# Adaptive Universal Generalized PageRank Graph Neural Network

Eli Chien[1] Jianhao Peng[1] Pan Li[2] Olgica Milenkovic[1]

ECE, University of Illinois Urbana-Champaign, USA[1]
CS, Purdue University, USA[2]


February 27, 2021

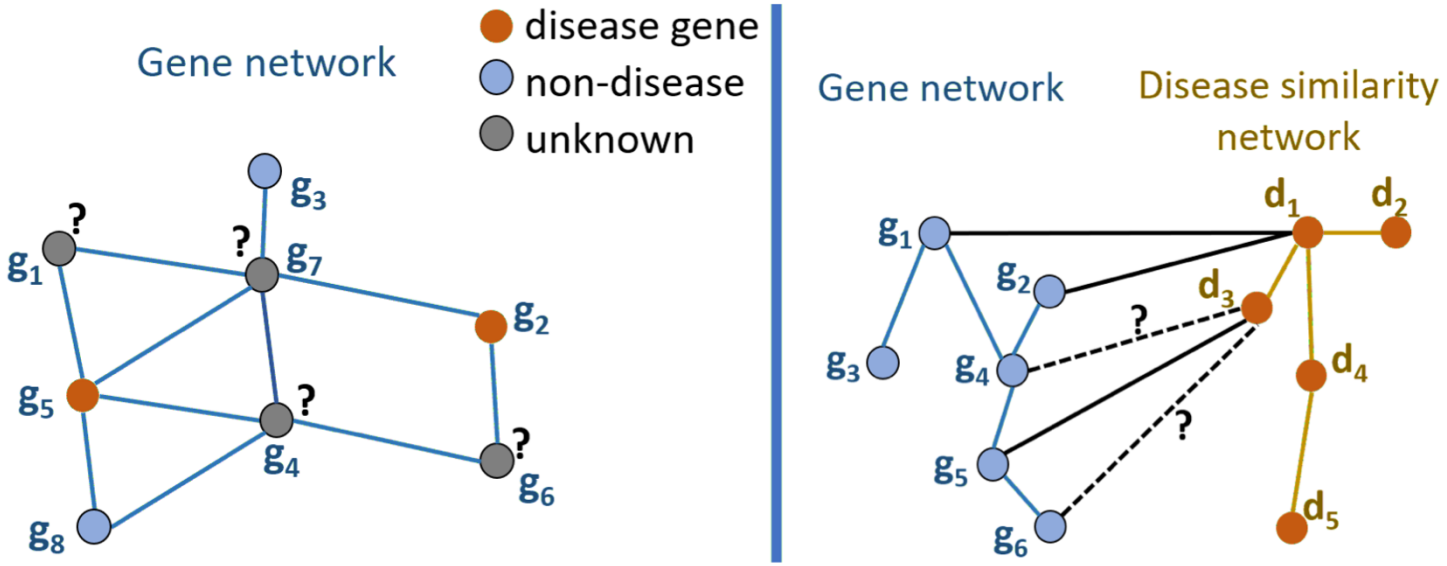
Our code can be found at :https://github.com/jianhao2016/GPRGNN

# Introduction

- Graph neural networks (GNNs) have shown great success in many graph related tasks.
  - Node classification, graph classification, link prediction... etc.
- Real world applications under the semi-supervised learning framework:
  - disease gene classification
  - drug repositioning

# Applications

- Gene disease asscoication, Ata, Sezin Kircali, et al. 2020

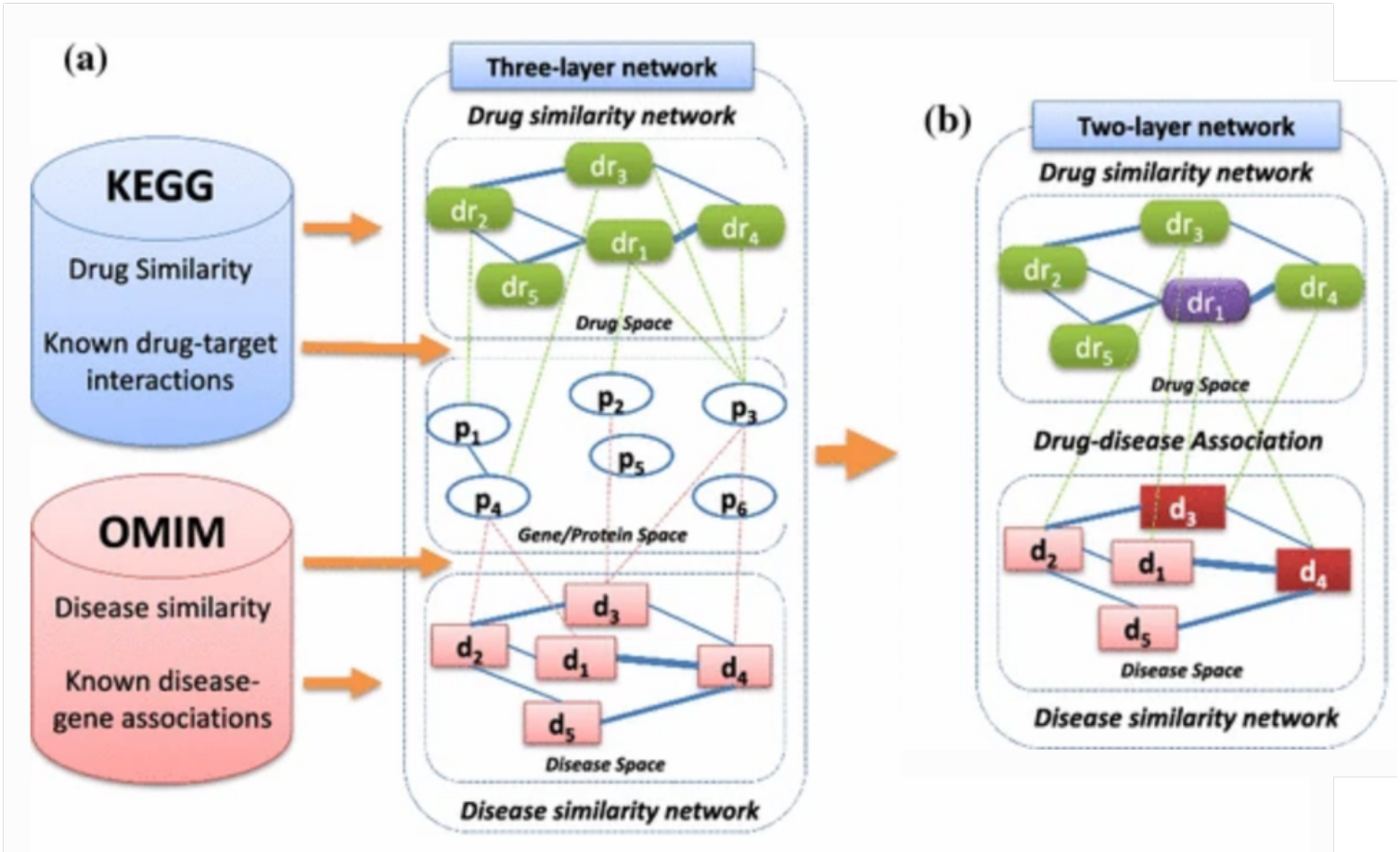



(a) Node classification.

(b) Link prediction.

# Applications

- Disease drug association Le, Duc-Hau, and Doanh Nguyen-Ngoc, 2018



# Outline

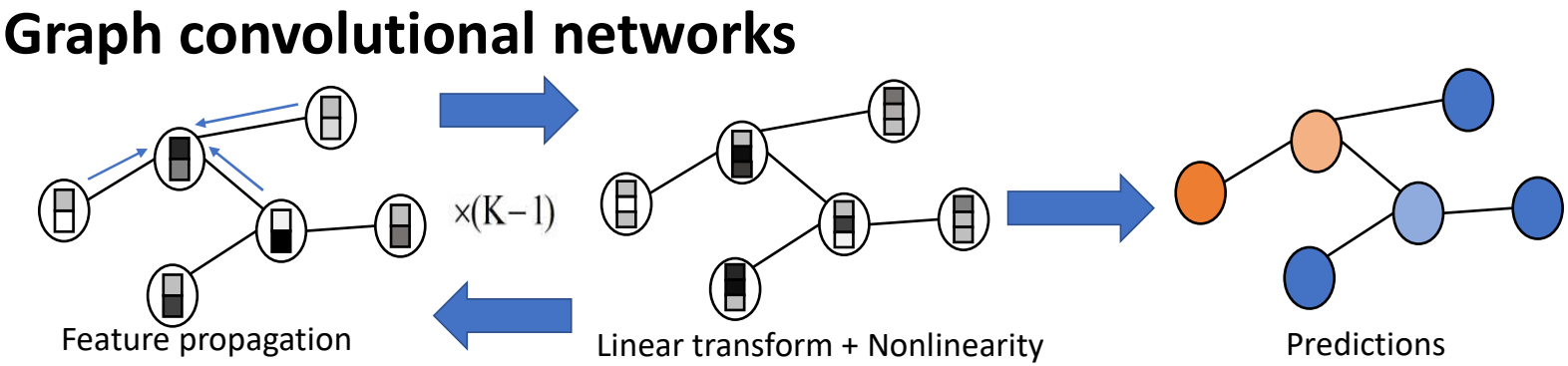
- Existing methodology for GNNs
- Fundamental weaknesses of existing GNNs: Universality and over-smoothing
- Our solution: GPR-GNN
- Experiments
- Conclusions and Future works

# General notations

- $A$: adjacency matrix of a graph $G = (V, E)$.
- $X \in \mathbb{R}^{|V| \times f}$: node features
- $Y \in \{0, 1\}^{|V| \times C}$: node labels, where $C$ is the number of classes.
- $\tilde{A} \triangleq I + A$ (adding self-loop)
- $\tilde{D}$: (diagonal) degree matrix of $\tilde{A}$
- $\tilde{A}_{\text{sym}} \triangleq \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$
- $\delta_{ij}$: Kronecker delta function

# Existing methodology: stacking GNN layers

- The most popular GNN: Graph Convolutional Networks (GCN) (Kipf et al. 2017)




$$\mathbf{H}_{\text{GCN}}^{(k)} = \text{ReLU}\left(\tilde{\mathbf{A}}_{\text{sym}}\mathbf{H}_{\text{GCN}}^{(k-1)}\mathbf{W}^{(k)}\right) \qquad \hat{\mathbf{P}}_{\text{GCN}} = \text{softmax}\left(\tilde{\mathbf{A}}_{\text{sym}}\mathbf{H}_{\text{GCN}}^{(K-1)}\mathbf{W}^{(k)}\right)$$

# Existing methodology: stacking GNN layers

- Inspired by GCN, there are tons of follow up works.
- Graph Attention Networks (GAT) (Velickovic et al. 2018), GraphSAGE (Hamilton et al. 2017) and many others.
- Most of these works define their own GNN layer and stack them.
- Common ground: stacking GNN layers and use the final layer as output.

# Fundamental weaknesses of existing GNNs

- However, most of the existing GNN architectures have two fundamental weaknesses.
  - Universality (homophily and heterophily)
  - Over-smoothing problem
- Universality issue restricts their learning ability on general graph-structured data.
- Over-smoothing problem prevents them being "deep models".

# Fundamental weaknesses of existing GNNs: Universality

- Most of the existing GNNs only work on homophilic (associative) graphs.
- Homophily principle: nodes from the same class tend to form edges.
- Counterexamples (heterophily/weak homophily):
  - Dating graphs: preferential connection with people of the opposite sex.
  - Protein structure: different classes of amino acids are more likely to connect.
- Design GNN that works in both cases: Universality.

# Fundamental weaknesses of existing GNNs: Over-smoothing

- Most of the existing GNNs fail to be "deep enough"
- Practical GNNs are mostly shallow (2-4 layers): better empirical performance than deep model.
- One theoretical reason: over-smoothing.

# Fundamental weaknesses of existing GNNs: Over-smoothing

- If one drop all non-linearity and stack infinitely many layers in GCN:

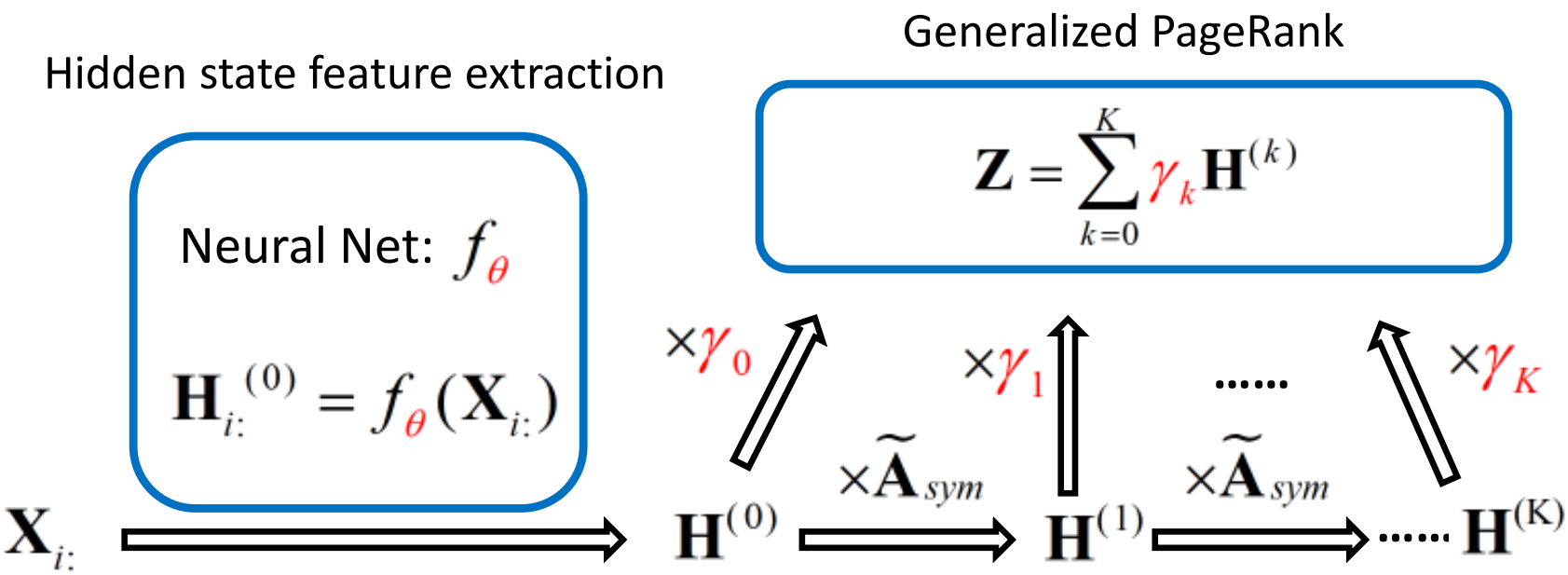
$$H_{\text{GCN}}^{(\infty)} = \tilde{A}_{\text{sym}}^{\infty} XW = v_1 v_1^T XW,$$

  where $v_1$ is the first eigenvector of $\tilde{A}_{\text{sym}}$.

- Each row of $H_{\text{GCN}}^{(\infty)}$ becomes independent of $X$ (up to a scalar).
- $\Rightarrow$ Lose feature information.

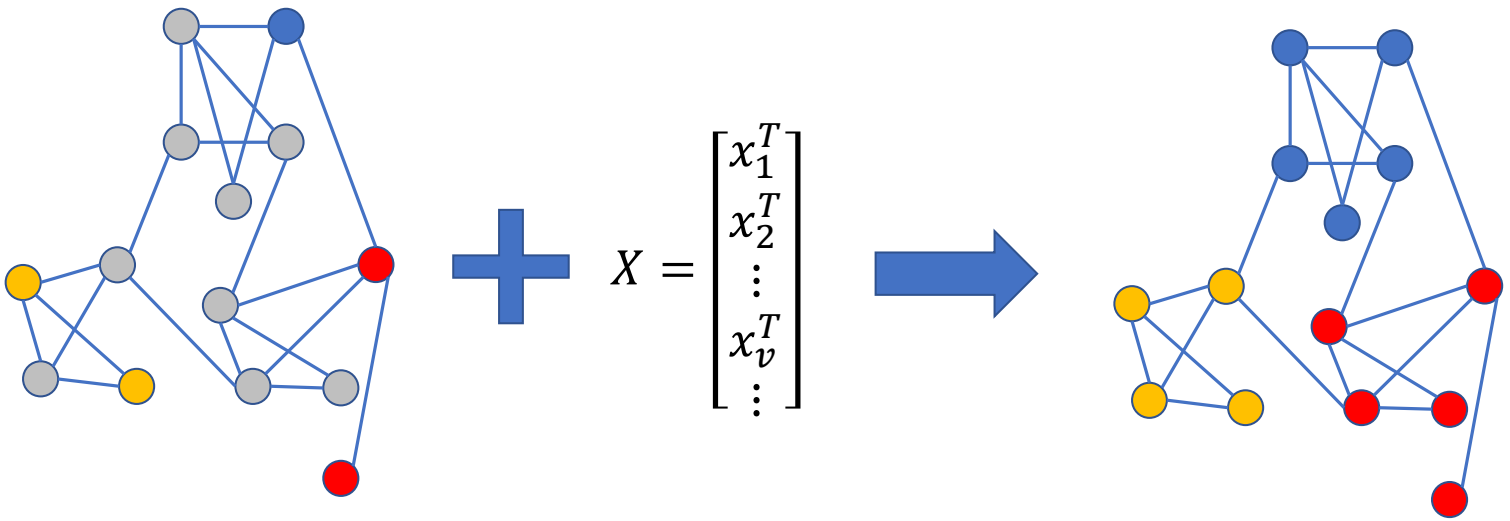
# Our solution: GPR-GNN

- We propose a succinct solution to resolve these fundamental weaknesses: Generalized PageRank GNN (GPR-GNN).
- GPR-GNN further prevents the over-fitting problem due to its succinctness.
- GPR-GNN also has interpretability on its learned weights.



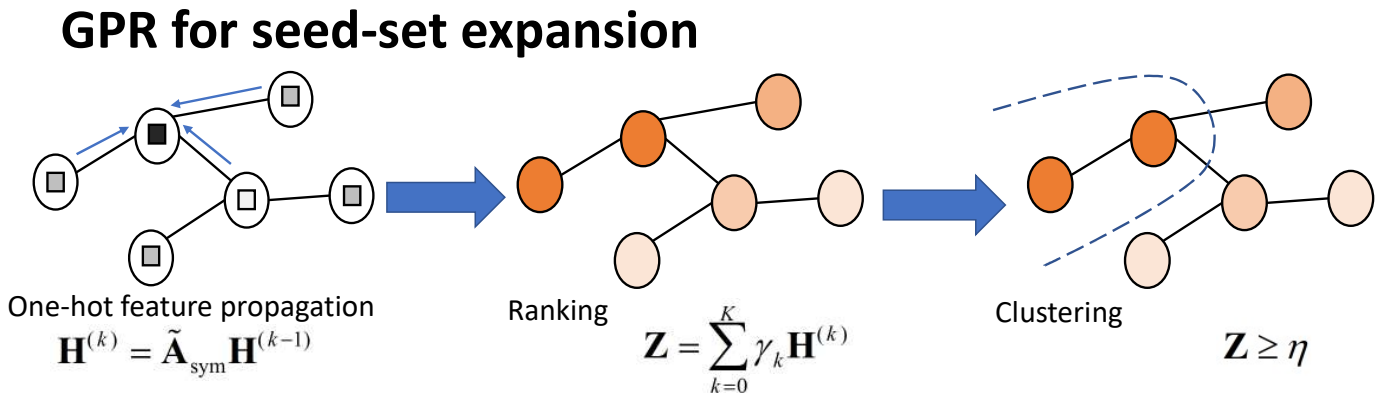
# Node classification

- In this talk, we focus on the node classification problem.
- Input data: adjacency matrix $A$, node features $X$ and partial observation of node labels $Y$.
- Goal: Predict the rest node labels.



# Genralized PageRank

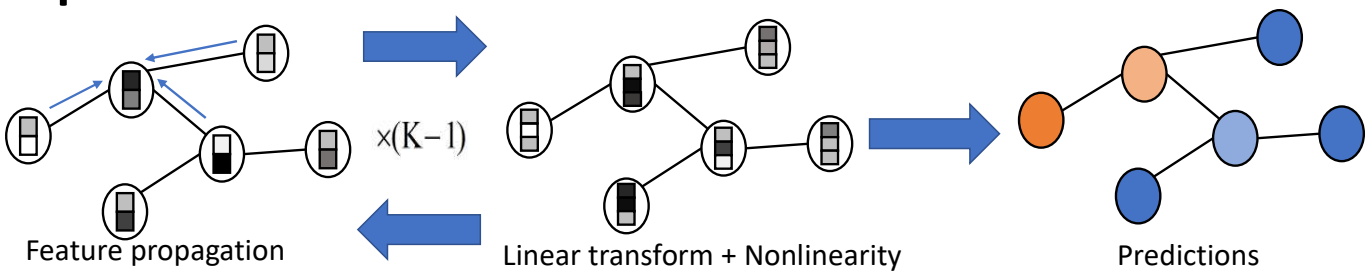
- GPR was used for unsupervised graph clustering, which is shown to significantly improve Personalized PageRank (PPR). (Li et al. 2019)
- Given a seed node $s \in V$, set $\mathrm{H}_{v:}^{(0)} = \delta_{vs}\ \forall v \in V\ (f = 1)$.
- $\gamma_k$ : GPR weights



# Relations between graph convolution and random walk

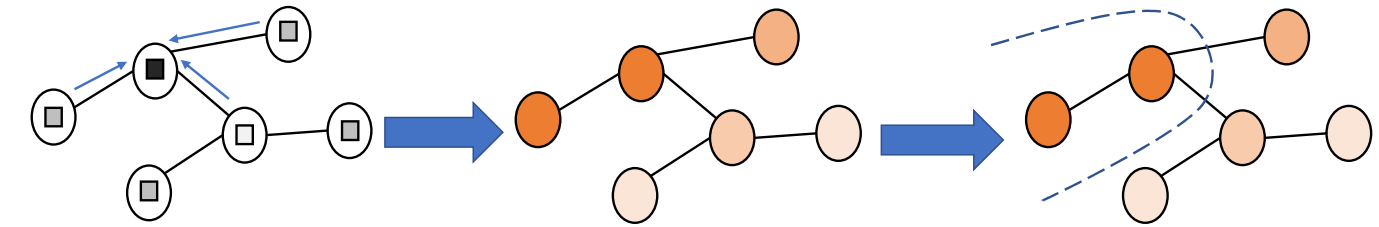
- One GCN layer: one step random walk (propagation) + some transformation.
- Relate to Generalized PageRanks!

**Graph convolutional networks**




$$\mathbf{H}_{\text{GCN}}^{(k)} = \text{ReLU}\left(\tilde{\mathbf{A}}_{\text{sym}}\mathbf{H}_{\text{GCN}}^{(k-1)}\mathbf{W}^{(k)}\right) \qquad \hat{\mathbf{P}}_{\text{GCN}} = \text{softmax}\left(\tilde{\mathbf{A}}_{\text{sym}}\mathbf{H}_{\text{GCN}}^{(K-1)}\mathbf{W}^{(k)}\right)$$

**GPR for seed-set expansion**




$$\mathbf{H}^{(k)} = \tilde{\mathbf{A}}_{\text{sym}}\mathbf{H}^{(k-1)} \qquad \mathbf{Z} = \sum_{k=0}^{K}\gamma_k\mathbf{H}^{(k)} \qquad \mathbf{Z} \geq \eta$$

# Why GPR-GNN is universal?

- Equivalence of GPR and polynomial graph filtering
- For $K \in \mathbb{N}$, $\sum_{k=0}^{K} \gamma_k \tilde{A}_{\text{sym}}^k$ is exactly the polynomial graph filter of order $K$!
- Learning optimal GPR weights $\Leftrightarrow$ optimal polynomial graph filter.
- From classical DSP and also GSP: polynomial filter can approximate arbitrary filter.

# Why GPR-GNN is universal?

- Define $\sum_{k=0}^{K} \gamma_k \tilde{A}_{\text{sym}}^k = U g_{\gamma,K}(\Lambda) U^T$, where $g_{\gamma,K}(\Lambda)$ is applied element-wise and $g_{\gamma,K}(\lambda) = \sum_{k=0}^{K} \gamma_k \lambda^k$.

**Theorem (Informal)**

*Assume that the graph $G$ is connected. If $\gamma_k \geq 0 \; \forall k \in \{0, 1, ..., K\}$, $\sum_{k=0}^{K} \gamma_k = 1$ and $\exists k' > 0$ such that $\gamma_{k'} > 0$, then $g_{\gamma,K}(\cdot)$ is a low-pass graph filter. Also, if $\gamma_k = (-\alpha)^k, \alpha \in (0, 1)$ and $K$ is large enough, then $g_{\gamma,K}(\cdot)$ is a high-pass graph filter.*

- To be universal: need some $\gamma_k$ to be negative.

# Why GPR-GNN resolves over-smoothing

- Key idea: If the $k^{\text{th}}$ step $H^{(k)}$ is not helpful (increase training loss)$\Rightarrow$ the magnitude of $\gamma_k$ should decay (done by the optimizer).

## Theorem (Informal)

*Assume the graph $G$ is connected and the training set contains nodes from each of the classes. Also assume that $k'$ is large enough so that the over-smoothing effect occurs for $\mathrm{H}^{(k)}, \forall k \geq k'$ which dominate the contribution to the final output $\mathrm{Z}$. Then, the gradients of $\gamma_k$ and $\gamma_k$ are identical in sign for all $k \geq k'$.*

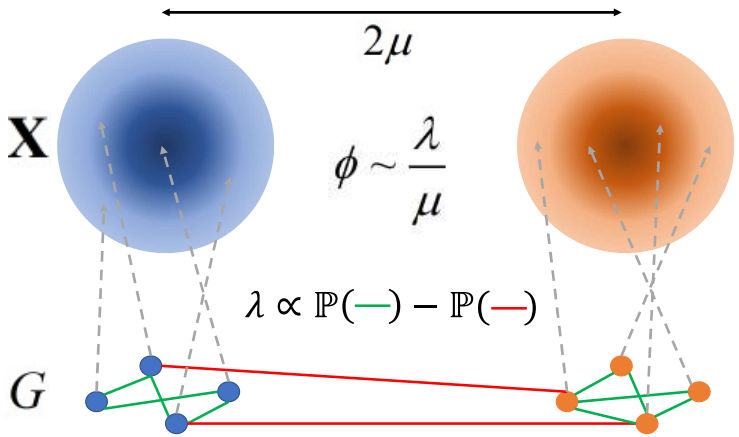
# Comparison to related works

- GCN-like models:
  - JK-Net: make use of different steps in final layers.
  - GCN-Cheby: use more then one steps in a GNN layers.
  - The depth of these models is limited in practice, no interpretability in learned weights.
- Graph augamented MLP:
  - APPNP: fix $\gamma_k = (1-\alpha)\alpha^k$ (PPR weights) $\Rightarrow$ Low pass filter.
  - SGC: fix $\gamma_k = \delta_{kK}$ and drop all non-linearity $\Rightarrow$ Low pass filter.
  - C&S frameworks[1]
  - Only works for homophilic graphs.

[1] Huang et al. Combining Label Propagation and Simple Models out-performs Graph Neural Networks. ICLR 2021.

# Experiments: Synthetic data

- To test GNNs on graphs with arbitrary levels of homophily and heterophily, we propose to use contextual Stochastic Block Model (cSBM) (Deshpande et al., 2018).
- Node features X: Gaussian vectors; Graph $G$: from SBM.
- $\phi \in [-1, 1]$:
  - $\phi = 1$: Strong homophily, node features are not informative.
  - $\phi = 0$: Graph topology is not informative.
  - $\phi = -1$: Strong heterophily, node features are not informative.



# Experiments: Synthetic data

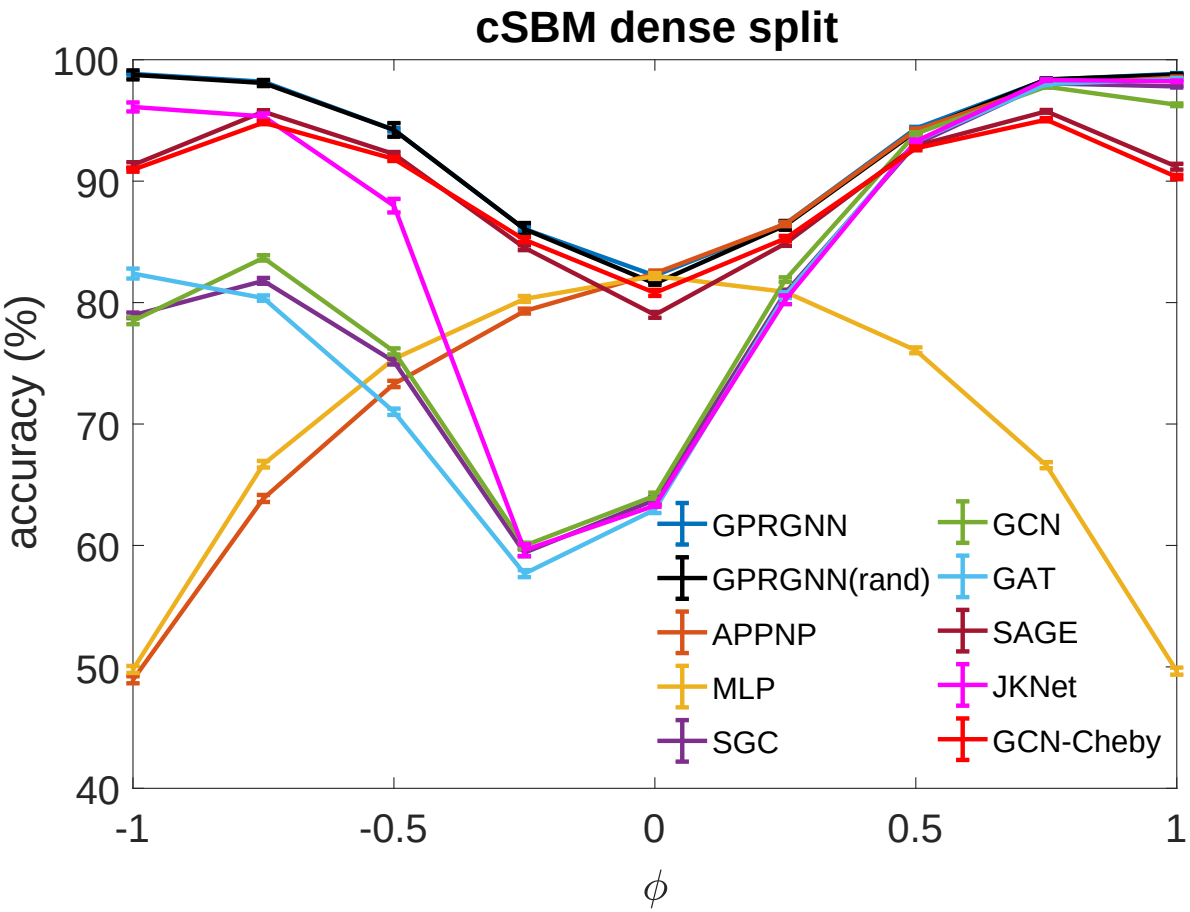



Figure: Accuracy of tested models on cSBM. Error bars indicate 95% confidence interval.

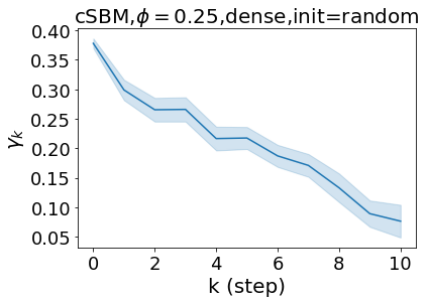
# Experiments: real world datasets

Table 2: Results on real world benchmark datasets: Mean accuracy (%) $\pm$ 95% confidence interval. Boldface letters are used to mark the best results while underlined boldface letters indicate results within the given confidence interval of the best result.

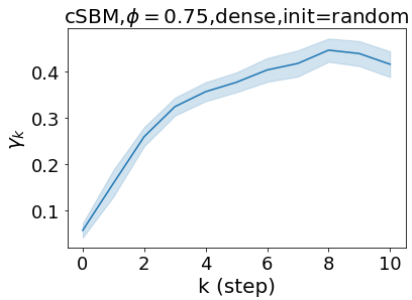
| | Cora | Citeseer | PubMed | Computers | Photo | Chameleon | Actor | Squirrel | Texas | Cornell |
|---|---|---|---|---|---|---|---|---|---|---|
| GPRGNN | **79.51±0.36** | 67.63±0.38 | **85.07±0.09** | **82.90±0.37** | **91.93±0.26** | **67.48±0.40** | **39.30±0.27** | **49.93±0.53** | **92.92±0.61** | **91.36±0.70** |
| APPNP | **79.41±0.38** | **68.59±0.30** | **85.02±0.09** | 81.99±0.26 | 91.11±0.26 | 51.91±0.56 | 38.86±0.24 | 34.77±0.34 | 91.18±0.70 | **91.80±0.63** |
| MLP | 50.34±0.48 | 52.88±0.51 | 80.57±0.12 | 70.48±0.28 | 78.69±0.30 | 46.72±0.46 | 38.58±0.25 | 31.28±0.27 | 92.26±0.71 | **91.36±0.70** |
| SGC | 70.81±0.67 | 58.98±0.47 | 82.09±0.11 | 76.27±0.36 | 83.80±0.46 | 63.02±0.43 | 29.39±0.20 | 43.14±0.28 | 55.18±1.17 | 47.80±1.50 |
| GCN | 75.21±0.38 | 67.30±0.35 | 84.27±0.01 | 82.52±0.32 | 90.54±0.21 | 60.96±0.78 | 30.59±0.23 | 45.66±0.39 | 75.16±0.96 | 66.72±1.37 |
| GAT | 76.70±0.42 | 67.20±0.46 | 83.28±0.12 | 81.95±0.38 | 90.09±0.27 | 63.9±0.46 | 35.98±0.23 | 42.72±0.33 | 78.87±0.86 | 76.00±1.01 |
| SAGE | 70.89±0.54 | 61.52±0.44 | 81.30±0.10 | **83.11±0.23** | 90.51±0.25 | 62.15±0.42 | 36.37±0.21 | 41.26±0.26 | 79.03±1.20 | 71.41±1.24 |
| JKNet | 73.22±0.64 | 60.85±0.76 | 82.91±0.11 | 77.80±0.97 | 87.70±0.70 | 62.92±0.49 | 33.41±0.25 | 44.72±0.48 | 75.53±1.16 | 66.73±1.73 |
| GCN-Cheby | 71.39±0.51 | 65.67±0.38 | 83.83±0.12 | 82.41±0.28 | 90.09±0.28 | 59.96±0.51 | 38.02±0.23 | 40.67±0.31 | 86.08±0.96 | 85.33±1.04 |
| GeomGCN | 20.37±1.13 | 20.30±0.90 | 58.20±1.23 | NA | NA | 61.06±0.49 | 31.81±0.24 | 38.28±0.27 | 58.56±1.77 | 55.59±1.59 |

Homophily | Heterophily

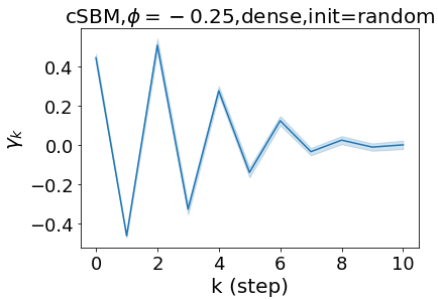
# Experiments: interpretability of learnt GPR weights



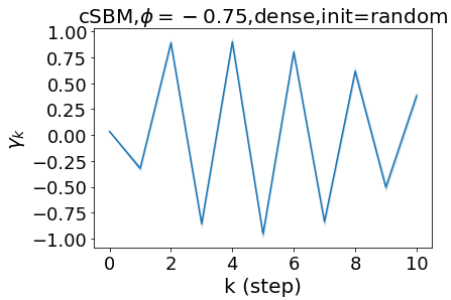

(a) $\phi = 0.25$



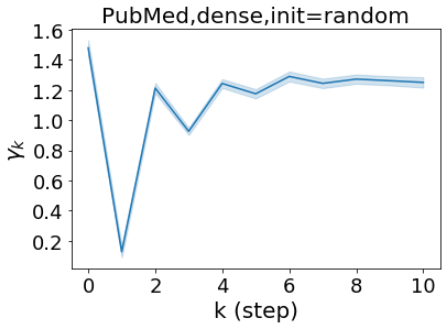

(b) $\phi = 0.75$



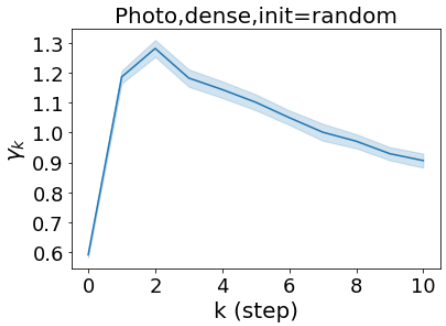

(c) $\phi = -0.25$



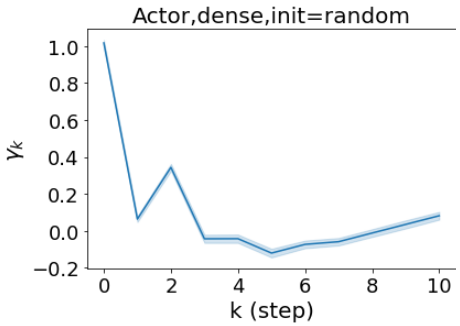

(d) $\phi = -0.75$



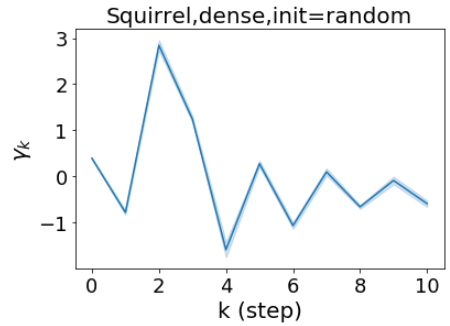

(e) PubMed




(f) Photo



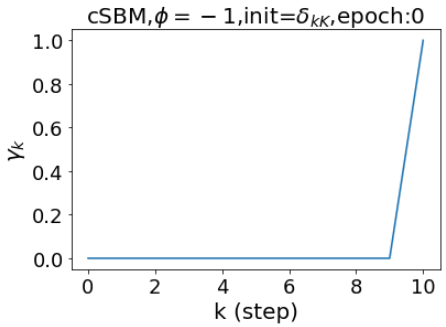

(g) Actor



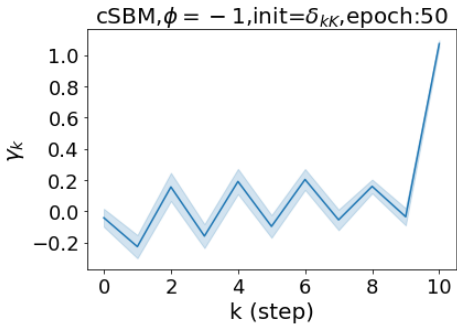

(h) Squirrel

- These match our theorem!

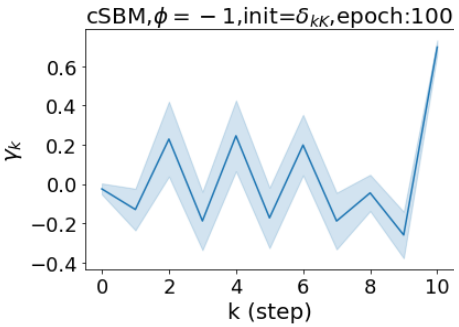
# Experiments: prevent over-smoothing



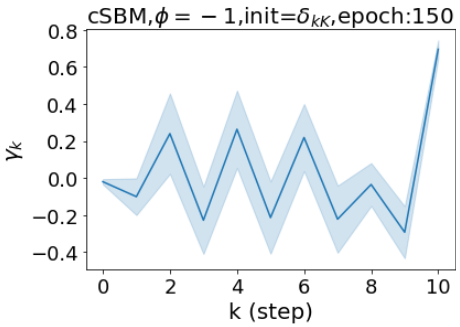

(i) Epoch= 0




(j) Epoch= 50




(k) Epoch= 100




(l) Epoch= 150

- We force GPR-GNN start with over-smoothing with high probability. (96 out of 100 runs)
- Initial accuracy at epoch 0: 50.07%; Final accuracy: 98.79%.
- GPR-GNN indeed can prevent over-smoothing!

# Conclusions and Future works

- GPR-GNN :
  - is universal
  - prevents over-smoothing
  - has small number of parameters $\Rightarrow$ prevent over-fitting
  - has interpretability
- Propose cSBM to test universality rigorously.
- Potential future direction:
  - MLP$\rightarrow$ ??
  - Use attention to learn GPR weights?
  - Extend to graph representation learning? (combine with some Graph Pooling??)

Thanks for your attention! Questions?

# Extra slides: Graph signal processing (GSP)

- Eigenvalue decomposition: $\tilde{A}_{\text{sym}} = U\Lambda U^T$.
- Eigenvalues: (low frequency) $1 = \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n > -1$ (high frequency).
- Low frequency $\Leftrightarrow$ Homophily.
- High frequency $\Leftrightarrow$ Heterophily.

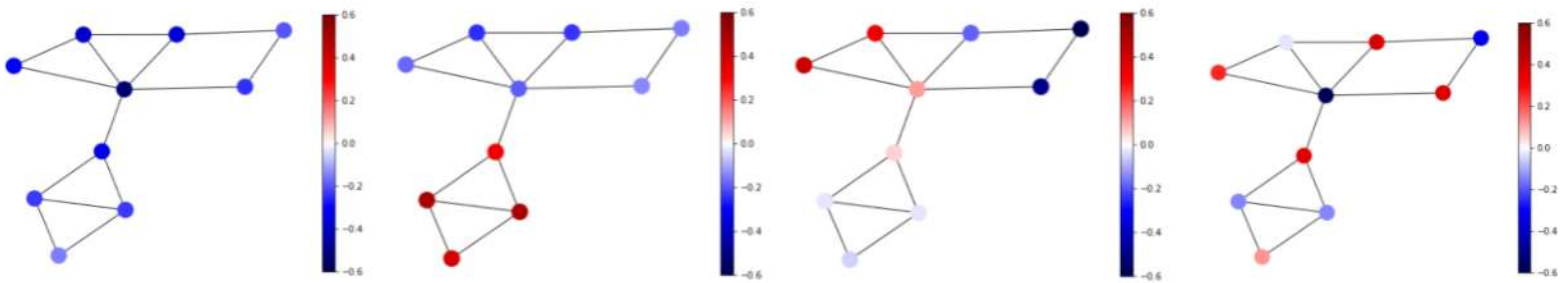


**Fig. 1**. The values of entries of different eigenvectors for $A$. From left to right are the first to the forth eigenvector correspond to the first to the forth largest eigenvalues respectively.

# Extra slides: Genralized PageRank

- Different choices of GPR weights lead to different PageRank methods.
- Some specific examples:
  1. standard PR: $\gamma_k = (1-\alpha)\alpha^k$, and some $\alpha \in [0, 1)$ (Page 1999).
  2. personalized PR: a special case of standard PR where $\mathbf{H}^{(0)}$ has a few non-zero entries (i.e. $\delta_{:s}$) (Andersen et al. 2006).
  3. heat-kernel PR: $\gamma_k = h^k \frac{e^{-h}}{k!}$, and some $h \geq 0$ (Chung 2007).
  4. totalRank: $\gamma_k = \frac{1}{k+1} - \frac{1}{k+2}$ (Boldi 2005).

# Extra slides: Genralized PageRank

- In (Li et al. 2019), we **theoretically** analyzed a special form of GPR, Inverse PR (IPR), for homophilic graphs.
- Key insights of (Li et al. 2019):
  - Including large-step random walks is beneficial for clustering.
  - If one view $x = [H_v^{(0)}, H_v^{(1)} \cdots, H_v^{(K)}]$ as "node features" for each node $v \in V$, then GPR is a linear classifier in this feature space.
  - In two blocks stochastic block model (SBM):