On over-squashing in Graph Neural Networks: from theoretical understanding to graph-rewiring solutions

Francesco Di Giovanni

University of Cambridge

Complex Network Analysis Discussion Group 6 March 2023, 03/04/2023

Table of contents

Message Passing Neural Networks

Notations and conventions: a useful shopping list

- Graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ with n nodes and $\mathbf{A} \in \mathbb{R}^{n \times n}$ s.t. $(v, u) \in \mathsf{E} \iff A_{vu} > 0$
- The diagonal degree matrix is $\mathbf{D} = \mathrm{diag}(d_v)$, with $d_v = \sum_u A_{vu}$.
- Symmetrically normalized adjacency: $\hat{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$.
- Node-wise features $H : \mathsf{V} \to \mathbb{R}^p$ with matrix representation $\mathbf{H} \in \mathbb{R}^{n \times p}$

An important class of GNNs: the MPNN-paradigm

- Combine function $\mathsf{com}^{(t)}$
- Permutation-invariant aggregation map $\mathsf{agg}^{(t)}$
- For any layer t we compute node embeddings $\mathbf{h}_v^{(t)}$ by

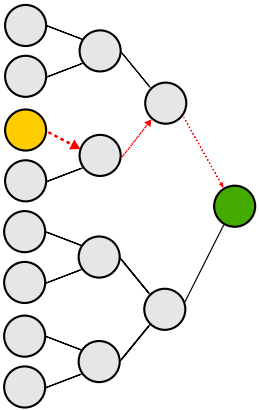
$$\mathbf{h}_v^{(t)} = \mathsf{com}^{(t)}\left(\mathbf{h}_v^{(t-1)}, \mathsf{agg}^{(t)}\left(\{\{\mathbf{h}_u^{(t-1)},\ (u,v) \in \mathsf{E}\}\}\right)\right)$$

Message Passing Neural Network (MPNN) of m layers is then a map

$$\mathsf{MPNN}_\theta^{(m)} = \mathsf{com}^{(m)} \circ \mathsf{agg}^{(m)} \circ \cdots \circ \mathsf{com}^{(1)} \circ \mathsf{agg}^{(1)}$$

A limitation of MPNNs: over-squashing

Second-order effects: Over-squashing (Alon and Yahav, 2021)



- Depending on the graph-topology but independent of the chosen MPNN, the size of the $r-$hop of a node may grow **exponentially**
- As messages are sent through fixed-dimension channels, node green is no longer *sensitive* to information contained at node yellow

Why over-squashing is an interesting problem?

- Over-squashing is intrinsic to the underlying graph topology and cannot be easily fixed by different choices of $\mathsf{com}^{(t)}$ or $\mathsf{agg}^{(t)}$

Why over-squashing is an interesting problem?

- Over-squashing is intrinsic to the underlying graph topology and cannot be easily fixed by different choices of $\mathsf{com}^{(t)}$ or $\mathsf{agg}^{(t)}$
- Even simple problems cannot be solved by an MPNN if the required radius of interactions is 'too' large $\rightarrow$ *Expressive power only tells a partial angle of the story?*

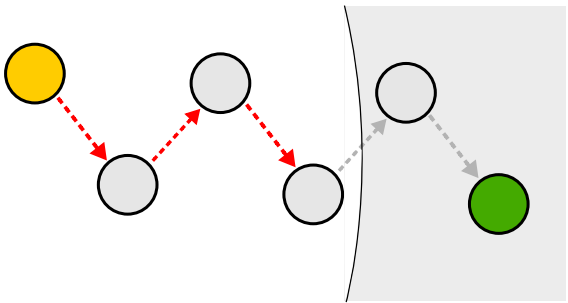
Why over-squashing is an interesting problem?

- Over-squashing is intrinsic to the underlying graph topology and cannot be easily fixed by different choices of $\mathsf{com}^{(t)}$ or $\mathsf{agg}^{(t)}$
- Even simple problems cannot be solved by an MPNN if the required radius of interactions is 'too' large $\rightarrow$ *Expressive power only tells a partial angle of the story?*
- If the topology is harmful, can we 'decouple' the computational graph from the input one? *The bigger picture: rewiring, graph-transformers, and the role of the graph* G?

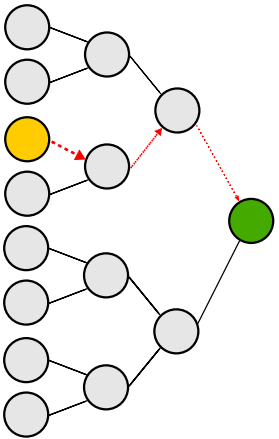
Understanding and correcting over-squashing: part I

Under-reaching

- Information cannot propagate further than there are layers in the MPNN
- Stack many layers m so that messages can be exchanged among distant pair of nodes



The over-squashing phenomenon[1]: heuristic point of view



- Depending on the graph-topology, the size of the $r-$hop of a node may grow **exponentially**
- As messages are sent through fixed-dimension channels, node green is no longer *sensitive* to information contained at node yellow

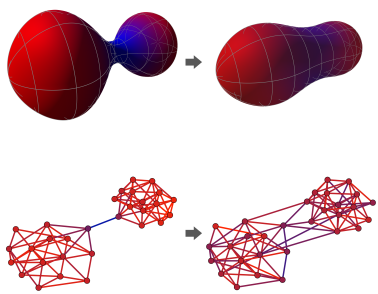
[1] Alon and Yahav (2021)

A first theoretical understanding of over-squashing

Published as a conference paper at ICLR 2022

UNDERSTANDING OVER-SQUASHING AND BOTTLENECKS ON GRAPHS VIA CURVATURE

Jake Topping[1,2†], **Francesco Di Giovanni**[3†], **Benjamin P. Chamberlain**[3], **Xiaowen Dong**[1], **and Michael M. Bronstein**[2,3]
[1]University of Oxford [2]Imperial College London [3]Twitter



- Sensitivity analysis to assess 'information flow' in an MPNN
- Proved that over-squashing is independent of the choice of the model, but arises when we have edges with high negative curvature
- Proposed first rewiring method directly aimed at correcting the graph topology along bottlenecks

Sensitivity analysis

- Consider the MPNN of the form

$$\mathbf{h}_v^{(t)} = \mathsf{com}^{(t)}\left(\mathbf{h}_v^{(t-1)}, \sum_u \hat{A}_{vu}\mathrm{mss}^{(t)}(\mathbf{h}_v^{(t-1)}, \mathbf{h}_u^{(t-1)})\right).$$

Sensitivity analysis

- Consider the MPNN of the form

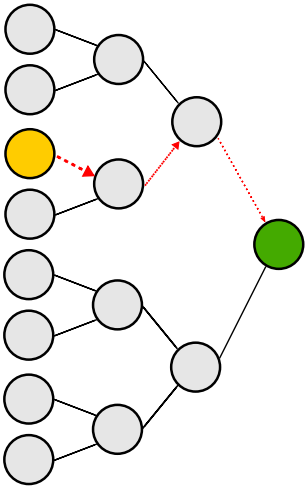
$$\mathbf{h}_v^{(t)} = \mathsf{com}^{(t)}\left(\mathbf{h}_v^{(t-1)}, \sum_u \hat{A}_{vu}\mathsf{mss}^{(t)}(\mathbf{h}_v^{(t-1)}, \mathbf{h}_u^{(t-1)})\right).$$

Lemma

Let $v, u \in \mathsf{V}$ with $d(u,v) = r$. If $|\nabla \mathsf{com}^{(t)}| \leq \alpha$, $|\nabla \mathsf{mss}^{(t)}| \leq \beta$ for $t \leq r$, then

$$\left|\left|\frac{\partial \mathbf{h}_v^{(r)}}{\partial \mathbf{h}_u^{(0)}}\right|\right|_\infty \leq (\alpha\beta)^r(\hat{\mathbf{A}}^r)_{vu}.$$

Over-squashing example: binary tree



- $(\hat{\mathbf{A}}^r)_{vu} = \frac{1}{2} \cdot 3^{-(r-1)}$
- Demonstrated in Tree-NeighborsMatch experiment in Alon and Yahav (2021)
- Sensitivity analysis is now the standard approach to monitor over-squashing in GNNs (Xu et al., 2018)
- If the graph topology induces over-squashing, can we identify the edges responsible for bottlenecks?

Edges with *high* negative curvature induce over-squashing

We introduce a curvature map Ric over the edges s.t. $(u, v) \mapsto \mathrm{Ric}(u, v)$ depends on triangles and 4-cycles based at (u, v)

	Cycle $C_{n\geq 5}$	Grid G_n	Clique K_n	Tree T_r
Graph				
Ric_G	0	0	$\frac{n}{n-1}$	$\frac{4}{r+1} - 2$

Theorem (Informal)

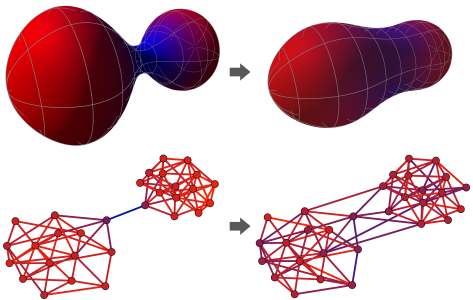
Over-squashing occurs along edges $u \sim v$ s.t. $\mathrm{Ric}(u, v)$ *has* **high** *negative curvature.*

Ricci flow is cooler than transformers

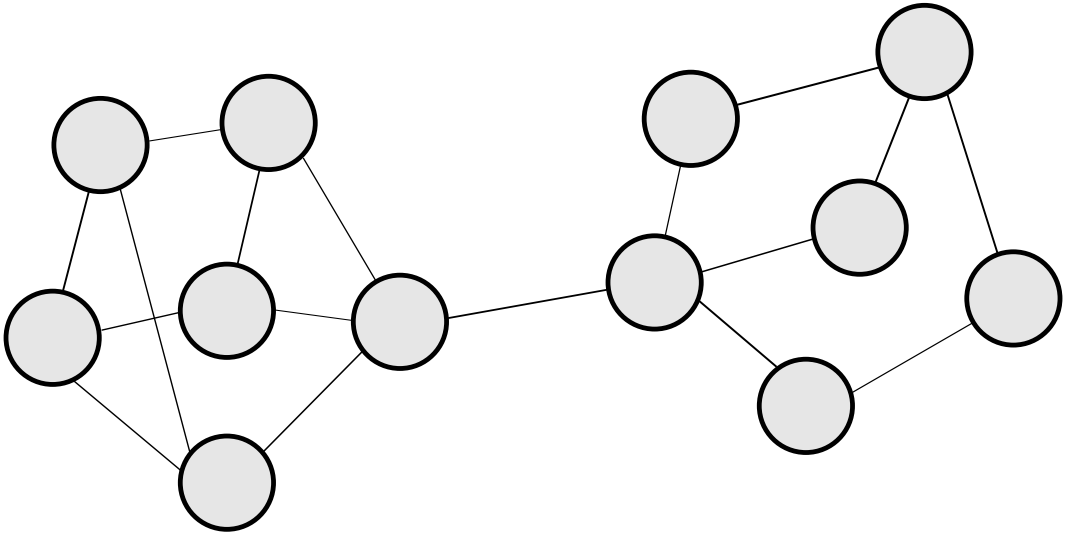
Negatively curved edges $\longrightarrow$ *bottlenecks* $\longrightarrow$ *over-squashing.*

Benefit of this analysis:

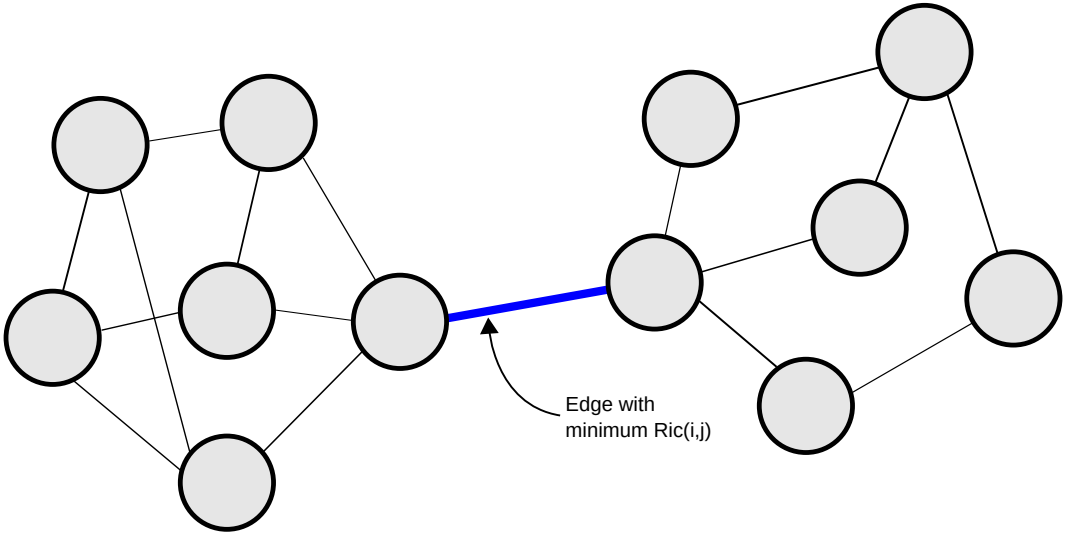
- we can surgically identify bottlenecks by studying the curvature
- we can add/remove edges accordingly $\longrightarrow$ we propose SDRF algorithm: $\mathcal{R}$ acts as discrete (backwards) Ricci flow



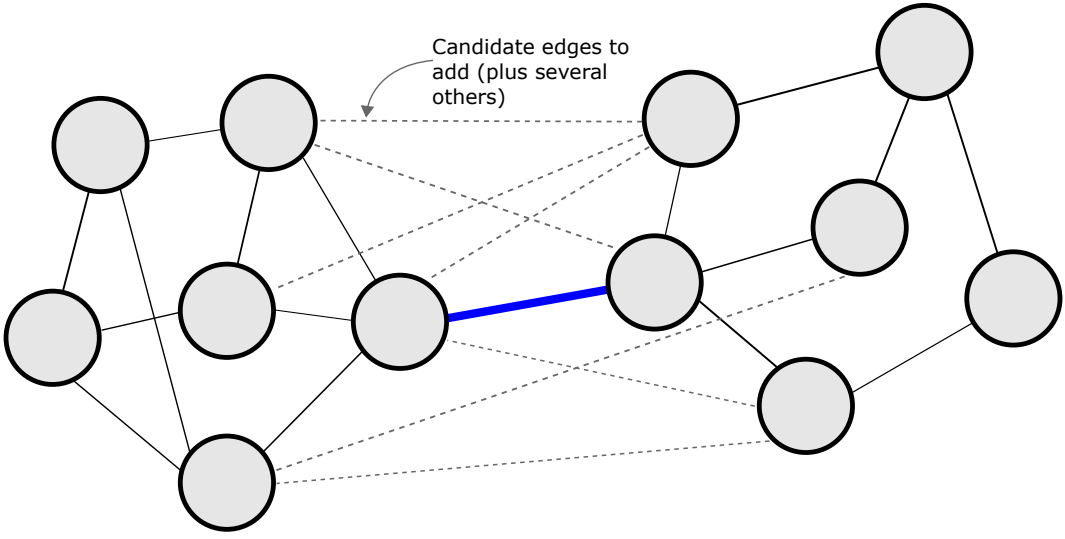
SDRF: Example



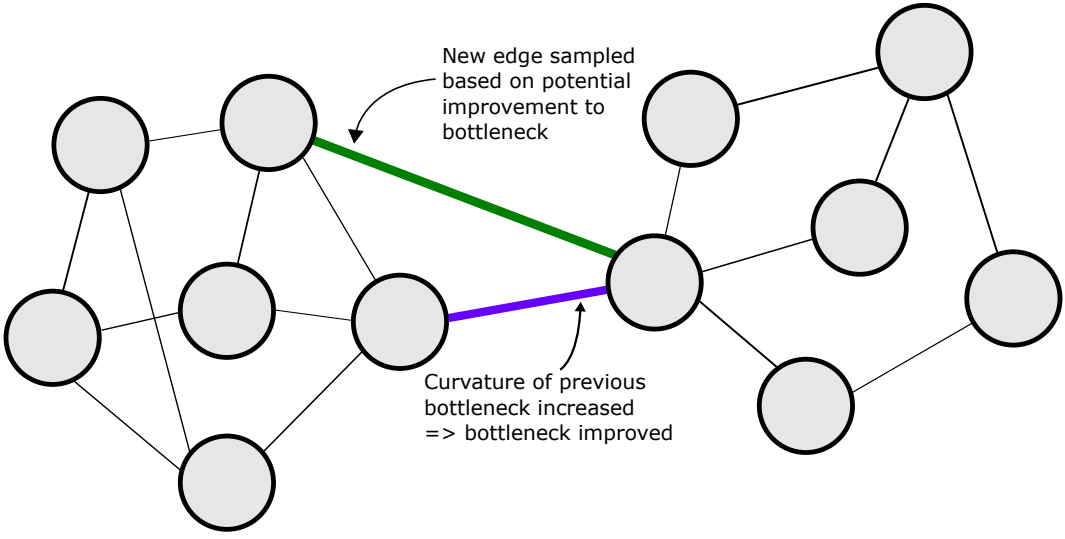
SDRF: Example



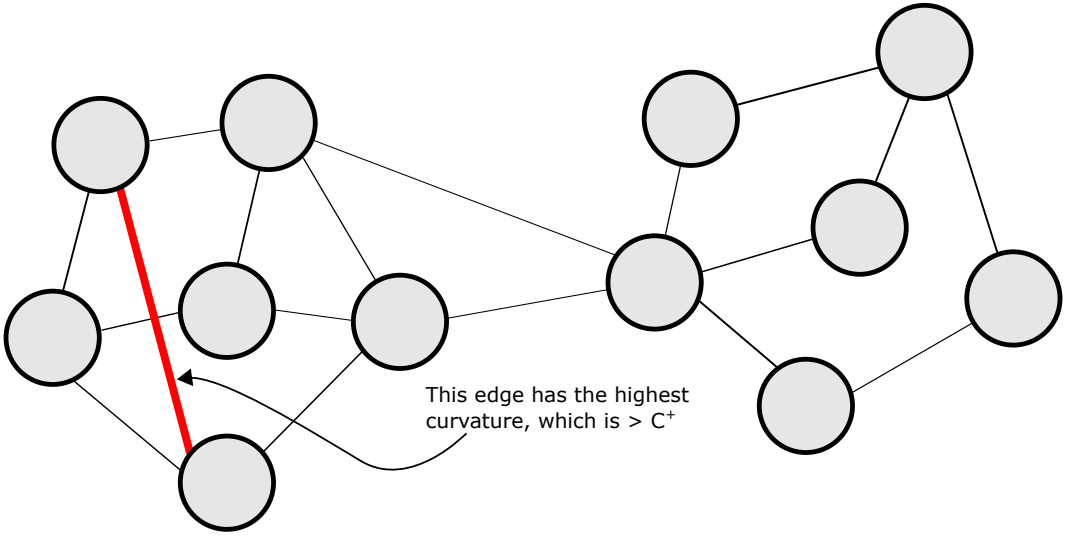
SDRF: Example



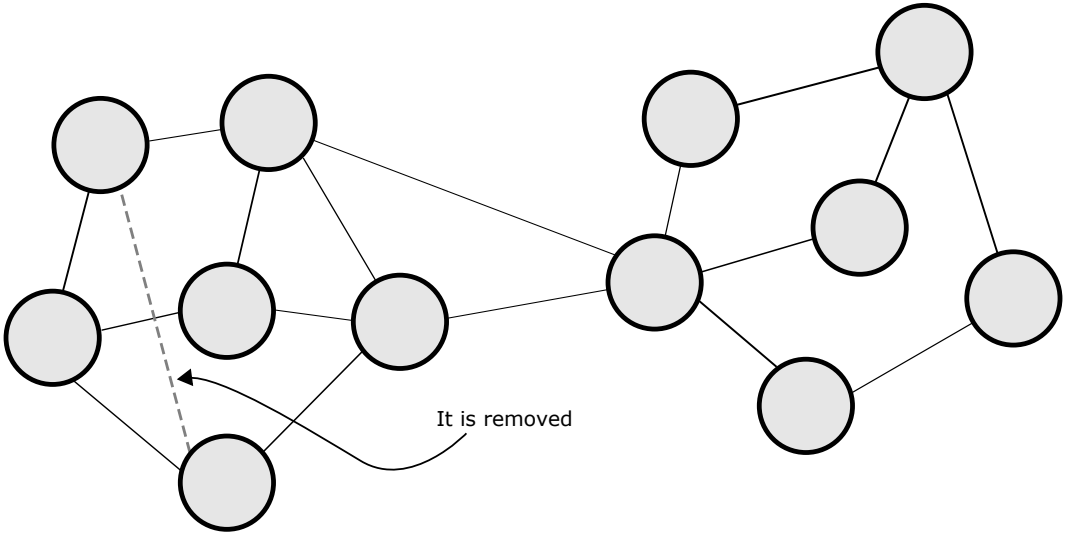
SDRF: Example



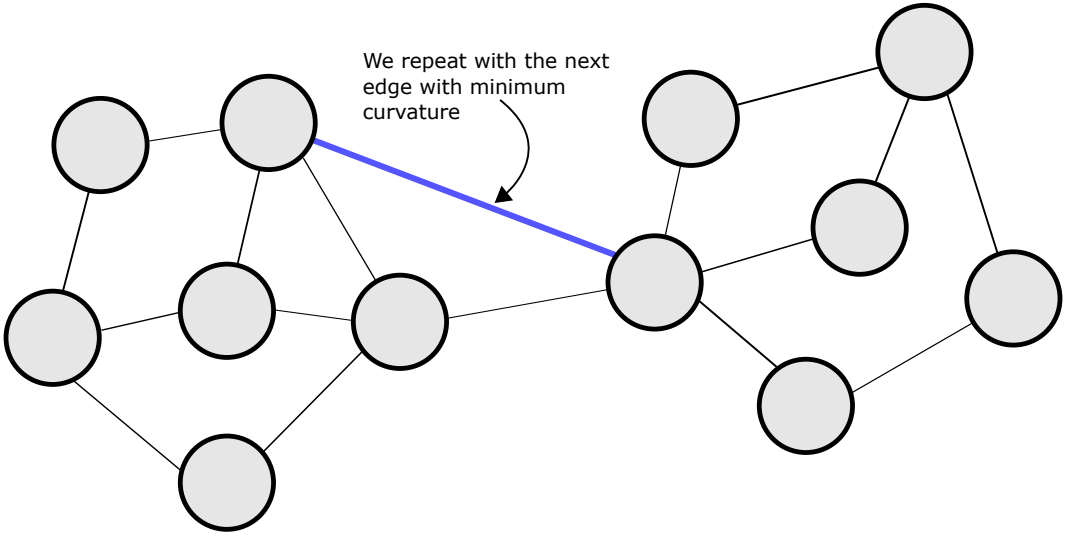
SDRF: Example



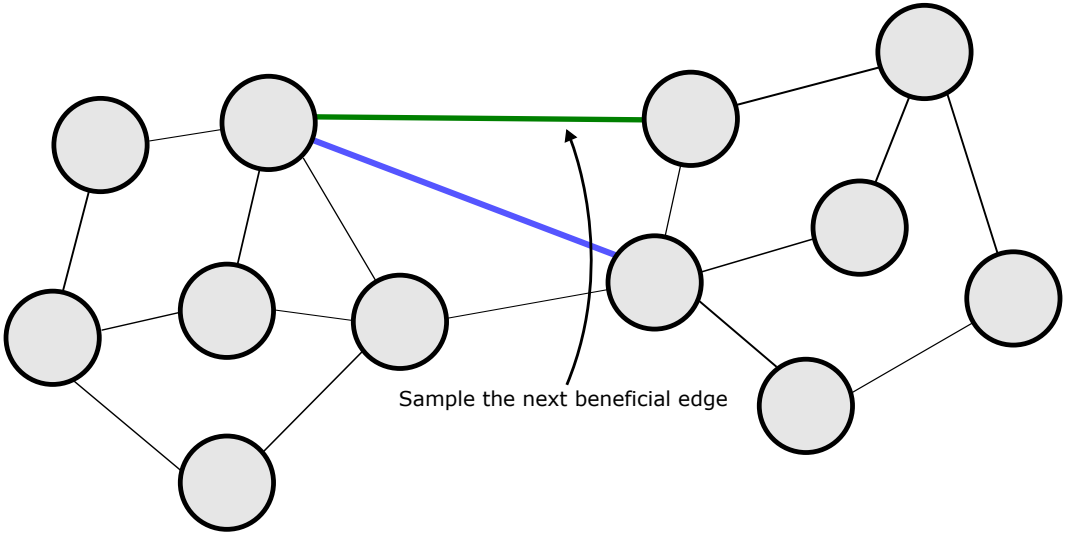
SDRF: Example



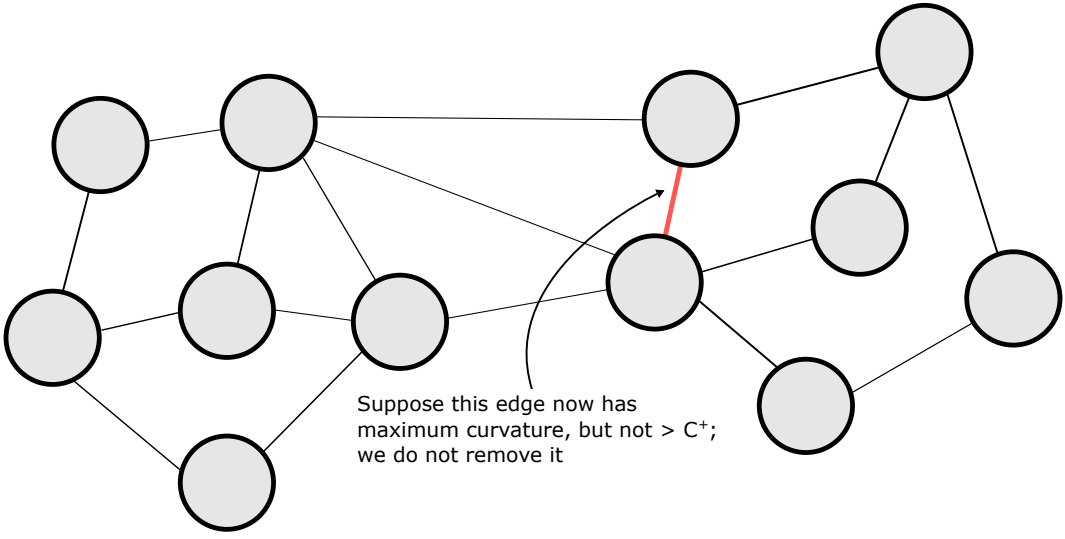
SDRF: Example



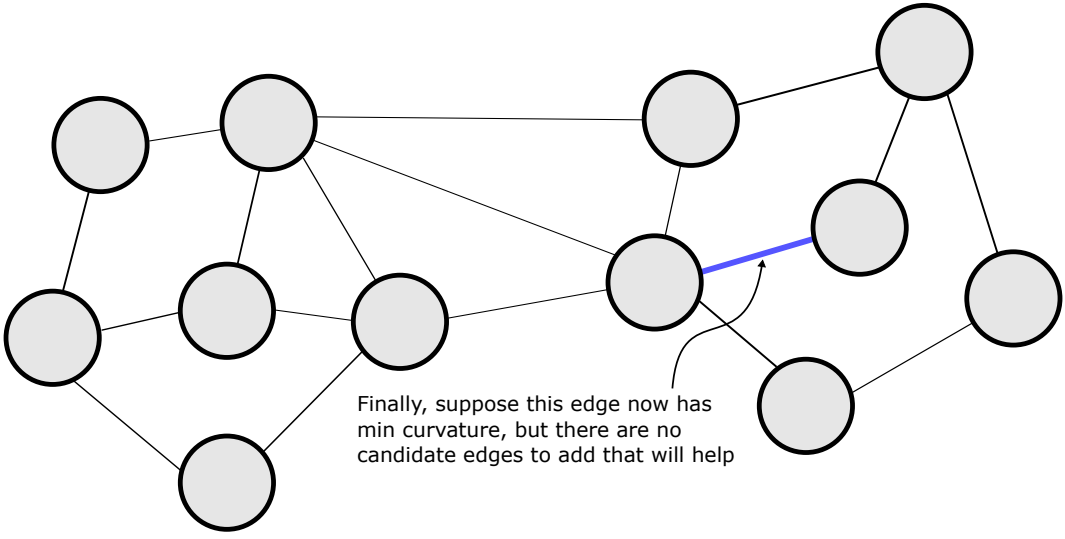
SDRF: Example



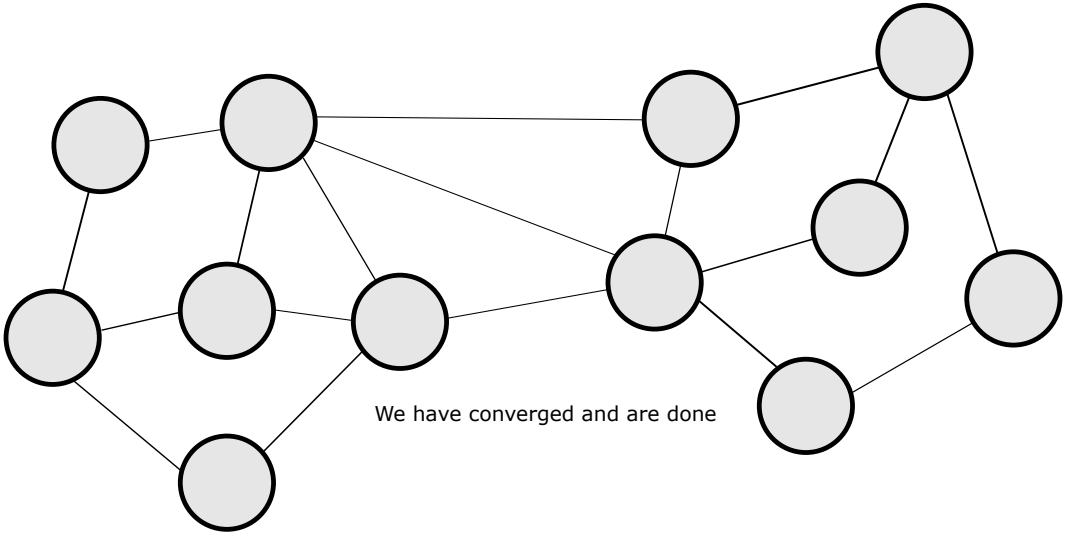
SDRF: Example



SDRF: Example

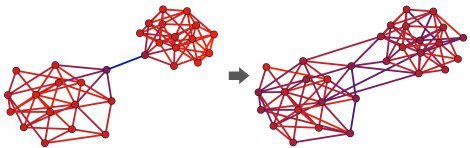


SDRF: Example



Recent ‘rewiring’ strategies to mitigate over-squashing: spectral approaches

The **Cheeger constant** h_{G} measures the ‘energy’ required to separate a graph into two communities $\rightsquigarrow$ related to the positive, smallest eigenvalue of $\boldsymbol{\Delta}$



Theorem (Di G. et al)

Some existing rewiring methods cannot improve h_{G} without making the graph too dense.

Deac et al. (2022), Arnaiz-Rodríguez et al. (2022), Karhadkar et al. (2022) $\rightarrow h_{\mathcal{R}(\mathsf{G})} > h_{\mathsf{G}}$

Why ‘improving’ h_{G} actually helps with over-squashing?

Understanding over-squashing: part II

Some questions we provide an answer to

Some important *questions left open* from the analysis in Topping et al. (2022):

(i) What is the impact of the **width** in mitigating over-squashing?

(ii) Can over-squashing be avoided by sufficiently **deep** models?

(iii) How does over-squashing relate to the graph **spectrum** and the underlying topology beyond curvature bounds that only apply to 2-hop propagation?

$\rightsquigarrow$ Provide a unified framework for both spatial and spectral rewiring approaches

Where you can find the answers

These questions are addressed in

On Over-Squashing in Message Passing Neural Networks: The Impact of Width, Depth, and Topology

Francesco Di Giovanni
University of Cambridge
fd405@cam.ac.uk

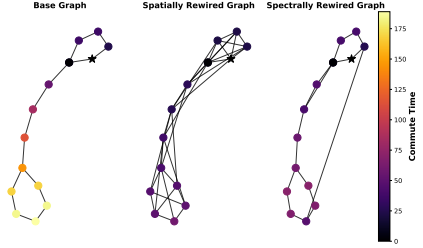
Lorenzo Giusti
Sapienza University
name.surname@uniroma1.it

Federico Barbero
University of Oxford
name.surname@cs.ox.ac.uk

Giulia Luise
Microsoft Research
namesurname@microsoft.com

Pietro Liò
University of Cambridge
pl219@cam.ac.uk

Michael Bronstein
University of Oxford
name.surname@cs.ox.ac.uk



Setting

Key assumption: We consider (isotropic) MPNNs of the form

$$\mathbf{h}_v^{(t)} = \sigma(\underbrace{c_{\mathsf{r}}\mathbf{W}_{\mathsf{r}}^{(t)}\mathbf{h}_v^{(t-1)}}_{\text{self}} + \underbrace{c_{\mathsf{a}}\mathbf{W}_{\mathsf{a}}^{(t)}\sum_u \mathsf{A}_{vu}\mathbf{h}_u^{(t-1)}}_{\text{aggregation}})$$

- σ is a pointwise nonlinear map, $\mathbf{W}^{(t)}$ are learnable weight matrices
- $c_{\mathsf{r}}, c_{\mathsf{a}} \geq 0$ measure the 'strength' of residual and aggregation terms
- $\mathbf{A}$ is a message-passing matrix typically chosen in $\{\mathbf{D}^{-1}\mathbf{A}, \mathbf{A}, \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\}$

$\rightsquigarrow$ include standard models as GCN, SAGE, GIN

The impact of width

We study over-squashing via gradient sensitivity: if $|\partial \mathbf{h}_v^{(m)} / \partial \mathbf{h}_u^{(0)}|$ is small, then **the feature at v is mostly insensitive to the information initially contained at u, after m layers**

Theorem (Sensitivity bounds)

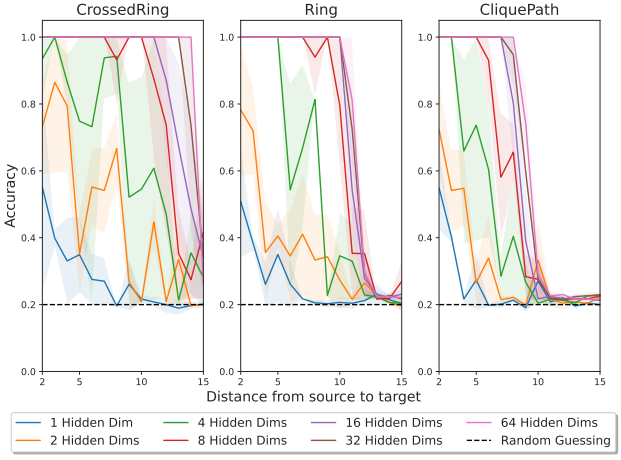
Consider an MPNN *as above for m layers, with c_σ the Lipschitz constant of σ and w the maximal entry-value over all weight matrices. For $v, u \in \mathsf{V}$ and width p, we have*

$$\left\| \frac{\partial \mathbf{h}_v^{(m)}}{\partial \mathbf{h}_u^{(0)}} \right\|_{L_1} \leq \underbrace{(c_\sigma w p)^m}_{\text{model}} \underbrace{(c_{\mathsf{r}} \mathbf{I} + c_{\mathsf{a}} \mathbf{A})^m_{vu}}_{\text{topology}}.$$

$\rightsquigarrow$ Differently from existing bounds, these apply to *any* number of layers m

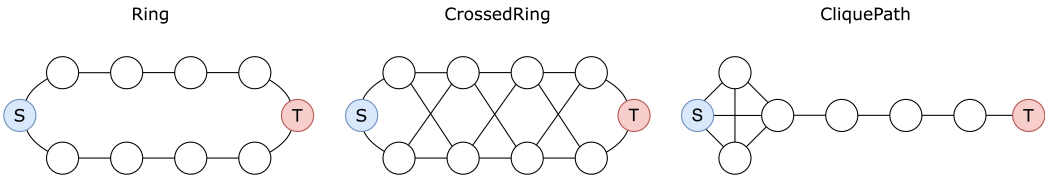
The impact of width

- Width can help, yet at the expense of over-fitting and generalization



What does it mean ‘long-range’?

Synthetic graph-transfer task (Bodnar et al., 2021):

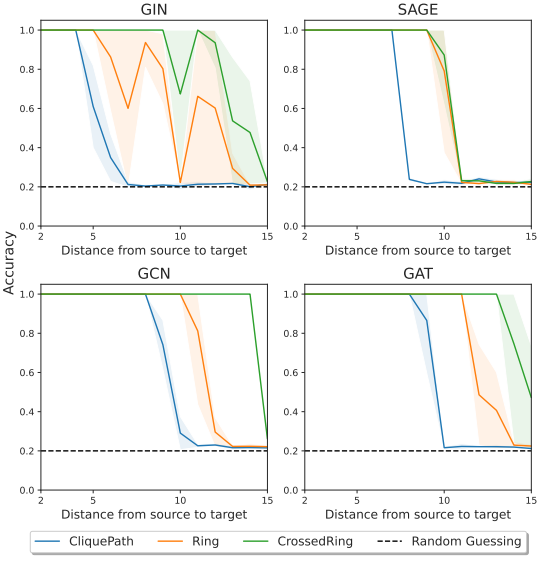


$$(c_{\mathsf{r}}\mathbf{I} + c_{\mathsf{a}}\mathbf{A})^r_{vu} = \left(\frac{1}{2}\right)^{r-1} \quad \text{for Ring}$$

$$(c_{\mathsf{r}}\mathbf{I} + c_{\mathsf{a}}\mathbf{A})^r_{vu} = \left(\frac{2}{3}\right)^{r-1} \quad \text{for CrossedRing}$$

$$(c_{\mathsf{r}}\mathbf{I} + c_{\mathsf{a}}\mathbf{A})^r_{vu} = \left(\frac{1}{2}\right)^{r-1} \frac{2}{(r\sqrt{r-2})} \quad \text{for CliquePath.}$$

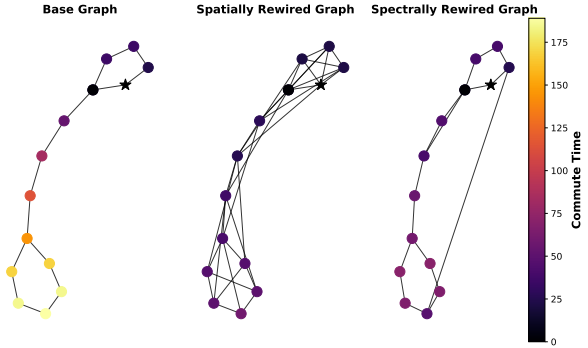
What does it mean ‘long-range’?



One slide on random-walk properties

A Random Walk (RW) on G is a Markov chain which, at each step, moves from a node v to one of its neighbours with probability $1/d_v$

- **Commute time** $\tau(v,u)$ is the expected number of steps in a RW starting at v to reach node u and *come back*. Effective Resistance (ER) is $\mathsf{Res}(v,u) = \tau(v,u)/2|\mathsf{E}|$



Over-squashing is aligned with access (commute) time metrics

- $\mathbf{A} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ and $\sigma = \mathsf{ReLU}$

Theorem (Informal)

In 'expectation', the 'obstruction' for two nodes to exchange messages in an MPNN *is approximated by* $\tau(v,u)/2|\mathsf{E}| = \mathsf{Res}(v,u)$.

Over-squashing is aligned with access (commute) time metrics

- $\mathbf{A} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ and $\sigma = \mathsf{ReLU}$

Theorem (Informal)

In 'expectation', the 'obstruction' for two nodes to exchange messages in an MPNN *is approximated by* $\tau(v,u)/2|\mathsf{E}| = \mathsf{Res}(v,u)$.

This has some **important consequences**:

(i) Reducing the sensitivity to messages from nodes with high effective resistance (ER) *can* be beneficial when the task only depends on 'local' interactions.

(ii) Over-squashing is an issue if the task depends on the interaction of nodes at high ER.

(iii) The ER represents an obstruction which is *independent of the number of layers*.

Concurrent work that attains similar conclusions (Black et al., 2023)

A unified framework

By (Lovász, 1993), a larger Cheeger constant $\Rightarrow$ lower overall effective resistance

- ***Spectral and spatial-rewiring methods manage to mitigate over-squashing since they decrease ER of*** G

A unified framework

By (Lovász, 1993), a larger Cheeger constant $\Rightarrow$ lower overall effective resistance

- ***Spectral and spatial-rewiring methods manage to mitigate over-squashing since they decrease ER of*** G
- ***If ER is bounded from above, then curvature is generally bounded from below (Devriendt and Lambiotte, 2022; Ollivier, 2007)***

What did we show?

- Width help mitigate over-squashing $\rightsquigarrow$ poor generalization

What did we show?

- Width help mitigate over-squashing $\rightsquigarrow$ poor generalization
- Depth m is (generally) not helpful: (i) when $m \sim \mathrm{diam}(\mathsf{G})$, over-squashing occurs among 'distant' nodes; (ii) when $m \gg 1$, sensitivity is generally lost $\rightsquigarrow$ *vanishing gradients*

What did we show?

- Width help mitigate over-squashing $\rightsquigarrow$ poor generalization
- Depth m is (generally) not helpful: (i) when $m \sim \mathrm{diam}(\mathsf{G})$, over-squashing occurs among 'distant' nodes; (ii) when $m \gg 1$, sensitivity is generally lost $\rightsquigarrow$ *vanishing gradients*
- Over-squashing is a topological issue: nodes at high access time (effective-resistance) struggle to exchange messages

What did we show?

- Width help mitigate over-squashing $\rightsquigarrow$ poor generalization
- Depth m is (generally) not helpful: (i) when $m \sim \mathrm{diam}(\mathsf{G})$, over-squashing occurs among 'distant' nodes; (ii) when $m \gg 1$, sensitivity is generally lost $\rightsquigarrow$ *vanishing gradients*
- Over-squashing is a topological issue: nodes at high access time (effective-resistance) struggle to exchange messages
- Relating over-squashing to access time and ER $\rightsquigarrow$ unified framework for graph-rewiring techniques

Towards new dynamic rewiring approaches

Some limitations of ‘static’ graph-rewiring techniques

Add edges among each pair of nodes within a certain distance (Graph-Transformers) (Abboud et al., 2022; Brüel-Gabrielsson et al., 2022; Ying et al., 2021; Rampášek et al., 2022) $\rightsquigarrow$ The new graph $\mathcal{R}(\mathsf{G})$ is much denser than G

- More sensitive to over-smoothing
- Higher impact on training time
- Loses inductive bias afforded by the distance

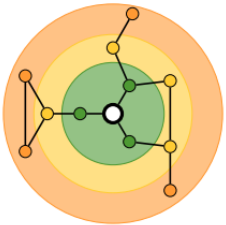
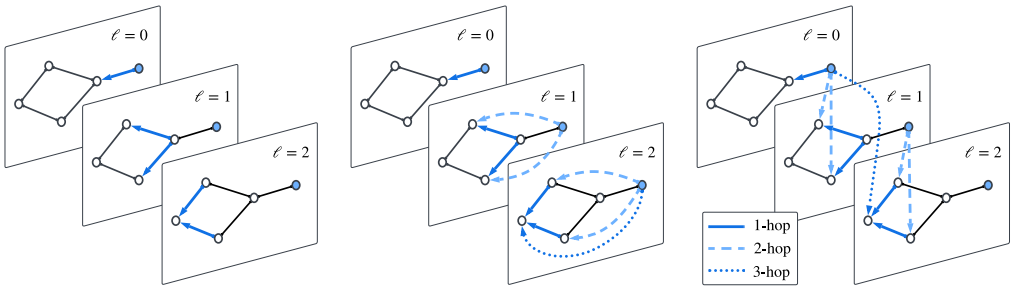


Figure 1: Figure from Abboud et al. (2022)

$\longleftrightarrow$

$\rightsquigarrow$ **New framework**: joint w/ Ben Gutteridge*, Xiaowen Dong, and Michael Bronstein

Dynamic edge-addition and delay: the new DREW and νDREW frameworks



(a) Classical MPNN (b) DREW − MPNN (c) νDREW − MPNN

- For classic MPNNs, information only travels from a node to its neighbours
- In DREW, at layer $r - 1$ we add edges connecting nodes at distance r
- In ν-DREW we introduce **delay depending on the distance between nodes**

ν-DREW in equations

Introduce $\tau_\nu(k) = \max(0, k - \nu)$ and consider the family of νDREW-MPNN

$$\mathbf{a}_{v,k}^{(t-1)} = \mathsf{agg}_k^{(t)}\left(\{\mathbf{h}_u^{(t-1-\tau_\nu(k))} : u \in \mathcal{N}_k(v)\}\right), 1 \leq k \leq t$$
$$\mathbf{h}_v^{(t)} = \mathsf{com}_k^{(t)}\left(\mathbf{h}_v^{(t-1)}, \mathbf{a}_{v,1}^{(t-1)}, \ldots, \mathbf{a}_{v,t}^{(t-1)}\right).$$

- The no delay case corresponds to $\nu = \infty$
- The agg and com are parameterised using weights depending on the layer t and the distance k

The advantages of νDREW

- The graph is progressively filled with each layer making for a more efficient framework

The advantages of νDREW

- The graph is progressively filled with each layer making for a more efficient framework
- We preserve the inductive bias provided by the input graph, since nodes that are closer still interact earlier

The advantages of νDREW

- The graph is progressively filled with each layer making for a more efficient framework
- We preserve the inductive bias provided by the input graph, since nodes that are closer still interact earlier
- The delay mechanism allow messages to travel both horizontally (i.e. across a layer) and vertically (i.e. across multiple layers) $\rightarrow$ we are 'rewiring' the graph by adding skip connections among different nodes based on their distance.

The advantages of νDREW

- The graph is progressively filled with each layer making for a more efficient framework
- We preserve the inductive bias provided by the input graph, since nodes that are closer still interact earlier
- The delay mechanism allow messages to travel both horizontally (i.e. across a layer) and vertically (i.e. across multiple layers) $\rightarrow$ we are 'rewiring' the graph by adding skip connections among different nodes based on their distance.
- Over-squashing is alleviated since now distant nodes can exchange messages directly

The advantages of νDREW

- The graph is progressively filled with each layer making for a more efficient framework
- We preserve the inductive bias provided by the input graph, since nodes that are closer still interact earlier
- The delay mechanism allow messages to travel both horizontally (i.e. across a layer) and vertically (i.e. across multiple layers) $\rightarrow$ we are 'rewiring' the graph by adding skip connections among different nodes based on their distance.
- Over-squashing is alleviated since now distant nodes can exchange messages directly
- We mitigate over-smoothing: a larger delay means that v aggregates the features from u before they are (significantly) 'smoothed' by repeated message passing

Long-Range-Graph-Benchmarks

In Dwivedi et al. (2022) new graph benchmarks have been introduced, with tasks likely to exhibit long-range dependencies

- Graph-Transformers were shown to be superior to standard MPNNs across all tasks
- The message was: we need Graph-Transformers whenever the task has long-range interactions

Can we beat Graph-Transformers on these tasks using the νDREW formalism?

The framework works: graph-level tasks

Model	FUNC	STRUCT
	Test AP ↑	Test MAE ↓
GCN	0.5930±0.0023	0.3496±0.0013
GINE	0.5498±0.0079	0.3547±0.0045
GatedGCN+RWSE	0.6069±0.0035	0.3357±0.0006
Transformer+LapPE	0.6326±0.0126	**0.2529±0.0016**
SAN+LapPE	0.6384±0.0121	*0.2683±0.0043*
SAN+RWSE	0.6439±0.0075	**0.2545±0.0012**
νDRew-GCN	0.6996±0.0076	0.2781±0.0028
νDRew-GCN+LapPE	**0.7150±0.0044**	**0.2536±0.0015**
νDRew-GIN+LapPE	*0.7126±0.0045*	*0.2697±0.0034*

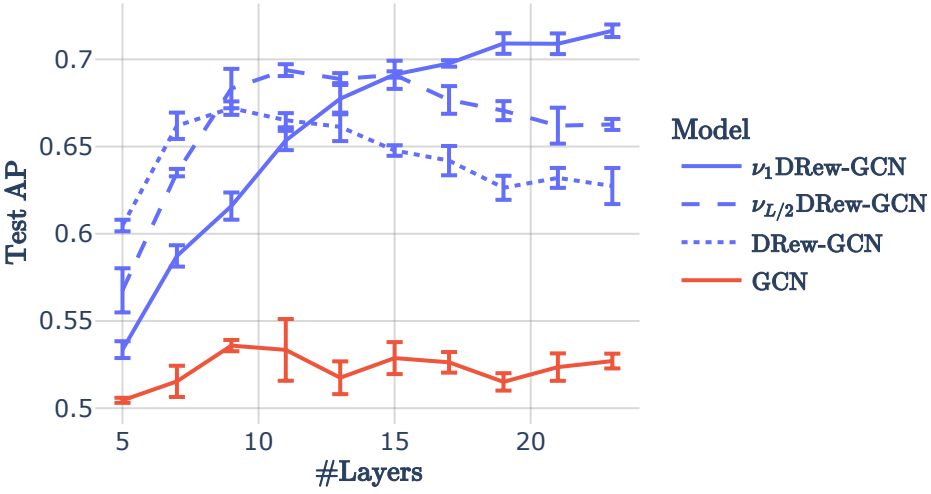
The framework works: node-level tasks

Model	PascalVOC-SP Test F1 ↑
GCN	0.1268±0.0060
GatedGCN	0.2873±0.0219
Transformer+LapPE	0.2694±0.0098
SAN+LapPE	*0.3230±0.0039*
DIGL+GatedGCN	0.2824±0.0039
νDRew-GCN	0.1848±0.0107
νDRew-GatedGCN	*0.3214±0.0021*
νDRew-GatedGCN+LapPE	**0.3314±0.0024**

The framework works: link-prediction

PCQM-CONTACT	
Model	**Test MRR ↑**
GCN	0.3234±0.0006
GINE	0.3180±0.0027
GatedGCN	0.3218±0.0011
GatedGCN+RWSE	0.3242±0.0008
Transformer+LapPE	0.3174±0.0020
SAN+LapPE	0.3350±0.0003
SAN+RWSE	0.3341±0.0006
DRew-GCN	**0.3444±0.0017**
DRew-GCN+LapPE	**0.3442±0.0006**

The framework works over deep models



The framework works: classical regression tasks on QM9

Property	R-GIN+FA	R-GAT+FA	SPN	DREW-GIN	ν_1DREW-GIN
mu	2.54±0.09	2.73±0.07	2.32±0.28	**1.95±0.07**	1.98±0.01
alpha	2.28±0.04	2.32±0.16	1.77±0.09	**1.64±0.03**	1.67±0.02
HOMO	1.26±0.02	1.43±0.02	1.26±0.09	**1.17±0.01**	**1.16±0.01**
LUMO	1.34±0.04	1.41±0.03	1.19±0.05	**1.12±0.02**	1.15±0.01
gap	1.96±0.04	2.08±0.05	1.89±0.11	1.74±0.02	**1.72±0.01**
R2	12.61±0.37	15.76±1.17	10.66±0.40	**9.40±0.12**	9.96±0.1
ZPVE	5.03±0.36	5.98±0.43	2.77±0.17	**2.73±0.19**	2.91±0.31
U0	2.21±0.12	2.19±0.25	1.12±0.13	1.01±0.08	**0.98±0.04**
U	2.32±0.18	2.11±0.10	**1.03±0.09**	**1.02±0.07**	**1.03±0.07**
H	2.26±0.19	2.27±0.29	**1.05±0.04**	1.12±0.06	**1.04±0.07**
G	2.04±0.24	2.07±0.07	**0.97±0.06**	1.01±0.17	1.03±0.02
Cv	1.86±0.03	2.03±0.14	1.36±0.06	**1.24±0.02**	**1.25±0.02**

What’s next and the bigger picture

The bigger picture: the benchmarks

Do the existing benchmarks have long-range interactions? Common datasets like ZINC, OGBG most likely don’t

The bigger picture: the benchmarks

Do the existing benchmarks have long-range interactions? Common datasets like ZINC, OGBG most likely don't

Most rewiring approaches work as follows:

- Propose rewiring $\mathcal{R}$
- Show that $\mathsf{MPNN} \circ \mathcal{R}$ is better than MPNN
- Conclude that over-squashing is being mitigated

This is a dangerous approach

The bigger picture: graph-rewiring beyond over-squashing

It is easy[2] to verify that given a 'standard' non-local class $\{\mathrm{GNN}_\theta\}$:

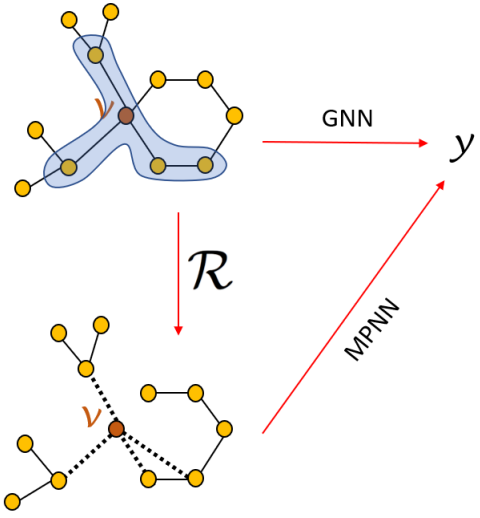
- There exists a **rewiring** map $\mathcal{R} : (\mathsf{G}, \mathbf{H}) \mapsto \mathcal{R}(\mathsf{G}, \mathbf{H})$ such that

$$\mathrm{GNN}_\theta(\mathsf{G}, \mathbf{H}) = \mathrm{MPNN}_\theta^{(m)}(\mathcal{R}(\mathsf{G}, \mathbf{H}), \mathbf{H})$$

- Typically $\mathcal{R}(\mathsf{G}) = (\tilde{\mathsf{V}}, \tilde{\mathsf{E}}, \mathcal{W})$, for some edge weights $\mathcal{W}$ depending on GNN_θ

[2] Veličković (2022)

Non-local MPNNs



A paradigm shift

$$\mathrm{GNN}_\theta(\mathsf{G}, \mathbf{H}_0) = \mathrm{MPNN}_\theta^{(m)}(\mathcal{R}(\mathsf{G}), \mathbf{H}_0)$$

Focus on GNN_θ to be *more expressive* $\rightarrow$ $\mathcal{R}$ is often geometrically meaningless

- Loss of information (what is local/non-local) unless 'augmentations'
- $\mathcal{R}(\mathsf{G})$ is usually *denser* $\rightarrow$ models not scalable and/or prone to *over-smoothing*
- Limitations of local MPNNs like *over-squashing* originate from the topology of G, why not choosing $\mathcal{R}$ to 'improve' the topology?
- Do we actually need G and what information out of G is actually necessary in the message-passing?

Thank you!

For any question do not hesitate to contact me:

Email : fd405 at cam.ac.uk

Social : @*Francesco_dgv*

References

Abboud, R., Dimitrov, R., and Ceylan, İ. İ. (2022). Shortest path networks for graph property prediction. *arXiv preprint arXiv:2206.01003*.

Alon, U. and Yahav, E. (2021). On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*.

Arnaiz-Rodríguez, A., Begga, A., Escolano, F., and Oliver, N. (2022). Diffwire: Inductive graph rewiring via the lov\'asz bound. *arXiv preprint arXiv:2206.07369*.

Black, M., Nayyeri, A., Wan, Z., and Wang, Y. (2023). Understanding oversquashing in gnns through the lens of effective resistance. *arXiv preprint arXiv:2302.06835*.

Bodnar, C., Frasca, F., Otter, N., Wang, Y., Lio, P., Montufar, G. F., and Bronstein, M. (2021). Weisfeiler and lehman go cellular: Cw networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 2625–2640.

Brüel-Gabrielsson, R., Yurochkin, M., and Solomon, J. (2022). Rewiring with positional encodings for graph neural networks. *arXiv preprint arXiv:2201.12674*.

Deac, A., Lackenby, M., and Veličković, P. (2022). Expander graph propagation. *arXiv preprint arXiv:2210.02997*.

Devriendt, K. and Lambiotte, R. (2022). Discrete curvature on graphs from the effective resistance. *Journal of Physics: Complexity*.

Dwivedi, V. P., Rampášek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., and Beaini, D. (2022). Long range graph benchmark. *arXiv preprint arXiv:2206.08164*.

Karhadkar, K., Banerjee, P. K., and Montufar, G. (2022). Fosr: First-order spectral rewiring for addressing oversquashing in gnns. *arXiv preprint arXiv:2210.11790*.

Lovász, L. (1993). Random walks on graphs. *Combinatorics, Paul erdos is eighty*, 2(1-46):4.

Ollivier, Y. (2007). Ricci curvature of metric spaces. *Comptes Rendus Mathematique*, 345(11):643–646.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. (2022). Recipe for a general, powerful, scalable graph transformer. *arXiv preprint arXiv:2205.12454*.

Topping, J., Di Giovanni, F., Chamberlain, B. P., Dong, X., and Bronstein, M. M. (2022). Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*.

Veličković, P. (2022). Message passing all the way up. *arXiv preprint arXiv:2202.11097*.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., and Jegelka, S. (2018). Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pages 5453–5462. PMLR.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. (2021). Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888.