Curvature and over-squashing in Graph Neural Networks

Francesco Di Giovanni

Twitter

First Italian School in GDL: July 25–28, Pescara

Overview of the presentation

- Review of MPNN problems with focus on **over-squashing**

Overview of the presentation

- Review of MPNN problems with focus on **over-squashing**
- (Long) discussion on curvature on graphs: old and new

Overview of the presentation

- Review of MPNN problems with focus on **over-squashing**
- (Long) discussion on curvature on graphs: old and new
- How curvature helps understanding information flow in MPNNs

Overview of the presentation

- Review of MPNN problems with focus on **over-squashing**
- (Long) discussion on curvature on graphs: old and new
- How curvature helps understanding information flow in MPNNs
- *Graph-rewiring* and future directions

Introduction

Preliminaries on graph operators

- $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ is an *undirected* graph with $|\mathsf{V}| = n$ and $i \sim j$ if $(i, j) \in \mathsf{E}$

Preliminaries on graph operators

- $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ is an *undirected* graph with $|\mathsf{V}| = n$ and $i \sim j$ if $(i, j) \in \mathsf{E}$
- $\mathbf{A}, \mathbf{D}$ are the adjacency and (diagonal) degree matrices

Preliminaries on graph operators

- $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ is an *undirected* graph with $|\mathsf{V}| = n$ and $i \sim j$ if $(i, j) \in \mathsf{E}$
- $\mathbf{A}, \mathbf{D}$ are the adjacency and (diagonal) degree matrices
- We write $d_i := \mathbf{D}_{ii}$ for the degree of node i

Preliminaries on graph operators

- $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ is an *undirected* graph with $|\mathsf{V}| = n$ and $i \sim j$ if $(i, j) \in \mathsf{E}$
- $\mathbf{A}, \mathbf{D}$ are the adjacency and (diagonal) degree matrices
- We write $d_i := \mathbf{D}_{ii}$ for the degree of node i
- $d_{\mathsf{G}}(i, j)$ is the *shortest walk* distance between nodes i, j

Preliminaries on graph operators

- $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ is an *undirected* graph with $|\mathsf{V}| = n$ and $i \sim j$ if $(i, j) \in \mathsf{E}$
- $\mathbf{A}, \mathbf{D}$ are the adjacency and (diagonal) degree matrices
- We write $d_i := \mathbf{D}_{ii}$ for the degree of node i
- $d_{\mathsf{G}}(i, j)$ is the *shortest walk* distance between nodes i, j
- $S_r(i) : \{j \in \mathsf{V} : \ d_{\mathsf{G}}(i, j) = r\}$

Preliminaries on graph operators

- The *normalized* adjacency is $\bar{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$

Preliminaries on graph operators

- The *normalized* adjacency is $\bar{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$
- The Laplacian $\boldsymbol{\Delta} = \mathbf{I} - \bar{\mathbf{A}}$ is an operator acting on signals $\mathbf{f} : \mathsf{V} \to \mathbb{R}$ as

$$\boxed{(\boldsymbol{\Delta}\mathbf{f})_i = f_i - \sum_{j \sim i} \frac{f_j}{\sqrt{d_i d_j}}}$$

Preliminaries on graph operators

- The *normalized* adjacency is $\bar{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$
- The Laplacian $\boldsymbol{\Delta} = \mathbf{I} - \bar{\mathbf{A}}$ is an operator acting on signals $\mathbf{f} : \mathsf{V} \to \mathbb{R}$ as

$$\boxed{(\boldsymbol{\Delta}\mathbf{f})_i = f_i - \sum_{j\sim i} \frac{f_j}{\sqrt{d_i d_j}}}$$

The Laplacian $\boldsymbol{\Delta} \succeq 0 \to$ eigenvalues satisfy $0 = \lambda_0^{\boldsymbol{\Delta}} \leq \ldots \leq \lambda_{n-2}^{\boldsymbol{\Delta}} \leq \rho_{\boldsymbol{\Delta}}$, with $\rho_{\boldsymbol{\Delta}} \leq 2$, and are called (graph) *frequencies*, eigenvectors are denoted by $\{\phi_\ell^{\boldsymbol{\Delta}}\}_{\ell=0}^{n-1}$

Information encoded in λ_ℓ^{Δ}[1]

- The multiplicity $\lambda_0^{\Delta} = 0$ represents the number of connected components of G

[1] Chung and Graham (1997)

Information encoded in $\lambda_\ell^{\boldsymbol{\Delta}}$[1]

- The multiplicity $\lambda_0^{\boldsymbol{\Delta}} = 0$ represents the number of connected components of G
- $S \subset \mathsf{V}$, $\partial S = \{(i,j) : i \in S,\ j \in \mathsf{V} \setminus S\}$ and $\mathrm{vol}(S) = \sum_{i \in S} d_i$.

$$h_{\mathsf{G}} := \min_{S \subset \mathsf{V}} \frac{|\partial S|}{\min\{\mathrm{vol}(S), \mathrm{vol}(\mathsf{V} \setminus S)\}} \implies \textbf{Cheeger inequality} : 2h_{\mathsf{G}} \geq \lambda_1 \geq \frac{h_{\mathsf{G}}^2}{2}$$

$\rightarrow \lambda_1^{\boldsymbol{\Delta}} := \mathrm{gap}(\boldsymbol{\Delta})$ measures the ‘required energy’ to separate G into two communities

[1] Chung and Graham (1997)

Information encoded in $\lambda_\ell^{\mathbf{\Delta}}$[1]

- The multiplicity $\lambda_0^{\mathbf{\Delta}} = 0$ represents the number of connected components of G
- $S \subset \mathsf{V}$, $\partial S = \{(i,j) : i \in S,\ j \in \mathsf{V} \setminus S\}$ and $\mathrm{vol}(S) = \sum_{i \in S} d_i$.

$$h_{\mathsf{G}} := \min_{S \subset \mathsf{V}} \frac{|\partial S|}{\min\{\mathrm{vol}(S), \mathrm{vol}(\mathsf{V} \setminus S)\}} \implies \textbf{Cheeger inequality} : 2h_{\mathsf{G}} \geq \lambda_1 \geq \frac{h_{\mathsf{G}}^2}{2}$$

$\rightarrow \lambda_1^{\mathbf{\Delta}} := \mathrm{gap}(\mathbf{\Delta})$ measures the 'required energy' to separate G into two communities

- $2 - \rho_{\mathbf{\Delta}}$ measures the deviation of G from a *bipartite* graph

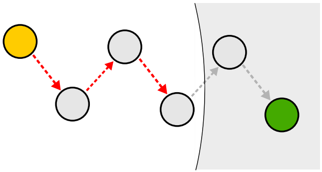
[1] Chung and Graham (1997)

The formalism of MPNNs

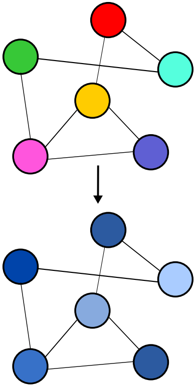
- Graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$
- $\mathbf{F}_{\text{input}} \in \mathbb{R}^{n \times p}$ matrix representation of input node features, with rows $\{(\mathbf{f}_i)_{\text{input}}^\top\}_{i=1}^n$
- Encoding map $\psi_{\text{EN}} : \mathbb{R}^p \to \mathbb{R}^{d_0}$
- Update functions $\{\phi_{\text{UP}}^t : \mathbb{R}^{d_t} \to \mathbb{R}^{d_{t+1}}\}$ for $0 \leq t \leq T-1$, with T the *depth*

$$\text{MPNN}: \quad \mathbf{f}_i(t+1) = \phi_{\text{UP}}^t\left(\mathbf{f}_i(t), \{\{\mathbf{f}_j(t) : \; j \sim i\}\}\right), \quad \mathbf{f}_i(0) = \psi_{\text{EN}}((\mathbf{f}_i)_{\text{input}}).$$

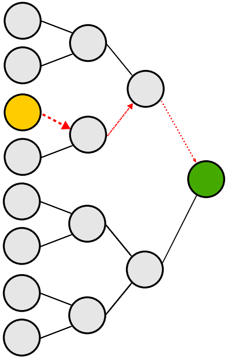
Common problems in MPNNs



Under-reaching

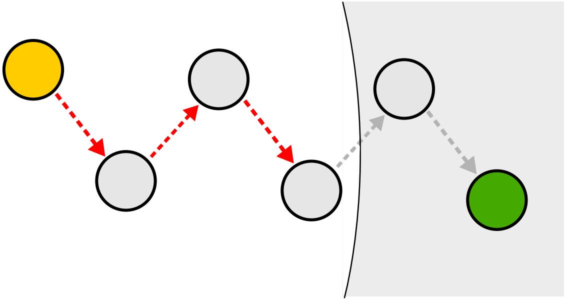


Over-smoothing



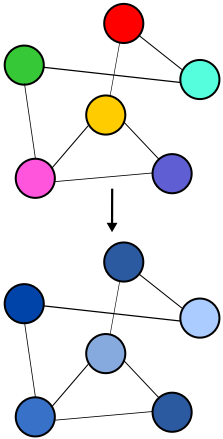
Over-squashing

Under-reaching



- Information cannot propagate further than there are layers in the MPNN (shown above with three layers): *in principle this can be fixed by increasing depth*

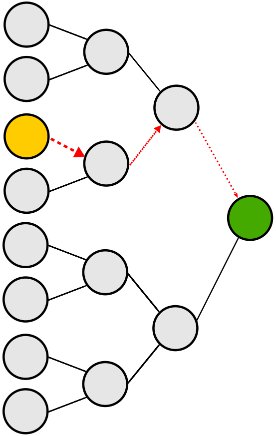
Over-smoothing



- In a deep MPNN, node representations can become similar (smoothed out) and weaken influence of graph structure

- *In principle, this can be fixed by choosing message passing functions that do not act as low-pass filters and is* ***independent of the topology***

The over-squashing phenomenon[2]



- Depending on the graph-topology, the size of the $r-$hop of a node may **grow exponentially**
- As messages are sent through the 'same structural edges' via *fixed-size* node representations we lose information

[2] Alon and Yahav (2021)

Understanding over-squashing

Questions:

- Can we formalize over-squashing a bit better?

Understanding over-squashing

Questions:

- Can we formalize over-squashing a bit better?
- Where in the graph the information is ‘getting stuck’?

Questions:

- Can we formalize over-squashing a bit better?
- Where in the graph the information is ‘getting stuck’?
- What are viable solutions?

Understanding over-squashing

Questions:

- Can we formalize over-squashing a bit better?
- Where in the graph the information is 'getting stuck'?
- What are viable solutions?

Upshot:

- Use sensitivity analysis to monitor information propagation

Understanding over-squashing

Questions:

- Can we formalize over-squashing a bit better?
- Where in the graph the information is 'getting stuck'?
- What are viable solutions?

Upshot:

- Use sensitivity analysis to monitor information propagation
- **Negatively curved** edges are responsible for the over-squashing phenomenon

Understanding over-squashing

Questions:

- Can we formalize over-squashing a bit better?
- Where in the graph the information is 'getting stuck'?
- What are viable solutions?

Upshot:

- Use sensitivity analysis to monitor information propagation
- **Negatively curved** edges are responsible for the over-squashing phenomenon
- Over-squashing only depends on topology and is unavoidable for MPNNs $\rightarrow$ how about changing the graph?

Ricci curvature on manifolds

Consider a Riemannian manifold $(M, g) \to p \in M, \mathbf{v}, \mathbf{w} \in T_pM : g_p(\mathbf{v}, \mathbf{w}) = 0$

Ricci curvature on manifolds

Consider a Riemannian manifold $(M, g) \to p \in M, \mathbf{v}, \mathbf{w} \in T_pM : g_p(\mathbf{v}, \mathbf{w}) = 0$

Sectional curvature $K_p(\mathbf{v}, \mathbf{w})$ is the *Gaussian curvature* of the surface with tangent plane at p spanned by $\mathbf{v}, \mathbf{w}$

Ricci curvature on manifolds

Consider a Riemannian manifold $(M, g) \to p \in M, \mathbf{v}, \mathbf{w} \in T_pM : g_p(\mathbf{v}, \mathbf{w}) = 0$

Sectional curvature $K_p(\mathbf{v}, \mathbf{w})$ is the *Gaussian curvature* of the surface with tangent plane at p spanned by $\mathbf{v}, \mathbf{w}$

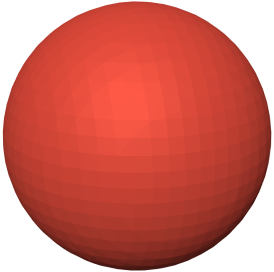
Ricci curvature $\mathrm{Ric}_p : T_pM \times T_pM \to \mathbb{R}$ bilinear map

Given $\mathbf{v}$ unit vector in $T_pM \to \{\mathbf{v}, \mathbf{e}_2, \ldots, \mathbf{e}_n\} \subset T_pM$ *orthonormal basis*

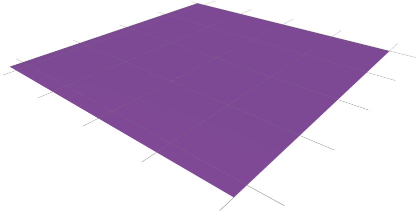
$$\boxed{\mathrm{Ric}_p(\mathbf{v}, \mathbf{v}) = \sum_{i=2}^{n} K_p(\mathbf{v}, \mathbf{e}_i)}$$

Ricci curvature of space-forms

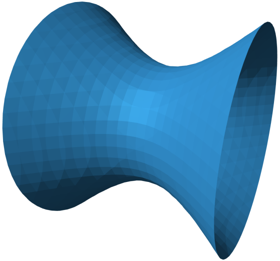
- *Ricci curvature* on ‘prototypical’ manifolds



Spherical (> 0)



Euclidean ($= 0$)



Hyperbolic (< 0)

What is the meaning of Ricci curvature? Part I

- On a sphere: $|B(p, r)| \sim C$
- In Euclidean space: $|B(p, r)| \sim \mathrm{poly}(r)$
- In Hyperbolic space: $|B(p, r)| \sim \exp(r)$

What is the meaning of Ricci curvature? Part I

- On a sphere: $|B(p,r)| \sim C$
- In Euclidean space: $|B(p,r)| \sim \mathrm{poly}(r)$
- In Hyperbolic space: $|B(p,r)| \sim \exp(r)$

Volume comparison results:

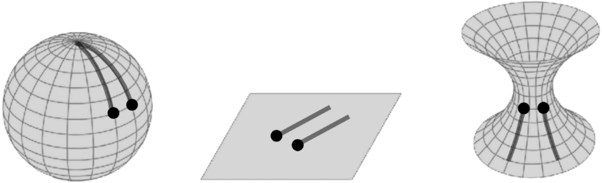
Theorem (Bishop-Cheeger-Gromov)

If $\mathrm{Ric} \geq (n-1)k$, *then* $r \mapsto |B(p,r)|/v(n,k,r)$ *is a nonincreasing function, with* $v(n,k,r)$ *the volume of space-form of dimension* n, *constant curvature* k *and radius* r.

What is the meaning of Ricci curvature? Part II

Pick geodesics starting at nearby points with *parallel* velocity

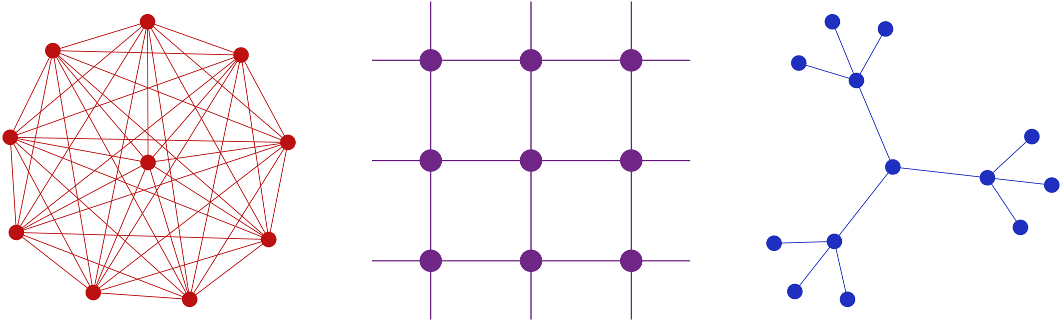
- With **positive curvature** geodesics **converge**
- With **zero curvature**, geodesics stay **parallel**
- With **negative curvature**, geodesics **diverge**



Sperical (>0) Euclidean (=0) Hyperbolic (<0)

Intuition

- Swapping geodesics for edges, we can take discrete analogues on graphs



Clique (> 0) Grid ($= 0$) Tree (< 0)

What is Ric capturing then? A transport point of view

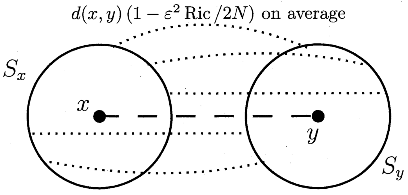


Figure 1: Figure taken from Ollivier (2009)

Let $x \in M$, $\mathbf{v} \in T_xM$ and $\gamma_{x,v}$ the geodesic starting at x with initial velocity $\mathbf{v}$

What is Ric capturing then? A transport point of view

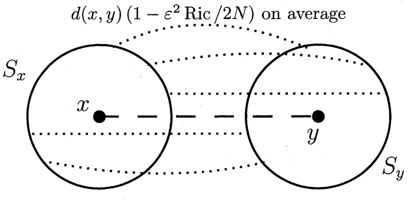


Figure 1: Figure taken from Ollivier (2009)

Let $x \in M$, $\mathbf{v} \in T_x M$ and $\gamma_{x,v}$ the geodesic starting at x with initial velocity $\mathbf{v}$

Theorem (Ollivier)

Let $\epsilon, \delta > 0$. Let $S_x = \{\gamma_{x,\mathbf{v}}(\epsilon) : \ \mathbf{v} \in T_x M, \ |\mathbf{v}| = 1\}$ and similarly for S_y with $y = \gamma_{x,\mathbf{v}}(\delta)$. If we map S_x to S_y using parallel transport, the average travelled distance is

$$\delta\left(1 - \frac{\epsilon^2}{2n}\mathrm{Ric}|_x(v,v) + \mathcal{O}(\epsilon^3 + \epsilon^2\delta)\right), \quad \delta, \epsilon \to 0.$$

Can we extend the same idea to graphs?[3]

In a nutshell: if Ric is positive (negative), balls are closer (farther) than their centers

[3] Description based on Samal et al. (2018)

Can we extend the same idea to graphs?[3]

In a nutshell: if Ric is positive (negative), balls are closer (farther) than their centers

Represent balls with *volume measures* $\to$

Idea: *how does distance between measures compare with distance between centers?*

- Given a metric space, use underlying structure to measure distance among points
- For distance among measures, use **Wasserstein** metric

[3] Description based on Samal et al. (2018)

Ollivier curvature on graphs

For $i \in V$ and $\alpha \in [0,1) \to$ *lazy RW*-probability measure

$$\boxed{\mu_i^\alpha : j \mapsto \begin{cases} \alpha,\ \ j = i \\ \frac{1-\alpha}{d_i},\ \ j \sim i, \\ 0,\ \ \text{otherwise} \end{cases}}$$

Ollivier curvature on graphs

For $i \in V$ and $\alpha \in [0,1) \to$ *lazy RW*-probability measure

$$\mu_i^\alpha : j \mapsto \begin{cases} \alpha, \ j = i \\ \frac{1-\alpha}{d_i}, \ j \sim i, \\ 0, \ \text{otherwise} \end{cases}$$

Problem: what is the coupling that moves mass from μ_i^α to μ_j^α while *minimizing* the travelled distance?

Ollivier curvature on graphs

The transportation distance between $\mu_i^\alpha, \mu_j^\alpha$ is

$$\boxed{W_1(\mu_i^\alpha, \mu_j^\alpha) := \inf_M \sum_{k \in S_1(i)} \sum_{w \in S_1(j)} M_{kw} d_G(k, w),}$$

where $\inf$ is over M satisfying the **marginal constraints**:

$$\sum_{k \in S_1(i)} M_{kw} = \mu_j^\alpha(w), \qquad \sum_{w \in S_1(j)} M_{kw} = \mu_i^\alpha(k).$$

Ollivier curvature on graphs

Definition (Lin et al.)

Given $i \sim j$ we define the α-*Ollivier curvature* by

$$\kappa_\alpha(i,j) := 1 - \frac{W_1(\mu_i^\alpha, \mu_j^\alpha)}{d_{\mathsf{G}}(i,j)}.$$

Since $\kappa_\alpha(1-\alpha)^{-1}$ is increasing and bounded the quantity below is well-defined:

$$\kappa(i,j) := \lim_{\alpha \to 1} \frac{\kappa_\alpha(i,j)}{1-\alpha}.$$

Does it make sense?

Analogy with results on Ric in the continuous case

Theorem (Lin et al.)

Let $\kappa(i,j) \geq \kappa > 0$ *for each edge* $(i,j) \in \mathsf{E}$. *Then*

- *The diameter of the graph is bounded by* $\mathrm{diam}(G) \leq \frac{2}{\kappa}$
- *The spectral gap is controlled from below:* $\mathrm{gap}(\boldsymbol{\Delta}) \geq \kappa$

What is κ capturing then?

Informal characterization: $\kappa(i, j)$ measures the structural importance of (i, j) for the connectedness of $S_1(i) \cup S_1(j)$

What is κ capturing then?

Informal characterization: $\kappa(i,j)$ measures the structural importance of (i,j) for the connectedness of $S_1(i) \cup S_1(j)$

$\rightarrow$ The more positive $\kappa(i,j)$ the more overlapping between $S_1(i)$ and $S_1(j)$

What is κ capturing then?

Informal characterization: $\kappa(i,j)$ measures the structural importance of (i,j) for the connectedness of $S_1(i) \cup S_1(j)$

$\rightarrow$ The more positive $\kappa(i,j)$ the more overlapping between $S_1(i)$ and $S_1(j)$

$\rightarrow$ The more negative $\kappa(i,j)$ the fewer shortcuts between $S_1(i)$ and $S_1(j)$

What is κ capturing then?

Informal characterization: $\kappa(i,j)$ measures the structural importance of (i,j) for the connectedness of $S_1(i) \cup S_1(j)$

$\rightarrow$ The more positive $\kappa(i,j)$ the more overlapping between $S_1(i)$ and $S_1(j)$

$\rightarrow$ The more negative $\kappa(i,j)$ the fewer shortcuts between $S_1(i)$ and $S_1(j)$

$\kappa(i,j)$ **is 'local' and will not detect if there are cycles with length** > 5 **based at** (i,j)

Expressive power vs computational cost

Ollivier is expressive $\to \kappa(i, j)$ function of cycles of up to length 5 based at (i, j)

Expressive power vs computational cost

Ollivier is expressive $\rightarrow \kappa(i,j)$ function of cycles of up to length 5 based at (i,j)

Caveat: Complexity of $\kappa : \mathsf{E} \rightarrow \mathbb{R}$ is $\mathcal{O}(|E| d_{\max}^3)$

Expressive power vs computational cost

Ollivier is expressive $\to$ $\kappa(i,j)$ function of cycles of up to length 5 based at (i,j)

Caveat: Complexity of $\kappa : \mathsf{E} \to \mathbb{R}$ is $\mathcal{O}(|E|d_{\max}^3)$

Can we approximate Ollivier? Let $\sharp_\Delta(i,j) := |S_1(i) \cap S_1(j)|$

Theorem (Jost and Liu)

If $\min\{d_i, d_j\} > 1$, *then*

$$\kappa(i,j) \geq \frac{2}{d_i} + \frac{2}{d_j} - 2 + 2\frac{|\sharp_\Delta(i,j)|}{\max\{d_i,d_j\}} + \frac{|\sharp_\Delta(i,j)|}{\min\{d_i,d_j\}}.$$

Forman curvature

Other curvature candidates? **(Augmented) Forman curvature**[4]

$$\boxed{\mathrm{F}(i,j) = 4 - d_i - d_j + 3\,\sharp_{\Delta}(i,j)}$$

[4] Forman (2003); Samal et al. (2018)

Forman curvature

Other curvature candidates? **(Augmented) Forman curvature**[4]

$$\boxed{\mathrm{F}(i,j) = 4 - d_i - d_j + 3\,\sharp_{\Delta}(i,j)}$$

- Computationally cheap..
- ..but limited power (can only distinguish triangles, gives grids negative curvature)

Can we strike a balance?

[4] Forman (2003); Samal et al. (2018)

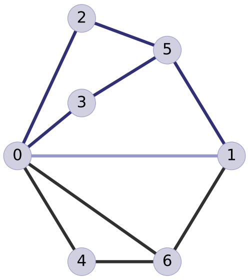
Balanced Forman curvature: preliminaries

(i) $\sharp_{\Delta}(i,j) := S_1(i) \cap S_1(j)$ are the triangles based at $i \sim j$.

(ii) $\sharp_{\square}^{i}(i,j)$ are neighbors of i forming a 4-cycle based at $i \sim j$ *without* diagonals

(iii) $\gamma_{\max}(i,j)$ is the maximal number of 4 cycles based at $i \sim j$ traversing a common node.

The degeneracy factor $\gamma_{\max}(0,1) = 2$ since there exist two 4 cycles passing the same node (5)



Balanced Forman curvature: preliminaries

(i) $\sharp_{\Delta}(i,j) := S_1(i) \cap S_1(j)$ are the triangles based at $i \sim j$

(ii) $\sharp_{\square}^{i}(i,j)$ are neighbors of i forming a 4-cycle based at $i \sim j$ *without* diagonals

(iii) $\gamma_{\max}(i,j)$ is the maximal number of 4 cycles based at $i \sim j$ traversing a common node

We introduce a new[5] combinatorial curvature named **Balanced Forman**

[5] Topping*, Di G.*, et al. (2021)

Balanced Forman curvature: preliminaries

(i) $\sharp_{\Delta}(i,j) := S_1(i) \cap S_1(j)$ are the triangles based at $i \sim j$

(ii) $\sharp_{\square}^{i}(i,j)$ are neighbors of i forming a 4-cycle based at $i \sim j$ *without* diagonals

(iii) $\gamma_{\max}(i,j)$ is the maximal number of 4 cycles based at $i \sim j$ traversing a common node

We introduce a new[5] combinatorial curvature named **Balanced Forman**

$$\mathrm{BF}(i,j) := \frac{2}{d_i} + \frac{2}{d_j} - 2$$

[5] Topping*, Di G.*, et al. (2021)

Balanced Forman curvature: preliminaries

(i) $\sharp_{\Delta}(i,j) := S_1(i) \cap S_1(j)$ are the triangles based at $i \sim j$

(ii) $\sharp_{\square}^{i}(i,j)$ are neighbors of i forming a 4-cycle based at $i \sim j$ *without* diagonals

(iii) $\gamma_{\max}(i,j)$ is the maximal number of 4 cycles based at $i \sim j$ traversing a common node

We introduce a new[6] combinatorial curvature named **Balanced Forman**

$$\mathrm{BF}(i,j) := \frac{2}{d_i} + \frac{2}{d_j} - 2 + 2\frac{|\sharp_{\Delta}(i,j)|}{\max\{d_i,d_j\}} + \frac{|\sharp_{\Delta}(i,j)|}{\min\{d_i,d_j\}}$$

[6] Topping*, Di G.*, et al. (2021)

Balanced Forman curvature: preliminaries

(i) $\sharp_\Delta(i,j) := S_1(i) \cap S_1(j)$ are the triangles based at $i \sim j$

(ii) $\sharp^i_\square(i,j)$ are neighbors of i forming a 4-cycle based at $i \sim j$ *without* diagonals

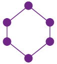
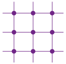
(iii) $\gamma_{\max}(i,j)$ is the maximal number of 4 cycles based at $i \sim j$ traversing a common node

We introduce a new[7] combinatorial curvature named **Balanced Forman**

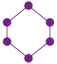
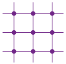
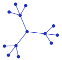
$$\mathrm{BF}(i,j) := \frac{2}{d_i} + \frac{2}{d_j} - 2 + 2\frac{|\sharp_\Delta(i,j)|}{\max\{d_i,d_j\}} + \frac{|\sharp_\Delta(i,j)|}{\min\{d_i,d_j\}} + \frac{\gamma_{\max}^{-1}(i,j)}{\max\{d_i,d_j\}}(|\sharp^i_\square(i,j) + \sharp^j_\square(i,j))$$

[7] Topping*, Di G.*, et al. (2021)

Does it make sense?

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

Does it make sense?

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

Theorem (Topping*, Di G.*, et al.)

Given an unweighted graph G, for any edge $i \sim j$ we have $\kappa(i,j) \geq \mathrm{BF}(i,j)$.

$\rightarrow$ Generalizes Jost and Liu (2014) to include 4-cycle contributions

Other curvature notions

- Bakry-Emery: rich theory thanks to its formulation[8] but computationally expensive

[8] Keller and Münch (2018)
[9] Devriendt and Lambiotte (2022)

Other curvature notions

- Bakry-Emery: rich theory thanks to its formulation[8] but computationally expensive
- Effective resistance curvature[9]: this is expressive but global in nature meaning that value along an edge affected by distant nodes

[8] Keller and Münch (2018)
[9] Devriendt and Lambiotte (2022)

Over-squashing and curvature

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

ABSTRACT

Most graph neural networks (GNNs) use the message passing paradigm, in which node features are propagated on the input graph. Recent works pointed to the distortion of information flowing from distant nodes as a factor limiting the efficiency of message passing for tasks relying on long-distance interactions. This phenomenon, referred to as 'over-squashing', has been heuristically attributed to graph bottlenecks where the number of k-hop neighbors grows rapidly with k. We provide a precise description of the over-squashing phenomenon in GNNs and analyze how it arises from bottlenecks in the graph. For this purpose, we introduce a new edge-based combinatorial curvature and prove that negatively curved edges are responsible for the over-squashing issue. We also propose and experimentally test a curvature-based graph rewiring method to alleviate the over-squashing.

Figure 2: Our work received an *outstanding paper honorable mention* at ICLR22!

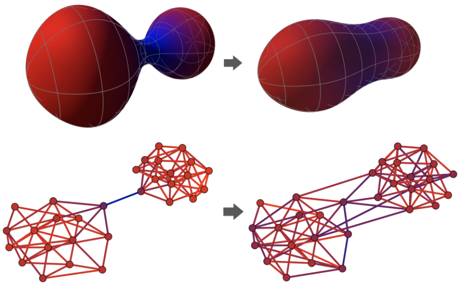
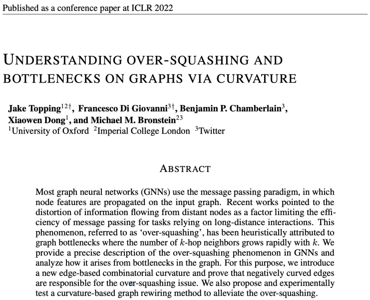


Figure 3: An example of a curvature-inspired flow to improve the propagation of information and alleviate over-squashing

Over-squashing and curvature


Published as a conference paper at ICLR 2022

UNDERSTANDING OVER-SQUASHING AND BOTTLENECKS ON GRAPHS VIA CURVATURE

Jake Topping[12†], **Francesco Di Giovanni**[3†], **Benjamin P. Chamberlain**[3], **Xiaowen Dong**[1], **and Michael M. Bronstein**[23]
[1]University of Oxford [2]Imperial College London [3]Twitter

ABSTRACT

Most graph neural networks (GNNs) use the message passing paradigm, in which node features are propagated on the input graph. Recent works pointed to the distortion of information flowing from distant nodes as a factor limiting the efficiency of message passing for tasks relying on long-distance interactions. This phenomenon, referred to as 'over-squashing', has been heuristically attributed to graph bottlenecks where the number of k-hop neighbors grows rapidly with k. We provide a precise description of the over-squashing phenomenon in GNNs and analyze how it arises from bottlenecks in the graph. For this purpose, we introduce a new edge-based combinatorial curvature and prove that negatively curved edges are responsible for the over-squashing issue. We also propose and experimentally test a curvature-based graph rewiring method to alleviate the over-squashing.

Figure 2: Our work received an *outstanding paper honorable mention* at ICLR22!

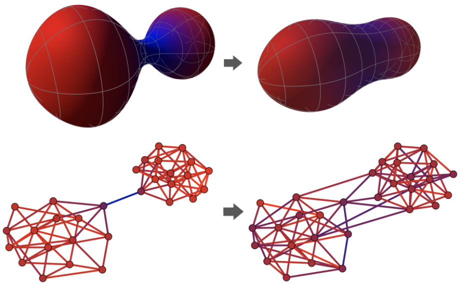


Figure 3: An example of a curvature-inspired flow to improve the propagation of information and alleviate over-squashing

Joint with J. Topping, B. Chamberlain, X. Dong, and M. Bronstein

Sensitivity analysis

- Symmetrically normalized adjacency $\rightarrow \bar{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$
- Message functions $\{\psi_t\}$ and update functions $\{\phi_t\}$

$$\text{MPNN}: \quad \mathbf{f}_i^{(t+1)} = \phi_t\left(\mathbf{f}_i^{(t)}, \sum_{j=1}^{n} \bar{\mathbf{A}}_{ij}\psi_t(\mathbf{f}_i^{(t)}, \mathbf{f}_j^{(t)})\right)$$

Sensitivity analysis

- Symmetrically normalized adjacency $\rightarrow \bar{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$
- Message functions $\{\psi_t\}$ and update functions $\{\phi_t\}$

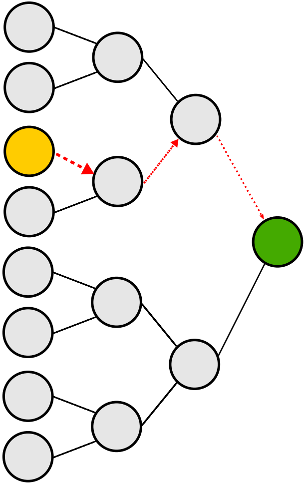
$$\text{MPNN}: \quad \mathbf{f}_i^{(t+1)} = \phi_t\left(\mathbf{f}_i^{(t)}, \sum_{j=1}^{n} \bar{\mathbf{A}}_{ij}\psi_t(\mathbf{f}_i^{(t)}, \mathbf{f}_j^{(t)})\right)$$

Lemma (Topping*, Di G.*, et al.)

Let $i, s \in \mathsf{V}$ *with* $d_G(i, s) = T + 1$. *If* $|\nabla\phi_t| \leq \alpha$ *and* $|\nabla\psi_t| \leq \beta$ *for* $0 \leq t \leq T$, *then*

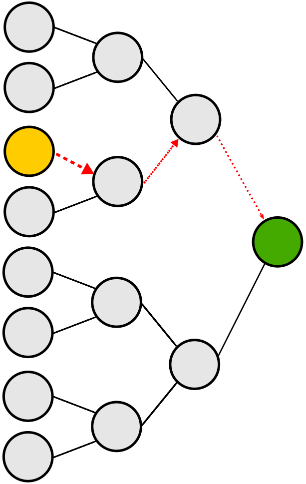
$$\left|\frac{\partial \mathbf{f}_i^{(T+1)}}{\partial \mathbf{f}_s^{(0)}}\right| \leq (\alpha\beta)^{T+1}(\bar{\mathbf{A}}^{T+1})_{is}.$$

Over-squashing example: binary tree



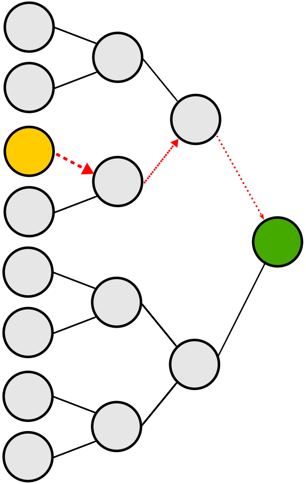
- Node s is one node in i's exponentially-growing receptive field $\rightarrow (\bar{\mathbf{A}}^{T+1})_{is} = \frac{1}{2} \cdot 3^{-T}$

Over-squashing example: binary tree



- Node **s** is one node in **i**'s exponentially-growing receptive field $\rightarrow (\bar{\mathbf{A}}^{T+1})_{is} = \frac{1}{2} \cdot 3^{-T}$
- Demonstrated in Tree-NeighborsMatch experiment in Alon and Yahav (2021)

Over-squashing example: binary tree



- Node **s** is one node in **i**'s exponentially-growing receptive field $\rightarrow (\bar{\mathbf{A}}^{T+1})_{is} = \frac{1}{2} \cdot 3^{-T}$
- Demonstrated in Tree-NeighborsMatch experiment in Alon and Yahav (2021)
- If the graph topology induces over-squashing, can we identify the edges responsible for **bottlenecks**[a]?

[a] Defined as those regions in the graph where MPNNs 'struggle' to send messages

How to identify bottlenecks?

If $\bar{\mathbf{A}}_{is}^{T+1}$ is small $\rightarrow \left|\frac{\partial \mathbf{f}_i^{(T+1)}}{\partial \mathbf{f}_s^{(0)}}\right|$ small $\rightarrow$ data at s fail to propagate to i in $T+1$ layers

How to identify bottlenecks?

If $\bar{\mathbf{A}}_{is}^{T+1}$ is small $\rightarrow \left|\frac{\partial \mathbf{f}_i^{(T+1)}}{\partial \mathbf{f}_s^{(0)}}\right|$ small $\rightarrow$ data at s fail to propagate to i in $T+1$ layers

$\rightarrow$ *The graph topology induces over-squashing in MPNN independent of the choice of update and activation functions*

How to identify bottlenecks?

If $\bar{\mathbf{A}}_{is}^{T+1}$ is small $\rightarrow \left|\frac{\partial \mathbf{f}_i^{(T+1)}}{\partial \mathbf{f}_s^{(0)}}\right|$ small $\rightarrow$ data at s fail to propagate to i in $T+1$ layers

$\rightarrow$ *The graph topology induces over-squashing in MPNN independent of the choice of update and activation functions*

Main question: Can we actually identify which edges cause bottlenecks?

How to identify bottlenecks?

If $\bar{\mathbf{A}}_{is}^{T+1}$ is small $\rightarrow \left|\frac{\partial \mathbf{f}_i^{(T+1)}}{\partial \mathbf{f}_s^{(0)}}\right|$ small $\rightarrow$ data at s fail to propagate to i in $T+1$ layers

$\rightarrow$ *The graph topology induces over-squashing in MPNN independent of the choice of update and activation functions*

Main question: Can we actually identify which edges cause bottlenecks?

Idea: Use curvature! We know it is related to 'dispersion' of edges and locally measures connectedness of neighbourhoods via edges

Conventions

We adopt the Balanced Forman curvature

$$\mathrm{BF}(i,j) := \frac{2}{d_i} + \frac{2}{d_j} - 2 + 2\frac{|\sharp_\Delta|}{\max\{d_i,d_j\}} + \frac{|\sharp_\Delta|}{\min\{d_i,d_j\}} + \frac{\gamma_{\max}^{-1}}{\max\{d_i,d_j\}}(|\sharp_\Box^i + \sharp_\Box^j)$$

Conventions

We adopt the Balanced Forman curvature

$$\mathrm{BF}(i,j) := \frac{2}{d_i} + \frac{2}{d_j} - 2 + 2\frac{|\sharp_\Delta|}{\max\{d_i,d_j\}} + \frac{|\sharp_\Delta|}{\min\{d_i,d_j\}} + \frac{\gamma_{\max}^{-1}}{\max\{d_i,d_j\}}(|\sharp_\square^i + \sharp_\square^j)$$

Convention: We say that $\mathrm{BF}(i,j)$ is *very negative* if there exists $\delta > 0$ s.t. $0 < \delta < (\max\{d_i,d_j\})^{-\frac{1}{2}}$, $\delta < \gamma_{\max}^{-1}$ and $\mathrm{BF}(i,j) \leq -2 + \delta$.

$\rightarrow$ exclude pathological cases with many 4-cycles traversing the same node

Negatively curved edges cause over-squashing

Informal version: If $\mathrm{BF}(i, j)$ is very negative, then there exist many nodes at hop-distance 2 from i such that MPNNs struggle to send messages from i to such nodes in 2 layers

Negatively curved edges cause over-squashing

Informal version: If $\mathrm{BF}(i,j)$ is very negative, then there exist many nodes at hop-distance 2 from i such that MPNNs struggle to send messages from i to such nodes in 2 layers

Theorem (Topping*, Di G.*, et al.)

Let $i \sim j$ with $d_i \leq d_j$ and assume that $\mathrm{BF}(i,j)$ is very negative. Then there exists $Q_j \subset S_2(i)$ satisfying $|Q_j| > \delta^{-1}$ and for $0 \leq t_0 \leq T-2$ we have

$$\frac{1}{|Q_j|} \sum_{k \in Q_j} \left| \frac{\partial \mathbf{f}_k^{(t_0+2)}}{\partial \mathbf{f}_i^{(t_0)}} \right| < (\alpha\beta)^2 \delta^{\frac{1}{4}}.$$

Surgical analysis: graph-rewiring

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

General idea: If the topology only is responsible for over-squashing $\rightarrow$ what if we change it?

Surgical analysis: graph-rewiring

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

General idea: If the topology only is responsible for over-squashing $\rightarrow$ what if we change it?

$\rightarrow$ resonates with ideas from geometric flows as **Ricci flow**

$$\boxed{\partial_t g(t) = -2\mathrm{Ric}(g(t))}$$

Very high-level description:

- Negatively curved directions are stretched
- Positively curved regions become rounder and collapse

Ricci flow for tackling over-squashing?

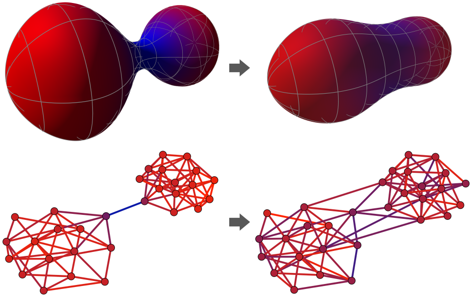
Benefit of this analysis:

- we can surgically identify bottlenecks by studying the curvature
- we can add/remove edges accordingly $\longrightarrow$ we propose SDRF algorithm

Ricci flow for tackling 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



Stochastic Discrete Ricci Flow

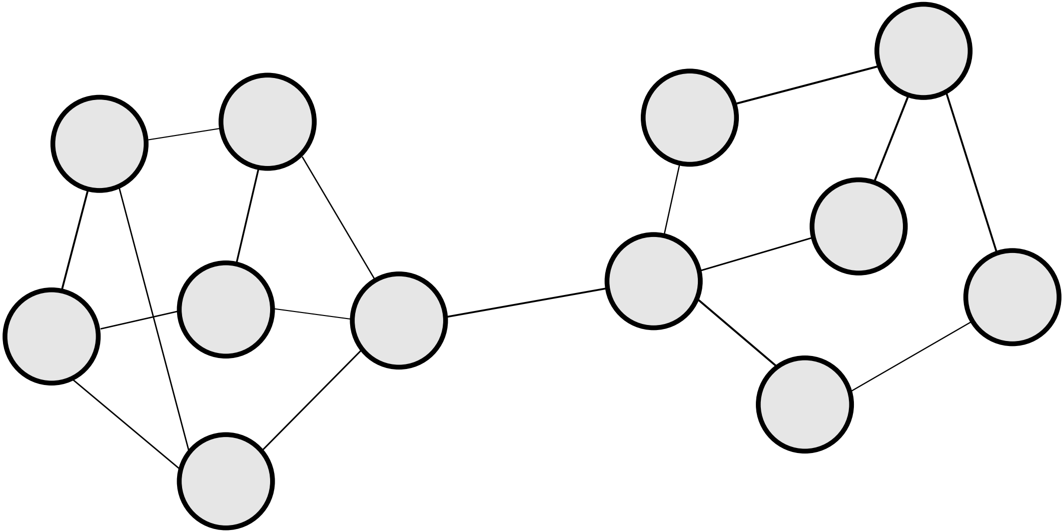
Algorithm 1: Stochastic Discrete Ricci Flow (SDRF)

Input: graph G, temperature $\tau > 0$, max number of iterations, optional Ric upper-bound C^+
Repeat
1) For edge $i \sim j$ with minimal Ricci curvature $\mathrm{Ric}(i, j)$:
   Calculate vector $\boldsymbol{x}$ where $x_{kl} = \mathrm{Ric}_{kl}(i, j) - \mathrm{Ric}(i, j)$, the improvement to $\mathrm{Ric}(i, j)$ from adding edge $k \sim l$ where $k \in B_1(i)$, $l \in B_1(j)$;
   Sample index k, l with probability $\mathrm{softmax}(\tau \boldsymbol{x})_{kl}$ and add edge $k \sim l$ to G.
2) Remove edge $i \sim j$ with maximal Ricci curvature $\mathrm{Ric}(i, j)$ if $\mathrm{Ric}(i, j) > C^+$.

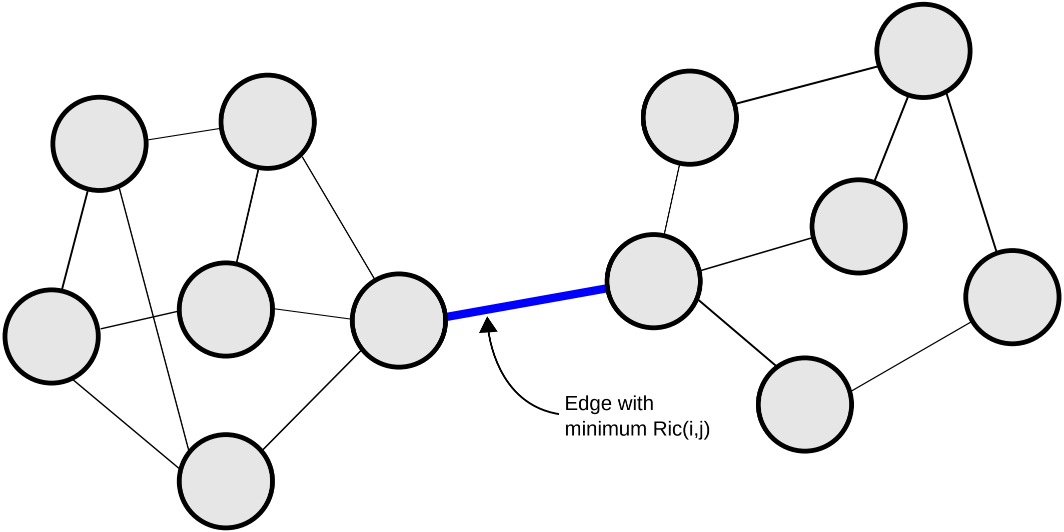
Until convergence, or max iterations reached;

This is more like a '**backwards**' Ricci flow

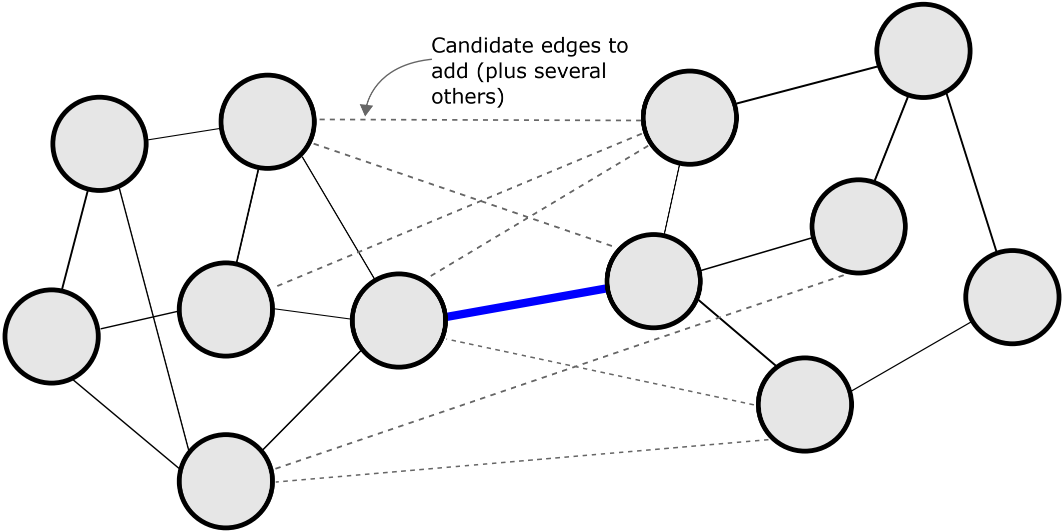
SDRF: Example



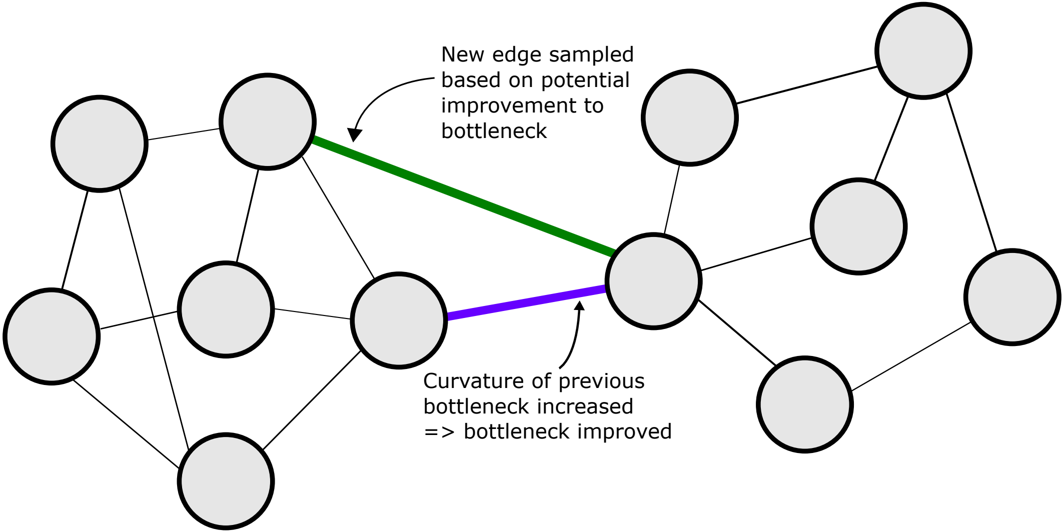
SDRF: Example



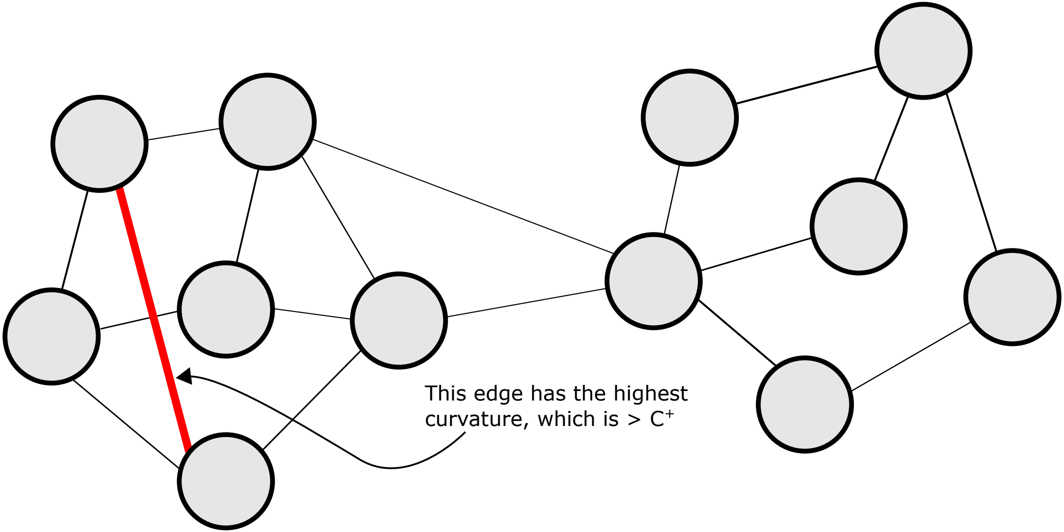
SDRF: Example



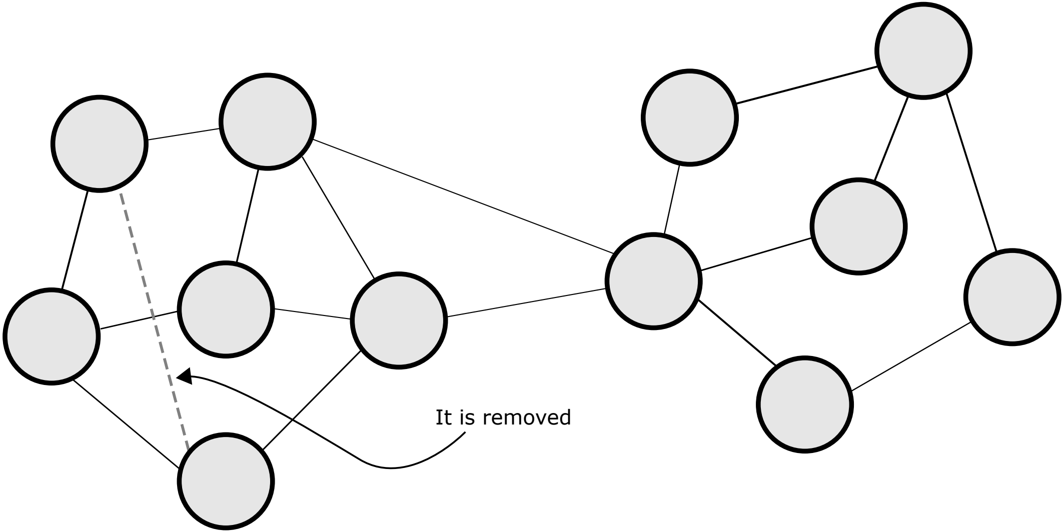
SDRF: Example



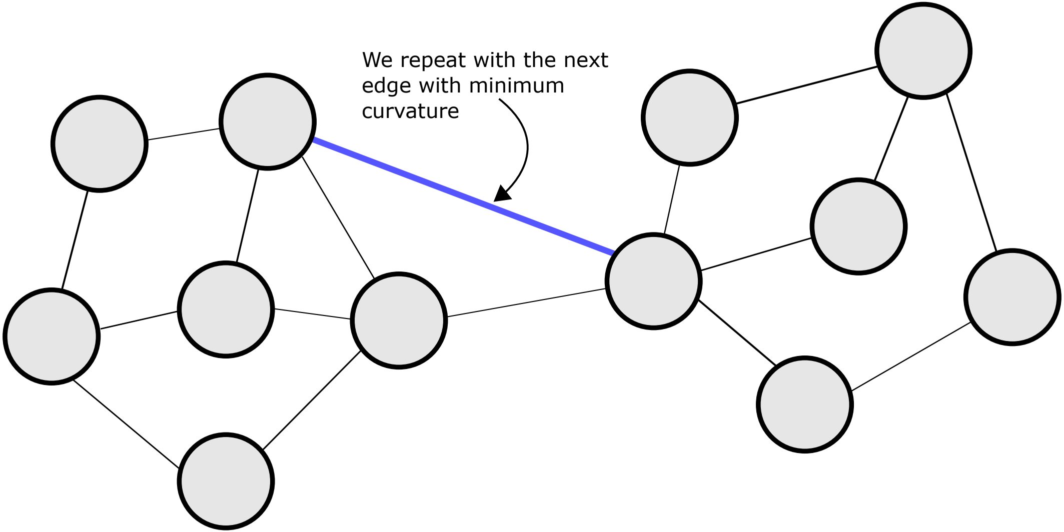
SDRF: Example



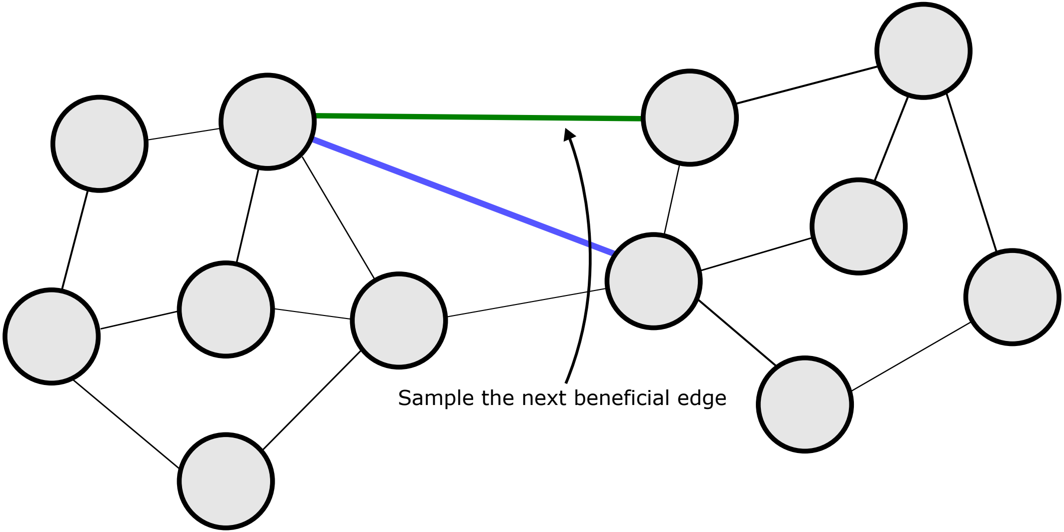
SDRF: Example



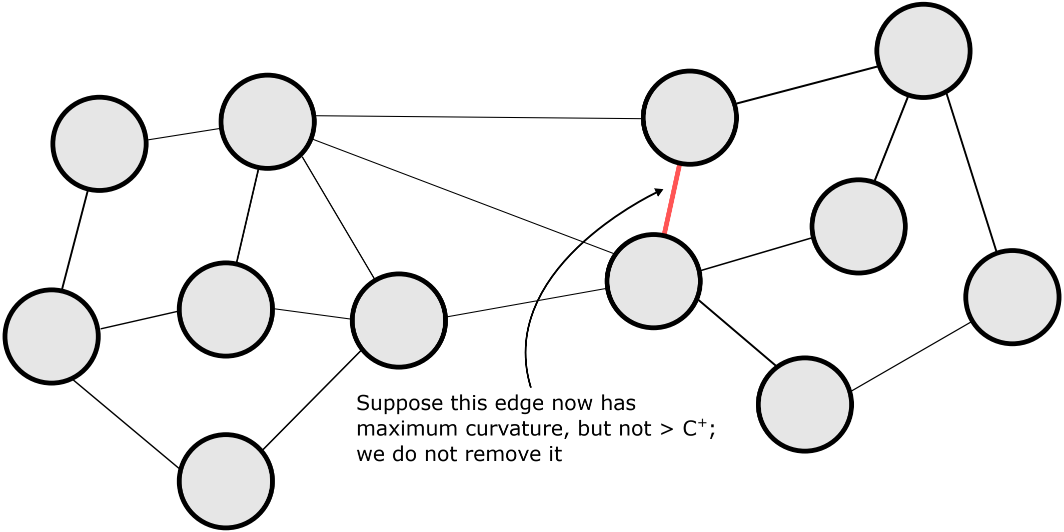
SDRF: Example



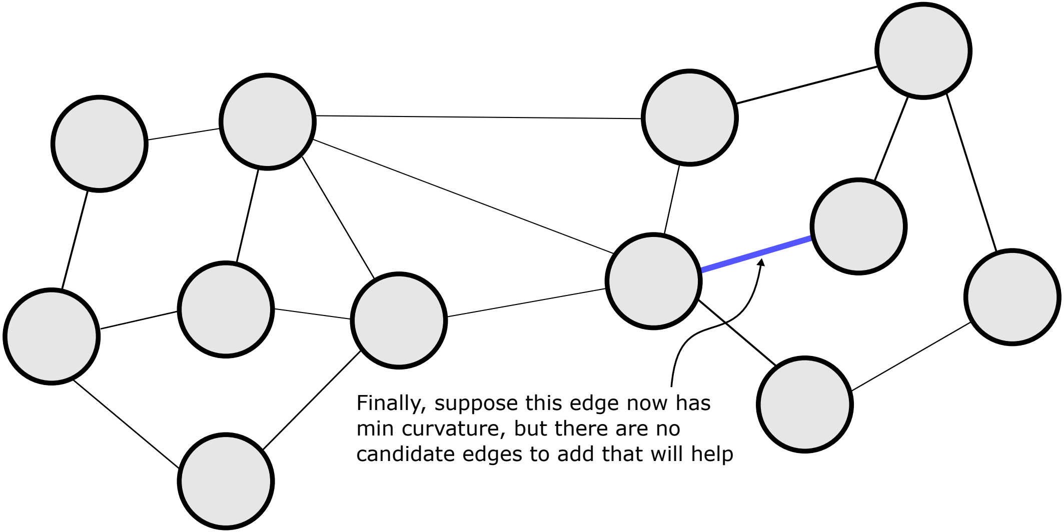
SDRF: Example



SDRF: Example

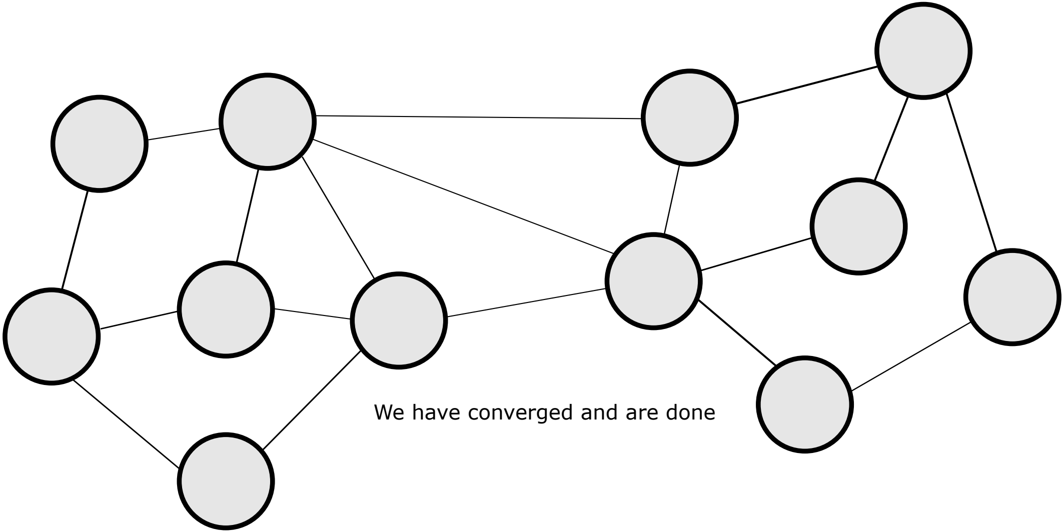


SDRF: Example



Finally, suppose this edge now has min curvature, but there are no candidate edges to add that will help

SDRF: Example



We have converged and are done

But are we allowed to change the input graph-topology?

- Graph Attention Networks learn to re-weight the input graph based on data[10]

[10] Veličković et al. (2018)
[11] Klicpera et al. (2019)
[12] Arnaiz-Rodríguez et al. (2022)

Graph-rewiring for GNNs

But are we allowed to change the input graph-topology?

- Graph Attention Networks learn to re-weight the input graph based on data[10]
- DIGL[11] proposes to smooth the graph out as pre-processing

[10] Veličković et al. (2018)
[11] Klicpera et al. (2019)
[12] Arnaiz-Rodríguez et al. (2022)

Graph-rewiring for GNNs

But are we allowed to change the input graph-topology?

- Graph Attention Networks learn to re-weight the input graph based on data[10]
- DIGL[11] proposes to smooth the graph out as pre-processing
- Methods that directly 'access' higher-order information contained in distant hops *effectively rewire the graph*

---

[10] Veličković et al. (2018)

[11] Klicpera et al. (2019)

[12] Arnaiz-Rodríguez et al. (2022)

But are we allowed to change the input graph-topology?

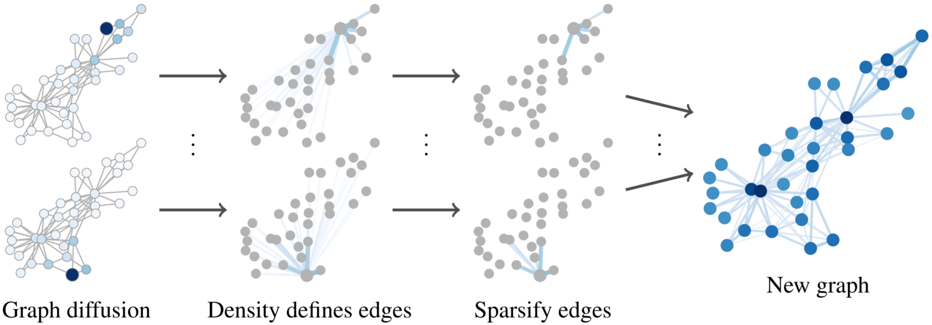
- Graph Attention Networks learn to re-weight the input graph based on data[10]
- DIGL[11] proposes to smooth the graph out as pre-processing
- Methods that directly 'access' higher-order information contained in distant hops *effectively rewire the graph*
- Learnable approaches to make the graph increasingly look like an expander [12]

$\rightarrow$ It works in the continuous case, how about the discrete setting?

[10] Veličković et al. (2018)
[11] Klicpera et al. (2019)
[12] Arnaiz-Rodríguez et al. (2022)

Can random-walk based rewiring alleviate over-squashing?

- DIGL[13] rewires the graph by graph diffusion
- Leads to significant improvements in performance on a range of models and datasets



[13] Klicpera et al. (2019)

What about DIGL / Graph Diffusion Convolution?

- Based on an assumption of homophily - common but not guaranteed
- Consider DIGL with the PPR kernel: replace input adjacency with

$$\mathbf{R}_\alpha := \sum_{k=0}^{\infty} \theta_k^{PPR} (\mathbf{D}^{-1}\mathbf{A})^k = \alpha \sum_{k=0}^{\infty} \Big((1-\alpha)(\mathbf{D}^{-1}\mathbf{A}) \Big)^k$$

Recall that a **global measure of connectivity** of G is

$$h_S = \frac{|\partial S|}{\min\{\mathrm{vol}(S), \mathrm{vol}(\mathsf{V} \setminus S)\}}, \quad h_{\mathsf{G}} = \min_S h_S$$

$\rightarrow$ related to spectral gap $\lambda_1^{\mathbf{\Delta}}$

What about DIGL / Graph Diffusion Convolution?

Theorem (Topping*, Di G.*, et al.)

Let $S \subset \mathsf{V}$ *with* $vol(S) \leq vol(\mathsf{G})/2$. *Then* $h_{S,\alpha} \leq \left(\frac{1-\alpha}{\alpha}\right) \frac{d_{avg}(S)}{d_{min}(S)} h_S$, *where* $d_{avg}(S)$ *and* $d_{min}(S)$ *are the average and minimum degree on* S, *respectively.*

What about DIGL / Graph Diffusion Convolution?

Theorem (Topping*, Di G.*, et al.)

Let $S \subset \mathsf{V}$ *with* $vol(S) \leq vol(\mathsf{G})/2$. *Then* $h_{S,\alpha} \leq \left(\frac{1-\alpha}{\alpha}\right) \frac{d_{avg}(S)}{d_{min}(S)} h_S$, *where* $d_{avg}(S)$ *and* $d_{min}(S)$ *are the average and minimum degree on* S, *respectively.*

Thanks to Lin et al. (2011) and our comparison result

Proposition (Topping*, Di G.*, et al.)

If $\mathrm{BF}(i,j) \geq \kappa > 0$ *for each edge* $(i,j) \in \mathsf{E}$, *then* $h_{\mathsf{G}} \geq \kappa/2$.

Experimental results

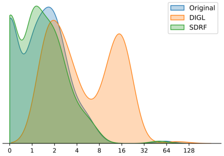
- Experiment: semi-supervised node classification with a simple GCN
- DIGL $>$ SDRF with homophily and DIGL $<$ SDRF with heterophily

Experimental results

- Experiment: semi-supervised node classification with a simple GCN
- DIGL $>$ SDRF with homophily and DIGL $<$ SDRF with heterophily

	DIGL	SDRF
Cornell	351.1% / 0.0%	7.8% / 33.3%
Texas	483.3% / 0.0%	2.4% / 10.4%
Wisconsin	300.6% / 0.0%	1.4% / 7.5%
Chameleon	336.1% / 11.8%	6.4% / 6.4%
Squirrel	228.8% / 1.9%	0.7% / 0.7%
Actor	2444.0% / 2.3%	5.4% / 9.9%
Cora	3038.0% / 0.5%	1.0% / 1.0%
Citeseer	2568.3% / 0.0%	1.1% / 1.1%
Pubmed	2747.1% / 0.1%	0.2% / 0.2%

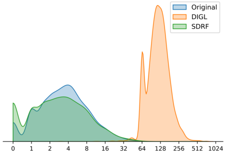
% edges added / removed by method.



(a) Wisconsin:

$W_1(\text{Original, DIGL}) = 11.83$

$W_1(\text{Original, SDRF}) = 0.28$



(b) Actor:

$W_1(\text{Original, DIGL}) = 243.81$

$W_1(\text{Original, SDRF}) = 1.03$

Recap

- *Over-squashing operational formulation: study Jacobian* $\partial \mathbf{f}^{(t)} / \partial \mathbf{f}^{(t_0)}$

Recap

- *Over-squashing operational formulation: study Jacobian* $\partial \mathbf{f}^{(t)} / \partial \mathbf{f}^{(t_0)}$
- *Over-squashing is an issue iff the given task is a function of* ***long-range relations*** *in* G

Recap

- *Over-squashing operational formulation: study Jacobian* $\partial\mathbf{f}^{(t)}/\partial\mathbf{f}^{(t_0)}$
- *Over-squashing is an issue iff the given task is a function of* ***long-range relations*** *in* G
- *Messages sent along* ***negatively curved edges*** *fail to propagate effectively in an MPNN*

Recap

- *Over-squashing operational formulation: study Jacobian* $\partial \mathbf{f}^{(t)} / \partial \mathbf{f}^{(t_0)}$
- *Over-squashing is an issue iff the given task is a function of* ***long-range relations*** *in* G
- *Messages sent along* ***negatively curved edges*** *fail to propagate effectively in an MPNN*
- *The over-squashing phenomenon is* ***independent of the chosen MPNN*** *architecture*

Recap

- *Over-squashing operational formulation: study Jacobian* $\partial \mathbf{f}^{(t)} / \partial \mathbf{f}^{(t_0)}$
- *Over-squashing is an issue iff the given task is a function of* ***long-range relations*** *in* G
- *Messages sent along* ***negatively curved edges*** *fail to propagate effectively in an MPNN*
- *The over-squashing phenomenon is* ***independent of the chosen MPNN*** *architecture*
- *Curvature-aware rewiring methods alleviate the over-squashing by surgical operations, while diffusion approaches could fail*

What's next?

How to account for long-range dependencies in an ideal world?

- **Sparsity**: mitigates computational cost and have MPNNs (roughly) scaling as $\mathcal{O}(\mathsf{E})$

What's next?

How to account for long-range dependencies in an ideal world?

- **Sparsity**: mitigates computational cost and have MPNNs (roughly) scaling as $\mathcal{O}(\mathsf{E})$
- **'Good information flow'**: if the interaction of $i, j \in \mathsf{V}$ is *important for the task*, then messages sent from i should 'quickly' reach j

What's next?

How to account for long-range dependencies in an ideal world?

- **Sparsity**: mitigates computational cost and have MPNNs (roughly) scaling as $\mathcal{O}(\mathsf{E})$
- **'Good information flow'**: if the interaction of $i, j \in \mathsf{V}$ is *important for the task*, then messages sent from i should 'quickly' reach j

A class of graphs that **may** satisfy both requirements are expanders:

Definition

A family of finite, connected graphs $\{\mathsf{G}_n\}$ is an expander family if there exist positive constants D, ε s.t. $d_{\max}(\mathsf{G}_n) \leq D$ and $\lambda_1^{\boldsymbol{\Delta}_n} \geq \varepsilon$.

What's next?

Is positive curvature enforcing an 'expander' type of property?

Theorem (Salez[14])

There are no sparse *expanders with positive Ollivier curvature.*

$\rightarrow$ positive curvature reduces bottleneck but at the cost of sparsity

How to have both sparsity and good information flow? $\rightarrow$ We proposed a surgical approach but could be improved in many ways

[14] Salez (2021)

What's next?

- Is the notion of expander graph the right one?

What's next?

- Is the notion of expander graph the right one?
- Can we find the right graph structure for the task in the expander family?

What’s next?

- Is the notion of expander graph the right one?
- Can we find the right graph structure for the task in the expander family?

In general we might not want to increase information flow across **any** pair of nodes!

What's next?

- Is the notion of expander graph the right one?
- Can we find the right graph structure for the task in the expander family?

In general we might not want to increase information flow across **any** pair of nodes!

Design data-driven graph-rewiring methods: a foray into this direction is Arnaiz-Rodríguez et al. (2022) $\rightarrow$ rewire graph to make it 'behave like an expander' but in learnable way

What's next?

- Is the notion of expander graph the right one?
- Can we find the right graph structure for the task in the expander family?

In general we might not want to increase information flow across **any** pair of nodes!

Design data-driven graph-rewiring methods: a foray into this direction is Arnaiz-Rodríguez et al. (2022) $\rightarrow$ rewire graph to make it 'behave like an expander' but in learnable way

Using λ_1^{Δ} as an indirect measure of *over-squashing* may be 'too rough'

What's next

Alon and Yahav (2021) $\rightarrow$ add an FC-layer at the end of the architecture to *break bottlenecks*
$\rightarrow$ 'post-processing' type of rewiring that leads to improvement on molecular datasets

- Are we alleviating over-squashing or just counting cycles?

What's next

Alon and Yahav (2021) $\rightarrow$ add an FC-layer at the end of the architecture to *break bottlenecks*
$\rightarrow$ 'post-processing' type of rewiring that leads to improvement on molecular datasets

- Are we alleviating over-squashing or just counting cycles?
- Is it enough to make node-representations interact after graph has been leveraged?

What's next

Alon and Yahav (2021) $\rightarrow$ add an FC-layer at the end of the architecture to *break bottlenecks*
$\rightarrow$ 'post-processing' type of rewiring that leads to improvement on molecular datasets

- Are we alleviating over-squashing or just counting cycles?
- Is it enough to make node-representations interact after graph has been leveraged?
- Was SDRF solving over-squashing for node-classification tasks?

What's next

Alon and Yahav (2021) $\rightarrow$ add an FC-layer at the end of the architecture to *break bottlenecks*
$\rightarrow$ 'post-processing' type of rewiring that leads to improvement on molecular datasets

- Are we alleviating over-squashing or just counting cycles?
- Is it enough to make node-representations interact after graph has been leveraged?
- Was SDRF solving over-squashing for node-classification tasks?
- How to actually test solution to over-squashing?

What's next

(Graph)-Transformers let **any pair of nodes interact with each other**

$\rightarrow$ solves over-squashing at the cost of complexity

Key question: What are we losing here when dropping the graph bias?

Is it enough to follow a **from-local-to-global** approach?

References

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*.

Chung, F. R. and Graham, F. C. (1997). *Spectral graph theory*. Number 92. American Mathematical Soc.

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

Forman, R. (2003). Discrete and computational geometry.

Jost, J. and Liu, S. (2014). Ollivier’s ricci curvature, local clustering and curvature-dimension inequalities on graphs. *Discrete & Computational Geometry*, 51(2):300–322.

Keller, M. and Münch, F. (2018). Gradient estimates, bakry-emery ricci curvature and ellipticity for unbounded graph laplacians. *arXiv preprint arXiv:1807.10181*.

Klicpera, J., Weißenberger, S., and Günnemann, S. (2019). Diffusion improves graph learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*.

Lin, Y., Lu, L., and Yau, S.-T. (2011). Ricci curvature of graphs. *Tohoku Mathematical Journal, Second Series*, 63(4):605–627.

Ollivier, Y. (2009). Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864.

Salez, J. (2021). Sparse expanders have negative curvature. *arXiv preprint arXiv:2101.08242*.

Samal, A., Sreejith, R., Gu, J., Liu, S., Saucan, E., and Jost, J. (2018). Comparative analysis of two discretizations of ricci curvature for complex networks. *Scientific reports*, 8(1):1–16.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations*.