAMI.2020.3039374.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Teh, Y. W. and Titterington, M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/gutmann10a.html.

Huang, J. and Yang, J. Unignn: a unified framework for graph and hypergraph neural networks, 2021.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Kirkland, S. Two-mode networks exhibiting data loss. *Journal of Complex Networks*, 6(2):297–316, 08 2017. ISSN 2051-1329. doi: 10.1093/comnet/cnx039. URL https://doi.org/10.1093/comnet/cnx039.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., and Zhang, X. Parameterized explainer for graph neural network, 2020. URL https://arxiv.org/abs/2011.04573.

Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E., and Hoffmann, H. Explainability methods for graph convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10764–10773, 2019. doi: 10.1109/CVPR.2019.01103.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Sun, L., Ji, S., and Ye, J. Hypergraph spectral learning for multi-label classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pp. 668–676, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581934. doi: 10.

1145/1401890.1401971. URL https://doi.org/10.1145/1401890.1401971.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328, 2017.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding, 2019. URL https://arxiv.org/abs/1807.03748.

Wei, T., You, Y., Chen, T., Shen, Y., He, J., and Wang, Z. Augmentations in hypergraph contrastive learning: Fabricated and generative. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=igMc_C9pgYG.

Wenping, Z., Meilin, L., and Jiye, L. Hypergraphs: Concepts, applications and analysis. In *2022 IEEE 13th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, pp. 1–6, 2022. doi: 10.1109/PAAP56126.2022.10010428.

Wu, L., Cui, P., Pei, J., and Zhao, L. *Graph Neural Networks: Foundations, Frontiers, and Applications*. Springer Singapore, Singapore, 2022.

Xie, Y., Katariya, S., Tang, X., Huang, E., Rao, N., Subbian, K., and Ji, S. Task-agnostic graph explanations. In *Advances in Neural Information Processing Systems*, volume 35, pp. 12027–12039. Curran Associates, Inc., 2022.

Yadati, N., Nimishakavi, M., Yadav, P., Nitin, V., Louis, A., and Talukdar, P. Hypergcn: A new method of training graph convolutional networks on hypergraphs, 2019.

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

Yuan, H., Tang, J., Hu, X., and Ji, S. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 430–438, 2020.

Yuan, H., Yu, H., Gui, S., and Ji, S. Explainability in graph neural networks: A taxonomic survey. *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, 45(05): 5782–5799, may 2023. ISSN 1939-3539. doi: 10.1109/TPAMI.2022.3204236.

Zhang, R. and Ma, J. MATCHA: Probing multi-way chromatin interaction with hypergraph representation learning. *Cell Syst*, 10(5):397–407.e5, May 2020.

Zhang, R., Zou, Y., and Ma, J. Hyper-sagnn: a self-attention based graph neural network for hypergraphs. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryeHuJBtPH.

Zhou, D., Huang, J., and Schölkopf, B. Learning with hypergraphs: Clustering, classification, and embedding. In Schölkopf, B., Platt, J., and Hoffman, T. (eds.), *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL https://proceedings.neurips.cc/paper_files/paper/2006/file/dff8e9c2ac33381546d96deea9922999-Paper.pdf.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.01.001. URL https://www.sciencedirect.com/science/article/pii/S2666651021000012.

Zien, J., Schlag, M., and Chan, P. Multilevel spectral hypergraph partitioning with arbitrary vertex sizes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(9):1389–1399, 1999. doi: 10.1109/43.784130.

A. Related Work

Both hypergraph neural networks and explainability in machine learning are active areas of research. In this section, we concisely review related work relevant to our method:

A.1. Hypergraph Neural Networks

Hypergraphs are a powerful representational structure, especially for higher-order relationships (Benson et al., 2016; Wenping et al., 2022). Learning approaches for clustering, classification, and embedding on hypergraphs have previously been developed (Zhou et al., 2006; Benson et al., 2016). Along with the rise of graph neural networks, hypergraph neural networks have gained considerable traction in recent years due to their ability to learn complex relationships among entities (Gao et al., 2022). A basic approach to hypergraph modeling centers on restructuring hypergraphs as graphs. For example, clique expansion replaces all hyperedges with cliques that connect its member nodes, and produces a graph representation that can then be used by traditional graph-based models. Feng et al. (2019) first describe HGNN, which adopts this approach to then learn using graph spectral convolutional methods (Defferrard et al., 2016). Similarly, **?** HyperGCN use a modified clique-based expansion strategy and applies a graph convolutional network (GCN) (Kipf & Welling, 2017) to the resulting graph representation. Unfortunately, clique-expansion methods are inherently lossy conversions (Kirkland, 2017), and such ad-hoc approximations add biased to the system and cannot incorporate higher order information in the hypergraph.

Recently, several studies have proposed to work with hypergraph structure instead of converting it to a graph to avoid information loss in previous works. HyperSAGE (Arya et al., 2020) generalizes the attention mechanism to hypergraphs for performing link prediction in hypergraphs. UniGNN (Huang & Yang, 2021) performs message passing directly on a hypergraph. AllSet (Chien et al., 2022) operates by representing hyperedges as sets and performing aggregations using deep multi-set functions. This approach enables AllSet to incorporate a wide range of previous hypergraph convolution methods. HyperGCL (Wei et al., 2022) furthermore, constructs contrastive views for hypergraphs via augmentations using two schemes: fabricated and generative. Their framework is developed on top of Chien et al. (2022) and achieved the-state-of-the-art results on various hypergraph datasets. Our work here tackles the challenge of explaining these recent models to better understand their predictions, strengths, and weaknesses to ultimately improve learning on hypergraphs.

A.2. Explainability for Graph Neural Networks

GNNs have gained significant attention in recent years due to their widespread utility and application for graph-structured data (Bronstein et al., 2021; Wu et al., 2022). With their application has also come the need for explainability methods, as they play a crucial role in understanding GNN models and their decisions (Pope et al., 2019). As a result, many approaches have been proposed to explain the predictions of GNN models (Zhou et al., 2020). These models can broadly be classified based on their scope of explanation (i.e. local vs global), their method (e.g. gradient vs. perturbation-based), or their usage (model specific vs. model agnostic) (Yuan et al., 2023). In particular, XGNN (Yuan et al., 2020), GNNExplainer (Ying et al., 2019), PGExplainer (Luo et al., 2020), and SubgraphX (Xie et al., 2022), etc. are all model-agnostic approaches that have been developed from different angles to provide different levels of explanations. These approaches can all be used to explain any trained GNN model.

However, these methods mainly aim to explain GNNs by identifying important nodes, edges, or node features of the input graphs. As a result, these methods do not naturally extend to the challenges presented when generalizing beyond binary edges. Currently, none of these approaches are able to account for the heterogeneous nature of hypergraphs.

B. Addictional Experimental Setting

In this section, complete information on the experimental setup and hyperparameters can be find.

B.1. Datasets and Training

We use five standard hypergraph datasets commonly used in evaluating HGNN-based models (Wei et al., 2022), with descriptions of the datasets are shown in Table 2. In this paper, we focus on explaining predictions made by HGNNs for node classification. For each node in our training set, we set its 3-hop neighborhood for explanation. Similar to Wei et al. (2022), we split the data into 10% training, 10% validation, and 80% test for each dataset. Notably, using only a small proportion of the dataset for training ensures meaningful and generalizable explanations.

Table 2. Five hypergraph datasets used for explanability. All datasets are used for node classification and test the methods against a range of hypergraph sizes.

DATA SET	NODES	HYPEREDGES	FEATURE	CLASSES
ZOO	101	43	16	8
CORA-CA	2,708	1,072	1,433	7
CORA	2,708	1,579	1,433	7
CITESEER	3,312	1,079	3,703	6
PUBMED	19,717	7,963	500	3

B.2. Evaluation Metrics

In this section, we summarizes our evaluation metrics.

Fidelity. With post hoc graph explanations, *fidelity*$^+$ measures whether an identified sub-hypergraph is important for a model explanation. Here, we adapt the standard soft *fidelity* score (Pope et al., 2019) for a hard *fidelity*$^+$ score, which concentrates on whether removal of the key sub-hypergraph alters the model's prediction:

$$\text{fidelity}^+ = \frac{1}{N}\sum_{i=1}^{N}\left[\text{argmax}\left(f_i(\mathcal{H})\right) \oplus \text{argmax}\left(f_i(\mathcal{H}^{1-m_i})\right)\right].$$

Here, N is the number of nodes to be explained, $f_i(\mathcal{H})$ is the model's original prediction for node i given hypergraph $\mathcal{H}$, and $f_i(\mathcal{H}^{1-m_i})$ in the new prediction after removing the important sub-hypergraph, $\mathcal{H}^{m_i}$, and $\oplus$ is the XOR operator.

In contrast, we also define *fidelity*$^-$, which corresponds to keeping *only* the important sub-hypergraph:

$$\text{fidelity}^- = \frac{1}{N}\sum_{i=1}^{N}\left[\text{argmax}\left(f_i(\mathcal{H})\right) \circ \text{argmax}\left(f_i(\mathcal{H}^{m_i})\right)\right].$$

where $\circ$ is the XNOR operator and m_i is the learned sub-hypergraph. Higher *fidelity*$+$ and *fidelity*$-$ scores indicate a better explanation, as they suggest the model has identified a critical sub-hypergraph.

Sparsity. Effective explanations will encompass the most crucial sub-hypergraphs while disregarding the nonessential ones, and hence will be sparse compared to full hypergraph. The *sparsity* score quantifies the fraction of structures deemed significant by the explanation method (Pope et al., 2019). It further provides a balanced comparison by standardizing the sizes of explanations, as larger sub-hypergraphs typically increase *fidelity*. Therefore, explanations of different sizes are not directly comparable.

$$\text{sparsity} = \frac{1}{N}\sum_{i=1}^{N}(1 - \frac{|m_i|}{|V_i|})$$

where $|V_i|$ is the total number of nodes in the k-hop neighborhood of node i.

HGNN models. We focus our current work on the AllSet model as our primary HGNN model, given its demonstrated expressivity and flexibility on hypergraphs (Chien et al., 2022). To augment this model, we also apply fabricated and generated augmentations from recent self-supervised pretraining literature on hypergraphs (Wei et al., 2022), as they achieve state-of-the-art results on most hypergraph datasets.

B.3. Baselines.

Since currently there are no explainable models for HGNNs, we adopt two common gradient-based approaches, Saliency (Simonyan et al., 2013) and Integrated Gradients (IG) (Sundararajan et al., 2017), as our baselines to evaluate against HyperEX.

Each method provides a node score with respect to a single output, and we pick top-k nodes based on their score. This allows us to control the relative sparsity of the explanation for evaluation. In order to compute the *fidelity*+ score, we mask out these top-k nodes from the hypergraph, and apply the HGNN model on the remaining sub-hypergraph.

Saliency. Saliency is an explainability method that aims to identify the most important nodes in a hypergraph for a given prediction made by an HGNN model. It computes a relevance score for each node, indicating their impact on the final prediction. Formally, given a node i represented by the embedding $\mathbf{x}_i$, its saliency $S_{SM}(i)$ is defined as the absolute value of the gradients of the class-specific output score y_c with respect to $\mathbf{x}_i$:

$$S_{SM}(i) = |\nabla_{\mathbf{x}_i} y_c|. \tag{4}$$

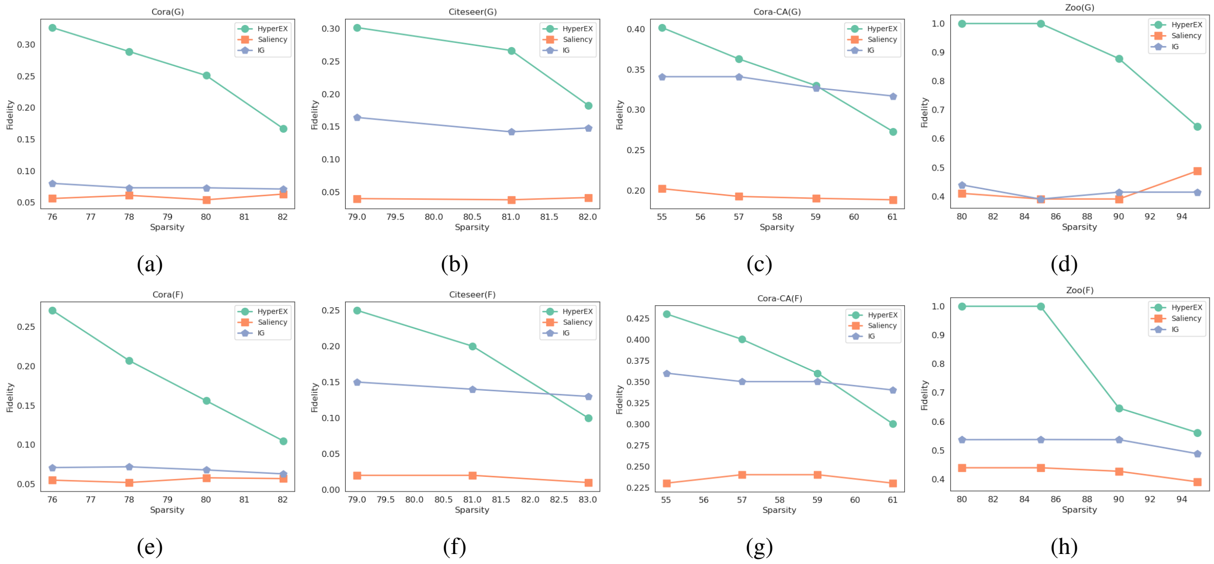
The resulting saliency scores are used for identifying the relevant sub-hypergraph.

Integrated Gradients. Integrated Gradients (IG) is an explainability method that provides insights into the contribution of each node in a hypergraph towards a specific prediction made by an HGNN. It considers a baseline input and target input to compute a path integral along a straight-line connecting these two inputs. By integrating the gradients of the prediction, IG assigns importance values to different nodes in the hypergraph. These importance values help us understand how the input features influence the HGNN's prediction. Given a node i, represented by the embedding $\mathbf{x}_i$, the IG score $SIG(i)$ is defined as:

$$S_{IG}(i) = (\mathbf{x}_i - \mathbf{x}_{i_0}) \cdot \int_{\alpha=0}^{1} \nabla \alpha \mathbf{z}_i + (1-\alpha)\mathbf{z}_{i_0} y_c, d\alpha, \tag{5}$$

where $\mathbf{x}_{i_0}$ is the embedding of a baseline node, $\nabla \alpha \mathbf{x}_i + (1-\alpha)\mathbf{x}_{i_0} y_c$ is the gradient at point $\alpha \mathbf{z}_i + (1-\alpha)\mathbf{z}_{i_0}$ along the straight-line path in the embedding space from $\mathbf{x}_{i_0}$ to $\mathbf{x}_i$, and the integral aggregates these gradients along the path. As above, we use the resulting IG score to define the sub-hypergraph.

B.4. Additional Experiments with Fidelity and Sparsity



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5. Fidelity^{+} score results on various sparsity levels. The HGNN model used in the first row is HyperGCL with generative augmentation (G), and HGNN used in the second row is HyperGCL with fabricated augmentation (F). Fidelity score is expected to drop as the sparsity level goes up. Sparsity is in %. HyperEX results are shown in green above Saliency (orange) and IG (blue).