

Appendix for “HyperGOOD: Towards Out-of-Distribution Detection in Hypergraphs”

Appendix A: Further Discussion

A.1 Experimental Details Our code is available at <https://anonymous.4open.science/r/HyperGOOD-FD64/>.

As the first method designed for the HOOD task, we conduct comprehensive experiments on nine open datasets to validate the generality and effectiveness of our approach. These datasets span a variety of real-world scenarios, including co-citation networks (Cora, Citeseer, PubMed), co-authorship networks (Cora-CA) (Yadati et al. 2019), political voting networks (Senate (Fowler 2006), House (Chodrow, Veldt, and Benson 2021)), a multi-label semantic dataset (NTU2012 (Chen et al. 2003)), a 3D object recognition dataset (ModelNet40 (Wu et al. 2015)), and a mushroom classification dataset (Mushroom (Dua, Graff et al. 2017)). Detailed statistics of these benchmarking hypergraphs are provided in Table 1.

Table 1: The statistics of datasets.

Datasets	# nodes	# hyperedges	# classes	# features
Cora-CA	2,708	1,072	7	1,433
Cora	2,708	1,579	7	1,433
Citeseer	3,312	1,709	6	3,703
Pubmed	19,717	7,963	3	500
Senate	282	315	2	100
House	1,290	340	2	100
NTU2012	2,012	2,012	67	100
ModelNet40	12,311	12,311	40	100
Mushroom	8,124	298	2	22

We evaluate HyperGOOD against four representative *generic OOD detection methods* — MSP (Hendrycks and Gimpel 2016), ODIN (Liang, Li, and Srikant 2018), Mahalanobis (Lee et al. 2018), and Energy (Liu et al. 2020) — as well as two recent *graph-based approaches*, GNNSafe (Wu et al. 2023) and NODESAFE (Yang et al. 2024).

For fair comparison:

- All generic methods are implemented with a standard multi-layer perceptron (MLP) as the backbone, except for Energy, where we adopt HGNN (Feng et al. 2019) to allow fairer comparison under hypergraph inputs.
- For graph-based baselines, which assume 2-graph inputs, we convert each hypergraph dataset into a pairwise graph format by flattening hyperedges into clique connections, i.e., connecting all node pairs that co-occur in the same hyperedge.

All baseline implementations follow their respective official settings or recommended default hyperparameters. Our comparison focuses on their OOD detection performance under various perturbation settings.

To apply graph-based OOD methods on hypergraph datasets, we transform hypergraph inputs into graphs via *clique expansion*. Specifically, for each hyperedge in the original hypergraph, we connect all pairs of nodes within that hyperedge to form a complete subgraph. This transformation enables graph-based methods to approximate higher-order relationships through pairwise connections, thereby allowing a unified evaluation across different OOD detection paradigms. However, both our experimental results and prior studies (our discussion in the Sec.Related Works) consistently indicate that such a naive conversion is insufficient for capturing the complexities of hypergraph data in OOD scenarios, underscoring the necessity of dedicated approaches tailored to hypergraph structures.

During conversion, we also preserve the node features, labels, and train/val/test splits, including OOD/IND masks where applicable, to ensure comparability in OOD detection experiments. The IND dataset is randomly split into training, validation, and test sets, with proportions of 50%, 25%, and 25%, respectively. The OOD dataset is generated by perturbing the IND dataset, following the standard practice of graph OOD detection (Cai et al. 2025), i.e., feature perturbation and label perturbation. All results are reported on the commonly used metric area under the receiver operating characteristic curve (AUC) and all experimental procedures are conducted on NVIDIA RTX A600 GPU device with 48 GB memory. All experiments are conducted using PyTorch and PyTorch Geometric. Converted datasets are saved in a consistent format and reused across relevant baselines to ensure reproducibility.

Note: Discussion on Future Work

We perform evaluation under multiple perturbation settings, including feature shift and label noise. In preliminary experiments, we also considered perturbing the hypergraph structure (e.g., modifying hyperedges or connectivity patterns) as a potential form of OOD signal. However, we found that such structural changes often led to unpredictable and non-reproducible behaviors, and lacked consistent correlation with OOD likelihood under our framework. We therefore focus our study on feature-level and semantic OOD shifts under a fixed hypergraph structure, which are more well-defined and aligned with existing literature. We believe the formalization and modeling of structural OOD in hypergraphs remains an open and important research direction.

Furthermore, many real-world scenarios may involve more complex forms of OOD shifts that remain underexplored in context of hypergraphs. These include heterophily, heterogeneous, and temporally evolving structures (Kim et al. 2024; Khan et al. 2025). Addressing such forms of distributional uncertainty presents important open challenges for future HOOD research.

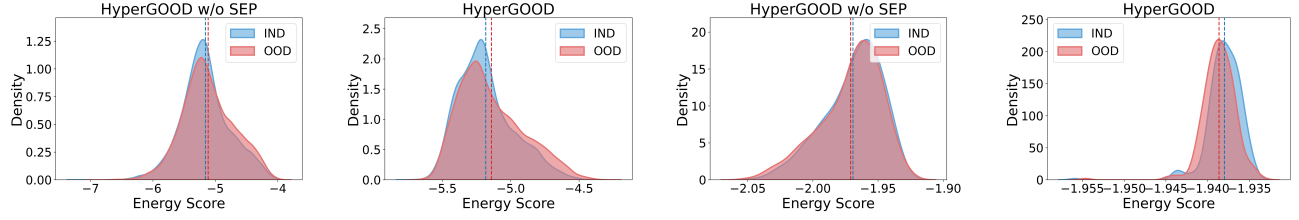


Figure 1: Visualization of energy score distributions used for HOOD detection on ModelNet40 (left) and Senate (right), under feature perturbation settings. (SEP: Structure-aware Energy Propagation)

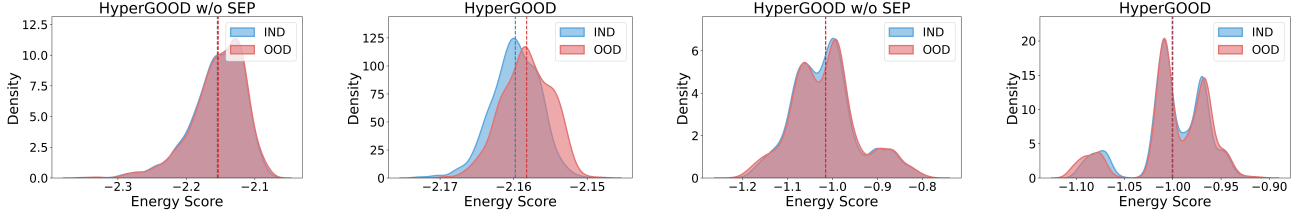


Figure 2: Visualization of energy score distributions used for HOOD detection on House (left), under label perturbation settings. On Mushroom (right), under feature perturbation settings.

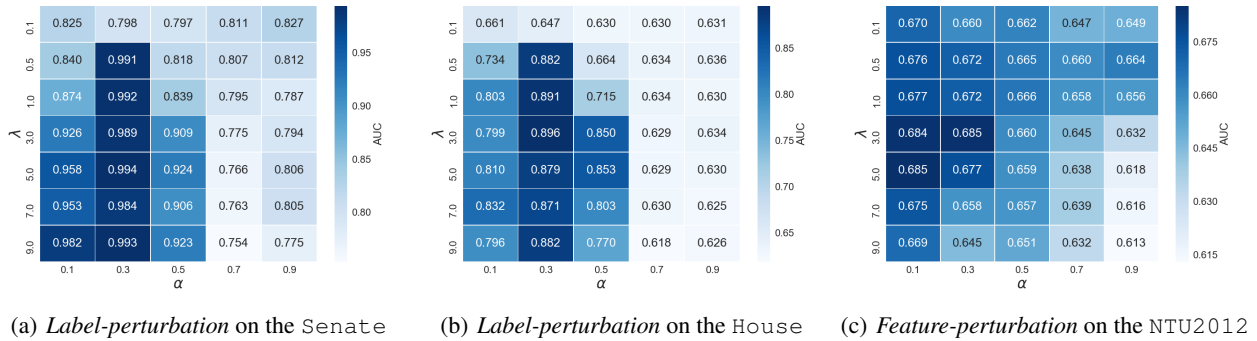


Figure 3: Sensitivity analysis of HyperGOOD with respect to λ and α .

A.2 Experimental Results

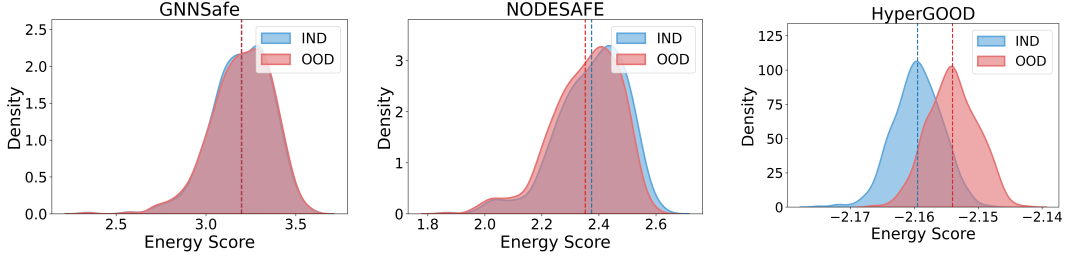


Figure 4: Visualization of energy score distributions used for HOOD detection on House with feature perturbation.

Appendix B: Theoretical Analysis

Proof of Lemma 1

Proof. We define the local deviation at step t for node v_i as:

$$\delta_i^{(t)} := s_i^{(t)} - \bar{s}_i^{(t)}, \quad \text{where } \bar{s}_i^{(t)} := \sum_j P_{ij} s_j^{(t)}.$$

Now, using the update rule:

$$s_i^{(t+1)} = \gamma s_i^{(t)} + (1 - \gamma) \bar{s}_i^{(t)},$$

we compute the new deviation:

$$\begin{aligned} \delta_i^{(t+1)} &= s_i^{(t+1)} - \sum_j P_{ij} s_j^{(t+1)} \\ &= \gamma s_i^{(t)} + (1 - \gamma) \bar{s}_i^{(t)} - \sum_j P_{ij} \left(\gamma s_j^{(t)} + (1 - \gamma) \bar{s}_j^{(t)} \right) \\ &= \gamma \left(s_i^{(t)} - \sum_j P_{ij} s_j^{(t)} \right) + (1 - \gamma) \left(\bar{s}_i^{(t)} - \sum_j P_{ij} \bar{s}_j^{(t)} \right). \end{aligned}$$

Now observe that: - $\sum_j P_{ij} s_j^{(t)} = \bar{s}_i^{(t)}$, by definition; - $\sum_j P_{ij} \bar{s}_j^{(t)}$ is a second-order average, and under mild assumptions on symmetry and stochasticity, $\bar{s}_i^{(t)} \approx \sum_j P_{ij} \bar{s}_j^{(t)}$, or in exact case (e.g., symmetric P), they cancel.

Thus, we have:

$$\delta_i^{(t+1)} \approx \gamma \cdot \delta_i^{(t)}.$$

Therefore:

$$|\delta_i^{(t+1)}| \approx \gamma \cdot |\delta_i^{(t)}| < |\delta_i^{(t)}|, \quad \text{since } \gamma \in (0, 1).$$

This proves that local deviation shrinks monotonically under propagation. \square

References

- Cai, T.; Jiang, Y.; Li, M.; Huang, C.; Wang, Y.; and Huang, Q. 2025. ML-GOOD: Towards multi-label graph out-of-distribution detection. In *AAAI*, 15, 15650–15658.
- Chen, D.-Y.; Tian, X.-P.; Shen, Y.-T.; and Ouhyoung, M. 2003. On visual similarity based 3D model retrieval. In *Computer Graphics Forum*, volume 22, 223–232.
- Chodrow, P. S.; Veldt, N.; and Benson, A. R. 2021. Generative hypergraph clustering: From blockmodels to modularity. *Science Advances*, 7(28): eabh1303.
- Dua, D.; Graff, C.; et al. 2017. UCI machine learning repository. URL: <http://archive.ics.uci.edu/ml>.
- Feng, Y.; You, H.; Zhang, Z.; Ji, R.; and Gao, Y. 2019. Hypergraph neural networks. In *AAAI*, 3558–3565.
- Fowler, J. H. 2006. Legislative cosponsorship networks in the US House and Senate. *Social Networks*, 28(4): 454–465.
- Hendrycks, D.; and Gimpel, K. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*.
- Khan, B.; Wu, J.; Yang, J.; and Ma, X. 2025. Heterogeneous hypergraph neural network for social recommendation using attention network. *ACM Transactions on Recommender Systems*, 3(3): 1–22.

- Kim, S.; Lee, S. Y.; Gao, Y.; Antelmi, A.; Polato, M.; and Shin, K. 2024. A survey on hypergraph neural networks: An in-depth and step-by-step guide. In *KDD*, 6534–6544.
- Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 7167–7177.
- Liang, S.; Li, Y.; and Srikant, R. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*.
- Liu, W.; Wang, X.; Owens, J.; and Li, Y. 2020. Energy-based out-of-distribution detection. In *NeurIPS*, 21464–21475.
- Wu, Q.; Chen, Y.; Yang, C.; and Yan, J. 2023. Energy-based out-of-distribution detection for graph neural networks. In *ICLR*.
- Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*, 1912–1920.
- Yadati, N.; Nimishakavi, M.; Yadav, P.; Nitin, V.; Louis, A.; and Talukdar, P. 2019. HyperGCN: A new method for training graph convolutional networks on hypergraphs. In *NeurIPS*, 1511–1522.
- Yang, S.; Liang, B.; Liu, A.; Gui, L.; Yao, X.; and Zhang, X. 2024. Bounded and uniform energy-based out-of-distribution detection for graphs. In *ICML*, 56216–56234.