Graph Rewiring: From theory to Applications in Fairness

Tutorial on the 1st Learning on Graphs Conference 2022

Adrián Arnaiz Rodríguez
ELLIS Alicante

adrian@ellisalicante.org
@arnaiztech

Francisco Escolano
ELLIS Alicante

@Fescolano

Nuria Oliver
ELLIS Alicante

@nuriaoliver

Edwin Hancock
University of York

Resources

Tutorial Webpage

https://ellisalicante.org/tutorials/GraphRewiring

Slides

https://ellisalicante.org/tutorials/GraphRewiring

Video

https://ellisalicante.org/tutorials/GraphRewiring

Code

https://github.com/ellisalicante/GraphRewiring-Tutorial

Outline

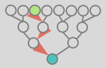
1. Motivation
- Graph Classification and Expressiveness
- Node Classification and Over-smoothing
- Desiderates

2. Graph Spectral Theory
- Average Cut Problem
- Fiedler Vector
- Laplacian and Dirichlet Energies
- Laplacian Eigenfunctions and Spectrum
- Spectral Theorem
- Spectral Commute times

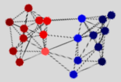
3. Transductive Graph Rewiring
- Diffusive Rewiring
- Cheeger constant
- Curvature Rewiring

4. Inductive Graph Rewiring
- CT and Lovász Bound
- CT and Sparsification
- CT and Directional Graph Networks
- CT-Layer
  - Loss function and CT-Layer
  - Learned CTE and CT distance
  - Experiments in Graph and Node Classification
  - CT-Layer as Differentiable Curvature
  - CT and Cheeger Constant
- GAP Layer
  - Network derivatives / Spectral Derivatives
  - Approximation of fielder vector

5. Rewiring in Graph Fairness
- Algorithmic Fairness
- Structure: a New Dimension
- Cause of Graph Bias
- Taxonomy of Definitions
- Graph Rewiring Methods for Fairness

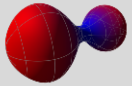


Motivation and Challenges



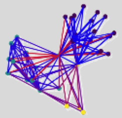
Introduction to Spectral Theory

Introduction to Spectral Theory



Transductive Rewiring

Transductive Graph Rewiring



Inductive Rewiring

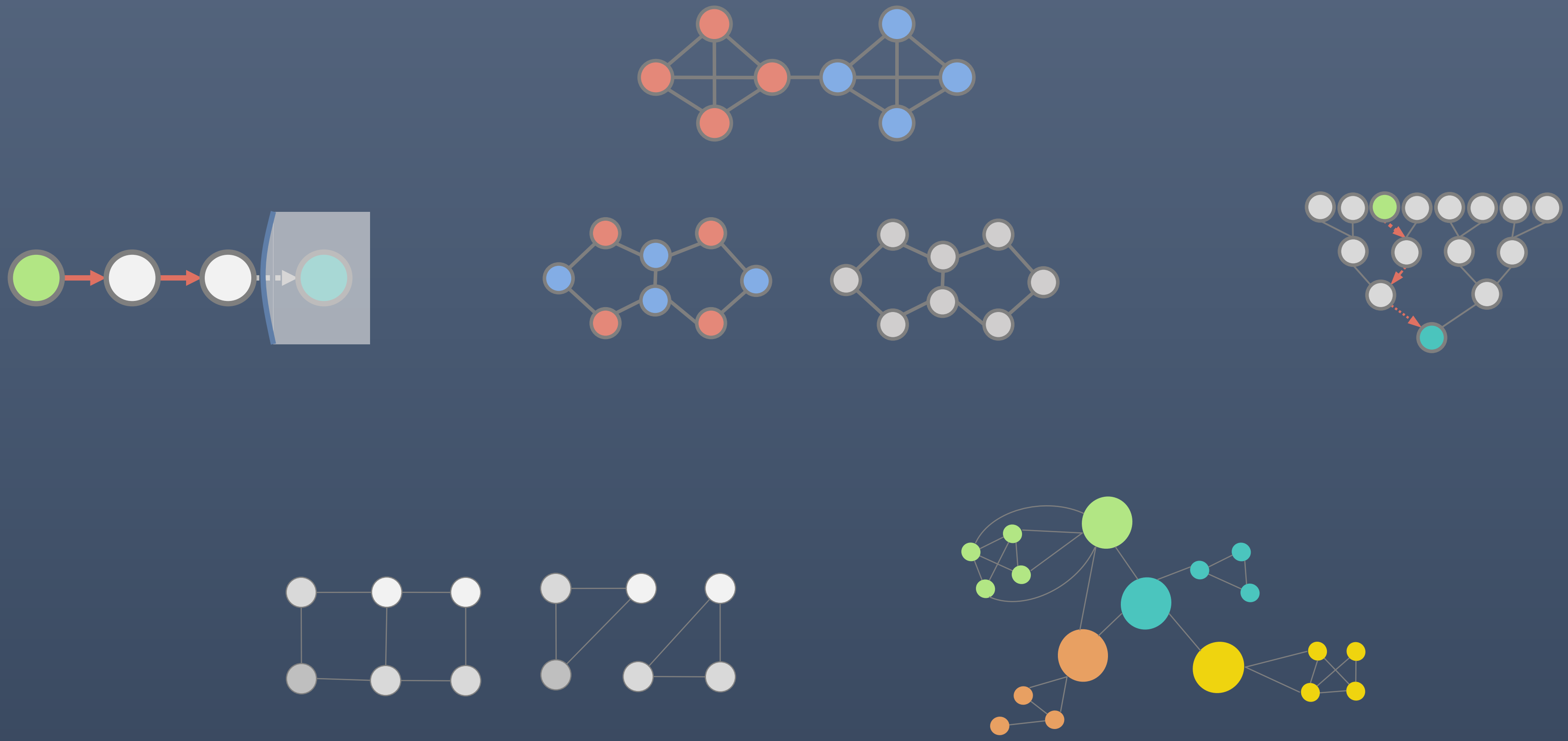
Lovász Bound and CT

CT-Layer

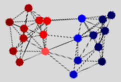
Graph Fairness

Panel Discussion

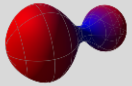
Motivation



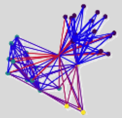
Motivation and Challenges



Introduction to Spectral Theory



Transductive Rewiring



Inductive Rewiring

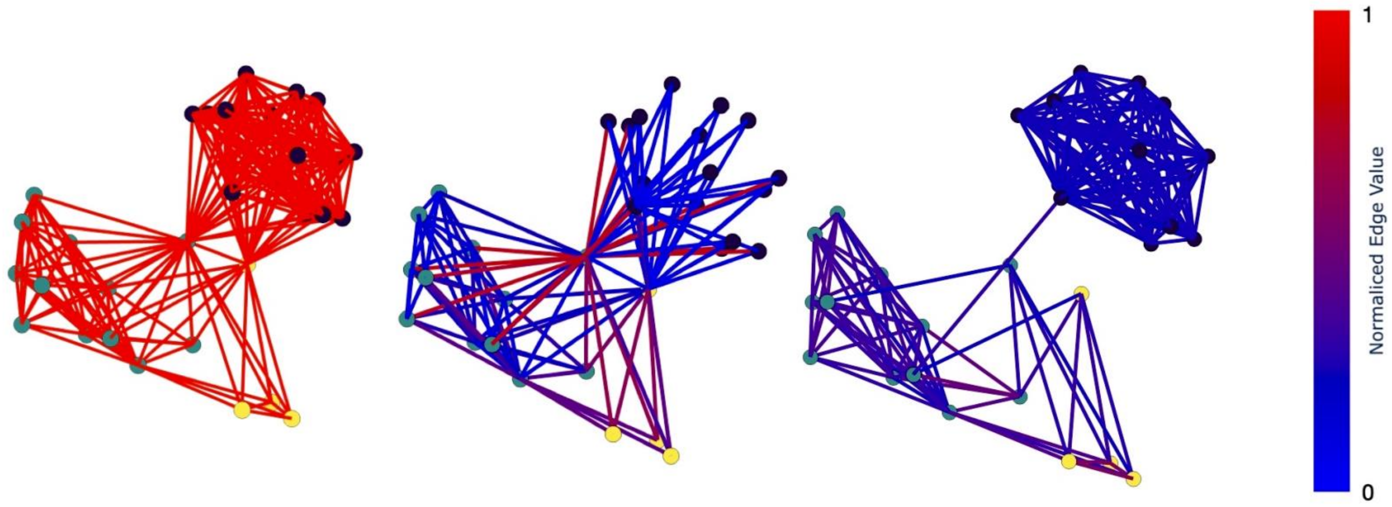
Graph Fairness

Panel Discussion

Motivation

What is (or should be) graph rewiring?

Graph Rewiring pursuits the **optimal graph structure** for the **downstream task**



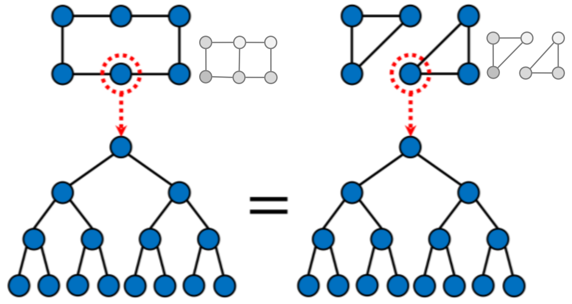
In GRAPH CLASSIFICATION, graph rewiring skeletonizes the graph so that the structure becomes more informative

- Given an input graph (left), **bottleneck-preserving** rewiring (center) discriminates graphs whose differences are in the bottlenecks themselves since intra-class edges are often removed or down-weighted.
- **Gap-minimization** rewiring (right) however, discriminates graphs whose differences are in the communities.
- **Example**: Web networks such as COLLAB are better discriminated by 'bottleneck-preserving rewiring but SBM-like networks with large bottlenecks are better discriminated by gap-minimization rewiring.

Motivation

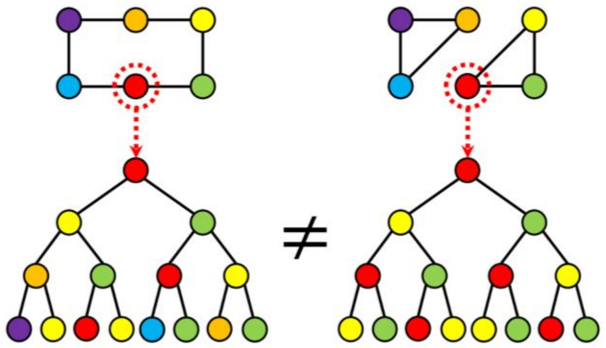
Graph Classification and Isomorphism

Most GNNs are as powerful as 1-WL test



Distance Encodings (DE) or **Positional Encodings** (PE) **make GNNs more powerful than 1-WL**

- **PE**: Random walk measures (e.g. shortest path, diameter, commute times), Spectral metrics (e.g. eigenvectors)
- **Expressiveness**: DE or PE provides strictly more expressive power than 1-WL test [Li, P. et al. 2020] [Velingker, A. et al. 2022]
- **Invariance**: Spectral GCN are permutation and sign equivariant [Lim, D. et al. 2022]
- **Usage**: Usually used as an extra node feature **or to control message aggregation**



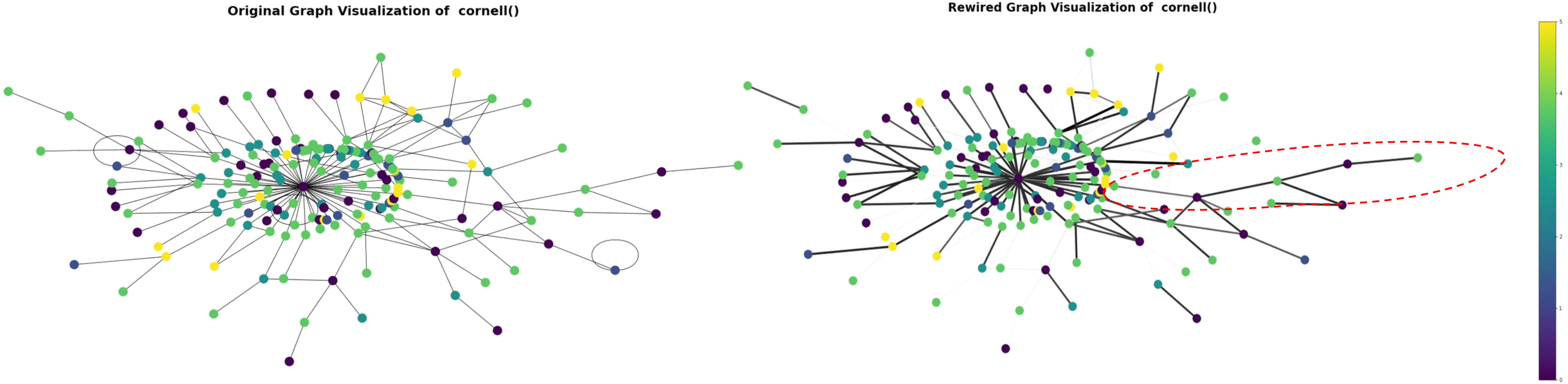
Use Spectral metrics to perform
Graph Rewiring

Bronstein, M. GNNs through the lens of differential geometry and algebraic topology. Blog Post, 2021. [Link]
Li, P., et al. "Distance encoding: Design provably more powerful neural networks for graph representation learning". In NeurIPS, 2020.
Lim, D., et al. "Sign and Basis Invariant Networks for Spectral Graph Representation Learning." arXiv preprint arXiv:2202.13013, 2022.
Velingker, A., et al. "Affinity-Aware Graph Networks." arXiv preprint arXiv:2206.11941, 2022.

Motivation

What is (or should be) graph rewiring?

Graph Rewiring pursuits the **optimal graph structure** for the **downstream task**



In NODE CLASSIFICATION, graph rewiring enables/disables information flow between nodes.

- **Homophilic** networks (where structure is correlated with class labels) are easy to rewire (e.g. reduce the gap).
- **Heterophilic** networks often require to increase the flow between heterophilic nodes.
- **In the figure above (Cornell):** distant green nodes can access the periphery of the hub while the gap is preserved.
- **Result**: classes with high heterophilic index are better classified

Motivation

Node Classification. Heterophily and Over-squashing.

GNNs were originally designed based on the smoothness principle

Homophily

Short-range tasks

[Zhu, J., et al., 2020]

$$h_{edges} = \frac{|\{(u,v) \in E: y_u = y_v\}|}{|E|}$$

$$\mathrm{H}_{ij}(E) = \frac{|\{(u,v) \in E: y_u i \wedge y_v = j\}|}{|\{(u,v) \in E: y_u = i\}|}$$

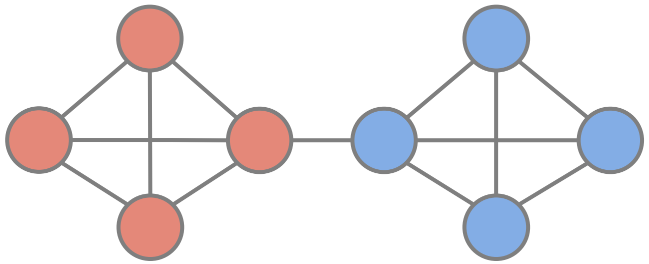
[Pei, H. et al., 2019]

$$h_{nodes} = \frac{1}{|V|} \sum_{v \in V} \frac{|\{u \in N(v): y_u = y_v\}|}{|N(v)|}$$

$$h_{class} = \frac{1}{|C|-1} \sum_{c \in C} \left[h_c - \frac{|C_c|}{n} \right]_+ , h_c = \frac{\sum_{v \in c} |\{u \in N(v): y_u = y_v\}|}{\sum_{v \in c} |N(v)|}$$

[Lim, D. et al., 2021]

i.e. Correlation between structure and labels



Dirichlet energies

$$h_{smooth} = \mathcal{E}(\mathbf{y}) = Tr[\mathbf{y}^T \mathbf{L} \mathbf{y}]$$

Assortativity

[Newman, M., 2002]

$h = \mathrm{r} =$ Pearson correlation coefficient between the degrees of linked nodes

Newman, M. "Assortative mixing in networks". Phys. Rev. Lett., 89, 2002.
Pei, H. et al. "Geom-GCN: Geometric GCNs". In ICLR, 2019.
Zhu, J., et al. "Beyond homophily in graph neural networks: Current limitations and effective designs". in NeurIPS, 2020
Lim, D., et al. "New benchmarks for learning on non-homophilous graphs". In WWW Workshop on GLB, 2021.

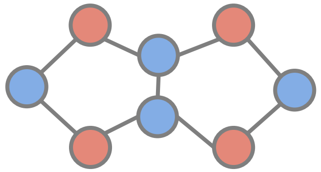
Motivation

Node Classification. Heterophily and Over-squashing.

r = radius
k = n layers
d = graph diameter

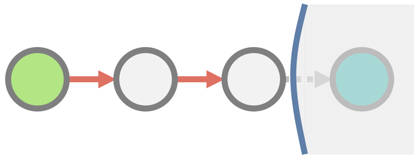
Heterophily → **Problem with higher radius**

Long-range tasks



Under-reaching

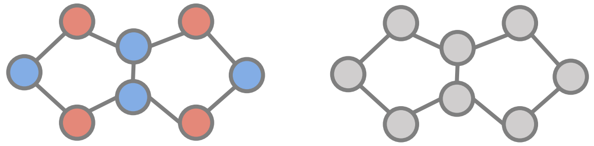
k < r



$\uparrow k$

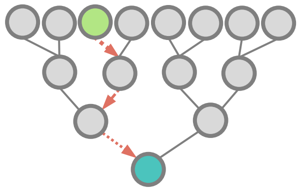
Over-smoothing

k > r ≈ d



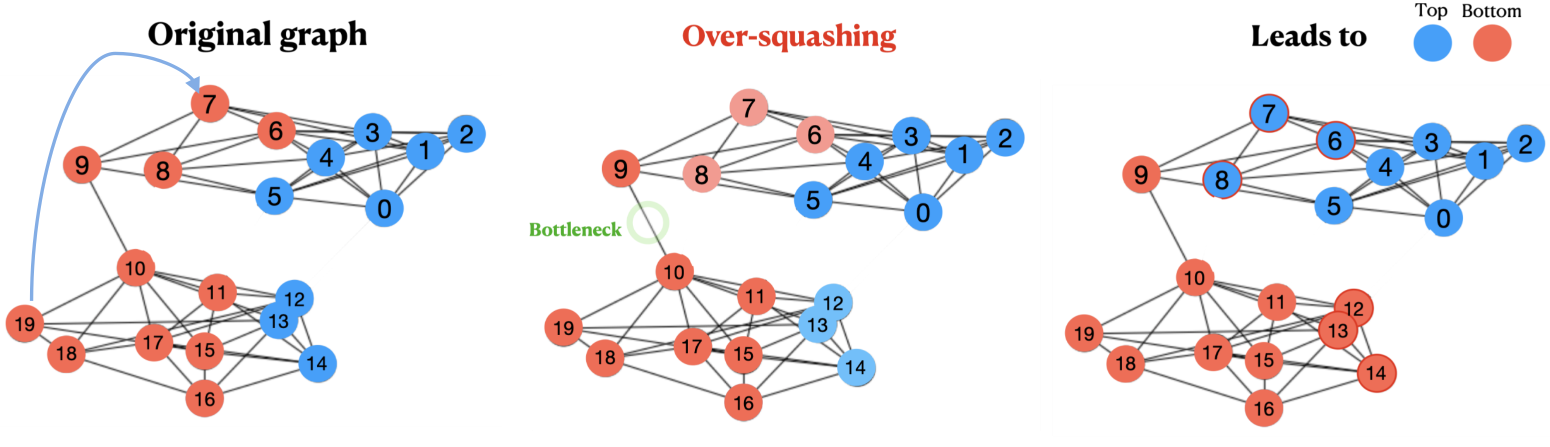
Over-squashing

node's receptive field increases exponentially



Graph Rewiring as a solution

- Connect distant nodes to overcome the three problems.
- E.g. increase bottleneck



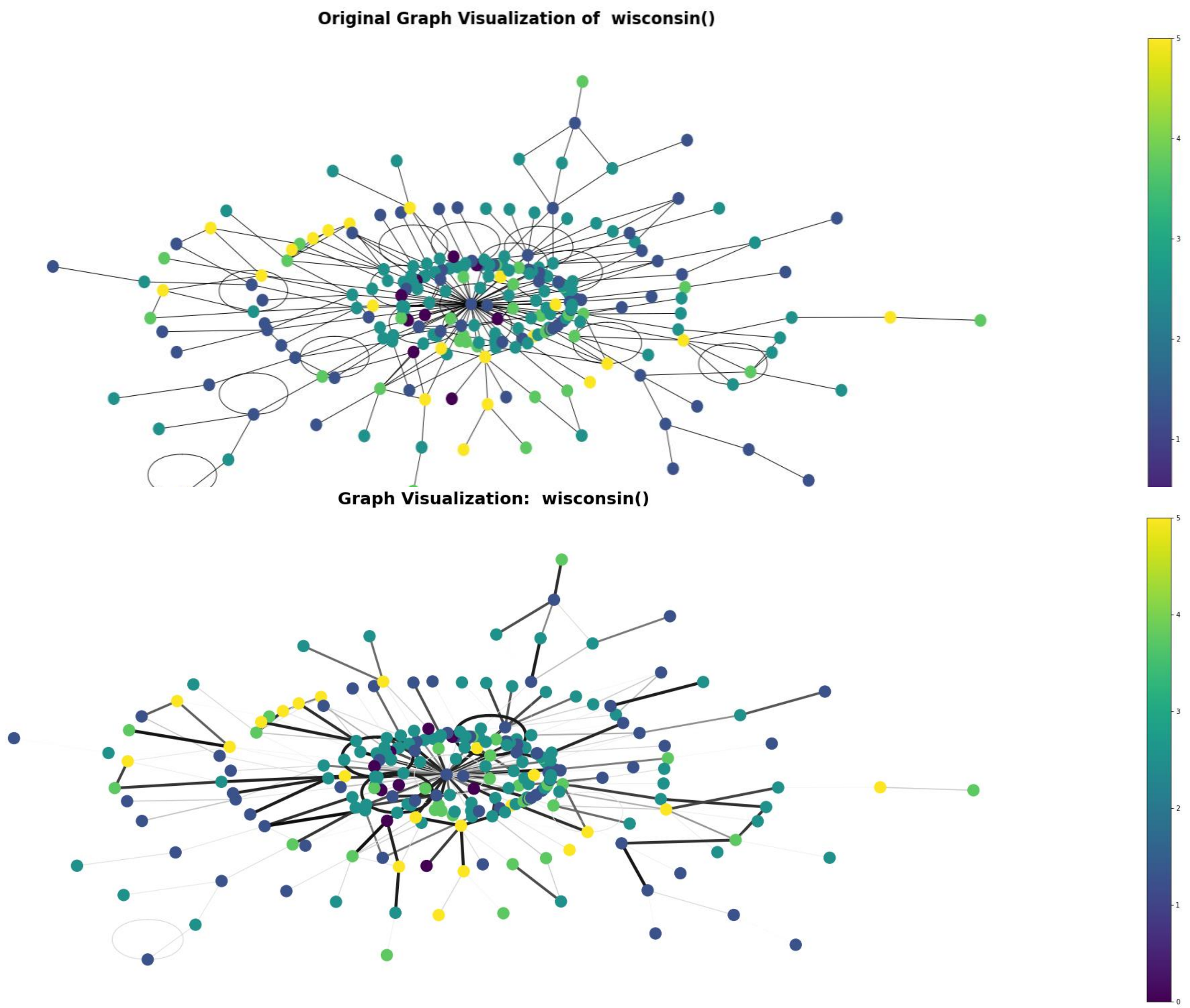
Dominant class in each community absorbs the other

Alon, U. and Yahav, E. "On the bottleneck of graph neural networks and its practical implications". In ICLR 2021

ellis ALICANTE

Motivation

Key Challenges – Desiderates of Graph Rewiring



- Principled
- Expressive
- Interpretable
- Task-aligned
- Parameter-Free

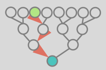
[Deac, A. et al., 2019]

- Global Propagation
- Remove bottlenecks
- Low Complexity
- No dedicated preprocessing
- Preserve the original structure

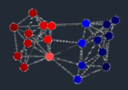
Deac, A., Lackenby, M. and Veličković, P. "Expander Graph Propagation". In Learning on Graphs Conference (LoG), 2022.

$$
\begin{array}{cccc}
 & \mathbf{\Phi} & \mathbf{\Lambda} & \mathbf{\Phi}^T \\
= & \begin{pmatrix} | & \vdots & | \\ \mathbf{v}_1 & \vdots & \mathbf{v}_n \\ | & \vdots & | \end{pmatrix} & \begin{pmatrix} \lambda_0 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{pmatrix} & \begin{pmatrix} - & \mathbf{v}_1 & - \\ \cdots & \cdots & \cdots \\ - & \mathbf{v}_n & - \end{pmatrix}
\end{array}
$$

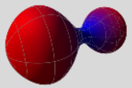
Introduction to Spectral Theory



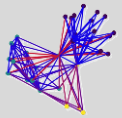
Motivation and Challenges



Introduction to Spectral Theory



Transductive Rewiring



Inductive Rewiring

Graph Fairness

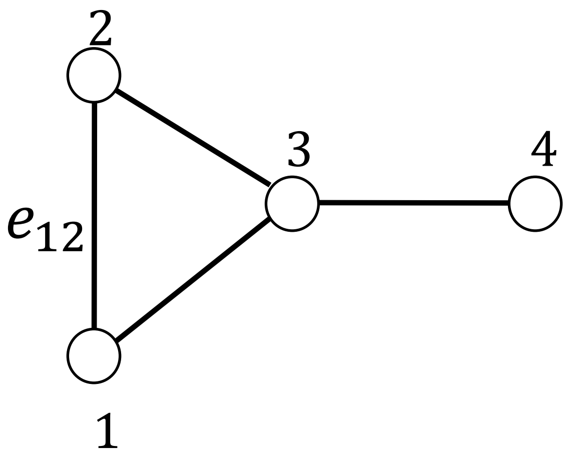
Panel Discussion

Graphs as Combinatorial Objects

Understanding the Graph Laplacian

- Undirected Graph

$$G = (V, E), V = \{1,2,\dots,n\} e_{ij} \in E \subseteq V \times V$$



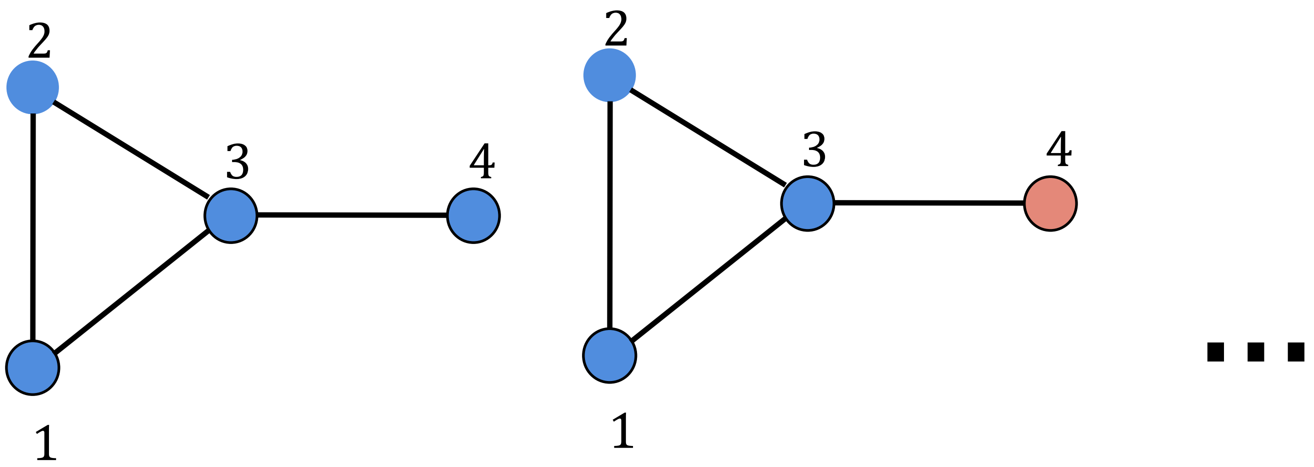
- Adjacency Matrix:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

A_{ij} As a variable

a_{ij}, w_{ij} As a weight

G as a Combinatorial Object: 2^n functions f



- Function over the nodes: $f: V \to \mathbb{R}$

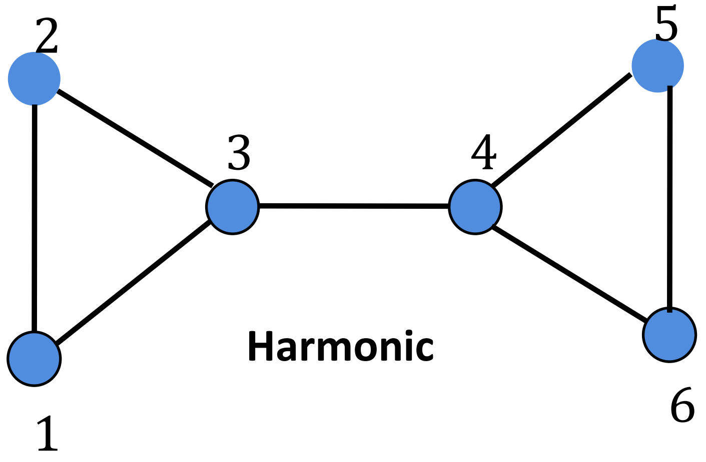
- Example: $f: V \to \{-1,1\}$

The Average Cut Problem

Understanding the Graph Laplacian

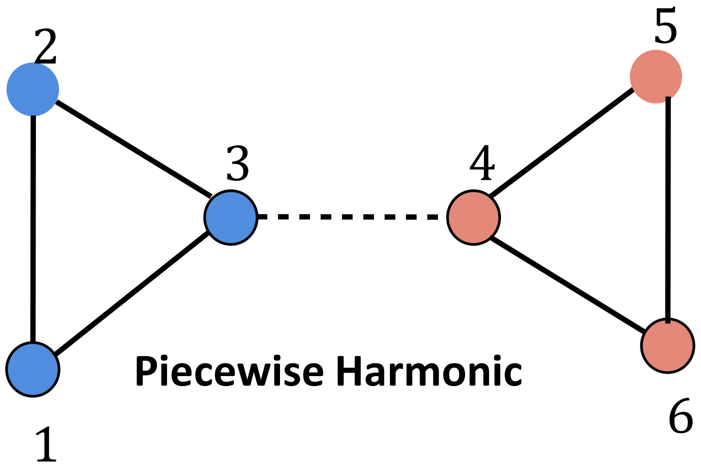
- What fs are more informative about G?

$$f(i) = f(j) \forall e_{ij}$$



$$f(i) = \frac{1}{|\mathcal{N}_i|} \sum_j a_{ij} f(i) = \frac{1}{d_i} \sum_j a_{ij} f(i)$$

$f(i) = f(j) \; \forall e_{ij}$ **BUT for** $i \in C_A, j \in C_B$



Vertex Partition

$$A \cup B = V$$
$$A \cap B = \emptyset$$

Cut

$$cut(A, B) = \sum_{i \in A, j \in B} a_{ij}$$

$$f = \min_{f \in \Omega} Acut(A, B)$$

$$Acut(A, B) = \frac{cut(A, B)}{|A|} + \frac{cut(A, B)}{|B|}$$

NP-Hard!

The Fiedler Vector

Understanding the Graph Laplacian

Fiedler's Theorem: Measures the **variability** of the optimal solution

$$\mathbf{x} \in \{-1,1\}^n \quad x_i = +1 \rightarrow i \in A, x_i = -1 \rightarrow i \in B$$

$\mathbf{x} \perp \mathbf{1}$ Minimal variability is $\lambda_1 = 0$, i.e. that of the harmonic function

The variability λ_2 of the **optimal partition** minimizes the ratio between the variability imposed by the structure of the graph and the unconstrained one!

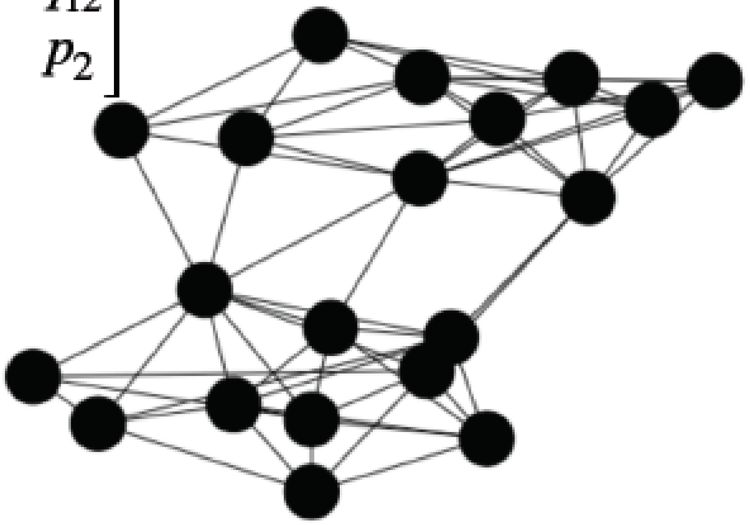
$$\lambda_2 = n \cdot min\left\{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}(x_i - x_j)^2}{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2} : \mathbf{x} \neq c \cdot \mathbf{1}, c \in \mathbb{R}\right\}.$$

The Fiedler Vector

Understanding the Graph Laplacian

$G \in SBM(\mathbf{P})$

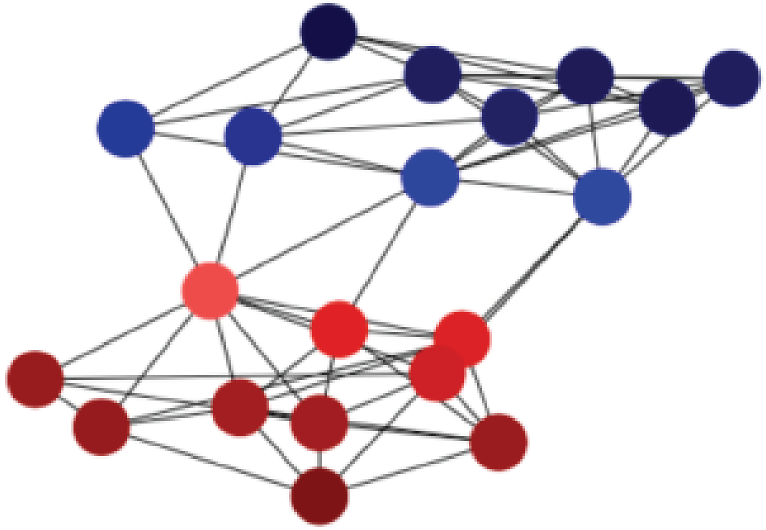
$$\mathbf{P} = \begin{bmatrix} p_1 & q_{12} \\ q_{12} & p_2 \end{bmatrix}$$



$\lambda_1 = 0$

Fiedler vector and its mapping over G

$\mathbf{x} =$
```
[[ 0.25351771]
 [ 0.25653022]
 [ 0.25387927]
 [ 0.16959223]
 [ 0.17525303]
 [ 0.24266911]
 [ 0.24454324]
 [ 0.1106077 ]
 [ 0.2855154 ]
 [ 0.19132828]
 [-0.25326518]
 [-0.24898363]
 [-0.15099328]
 [-0.17707374]
 [-0.1441682 ]
 [-0.2553871 ]
 [-0.24018788]
 [-0.25043498]
 [-0.19223463]
 [-0.27070759]]
```



$\lambda_2 = 8.37042078e-01$

The Combinatorial Laplacian
Understanding the Graph Laplacian

The combinatorial Laplacian
$\mathbf{L} = \mathbf{D} - \mathbf{A}, \mathbf{D} = diag(d_1, \dots, d_n)$

$$\mathbf{L} = \begin{bmatrix} d_1 & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & d_2 & \cdots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & d_n \end{bmatrix}$$

$$\forall i : \sum_j \mathbf{L}_{ij} = 0$$

Semidefinite Positive

$$Tr(\mathbf{f}^T \mathbf{L}\mathbf{f}) \geq 0, \forall \mathbf{f} \in \mathbb{R}^{n}$$

The trace of $\mathbf{L}$ is $\propto$ to the variability imposed by the structure of the graph (Fiedler's Thm)

$$Tr(\mathbf{f}^T \mathbf{L}\mathbf{f}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}(f_i - f_j)^2 = \frac{1}{2}\sum_{i\sim j}^{n}(f_i - f_j)^2$$

Dirichlet Energies!

Harmonicity

$$\boldsymbol{Lf} = \mathbf{f} - \mathbf{D}^{-1}\mathbf{A}\mathbf{f} \rightarrow f(i) = \frac{1}{d_i}\sum_j a_{ij} f(i)$$

Eigenfunctions and Spectrum

Understanding the Graph Laplacian

The **spectrum** and **eigenfunctions** of $\mathbf{L}$

Courant-Fisher Theorem:

$$\lambda_i = \max_{\mathcal{X}^i} \left(\min_{\mathbf{x} \perp \mathcal{X}^i, \mathbf{x} \neq \mathbf{0}} R(\mathbf{x}) \right)$$

Rayliegh Quotient:

$$R(\mathbf{x}) = \frac{\mathbf{x}^T \mathbf{L} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \in \mathbb{R}$$

Unconstrained Variability:

$$\mathbf{x}^T \mathbf{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x_j)^2$$

$$\mathbf{L}\mathbf{f}_1 = 0 \rightarrow \lambda_1 = 0$$

$$\mathbf{L}\mathbf{f}_2 = \lambda_2 \mathbf{f}_2$$ **Fiedler vector**

$$\vdots$$

$$\mathbf{L}\mathbf{f}_n = \lambda_n \mathbf{f}_n$$

$$\lambda_2 = n \cdot min \left\{ \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} (x_i - x_j)^2}{\sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x_j)^2} : \mathbf{x} \neq c \cdot \mathbf{1}, c \in \mathbb{R} \right\}$$

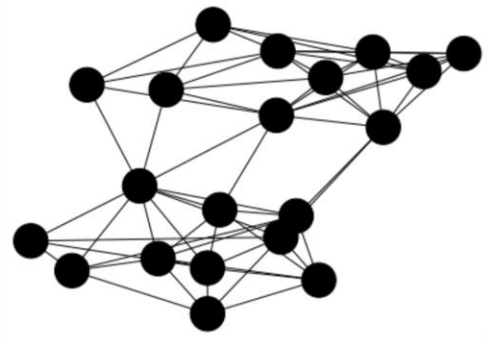
$$f_2 \perp \mathbf{f}_1, f_3 \perp \{\mathbf{f}_1, \mathbf{f}_2\}, \ldots \rightarrow \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$$

For connected graphs

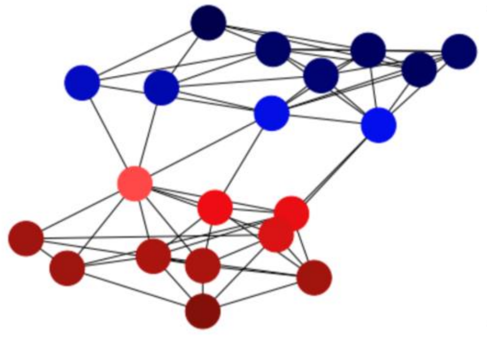
$$\lambda_i = \frac{\mathbf{f}_i^T \mathbf{L} \mathbf{f}_i}{\mathbf{f}_i^T \mathbf{f}_i}$$

Eigenfunctions of the Laplacian

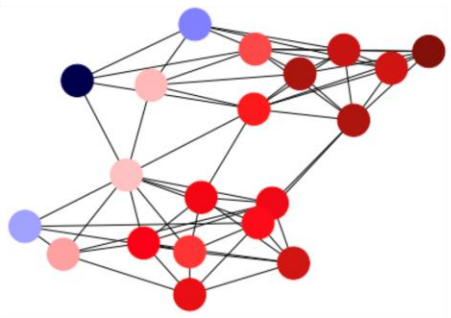
Understanding the Graph Laplacian



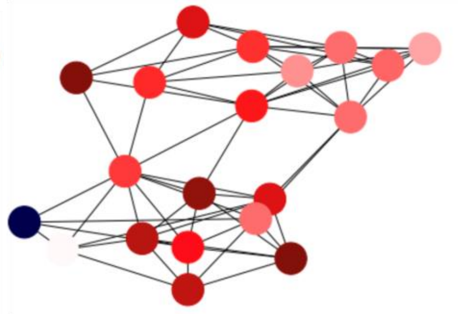
$\lambda_1 = 0$



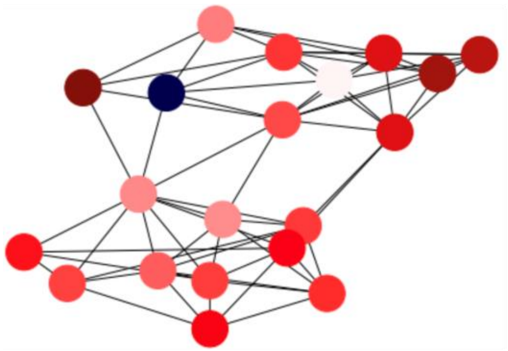
$\lambda_2 = 0.837$



$\lambda_3 = 3.389$

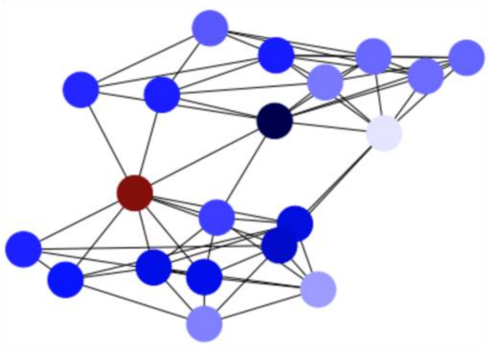


$\lambda_4 = 3.635$



$\lambda_5 = 4.514$

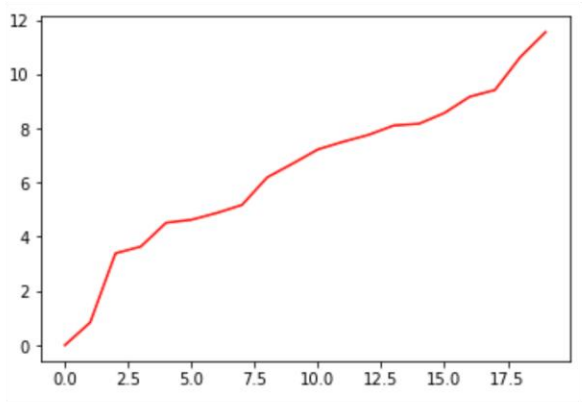
...



$\lambda_n = 11.547$

...

Spectrum



Spectral Theorem and Heat Kernels

Understanding the Graph Laplacian

Diffusion through Heat Kernels

Spectral Decomposition on L

$$\mathbf{L} = \mathbf{\Phi\Lambda\Phi}^T, \quad \mathbf{\Phi} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n], \ \mathbf{\Lambda} = diag(\lambda_1, \lambda_2, \dots, \lambda_n) \quad \rightarrow \quad \mathbf{L} = \sum_{i=1}^{n} \lambda_i \, \mathbf{f}_i \mathbf{f}_i^T$$

Solution of heat equation and measures information flow across edges of graph with time: $\boxed{\dfrac{\partial h_t}{\partial t} = -Lh_t}$

Matricial Exponential: Solution found by exponentiating Laplacian eigensystem

$$\mathbf{K}_t = \exp(-t\mathbf{L}) \ \rightarrow \Phi \exp(-t\Lambda)\Phi^T \ \rightarrow \Phi \begin{bmatrix} e^{-t\lambda_1} & 0 & \dots & 0 \\ 0 & e^{-t\lambda_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & e^{-t\lambda_n} \end{bmatrix} \Phi^T$$

$$\mathbf{K}_t = \sum_{i=1}^{n} e^{-t\lambda_i} \mathbf{f}_i \mathbf{f}_i^T$$

$$exp(-t\mathbf{L}) = \mathbf{I} - t\mathbf{L} + \frac{t^2}{2!}\mathbf{L}^2 - \frac{t^3}{3!}\mathbf{L}^3 + \dots$$

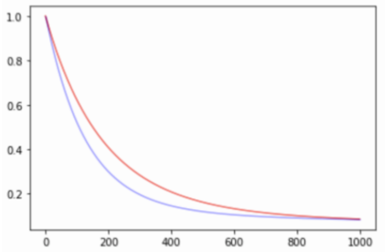
$$t \approx 0 : \mathbf{K}_t \approx \mathbf{I} - t\mathbf{L}$$

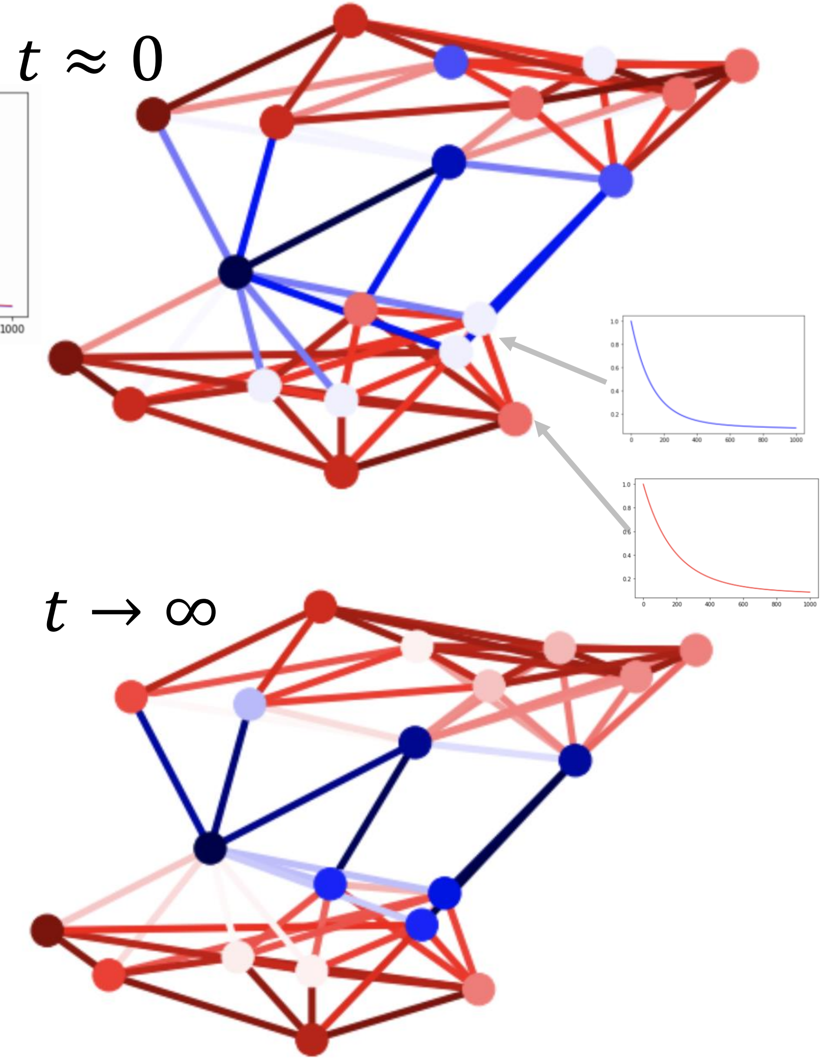
$$t \rightarrow \infty : K_t \approx e^{-t\lambda_2} \mathbf{f}_2 \mathbf{f}_2^T$$

Spectral Theorem and Heat Kernels

Understanding the Graph Laplacian

Diffusion through Heat Kernels





Motivation

- At time $t = 0$, each node has a unit of heat.
- The heat diffuses as $t \to \infty$ driven by $\mathbf{L}$ (actually by its harmonization behavior).
- The Heat Kernel Signature (HKS) of a node is its heat trace over time.
- Heat $H_t(i,j)$ is the probability that a lazy random walk starting at node i hits node j at time t.

Commute Time and its Embedding

Motivation

$$CT(i,j) = H(i,j) + H(j,i)$$

Time needed by a random walk to hit j (Hitting time) and return. More respectful with G's structure than SP !

$$R(i,j) = \frac{CT(i,j)}{vol(G)}$$

$$CT(u,v) = vol(G)\sum_{i=2}^{n}\frac{1}{\lambda_i}(\mathbf{f}_i(u) - \mathbf{f}_i(v))^2$$

Sum of divergences between eigenfunctions pinpointed at u and v but downweighed by the eigenvalue

Smoothest eigenfunctions contribute less (btw their λ_i is smaller) whereas the contribution of high variance eigenfunctions is reduced by their large inverse λ_i

Commute Times

Understanding the Graph Laplacian

Commute Time and its Embedding

$$CT(u,v) = vol(G)\sum_{i=2}^{n}\frac{1}{\lambda_i}(\mathbf{f}_i(u) - \mathbf{f}_i(v))^2$$

$$\downarrow$$

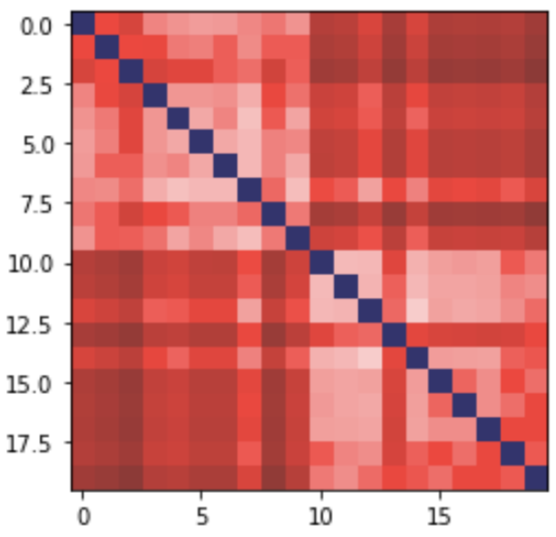
$$\Theta = \sqrt{vol(G)}\Lambda^{-1/2}\Phi^T$$

$$CT(u,v) = vol(G)\sum_{i=2}^{n}\frac{1}{\lambda_i'}\left(\frac{\mathbf{g}_i(u)}{\sqrt{d_u}} - \frac{\mathbf{g}_i(v)}{\sqrt{d_v}}\right)^2$$

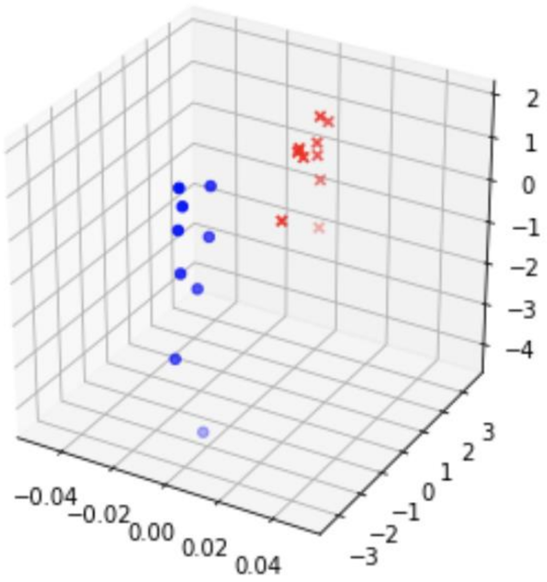
$$\downarrow$$

$$\Theta = \sqrt{vol(G)}\Lambda'^{-1/2}\Phi'^T\mathbf{D}^{1/2}$$

CT matrix:



CT Embedding:



CTEmbedding in the cols of Θ

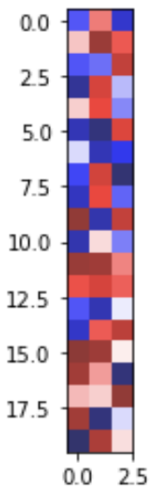
Commute Times

Understanding the Graph Laplacian

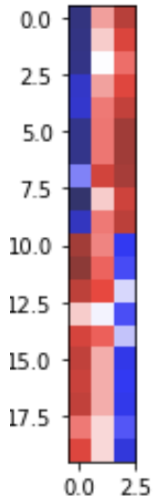
Commute Time and its Embedding

$$\Theta^* = \min_{\Theta\in\mathbb{R}^{n\times d}:\Theta\Theta^T=\mathbf{I}} Tr[\Theta^T\mathcal{L}\Theta] \rightarrow min L = Tr[\Theta^T\mathcal{L}\Theta] + \lambda_{reg} \parallel \Theta\Theta^T-\mathbf{I} \parallel^2$$

$$\frac{\partial L}{\partial \Theta} = 2\mathcal{L}\Theta + 4\lambda_{reg}\Theta(\Theta\Theta^T-\mathbf{I})$$

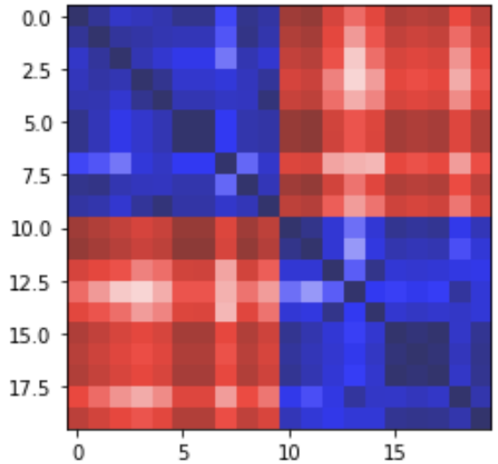


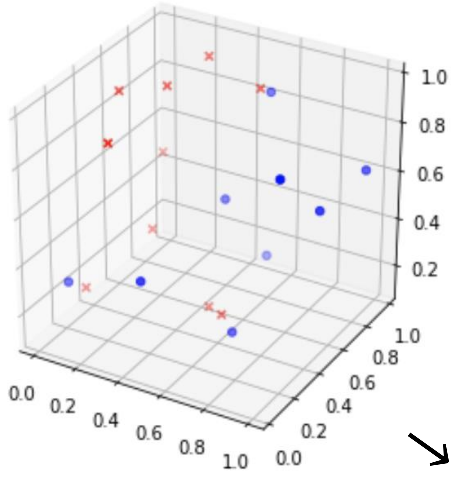
→

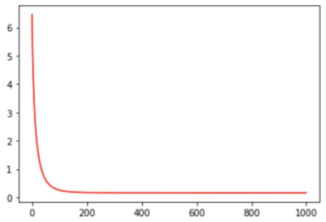


→

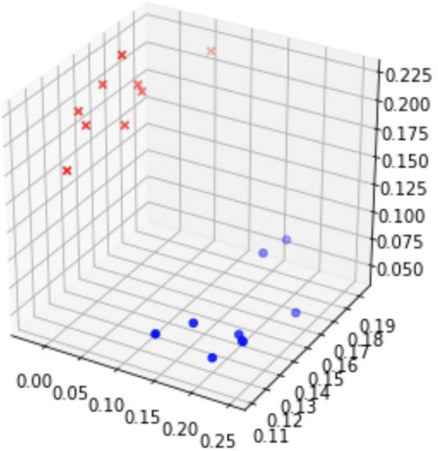
Distance matrix:

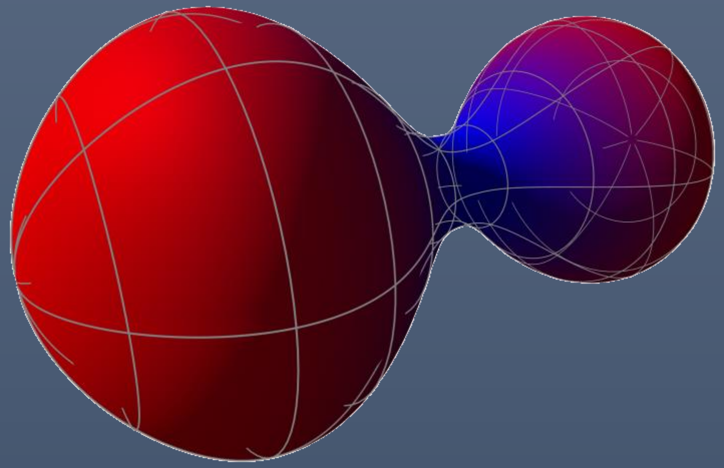




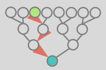


↘

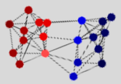




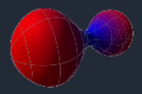
Transductive Graph Rewiring



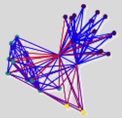
Motivation and Challenges



Introduction to Spectral Theory



Transductive Rewiring



Inductive Rewiring

Graph Fairness

Panel Discussion

Diffusive Rewiring

Motivation and basic equations

Diffusion processes provide principled methods for linking distant nodes [Klicpera et al. 2019]

- **Improving Message Passing**: Spatial MPNNs **need** deep layers to leverage high-order (distant) neighborhoods.
- **Structural Noise**: Edges in real graphs are often noisy or not correlated with the distribution of nodal features.
- **Spectral principles**: Spectral GNNs allow high-order neighborhoods but are not inductive for unseen graphs
- **GDC/DIGL**: Diffuse (PageRank/RW with restart, Heat Kernels) + sparsify + threshold as an **alternative message passing.**

Parameterized

PPR: $S = \alpha \, (I_n + (\alpha - 1)A)^{-1}$

Alpha

Top-K or epsilon for thresholding edges

Heat: $S = e^{t(A - I_n)}$

t

Top-K or epsilon for thresholding edges

Powers to the transition matrix

$$\boldsymbol{S} = \sum_{k=0}^{\infty} \theta_k \boldsymbol{T}^k$$

$$\sum_{k=0}^{\infty} \boldsymbol{T}^k = (\boldsymbol{I} - \boldsymbol{T})^{-1}$$

$$\theta_k = \alpha(1-\alpha)^k$$

$$\boldsymbol{S} = \alpha \sum_{k=0}^{\infty} ((1-\alpha)\boldsymbol{T})^k$$

$$\boldsymbol{S} = \alpha(\boldsymbol{I} - (1-\alpha)\boldsymbol{T})^{-1}$$

Row-stochastic matrix

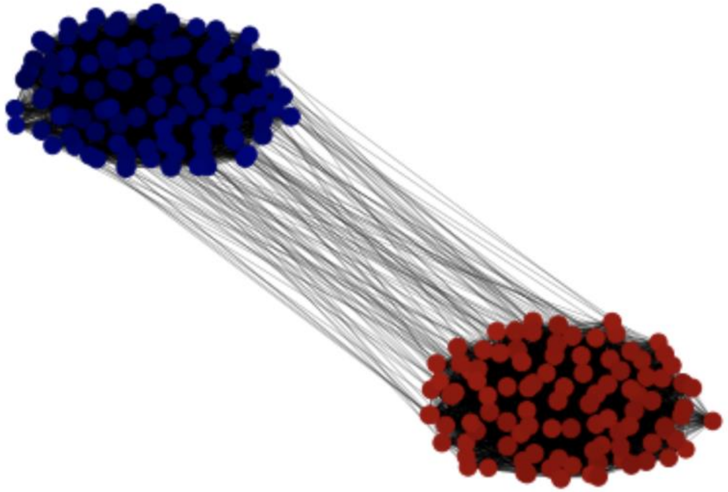
Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In Advances in Neural Information Processing Systems, 2019. URL https://proceedings.neurips.cc/paper/ 2019/file/23c894276a2c5a16470e6a31f4618d73-Paper.pdf. 3, 8, 23

Diffusive Rewiring

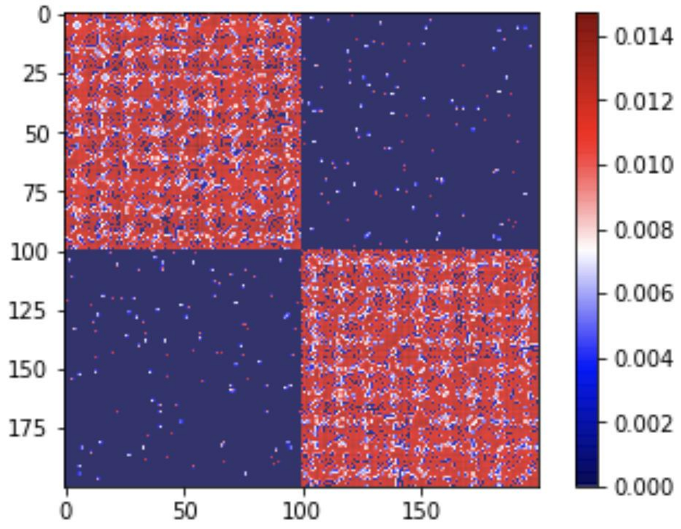
Diffusion works as a low—pass filter of structural noise [Klicpera et al. 2019]

- **Trivial choice of T (random walker):** $T \equiv T_{rw} = D^{-1}A$
- **Interpretation of:** T^k(i,j) probability of hitting j from i in k-steps. Hop aggregation: $\theta_1 T + \theta_2 T^2 + \theta_3 T^3 + \cdots$
- $k \to \infty$: Hitting probability is proportional to degree. But more distant modes can be reached -> Structural Smoothing

Basic SBM

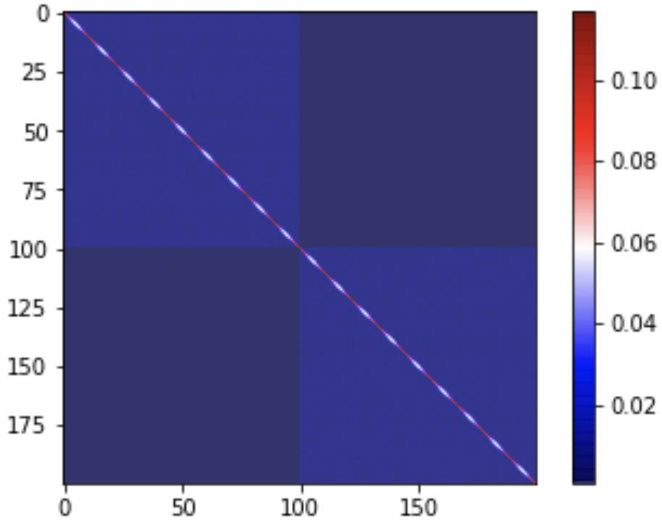


Structural Noise (white pixels)



$$T \equiv T_{sym} = (I + D)^{-1/2}(I + A)(I + D)^{-1/2}$$

Structural smoothing



Self Loops

$$S = \alpha(I - (1-\alpha)T)^{-1}$$

Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In Advances in Neural Information Processing Systems, 2019. URL https://proceedings.neurips.cc/paper/ 2019/file/23c894276a2c5a16470e6a31f4618d73-Paper.pdf. 3, 8, 23

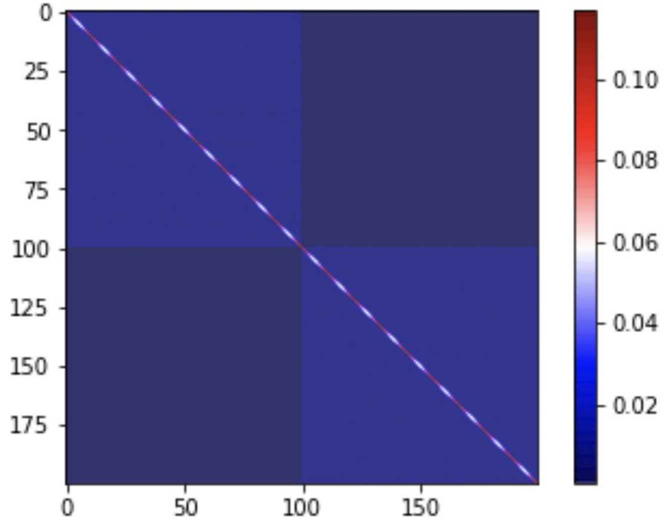
Diffusive Rewiring

Analysis

Sparsification and thresholding after diffusion [Nassar et al. 2015]

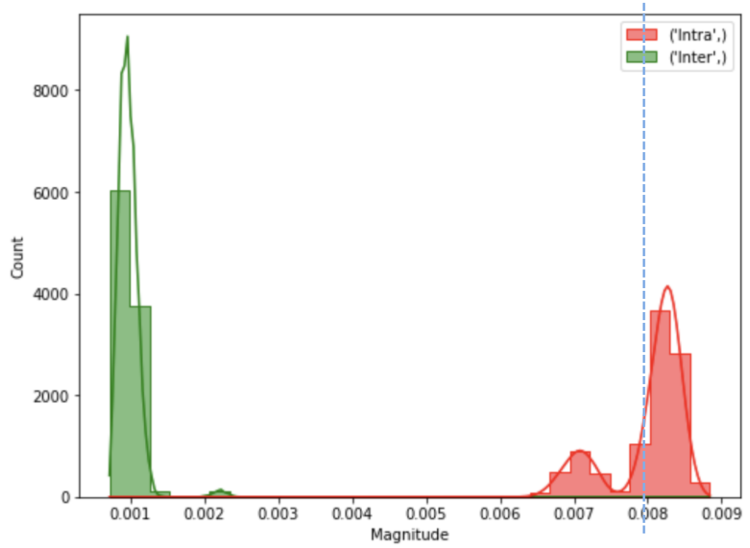
- Sparsification and thresholding: $\widetilde{\boldsymbol{S}} = \boldsymbol{S} * (S \geq \varepsilon)$
- Why $\widetilde{\boldsymbol{S}}$? Limit distribution of $\boldsymbol{S}$ is somewhat sparse (some nodes maybe not visited). This is "localization".
- Sparsification is enabled by localization! Perturbation mostly affects to highest and lowest eigenvalues.

After sparsification



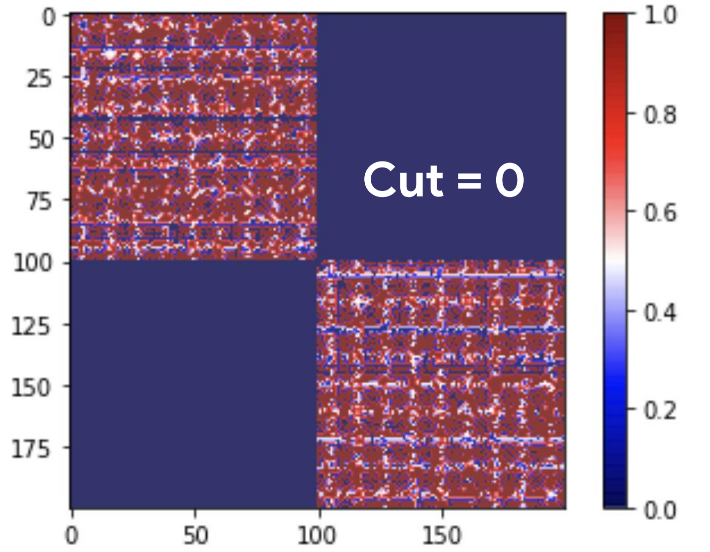
$T_{sym}^{\tilde{S}} \equiv D_{\tilde{S}}^{-1/2} \widetilde{\boldsymbol{S}} \, \boldsymbol{D}_{\tilde{\boldsymbol{S}}}^{-1/2}$

Edge magnitude



$\boldsymbol{S}(\boldsymbol{i},\boldsymbol{j}), (i \in \boldsymbol{V}_a, j \in \boldsymbol{V}_b)$ vs $\boldsymbol{S}(\boldsymbol{i},\boldsymbol{j}), (i \in \boldsymbol{V}_a, j \in \boldsymbol{V}_a)$

Final thresholding




```
T_S_th = T_S_zeroD>0.008
```

Huda Nassar, Kyle Kloster, and David F. Gleich. Strong Localization in Personalized PageRank Vectors. In International Workshop on Algorithms and Models for the Web Graph (WAW), 2015.
GITHUB: https://github.com/gasteigerjo/gdc and also recently incorporated to Pytorch Geometric.

Curvature

The Cheeger Constant

The Cheeger Constant is a separator problem

- Given a graph G, remove as few edges as possible to disconnect the graph into two parts of almost equal size
- Solving this problem implies exploring the $2^{|V|}$ subsets $S \subseteq V$ of the graph.
- Each one induces a partition $S \cup \bar{S} = \mathrm{V}, S \cap \bar{S} = \emptyset$

edges in the bottleneck

$$\boldsymbol{h_G} = \min_{S \subseteq V} h_S \,, \qquad h_S = \frac{cut(S,\bar{S})}{\min(\mathrm{vol}(S), \mathrm{vol}(\bar{S}))}$$

Volume of the community

$$cut(S,\bar{S}) = |\{(u,v) : u \in S, v \in \bar{S}\}|$$

Number of edges in the bottleneck

Minimal edge density in the partition

However, this quantity can be **spectrally bounded (and it bounds the spectra)**

$$\frac{\lambda_2}{2} \leq h_G < \sqrt{2\lambda_2} \quad \text{and} \quad 2h_G \leq \lambda_2 < \frac{h_G^2}{2}$$

λ_2 is the first non-trivial eigenvalue of the normalized Laplacian of G

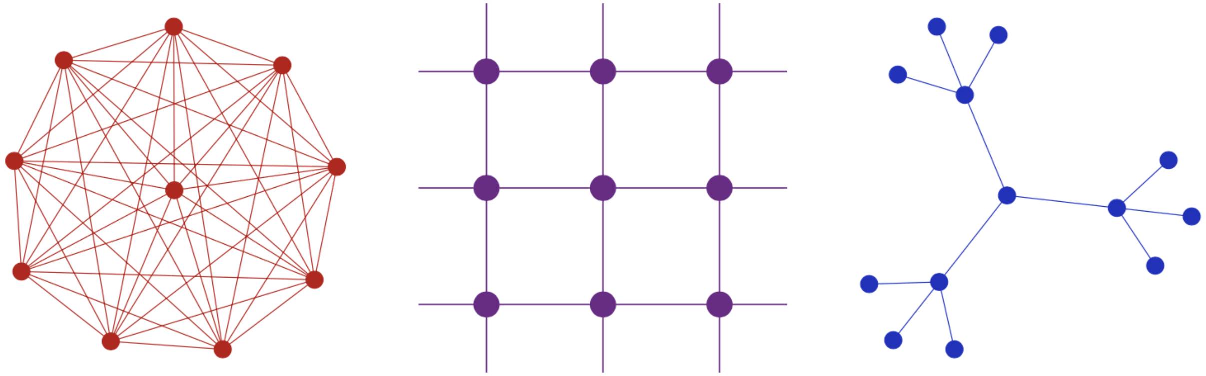
Fan RK Chung. Spectral Graph Theory. American Mathematical Society, 1997. URL https://mathweb.ucsd.edu/~fan/research/revised.html

Curvature

The Cheeger Constant

Since graphs encode manifolds, curvature (positive, negative or zero) quantifies the dispersion of geodesics (e.g. shortest paths) : [Devrient and Lambiotte. 2022]

- Zero: geodesics remain parallel (e.g. grid)
- Positive: geodesics converge (e.g. clique)
- Negative: geodesics diverge (e.g. trees)



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

Edge curvature : [Topping el al., 2022]

- $\#_{\triangle}(i,j)$: Triangles based at (i,j)
- $\#^{i}_{\blacksquare}(i,j)$: Neighbors of i forming a 4-cycle based on (i,j) without diagonals inside.
- $\gamma_{max}(i,j)$: Maximal number of 4-cycles based at (i.j) traversing a common node

Karel Devriendt and Renaud Lambiotte. Discrete curvature on graphs from the effective resistance. arXiv preprint arXiv:2201.06385, 2022. doi: 10.48550/ARXIV.2201.06385. URL https://arxiv.org/abs/2201. 06385. 2, 6, 7, 18

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=7UmjRGzp-A. 2, 3, 6, 8, 18, 23

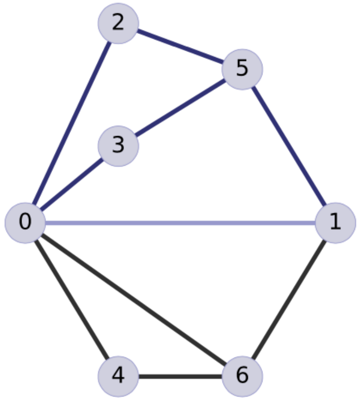
Curvature

Intuition

Balanced forman curvature

$Ric(i,j) = 0$ if $\min\{d_i, d_j\} = 1$

$$Ric(i,j) = \frac{2}{d_i} + \frac{2}{d_j} - 2 + 2\frac{|\#_{\Delta}(i,j)|}{\max\{d_i,d_j\}} + \frac{|\#_{\Delta}(i,j)|}{min\{d_i,d_j\}} + \frac{(\gamma_{\max})^{-1}}{\max\{d_i,d_j\}}(|\#_{\blacksquare}^{i}| + |\#_{\blacksquare}^{j}|)$$



d_0=5, d_1=3

$|\#_{\Delta}(0,1)|$ =1 given by triangle {1, 6, 0}

$\#_{\blacksquare}^{0}(0,1)$ = {2, 3} without 4,6 because triangle {1,6,0}

$\#_{\blacksquare}^{1}(0,1)$ = {5} without 4,6 because triangle {1,6,0}

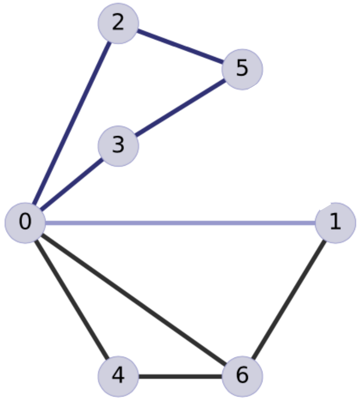
$\gamma_{max}(0,1)$ = 2 from the two 4-cycles passing through node 5.

$$Ric(0,1) = \frac{2}{5} + \frac{2}{3} - 2 + 2\frac{1}{5} + \frac{1}{3} + \frac{(2)^{-1}}{5}(2+1) = \frac{6+10}{15} - 2 + \frac{6+5}{15} + \frac{5}{10} = -2 + \frac{22}{15} + \frac{3}{10} = -2 + \frac{44+9}{30} > = -2 + \frac{51}{30} = -0.23 < 0$$

Balanced forman curvature

$Ric(i,j) = 0$ if $\min\{d_i, d_j\} = 1$

$$Ric(i,j) = \frac{2}{d_i} + \frac{2}{d_j} - 2 + 2\frac{|\#_\Delta(i,j)|}{\max\{d_i,d_j\}} + \frac{|\#_\Delta(i,j)|}{min\{d_i,d_j\}} + \frac{(\gamma_{\max})^{-1}}{\max\{d_i,d_j\}}(|\#^i_\blacksquare| + |\#^j_\blacksquare|)$$



d_0=5, d_1=2

$|\#_\Delta(0,1)|$ =1 given by triangle {1, 6, 0}

$\#^0_\blacksquare(0,1)$ = {2} without 3 and without 4,6 because triangle {1,6,0}

$\#^1_\blacksquare(0,1)$ = ∅ without 5 and without 4,6 because triangle {1,6,0}

$\gamma_{max}(0,1)$ = 1 from the 4-cycle passing through node 5.

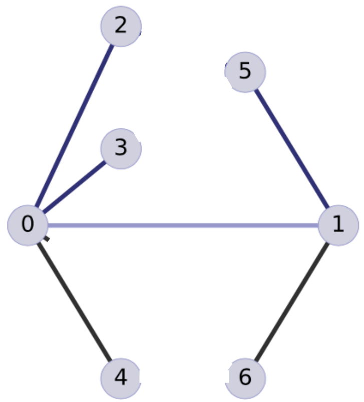
$$Ric(0,1) = \frac{2}{5} + \frac{2}{2} - 2 + 2\frac{1}{5} + \frac{1}{2} + \frac{(1)^{-1}}{5}(1 + 0) = \frac{4+10}{10} - 2 + \frac{4+5}{10} + \frac{1}{5} = -2 + \frac{23}{10} + \frac{1}{5} = -2 + \frac{25}{10} = 2.5 > 0$$

Jake Topping, et al. "Understanding over-squashing and bottlenecks on graphs via curvature". In ICLR, 2022. URL.

Balanced forman curvature

$Ric(i,j) = 0$ if $\min\{d_i, d_j\} = 1$

$$Ric(i,j) = \frac{2}{d_i} + \frac{2}{d_j} - 2 + 2\frac{|\#_{\Delta}(i,j)|}{\max\{d_i,d_j\}} + \frac{|\#_{\Delta}(i,j)|}{\min\{d_i,d_j\}} + \frac{(\gamma_{\max})^{-1}}{\max\{d_i,d_j\}}(|\#^{i}_{\blacksquare}| + |\#^{j}_{\blacksquare}|)$$



d_0=4, d_1=3

$|\#_{\Delta}(0,1)|$ =0 no triangle based at (0,1)

$\#^{0}_{\blacksquare}(0,1)$ = Ø no 4-cycle based at (0,1)

$\#^{1}_{\blacksquare}(0,1)$ = Ø no 4-cycle based at (0,1)

$\gamma_{max}(0,1)$ = 0 no 4-cycle

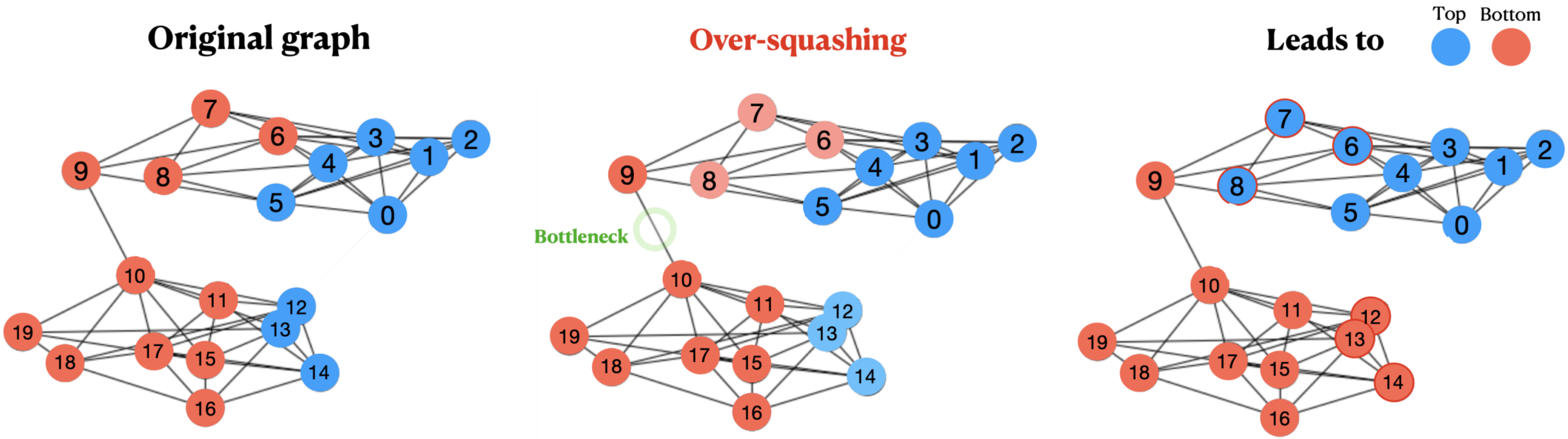
Strong bottleneck

$$Ric(0,1) = \frac{2}{4} + \frac{2}{3} - 2 + 2\frac{0}{4} + \frac{0}{3} + \frac{(0)^{-1}}{4}(0 + 0) = \frac{6+8}{12} - 2 + 0 + 0 = -2 + \frac{14}{12} = -2 + \frac{14}{12} = -0.83<0$$

Curvature
Intuition

Balanced forman curvature

- Edges with very negative curvature (>-2) create bottlenecks and thus over-squashing

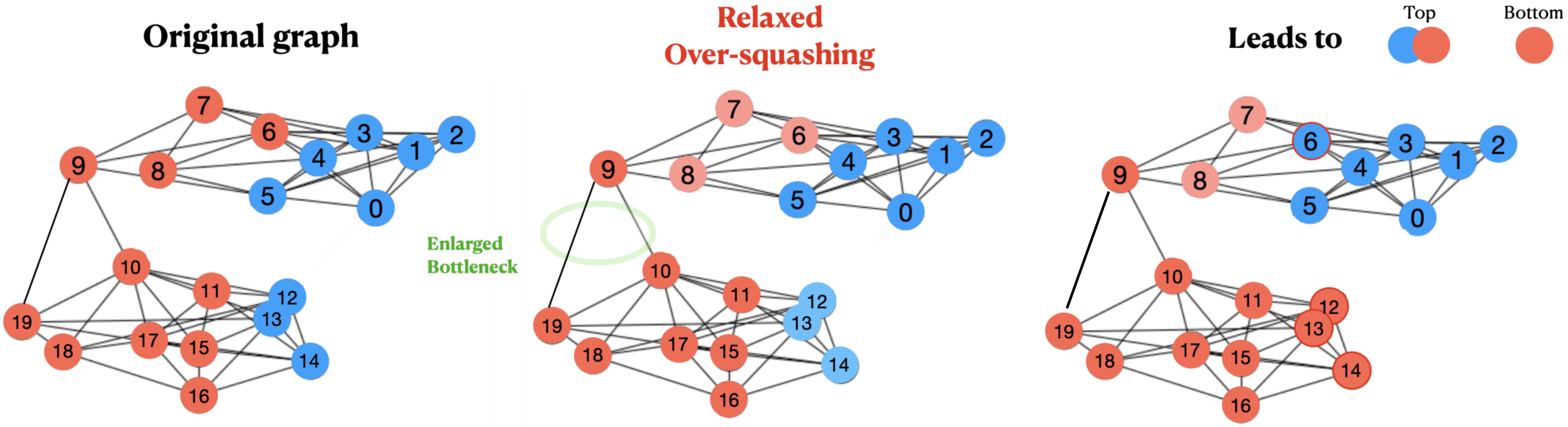


Dominant class in each community absorbs the other

Jake Topping, et al. “Understanding over-squashing and bottlenecks on graphs via curvature”. In ICLR, 2022. URL.

Balanced forman curvature

- Enlarging the bottlenecks reduces over-squashing



Dominant class in each community may NOT absorb the other

Jake Topping, et al. "Understanding over-squashing and bottlenecks on graphs via curvature". In ICLR, 2022. URL.

Curvature

The SRDF ALGORITHM

Stochastic Discrete Ricci Flow (SDRF)

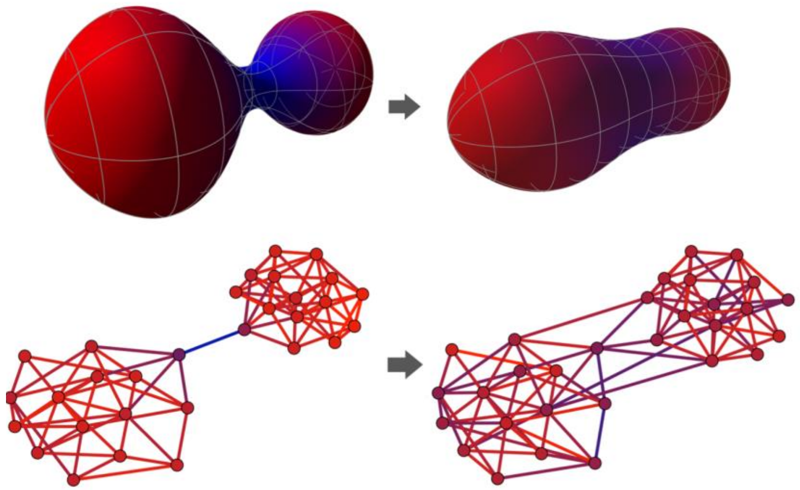
Algorithm 1: Stochastic Discrete Ricci Flow (SDRF)

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;



SURGICAL REWIRING:

Minimal Ricci curvature: Best candidate edge to improve.

Sample neighboring edges with probability propto improvement

Remove Edge with maximal Ricci curvature

CUDA REQUIRED

$$Ric(i,j) > k > 0 \; \forall (i,j) \Rightarrow \frac{\lambda_2}{2} \geq h_G \geq \frac{k}{2}$$

Jake Topping, et al. "Understanding over-squashing and bottlenecks on graphs via curvature". In ICLR, 2022. URL.

Curvature vs Diffusive Rewiring

Analysis

Diffusion works as a low—pass filter of structural noise [Klicpera et al. 2019]

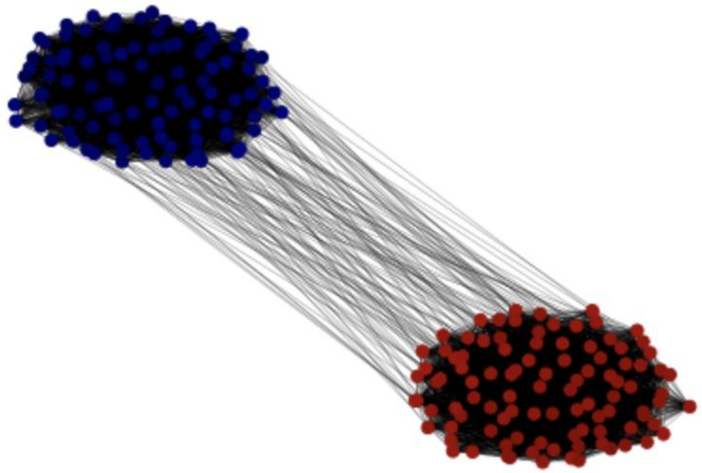
SDRF is quirurgical on behalf of a structural test for each edge [Topping el al., 2022]

- **The Cheeger constant of SGD/DIGL is controlled by that of SDRF:** $h_{S,\alpha} \leq \frac{(1-\alpha)}{\alpha}\frac{d_{avg}(S)}{d_{min}(S)} h_S$
- **SDRF preserves more the structure than SGD/DIGL (which may remove the cut)**

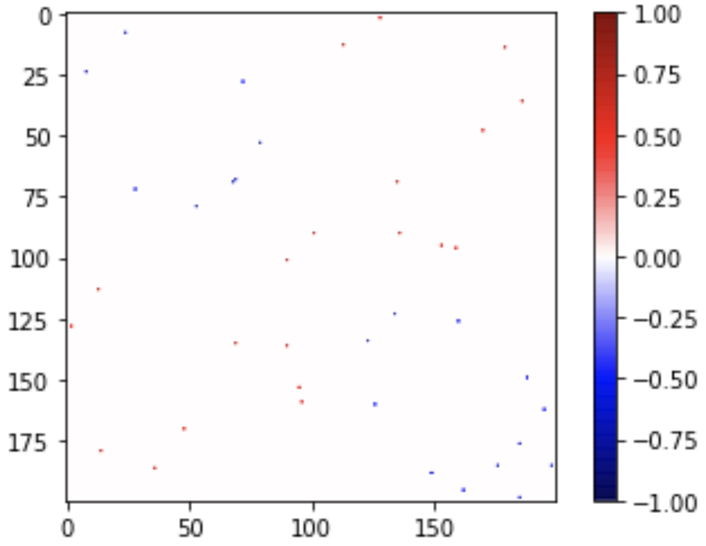
$\lambda_{2,SDRF} = 0.0297$

$\lambda_{2,G} = 0.0272$

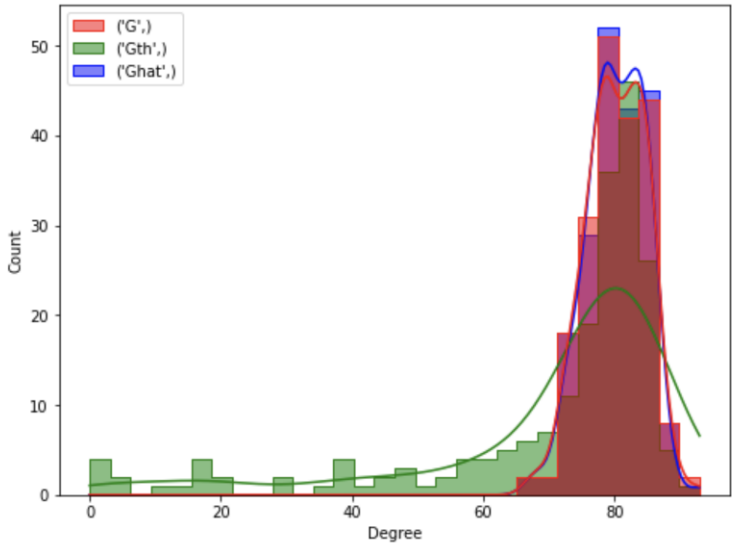
After SDRF



Removes intra & Adds inter



Degree Distributions



Diffusion improves graph learning. In Advances in Neural Information Processing Systems, 2019. https://proceedings.neurips.cc/paper/ 2019/file/23c894276a2c5a16470e6a31f4618d73-Paper.pdf
Jake Topping, et al. "Understanding over-squashing and bottlenecks on graphs via curvature". In ICLR, 2022. URL.

Curvature vs Diffusive Rewiring

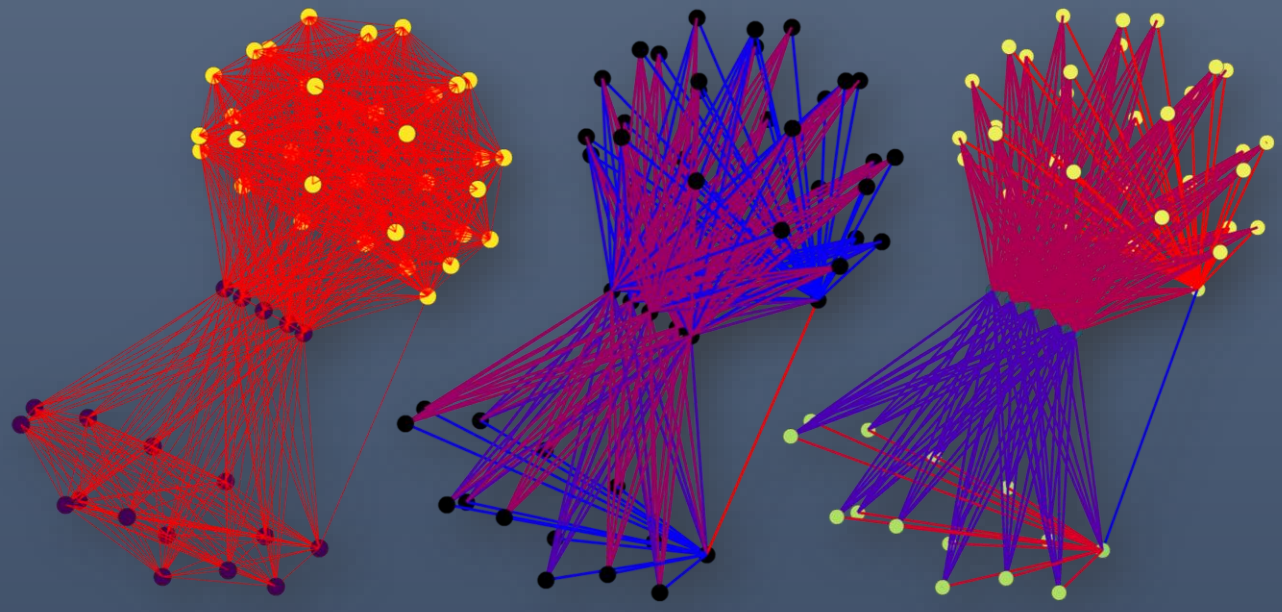
Analysis

Diffusion works better in homophilic graphs [Klicpera et al. 2019] Needs parameters α (or t) and ϵ

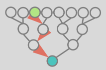
SDRF works better in heterophilic graphs [Topping el al., 2022] Needs parameters τ and C^+

$\mathcal{H}(G)$	Cornell 0.11	Texas 0.06	Wisconsin 0.16	Chameleon 0.25	Squirrel 0.22	Actor 0.24	Cora 0.83	Citeseer 0.71	Pubmed 0.79
None	52.69 ± 0.21	61.19 ± 0.49	54.60 ± 0.86	41.33 ± 0.18	30.32 ± 0.99	23.84 ± 0.43	81.89 ± 0.79	72.31 ± 0.17	78.16 ± 0.23
Undirected	53.20 ± 0.53	63.38 ± 0.87	51.37 ± 1.15	42.02 ± 0.30	35.53 ± 0.78	21.45 ± 0.47	-	-	-
+FA	**58.29 ± 0.49**	**64.82 ± 0.29**	55.48 ± 0.62	42.67 ± 0.17	36.86 ± 0.44	24.14 ± 0.43	81.65 ± 0.18	70.47 ± 0.18	**79.48 ± 0.12**
DIGL (PPR)	58.26 ± 0.50	62.03 ± 0.43	49.53 ± 0.27	42.02 ± 0.13	33.22 ± 0.14	24.77 ± 0.32	**83.21 ± 0.27**	**73.29 ± 0.17**	78.84 ± 0.08
DIGL + Undirected	**59.54 ± 0.64**	63.54 ± 0.38	52.23 ± 0.54	42.68 ± 0.12	32.48 ± 0.23	25.45 ± 0.30	-	-	-
SDRF	54.60 ± 0.39	64.46 ± 0.38	**55.51 ± 0.27**	**42.73 ± 0.15**	**37.05 ± 0.17**	**28.42 ± 0.75**	**82.76 ± 0.23**	**72.58 ± 0.20**	**79.10 ± 0.11**
SDRF + Undirected	57.54 ± 0.34	**70.35 ± 0.60**	**61.55 ± 0.86**	**44.46 ± 0.17**	**37.67 ± 0.23**	**28.35 ± 0.06**	-	-	-

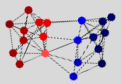
Diffusion improves graph learning. In Advances in Neural Information Processing Systems, 2019.
https://proceedings.neurips.cc/paper/ 2019/file/23c894276a2c5a16470e6a31f4618d73-Paper.pdf
Jake Topping, et al. "Understanding over-squashing and bottlenecks on graphs via curvature". In ICLR, 2022. URL.



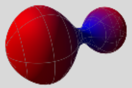
Inductive Graph Rewiring



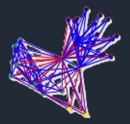
Motivation and Challenges



Introduction to Spectral Theory



Transductive Rewiring



Inductive Rewiring

Graph Fairness

Panel Discussion

The Lovász Bound

Motivation and basic equations

The Lovász bound explains the expressiveness of commute times [Lovász, 1993]

$$\left|\frac{CT(u,v)}{vol(G)} - \left(\frac{1}{d_u} + \frac{1}{d_v}\right)\right| \le \frac{1}{\lambda_2}\frac{2}{d_{min}}$$

Effective Resistance Local resistance

- **Deviation from Local resistance**: The **global** effective resistance should be far from its **local estimation** to be **informative.**
- **Inverse of the bottleneck**: High **spectral gaps** induce uninformative effective resistances. (Link to Cirvature)
- **High probability of getting lost** in (some) large graphs [von Luxburg et al., 2014]

Some facts:

- Effective resistances are also given by the **Laplacian's pseudoinverse or Green's function**
  $R(u,v) = (e_u - e_v)^T L^+ (e_u - e_v), \;\; L^+ = \sum_{i>2}^{n} \lambda_i^{-1} f_i f_i^T$
- Effective resistances are **upper bounded by shortest paths**
  (and they are by far more informative about the role of the Edge (u,v) in the graph since all paths are considered)

László Lovász. Random walks on graphs. Combinatorics, Paul Erdös is eighty, 2(1-46):4, 1993. URL https://web.cs.elte.hu/~lovasz/erdos.pdf. 2, 4

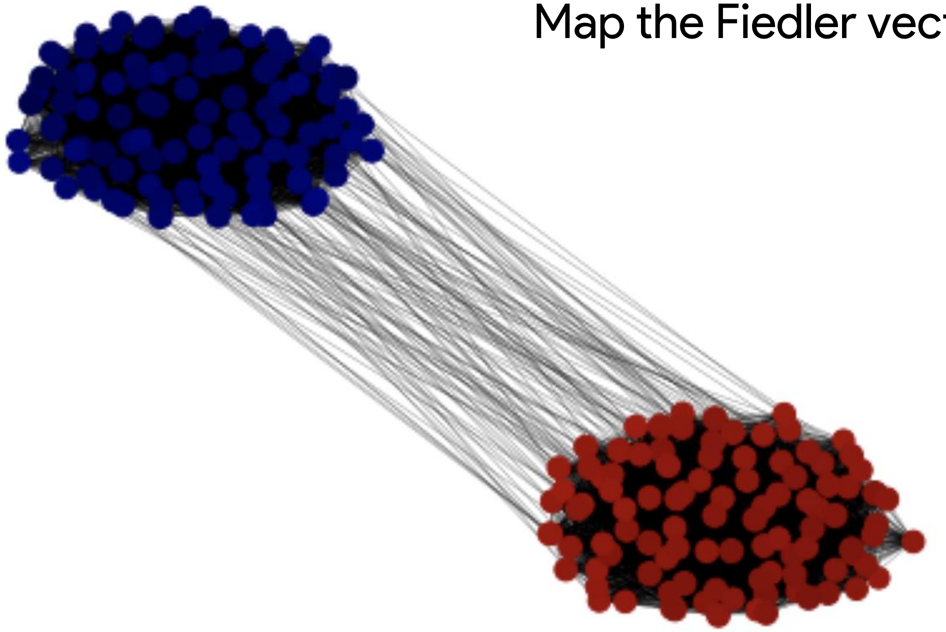
Ulrike von Luxburg, Agnes Radl, and Matthias Hein. Hitting and commute times in large random neighborhood graphs. Journal of Machine Learning Research, 15(52):1751–1798, 2014. URL http://jmlr.org/papers/v15/vonluxburg14a.html. 4, 20

The Lovász Bound

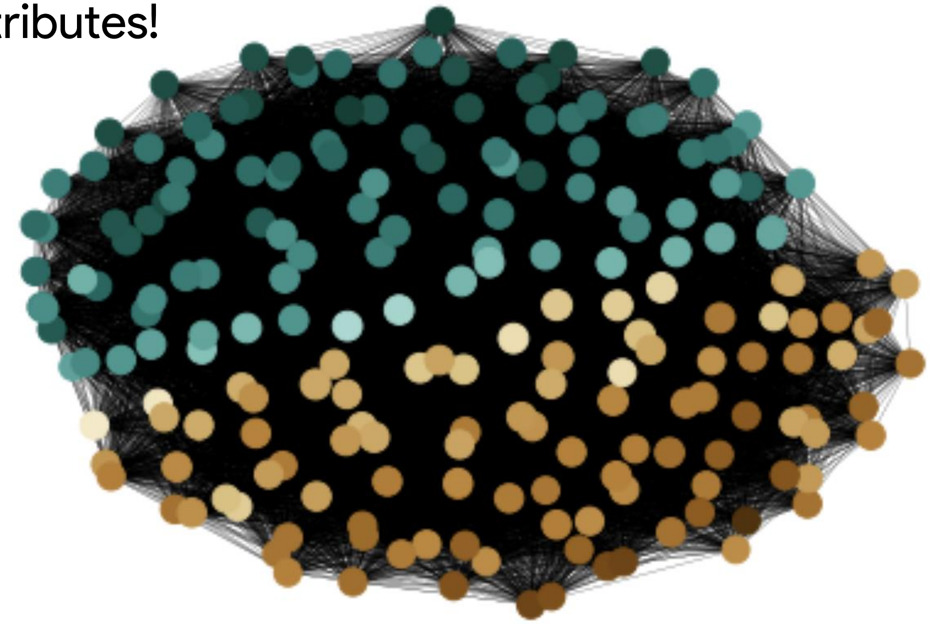
Impact of the bound

Consider two SBMs with small and large gap respectively:

Map the Fiedler vector as node attributes!



`Bottleneck of G is 0.027295784924703657`



`Bottleneck of H is 0.7588701310820082`

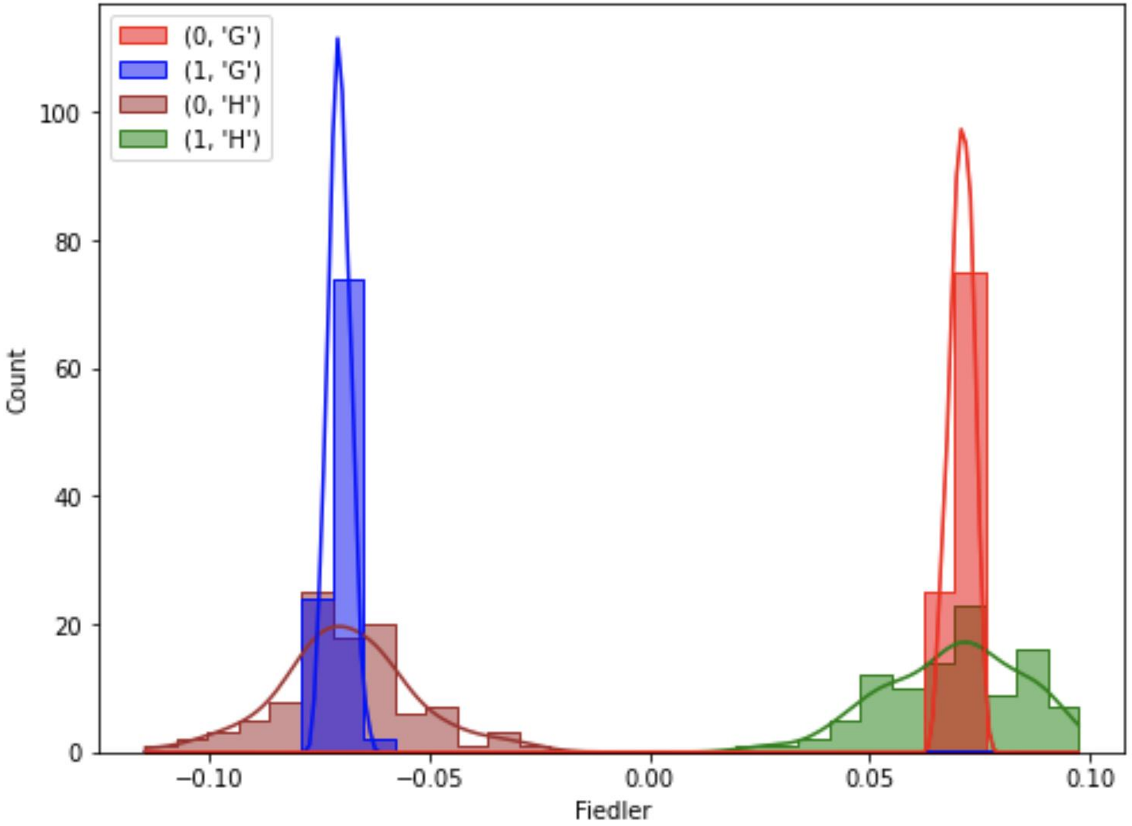
László Lovász. Random walks on graphs. Combinatorics, Paul Erdös is eighty, 2(1-46):4, 1993. URL https://web.cs.elte.hu/~lovasz/erdos.pdf. 2, 4

Ulrike von Luxburg, Agnes Radl, and Matthias Hein. Hitting and commute times in large random neighborhood graphs. Journal of Machine Learning Research, 15(52):1751–1798, 2014. URL http://jmlr.org/papers/v15/vonluxburg14a.html. 4, 20

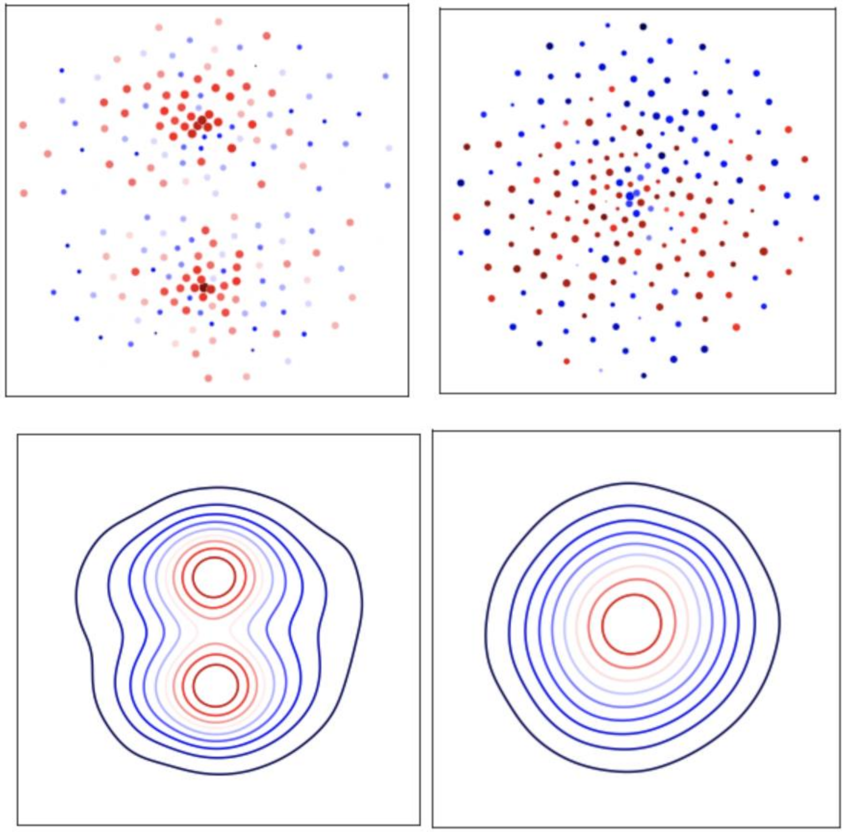
The Lovász Bound

Impact of the bound

The spectral gap (i.e. the Dirichlet energy of the Fiedler vector) controls the variance of f_2 and consequently the scatter in the latent space:



Latent spaces: Nodes and KDEs



Huaijun Qiu and Edwin R. Hancock. Clustering and embedding using commute times. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(11):1873–1890, 2007. doi: 10.1109/TPAMI.2007.1103. URL https://ieeexplore.ieee.org/document/4302755. 6

The Lovász Bound

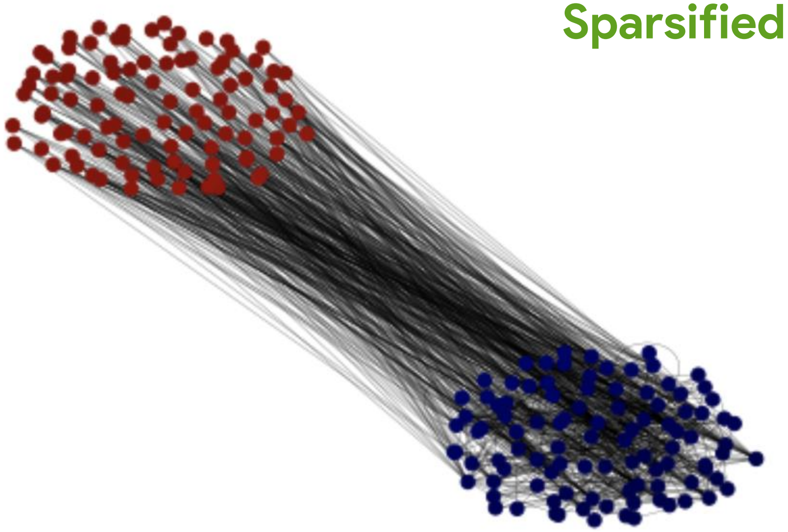
Sparsification

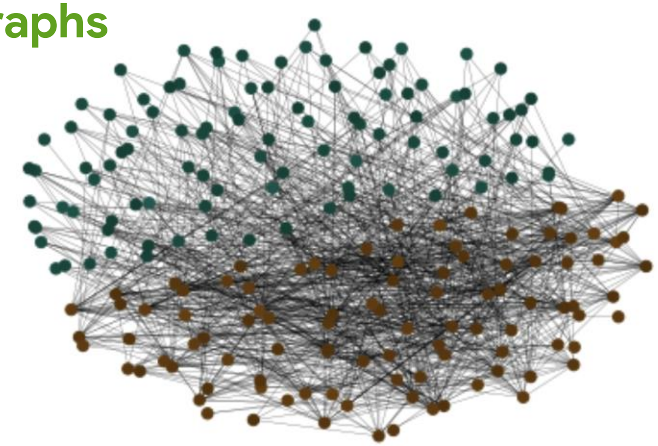
Effective resistances (when informative) R(u,v) reveal the impact of each Edge (u,v) in the topology of the Graph. Therefore, sampling edges with a probability proportional to the effective resistance results in a sparse versión of the graph. [Spielman and Srivastava, 2011]

$\forall \mathbf{x} \in \mathbb{R}^n: (1-\epsilon)\mathbf{x}^T\mathbf{L}_G\mathbf{x} \le \mathbf{x}^T\mathbf{L}_{G'}\mathbf{x} \le (1+\epsilon)\mathbf{x}^T\mathbf{L}_G\mathbf{x}$

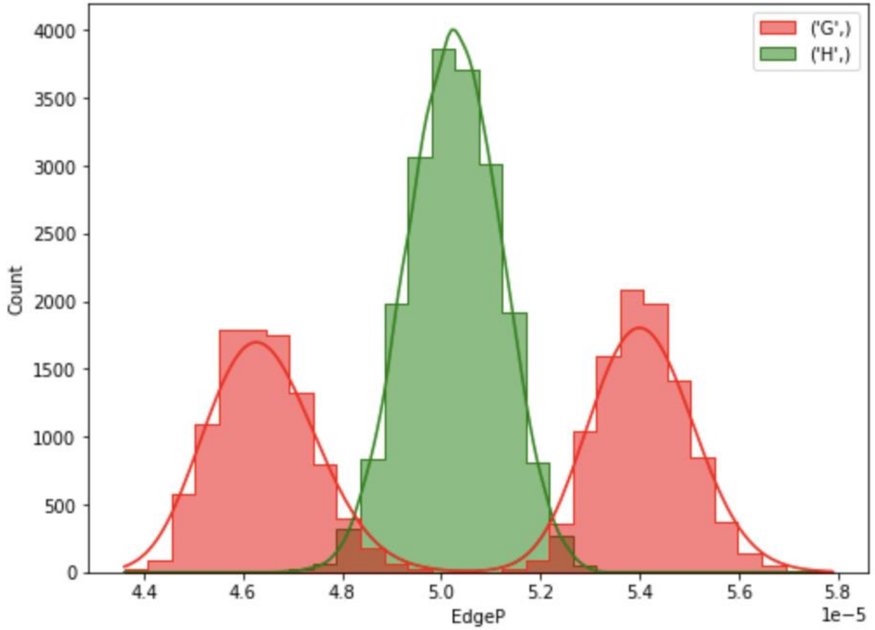
$O\left(\frac{nlogn}{\varepsilon}\right)$ samples suffice to satisfy

Sparsified graphs





KDE of probabilities



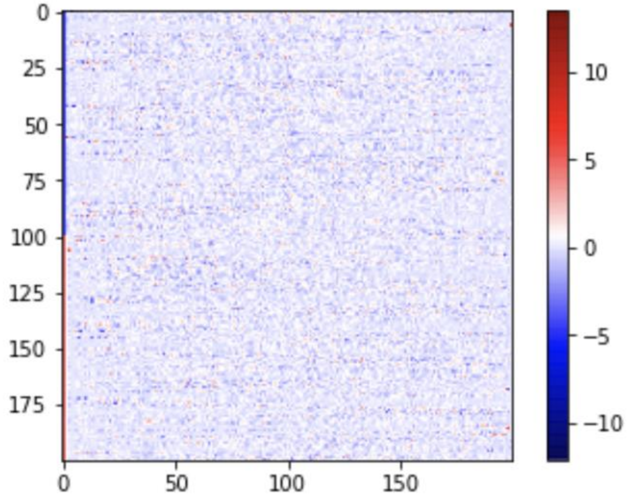
Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. SIAM Journal on Computing, 40(6):1913–1926, 2011. doi: 10.1137/080734029. URL https://doi.org/10.1137/080734029. 5

Commute Times embeddings rely on **down-scaled versions** **of the eigenvectors $\boldsymbol{F}$ and the scale factor is the corresponding eigenvalue.**

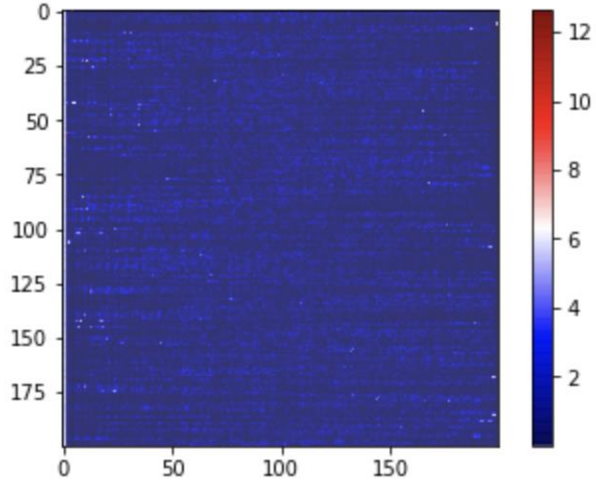
CT Embedding → $CT_{uv} = ||\mathbf{z}_u - \mathbf{z}_v||_2^2$ **comes from** $Z = \sqrt{vol(G)}\Lambda^{-1/2}\boldsymbol{F}^T$

$\mathbf{z}_u = \sqrt{vol(G)}\left(0 \;\; \frac{\mathrm{f}_2(u)}{\sqrt{\lambda_2}} \;\; \frac{\mathrm{f}_3(u)}{\sqrt{\lambda_3}} \;\; \dots \;\; \frac{\mathrm{f}_n(u)}{\sqrt{\lambda_n}}\right)^T$ and consequently Z^T gets the scaled non-trivial eigenvectors

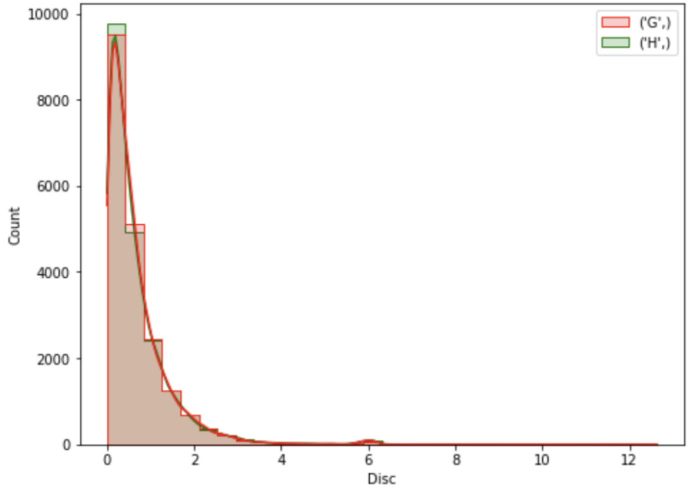
Z^T



$|Z^T - \boldsymbol{F}|$



KDE of $|Z^T - \boldsymbol{F}|$

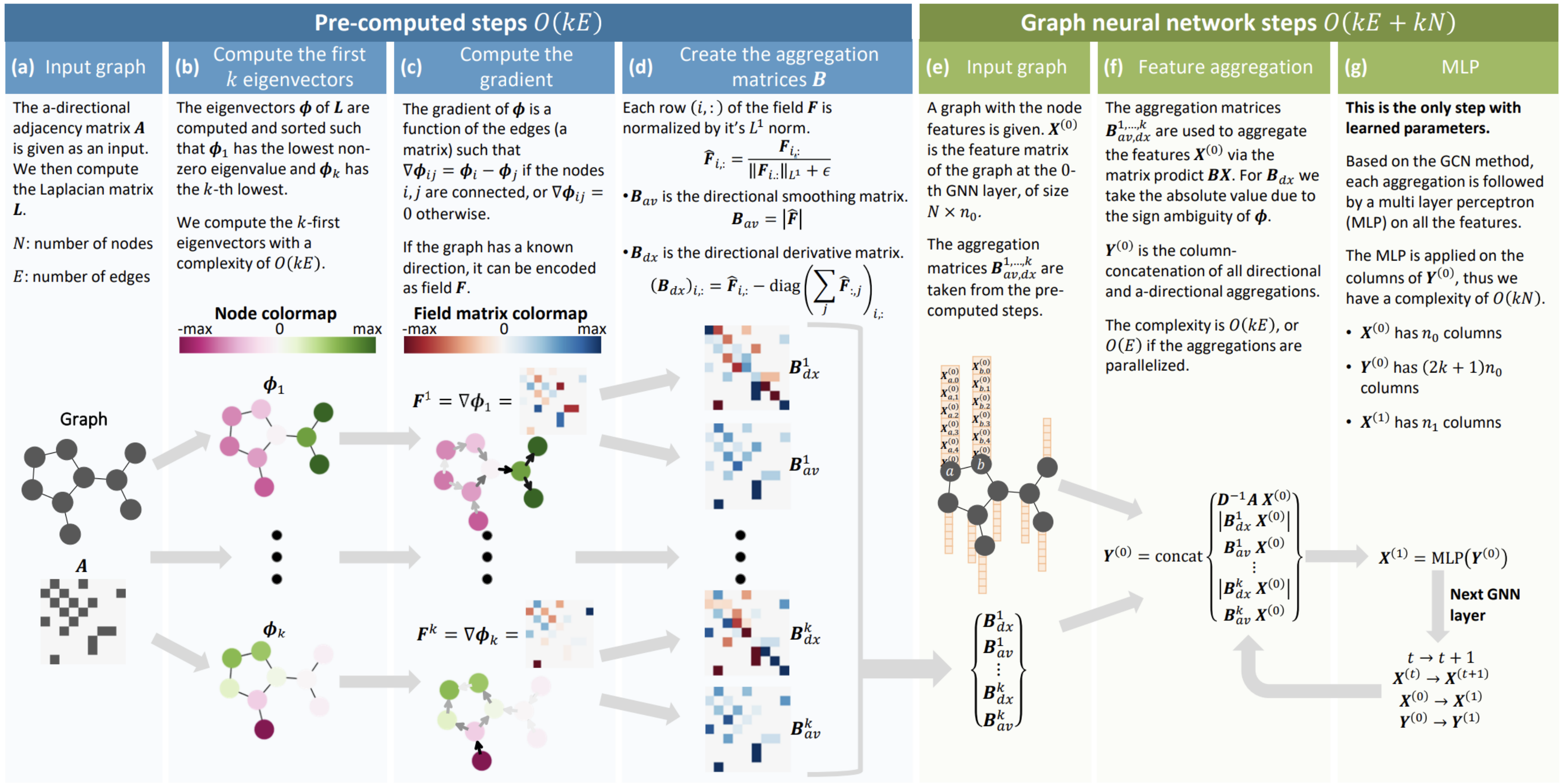


Huaijun Qiu and Edwin R. Hancock. Clustering and embedding using commute times. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(11):1873–1890, 2007. doi: 10.1109/TPAMI.2007.1103. URL https://ieeexplore.ieee.org/document/4302755. 6

The Lovász Bound

Link with Directional Graph Networks

Directional Graph Networks



Dominique Beaini, Saro Passaro, Vincent Létourneau, Will Hamilton, Gabriele Corso, and Pietro Liò. Directional graph networks. In International Conference on Machine Learning (ICML), pages 748–758. PMLR, 2021. http://proceedings.mlr.press/v139/beaini21a/beaini21a.pdf

CT-Layer

Why commute times for rewiring? Quick Recap!

$\boldsymbol{CT_{uv} \propto R_{uv} = H_{uv} + H_{vu}}$

→ Expected time to from u to v and come back to u

CT Embedding → $\boldsymbol{CT_{uv} = ||\mathbf{z}_u - \mathbf{z}_v||_2^2}$

→ Node embedding which pairwise Euclidean distance is CT_{uv}

Direct relationship with

- Eigenvectors

$$R_{uv} = \frac{CT_{uv}}{vol(G)} = \sum_{i=2}^{n} \frac{1}{\lambda_i}\left(\mathbf{f}_i(u) - \mathbf{f}_i(v)\right)^2$$

- Dirichlet Energies

$$\mathcal{E}_G(\mathbf{x}) = \mathbf{x}^T \mathbf{L}_G \mathbf{x} = \sum_{(u,v)\in E} (\mathrm{x_u} - \mathrm{x_v})^2 = Tr[\mathbf{X}^{\mathrm{T}}\mathbf{L}_G\mathbf{X}]$$

- Expanders and Sparsifiers

$$\forall \mathbf{x} \in \mathbb{R}^{\mathrm{n}}: (1-\epsilon)\mathbf{x}^{\mathrm{T}}\mathbf{L}_G\mathbf{x} \le \mathbf{x}^{\mathrm{T}}\mathbf{L}_{G'}\mathbf{x} \le (1+\epsilon)\mathbf{x}^{\mathrm{T}}\mathbf{L}_G\mathbf{x}$$

- Cheeger Constant

$$\boldsymbol{h_G} = \min_{S\subseteq V} h_S\,, h_S = \frac{|\{(u,v): u \in S, v \in \bar{S}\}|}{\min(\mathrm{vol}(S), \mathrm{vol}(\bar{S}))}$$

- Curvature

$$p_{\mathrm{u}} := 1 - \tfrac{1}{2}\Sigma_{\mathrm{v}\in \mathrm{N(u)}} R_{\mathrm{uv}} \qquad \kappa_{\mathrm{uv}} := \frac{2(p_{\mathrm{u}} + p_{\mathrm{v}})}{R_{\mathrm{uv}}}$$

Spectral computation

$$CT_{uv} = \sum_{i=2}^{n} \frac{1}{\lambda_i}\left(\mathbf{f}_i(u) - \mathbf{f}_i(v)\right)^2$$

$$\mathbf{Z} = \sqrt{vol(G)}\boldsymbol{\Lambda}^{-1/2}\mathbf{F}^T \text{ given } \mathbf{L} = \mathbf{F}\boldsymbol{\Lambda}\mathbf{F}^T$$

or

$$R_{uv} = (\mathbf{e}_u - \mathbf{e}_v)\mathbf{L}^{+}(\mathbf{e}_u - \mathbf{e}_v)$$

$$\mathbf{L}^{+} = \sum_{i=2}^{n} \frac{1}{\lambda_i}\mathbf{f}_i\mathbf{f}_i^{\mathrm{T}}$$

Optimization problem

$$\mathbf{Z} = \arg\min_{s.t.\ \mathbf{Z}^T\mathbf{Z}=\mathbb{I}} \frac{Tr[\mathbf{Z}^T\mathbf{L}_G\mathbf{Z}]}{Tr[\mathbf{Z}^T\mathbf{D}_G\mathbf{Z}]}$$

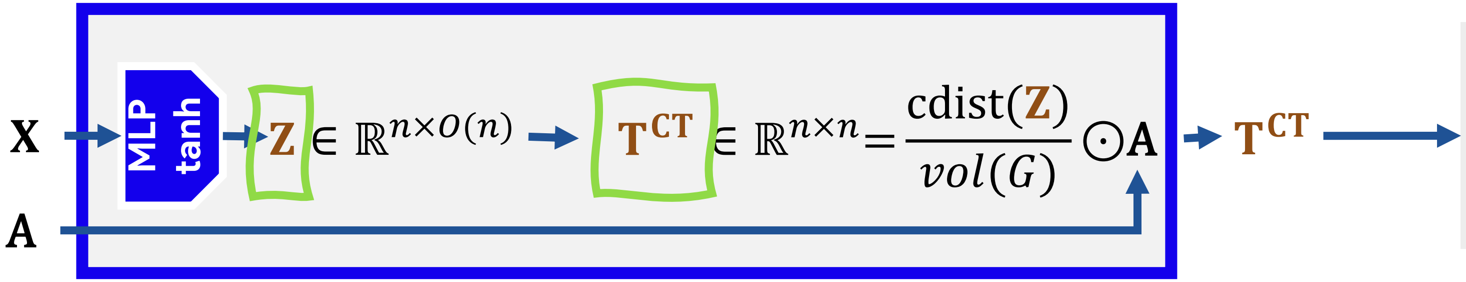
$$CT_{uv} = \|\mathbf{z}_u - \mathbf{z}_v\|_2^2$$

e l l i s ALICANTE

CT-Layer

From Spectral CT to CT-Layer

$$\mathbf{Z} = \sqrt{vol(G)}\mathbf{\Lambda}^{-1/2}\mathbf{F}^T \rightarrow \mathbf{Z} = \arg\min_{s.t.\,\mathbf{Z}^T\mathbf{Z}=\mathbb{I}} \frac{Tr[\mathbf{Z}^T\mathbf{L}_G\mathbf{Z}]}{Tr[\mathbf{Z}^T\mathbf{D}_G\mathbf{Z}]} \longrightarrow \boxed{L_{CT} = \frac{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{L}\mathbf{Z}]}{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{D}\mathbf{Z}]} + \left\| \frac{\mathbf{Z}^{\mathbf{T}}\mathbf{Z}}{\|\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\|_F} - \mathbf{I}_N \right\|_F}$$



$$L_{CT} = \frac{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{L}\mathbf{Z}]}{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{D}\mathbf{Z}]} + \left\| \frac{\mathbf{Z}^{\mathbf{T}}\mathbf{Z}}{\|\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\|_F} - \mathbf{I}_N \right\|_F$$

Use Effective Resistances matrix (commute times) as the input adjacency matrix for new layers

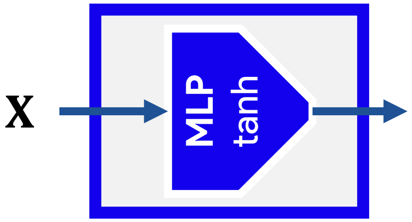
CT-layer can be added as the first layer or as the # desired layer

Arnaiz-Rodriguez, A., Begga, A., Escolano, F. & Oliver, N. "**DiffWire: Inductive Graph Rewiring via the Lovász Bound**". Proceedings of the First Learning on Graphs Conference (LoG 2022), PMLR 198, Virtual Event, December, 2022

CT-Layer

From Spectral CT to CT-Layer

$$L_{CT} = \frac{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{L}\mathbf{Z}]}{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{D}\mathbf{Z}]} + \left\| \frac{\mathbf{Z}^{\mathbf{T}}\mathbf{Z}}{\|\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\|_F} - \mathbf{I}_N \right\|_F$$



Simplified version of the previous slide

$\mathbf{Z} \in \mathbb{R}^{n \times O(n)}$

CT as difusion **(A)**
(DiffWire – actual rewiring – previous slide)

CT as edge feat
(Affinity-Aware approach)

CTE as node feat
(preliminary work in both papers)

Arnaiz-Rodriguez, A., Begga, A., Escolano, F. & Oliver, N. "**DiffWire: Inductive Graph Rewiring via the Lovász Bound**". Proceedings of the First Learning on Graphs Conference (LoG 2022), PMLR 198, Virtual Event, December, 2022

CT-Layer

From Spectral CT to CT-Layer

$\mathbf{X}$

MLP tanh

$\mathbf{Z} \in \mathbb{R}^{n\times O(n)} \rightarrow \mathbf{T}^{\mathbf{CT}} \in \mathbb{R}^{n\times n} = \dfrac{\mathrm{cdist}(\mathbf{Z})}{vol(G)} \odot \mathbf{A} \rightarrow \mathbf{T}^{\mathbf{CT}}$

$\mathbf{A}$

$$L_{CT} = \frac{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{L}\mathbf{Z}]}{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{D}\mathbf{Z}]} + \left\| \frac{\mathbf{Z}^{\mathbf{T}}\mathbf{Z}}{\|\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\|_F} - \mathbf{I}_N \right\|_F$$

```
# Pooling for CT embedding
num_of_centers1 =  k_centers # k1 #order of number of nodes
self.pool1 = Linear(hidden_channels, num_of_centers1)

# CT REWIRING
s  = self.pool1(x)
```

https://github.com/AdrianArnaiz/DiffWire/blob/main/layers/CT_layer.py

Arnaiz-Rodriguez, A., Begga, A., Escolano, F. & Oliver, N. "**DiffWire: Inductive Graph Rewiring via the Lovász Bound**". Proceedings of the First Learning on Graphs Conference (LoG 2022), PMLR 198, Virtual Event, December, 2022

CT-Layer

From Spectral CT to CT-Layer

X → MLP tanh → $\mathbf{Z} \in \mathbb{R}^{n \times O(n)}$ → $\mathbf{T}^{\mathbf{CT}} \in \mathbb{R}^{n \times n} = \dfrac{\mathrm{cdist}(\mathbf{Z})}{vol(G)} \odot \mathbf{A}$ → $\mathbf{T}^{\mathbf{CT}}$

A

$$L_{CT} = \frac{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{L}\mathbf{Z}]}{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{D}\mathbf{Z}]} + \left\| \frac{\mathbf{Z}^{\mathbf{T}}\mathbf{Z}}{\|\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\|_F} - \mathbf{I}_N \right\|_F$$

```python
# Calculate CT_dist as cdist(s,s)/vol(G)
CT_dist = torch.cdist(s,s) # [b, N, k], [b, N, k]-> [20,N,N]
## Calculate degree d_flat and degree matrix d
d_flat = torch.einsum('ijk->ij', adj) # torch.Size([b, N])
d = _rank3_diag(d_flat)+EPS  # d torch.Size([b, N, N])
## Calculate Vol (volumes): one per graph
vol = _rank3_trace(d) # torch.Size([b])
## Calculate out_adj as CT_dist*(N-1)/vol(G)
N = adj.size(1)
CT_dist = (CT_dist) / vol.unsqueeze(1).unsqueeze(1)
## Mask with adjacency
adj = CT_dist*adj
```

https://github.com/AdrianArnaiz/DiffWire/blob/main/layers/CT_layer.py

Arnaiz-Rodriguez, A., Begga, A., Escolano, F. & Oliver, N. "**DiffWire: Inductive Graph Rewiring via the Lovász Bound**". Proceedings of the First Learning on Graphs Conference (LoG 2022), PMLR 198, Virtual Event, December, 2022

CT-Layer

From Spectral CT to CT-Layer

X → MLP tanh → $\mathbf{Z} \in \mathbb{R}^{n\times O(n)}$ → $\mathbf{T}^{\mathbf{CT}} \in \mathbb{R}^{n\times n} = \dfrac{\mathrm{cdist}(\mathbf{Z})}{vol(G)} \odot \mathbf{A}$ → $\mathbf{T}^{\mathbf{CT}}$

A

$$L_{CT} = \frac{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{L}\mathbf{Z}]}{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{D}\mathbf{Z}]} + \left\| \frac{\mathbf{Z}^{\mathbf{T}}\mathbf{Z}}{\|\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\|_F} - \mathbf{I}_N \right\|_F$$

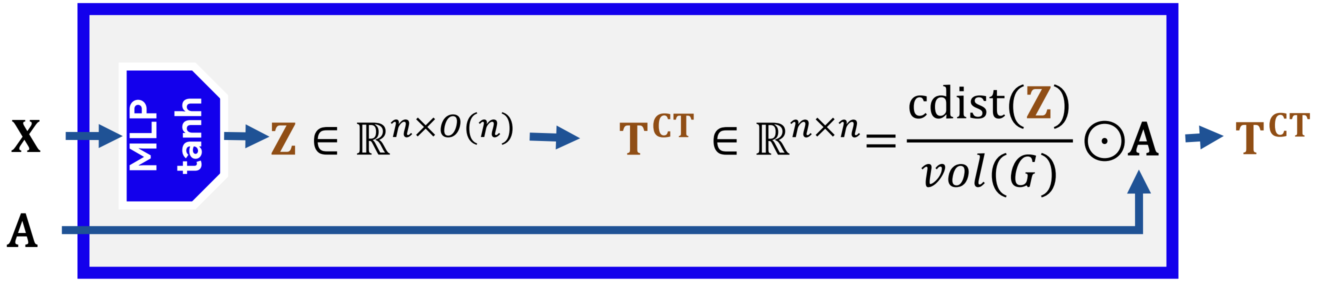
```python
# Losses
## Loss cut
### Calculate Laplacian L = D - A
L = d - adj
### Calculate loss num as Tr[S.T*L*S]
num = torch.matmul(torch.matmul(s.transpose(1, 2), L), s)
CT_num = _rank3_trace(num) # mincut_num torch.Size([b]) one sum over each graph
### Calculate CT_den
CT_den = _rank3_trace(
    torch.matmul(torch.matmul(s.transpose(1, 2), d ), s))+EPS
### Calculate loss cut
CT_loss = CT_num / CT_den
CT_loss = torch.mean(CT_loss) # Mean over batch!
```

https://github.com/AdrianArnaiz/DiffWire/blob/main/layers/CT_layer.py

Arnaiz-Rodriguez, A., Begga, A., Escolano, F. & Oliver, N. "**DiffWire: Inductive Graph Rewiring via the Lovász Bound**". Proceedings of the First Learning on Graphs Conference (LoG 2022), PMLR 198, Virtual Event, December, 2022

CT-Layer

From Spectral CT to CT-Layer



$$L_{CT} = \frac{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{L}\mathbf{Z}]}{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{D}\mathbf{Z}]} + \left\| \frac{\mathbf{Z}^{\mathbf{T}}\mathbf{Z}}{\|\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\|_F} - \mathbf{I}_N \right\|_F$$

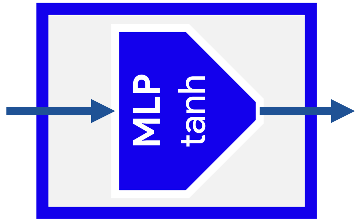
```
## Loss Orthogonality regularization.
ss = torch.matmul(s.transpose(1, 2), s)  #[b, k, N]*[b, N, k]-> [b, k, k]

i_s = torch.eye(k).type_as(ss) # [k, k]
ortho_loss = torch.norm(
    ss / torch.norm(ss, dim=(-1, -2), keepdim=True) -
    i_s)
ortho_loss = torch.mean(ortho_loss) # Mean over batch!
```

https://github.com/AdrianArnaiz/DiffWire/blob/main/layers/CT_layer.py

Arnaiz-Rodriguez, A., Begga, A., Escolano, F. & Oliver, N. "**DiffWire: Inductive Graph Rewiring via the Lovász Bound**". Proceedings of the First Learning on Graphs Conference (LoG 2022), PMLR 198, Virtual Event, December, 2022

CT-Layer

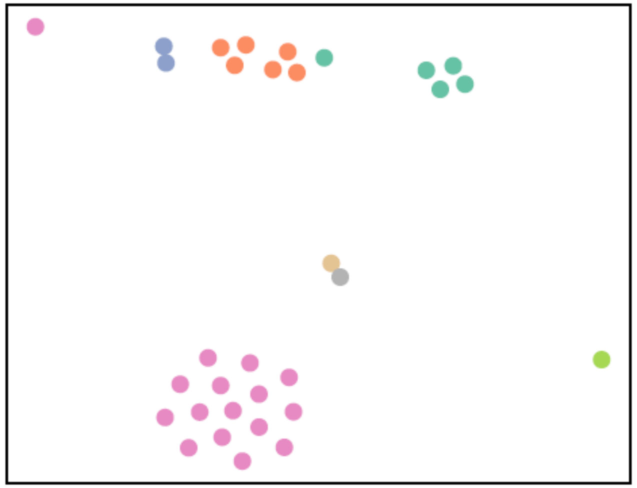
From Spectral CT to CT-Layer

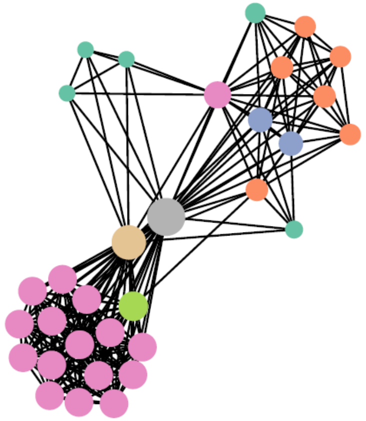
X 

$\mathbf{Z} \in \mathbb{R}^{n \times O(n)}$

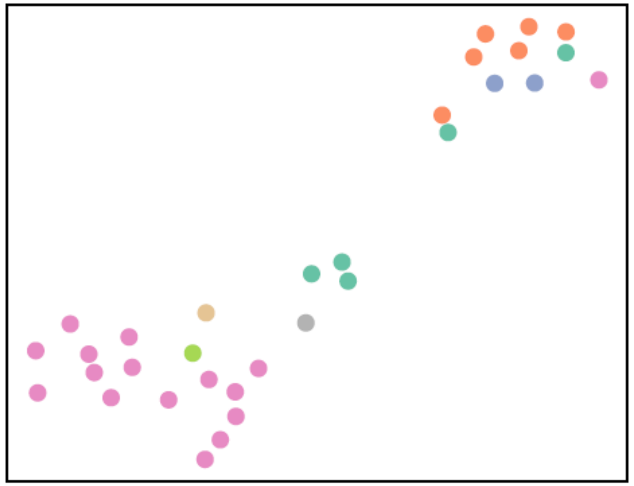
$$L_{CT} = \frac{Tr[\mathbf{Z}^{\mathrm{T}}\mathbf{L}\mathbf{Z}]}{Tr[\mathbf{Z}^{\mathrm{T}}\mathbf{D}\mathbf{Z}]} + \left\| \frac{\mathbf{Z}^{\mathrm{T}}\mathbf{Z}}{\|\mathbf{Z}^{\mathrm{T}}\mathbf{Z}\|_F} - \mathbf{I}_N \right\|_F$$

CT-Layer CTE





Spectral CTE



Arnaiz-Rodriguez, A., Begga, A., Escolano, F. & Oliver, N. “**DiffWire: Inductive Graph Rewiring via the Lovász Bound**”. Proceedings of the First Learning on Graphs Conference (LoG 2022), PMLR 198, Virtual Event, December, 2022

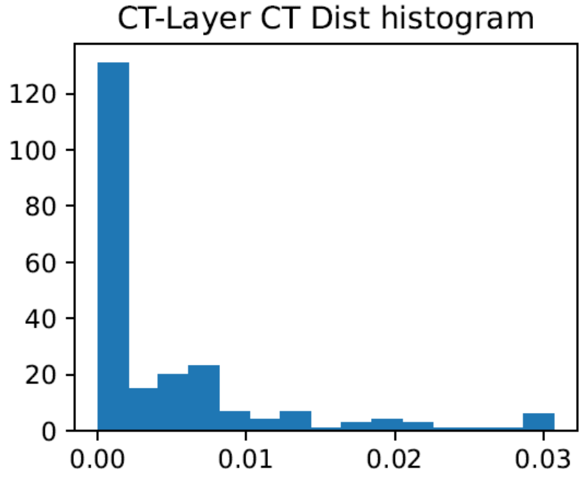
CT-Layer

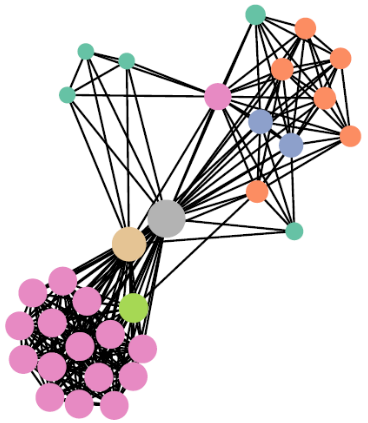
From Spectral CT to CT-Layer

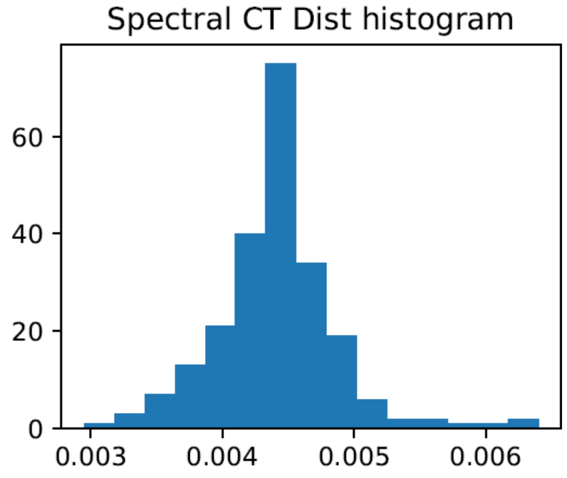
$$\mathbf{X} \rightarrow \text{MLP tanh} \rightarrow \mathbf{Z} \in \mathbb{R}^{n\times O(n)} \rightarrow \frac{\text{cdist}(\mathbf{Z})}{vol(G)} \odot \mathbf{A} \rightarrow \mathbf{T}^{\mathbf{CT}} \in \mathbb{R}^{n\times n}$$

$$L_{CT} = \frac{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{L}\mathbf{Z}]}{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{D}\mathbf{Z}]} + \left\| \frac{\mathbf{Z}^{\mathbf{T}}\mathbf{Z}}{\|\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\|_F} - \mathbf{I}_N \right\|_F$$

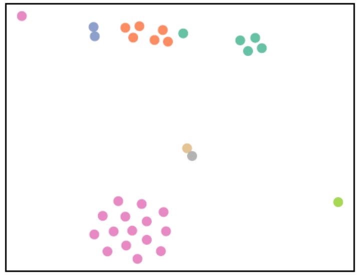
New Learned Adjacency $\mathbf{T}^{\mathbf{CT}}$

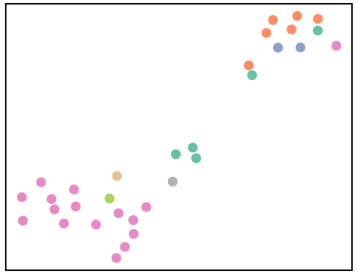






Learned PE $\mathbf{Z}$





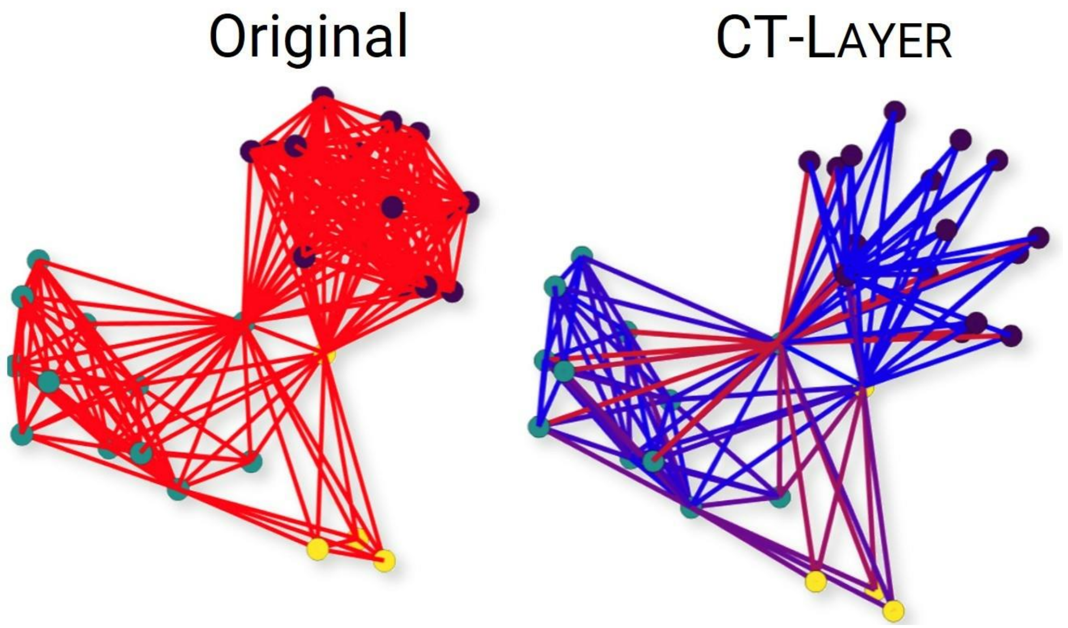
Arnaiz-Rodriguez, A., Begga, A., Escolano, F. & Oliver, N. "**DiffWire: Inductive Graph Rewiring via the Lovász Bound**". Proceedings of the First Learning on Graphs Conference (LoG 2022), PMLR 198, Virtual Event, December, 2022

CT-Layer

From Spectral CT to CT-Layer

$$\mathbf{X} \rightarrow \boxed{\text{MLP tanh} \rightarrow \mathbf{Z} \in \mathbb{R}^{n\times O(n)} \rightarrow \frac{\text{cdist}(\mathbf{Z})}{vol(G)} \odot \mathbf{A}} \rightarrow \mathbf{T}^{\mathbf{CT}} \in \mathbb{R}^{n\times n}$$

$$\boxed{L_{CT} = \frac{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{L}\mathbf{Z}]}{Tr[\mathbf{Z}^{\mathbf{T}}\mathbf{D}\mathbf{Z}]} + \left\| \frac{\mathbf{Z}^{\mathbf{T}}\mathbf{Z}}{\|\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\|_F} - \mathbf{I}_N \right\|_F}$$



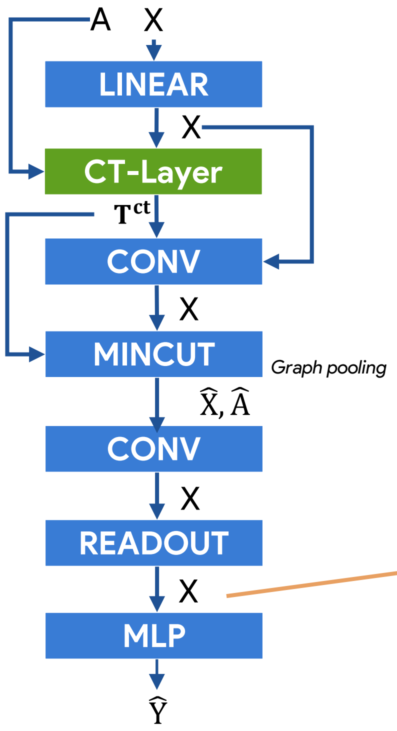
Graph from COLLAB Test Set

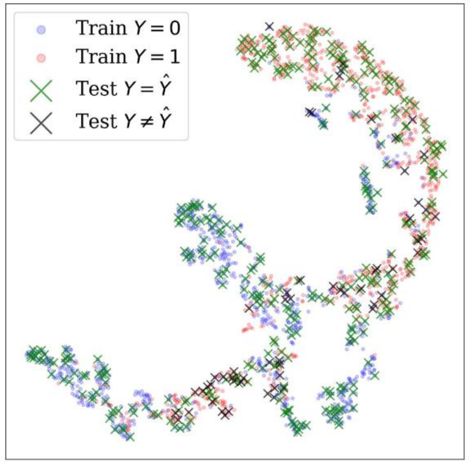
Arnaiz-Rodriguez, A., Begga, A., Escolano, F. & Oliver, N. "**DiffWire: Inductive Graph Rewiring via the Lovász Bound**". Proceedings of the First Learning on Graphs Conference (LoG 2022), PMLR 198, Virtual Event, December, 2022

CT-Layer

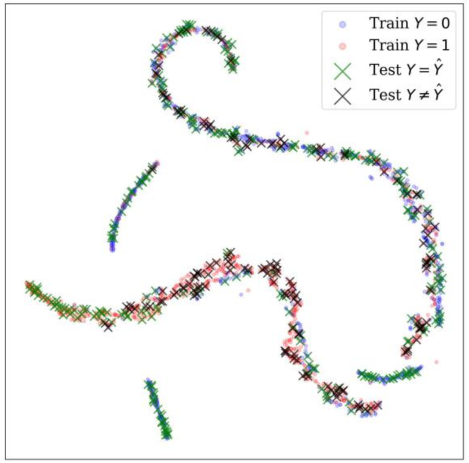
Experiments on Graph Classification

	MinCutPool	k-NN	DIGL	SDRF	CT-LAYER
REDDIT-B*	66.53±4.4	64.40±3.8	76.02±4.3	65.3±7.7	**78.45**±4.5
IMDB-B*	60.75±7.0	55.20±4.3	59.35±7.7	59.2±6.9	**69.84**±4.6
COLLAB*	58.00±6.2	58.33±11	57.51±5.9	56.60±10	**69.87**±2.4
MUTAG	84.21±6.3	**87.58**±4.1	85.00±5.6	82.4±6.8	**87.58**±4.4
PROTEINS	74.84±2.3	**76.76**±2.5	74.49±2.8	74.4±2.7	**75.38**±2.9
SBM*	53.00±9.9	50.00±0.0	56.93±12	54.1±7.1	81.40±11
Erdös-Rényi*	81.86±6.2	63.40±3.9	**81.93**±6.3	73.6±9.1	79.06±9.8

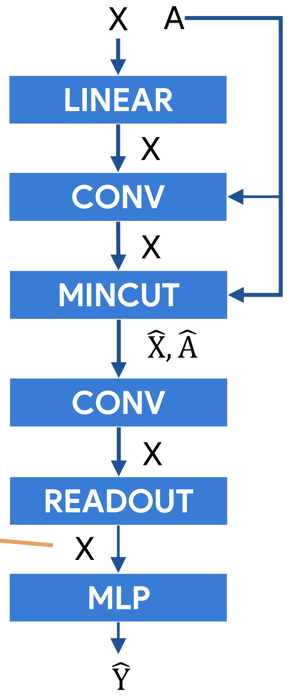




(a) CT-LAYER



(b) MinCut



EXPRESSIVENESS

More sparse Graph Readouts → greater ability to detect differences between graphs

DE or PE provides strictly more expressive power than 1-WL test [Li, P. et al. 2020] [Velingker, A. et al. 2022]

Arnaiz-Rodriguez, A., et al. "DiffWire: Inductive Graph Rewiring via the Lovász Bound". In Learning on Graphs Conference (LoG), 2022.
Li, P., et al. "Distance encoding: Design provably more powerful neural networks for graph representation learning". In NeurIPS, 2020.
Velingker, A., et al. "Affinity-Aware Graph Networks." arXiv preprint arXiv:2206.11941, 2022.

CT-Layer

Implications in Cheeger constant

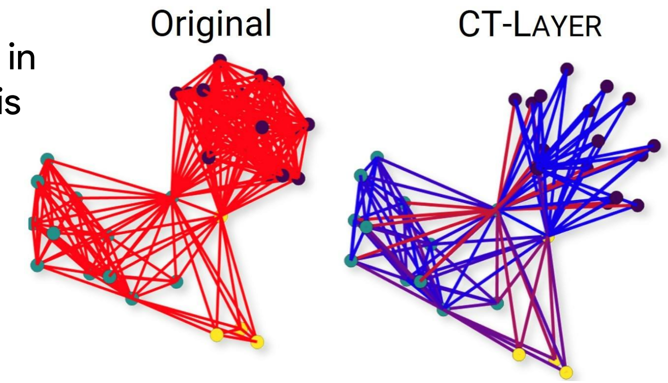
$R_{\mathrm{uv}} \propto CT_{\mathrm{uv}} = \mathbf{T}^{\mathbf{CT}}_{uv}$

CT prioritizes edges in the bottleneck while it sparsifies the communities

$$\boldsymbol{h_G} = \min_{S \subseteq V} h_S\,, \qquad h_S = \frac{\overbrace{|\{(u,v): u \in S, v \in \bar{S}\}|}^{\text{\# edges in the bottleneck}}}{\underbrace{\min(\,\mathrm{vol}(S), \mathrm{vol}(\bar{S}))}_{\text{Volume of the community}}}$$

→ Giving priority to the edges in the bottleneck maintains this

→ Community sparsification minimizes this



In the rewired graph G': bottleneck is wider → resistances are lower in G'

[Alev. et al., 2018]

$$R_{diam} \coloneqq \max_{u,v} \boldsymbol{R}_{\mathbf{uv}}$$

→ Edges in the bottleneck

$$h_G \leq \frac{\alpha^{\epsilon}}{\sqrt{\boldsymbol{R_{diam}} \cdot \epsilon}} \mathrm{vol}(S)^{\epsilon - \frac{1}{2}}$$

→ Prioritizing edges in the bottleneck maintains upper bound (at least)

Rewiring using CT dist

Community sparsification

Preserve bottleneck

Preserve R_{diam} in G'

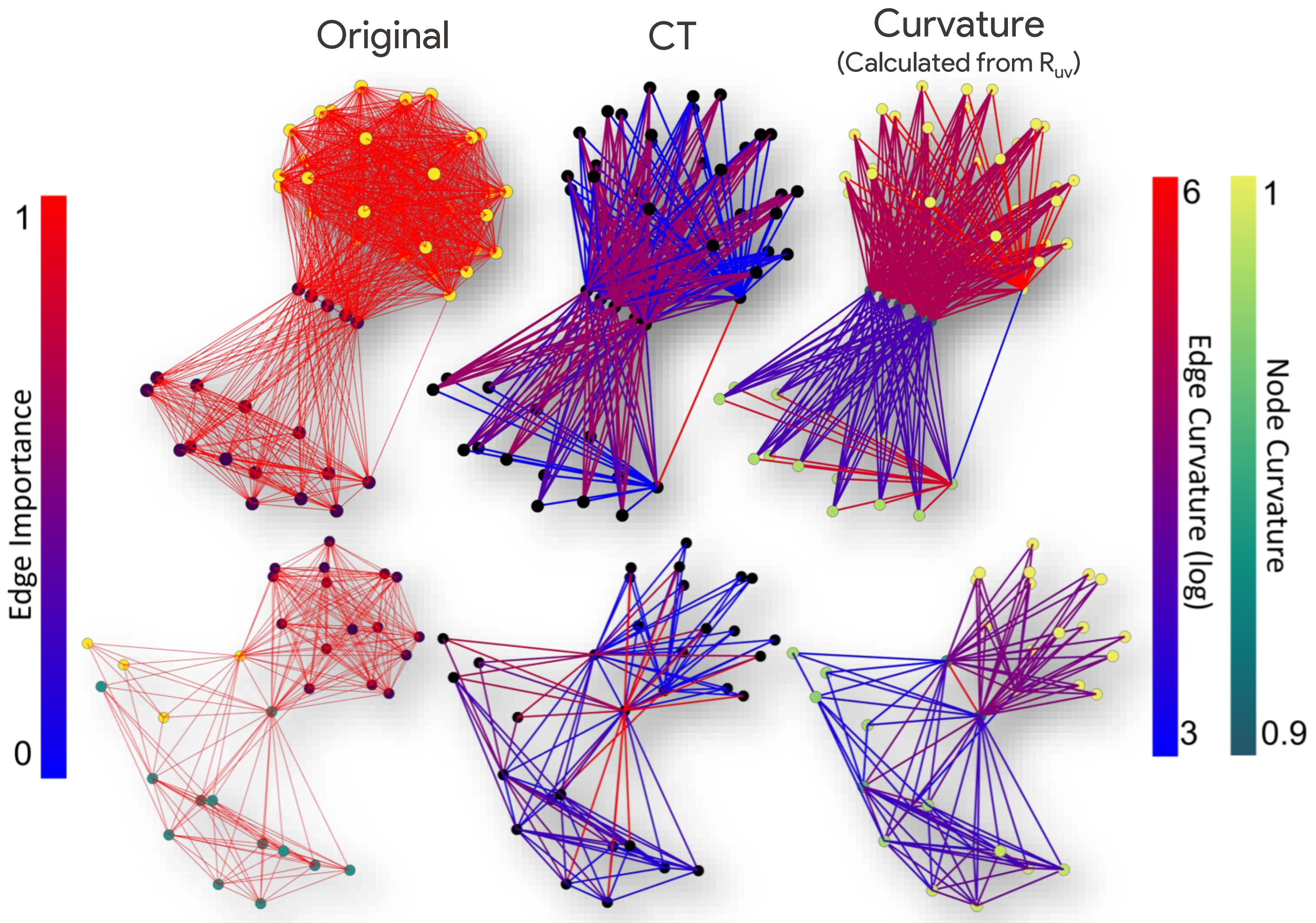
No reduction of h_G upper bound

Quirurgical?

Alev, VL., et al. "Graph Clustering using Effective Resistance". In ITCS, 2018.

CT-Layer

Relationship with Curvature



$$\mathbf{X} \rightarrow \text{MLP} \rightarrow \mathbf{Z} \rightarrow \frac{\text{cdist}(\mathbf{Z})}{vol(G)} \odot \mathbf{A} \rightarrow \mathbf{T}^{\mathbf{CT}}$$

$$R_{uv} = \mathbf{T}^{\mathbf{CT}}_{uv}$$

[Devriendt. et al., 2022]

Node Curvature

$$p_u := 1 - \frac{1}{2}\sum_{v \in N(u)} R_{uv}$$

Edge Curvature

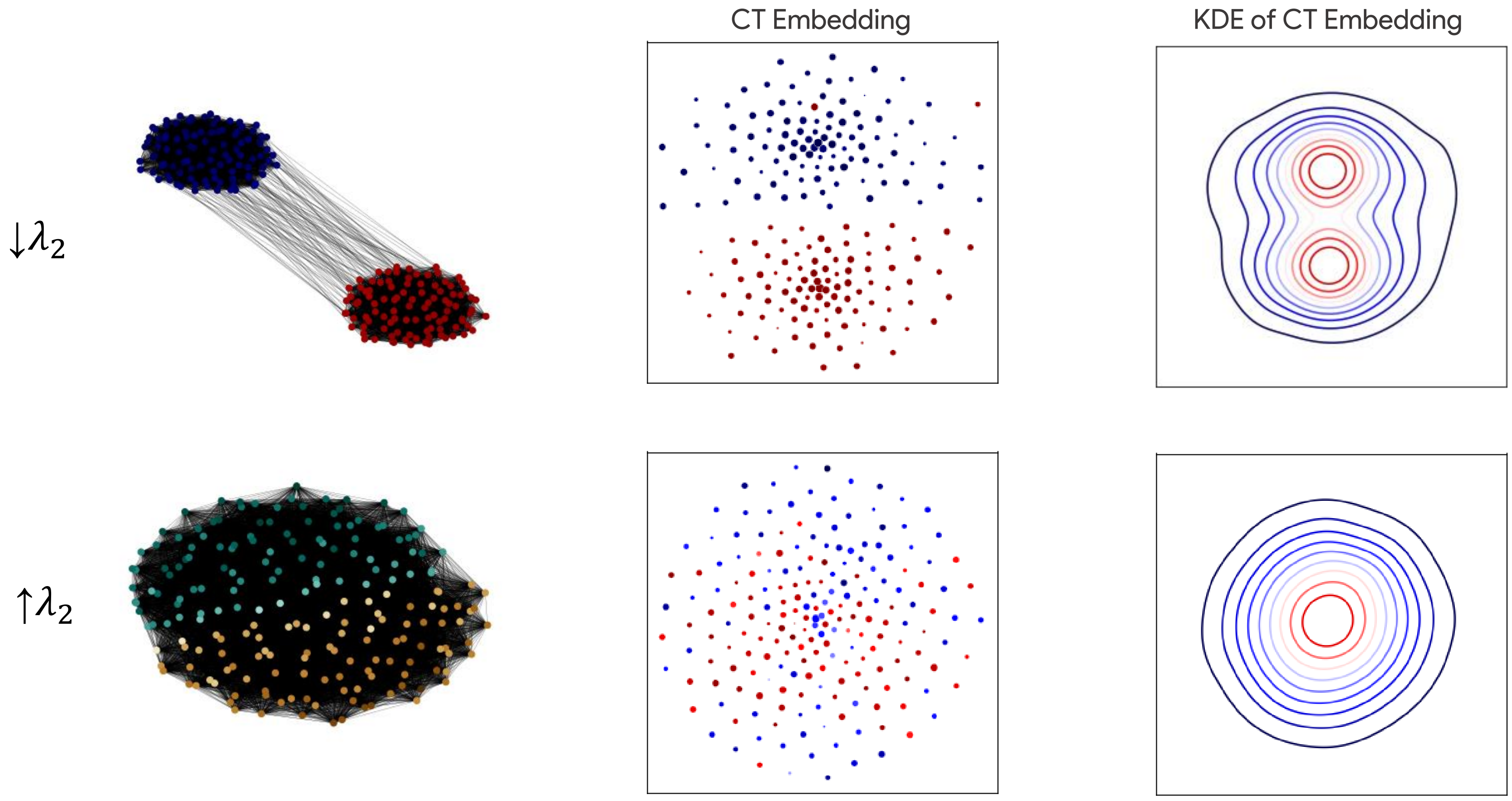
$$\kappa_{uv} := \frac{2(p_u + p_v)}{R_{uv}}$$

CT-Layer as differentiable curvature

Devriendt, K. and Lambiotte, R. "Discrete curvature on graphs from the effective resistance". Journal of Physics: Complexity, 2022.
Topping, J., et al. "Understanding over-squashing and bottlenecks on graphs via curvature". In ICLR, 2022.

CT-Layer

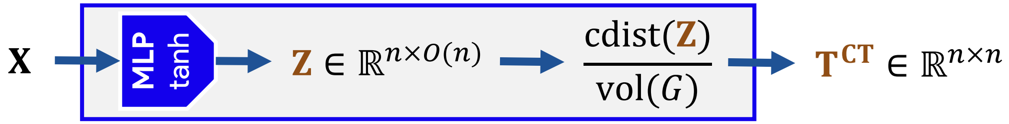
Relationship with Curvature



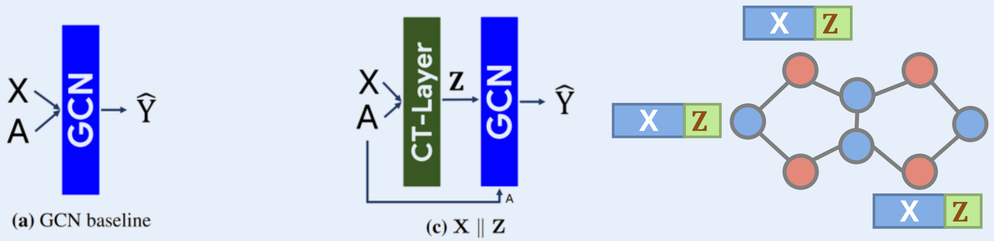
Devriendt, K. and Lambiotte, R. "Discrete curvature on graphs from the effective resistance". Journal of Physics: Complexity, 2022.
Topping, J., et al. "Understanding over-squashing and bottlenecks on graphs via curvature". In ICLR, 2022.

CT-Layer

Node Classification. CT-Diffusions vs CT as Positional Encoding.



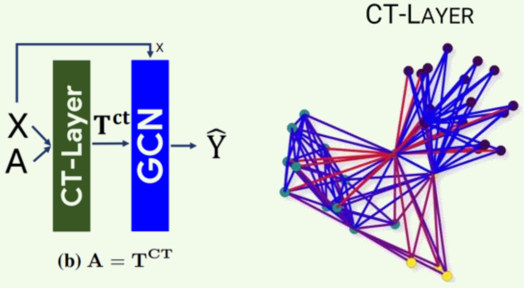
- **CTE** as structural feature (PE) reinforces performance in **homophily** tasks



(a) GCN baseline

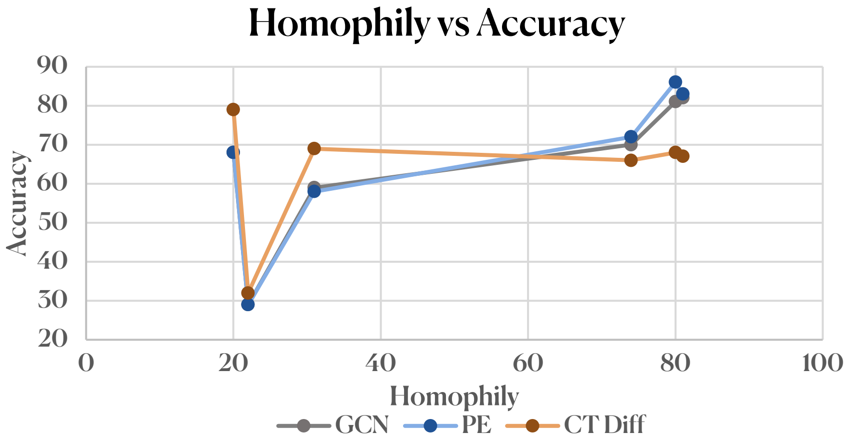
(c) X || Z

- **CT Distance** for diffusion helps in **heterophilic** tasks



(b) $\mathbf{A} = \mathbf{T}^{\mathbf{CT}}$

Dataset	GCN (baseline)	*model 1:* $\mathbf{X} \parallel \mathbf{Z}$	*model 2:* $\mathbf{A} = \mathbf{T}^{\mathbf{CT}}$	Homophily
Cora	82.01±0.8	**83.66**±0.6	67.96±0.8	*81.0%*
Pubmed	81.61±0.3	**86.07**±0.1	68.19±0.7	*80.0%*
Citeser	70.81±0.5	**72.26**±0.5	66.71±0.6	*73.6%*
Cornell	59.19±3.5	58.02±3.7	**69.04**±2.2	*30.5%*
Actor	29.59±0.4	29.35±0.4	**31.98**±0.3	*21.9%*
Wisconsin	68.05±6.2	69.25±5.1	**79.05**±2.1	*19.6%*



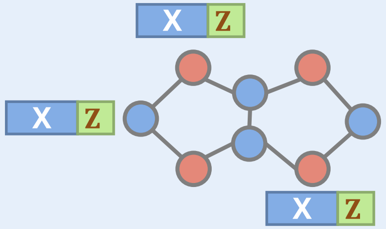
Arnaiz-Rodriguez, A., et al. "DiffWire: Inductive Graph Rewiring via the Lovász Bound". In Learning on Graphs Conference (LoG), 2022.
Li, P., et al. "Distance encoding: Design provably more powerful neural networks for graph representation learning". In NeurIPS, 2020.
Velingker, A., et al. "Affinity-Aware Graph Networks." arXiv preprint arXiv:2206.11941, 2022.

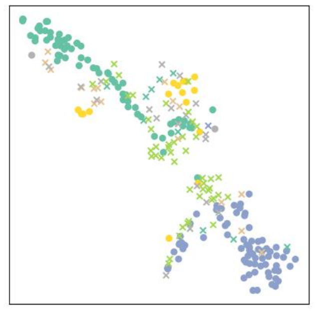
CT-Layer

Node Classification. CT-Diffusions vs CT as Positional Encoding.

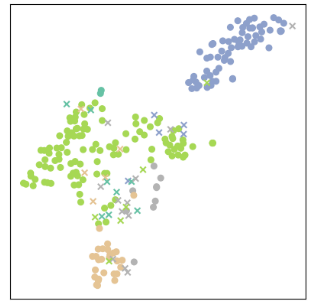
$$\mathbf{X} \rightarrow \boxed{\text{MLP tanh} \rightarrow \mathbf{Z} \in \mathbb{R}^{n \times O(n)} \rightarrow \frac{\text{cdist}(\mathbf{Z})}{\text{vol}(G)}} \rightarrow \mathbf{T}^{\mathbf{CT}} \in \mathbb{R}^{n \times n}$$

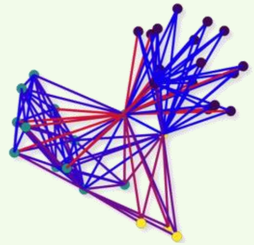
- **CTE** as structural feature (PE) reinforces performance in **homophily** tasks



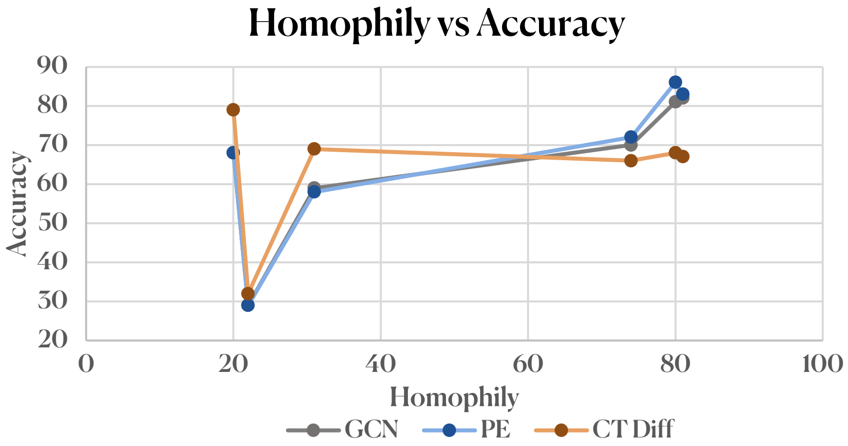


- **CT Distance** for diffusion helps in **heterophilic** tasks





Dataset	GCN (baseline)	*model 1:* $\mathbf{X} \parallel \mathbf{Z}$	*model 2:* $\mathbf{A} = \mathbf{T}^{\mathbf{CT}}$	Homophily
Cora	82.01±0.8	**83.66**±0.6	67.96±0.8	*81.0%*
Pubmed	81.61±0.3	**86.07**±0.1	68.19±0.7	*80.0%*
Citeser	70.81±0.5	**72.26**±0.5	66.71±0.6	*73.6%*
Cornell	59.19±3.5	58.02±3.7	**69.04**±2.2	*30.5%*
Actor	29.59±0.4	29.35±0.4	**31.98**±0.3	*21.9%*
Wisconsin	68.05±6.2	69.25±5.1	**79.05**±2.1	*19.6%*



Arnaiz-Rodriguez, A., et al. "DiffWire: Inductive Graph Rewiring via the Lovász Bound". In Learning on Graphs Conference (LoG), 2022.
Li, P., et al. "Distance encoding: Design provably more powerful neural networks for graph representation learning". In NeurIPS, 2020.
Velingker, A., et al. "Affinity-Aware Graph Networks." arXiv preprint arXiv:2206.11941, 2022.

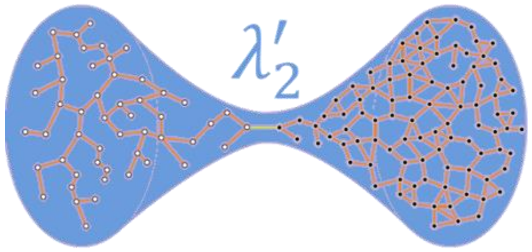
ellis ALICANTE

GAP-Layer

Spectral Derivatives

- **Goal**: Optimize bottleneck width

λ_2 := spectral gap **or bottleneck size**



- Search $\widetilde{\mathbf{A}}$ as similar as $\mathbf{A}$ but **minimizing bottleneck size**
  - **Spectral derivatives**

$$L_{Fiedler} = \left\|\widetilde{\mathbf{A}} - \mathbf{A}\right\|_F + \alpha(\boldsymbol{\lambda}_2)^2$$

$$\nabla_{\widetilde{\mathbf{A}}}\lambda_2 := Tr[(\nabla_{\tilde{\mathbf{L}}}\lambda_2)^T \nabla_{\widetilde{\mathbf{A}}}\tilde{\mathbf{L}}] = \mathrm{diag}(\mathbf{f}_2\mathbf{f}_2^T)\mathbf{1}\mathbf{1}^T - \mathbf{f}_2\mathbf{f}_2^T$$

But λ_2 and f_2 are spectrally computed? NO!
GAP-Layer learns them

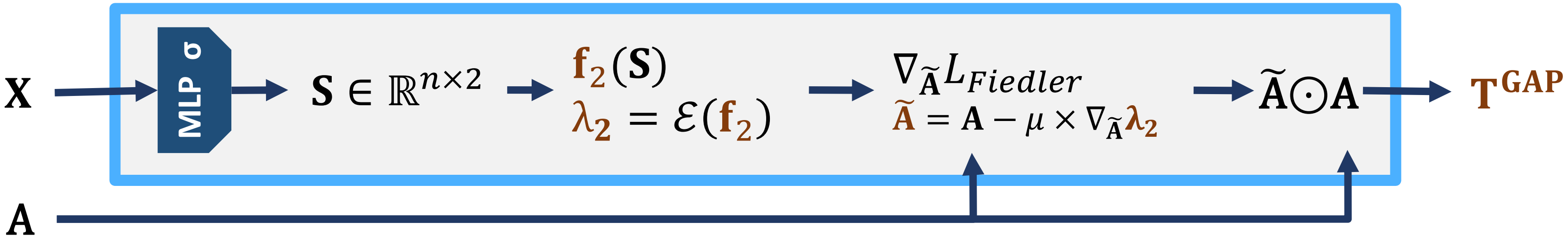
- $\mathbf{f}_2 \in \mathbb{R}^n$:= Fiedler vector
  - $\mathbf{f}_2$: Node membership to each of the 2 clusters
  - λ_2: Eigenvalue of $\mathbf{f}_2$ (Dirichlet energies of $\mathbf{f}_2$)

- Main problem: λ_2 and $\mathbf{f}_2$ are **usually spectrally computed**

Kang, J. and Tong, H. "N2n: Network derivative mining". In CIKM, 2019.

GAP-Layer

Gap-Layer: Approximating the Fiedler vector



$$L_{cut} = \frac{\mathrm{Tr}[\mathbf{S}^{\mathbf{T}}\mathbf{L}\mathbf{S}]}{\mathrm{Tr}[\mathbf{S}^{\mathbf{T}}\mathbf{D}\mathbf{S}]} + \left\| \frac{\mathbf{S}^{\mathbf{T}}\mathbf{S}}{\|\mathbf{S}^{\mathbf{T}}\mathbf{S}\|_F} - \frac{\mathbf{I}_N}{\sqrt{2}} \right\|_F$$

$$L_{fiedler} = \|\tilde{\mathbf{A}} - \mathbf{A}\|_F + \alpha(\lambda_2)^2$$

$$\nabla_{\tilde{\mathbf{A}}} \lambda_2 = \left[2(\tilde{\mathbf{A}} - \mathbf{A}) + (\mathrm{diag}(\mathbf{f}_2\mathbf{f}_2^T)\mathbf{1}\mathbf{1}^T - \mathbf{f}_2\mathbf{f}_2^T) \times \lambda_2\right]$$

How does GAP-Layer learn $\mathbf{f}_2$?

$\mathbf{S} \in \mathbb{R}^{n\times 2}$ → cluster membership

$$\mathbf{f}_2(\mathbf{S}) = \begin{cases} +1/\sqrt{n} \text{ if u belongs to cluster \#1} \\ -1/\sqrt{n} \text{ if u belongs to cluster \#2} \end{cases}$$

[Hoang. et al., 2020]

How does GAP-Layer learn λ_2?

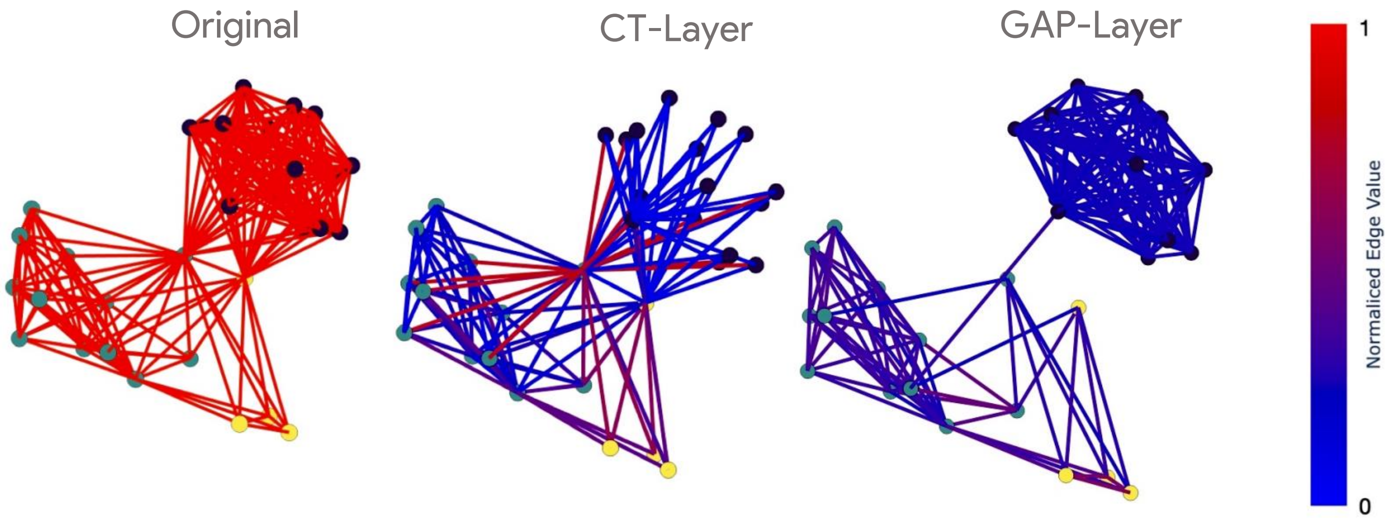
$$\lambda_2 = \mathcal{E}_G(\mathbf{f}_2) = \mathbf{f}_2^{\mathbf{T}} \mathbf{L}_G \mathbf{f}_2$$

Dirichlet energies of the approximated $\mathbf{f}_2$

Arnaiz-Rodriguez, A., et al. "DiffWire: Inductive Graph Rewiring via the Lovász Bound". In Learning on Graphs Conference (LoG), 2022.
Hoang, N.T., et al. "Revisiting graph neural networks: Graph filtering perspective". In ICPR, 2020.

GAP-Layer

Experiments



	MinCutPool	k-NN	DIGL	SDRF	CT-LAYER	GAP-LAYER (R)	GAP-LAYER (N)
REDDIT-B*	66.53±4.4	64.40±3.8	76.02±4.3	65.3±7.7	**78.45**±4.5	**77.63**±4.9	76.00±5.3
IMDB-B*	60.75±7.0	55.20±4.3	59.35±7.7	59.2±6.9	**69.84**±4.6	**69.93**±3.3	68.80±3.1
COLLAB*	58.00±6.2	58.33±11	57.51±5.9	56.60±10	**69.87**±2.4	64.47±4.0	**65.89**±4.9
MUTAG	84.21±6.3	**87.58**±4.1	85.00±5.6	82.4±6.8	**87.58**±4.4	**86.90**±4.0	**86.90**±4.0
PROTEINS	74.84±2.3	**76.76**±2.5	74.49±2.8	74.4±2.7	**75.38**±2.9	75.03±3.0	**75.34**±2.1
SBM*	53.00±9.9	50.00±0.0	56.93±12	54.1±7.1	81.40±11	**90.80**±7.0	**92.26**±2.9
Erdös-Rényi*	81.86±6.2	63.40±3.9	**81.93**±6.3	73.6±9.1	79.06±9.8	79.26±10	**82.26**±3.2

Arnaiz-Rodriguez, A., et al. "DiffWire: Inductive Graph Rewiring via the Lovász Bound". In Learning on Graphs Conference (LoG), 2022.

Future work

Rewiring

- Dynamic Rewiring wrt structure, homophily-heterophily and utility
  - Reduce or enforce over-squahing when needed (merge only util information)
- Rewiring with Interpretability

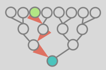
DiffWire

- Use of learned CT for different objectives
- Code to sparse → efficient computation
- Code to PyG → Easy use (even more)

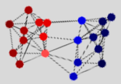
Graph Fairness

Algorithmic Fairness
with Graph Rewiring

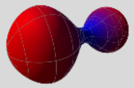
Illustration by **Justin Metz** in *Chouldechova, A. and Roth, A., 2020. A snapshot of the frontiers of fairness in machine learning. Communications of the ACM, 63(5), pp.82-89.*



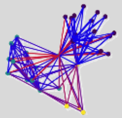
Motivation and Challenges



Introduction to Spectral Theory



Transductive Rewiring



Inductive Rewiring

Graph Fairness

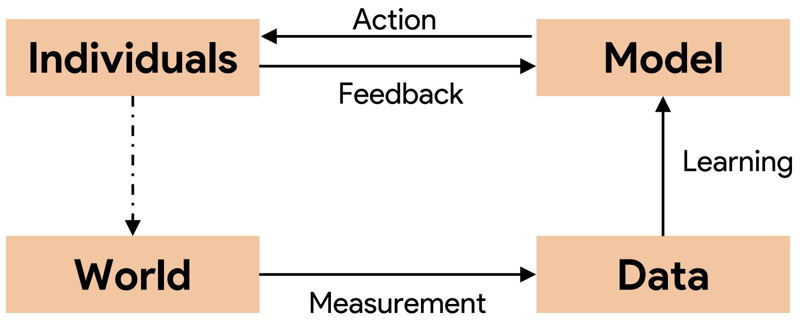
Panel Discussion

Algorithmic Fairness

ML for Critical Decision Making




- ✓ Privacy
- ✓ Transparency
- ✓ Accountability
- ✓ Reliability
- ✓ Autonomy
- ✓ **Fairness**



Social Biased decisions leads to

INDIVIDUAL HARMS (ILLEGAL DISCRIMINATION / UNFAIR PRACTICES)	COLLECTIVE SOCIAL HARMS
HIRING	LOSS OF OPPORTUNITY
EMPLOYMENT	
INSURANCE & SOCIAL BENEFITS	
HOUSING	
EDUCATION	
CREDIT	ECONOMIC LOSS
DIFFERENTIAL PRICES OF GOODS	
LOSS OF LIBERTY	SOCIAL STIGMATIZATION
INCREASED SURVEILLANCE	
STEREOTYPE REINFORCEMENT	
DIGNATORY HARMS	

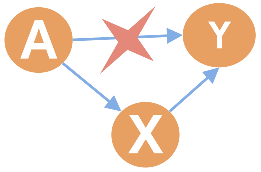
Buolamwini, J., et al. "Gender shades: Intersectional accuracy disparities in commercial gender classification". In FAccT, 2021

Algorithmic Fairness

Independence on the Protected Attributes

Ensure that the **outputs** of a model **DO NOT depend on sensitive attributes**

$$F(\mathbf{X}) = R, \qquad S \in \mathbf{X} \rightarrow R \perp S$$



Group Fairness

Groups (defined by sensitive attributes) are treated equally

$P(R\|S)$	$P(R\|Y,S)$	$P(Y\|R,S)$
Independence	*Separation*	*Sufficiency*
$R \perp S$	$R \perp S \mid Y$	$S \perp Y \mid R$

Demographic parity

P(R=1|S=*a*) = P(R=1|S=*b*)

Positive Predicted Ratio: Equal acceptance rate

Equalized odds

P(R=1 | Y=i, S=*a*) = P(R=1 | Y=i, S=*b*), i ∈ 0, 1

TPR – FPR Equal error rates

Predictive Parity

P(Y=1 | R=1, S=*a*) = P(Y=1 | R=1 , S=*b*)

PPV – NPV Equal success rate

Individual Fairness

Treat similar individuals in a similar way

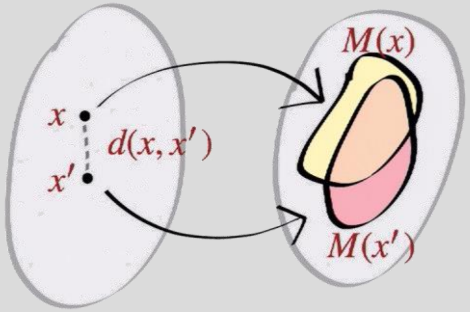
Our Dataset: $D = \{(x_i, y_i)\}_i^N$

Distance between x_i pairs: $k: V \times V \rightarrow R$.

Mapping from x_i to outcomes probability distribution $M: V \rightarrow \alpha S$

Distance between distributions of outputs D

$$D(M(x), M(y)) =< k(x, y)$$



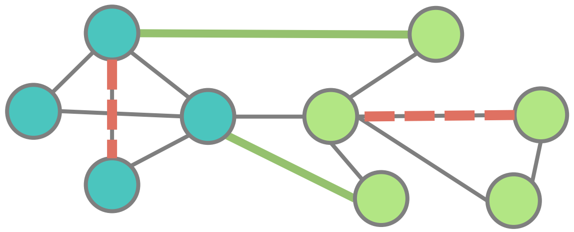
Barocas, S., et al. "Fairness in machine learning". NeurIPS tutorial, 2017
Dwork, C., et al. "Fairness through awareness". Proceedings of the 3rd innovations in theoretical computer science conference, 2012.

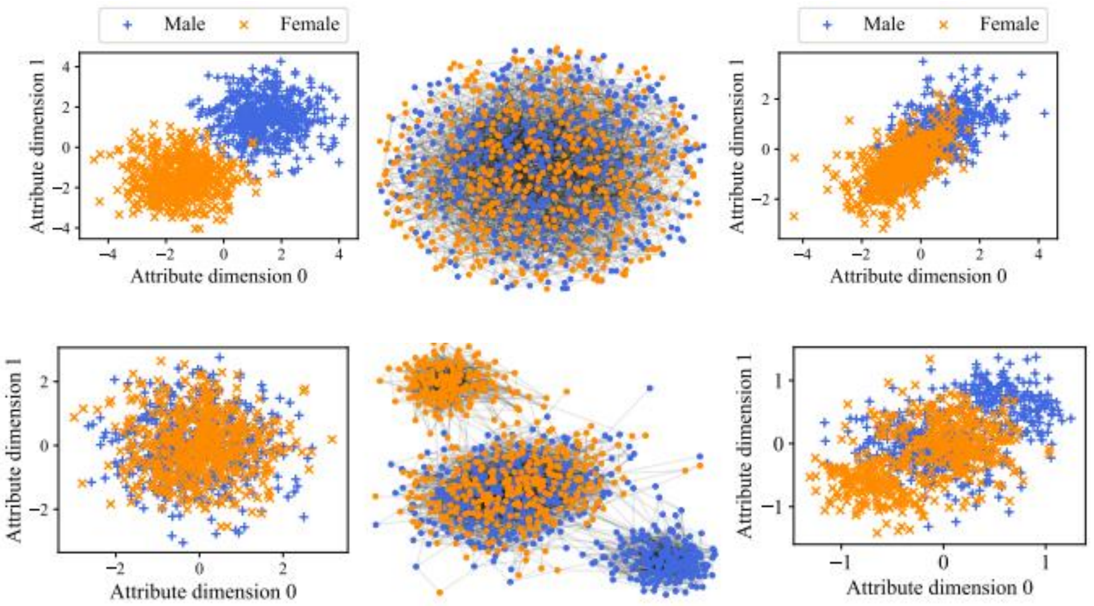
Why Graph Fairness?

The Graph Structure: a New Biased Element

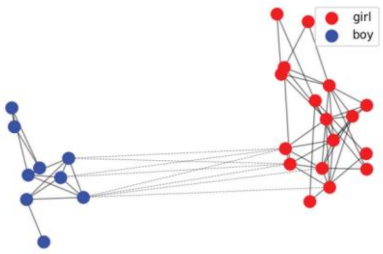
Topology of the graph (A) can be biased → correlated with sensitive attributes

- **Over-representing** **homophilic** **edges**
  (social stratification, fraudulent links, social homophily [McPherson, 2001])
- **Missing** **heterophilic** **edges** that
  would have been present in more fair settings





[Dong, 2022]



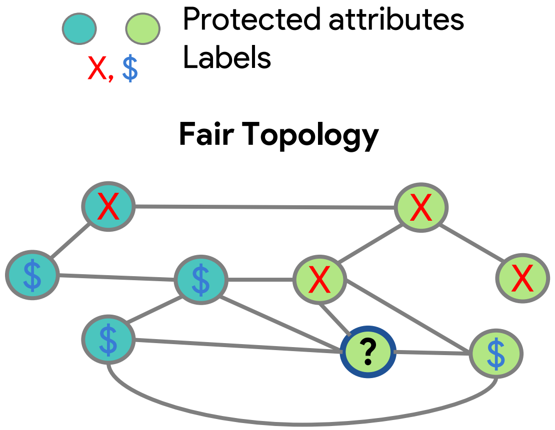
Friendship among students in a Dutch School
[Masrour, 2020]

McPherson, M., et al. "Birds of a feather: Homophily in social networks". Annual review of sociology 27, 2001
Dong, Y., et al. "Edits: Modeling and mitigating data bias for graph neural networks". In the Web Conference, 2022.
Masrour, F., et al. "Bursting the filter bubble: Fairness-aware network link prediction". In AAAI, 2020.

Why Graph Fairness?

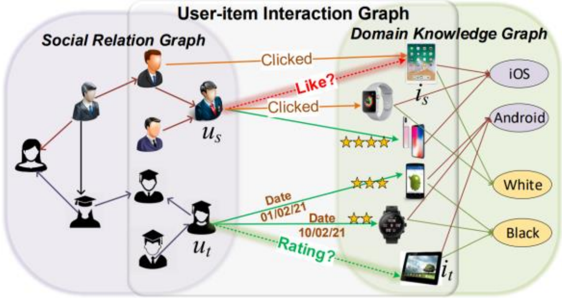
Consequences on the Real world

Decisions on the nodes	Recommendations	Influence Maximization

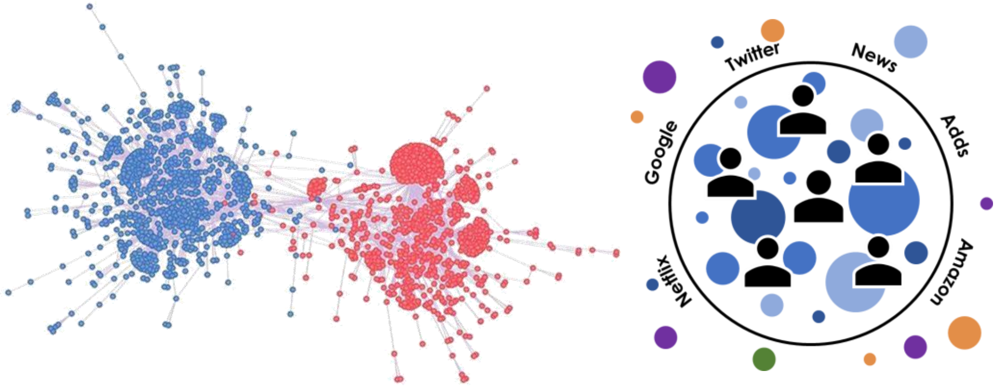


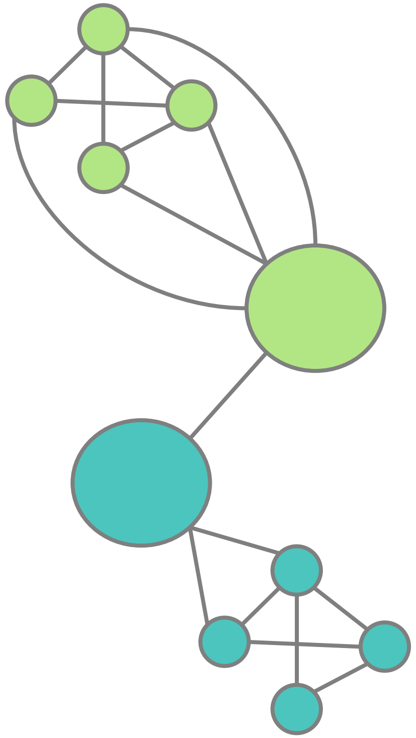
* *Also applies to community detection and Link prediction*

Biased recommendations
(Products, jobs, content...)



Echo Chambers and Filter bubbles

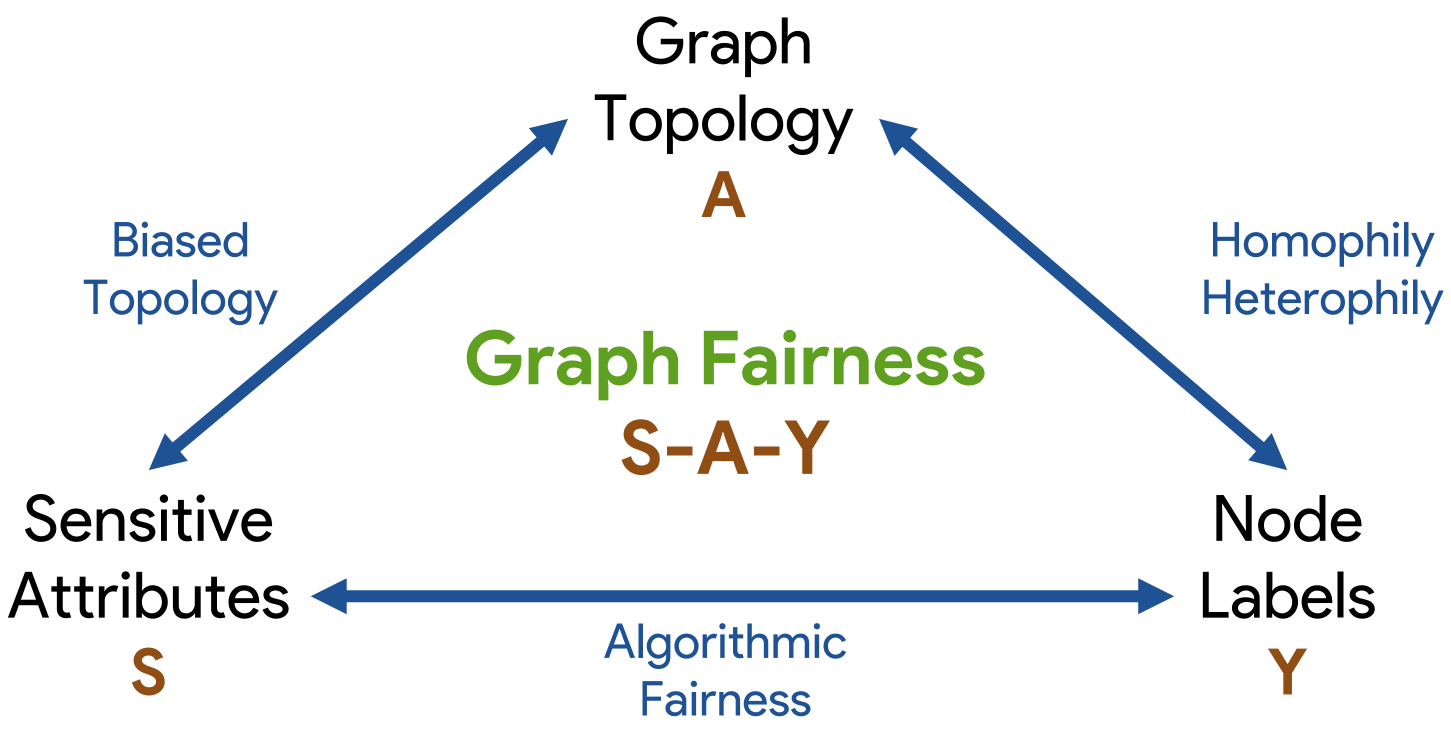




Hampson, M. "Smart Algorithm Bursts Social Networks' 'Filter Bubbles'. 2011. [Link]
Wang, S., et al. "Graph learning based recommender systems". In IJCAI, 2021

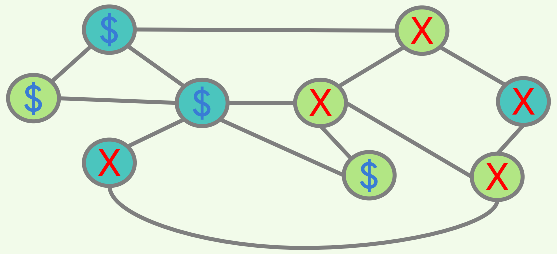
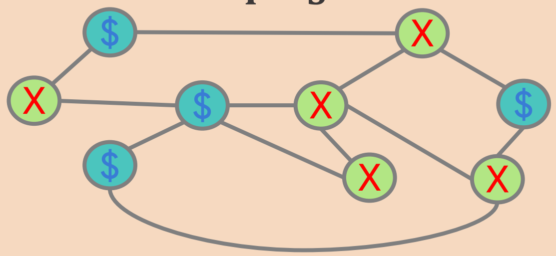
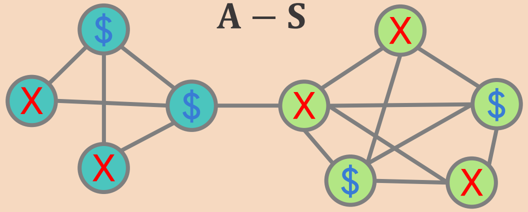
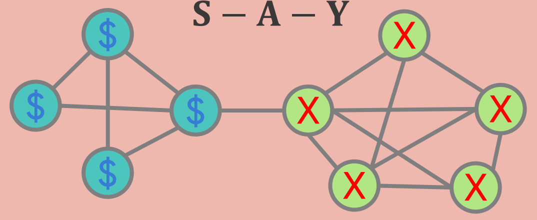
Graph Fairness

Causes



Graph Fairness

Causes

	Unbiased Attributes	Biased Attributes [Y-S]
Unbiased Structure		Label correlates with the protected attribute $\mathbf{Y - S}$  Individuals are biased, their relationships are not
Biased Structure [A-S]	Structure correlates with the protected attribute $\mathbf{A - S}$  **Graph generation process is Biased** People built **relationships** in correlation **only with A** Model will have **bad accuracy and biased** decision *Assortativity in protected attribute*	Both correlate with the protected attribute $\mathbf{S - A - Y}$  **Society is Biased** People built **relationships** in correlation with **Y and/or A** Model will have **good accuracy but biased** decision *Assortativity in protected attribute (S-A) and label homophily (A-Y)*

$\mathbf{S}$: Sensitive attribute $\mathbf{A}$: Adjacency, i.e. Matrix Structure $\mathbf{Y}$: Node Label

Dimensions of taxonomy

Perspectives to analyze graph fairness

Causes

- A-S correlation
- Y-S correlation
- Y-A correlation (homophily)

Fairness definitions

- Node-level decision
  - Group
    - ...
  - Individual
    - ...
- Structure segregation
  - Group
    - ...
  - Individual
    - ...

Tasks

- Topology analysis
- Representation Learning
- Classification/Regression
- Link prediction
- Community detection
- Application specific
  - Recommender systems
  - Influence maximization
  - Ranking

Techniques

- Constrained optimization
- Adversarial/orthogonal
- Rebalancing
- **Graph Rewiring**

ellis ALICANTE

Graph Fairness Definitions

Definitions and Metrics

- **Graph Fairness**
  - **Node Fairness** — *Focused on node-level decisions*
    - **Group**
      - Node clf
        - Embedding
      - Influence maximization
        - Counterfactual

      Node embedding $\perp\!\!\!\perp$ S
      Node prediction $\perp\!\!\!\perp$ S
    - **Individual**
      - Node clf
        - Embedding
      - Degree
        - Ranking or distances

      Similar* nodes should lead to similar predictions.
      ** Similarity can be n be given based on X, A...*
  - **Structural Fairness** — *Focused on structure segregation*
    - **Group**
      - Clustering
      - Original Structure
      - Link pred *Dyadic*

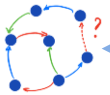
      Structure $\perp\!\!\!\perp$ S
      No Assortativity in S
      No communities based on S
      Link prediction $\perp\!\!\!\perp$ S
      $P((u,v)|A_u = A_v) = P((u,v)|A_u \neq A_v)$
    - **Individual**
      - Clustering
      - Original Structure
      - Link pred

      Similar nodes should be connected
      $P((u,v) \in E) = d(x_u, x_v)$
      or they should have highly overlapped neighboring node sets

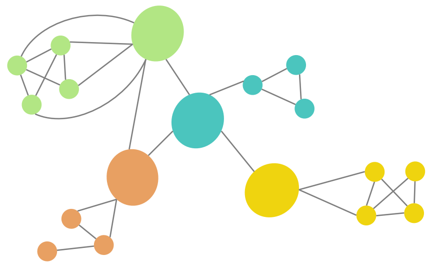
Graph Fairness Definitions

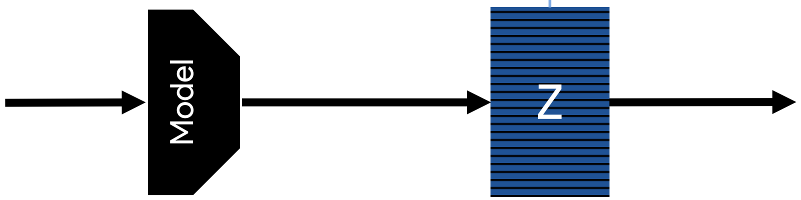
Definitions and metrics from a Pipeline Point of View

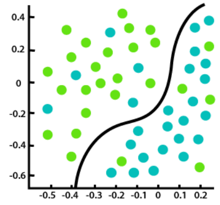
Link Prediction



- **Group**
  - *Dyadic Fairness*
  - *DI / EOd / EOp*
- **Individual**
  - $Tr(A^T \phi^T L_K \phi A)$







Classification

Classification

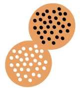
- **Group**
  - *DI/EOd/Eop*
  - *Counterfactual*
    (% change f(X) when change A)
- **Individual**
  - *Consistency*
  - *Degree*

Topology

- **Group**
  - *Assortativity*
  - *Attribute Homophily*

Embeddings

- **Group**
  - *Distributional Bias*
  - *Representation Bias (AUC predicting A)*
- **Individual**
  - *Pair Distance* $\mathrm{Tr}\left(\mathbf{Y}^{\mathbf{T}}\mathbf{L}_{\mathbf{s}}\mathbf{Y}\right)$
  - *Ranking similarity*



Clustering

- **Group**
  - *Balance (SC)*
  - *ModRed*
- **Individual**
  - *SC*

Influence Max

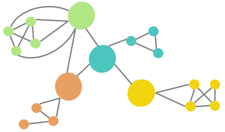
Rec Sys

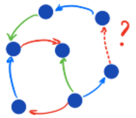
Ranking

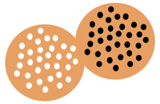
ellis ALICANTE

Graph Fairness Definitions

Definitions and metrics from a Pipeline Point of View







Topology

Group

- Assortativity [Newman, 2003]
- Modularity: *modred* [Masrour, 2020]

$$Q = \frac{1}{2|E|}\sum_{ij}\left(A_{ij} - \frac{d_i d_j}{2|E|}\right)(S_u \otimes S_v)$$

$$\text{modred} = \frac{\mathrm{Q_{ref}} - \mathrm{Q_{pred}}}{\mathrm{Q_{ref}}}$$

- Attribute Homophily

$$h_{edges} = \frac{|\{(u,v) \in E : S_u = S_v\}|}{|E|}$$

$$h_{nodes} = \frac{1}{|V|}\sum_{v\in V}\frac{|\{u \in N(v) : S_u = S_v\}|}{|N(v)|}$$

- Information Unfairness Score [Jalali 2020]

$$M = \sum \theta A^k\ ;\ \mathrm{Max}(\{d : d = D(M_{S_u=i,S_v=j}, M_{S_k=S_v=l})\})\forall i \in \{0,1\}$$

Individual

- Dirichlet energies wrt node features **X**

$$\mathcal{E}(\boldsymbol{y}) = Tr[\mathbf{X}^{\mathrm{T}}\mathbf{L}\mathbf{X}]$$

Link Prediction

Group – Dyadic Fairness

- Statistical Parity or Disparate Impact [Laclau, 2020; Rahman, 2019; Buyl, 2020; Li, 2021; Spinelli, 2021]

$$P((u,v)|S_u = S_v) = P((u,v)|S_u \neq S_v)$$

$$\frac{P\big((u,v)\big|S_u = S_v\big)}{P\big((u,v)\big|S_u \neq S_v\big)}$$

- Equal opportunity [Buyl, 2020 ; Li, 2021]

$$P\big((u,v)\big|y_{uv} = 1, S_u = S_v\big) = P\big((u,v)\big|y_{uv} = 1, S_u \neq S_v\big)$$

- Equalized odds [Li, 2021]

$$P\big((u,v)\big|y_{uv} = i, S_u = S_v\big) = P\big((u,v)\big|y_{uv} = i, S_u \neq S_v\big)\ \forall i \in \{0,1\}$$

** More group metrics like ARP [Rahman, 2019] or DI or EO based on group and subgroup dyadic-specific [Spinelli, 2021] (above metrics are mixed dyadic)*

Individual

$$\mathcal{E}(\boldsymbol{y}) = Tr[\mathbf{X}^{\mathrm{T}}\mathbf{L}'\mathbf{X}]$$

$$P\big((u,v) \in E\big) = d(x_u, x_v)$$

Clustering

Group

- *Same proportion of each group in each cluster as in the population as a whole*

$$\text{balance} = \min\frac{|\mathrm{V_s} \cap \mathrm{C_k}|}{|\mathrm{V_s'} \cap \mathrm{C_k}|}$$

- *For each node, each cluster must contain an adequate number of members similar* to the individual. *similar defined by some graph R built using sensitive attributes.*

$$\frac{\big|\{j : R_{ij} = 1 \wedge v_j \in \mathrm{C_k}\}\big|}{|C_k|} = \frac{\big|\{j : R_{ij} = 1\}\big|}{|V|}, \forall k \in K$$

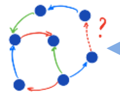
Individual

$$\frac{1}{|C(u)| - 1}\sum_{v\in C(u)} d(u,v) \leq \frac{1}{|C_k|}\sum_{v\in C_k} d(x,y), \forall k \in K$$

Masrour, F., et al. "Bursting the filter bubble: Fairness-aware network link prediction". In AAAI, 2020.
Buyl M., et al.. "Debayes: a bayesian method for debiasing network embeddings". In ICML, 2020.
Jalali Z. S., et al. "On the information unfairness of social networks". In SDM, 2020
Newman, M. "Assortative mixing in networks". Phys. Rev. Lett., 89, 2002.
Laclau, C., et al. "All of the Fairness for Edge Prediction with Optimal Transport". In ICAIS, 2020.
Rahman, T. et al. "FairWalk: Towards Fair Graph Embedding". In IJCAI, 2019.
Li, P., et al. "On dyadic fairness: Exploring and mitigating bias in graph connections". In ICLR, 2021.
Spinelli, I., et al. "FairDrop: Biased edge dropout for enhancing fairness in GRL". In TAI 2021

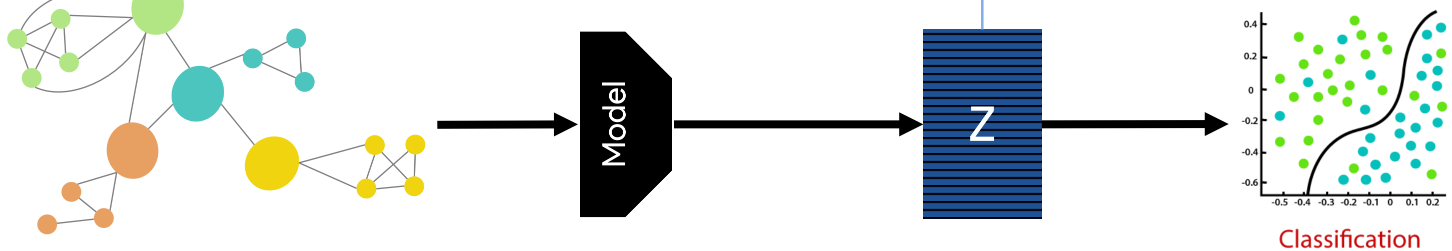
Graph Fairness Definitions

Definitions and metrics from a Pipeline Point of View



Link Prediction

- **Group**
  - *Dyadic Fairness*
  - *DI / EOd / EOp*
- **Individual**
  - $Tr(A^T \phi^T L_K \phi A)$



Classification

Classification

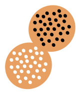
- **Group**
  - *DI/EOd/Eop*
  - *Counterfactual*
    (% change f(X) when change A)
- **Individual**
  - *Consistency*
  - *Degree*

Topology

- **Group**
  - *Assortativity*
  - *Attribute Homophily*

Embeddings

- **Group**
  - *Distributional Bias*
  - *Representation Bias (AUC predicting A)*
- **Individual**
  - *Pair Distance* $\mathrm{Tr}\left(\mathbf{Y}^{\mathbf{T}}\mathbf{L_s}\mathbf{Y}\right)$
  - *Ranking similarity*



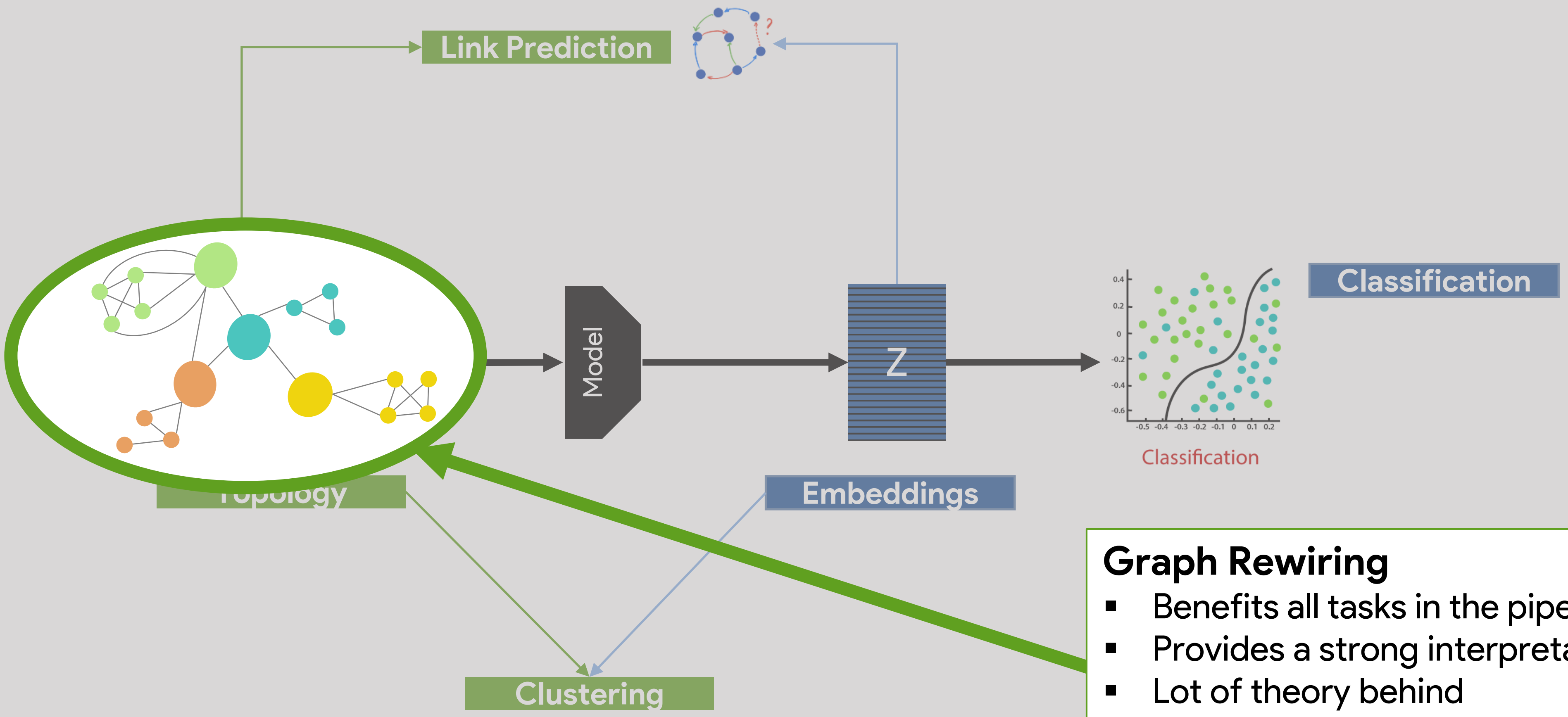
Clustering

- **Group**
  - *Balance (SC)*
  - *ModRes*
- **Individual**
  - *SC*

Influence Max
Rec Sys
Ranking

Graph Fairness Definitions

Definitions and metrics from a Pipeline Point of View



Graph Rewiring

- Benefits all tasks in the pipeline
- Provides a strong interpretability
- Lot of theory behind
- Aligned with other problems in GNNs
  - Homophily/Heterophily
  - Expressiveness

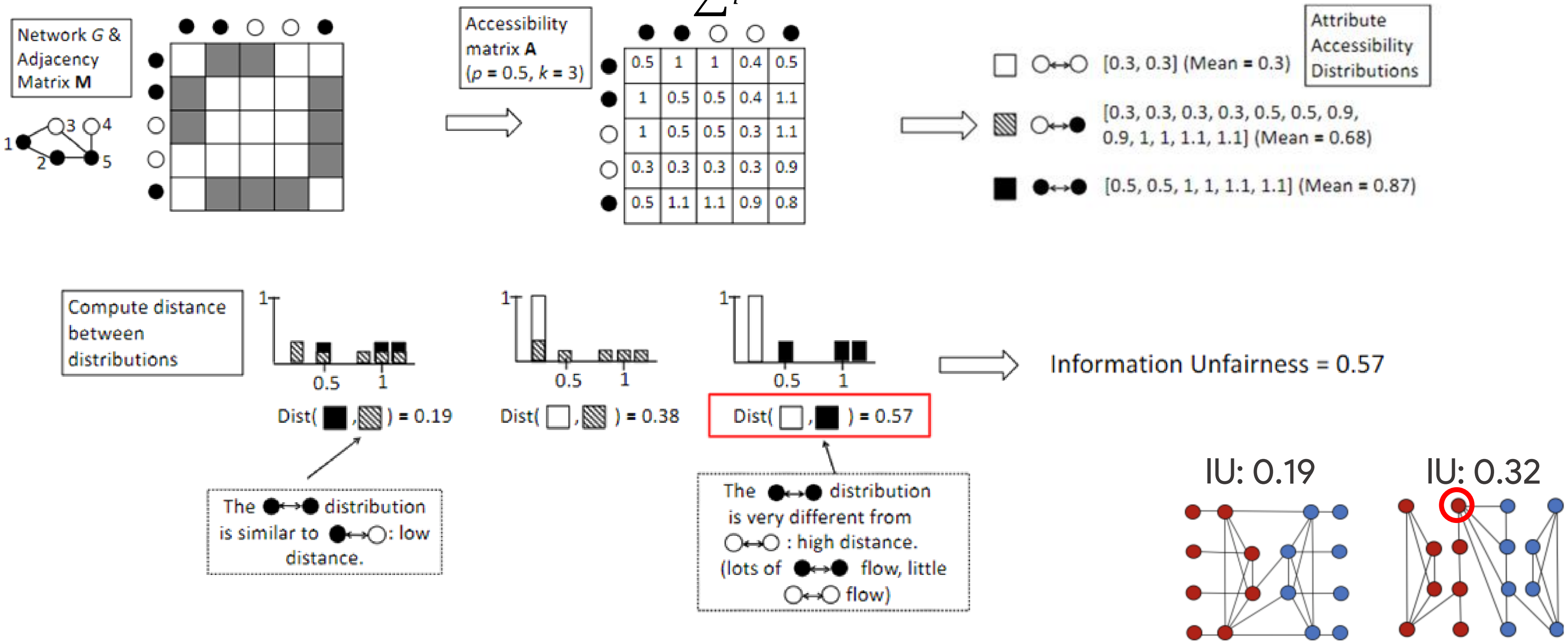
Information Unfairness

Maximum difference between distribution of intra and inter edge weights

$$\mathbf{A} = \sum p\mathbf{M}^k$$

joint attribute accessibility distribution

$$\mathbf{D}_{fg} = \{\mathbf{A}_{uv}: S_u = f, S_v = g\}$$



IU: 0.19 IU: 0.32

Assortativity = 0.66

*same intra-inter edges

Rewiring for Topology Debiasing

On the Information Unfairness of Social Networks

MaxFair

Find b edges such that the IU of $G' = (V, E \cup B)$ is minimized

1. **Calculate Node-Attribute centrality:** Quantify how well a node spreads information into a group
   - $vec_s \in \mathbb{R}^{1\times n} = \sum_k p^k \times vec_{s,k}$: each node's centrality with respect to sensitive group. One for each $s \in S$.
     - $\mathrm{vec}_{s,0}$: vector of node membership to sensitive group. i.e. $vec_{s,0}(u) = 1 \text{ if } S_u = \text{f else } 0$
     - $\mathrm{vec}_{s,k}(u) = sum\left(\left[vec_{s,k-1}(v)\right]_{j\in N(u)}\right)$: message passing using $\mathrm{vec}_{s,0}$ as initial feature

Weighted MultiHead Message passing using one hot encoded sensitive attribute as X

2. **Score unconnected pair of nodes using vec_s**

   - $\mathbf{A} = \sum p\mathbf{M}^k \rightarrow \mathbf{D}_{fg} = \{\mathbf{A}_{uv}: S_u = f, S_v = g\},\ \forall\, f, g \in S$, i.e.
   - $s_{fg} = \mathrm{mean}(\mathbf{A}) - \mathrm{mean}(\mathbf{D}_{fg})$. How each distribution deviate from the mean of all edges.
   - $\mathrm{score}(u, v) = \sum_{f,g\in S} s_{fg} * (vec_f(u) * vec_g(v) + vec_g(u) * vec_f(v))$

How a given edge would relief over-squashing between 2 different communities defined by sensitive attributes?

3. **Select the highest scoring edge**

Jalali Z. S., et al. "On the information unfairness of social networks". In SDM, 2020. [Video]

Rewiring for Fair Link Prediction

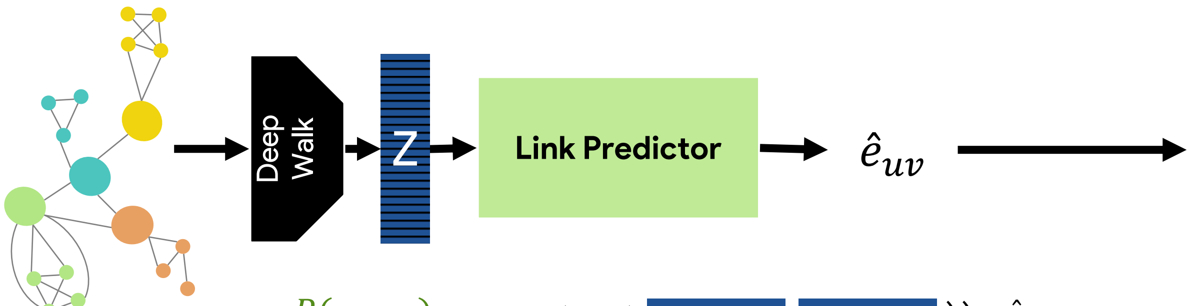
Bursting the Filter Bubble: Fairness-aware network link prediction

Evaluate Structural Fairness by change in modularity after link prediction

$$Q = \frac{1}{2|E|}\sum_{ij}\left(A_{ij} - \frac{d_i d_j}{2|E|}\right)(S_u \otimes S_v) \qquad \text{modred} = \frac{Q - Q'}{Q}$$

Greedy-FLIP

Greedy rewiring at post-processing



$$P(z_u, z_v) = \text{relu(MLP(} \; z_u \; z_v \; \text{))} = \hat{e}_{uv}$$

$$\mathcal{L}_P = \text{cross\_entropy}(e_{uv}, \hat{e}_{uv})$$

How flipping an edge prediction change the modularity?

Flip edge with the lowest score and repeat

$$score(\dot{e}_{xy}) = \frac{(-1)^{\delta(\dot{e}_{xy})}}{2m}\left(-1 + \frac{d_x + d_y - 1}{2m}\right)\delta(X_x^{(p)}, X_y^{(p)}) + \left(\sum_{\substack{v \in V, X_v^{(p)} \neq X_x^{(p)} \\ v \neq y}} d_v + \sum_{\substack{v \in V, X_v^{(p)} \neq X_y^{(p)} \\ v \neq x}} d_i\right)/4m^2$$

Masrour, F., et al. "Bursting the filter bubble: Fairness-aware network link prediction". In AAAI, 2020

Adversarial Learning for Fair Link Prediction

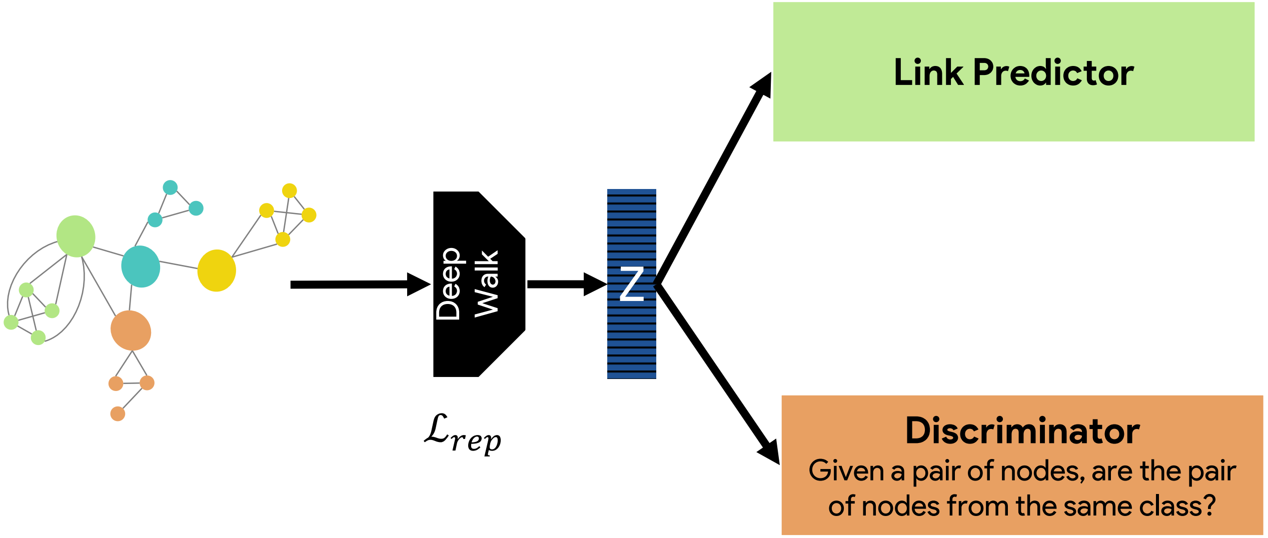
Bursting the Filter Bubble: Fairness-aware network link prediction

Evaluate Structural Fairness by change in modularity after link prediction

$$Q = \frac{1}{2|E|}\sum_{ij}\left(A_{ij} - \frac{d_i d_j}{2|E|}\right)(S_u \otimes S_v) \qquad \text{modred} = \frac{Q - Q'}{Q}$$

FLIP

Adversarial Rewiring



$$P(z_u, z_v) = \texttt{relu(MLP(}\ z_u\ \ z_v\ \texttt{))} = \hat{e}_{uv}$$

$$\mathcal{L}_P = \text{cross\_entropy}(e_{uv}, \hat{e}_{uv})$$

$$D(z_u, z_v) = \texttt{relu(MLP(}\ z_u\ \ z_v\ \texttt{))} = \hat{g}_{uv}$$

$$\mathcal{L}_D = \text{cross\_entropy}(g_{uv}, g_{uv})$$

$$\mathcal{L} = \alpha\mathcal{L}_{rep} - (1-\alpha)\mathcal{L}_D + \beta\mathcal{L}_P$$

Masrour, F., et al. "Bursting the filter bubble: Fairness-aware network link prediction". In AAAI, 2020

Rewiring for Fair Link Prediction

On dyadic fairness: Exploring and mitigating bias in graph connections

Dyadic Fairness: $P((u,v)|S_u = S_v) = P((u,v)|S_u \neq S_v)$ → predict equal number of 

** Also, same TPR, TNR, FPR and FNR*

FairAdj

Rewire the graph topology to get fair embeddings to perform fair link prediction using projected gradient descent → maintain $\mathbf{A}$ nature

- They prove that their rewiring reduces an upper bound of a constant that, if low, is a sufficient condition for Dyadic Fairness
  - It reduces the disparity of representation between nodes of different groups after message passing

Until convergence of $\boldsymbol{\theta}$ or $\tilde{\mathbf{A}}$

Train $\boldsymbol{\theta}$ *n* epochs for utility

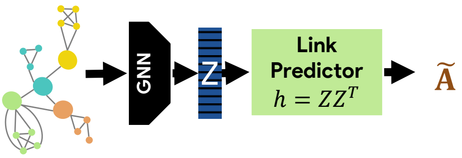
$$\max_{\boldsymbol{\theta}} \mathcal{L}_{VGAE} \coloneqq \mathrm{E}[\log p(\mathbf{A}|\mathbf{Z})] - K[\mathrm{GNN}(\mathbf{Z}|\mathbf{X},\tilde{\mathbf{A}})||N(0,1)]$$

Train $\tilde{\mathbf{A}}$ *m* epochs for fairness

$$\min_{\tilde{\mathbf{A}}} \mathcal{L}_{\mathrm{fair}} \coloneqq \left\| \mathrm{E}_{u,v\sim UxU}[h(u,v)|S_u = S_v] - \mathrm{E}_{u,v\sim UxU}[h(u,v)|S_u \neq S_v] \right\|^2$$

$$\text{s.t } [\tilde{\mathbf{A}}]_{uv} = 0 \text{ if } [\mathbf{A}]_{uv} = 0 \text{ and } \tilde{\mathbf{A}}\mathbf{1} = \mathbf{1}$$

Same probability of **inter** and **intra** links without adding edges and being a RW matrix



$$\tilde{\mathbf{A}} = \tilde{\mathbf{A}} - (\eta \nabla_{\tilde{\mathbf{A}}} \mathcal{L}_{\mathrm{fair}})$$

Modify $\tilde{\mathbf{A}}$ to **minimize** $\mathcal{L}_{\mathrm{fair}}$

Project $\tilde{\mathbf{A}} - (\eta \nabla_{\tilde{\mathbf{A}}} \mathcal{L}_{\mathrm{fair}})$ to the feasible space

$$\tilde{\mathbf{A}} - (\eta \nabla_{\tilde{\mathbf{A}}} \mathcal{L}_{\mathrm{fair}})\mathbf{1} = \mathbf{1}$$

Modify $\tilde{\mathbf{A}}$ to be row stochastic

Li, P., et al. "On dyadic fairness: Exploring and mitigating bias in graph connections". In ICLR, 2021

Rewiring for Fair Representation Learning

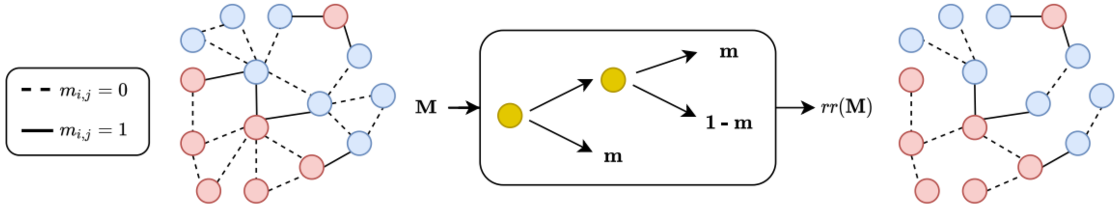
FairDrop: Biased edge dropout for enhancing fairness in Graph Rrepresentation Learning

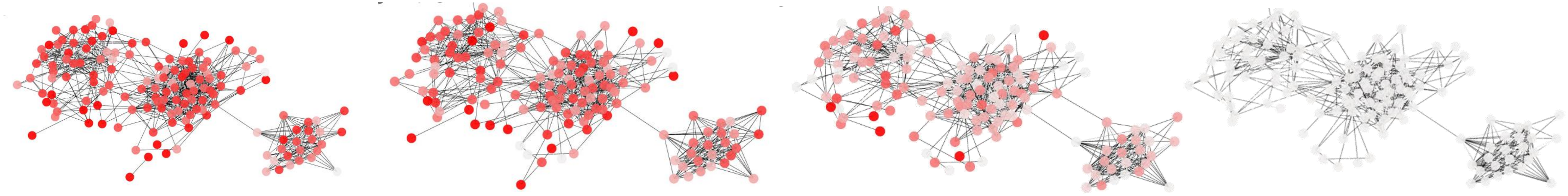
Fairness: AUC predicting S *they also perform link prediction evaluated with dyadic fairness

FairDrop

Fair edge dropout

- Dropout homophilic edges with prob ½ + $\boldsymbol{\delta}$
- Dropout heterophilic edges with prob ½ − $\boldsymbol{\delta}$





$\boldsymbol{\delta} = 0$ $\boldsymbol{\delta} = 0.1$ $\boldsymbol{\delta} = 0.35$ $\boldsymbol{\delta} = 0.5$

Spinelli, I., et al. "FairDrop: Biased edge dropout for enhancing fairness in GRL". In TAI 2021

That's not all Folks!

More Graph Rewiring Methods for Graph Fairness

<u>RW for topology debiasing</u>

- *MaxFair*
  Jalali Z. S., et al. "On the information unfairness of social networks". In SDM, 2020

<u>RW first for link prediction</u>

- *Greedy-FLIP*
  Masrour, F., et al. "Bursting the filter bubble: Fairness-aware network link prediction". In AAAI, 2020.
- *FairAdj*
  Li, P., et al. "On dyadic fairness: Exploring and mitigating bias in graph connections". In ICLR, 2021.
- *FairDrop*
  Spinelli, I., et al. "FairDrop: Biased edge dropout for enhancing fairness in GRL". In TAI 2021
- OT: Individual Fairness
  Laclau, C., et al. "All of the Fairness for Edge Prediction with Optimal Transport". In ICAIS, 2020.

<u>RW for fair representation learning</u>

- InForm: Individual Fairness
  Kang, J. et al. "Inform: Individual fairness on graph mining". In SIGKDD 2020.
- *FairDrop* – [Spinelli, I., 2021]
- *FairAdj* – [Li, P., 2021]

<u>RW for node classification</u>

- OT - [Laclau, C., 2020]
- EDITS
  Dong, Y., et al. "EDITS: Modeling and mitigating data bias for graph neural networks". In WWW, 2022.
- FairEdit
  Loveland, Donald, et al. "FairEdit: Preserving Fairness in Graph Neural Networks through Greedy Graph Editing." preprint, 2022.

<u>RW for specific applications</u>

- Recommender systems
  - Fabbri, F., et al. " Rewiring What-to-Watch-Next Recommendations to Reduce Radicalization Pathways". In WWW, 2022.

What can we do now?

- Normalization of benchmarks, evaluation metrics and pipelines
- Formalization of Graph Fairness as happens in Algorithmic Fairness
- Beyond Dyadic fairness
- Accuracy-fairness tradeoff in Graph Fairness?
- More efficient and Interpretable Rewiring Methods
- Causality Aware GNNs for fairness
- Ethical challenges:
  - Different values and philosophical fairness definitions
  - Human-in-the-loop
  - Robustness, XAI, privacy...
  - Go beyond known, measurable, discrete and static sensitive attributes*

Acknowledgments

ellis ALICANTE

The following authors…

Adrián Arnaiz Rodríguez
ELLIS Alicante

Speaker, panel moderator and content creator

Francisco Escolano
University of Alicante
ELLIS Alicante
Speaker and content creator

Nuria Oliver
ELLIS Alicante

Panel moderator and content creator

Edwin Hancock
University of York

Speaker and content creator

Ahmed Begga
University of Valencia

Content creator

… want to thanks to LoG reviewers of DiffWire and other researches that we got feedback from!

This work is partially funded by:

ellis ALICANTE

GENERALITAT VALENCIANA
Conselleria de Innovación, Universidades, Ciencia y Sociedad Digital

EUROPEAN UNION
European Regional Development Fund

Sabadell Fundación

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

Bibliography

Graph Rewiring

- **Arnaiz-Rodríguez, A., Begga, A., Escolano, F., & Oliver, N., DiffWire: Inductive Graph Rewiring via the Lovász Bound. Proceedings of the First Learning on Graphs Conference (LoG 2022), PMLR 198, Virtual Event, December 9–12, 2022.**
- Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In Proceedings. 2005 IEEE international joint conference on neural networks, volume 2, pages 729–734, 2005. URL https://ieeexplore.ieee.org/document/1555942. 1
- Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. IEEE transactions on neural networks, 20(1):61–80, 2008. URL https: //ieeexplore.ieee.org/document/4700287. 1
- Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations (ICLR), 2017. URL https://openreview.net/forum? id=SJU4ayYgl.
- Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning, ICML, page 1263–1272, 2017. 1
- Thomas N Kipf and Max Welling. Variational graph auto-encoders. In NeurIPS Workshop on Bayesian Deep Learning, 2016. URL http://bayesiandeeplearning.org/2016/papers/BDL_16.pdf. 1
- Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 30, 2016. URL https://ojs.aaai. org/index.php/AAAI/article/view/10179. 1
- Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In Proceedings of the AAAI Conference on Artificial Intelligence, 2014. URL https://ojs.aaai. org/index.php/AAAI/article/view/8916. 1
- Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32 (1):4–24, 2021. URL https://ieeexplore.ieee.org/document/9046288.
- Petar Velickovi c, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. ´Graph Attention Networks. International Conference on Learning Representations, 2018. URL https: //openreview.net/forum?id=rJXMpikCZ. 1
- Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=F72ximsx7C1. 1
- Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id= ryGs6iA5Km. 1
- Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Advances in Neural Information Processing Systems, 2017. URL https://proceedings.neurips.cc/paper/ 2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf. 1, 3
- Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11604. 2
- Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id= i80OPhOCVH2. 2
- László Lovász. Random walks on graphs. Combinatorics, Paul erdos is eighty, 2(1-46):4, 1993. URL https://web.cs.elte.hu/~lovasz/erdos.pdf. 2, 4
- Pablo Barceló, Egor V. Kostylev, Mikael Monet, Jorge Pérez, Juan Reutter, and Juan Pablo Silva. The logical expressiveness of graph neural networks. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=r1lZ7AEKvB. 2 11 DiffWire: Inductive Graph Rewiring via the Lovász Bound
- NT Hoang, Takanori Maehara, and Tsuyoshi Murata. Revisiting graph neural networks: Graph filtering perspective. In 25th International Conference on Pattern Recognition (ICPR), pages 8376–8383, 2021. URL https://ieeexplore.ieee.org/document/9412278. 2, 7, 19
- Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In International Conference on Learning Representations, 2020. URL https://openreview. net/forum?id=S1ldO2EFPr.
- Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. CoRR, abs/1812.08434, 2018. URL http://arxiv.org/ abs/1812.08434. 2
- Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=7UmjRGzp-A. 2, 3, 6, 8, 18, 23
- Petar Velickovic. Message passing all the way up. In ´ ICLR 2022 Workshop on Geometrical and Topological Representation Learning, 2022. URL https://openreview.net/forum?id=Bc8GiEZkTe5. 2
- Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=Hkx1qkrKPr. 2, 3
- Anees Kazi, Luca Cosmo, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein. Differentiable graph module (dgm) for graph convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1–1, 2022. URL https://ieeexplore.ieee.org/document/9763421. 2, 3
- Karel Devriendt and Renaud Lambiotte. Discrete curvature on graphs from the effective resistance. arXiv preprint arXiv:2201.06385, 2022. doi: 10.48550/ARXIV.2201.06385. URL https://arxiv.org/abs/2201. 06385. 2, 6, 7, 18
- Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In Advances in Neural Information Processing Systems, 2019. URL https://proceedings.neurips.cc/paper/ 2019/file/23c894276a2c5a16470e6a31f4618d73-Paper.pdf.
- Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261, 2018. URL https://arxiv.org/abs/1806.01261. 3
- Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Benjamin Chamberlain, Michael Bronstein, and Federico Monti. Sign: Scalable inception graph neural networks. In ICML 2020 Workshop on Graph Representation Learning and Beyond, 2020. URL https://grlplus.github.io/papers/77.pdf.

Bibliography

Graph Rewiring

- Pál András Papp, Karolis Martinkus, Lukas Faber, and Roger Wattenhofer. DropGNN: Random dropouts increase the expressiveness of graph neural networks. In Advances in Neural Information Processing Systems, 2021. URL https://openreview.net/forum?id=fpQojkIV5q8.
- Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the oversmoothing problem for graph neural networks from the topological view. Proceedings of the AAAI Conference on Artificial Intelligence, 34(04):3438–3445, Apr. 2020. doi: 10.1609/aaai.v34i04.5747. URL https://ojs.aaai.org/index.php/AAAI/article/view/5747.
- Yanqiao Zhu, Weizhi Xu, Jinghao Zhang, Yuanqi Du, Jieyu Zhang, Qiang Liu, Carl Yang, and Shu Wu. A survey on graph structure learning: Progress and opportunities. arXiv PrePrint, 2021. URL https://arxiv.org/abs/2103.03036.
- Diego Mesquita, Amauri Souza, and Samuel Kaski. Rethinking pooling in graph neural networks. In Advances in Neural Information Processing Systems, 2020. URL https://proceedings.neurips.cc/paper/ 2020/file/1764183ef03fc7324eb58c3842bd9a57-Paper.pdf.
- Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In Advances in Neural Information Processing Systems, 2018. URL https://proceedings.neurips.cc/paper/2018/file/ e77dbaf6759253c7c6d0efc5690369c7-Paper.pdf.
- Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In Proceedings of the 37th International Conference on Machine Learning, 2020. URL https://proceedings.mlr.press/v119/bianchi20a.html. 3, 8
- Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. arXiv:2205.12454, 2022. URL https://arxiv.org/pdf/2205.12454.pdf. 3 12
- Ameya Velingker, Ali Kemal Sinop, Ira Ktena, Petar Velickovi ˇ c, and Sreenivas Gollapudi. Affinity-aware ´ graph networks. arXiv preprint arXiv:2206.11941, 2022. URL https://arxiv.org/pdf/2206.11941.pdf.
- Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. AAAI Workshop on Deep Learning on Graphs: Methods and Applications, 2021. URL https://arxiv.org/pdf/ 2012.09699.pdf. 3, 23
- Derek Lim, Joshua David Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. In ICLR 2022 Workshop on Geometrical and Topological Representation Learning, 2022. URL https://openreview.net/ forum?id=BlM64by6gc.
- Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning. Advances in Neural Information Processing Systems, 33, 2020. URL https://proceedings.neurips.cc/paper/2020/file/ 2f73168bf3656f697507752ec592c437-Paper.pdf
- Fan RK Chung. Spectral Graph Theory. American Mathematical Society, 1997. URL https://www. bibsonomy.org/bibtex/295ef10b5a69a03d8507240b6cf410f8a/folke.
- Ulrike von Luxburg, Agnes Radl, and Matthias Hein. Hitting and commute times in large random neighborhood graphs. Journal of Machine Learning Research, 15(52):1751–1798, 2014. URL http: //jmlr.org/papers/v15/vonluxburg14a.html.
- Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. SIAM Journal on Computing, 40(6):1913–1926, 2011. doi: 10.1137/080734029. URL https://doi.org/10.1137/080734029.
- Huaijun Qiu and Edwin R. Hancock. Clustering and embedding using commute times. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(11):1873–1890, 2007. doi: 10.1109/TPAMI.2007.1103. URL https://ieeexplore.ieee.org/document/4302755.
- Vedat Levi Alev, Nima Anari, Lap Chi Lau, and Shayan Oveis Gharan. Graph Clustering using Effective Resistance. In 9th Innovations in Theoretical Computer Science Conference (ITCS 2018), volume 94, pages 1–16, 2018. doi: 10.4230/LIPIcs.ITCS.2018.41. URL http://drops.dagstuhl.de/opus/volltexte/2018/8369.
- Emmanuel Abbe. Community detection and stochastic block models: Recent developments. Journal of Machine Learning Research, 18(177):1–86, 2018. URL http://jmlr.org/papers/v18/16-480.html
- Thomas Bühler and Matthias Hein. Spectral clustering based on the graph p-laplacian. In Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, page 81–88, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553385. URL https://doi.org/10.1145/1553374.1553385. 7, 20
- Jian Kang and Hanghang Tong. N2n: Network derivative mining. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, page 861–870, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369763. doi: 10.1145/ 3357384.3357910. URL https://doi.org/10.1145/3357384.3357910. 7, 19
- Franco P Preparata and Michael I Shamos. Computational geometry: an introduction. Springer Science & Business Media, 2012. URL http://www.cs.kent.edu/~dragan/CG/CG-Book.pdf. 8
- Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019. 9
- Joshua Batson, Daniel A. Spielman, Nikhil Srivastava, and Shang-Hua Teng. Spectral sparsification of graphs: Theory and algorithms. Commun. ACM, 56(8):87–94, aug 2013. ISSN 0001-0782. doi: 10.1145/2492007.2492029. URL https://doi.org/10.1145/2492007.2492029.
- Morteza Alamgir and Ulrike Luxburg. Phase transition in the family of p-resistances. In Advances in Neural Information Processing Systems, 2011. URL https://proceedings.neurips.cc/paper/2011/file/ 07cdfd23373b17c6b337251c22b7ea57-Paper.pdf.
- Gregory Berkolaiko, James B Kennedy, Pavel Kurasov, and Delio Mugnolo. Edge connectivity and the spectral gap of combinatorial and quantum graphs. Journal of Physics A: Mathematical and Theoretical, 50(36):365201, 2017. URL https://doi.org/10.1088/1751-8121/aa8125.
- Zoran Stanic. Graphs with small spectral gap. ´ Electronic Journal of Linear Algebra, 26:28, 2013. URL https://journals.uwyo.edu/index.php/ela/article/view/1259. 20 [54] Douglas J Klein and Milan Randic. Resistance distance. ´ Journal of Mathematical Chemistry, 12(1):81–95, 1993. URL https://doi.org/10.1007/BF01164627. 24

ellis ALICANTE

Bibliography
Algorithmic Fairness

Rewiring Methods

- Jalali Z. S., et al. “On the information unfairness of social networks”. In SDM, 2020
- Masrour, F., et al. “Bursting the filter bubble: Fairness-aware network link prediction”. In AAAI, 2020.
- Li, P., et al. “On dyadic fairness: Exploring and mitigating bias in graph connections”. In ICLR, 2021.
- Spinelli, I., et al. “FairDrop: Biased edge dropout for enhancing fairness in GRL”. In TAI 2021
- Laclau, C., et al. “All of the Fairness for Edge Prediction with Optimal Transport”. In ICAIS, 2020.
- Kang, J. et al. “Inform: Individual fairness on graph mining”. In SIGKDD 2020. EDITS
- Dong, Y., et al. “EDITS: Modeling and mitigating data bias for graph neural networks”. In WWW, 2022.
- Loveland, Donald, et al. "FairEdit: Preserving Fairness in Graph Neural Networks through Greedy Graph Editing." preprint, 2022.

More Methods

- Maarten Buyl and Tijl De Bie. Debayes: a bayesian method fordebiasing network embeddings. In ICML, 2020
- Avishek Bose and William Hamilton. Compositional fairness constraints for graph embeddings. In ICML, 2019.
- Enyan Dai et al. Say no to the discrimination: Learning fair GNN with limited sensitive attribute information. In WSDM, 2021.
- Rahman, T., et al. Fairwalk: Towards fair graph embedding. In IJCAI, 2019.
- Khajehnejad, A., et al. Crosswalk: Fairness-enhanced node representation learning. In AAAI, 2022.
- Kleindessner, M., et al. “Guarantees for spectral clustering with fairness constraints”. In ICML, 2019.
- Shubham Gupta et al. Protecting individual interests across clusters: Spectral clustering with guarantees. Preprint, 2021.
- Ma j. Subgroup generalization and fairness of graph neural networks. In NeurIPS, 2021.
- Kang, J., et al. Rawlsgcn: Towards rawlsian difference principle on graph convolutional network. In WWW, 2022.
- Dong, Y., et al. GUIDE: Group Equality Informed Individual Fairness in Graph Neural Networks. In KDD 2022
- Dong, Y., et al. Individual fairness for graph neural networks: A ranking based approach. In SIGKDD, 2021.
- Agarwal, C., et al. Towards a unified framework for fair and stable graph representation learning. In UAI, 2021.
- Ma, J., et al. Learning fair node representations with graph counterfactual fairness. In ICWSDM. 2022.
- Zhang, X., et al. A Multi-view Confidence-calibrated Framework for Fair and Stable Graph Representation Learning. In ICDM, 2021.
- Kose, O.D. and Shen, Y. Fair Contrastive Learning on Graphs. In TSPN, 2022.

Topology Bias

- Masrour, F., et al. “Bursting the filter bubble: Fairness-aware network link prediction”. In AAAI, 2020.
- Jalali Z. S., et al. “On the information unfairness of social networks”. In SDM, 2020
- Fabbri, F., et al. “The effect of homophily on disparate visibility of minorities in people recommender systems”. In ICWSM, 2020.
- Zeng, Z., et al. “Fair Representation Learning for Heterogeneous Information Networks” In AAAI, 2021.
- Dong, Y., et al. “On Structural Explanation of Bias in Graph Neural Networks”. In KDD 2022.
- Dong, Y., et al. “EDITS: Modeling and mitigating data bias for graph neural networks”. In WWW, 2022.
- Loveland, D., et al. “On Graph Neural Network Fairness in the Presence of Heterophilous Neighborhoods”. Preprint, 2022.
- Jiang, Z, et al., “FMP: Toward Fair Graph Message Passing against Topology Bias”. Preprint, 2022.

Algorithmic Fairness and Network Science

- Barocas, S., et al. “Fairness in machine learning”. NeurIPS tutorial, 2017
- Dwork, C., et al. “Fairness through awareness”. Proceedings of the 3rd innovations in theoretical computer science conference, 2012.
- McPherson, M., et al. “Birds of a feather: Homophily in social networks”. Annual review of sociology 27, 2001.
- Hampson, M. “Smart Algorithm Bursts Social Networks' ‘Filter Bubbles’. 2011.
- Newman, M. “Assortative mixing in networks”. Phys. Rev. Lett., 89, 2002.

Surveys on Graph Fairness

- Dong, Y., et al. Fairness in Graph Mining: A Survey. Preprint, 2022.
- Choudhary, M., et al. A Survey on Fairness for Machine Learning on Graphs. Preprint, 2022.
- Zhang, W., et al. Fairness amidst non-iid graph data: A literature review. Preprint, 2022.
- Dai, Enyan, et al. A Comprehensive Survey on Trustworthy GNNs: Privacy, Robustness, Fairness, and Explainability. Preprint, 2022.
- Zhang, He, et al. Trustworthy Graph Neural Networks: Aspects, Methods and Trends. Preprint, 2022.

Tutorials

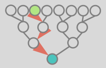
- Kang, J. et al. Algorithmic Fairness on Graphs: Methods and Trends. In KDD 2022.
- Venkatasubramanian, S., et al. Fairness in Networks: a tutorial. In KDD 2021.

Recommender Systems

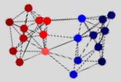
- Fabbri, F., et al. Rewiring What-to-Watch-Next Recommendations to Reduce Radicalization Pathways. In WWW, 2022.
- Fabbri, F., et al. The effect of homophily on disparate visibility of minorities in people recommender systems. In ICWSM, 2020.
- Wang, Y., et al. Streaming Algorithms for Diversity Maximization with Fairness Constraints. In ICDE, 2022.
- Fabbri, F., et al. Exposure Inequality in People Recommender Systems: The Long-Term Effects. In ICWSM, 2022.

Influence Maximization

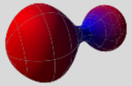
- Khajehnejad, M., et al. Adversarial Graph Embeddings for Fair Influence Maximization over Social Networks. In IJCAI, 2020.



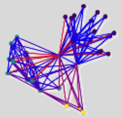
Motivation and Challenges



Introduction to Spectral Theory



Transductive Rewiring



Inductive Rewiring

Graph Fairness

Panel Discussion

Nuria Oliver

Petar Veličković

Marinka Zitnik

Francesco Fabbri

Francesco Di Giovanni

Panel

ellis

ALICANTE unit

The Institute for Humanity-Centric Artificial Intelligence

https://ellisalicante.org/tutorials/GraphRewiring

https://github.com/ellisalicante/GraphRewiring-Tutorial

Thanks

@arnaiztech
@Fescolano
@nuriaoliver
@ahmexbeg115

adrian@ellisalicante.org

GENERALITAT VALENCIANA
Conselleria de Innovación, Universidades, Ciencia y Sociedad Digital

EUROPEAN UNION
European Regional Development Fund

Sabadell Fundación

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES