Untying the Knots of Heterophily in Hypergraphs via Mixed-Curvature Manifolds

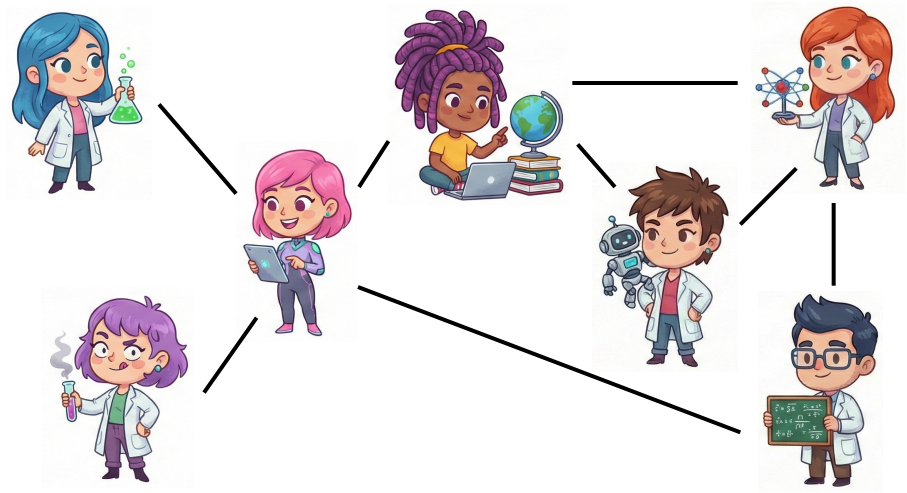
Lizhi Liu

China UnionPay

Hypergraph: Beyond Pairwise Interactions

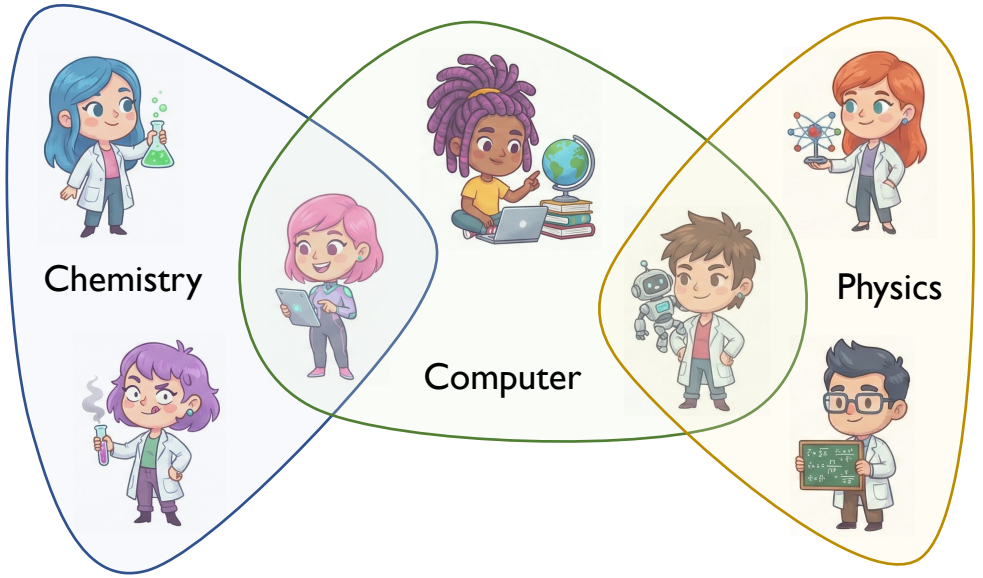
- In simple graphs, an edge connects exactly two nodes, limiting their capacity to model strictly pairwise relationships.
- Hypergraphs allow a single "hyperedge" to connect any number of nodes, capturing these high-order correlations (e.g. co-authorship groups) naturally.

Low-Order



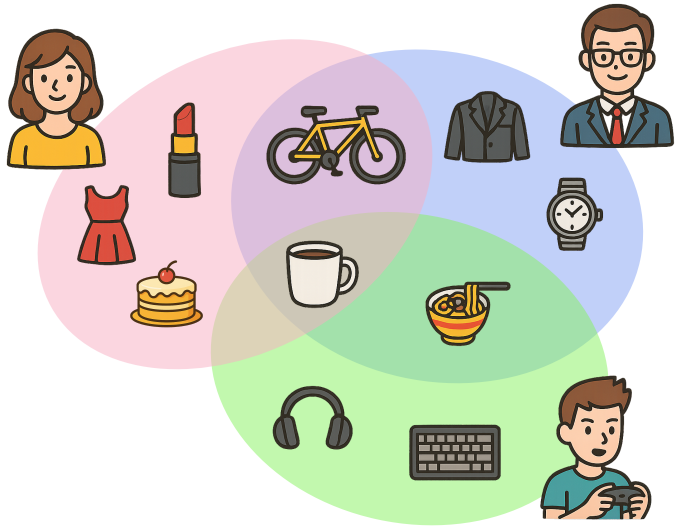
Graph

High-Order

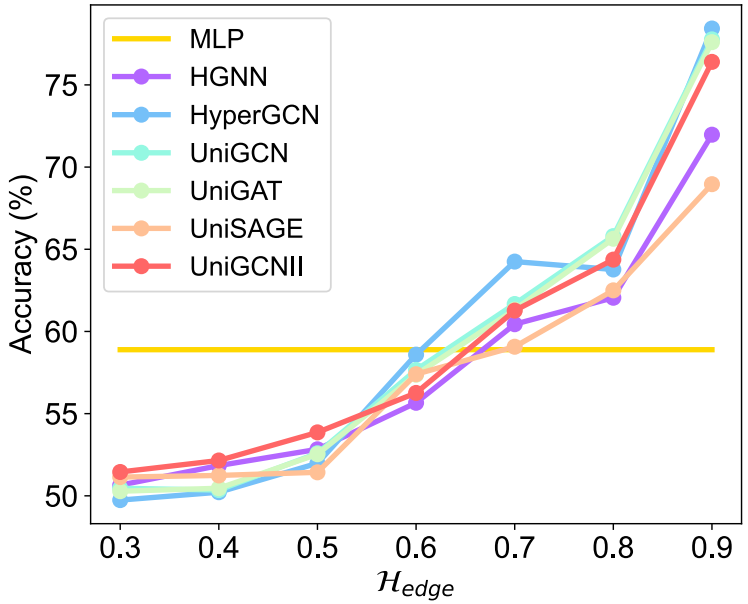


Hypergraph

The "Homophily Assumption" Trap



- Most Hypergraph Neural Networks (HNNs) assume **homophily**: connected nodes are expected to share the same label.
  - "Birds of a feather flock together."
- But reality is **heterophilic**: a single hyperedge connects nodes belonging to diverse classes.
  - A "Gamer" might buy *Coffee* and *Snacks* (unrelated categories).



- **Result:** Traditional HNNs perform worse than simple MLPs under high heterophily, revealing the fundamental limitations of classic message-passing architectures.

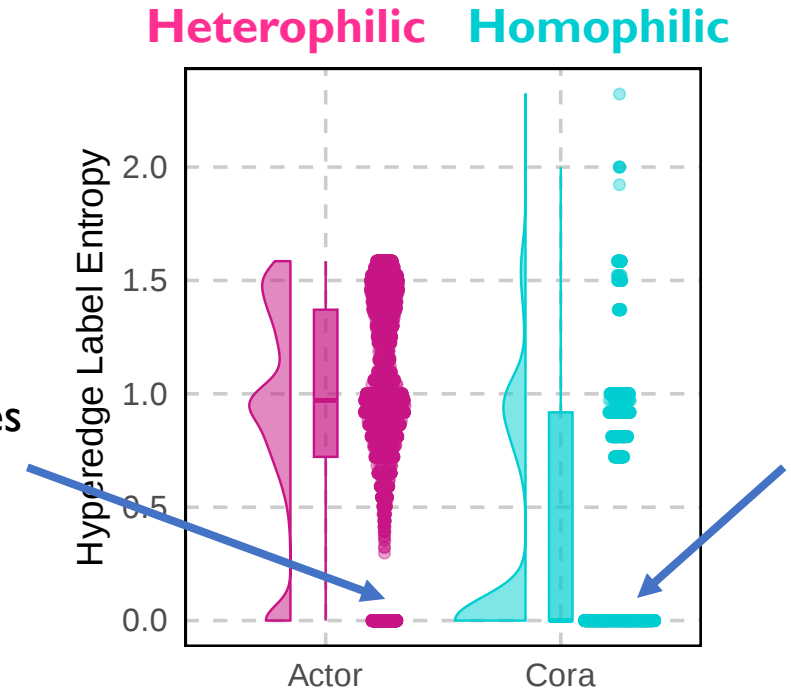
The Problem: Heterophily Mixing

Heterophily Mixing

In standard message passing, signals from neighbors of different classes are indiscriminately aggregated.

- Distinct class features mix together.
- Representations from different classes become indistinguishable (Over-smoothing).
- Discriminative power is eroded as depth increases.

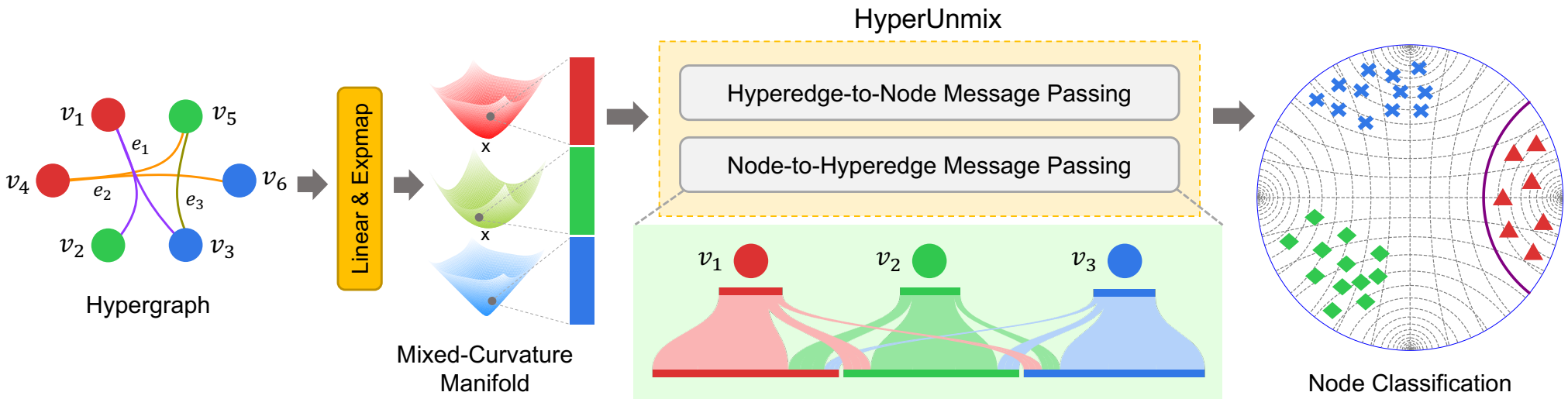
Only **14.7%** of hyperedges exhibit zero entropy
(Mixed / Hard)



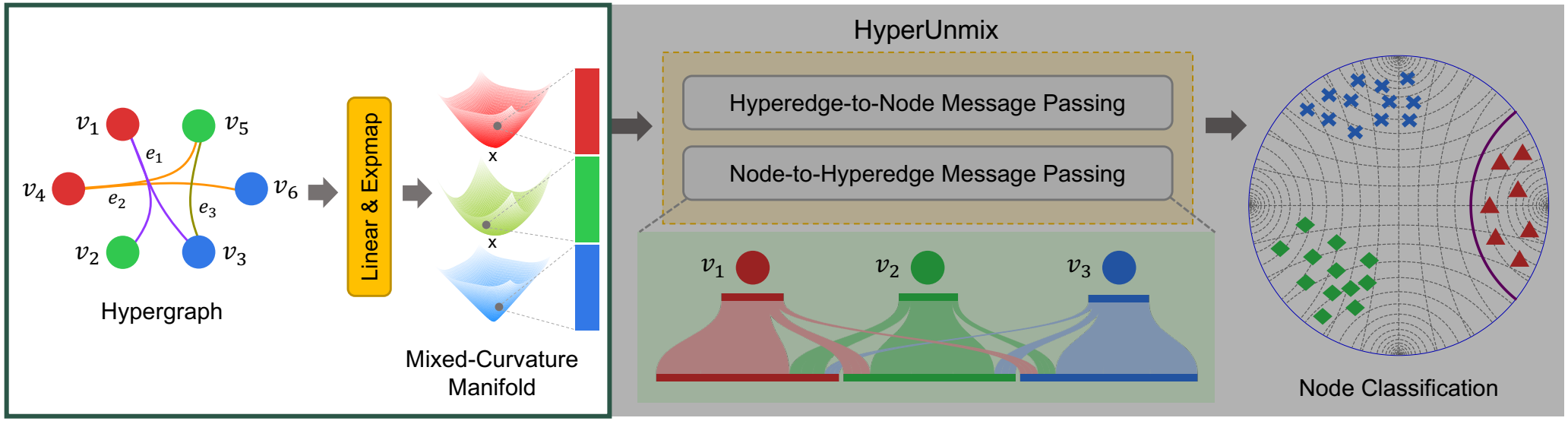
65.7% of hyperedges have zero entropy
(Pure / Easy)

HyperUnmix: Disentanglement via Mixed-Curvature Manifold

- **Key Idea:** Heterophily mixing can be alleviated if messages are constrained to propagate along class-specific semantic channels.
- **Assumption:** Nodes of different classes exhibit distinct distributional characteristics.
- **Solution:** We model the representation space as a Cartesian product of multiple Riemannian (hyperbolic) submanifolds, each associated with a particular class.
- **Mechanism:** By constraining the message flow to propagate primarily within class-aligned submanifolds, we prevent "bad" neighbors from polluting the representation.



Preprocessing: Projection & Decomposition



- Raw features (in Euclidean space) are projected via the Exponential Map and decomposed into C chunks (where C is the number of classes).
- Each chunk is aligned with a specific hyperbolic submanifold (Poincaré ball).

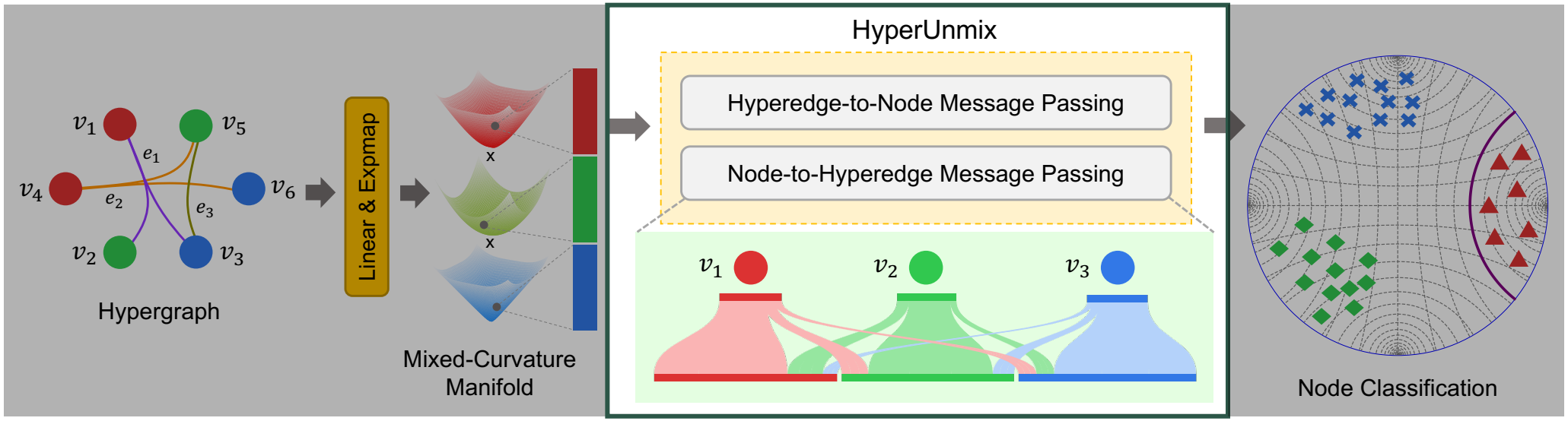
$$\mathbf{H}_n^{(0)} = \mathrm{Exp}_0^{\mathfrak{K}}\left(\mathbf{X}_n\mathbf{W}_n^{(0)} + \mathbf{b}_n^{(0)}\right), \qquad \mathbf{H}_e^{(0)} = \mathrm{Exp}_0^{\mathfrak{K}}\left(\mathbf{X}_e\mathbf{W}_e^{(0)} + \mathbf{b}_e^{(0)}\right)$$

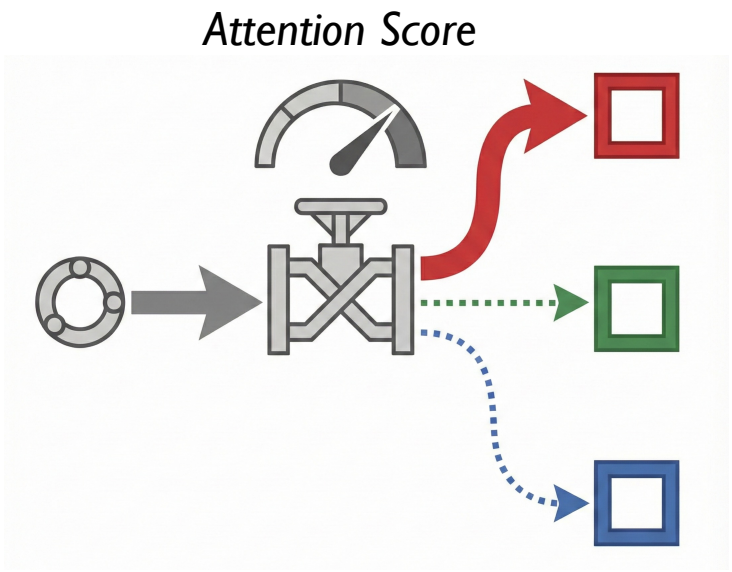
Node Feature *Hyperedge Feature*

Mixed-Curvature Product Manifold: $\mathbb{P} = \prod_{i=1}^{C} \mathbb{B}_{\kappa^{[i]}}^{d/C}$

Exponential Map: $\mathrm{Exp}_0^{\mathfrak{K}}(\mathbf{x}) = (\exp_{\mathbf{0}}^{\kappa^{[1]}}(\mathbf{x}^{[1]}), \cdots, \exp_{\mathbf{0}}^{\kappa^{[C]}}(\mathbf{x}^{[C]}))$

Methodology: Class-Aware Message Passing





- **Challenge:** We do not know the labels during training!
- **Solution:** Self-attention as a "soft-label" mechanism.

Methodology: Class-Aware Message Passing

Step 1: Assigning Attention Scores

- Calculate attention score between node v_j and hyperedge e_i:

$$\mathrm{S}_{n2e}^{(l)}[i,j,:] = \mathrm{Softmax}\left(\frac{\sigma\left(\mathbf{Z}_{e,n2e}^{(l)}[i,:] + \mathbf{Z}_{n,n2e}^{(l)}[j,:]\right)\mathbf{W}_{s,n2e}^{(l)}}{\sqrt{C}}\right)$$

➢ *Reflects the extent to which node v_j transmits information to the c-th submanifold of hyperedge e_i.*

Step 2: Class-Aware Message Aggregation

- Message (node $v_j \rightarrow$ hyperedge e_i, sent to the c-th submanifold):

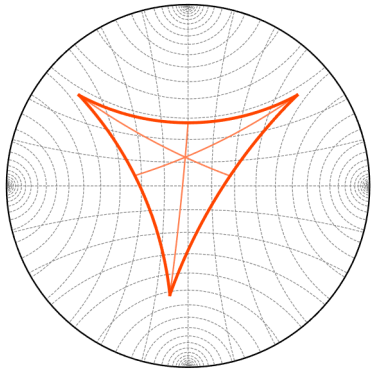
$$\mathrm{M}_{n2e}^{(l)}[i,j,c,:] = \mathrm{S}_{n2e}^{(l)}[i,j,c] \otimes_{\kappa^{[c]}} \exp_0^{\kappa^{[c]}}\left(\mathbf{Z}_{n,n2e}^{(l)}[j,:]\right)$$

- Aggregation (via Möbius gyromidpoint):

$$\mathrm{P}_e^{(l)}[i,c,:] = \frac{1}{2} \otimes_{\kappa^{[c]}} \left(\frac{\sum_{v_j \in e_i} \lambda^{\kappa^{[c]}}_{\mathrm{M}_{n2e}^{(l)}[i,j,c,:]} \mathrm{M}_{n2e}^{(l)}[i,j,c,:]}{\sum_{v_j \in e_i}\left(\lambda^{\kappa^{[c]}}_{\mathrm{M}_{n2e}^{(l)}[i,j,c,:]} - 1\right)}\right)$$

Möbius gyromidpoint

-- Weighted mean in hyperbolic space



Step 3: Residual Connections

- Adopt Möbius gyromidpoint to compute weighted average between initial feature and aggregated representation.

$$\mathbf{H}_e^{(l)}[i,:] = \frac{1}{2} \otimes_{\mathfrak{K}} \left(\frac{\alpha\lambda^{\mathfrak{K}}_{\mathbf{P}_e^{(l)}[i,:]}\mathbf{P}_e^{(l)}[i,:] + (1-\alpha)\lambda^{\mathfrak{K}}_{\mathbf{H}_e^{(0)}[i,:]}\mathbf{H}_e^{(0)}[i,:]}{\alpha\lambda^{\mathfrak{K}}_{\mathbf{P}_e^{(l)}[i,:]} + (1-\alpha)\lambda^{\mathfrak{K}}_{\mathbf{H}_e^{(0)}[i,:]} - 1}\right)$$

Note: Taking node-to-hyperedge message passing as an example, hyperedge-to-node propagation is the opposite.

Performance on Heterophilic Hypergraphs

Node Classification: Average Accuracy ± Standard Deviation

Method	Actor	Twitch-gamers	Pokec	Senate	House	*Avg. Rank*
MLP	85.45±1.21	52.77±1.81	56.92±2.46	52.25±5.17	51.86±2.34	8.6
HGNN	74.47±0.32	51.88±0.26	49.82±0.27	48.59±4.52	61.39±2.96	13
HyperGCN	68.67±4.38	51.32±1.02	52.43±3.68	42.45±3.67	48.32±2.93	14.6
HyperND	83.19±0.92	51.44±0.67	55.94±0.45	52.82±3.20	51.70±3.37	11.2
TF-HNN	85.96±0.41	52.39±0.60	57.65±1.08	58.31±7.59	67.55±1.33	5.8
UniGCNII	80.48±1.13	50.84±0.76	54.25±2.70	49.30±4.25	67.25±2.57	12.4
AllDeepSets	82.00±2.33	50.72±0.96	51.11±1.04	48.17±5.67	67.82±2.40	13
AllSetTransformer	83.39±1.73	50.45±0.76	58.40±0.42	51.83±5.22	69.33±2.20	10
HyperGT	62.43±0.55	52.10±0.40	57.55±0.54	48.14±5.15	46.93±2.60	12.8
Hypergraph-MLP	84.72±0.54	51.95±0.53	59.63±0.28	68.00±4.20	71.24±2.16	5.2
KHGNN	62.31±0.42	50.91±0.40	52.32±0.62	58.14±3.38	55.40±2.25	13
EHNN	86.21±0.49	52.14±0.76	58.23±1.07	53.80±7.26	65.45±2.21	6.8
T-HyperGNN	85.32±0.48	51.82±0.38	58.82±0.49	56.06±3.92	64.58±2.42	8
SheafHyperGNN	80.09±2.45	51.03±0.76	55.34±4.39	68.73±4.68	73.84±2.30	8.2
ED-HNN	85.77±0.46	50.86±0.88	59.11±0.57	64.79±5.14	72.45±2.28	6.4
HyperUFG	89.32±0.75	52.35±0.04	62.30±0.12	67.61±7.00	72.82±2.22	3
HyperUnmix	**90.57**±0.38	**53.94**±0.41	**62.98**±0.57	**70.29**±6.05	**73.91**±2.03	1

- Ranked **#1** across all 5 heterophilic benchmarks.
- HyperUnmix is the **ONLY** model to outperform the MLP baseline on Twitch-gamers.

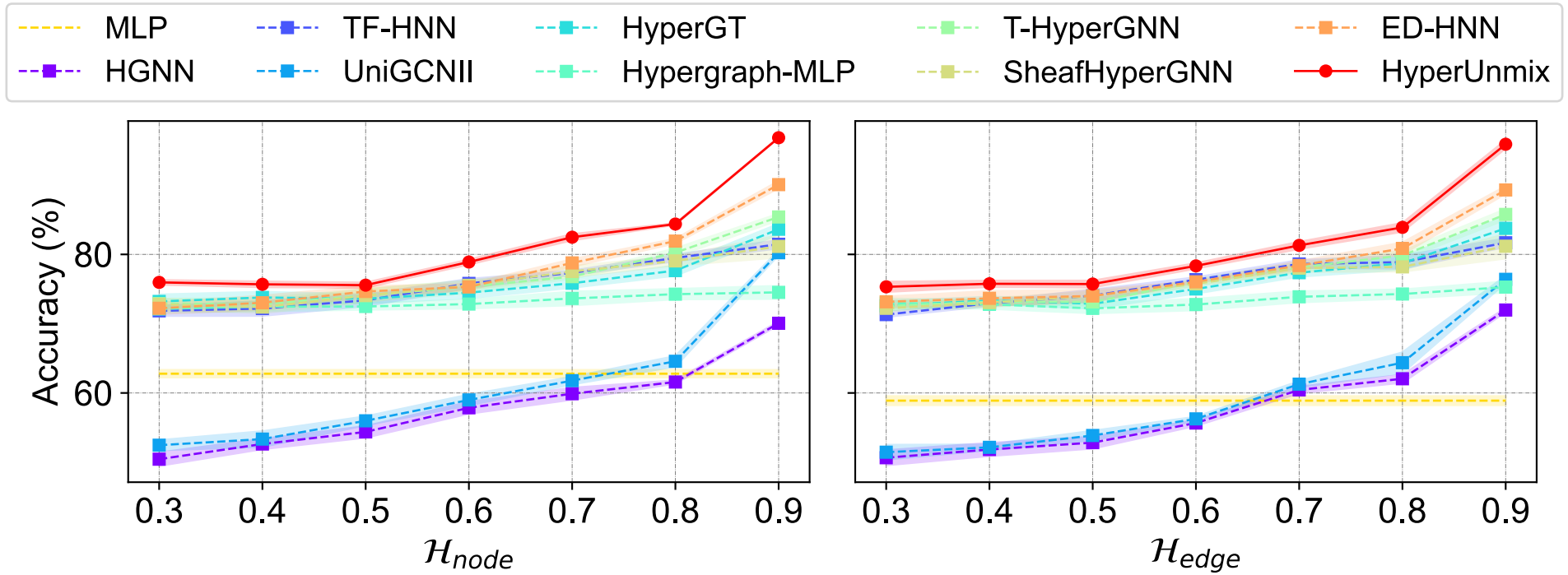
Performance on Homophilic Hypergraphs

Node Classification: Average Accuracy ± Standard Deviation

Method	Cora	Citeseer	Pubmed	Cora-CA	DBLP-CA	*Avg. Rank*
MLP	75.16±1.41	71.71±1.01	87.20±0.34	75.17±1.41	84.37±0.33	14.4
HGNN	79.39±1.36	72.45±1.16	86.44±0.44	82.64±1.65	91.03±0.20	10.2
HyperGCN	78.45±1.26	71.28±0.82	82.84±8.67	79.48±2.08	89.38±0.25	14
HyperND	79.20±1.14	72.62±1.49	86.68±0.43	80.62±1.32	90.35±0.26	11
TF-HNN	79.47±1.31	73.00±1.27	87.90±0.37	84.19±0.89	91.38±0.24	7.2
UniGCNII	78.81±1.05	73.05±2.21	88.25±0.40	83.60±1.14	91.69±0.19	7.2
AllDeepSets	76.88±1.80	70.83±1.63	88.75±0.33	81.97±1.50	91.27±0.27	10
AllSetTransformer	78.58±1.47	73.08±1.20	88.72±0.37	83.63±1.47	91.53±0.23	7.2
HyperGT	75.16±1.32	72.33±0.91	80.31±0.70	80.19±2.17	90.20±0.24	14
Hypergraph-MLP	79.80±1.82	73.90±1.57	87.89±0.55	78.57±1.12	90.29±0.26	9.4
KHGNN	80.67±0.76	<u>74.80±1.10</u>	88.47±0.47	84.25±0.74	87.34±0.92	6.2
EHNN	76.51±1.52	68.67±0.75	87.12±0.31	81.68±0.81	90.47±0.43	12.8
T-HyperGNN	74.20±1.37	71.21±0.87	86.28±0.62	75.01±1.44	85.44±0.14	16
SheafHyperGNN	81.30±1.70	74.71±1.23	87.68±0.60	<u>85.52±1.28</u>	91.59±0.24	4.8
ED-HNN	80.31±1.35	73.70±1.38	**89.03±0.53**	83.97±1.55	<u>91.90±0.19</u>	4
HyperUFG	<u>81.51±0.99</u>	74.72±2.10	88.73±0.42	85.18±0.69	91.67±0.31	3.2
HyperUnmix	**84.34**±1.55	**75.96**±1.19	<u>88.93±0.38</u>	**87.00**±1.25	**92.56**±0.45	1.2

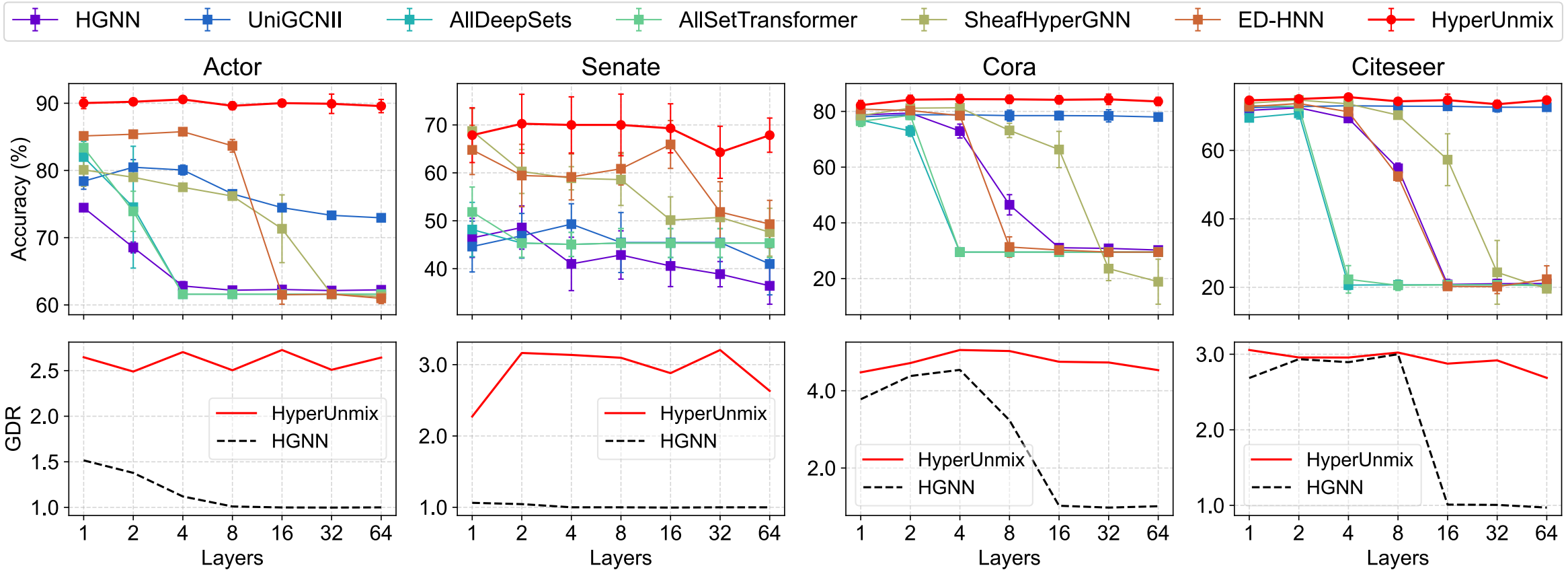
- HyperUnmix consistently ranks among the top-performing methods on most datasets.
- The convolutional design tailored for heterophily generalizes well to homophilic cases.

Results on Synthetic Heterophilic Datasets



- HyperUnmix consistently outperforms **ALL** baselines across **DIFFERENT** homophily levels, demonstrating the adaptability and effectiveness in handling diverse neighborhood contexts.

Mitigating Over-Smoothing



- HyperUnmix exhibits remarkably flat accuracy curves on **BOTH** homophilic and heterophilic hypergraphs as depth increases.

Summary & Limitations

Summary

- **Challenge: HNNs Fail under Heterophily**
  - The performance of classic HNNs drops sharply on heterophilic graphs, where connected nodes often belong to different classes.
- **Core Issue: Heterophily Mixing**
  - Messages from dissimilar classes become entangled during aggregation, diluting class-discriminative information.
- **Our Solution: HyperUnmix Routes Messages by Class**
  - We assume that nodes of different classes lie on distinct geometric manifolds, and thus model the representation space as a mixed-curvature product manifold.
  - During message passing, information is constrained to flow within class-specific submanifolds, preventing harmful signal entanglement.

Limitations

- **Scalability**: HyperUnmix becomes difficult to apply when dealing with hypergraphs with large class sets.
- **Efficiency**: Hyperbolic geometry operations have a relatively high computational cost, especially for large-scale hypergraphs.

Thank you!

GitHub

Homepage