Higher-Order Expander Graph Propagation

Thomas Christie *
Unaffiliated
thwc3@cantab.ac.uk

Yu He *
Stanford University
heyu@cs.stanford.edu

Abstract

Graph neural networks operate on graph-structured data via exchanging messages along edges. One limitation of this message passing paradigm is the over-squashing problem. Over-squashing occurs when messages from a node's expanded receptive field are compressed into fixed-size vectors, potentially causing information loss. To address this issue, recent works have explored using expander graphs, which are highly-connected sparse graphs with low diameters, to perform message passing. However, current methods on expander graph propagation only consider pair-wise interactions, ignoring higher-order structures in complex data. To explore the benefits of capturing these higher-order correlations while still leveraging expander graphs, we introduce *higher-order expander graph propagation*. We propose two methods for constructing bipartite expanders and evaluate their performance on both synthetic and real-world datasets.

1 Introduction

Graph neural networks (GNNs) [1–3] have gained significant attention for their effective applications across various domains [4–6]. They operate directly on graph-structured data, utilising the inherent symmetries of graphs [7]. GNNs commonly employ the message-passing paradigm [3, 8], where messages are exchanged along the edges of a graph. However, this approach faces some fundamental challenges, including limited expressivity [9, 10], over-smoothing [11] and over-squashing [12].

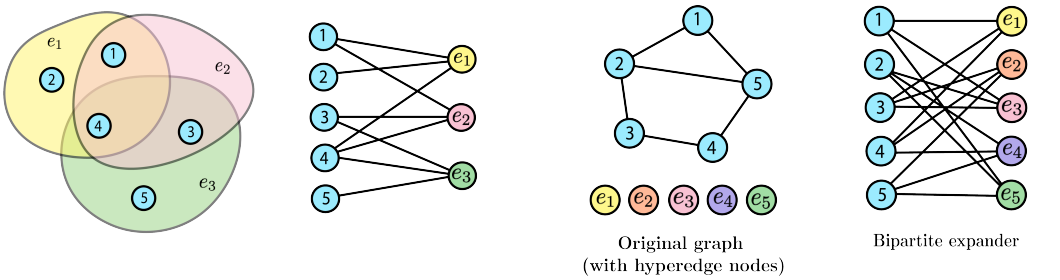
In a multi-layered GNN architecture, a node can aggregate information from neighbouring nodes within a certain radius determined by the number of layers, denoted as k. The over-squashing problem [12] becomes apparent as k increases. In such cases, nodes are compelled to compress information from an exponentially growing number of neighbouring nodes into their fixed-size feature vectors. This compression process can lead to the loss of important information, particularly in long-range interactions between nodes. Addressing this over-squashing issue is crucial for enhancing GNNs' expressivity [13], especially in tasks that require long-range interactions in order to be solved.

Graph rewiring is a technique which modifies the edges within the original graph to facilitate the exchange of messages between distant nodes and counteract the over-squashing issue. This approach has gained attention in recent research efforts [14–19]. More recently, researchers have explored the use of expander graphs as a solution to address the over-squashing problem [20, 21]. In this approach, message passing takes place alternately on the original graph and an expander graph. Expander graphs offer several advantageous properties for information propagation, such as sparsity, high connectivity, and logarithmic diameters, which enable efficient signal propagation across the graphs with minimal message passing steps.

On the other hand, hypergraphs provide a representation for capturing higher-order interactions within complex data [22–24]. While previous work [20] has utilised expander graphs in the context of pair-wise interactions, we aim to investigate the potential benefits of introducing higher-order interactions within expander graphs.

*Equal contribution. Work done when both authors were at the University of Cambridge.

NeurIPS 2023 New Frontiers in Graph Learning Workshop (NeurIPS GLFrontiers 2023).



(a) Hypergraphs as bipartite graphs

(b) Higher-Order Expander Graph Propagation

Figure 1: (a) Given a hypergraph with edges between sets of nodes of arbitrary cardinality, we can construct a corresponding bipartite representation, with one set of nodes corresponding to the original nodes in the hypergraph, and the other set representing hyperedges. (b) We first augment the input graph with disconnected hyperedge nodes. Then, we construct bipartite expanders where input graph nodes are on the left-hand side, and hyperedge nodes on the right-hand side. We perform message passing on the original graph and bipartite expander alternately. Message passing on the bipartite expander first goes from graph nodes to hyperedge nodes, and then back to graph nodes.

Our approach involves exchanging messages on bipartite expanders that represent hypergraphs. We construct random bipartite expanders using two algorithms: one based on perfect matchings, and the other using Ramanujan graphs. We demonstrate the effectiveness of our models by evaluating them on both synthetic and real-world data.

2 Related Work

Approaches to over-squashing Methods for addressing over-squashing in graph neural networks fall into two categories: spatial and spectral [25]. Spatial methods focus on reducing the distance between distant nodes, achieved through methods like adding explicit edges [14, 15], leveraging higher-order structures [16, 17], or reweighting edges with attention mechanisms [26]. On the other hand, spectral methods aim to increase the graph's Cheeger constant, measuring its "bottleneckness", such as differentiable rewiring for Lovász bound optimization [18], first-order spectral rewiring for spectral gap optimization [19], and the use of expander graphs [20, 21].

Expanders Expander graphs have gained attention as a means to mitigate the over-squashing problem in graph neural networks, primarily due to their favorable spectral properties. In recent research, 4-regular Cayley graphs have been utilised as templates for interleaving message passing on the input graph and the expander graph [20]. Concurrently, another approach utilises a random local edge flip algorithm based on expander graph construction [21]. Our work builds upon this existing research by focusing on the capture of higher-order interactions between nodes using bipartite expander graphs.

3 Theoretical Background

3.1 Hypergraphs

Standard graphs, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consist of a set of *vertices* $\mathcal{V}$ and pairwise *edges* between them: $\mathcal{E} \subseteq \{(u, v) | (u, v) \in \mathcal{V}^2\}$ [27]. Graphs are said to be *k-regular* if every vertex is of degree k. Hypergraphs $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ are a generalisation of graphs, whereby edges may occur between *sets* of nodes, which may be of arbitrary size i.e. $\mathcal{E} \subseteq \{\mathcal{X} | \mathcal{X} \subseteq \mathcal{V}\}$ [28]. A hypergraph is said to be *k-uniform* if each hyperedge is of cardinality k. An example can be seen in Figure 1(a).

Bipartite graphs, $\mathcal{B} = (\mathcal{L}, \mathcal{R}, \mathcal{E})$, are graphs whose vertices can be separated into two disjoint sets $\mathcal{L}$ and $\mathcal{R}$, with no edges between nodes within a single set. Hypergraphs may be represented as bipartite graphs: one set of nodes in the bipartite graph, say $\mathcal{L}$, corresponds to the original nodes in the hypergraph, and the other set of nodes in the bipartite graph, $\mathcal{R}$, represents the hyperedges. Edges in the bipartite graph (l, r) are formed if node l in the hypergraph belongs to hyperedge r. The

bipartite graph representation is particularly useful in graph representation learning, as it enables one to utilise the array of tools developed for standard graphs on hypergraphs with minimal adaptation.

3.2 Expander Graphs

Expander graphs have recently been identified as a promising approach to alleviate the over-squashing problem [20]. Expander graphs are *sparse* graphs ($|\mathcal{E}| = O(|\mathcal{V}|)$) which have a low *diameter*. Therefore, performing message passing on expander graphs enables messages to be propagated between any pair of nodes in a low number of hops, alleviating the issue of over-squashing.

We define an expander graph more formally as follows. Firstly, for any subset of vertices $\mathcal{A}$ from the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the *outer boundary* of $\mathcal{A}$, denoted $\partial_{\text{out}}(\mathcal{A})$, consists of the set of nodes adjacent to nodes in $\mathcal{A}$, which themselves don't belong to $\mathcal{A}$. Then, an expander graph is defined as follows [29]:

Definition 1 (Regular Expander Graphs) *A k-regular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be a c-expander graph if*

$$\frac{|\partial_{out}(\mathcal{A})|}{|\mathcal{A}|} \geq c \tag{1}$$

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

There are various known methods for constructing expander graphs ([30], [31]). Several of these constructions are algebraic, and utilise Cayley graphs [32] for deterministic construction. However, other construction methods rely on a family of graphs known as *Ramanujan graphs*, whose members have spectral properties that make them excellent candidates for expander graphs.

For an undirected graph with n vertices, it is known that its adjacency matrix has n real-valued eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. If the graph is k-regular, then the eigenvalues satisfy $k = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_n \geq -k$, and if the graph is also bipartite then $\lambda_n = -k$. Eigenvalues $\lambda_i \neq \pm k$ are referred to as *non-trivial* eigenvalues. The largest magnitude non-trivial eigenvalue is denoted $\lambda(\mathcal{G}) = \max_{|\lambda_i|<k} |\lambda_i|$, which leads to Ramanujan graphs being defined as follows [33]:

Definition 2 (Ramanujan Graphs) *A k-regular graph $\mathcal{G}$ is said to be Ramanujan if it satisfies the property $\lambda(\mathcal{G}) \leq 2\sqrt{k-1}$.*

We shall see that this property leads to Ramanujan graphs having low diameters, which follows from several results connecting the spectral properties of graphs with their diameters. The first important result was derived by Chung [34] (and was further refined by [35]), which presents a bound on the diameter of graph $\mathcal{G}$ in terms of its largest magnitude non-trivial eigenvalue $\lambda(\mathcal{G})$:

Theorem 1 *For a connected k-regular graph $\mathcal{G}$ with n vertices, its diameter is bounded by:*

$$\alpha + \frac{\log\left(\frac{2n}{\alpha}\right)}{log\left(\frac{k+\sqrt{k^2-\lambda(\mathcal{G})^2}}{\lambda(\mathcal{G})}\right)} \tag{2}$$

with $\alpha = 2$ in the case of bipartite graphs and 1 otherwise.

Hence, it follows that in order to minimise the diameter of the graph, we must minimise $\lambda(\mathcal{G})$. The Alon-Boppana bound [36] gives an asymptotic lower bound on $\lambda(\mathcal{G})$:

Theorem 2 (Alon-Boppana Bound) *All sufficiently large k-regular graphs $\mathcal{G}$ satisfy:*

$$\lambda(\mathcal{G}) \geq 2\sqrt{k-1} - o(1) \tag{3}$$

with the asymptotic behaviour in the $o(1)$ term coming from the number of nodes in the graph n going to infinity: $n \to \infty$.

Therefore, asymptotically, Ramanujan graphs have the smallest possible value of $\lambda(\mathcal{G})$, and so have asymptotically minimal diameters. This property can further be linked with the *expander constant* of the resulting graph, c, defined in Definition 1. It has been shown [29] that:

$$\frac{|\partial_{\text{out}}(\mathcal{A})|}{|\mathcal{A}|} \geq (k - \lambda_2(\mathcal{G}))\frac{|\mathcal{V} \setminus \mathcal{A}|}{|\mathcal{V}|} \tag{4}$$

It follows from Definition 1 that the resulting expander constant is $(k - \lambda_2(\mathcal{G}))/2$. In order to maximise this, and hence generate expander graphs with good expansion properties, it follows that we should minimise $\lambda_2(\mathcal{G})$. Since we have $\lambda_2(\mathcal{G}) \leq \lambda(\mathcal{G})$, Ramanujan graphs also asymptotically minimise $\lambda_2(\mathcal{G})$. Therefore, it is clear that Ramanujan graphs are excellent expanders with low diameters and high expander constants.

Furthermore, even k-regular graphs make good expanders. An alternative definition of expander graphs is given in terms of their *edge expansion*, $h(\mathcal{G})$. Given a subset of vertices $\mathcal{A} \subseteq \mathcal{V}$, the *edge boundary* of $\mathcal{A}$, denoted $\partial\mathcal{A}$, is the set of edges with one end in $\mathcal{A}$, and the other end outside of $\mathcal{A}$. Then, an alternative definition of an expander graph is the following [33]:

Definition 3 (Regular Expander Graphs - Edge Expansion Definition) *A k-regular graph of n vertices is an (n, k, δ)-expander if its edge expansion, $h(\mathcal{G})$ satisfies the following inequality:*

$$h(\mathcal{G}) = \min_{\mathcal{A} \subset \mathcal{V}: |\mathcal{A}| \leq n/2} \frac{|\partial\mathcal{A}|}{|\mathcal{A}|} \geq \delta \tag{5}$$

A theorem from Dodziuk [37] gives us a lower bound on δ for k-regular graphs:

Theorem 3 *If graph $\mathcal{G}$ is k-regular then:*

$$\frac{k - \lambda(\mathcal{G})}{2} \leq h(\mathcal{G}) \leq \sqrt{2k(k - \lambda(\mathcal{G}))} \tag{6}$$

4 Higher-Order Expander Graph Propagation

4.1 Bipartite Expanders

We rely on the fact that hypergraphs can be represented as bipartite graphs, and construct *bipartite expander graphs* to capture higher-order interactions and leverage expander graph properties.

Perfect matchings A *matching* on a graph is defined as a set of edges without common vertices [38], and a *perfect matching* is a matching which contains all vertices of the graph. One approach to constructing bipartite expander graphs is taking the union of k perfect matchings [39]. By ensuring that the k matchings are disjoint (i.e. contain no common edges), we guarantee that the resulting bipartite graph will be k-regular. As detailed in section 3.2, k-regular graphs have good expansion properties, and so k-regular random bipartite graphs are good candidates for expander graphs.

Random Ramanujan bipartite graphs In addition to generating random k-regular graphs, due to the utility of Ramanujan graphs as expanders, we also added the ability to check that the resulting graphs satisfied the Ramanujan property, given in definition 2. This incurs $O(|\mathcal{V}|^3)$ time complexity due to the need to calculate the eigenvalues of the graph's adjacency matrix.

4.2 Framework

As illustrated in Figure 1(b), given an input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we first augment it with $|\mathcal{V}|$ hyperedge nodes, call it the set $\mathcal{H}$. We construct a bipartite expander $\mathcal{B} = (\mathcal{L}, \mathcal{R}, \mathcal{E})$ where $\mathcal{L} = \mathcal{V}$ and $\mathcal{R} = \mathcal{H}$. Next, we connect nodes between the two sides by constructing a set of edges $\mathcal{E}$, either using perfect matchings or Ramanujan graphs as explained in section 4.1. In this way, we have a k-regular bipartite expander, where we set k as a hyperparameter.

To incorporate the expander graph into the model without losing the topology of the original graph, we follow [20] by interleaving message passing on the two graphs. Therefore, we perform message passing on the original graph in odd layers, and on the expander graph in even layers.

Lastly, we ignore all hyperedge node features when performing graph pooling, which means only node representations from the input graph are used to predict the final graph classification.

4.3 Message passing on bipartite expanders

We perform message passing in two directions sequentially, first from original graph nodes to hyperedge nodes, and then back, as shown in Figure 1(b). This allows each hyperedge node to serve as a communication hub for k graph nodes, enabling *higher-order* message passing which goes beyond pair-wise interactions. In practice, the hyperedge node features are initialised with 0s. During message passing, we experiment with two handling methods. One allows hyperedge node features to be learnt end-to-end, whilst the other simply aggregates the messages from the original graph nodes at the hyperedge nodes via summation followed by a linear layer.

5 Evaluation

5.1 Models

GIN We used Graph Isomorphism Network (GIN) [9] as a baseline. Given $h_v^{(l)}$ as the representation for node v at layer l, $\mathcal{N}(v)$ as the neighbours for node v, and ϵ as a learnable parameter, a GIN convolutional layer can be formulated as $h_v^{(l)} = \text{MLP}^{(l)}\left((1+\epsilon^{(l)}) \cdot h_v^{(l-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(l-1)}\right)$. We perform graph-level pooling by averaging the node representations after the final GIN layer, and then use a task-specific loss function to evaluate the graph classification results.

Our models We use GIN to perform message-passing on the augmented bipartite expander graph. We experiment with two bipartite expander construction methods, one based on perfect matchings (**GIN+PM**), and the other one additionally imposes the Ramanujan condition (**GIN+RM**). Furthermore, we test two ways of handling the arbitrary hyperedge node features, depending on whether they are learned (**learned features**) or aggregated via summation (**summation**).

5.2 Set-up

We follow the same hyperparameter set-up as [40] for graph-property prediction tasks. For a fair comparison, we use the same number of layers for all models. Note that we treat message passing in two directions on the bipartite graph as one layer, because in principle nodes in the original graph only get updated once.

6 Results

6.1 Tree-NeighborsMatch

We ran experiments on the synthetic Tree-NeighborsMatch dataset [12] in the same manner as in the original expander graph propagation paper [20]. Namely, we ran the experiment on binary trees of `depth=5` with 6 GNN layers, at which depth standard GNN implementations begin to suffer from over-squashing. We ran experiments both without expander graphs (Plain GIN) and with a Ramanujan bipartite expander (GIN+RM), with two different methods of dealing with hyperedge node features. The regularity of bipartite expanders is set to $k = 3$. Experiments were run 3 times for 5400 epochs. The results presented in Figure 2 demonstrate clearly that interleaving message passing on the original graph and the bipartite expander graph helps to mitigate the problem of over-squashing.

6.2 OGB - `molhiv`

For real-world datasets, we first evaluated our models on the `ogbg-molhiv` dataset with 5-regular bipartite expanders. As shown in Table 1, GIN+PM+Summation (bipartite expanders built with

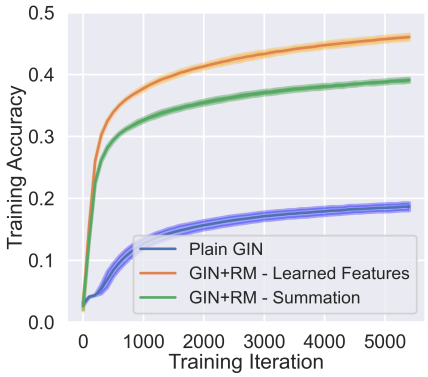


Figure 2: Mean training accuracy ($\pm$ STD) on the Tree-NeighborsMatch dataset [12] with binary trees of `depth=5`. It is clear that the interleaving message passing on the original graph and the higher-order expander graph helps mitigate the issue of over-squashing.

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

Table 1: Mean $\pm$ STD test ROC-AUC score on the `ogbg-molhiv` dataset [40], with various expander graph construction methods (PM, RM). Best, Second Best and Third Best results are coloured respectively.

perfect matchings, using summation to handle hyperedge node features) gives the best performance. Interestingly, no strategies reach the same performance as the original expander graph paper [20]. We suggest that this may be because higher-order interactions are not useful in the `ogbg-molhiv` dataset, or it may be due to how we handled the learning of hyperedge node features. Investigating this would be useful in future work, using the Long Range Graph Benchmark [41].

Effect of Ramanujan condition We observe in Table 1 that Ramanujan expanders do not bring much benefit to the performances. This may suggest that enforcing the Ramanujan condition does not lead to a noticeable performance increase beyond merely using random k-regular bipartite expanders.

6.3 OGB - `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

Table 2: Mean $\pm$ STD test F1 scores on the `ogbg-code2` dataset. Best, Second Best and Third Best results are coloured respectively.

We also evaluated the performance on `ogbg-code2`. We generated random 3-regular bipartite graphs using perfect matchings. As shown in Table 2, our best method, learning the features of the hyperedge nodes, outperforms both plain GIN [40] and GIN + EGP [20]. One possibile explanation for this is that higher-order interactions may be helpful for this task, giving rise to a benefit from our approach.

7 Conclusion

We explored bipartite expander graphs as a solution for addressing over-squashing in graph neural networks. These hypergraphs are easily constructed, and even randomly generated k-regular bipartite graphs show favourable expansion properties. Our experiments yielded promising results, especially on Tree-NeighborsMatch and OGB - `code2`, indicating that hypergraph expanders can help to mitigate over-squashing.

Acknowledgments and Disclosure of Funding

We would like to thank Prof. Pietro Liò and Dr Petar Veličković for creating and delivering the "L45: Representation Learning on Graphs and Networks" course in Part III of the Computer Science Tripos at Cambridge, which made this paper possible.

References

[1] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[2] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[3] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1263–1272. PMLR, 2017.

[4] Kien Do, Truyen Tran, and Svetha Venkatesha. Graph transformation policy network for chemical reaction prediction, 2018.

[5] Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. Knowledge transfer for out-of-knowledge-base entities : A graph neural network approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1802–1808, 2017.

[6] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference*. ACM, may 2019.

[7] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, abs/2104.13478, 2021.

[8] Petar Veličković. Message passing all the way up. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022.

[9] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[10] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks, 2018.

[11] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 3538–3545. AAAI Press, 2018.

[12] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications, 2020.

[13] Francesco Di Giovanni, T. Konstantin Rusch, Michael M. Bronstein, Andreea Deac, Marc Lackenby, Siddhartha Mishra, and Petar Veličković. How does over-squashing affect the power of gnns?, 2023.

[14] Rickard Brüel-Gabrielsson, Mikhail Yurochkin, and Justin Solomon. Rewiring with positional encodings for graph neural networks, 2022.

[15] Ralph Abboud, Radoslav Dimitrov, and Ismail Ilkan Ceylan. Shortest path networks for graph property prediction. In *The First Learning on Graphs Conference*, 2022.

[16] Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yu Guang Wang, Pietro Liò, Guido Montúfar, and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks, 2021.

[17] Cristian Bodnar, Fabrizio Frasca, Yu Guang Wang, Nina Otter, Guido Montúfar, Pietro Liò, and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks, 2021.

[18] Adrian Arnaiz-Rodriguez, Ahmed Begga, Francisco Escolano, and Nuria Oliver. Diffwire: Inductive graph rewiring via the lovász bound, 2022.

[19] Kedar Karhadkar, Pradeep Kr. Banerjee, and Guido Montufar. FoSR: First-order spectral rewiring for addressing oversquashing in GNNs. In *The Eleventh International Conference on Learning Representations*, 2023.

[20] Andreea Deac, Marc Lackenby, and Petar Veličković. Expander graph propagation, 2022.

[21] 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.

[22] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. *CoRR*, abs/1809.09401, 2018.

[23] Mehmet Emin Aktas, Thu Nguyen, Sidra Jawaid, Rakin Riza, and Esra Akbas. Identifying critical higher-order interactions in complex networks, 2021.

[24] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

[25] Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Lio', and Michael Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology, 2023.

[26] Luis Müller, Mikhail Galkin, Christopher Morris, and Ladislav Rampášek. Attending to graph transformers, 2023.

[27] Richard J Trudeau. *Introduction to graph theory*. Courier Corporation, 2013.

[28] Alain Bretto. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, 2013.

[29] Maruti Ram Murty. Ramanujan graphs. *J. Ramanujan Math. Soc. 18, No.1 (2003) 1–20*.

[30] Emmanuel Kowalski. *An introduction to expander graphs*. Société mathématique de France, 2019.

[31] Giuliana P Davidoff, Peter Sarnak, and Alain Valette. *Elementary number theory, group theory, and Ramanujan graphs*, volume 55. Cambridge university press Cambridge, 2003.

[32] Professor Cayley. Desiderata and suggestions: No. 2. the theory of groups: graphical representation. *American journal of mathematics*, 1(2):174–176, 1878.

[33] M Ram Murty. Ramanujan graphs: An introduction. *Indian J. Discrete Math*, 6(2):91–127, 2020.

[34] Fan RK Chung. Diameters and eigenvalues. *Journal of the American Mathematical Society*, 2(2):187–196, 1989.

[35] Edwin R Van Dam and Willem H Haemers. Eigenvalues and the diameter of graphs. *Linear and Multilinear Algebra*, 39(1-2):33–44, 1995.

[36] Alon Nilli. On the second eigenvalue of a graph. *Discrete Mathematics*, 91(2):207–210, 1991.

[37] Jozef Dodziuk. Difference equations, isoperimetric inequality and transience of certain random walks. *Transactions of the American Mathematical Society*, 284(2):787–794, 1984.

[38] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

[39] Ryan O'Donnell. A theorist's toolkit lecture series, lecture 12, cmu 2013. *https://www.cs.cmu.edu/ odonnell/toolkit13/lecture12.pdf*.

[40] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22118–22133. Curran Associates, Inc., 2020.

[41] Vijay Prakash Dwivedi, Ladislav Rampášek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.