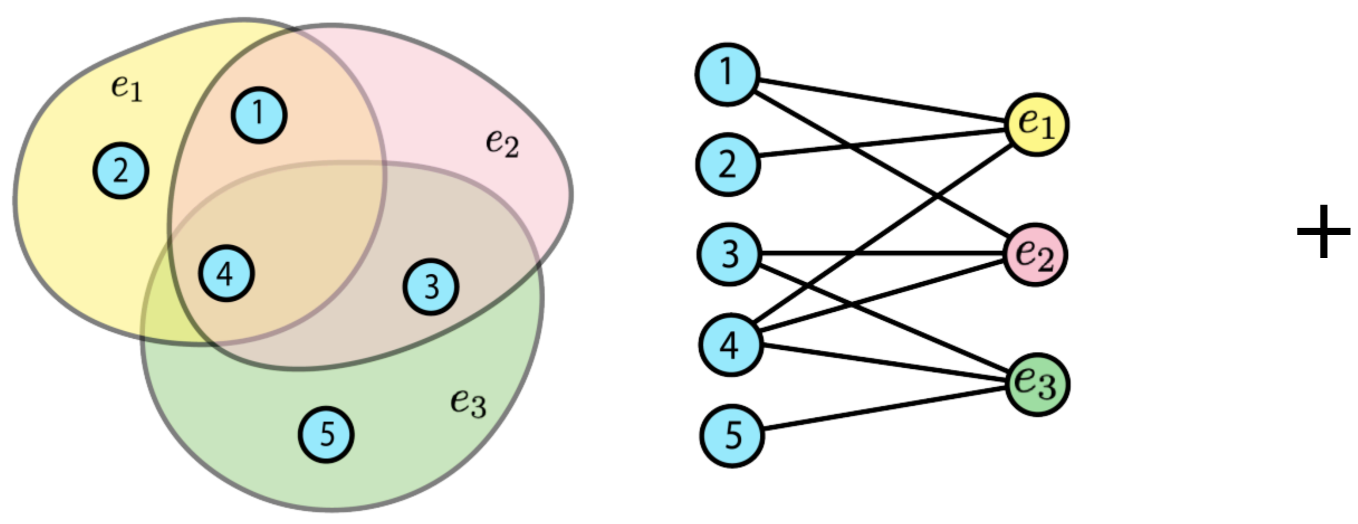
Higher-Order Expander Graph Propagation

Thomas Christie*, Yu He*,[1]

* Equal Contribution. Work done at University of Cambridge. [1] Stanford University

TLDR: We propose a framework to construct **bipartite expanders** that capture **higher-order interactions** while leveraging **expander properties**, in order to mitigate the **over-squashing** problem for GNNs.

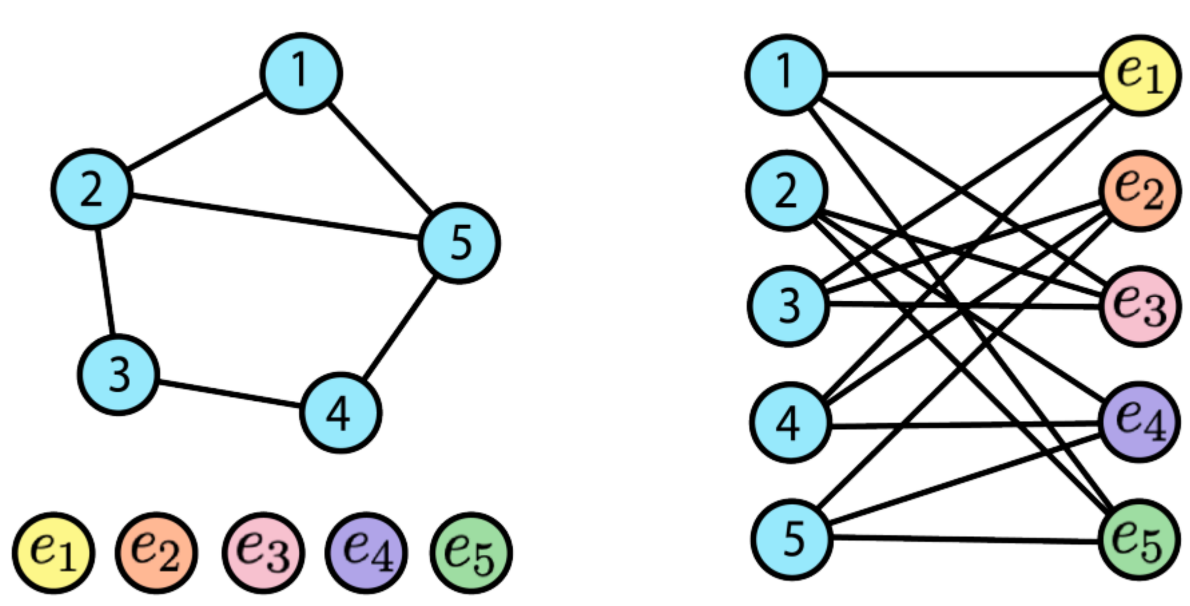
Hypergraphs as bipartite graphs



A hypergraph (left) can be represented as a bipartite graph (right), where nodes are at the left-hand side and hyperedges at the right-hand side.

Bipartite expanders to capture higher-order interactions.

Higher-Order Expander Graph Propagation



Original graph (with hyperedge nodes)

Bipartite expander

Construction of bipartite expanders:

(i) Perfect matchings

A matching on a graph is defined as a set of edges without common vertices, and a perfect matching is a matching which contains all vertices of the graph.

We construct bipartite expanders by taking the union of k disjoint perfect matchings, making them k-regular.

(ii) Ramanujan condition

A k-regular graph G is said to be Ramanujan if it satisfies the property $\lambda(\mathcal{G}) \leq 2\sqrt{k-1}$. Here, $\lambda(\mathcal{G})$ is the largest magnitude non-trivial eigenvalue.

We additionally impose Ramanujan condition that gives low diameters and high expander constants.

Message passing framework:

1. Augment the original graph with hyperedge nodes.
2. Construct bipartite expanders using perfect matchings or Ramanujan bipartite graphs.
3. Perform message-passing on the original graph.
4. Perform bi-directional message-passing on the bipartite expander graph.
5. Interleave two message-passing layers, with the original graph as the first and last layers.

Expander graphs

A k-regular graph $G = (V, E)$ is said to be a c-expander graph if

$$\frac{|\partial_{out}(\mathcal{A})|}{|\mathcal{A}|} \geq c$$

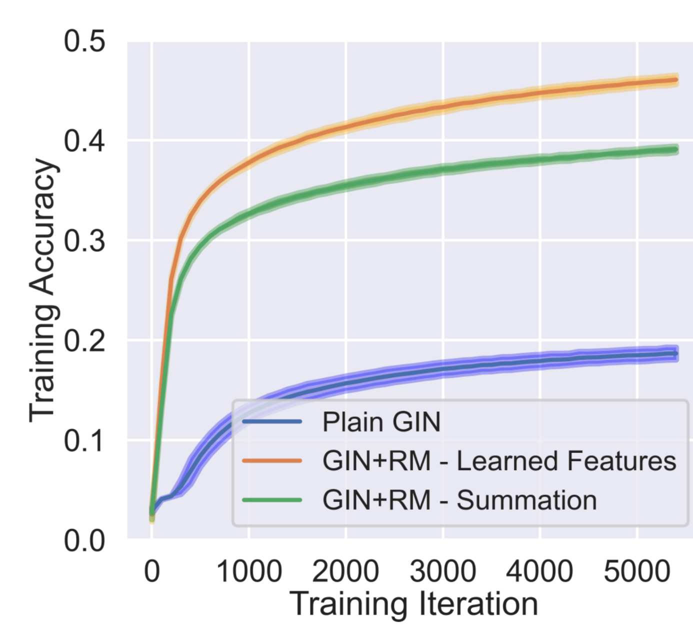
for all subsets $\mathcal{A} \subset \mathcal{V}$ with $|\mathcal{A}| \leq \frac{|\mathcal{V}|}{2}$.

Properties: highly connected, sparse graph, low diameter

Previous works [1, 2, 3] apply expander graphs in GNNs to overcome the **over-squashing problem** - where information from an exponential number of neighbors gets compressed into a fixed-size vector, leading to potential information loss .

Experimental results

(i) Tree Neighbors Match



(ii) OGBG-molhiv

Model	Test ROC-AUC
Plain GIN [40]	0.7558 ± 0.0140
EGP [20]	0.7934 ± 0.0035
GIN+PM+Learned Features	0.7742 ± 0.0104
GIN+PM+Summation	0.7751 ± 0.0138
GIN+RM+Learned Features	0.7628 ± 0.0132
GIN+RM+Summation	0.7737 ± 0.0138

Mean ± STD test ROC-AUC score. Best, Second Best and Third Best results are colored.

To deal with the hyperedge node features, we propose two methods: learn the features end-to-end (Learned Features) or perform summation during left-to-right message passing on the bipartite expander (Summation).

(iii) OGBG-code2

Model	Test F1 Score
Plain GIN [40]	0.1495 ± 0.0023
EGP [20]	0.1497 ± 0.0015
GIN + 3-Regular Bipartite Expander + Learned Features	0.1519 ± 0.0020
GIN + 3-Regular Bipartite Expander + Summation	0.1254 ± 0.0029

Mean ± STD test F1 score. Best, Second Best and Third Best results are colored.

We compare our model with GIN [4] and EGP [1], aggregating the results over 10 seeds with the same setup.

Conclusion & Future work

- We show bipartite expanders can help to alleviate over-squashing problem in GNNs by additionally capturing higher-order interactions.
- Datasets: long-range dependencies.
- Bipartite expanders: explicit construction methods.
- Bipartite message passing: hypergraph neural networks.

Bibliography

[1] Andreea Deac, Marc Lackenby, and Petar Veličković. *Expander graph propagation,* 2022.
[2] Pradeep Kr. Banerjee, Kedar Karhadkar, Yu Guang Wang, Uri Alon, and Guido Montúfar. *Oversquashing in gnns through the lens of information contraction and graph expansion,* 2022.
[3] Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J. Sutherland, Ali Kemal Sinop. *Exphormer: Sparse Transformers for Graphs,* 2023.
[4] Keyulu Xu, Weihua Hu, Jure Leskovec, Stefanie Jegelka. *How powerful are Graph Neural Networks?* 2018.