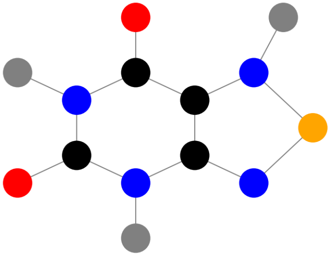
Understanding message passing: limitations and new paradigms

Francesco Di Giovanni

University of Oxford

Graphs are ubiquitous

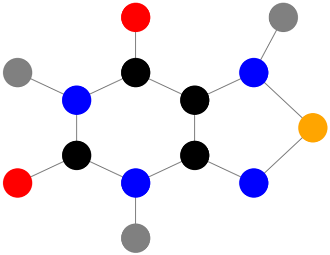
- **Graph** G is defined by nodes V and edges $E \subset V \times V$
- **Features** are signals over the graph, e.g. $X : V \to \mathbb{R}^d$

$(G, X) =$  $\longrightarrow$

Graphs are ubiquitous

- **Graph** G is defined by nodes V and edges $E \subset V \times V$
- **Features** are signals over the graph, e.g. $X : V \to \mathbb{R}^d$

$(G, X) =$



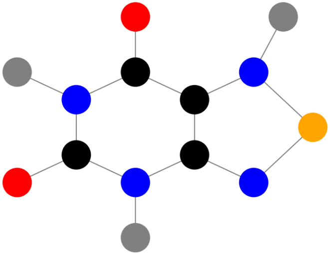
$$\longrightarrow \begin{cases} f(G, X) = \text{Is the molecular graph soluble?} \\ f(G, X) = \text{Polarity of the molecular graph} \\ f(G, X) = \text{Which user is a spammer?} \end{cases}$$

Graph Neural Networks (GNNs)

- **Graph** G is defined by nodes V and edges $E \subset V \times V$
- **Features** are signals over the graph, e.g. $X : V \to \mathbb{R}^d$
- **Labels**, i.e. we know the quantity f in a *training set*

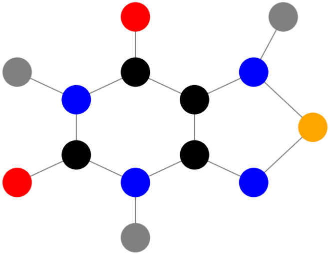
Graph Neural Networks (GNNs)

- **Graph** G is defined by nodes V and edges $E \subset V \times V$
- **Features** are signals over the graph, e.g. $X : V \to \mathbb{R}^d$
- **Labels**, i.e. we know the quantity f in a *training set*

$(G, X) =$  $\longrightarrow$

Graph Neural Networks (GNNs)

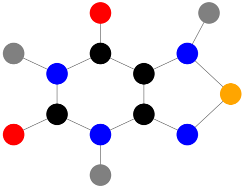
- **Graph** G is defined by nodes V and edges $E \subset V \times V$
- **Features** are signals over the graph, e.g. $X : V \to \mathbb{R}^d$
- **Labels**, i.e. we know the quantity f in a *training set*

$(G, X) =$  $\longrightarrow \quad \mathrm{GNN}_\Theta(G, X)$

- **GNN_Θ includes Neural Networks whose parameters Θ are learned through back-propagation by minimizing a loss over labels in the training set**

What is the correct graph structure?

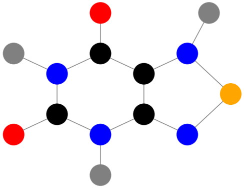
- The graph structure can be noisy or not aligned with the task

$(G, X) =$  $\longrightarrow$ $\boxed{\mathrm{GNN}_\Theta(G, X)}$

What is the correct graph structure?

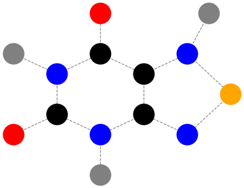
- The graph structure can be noisy or not aligned with the task

$(G, X) =$



$\longrightarrow$ $\mathrm{GNN}_\Theta(G, X)$

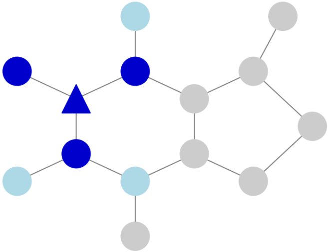
$\boxed{(G, X) =}$



$\longrightarrow$ $\mathrm{GNN}_\Theta(G, X)$

Long-range interactions

- Fast-decaying vs slowly-decaying forces



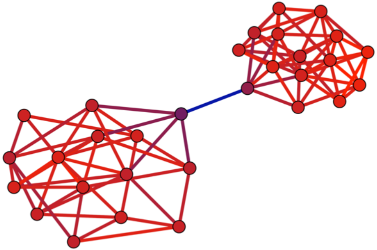
Long-range interactions

- Fast-decaying vs slowly-decaying forces

→ Long-range interactions on periodic systems, protein folding, molecular dynamics

Bottlenecks

- Bottleneck between clusters (communities)



→ Label assignments on heterophilic graphs

Motivating questions and outline

- Are GNNs capable of capturing long-range interactions and break bottlenecks?

Motivating questions and outline

- Are GNNs capable of capturing long-range interactions and break bottlenecks?

$\rightarrow$ In many cases, the answer is negative due to a phenomenon known as **oversquashing**

Motivating questions and outline

- Are GNNs capable of capturing long-range interactions and break bottlenecks?

$\rightarrow$ In many cases, the answer is negative due to a phenomenon known as **oversquashing**

- Can we *alter* the input graph in an optimal way?

Motivating questions and outline

- Are GNNs capable of capturing long-range interactions and break bottlenecks?

$\rightarrow$ In many cases, the answer is negative due to a phenomenon known as **oversquashing**

- Can we *alter* the input graph in an optimal way?

$\rightarrow$ Introduction of **graph rewiring**, a paradigm shift for designing sparse, powerful GNNs

Motivating questions and outline

- Are GNNs capable of capturing long-range interactions and break bottlenecks?

$\rightarrow$ In many cases, the answer is negative due to a phenomenon known as **oversquashing**

- Can we *alter* the input graph in an optimal way?

$\rightarrow$ Introduction of **graph rewiring**, a paradigm shift for designing sparse, powerful GNNs

- What's the state of things of GNNs and where to go from here?

Motivating questions and outline

- Are GNNs capable of capturing long-range interactions and break bottlenecks?

→ In many cases, the answer is negative due to a phenomenon known as **oversquashing**

- Can we *alter* the input graph in an optimal way?

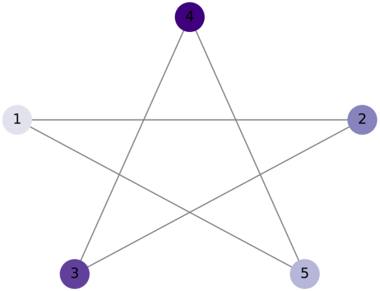
→ Introduction of **graph rewiring**, a paradigm shift for designing sparse, powerful GNNs

- What's the state of things of GNNs and where to go from here?

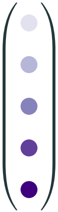
→ A perspective on the future, new directions and opportunities

The Message Passing paradigm: overview

Representing graphs and features



$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{X} =$$



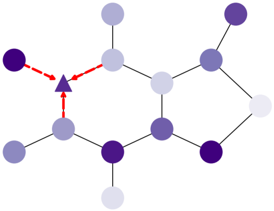
Need graph-level output of any neural operation to be **invariant** to permutations

The message-passing paradigm: MPNN $\subset$ GNN

Given m number of layers, also called **depth**

$$\mathbf{h}_v^{(t)} = f^{(t)}\Big(\mathbf{h}_v^{(t-1)}, \mathrm{AGG}(\{\mathbf{h}_u^{(t-1)} : (v,u) \in E\})\Big), \quad 1 \leq t \leq m$$

- f is a neural network (MLP), AGG is permutation invariant (e.g. sum, mean)



- Equivariant to permutations
- Locality inductive bias
- Linear complexity in the number of edges

The class of MPNNs

We consider Message Passing Neural Networks (MPNNs) of the form:

$$\mathbf{h}_v^{(t)} = \sigma\Big(\mathbf{\Omega}^{(t)}\mathbf{h}_v^{(t-1)} + \mathbf{W}^{(t)}\sum_u \mathsf{A}_{vu}\psi^{(t)}(\mathbf{h}_v^{(t-1)}, \mathbf{h}_u^{(t-1)})\Big), \quad \mathbf{h}_v^{(0)} = \mathbf{x}_v$$

- σ is a pointwise nonlinear map, $\mathbf{\Omega}^{(t)}$, $\mathbf{W}^{(t)}$ are learnable weight matrices
- $\mathsf{\mathbf{A}}$ is a message-passing matrix typically chosen in $\{\mathbf{D}^{-1}\mathbf{A}, \mathbf{A}, \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\}$
- $\psi^{(t)}$ is a learnable message function

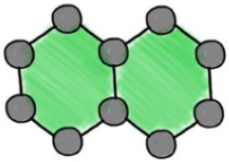
$\rightsquigarrow$ include standard models as GCN, GraphSAGE, GIN, GatedGCN

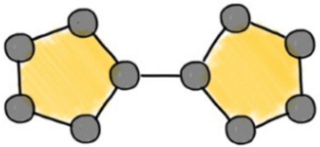
Pitfalls of message passing: limited expressive power

- **Expressive power**: typically studied through graph isomorphism and color refinement (Xu et al. (2018), Morris et al., (2018)):

  → MPNNs have limited expressive power i.e. there exist graph-functions that cannot be approximated

  E.g. MPNNs cannot distinguish regular graphs without features



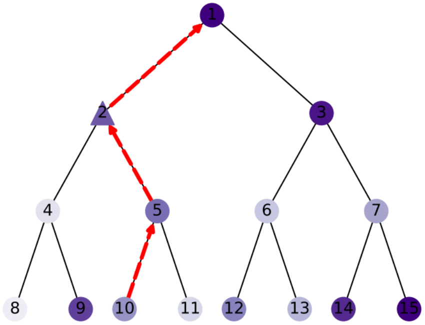


Shortcomings of using graph-isomorphism test

$\rightarrow$This approach identifies models that are better at counting substructures but...

- Not clear which functions of features can be learned, and *how easily*
- Type of expressivity that only matters when the graph is noiseless and the 2D info is crucial $\rightarrow$ not that useful for equivariant GNNs on point clouds

Pitfalls of message-passing: oversquashing



- Depending on the topology but independent of MPNN, the size of the $r-$hop of a node may grow **exponentially** (Alon & Yahav (2021))
- Messages have fixed dimensions $\rightarrow$ node 1 may fail to receive information from node 10

$\rightarrow$ Oversquashing highlights limitations on expressive power that go undetected by graph-isomorphism test and color refinement algorithms

Oversquashing: open questions

- How do we analytically measure and show the impact of oversquashing?

Oversquashing: open questions

- How do we analytically measure and show the impact of oversquashing?
- How does the graph topology lead to oversquashing?

Oversquashing: open questions

- How do we analytically measure and show the impact of oversquashing?
- How does the graph topology lead to oversquashing?
- How do we combat oversquashing?

Oversquashing: open questions

- How do we analytically measure and show the impact of oversquashing?
- How does the graph topology lead to oversquashing?
- How do we combat oversquashing?

Questions addressed in

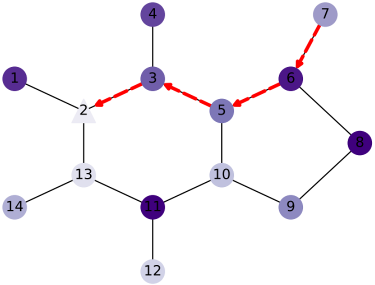
- Topping*, **Di G.***, Chamberlain, Dong, Bronstein, *Understanding over-squashing and bottlenecks on graphs via curvature*, ICLR 2022, Top 10/3300
- **Di G.**, Giusti, Barbero, Luise, Liò, Bronstein, *On Over-Squashing in Message Passing Neural Networks: The Impact of Width, Depth, and Topology*, ICML 2023
- Gutteridge, Dong, Bronstein, **Di G.**, *DRew: Dynamically Rewired Message Passing with Delay*, ICML 2023
- **Di G.***, Rusch*, Bronstein, Deac, Lackenby, Mishra, Velickvovic, *How does over-squashing affect the power of GNNs?*, TMLR 2024

Analyzing the problem: oversquashing

Understanding oversquashing: sensitivity analysis

Recall that $\mathbf{h}_v^{(t)}$ is the feature of node v at layer t

Question: How do we analytically measure and show the impact of oversquashing?



To measure how messages are propagated in MPNNs use derivatives of features e.g. $\partial \mathbf{h}_2^{(t_0+4)} / \partial \mathbf{h}_7^{(t_0)}$

Measuring the pairwise mixing

Question: How do we analytically measure and show the impact of oversquashing?

$\rightarrow$ Let y be a smooth graph-function of node features

- Taylor approx: $y(x_1, \ldots, x_n) = \mathcal{P}(x_1, \ldots, x_n) + \mathcal{R}$
- Nonlinear interactions between x_v, x_u w.r.t. y estimated by **mixed products** $x_v x_u$ in $\mathcal{P}$
- Monomials $x_v x_u$ are multiplied by the Hessian of y

Measuring the pairwise mixing

Question: How do we analytically measure and show the impact of oversquashing?

$\rightarrow$ Let y be a smooth graph-function of node features

- Taylor approx: $y(x_1, \dots, x_n) = \mathcal{P}(x_1, \dots, x_n) + \mathcal{R}$
- Nonlinear interactions between x_v, x_u w.r.t. y estimated by **mixed products** $x_v x_u$ in $\mathcal{P}$
- Monomials $x_v x_u$ are multiplied by the Hessian of y

Definition: For a *smooth* graph-function y of node features $\{\mathbf{x}_i\}$, the **maximal mixing** induced by y among the features $\mathbf{x}_v$ and $\mathbf{x}_u$ associated with nodes v, u is

$$\boxed{\mathsf{mix}_y(v,u) = \max_{\mathbf{x}_i} \max_{1 \leq \alpha, \beta \leq d} \left| \frac{\partial^2 y(\mathbf{X})}{\partial x_v^\alpha \partial x_u^\beta} \right|.}$$

Measuring oversquashing through derivatives

Question: How do we analytically measure and show the impact of oversquashing? Monitor derivatives of the features computed by layers of an MPNN

Measuring oversquashing through derivatives

Question: How do we analytically measure and show the impact of oversquashing? Monitor derivatives of the features computed by layers of an MPNN

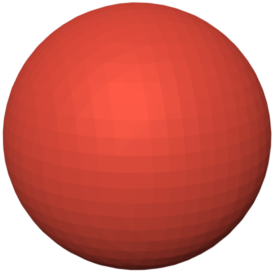
- Assess how $\partial \mathbf{h}_v^{(t_0+m)} / \partial \mathbf{h}_u^{(t_0)}$ is affected by the graph **topology**
- For MPNN output $\tilde{y}$ and ground-truth function y, show how **topology** prevents $\mathsf{mix}_{\tilde{y}}(v, u)$ from matching the value $\mathsf{mix}_{y}(v, u)$

$\rightarrow$ Now standard way of studying and quantifying oversquashing

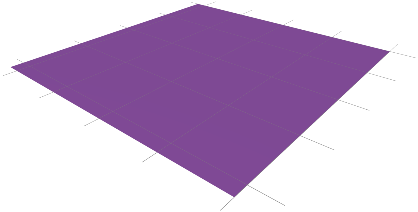
Large negative curvature induces oversquashing

Question: How does the graph topology lead to oversquashing?

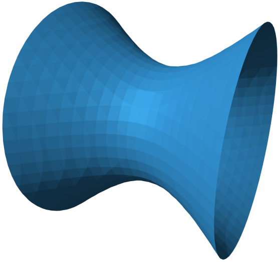
- In differential geometry, Ricci curvature on manifolds relates to information spreading



Spherical (> 0)



Euclidean ($= 0$)

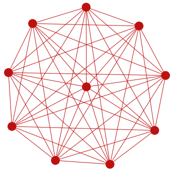


Hyperbolic (< 0)

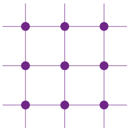
Large negative curvature induces oversquashing

Question: How does the graph topology lead to oversquashing?

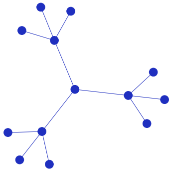
- **New** edge-curvature $\mathsf{Ric}(v, u) \in (-2, 1]$ related to Ollivier curvature



Clique ($\mathsf{Ric} > 0$)



Grid ($\mathsf{Ric} = 0$)

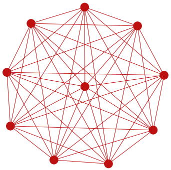


Tree ($\mathsf{Ric} < 0$)

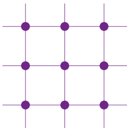
Large negative curvature induces oversquashing

Question: How does the graph topology lead to oversquashing?

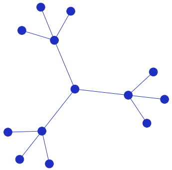
- **New** edge-curvature $\mathsf{Ric}(v, u) \in (-2, 1]$ related to Ollivier curvature



Clique ($\mathsf{Ric} > 0$)



Grid ($\mathsf{Ric} = 0$)



Tree ($\mathsf{Ric} < 0$)

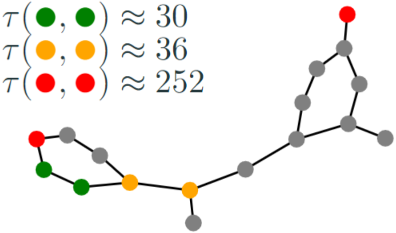
Theorem (Informal statement) *The sensitivity of node features is reduced around edges with large,* **negative curvature**.

$\rightarrow$ First result relating local properties of the topology to the sensitivity of features

Commute time

Question: How does the graph topology lead to oversquashing?

- Commute time $\tau : V \times V \to \mathbb{R}_+$ measures the expected number of steps for a random walk $v \to u \to v$
- τ is a distance on the graph and depending on the topology $\tau = \mathcal{O}(|V|^3)$



A measure of over-squashing

Oversquashing $\rightarrow$ inability of MPNNs to model interactions among **certain nodes**

A measure of over-squashing

Oversquashing → inability of MPNNs to model interactions among **certain nodes**

- Let m be the number of layers and w be the maximal norm of the weights

→ Given an MPNN with capacity (m,w), we define the **pairwise** oversquashing of v, u as

$$\mathsf{OSQ}_{v,u}(m, \mathsf{w}) = \left(\mathsf{mix}_{\tilde{y}}(v, u)\right)^{-1}.$$

What is the minimal capacity (m, w) required to induce certain mixing $\mathsf{mix}_y(v, u)$?

Minimal number of layers m required

Question: How does the graph topology lead to oversquashing?

$\rightarrow$ Recall $\mathsf{mix}_{y}(v,u) = \max_{\mathbf{x}_i} \max_{1\leq\alpha,\beta\leq d} \left| \frac{\partial^2 y(\mathbf{X})}{\partial x_v^{\alpha} \partial x_u^{\beta}} \right|$

Minimal number of layers m required

Question: How does the graph topology lead to oversquashing?

$\to$ Recall $\mathsf{mix}_y(v,u) = \max_{\mathbf{x}_i} \max_{1\le\alpha,\beta\le d} \left| \frac{\partial^2 y(\mathbf{X})}{\partial x_v^\alpha \partial x_u^\beta} \right|$

Theorem. *A* **necessary** *condition for an* MPNN *of bounded weights to induce* $\mathsf{mix}_y(v,u)$ *is:*

$$m \geq \frac{\tau(v,u)}{8} + \alpha\, \mathsf{mix}_y(v,u) - \beta. \quad (*)$$

Minimal number of layers m required

Question: How does the graph topology lead to oversquashing?

$\rightarrow$ Recall $\mathsf{mix}_y(v,u) = \max_{\mathbf{x}_i} \max_{1\leq\alpha,\beta\leq d} \left| \frac{\partial^2 y(\mathbf{X})}{\partial x_v^\alpha \partial x_u^\beta} \right|$

Theorem. *A* **necessary** *condition for an* MPNN *of bounded weights to induce* $\mathsf{mix}_y(v,u)$ *is:*

$$\boxed{m \geq \frac{\tau(v,u)}{8} + \alpha\, \mathsf{mix}_y(v,u) - \beta. \quad \Big(* \Big)}$$

Corollary. Given an MPNN of bounded weights, if m fails to satisfy $\Big(* \Big)$, then the MPNN cannot learn functions with mixing $\mathsf{mix}_y(v,u)$.

Consequence of the theoretical results

- Identifies **functions harder to learn** for MPNNs with **practical size**

Consequence of the theoretical results

- Identifies **functions harder to learn** for MPNNs with **practical size**
- Oversquashing more general than long-range interactions (if τ large for nearby nodes)

Consequence of the theoretical results

- Identifies **functions harder to learn** for MPNNs with **practical size**
- Oversquashing more general than long-range interactions (if τ large for nearby nodes)
- Result studying limitations on the MPNNs derivatives to match derivatives of the task

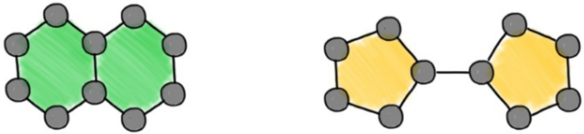
Consequence of the theoretical results

- Identifies **functions harder to learn** for MPNNs with **practical size**
- Oversquashing more general than long-range interactions (if τ large for nearby nodes)
- Result studying limitations on the MPNNs derivatives to match derivatives of the task
- Results also apply to MPNN models on geometric graphs (point clouds in 3D)

Consequence of the theoretical results

- Identifies **functions harder to learn** for MPNNs with **practical size**
- Oversquashing more general than long-range interactions (if τ large for nearby nodes)
- Result studying limitations on the MPNNs derivatives to match derivatives of the task
- Results also apply to MPNN models on geometric graphs (point clouds in 3D)

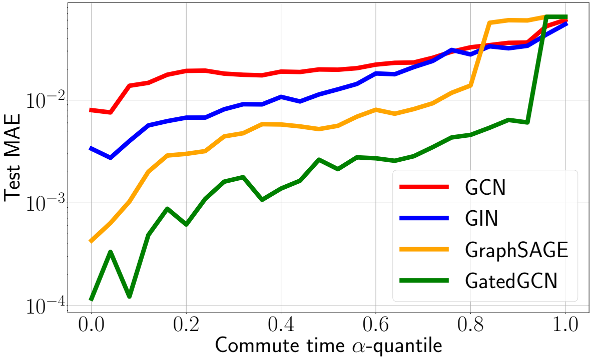
No assumption on the type of features: MPNNs can distinguish nodes, but they would still have **low mixing** between nodes at large commute time



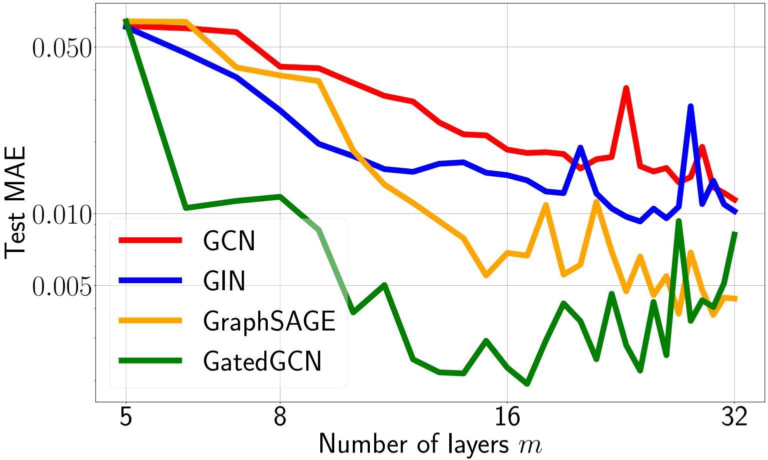
$\rightarrow$ **Mixing is a new paradigm for expressive power beyond graph isomorphism test**

Synthetic ZINC

- $\{\mathsf{G}^i\}$ is the ZINC molecular graphs
- $x_v^i = 0$, except for two, which are set to uniform random numbers $x_{u^i}^i, x_{v^i}^i$ in $(0, 1)$
- Regression output is $y^i = \tanh(x_{u^i}^i + x_{v^i}^i)$
- The two non-zero node features $x_{u^i}^i, x_{v^i}^i$ are positioned on G^i according to the α-quantile of the commute time τ distribution



The role of depth



On the level of mixing

Mixing	*input interval*	*maximal mixing*	GCN	GIN	GraphSAGE	GatedGCN
$\tanh(x^i_{u^i} + x^i_{v^i})$	$(0, 1)$	≈ 0.77	0.024	0.014	0.006	0.004
$\exp(x^i_{u^i} + x^i_{v^i})$	$(0, 1)$	≈ 7.4	0.043	0.021	0.033	0.008
$\exp(x^i_{u^i} + x^i_{v^i})$	$(0, 1.5)$	≈ 20.1	0.054	0.035	0.075	0.014

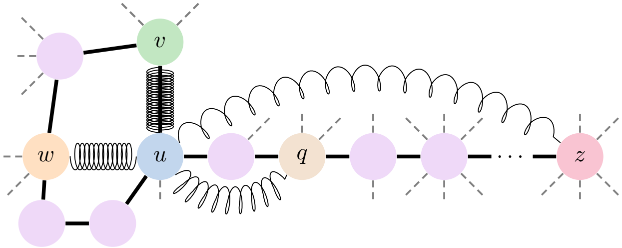
Impact of graph topology to message exchange

Question: How does the graph topology lead to oversquashing?

Impact of graph topology to message exchange

Question: How does the graph topology lead to oversquashing?

- Showed that large negative curvature reduces sensitivity of features (locally)
- New paradigm for expressive power based on mixed second-order derivatives $\rightarrow$ Oversquashing prevents MPNNs of bounded depth from learning graph functions inducing strong mixing among nodes with large commute time



Graph rewiring: where and when messages should be exchanged?

How to correct oversquashing: Graph Rewiring

Question: How do we combat oversquashing?

How to correct oversquashing: Graph Rewiring

Question: How do we combat oversquashing?

Negative curvature reduces sensitivity.. $\rightarrow$

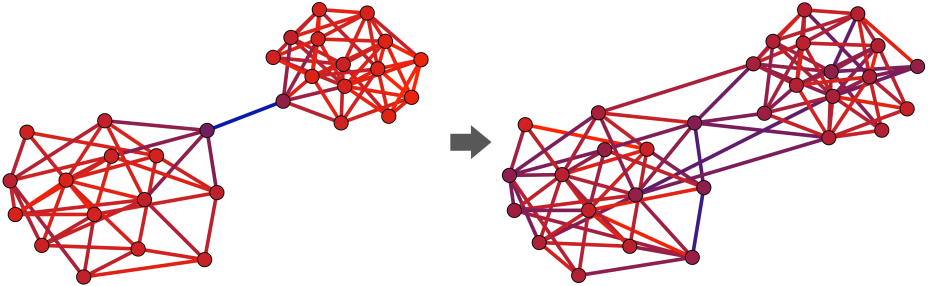
- Add edges to increase curvature of bottlenecks
- Remove edges where curvature is positive and large to retain sparsity

How to correct oversquashing: Graph Rewiring

Question: How do we combat oversquashing?

Negative curvature reduces sensitivity.. $\rightarrow$

- Add edges to increase curvature of bottlenecks
- Remove edges where curvature is positive and large to retain sparsity



A paradigm shift

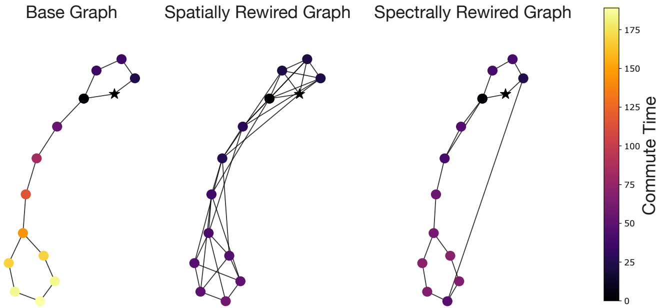
Question: How do we combat oversquashing? $\rightarrow$ Rewiring paradigm: find (**learn**) $\mathcal{R} : G \mapsto \mathcal{R}(G)$ to reduce oversquashing and combine MPNN over G + MPNN over $\mathcal{R}(G)$

(Deac et al., 2022; Nguyen et al., 2022; Black et al., 2023; Arnaiz-Rodríguez et al., 2022; Abboud et al., 2022; Karhadkar et al., 2023)...

A paradigm shift

Question: How do we combat oversquashing? $\rightarrow$ Rewiring paradigm: find (**learn**) $\mathcal{R} : G \mapsto \mathcal{R}(G)$ to reduce oversquashing and combine MPNN over G + MPNN over $\mathcal{R}(G)$

(Deac et al., 2022; Nguyen et al., 2022; Black et al., 2023; Arnaiz-Rodríguez et al., 2022; Abboud et al., 2022; Karhadkar et al., 2023)...



Recent works: spatial methods

Spatial Rewiring: $\mathcal{R}$ add edges among nodes within a certain distance and use different weights based on mutual distance

- **Oversquashing mitigated**
- **Can compute distance-based functions**

Recent works: spatial methods

Spatial Rewiring: $\mathcal{R}$ add edges among nodes within a certain distance and use different weights based on mutual distance

- **Oversquashing mitigated**
- **Can compute distance-based functions**
- **More sensitive to over-smoothing**
- **Higher impact on training time**
- **Loses inductive bias afforded by the distance**

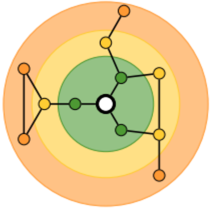


Figure 1: Figure from Abboud et al. (2022)

An extreme example of spatial rewiring: Graph Transformers

Graph Transformer is the extreme case of spatial rewiring with $\mathcal{R}(G) = (V, V \times V)$, (Kreuzer et al., 2021; Rampasek et al., 2022; Shirzad et al., 2023)...

- Quadratic memory cost $O(|V|^2) \rightsquigarrow$ Lot of interest in 'sparsifying Graph Transformers'

An extreme example of spatial rewiring: Graph Transformers

Graph Transformer is the extreme case of spatial rewiring with $\mathcal{R}(G) = (V, V \times V)$, (Kreuzer et al., 2021; Rampasek et al., 2022; Shirzad et al., 2023)...

- Quadratic memory cost $O(|V|^2)$ $\rightsquigarrow$ Lot of interest in 'sparsifying Graph Transformers'
- Need data augmentation $\rightsquigarrow$ $v \mapsto \mathbf{p}_v$ encoding positional/structural info; what is their expressive power?

An extreme example of spatial rewiring: Graph Transformers

Graph Transformer is the extreme case of spatial rewiring with $\mathcal{R}(G) = (V, V \times V)$, (Kreuzer et al., 2021; Rampasek et al., 2022; Shirzad et al., 2023)...

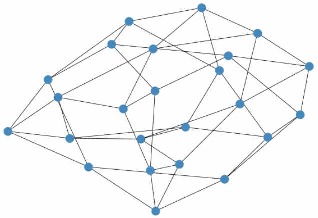
- Quadratic memory cost $O(|V|^2)$ $\rightsquigarrow$ Lot of interest in 'sparsifying Graph Transformers'
- Need data augmentation $\rightsquigarrow$ $v \mapsto \mathbf{p}_v$ encoding positional/structural info; what is their expressive power?
- Need enough data to recover inductive bias $\rightsquigarrow$ How GTs operate with fewer labels?

Spectral rewiring

$\rightarrow h_{\mathsf{G}}$ measures the 'energy' required to separate a graph into two communities

Deac et al. (2022), Arnaiz-Rodríguez et al. (2022), Karhadkar et al. (2023) $\rightarrow h_{\mathcal{R}(\mathsf{G})} > h_{\mathsf{G}}$

- Oversquashing is mitigated (worst-case commute time is reduced)
- Why improving the information flow among any pair of nodes?
- How 'close' $\mathcal{R}(\mathsf{G})$ is to G?



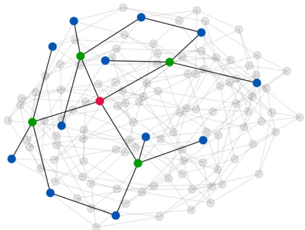


Figure 2: Figure taken from Deac et al. (2022)

Some desiderata

How to combat oversquashing in an ideal world?

(i) **Sparsity**: small computational cost and use local MPNNs

(ii) Avoiding to make the graph **too connected** too soon to mitigate oversmoothing and preserve the inductive bias afforded by the graph

(iii) **'Good information flow'**: if the interaction of $u, v \in \mathsf{V}$ is *important for the task*, then messages sent from u should 'quickly' reach v

Caveat: how do we ***actually validate*** the existence of long-range dependencies on tasks?

Rewiring over space and time

Question: How do we combat oversquashing?

- **Where**? $\rightarrow$ Traditional GNNs exchange messages over input edges, but rewiring adds and removes edges based on topology

Rewiring over space and time

Question: How do we combat oversquashing?

- **Where**? $\rightarrow$ Traditional GNNs exchange messages over input edges, but rewiring adds and removes edges based on topology

- **When?** $\rightarrow$ Do we need to send messages simultaneously? How can we use the graph topology to determine when nodes should interact?

Rewiring over space and time

Question: How do we combat oversquashing?

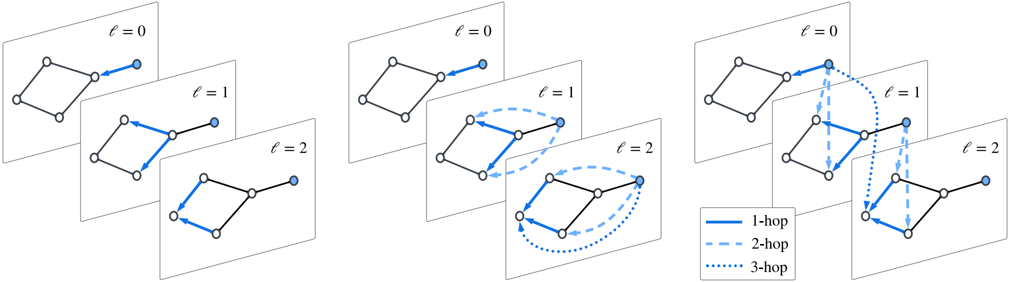
- **Where**? $\rightarrow$ Traditional GNNs exchange messages over input edges, but rewiring adds and removes edges based on topology

- **When?** $\rightarrow$ Do we need to send messages simultaneously? How can we use the graph topology to determine when nodes should interact?

$\rightarrow$ 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; Rampasek et al., 2022) $\rightsquigarrow \mathcal{R}(\mathsf{G})$ **becomes much denser**

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

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
- Preserve inductive bias of the input graph: nodes that are closer interact earlier

The advantages of νDREW

- The graph is progressively filled with each layer making for a more efficient framework
- Preserve inductive bias of the input graph: nodes that are closer interact earlier
- The delay allows messages to travel vertically (across multiple layers) $\rightarrow$ we are 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
- Preserve inductive bias of the input graph: nodes that are closer interact earlier
- The delay allows messages to travel vertically (across multiple layers) $\rightarrow$ we are 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
- Preserve inductive bias of the input graph: nodes that are closer interact earlier
- The delay allows messages to travel vertically (across multiple layers) $\rightarrow$ we are 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

The advantages of νDREW

- The graph is progressively filled with each layer making for a more efficient framework
- Preserve inductive bias of the input graph: nodes that are closer interact earlier
- The delay allows messages to travel vertically (across multiple layers) $\rightarrow$ we are 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
- Performance competitive with more complex GraphTransformers

Vision and future perspectives

Motivations: why Geometric Deep Learning and Challenges in Life Sciences

- Data scarcity and acquisition costs: Quantum mechanics simulations are more expensive than gathering images or sentences
- Non-Euclidean structures with physical constraints: Instead of grids or sequences we have E(3) symmetries to account for on point clouds
- Lack of solid theoretical foundations that limits trust: Are large models successful for language processing and computer vision the right tools for scientific domains too?

The Geometric Deep Learning blueprint

- Input $\mathcal{X}$ and output $\mathcal{Y}$ spaces
- There exists a group G acting from the left on $\mathcal{X}$ and $\mathcal{Y}$, respectively
- $f_\theta : \mathcal{X} \to \mathcal{Y}$ is a neural network model

The Geometric Deep Learning blueprint

- Input $\mathcal{X}$ and output $\mathcal{Y}$ spaces
- There exists a group G acting from the left on $\mathcal{X}$ and $\mathcal{Y}$, respectively
- $f_\theta : \mathcal{X} \to \mathcal{Y}$ is a neural network model

Geometric Deep Learning rests on the **equivariance** paradigm:

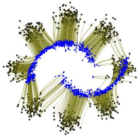
$$\begin{array}{ccc} \mathcal{X} & \xrightarrow{G \hookrightarrow} & \mathcal{X} \\ {\scriptstyle f_\theta}\downarrow & & \downarrow{\scriptstyle f_\theta} \\ \mathcal{Y} & \xrightarrow{G \hookrightarrow} & \mathcal{Y} \end{array}$$

GNNs is a special instance, where G is the permutation group

Pushing Geometric Deep Learning to new frontiers

New frontiers for Graph Theory and Graph Neural Networks

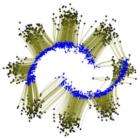
GNNs across samples for Deep Generative AI $\rightarrow$ Applications to single-cell RNA + new framework to learn geometry from data



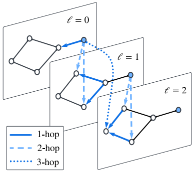
Pushing Geometric Deep Learning to new frontiers

New frontiers for Graph Theory and Graph Neural Networks

GNNs across samples for Deep Generative AI $\rightarrow$ Applications to single-cell RNA + new framework to learn geometry from data



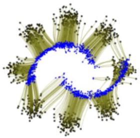
GNNs across layer-representations of neural networks and time-rewiring $\rightarrow$ Applications to Molecular Dynamics + numerical solutions of PDEs



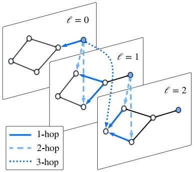
Pushing Geometric Deep Learning to new frontiers

New frontiers for Graph Theory and Graph Neural Networks

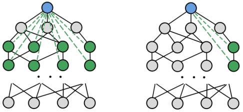
GNNs across samples for Deep Generative AI $\rightarrow$ Applications to single-cell RNA + new framework to learn geometry from data



GNNs across layer-representations of neural networks and time-rewiring $\rightarrow$ Applications to Molecular Dynamics + numerical solutions of PDEs



Graph rewiring to sparsify and study Transformers $\rightarrow$ Sparsifying Transformers with applications to protein generation



- Provide stronger theoretical foundations to existing approaches $\rightarrow$**Do we need to bake symmetries into our model or can symmetries emerge in large models?** How does enforcing symmetries affect the optimization of weights of neural networks?

Reassessing Geometric Deep Learning

- Provide stronger theoretical foundations to existing approaches $\rightarrow$**Do we need to bake symmetries into our model or can symmetries emerge in large models?** How does enforcing symmetries affect the optimization of weights of neural networks?

- Is it just a matter of data? $\rightarrow$ Assessing if there exists conditions on optimization and data, under which performance is **independent** of specifics of the model $\rightarrow$ Understanding foundational limits of GDL to propose new AI paradigms for science

Thank you!

QR-codes for papers

Contacts

Email : francesco.di.giovanni at cs.ox.ac.uk

Social : @*Francesco_dgv*

References

Abboud, R., Dimitrov, R., and Ceylan, I. I. (2022). Shortest path networks for graph property prediction. In *The First Learning on Graphs Conference*.

Arnaiz-Rodríguez, A., Begga, A., Escolano, F., and Oliver, N. (2022). DiffWire: Inductive Graph Rewiring via the Lovász Bound. In *The First Learning on Graphs Conference*.

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*.

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. In *The First Learning on Graphs Conference*.

Karhadkar, K., Banerjee, P. K., and Montufar, G. (2023). FoSR: First-order spectral rewiring for addressing oversquashing in GNNs. In *The Eleventh International Conference on Learning Representations*.

Kreuzer, D., Beaini, D., Hamilton, W., Létourneau, V., and Tossou, P. (2021). Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems*, volume 34, pages 21618–21629.

Nguyen, K., Nguyen, T., Ho, N., Nguyen, K., Nong, H., and Nguyen, V. (2022). Revisiting over-smoothing and over-squashing using ollivier's ricci curvature. *arXiv preprint arXiv:2211.15779*.

Rampasek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. (2022). Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*.

Shirzad, H., Velingker, A., Venkatachalam, B., Sutherland, D. J., and Sinop, A. K. (2023). Exphormer: Sparse transformers for graphs. *arXiv preprint arXiv:2303.06147*.

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? In *Advances in Neural Information Processing Systems*, volume 34, pages 28877–28888.