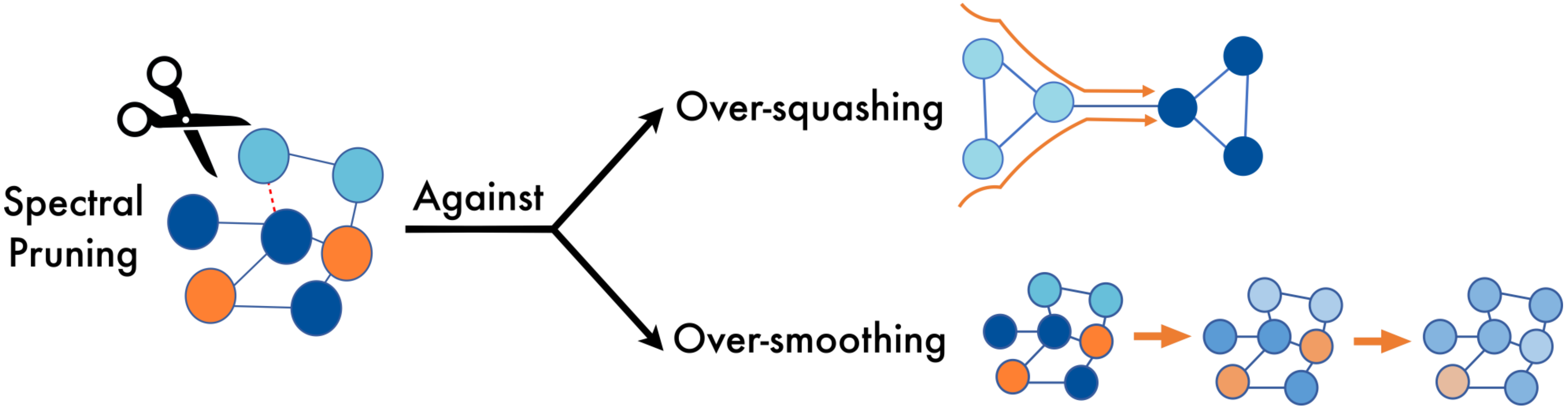
CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

Spectral Graph Pruning Against Over-Squashing and Over-Smoothing

*Adarsh Jamadandi * [1,2], Celia Rubio-Madrigal * [2], Rebekka Burkholz [2]*

[1] *Universität des Saarlandes* [2] *CISPA Helmholtz Center for Information Security*
** Equal contribution*



Background

- GNNs learn from graph-structured data
- Training GNNs → message passing on <u>input graph</u>

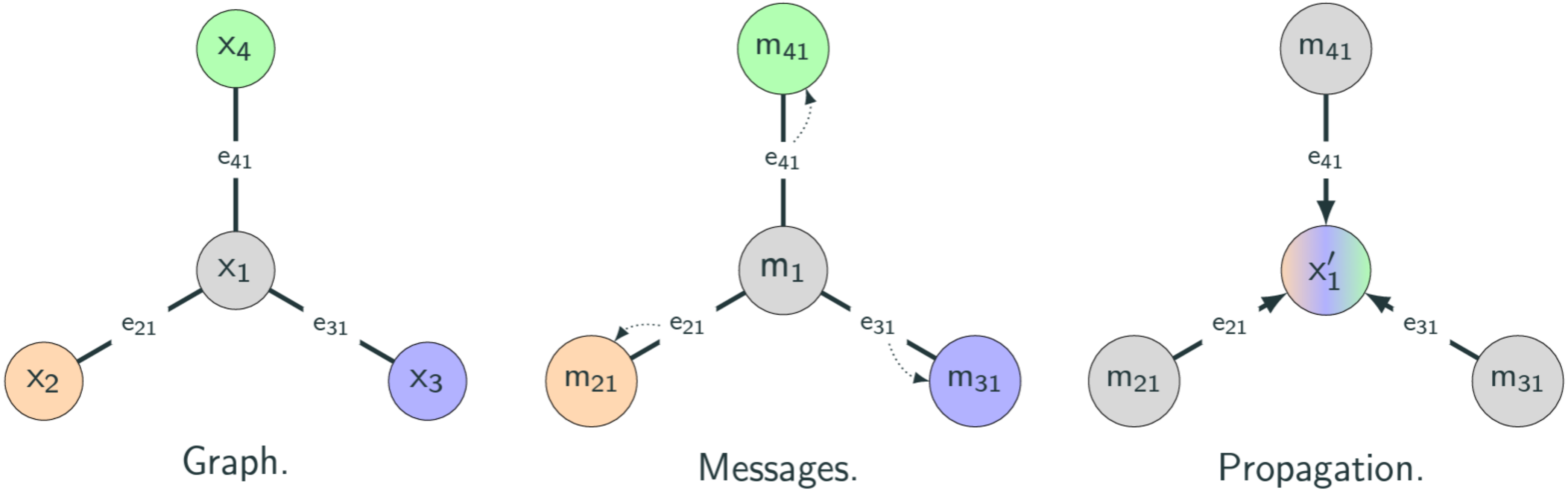
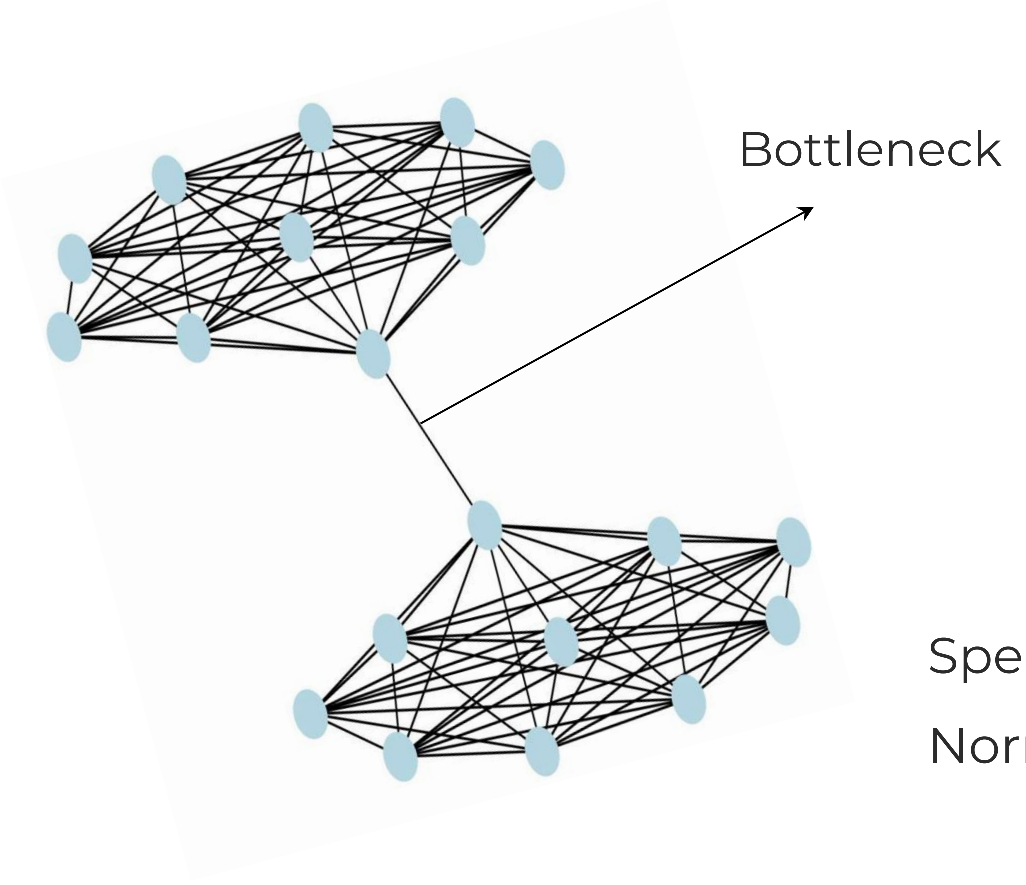


Image from: D. Grattarola, "Graph Deep Learning" (2021). https://danielegrattarola.github.io/files/talks/2021-03-01-USI_GDL_GNNs.pdf

Over-squashing

Bottlenecks obstruct the flow of information during message passing



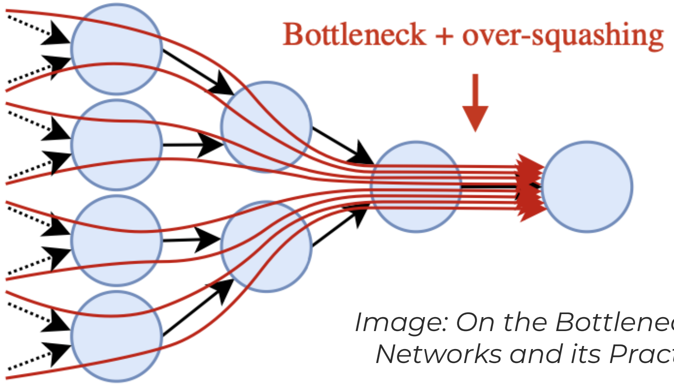


Image: On the Bottleneck of Graph Neural Networks and its Practical Implications

Spectral gap: $\lambda_1 - \lambda_0 \ (= \lambda_1)$

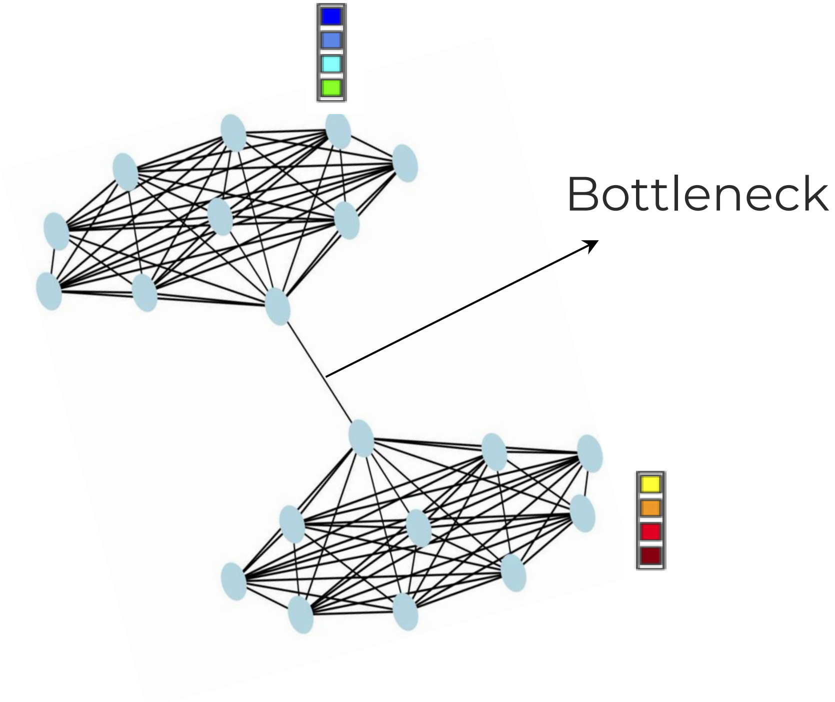
Normalized Laplacian: $L_G = I - D^{-1/2} A \ D^{-1/2}$

Small spectral gap ≡ bottlenecks

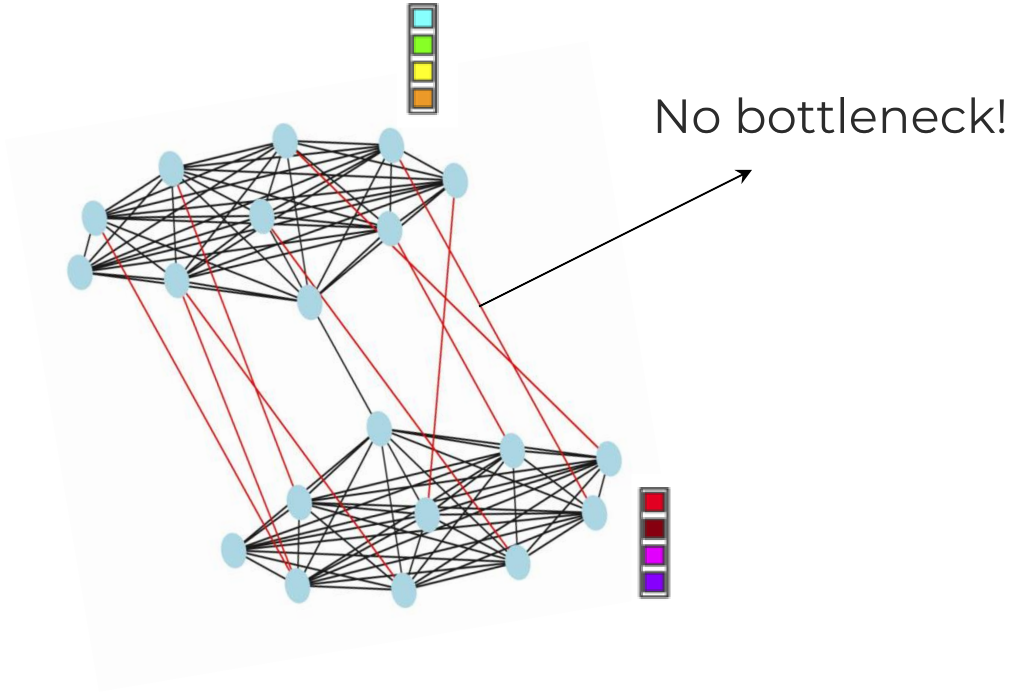
Trade-off?

Do we need to balance over-squashing and over-smoothing?

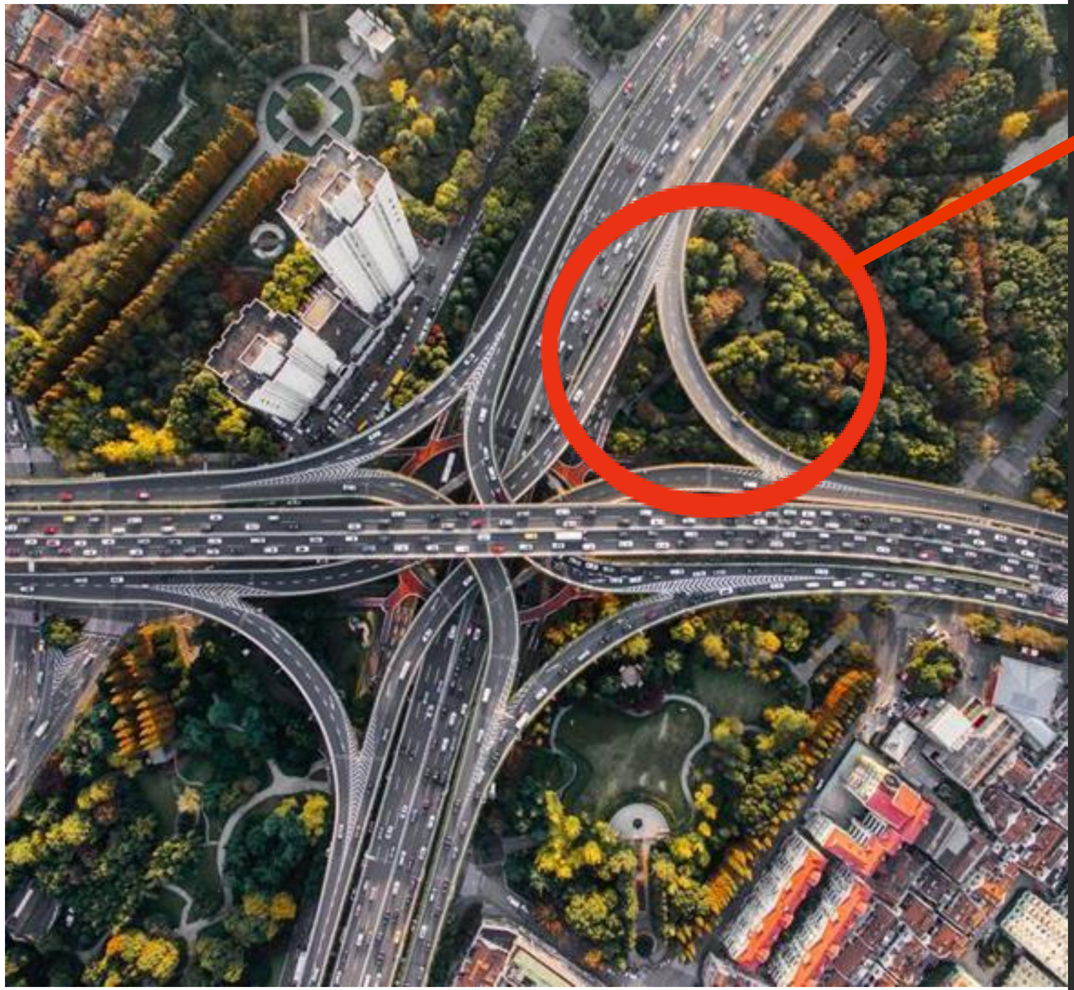
Over-squashing!



Over-smoothing?



Braess Paradox



Adding this extra road causes delays (Braess, 1968)

- Not all edge additions increase connectivity
- Not all edge deletions decrease connectivity

(Eldan et al., 2017) ⇒ there is a Braess Paradox for the **spectral gap** of the normalized Laplacian

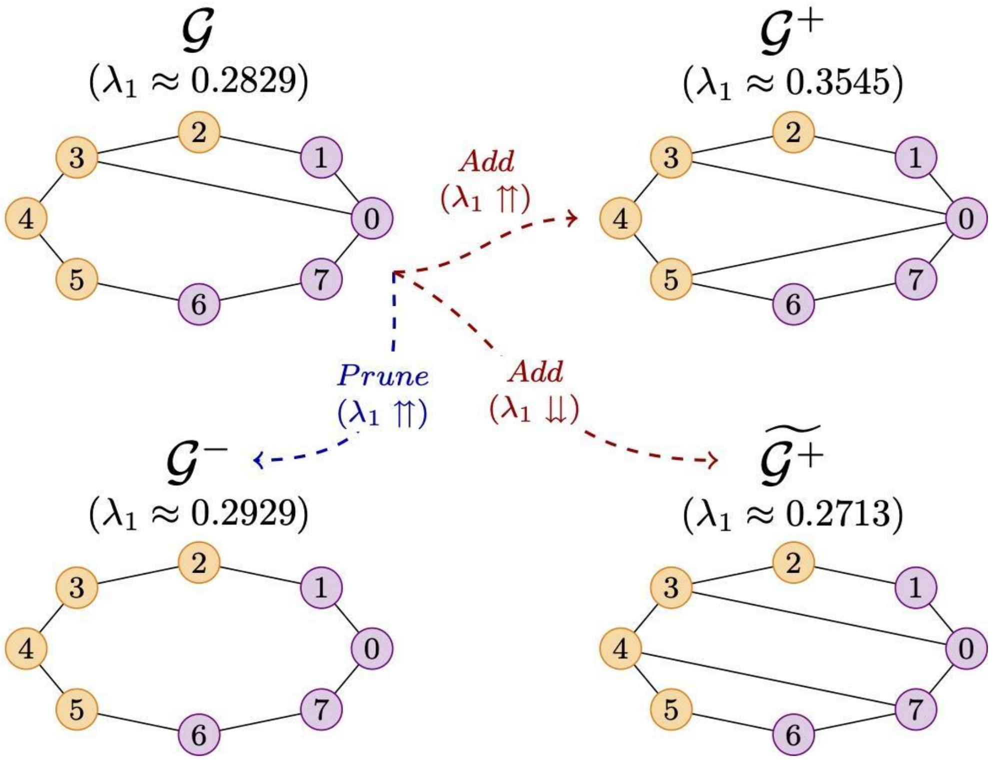
We can

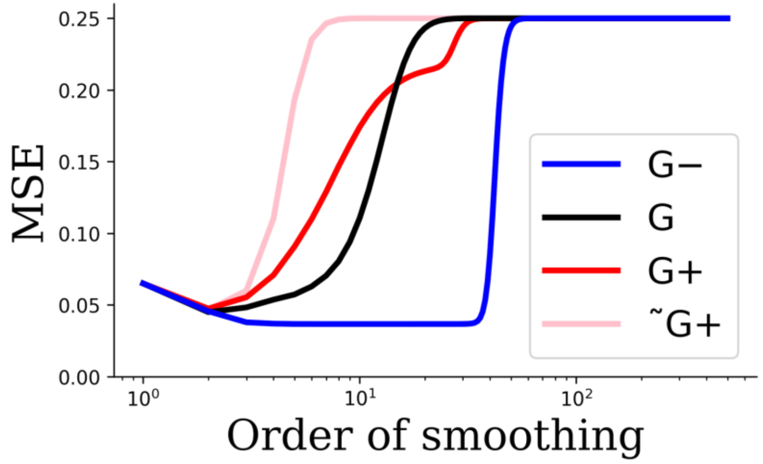
1. DELETE edges
2. INCREASE λ_1

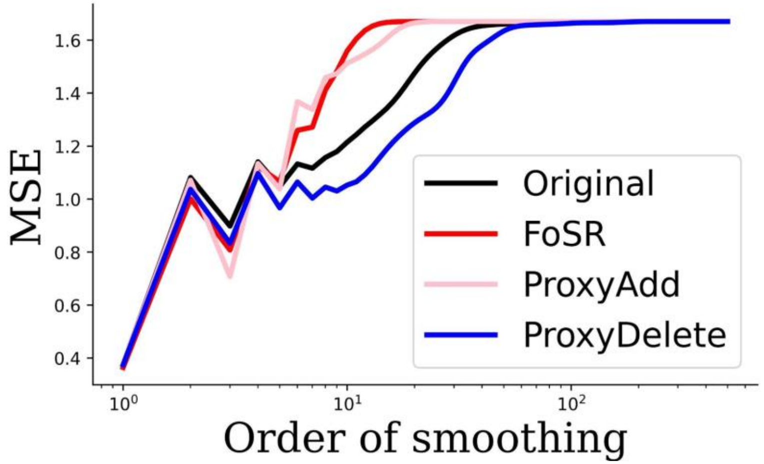
   (mitigating over-squashing)

Existence proof / Counterexample

1. The spectral gap increases (helping over-squashing)
2. The order of smoothing decreases (plot →)
   - Also in real-world datasets (Texas plot ↘)







Two criteria for rewiring

Calculating the spectral gap is <u>costly</u>. Two suggestions:

1. Eldan's sufficient criterion for the Braess paradox.
2. Proxy given by Matrix Perturbation theory (O(1)!)

Lemma 2.1. *Eldan et al. (2017): Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a finite graph, with f denoting the eigenvector and $\lambda_1(\mathcal{L}_\mathcal{G})$ the eigenvalue corresponding to the spectral gap. Let $\{u, v\} \notin \mathcal{V}$ be two vertices that are not connected by an edge. Denote $\hat{\mathcal{G}} = (\mathcal{V}, \hat{\mathcal{E}})$, the new graph obtained after adding an edge between $\{u, v\}$, i.e., $\hat{\mathcal{E}} := \mathcal{E} \cup \{u, v\}$. Denote with $\mathcal{P}_f := \langle f, \hat{f}_0 \rangle$ the projection of f onto the top eigenvector of $\hat{\mathcal{G}}$. Define $g\left(u, v, \mathcal{L}_\mathcal{G}\right) :=$*

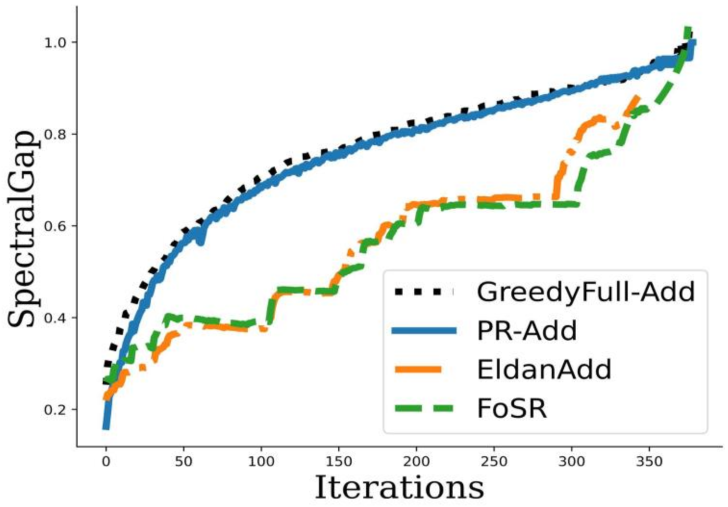
$$-\mathcal{P}_f^2 \lambda_1(\mathcal{L}_\mathcal{G}) - 2(1 - \lambda_1(\mathcal{L}_\mathcal{G})) \left(\frac{\sqrt{d_u + 1} - \sqrt{d_u}}{\sqrt{d_u + 1}} f_u^2 + \frac{\sqrt{d_v + 1} - \sqrt{d_v}}{\sqrt{d_v + 1}} f_v^2 \right) + \frac{2 f_u f_v}{\sqrt{d_u + 1}\sqrt{d_v + 1}}.$$

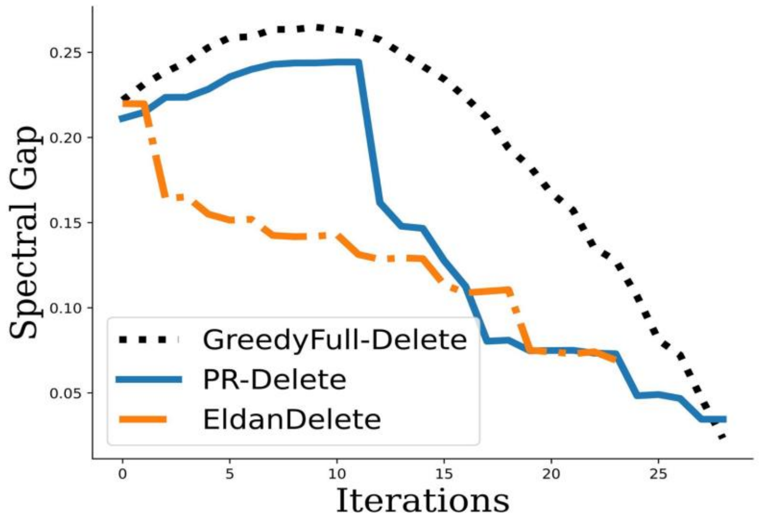
If $g\left(u, v, \mathcal{L}_\mathcal{G}\right) > 0$, then $\lambda_1(\mathcal{L}_\mathcal{G}) > \lambda_1(\mathcal{L}_{\hat{\mathcal{G}}})$.

$$\acute{\lambda} \approx \lambda + \Delta w_{u,v}((f_u - f_v)^2 - \lambda(f_u^2 + f_v^2));$$

Quality of approximation and runtime

Approximation compared to full calculation (black dots):





Seconds compared to other literature:

Method	Cora	Citeseer	Chameleon	Squirrel
FoSR	4.69	5.33	5.04	19.48
SDRF	19.63	173.92	17.93	155.95
ProxyAdd	4.30	3.13	1.15	9.12
ProxyDelete	1.18	0.86	1.46	7.26

We outperform baselines

- Benchmarks for over-squashing and over-smoothing

Table 1: Results on Long Range Graph Benchmark datasets.

Method	PascalVOC-SP (Test F1 ↑)	Peptides-Func (Test AP ↑)	Peptides-Struct (Test MAE ↓)
Baseline-GCN	0.1268±0.0060	0.5930±0.0023	0.3496±0.0013
DRew+GCN	0.1848±0.0107	**0.6996±0.0076**	0.2781±0.0028
FoSR+GCN	0.2157±0.0057	0.6526±0.0014	0.2499±0.0006
ProxyAdd+GCN	**0.2213±0.0011**	0.6789±0.0002	**0.2465±0.0004**
ProxyDelete+GCN	0.2170±0.0015	0.6908±0.0007	**0.2470±0.0080**

Table 3: Node classification on Amazon-Ratings.

Method	#EdgesAdded	Accuracy	#EdgesDeleted	Accuracy	Layers
GCN	-	47.20±0.33	-	47.20±0.33	10
GCN+FoSR	25	49.68±0.73	-	-	10
GCN+Eldan	25	48.71±0.99	100	**50.15±0.50**	10
GCN+ProxyGap	10	49.72±0.41	50	**49.75±0.46**	10
GAT	-	47.43±0.44	-	47.43±0.44	10
GAT+FoSR	25	51.36±0.62	-	-	10
GAT+Eldan	25	51.68±0.60	50	**51.80±0.27**	10
GAT+ProxyGap	20	49.06±0.92	100	**51.72±0.30**	10
GCN	-	47.32±0.59	-	47.32±0.59	20
GCN+FoSR	100	49.57±0.39	-	-	20
GCN+Eldan	50	**49.66±0.31**	20	48.32±0.76	20
GCN+ProxyGap	50	49.48±0.59	500	**49.58±0.59**	20
GAT	-	47.31±0.46	-	47.31±0.46	20
GAT+FoSR	100	51.31±0.44	-	-	20
GAT+Eldan	20	51.40±0.36	20	**51.64±0.44**	20
GAT+ProxyGap	50	47.53±0.90	20	**51.69±0.46**	20

Table 4: Node classification on Minesweeper.

Method	#EdgesAdded	Accuracy	#EdgesDeleted	Test ROC	Layers
GCN	-	88.57± 0.64	-	88.57± 0.64	10
GCN+FoSR	50	90.15±0.55	-	-	10
GCN+Eldan	100	**90.11±0.50**	50	89.49±0.60	10
GCN+ProxyGap	20	**89.59±0.50**	20	89.57±0.49	10
GAT	-	93.60±0.64	-	93.60±0.64	10
GAT+FoSR	100	93.14±0.43	-	-	10
GAT+Eldan	50	93.26±0.48	100	**93.82±0.56**	10
GAT+ProxyGap	20	93.60±0.69	20	**93.65±0.84**	10
GCN	-	87.41±0.65	-	87.41±0.65	20
GCN+FoSR	100	89.64±0.55	-	-	20
GCN+Eldan	72	**89.70±0.57**	10	88.90±0.44	20
GCN+ProxyGap	20	**89.46±0.50**	50	89.35±0.30	20
GAT	-	93.92±0.52	-	93.92±0.52	20
GAT+FoSR	50	93.56±0.64	-	-	20
GAT+Eldan	10	93.92±0.44	20	**95.48±0.64**	20
GAT+ProxyGap	20	**94.89±0.67**	20	94.64±0.81	20

Results for Graph Lottery Tickets

- Connecting fields with different objectives
- Deleting edges fights both challenges, explaining success of GLT

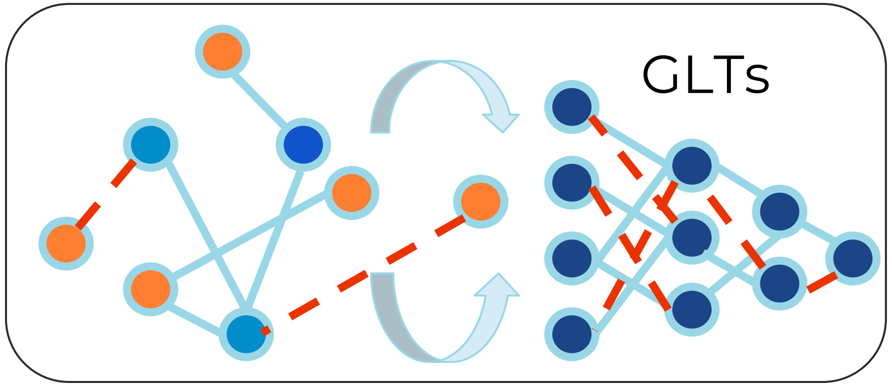


Table 3: Pruning for lottery tickets comparing UGS to our ELDANDELETE pruning and our PROXYDELETE pruning. We report Graph Sparsity (GS), Weight Sparsity (WS), and Accuracy (Acc).

Method	Cora			Citeseer			Pubmed		
Metrics	GS	WS	Acc	GS	WS	Acc	GS	WS	Acc
UGS	79.85%	97.86%	68.46±1.89	78.10%	97.50%	**66.50±0.60**	68.67%	94.52%	76.90±1.83
ELDANDELETE-UGS	79.70%	97.31%	68.73±0.01	77.84%	96.78%	64.60±0.00	70.11%	93.17%	**78.00±0.42**
PROXYDELETE-UGS	78.81%	97.24%	**69.26±0.63**	77.50%	95.83%	65.43±0.60	78.81%	97.24%	75.25±0.25

Summary

- Over-squashing and Over-smoothing not a trade-off. Can be mitigated simultaneously.
- We leverage Braess paradox to show deleting edges can also help alleviate over-squashing and slow down the rate of detrimental over-smoothing.
- We propose a computationally friendly spectral gap optimization scheme to rewire graphs.
- Additonally, as a bonus application we show our proposal can help identify graph lottery tickets!