p-Laplacian Based Graph Neural Networks

Guoji Fu, Peilin Zhao, Yatao Bian
Tencent AI Lab

Outline

1. **Introduction**
   - Graph Neural Networks
   - Motivation

2. **Preliminary**

3. **Methodology**
   - p-Laplacian Regularization Framework
   - p-Laplacian Message Passing & ^{p}GNN Architecture

4. **Spectral Analysis**

5. **Empirical Results**

Introduction — Graph Neural Networks

Our focus: Graph neural networks (GNNs) & **Semi-supervised node classification.**

Introduction — Graph Neural Networks

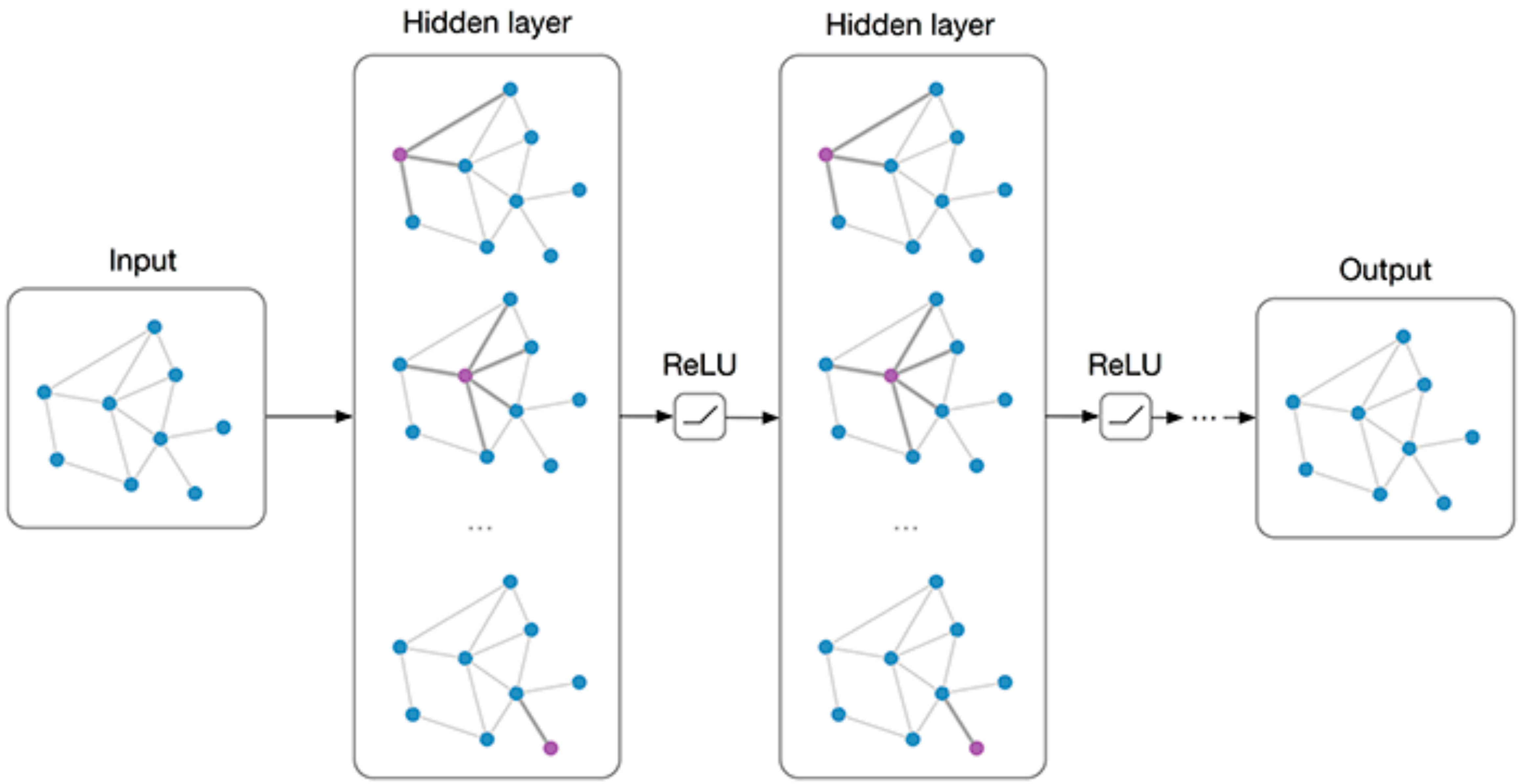
Our focus: Graph neural networks (GNNs) & **Semi-supervised node classification.**

Input: A graph G with node features $\mathbf{X}$, the adjacency matrix $\mathbf{A}$, labels of some nodes.
Output: The predicted labels of the rest of nodes.

Introduction — Graph Neural Networks

Our focus: Graph neural networks (GNNs) & **Semi-supervised node classification.**

Input: A graph G with node features $\mathbf{X}$, the adjacency matrix $\mathbf{A}$, labels of some nodes.
Output: The predicted labels of the rest of nodes.

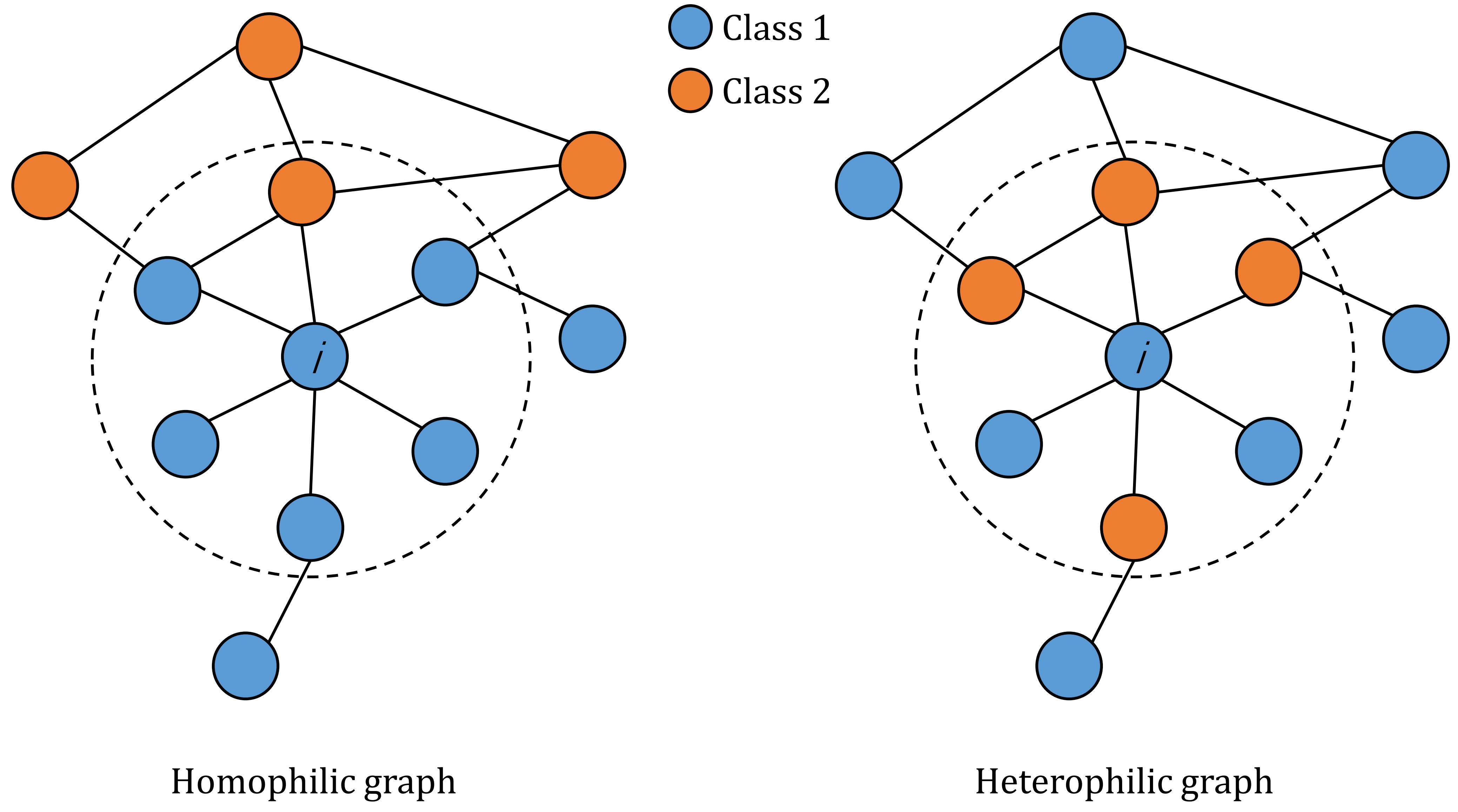


(The figure was collected from https://tkipf.github.io/graph-convolutional-networks/)

Introduction — Motivation

Homophily Assumption of GNNs:

The labels of linked nodes in a graph are the same or consistent.

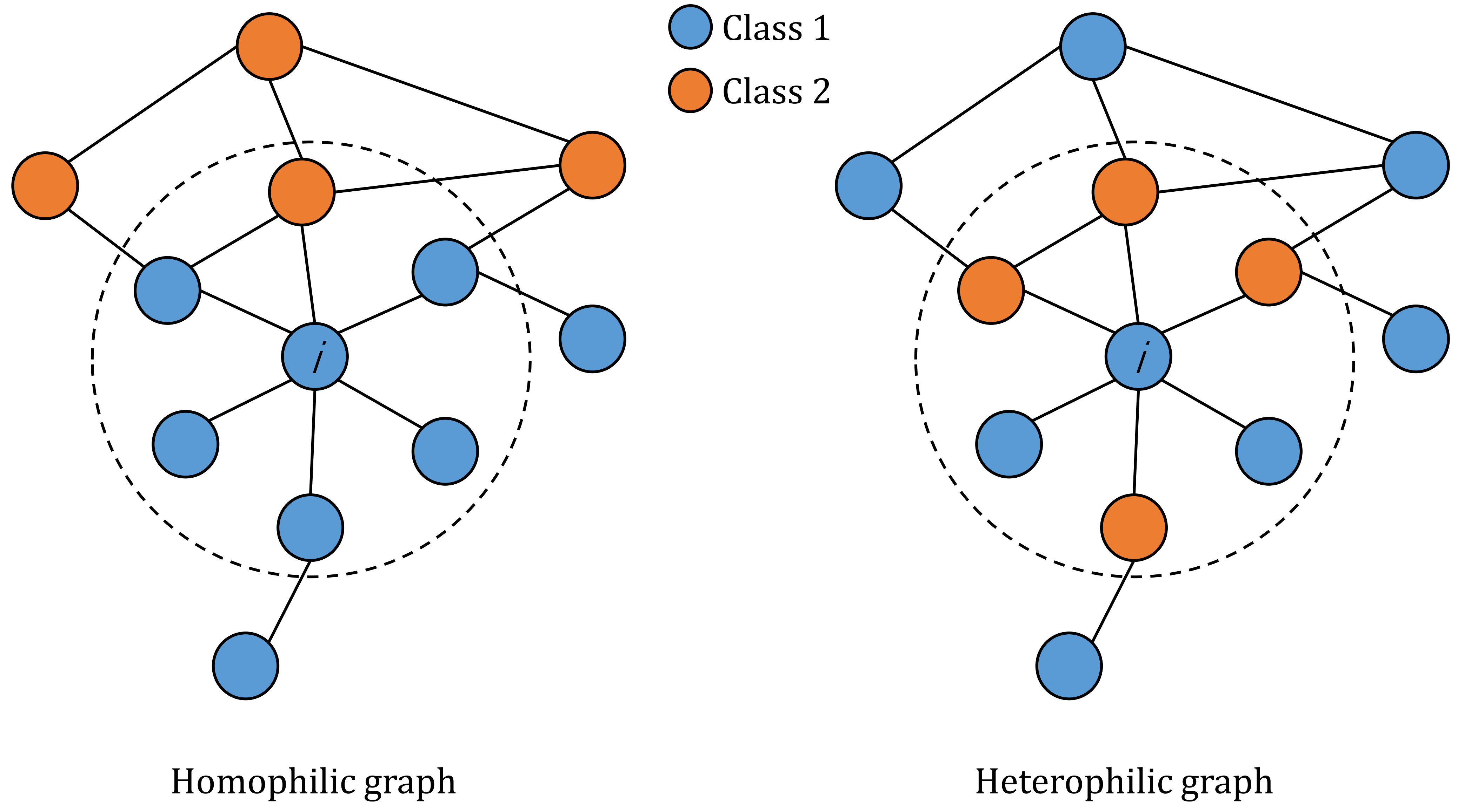


Introduction — Motivation

Homophily Assumption of GNNs:

The labels of linked nodes in a graph are the same or consistent.

Problem: The assumption does not hold for heterophilic graphs!



Introduction — Motivation

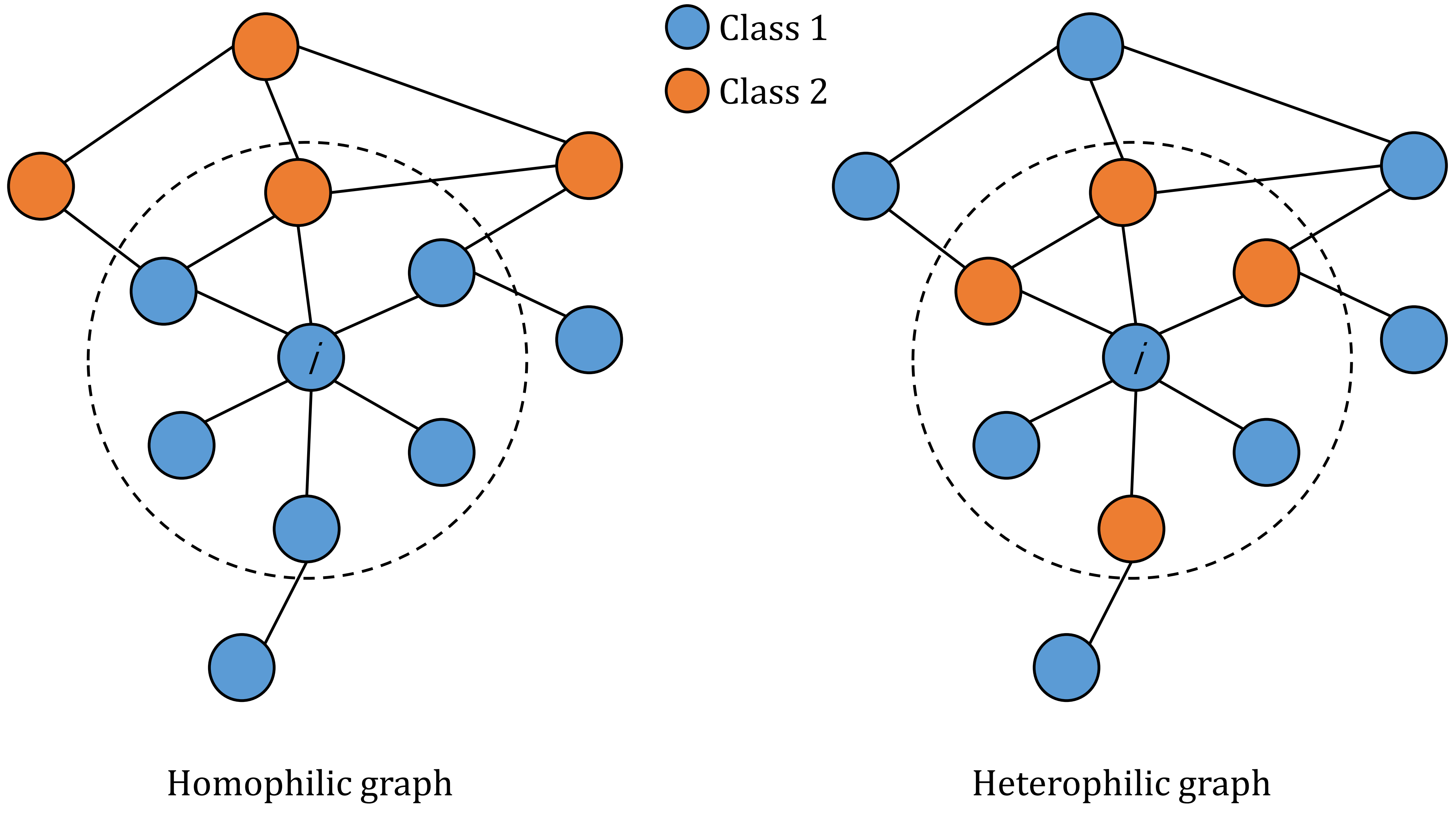
Homophily Assumption of GNNs:

The labels of linked nodes in a graph are the same or consistent.

Problem: The assumption does not hold for heterophilic graphs!

Low-Pass Filter Type GNNs

Some GNNs (e.g., GCN, SGC, APPNP) work as low-pass filters (The spectral range of Laplacian is [0, 2]).



Introduction — Motivation

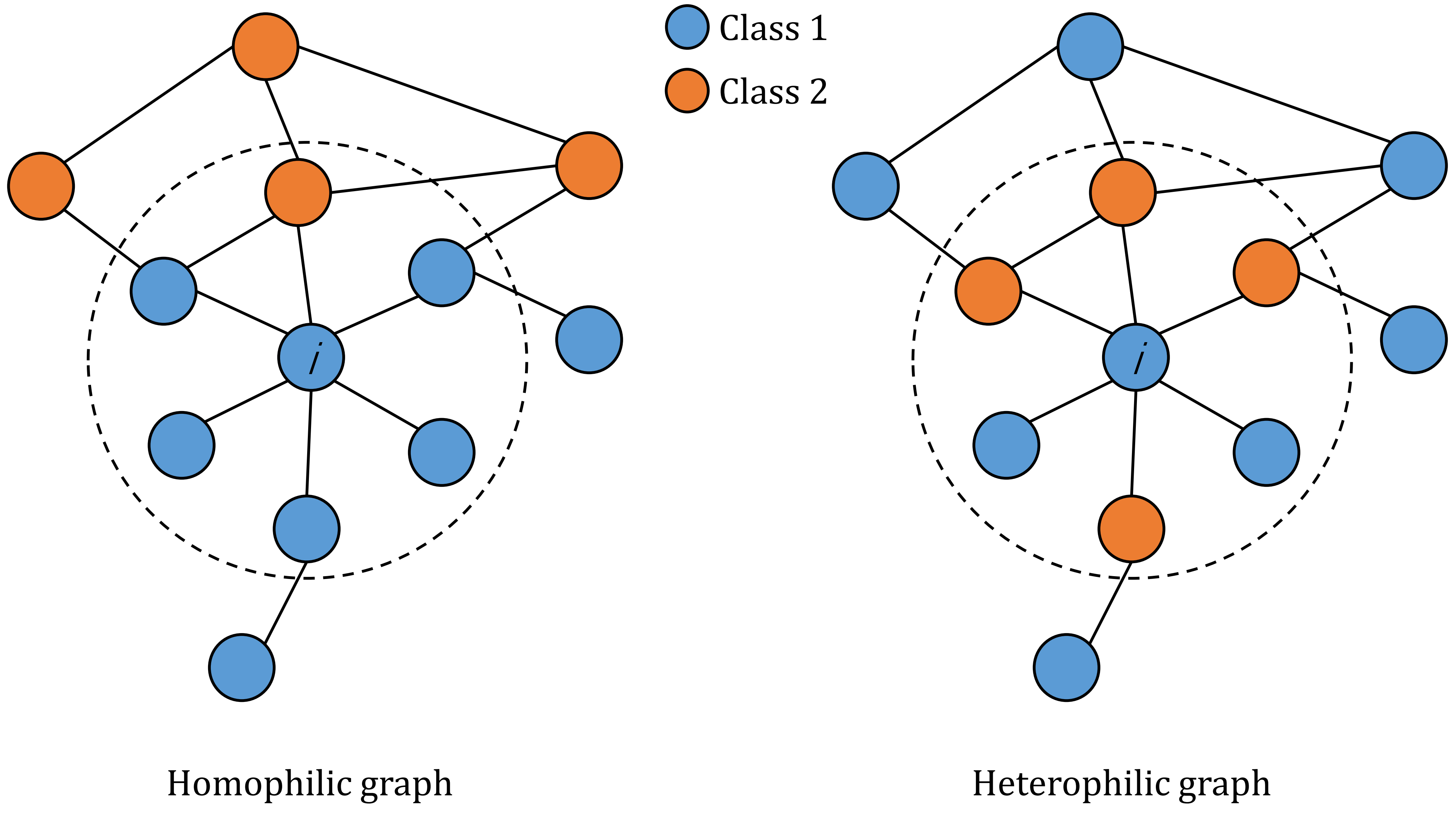
Homophily Assumption of GNNs:

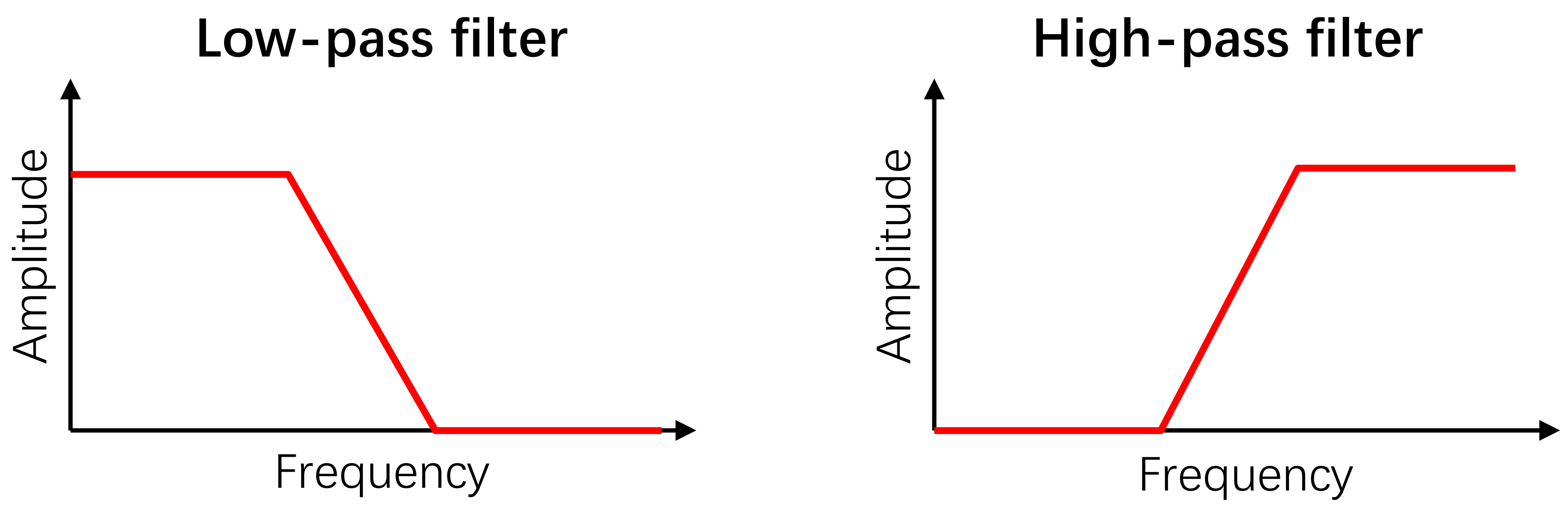
The labels of linked nodes in a graph are the same or consistent.

Problem: The assumption does not hold for heterophilic graphs!

Low-Pass Filter Type GNNs

Some GNNs (e.g., GCN, SGC, APPNP) work as low-pass filters (The spectral range of Laplacian is [0, 2]).





Introduction — Motivation

Homophily Assumption of GNNs:

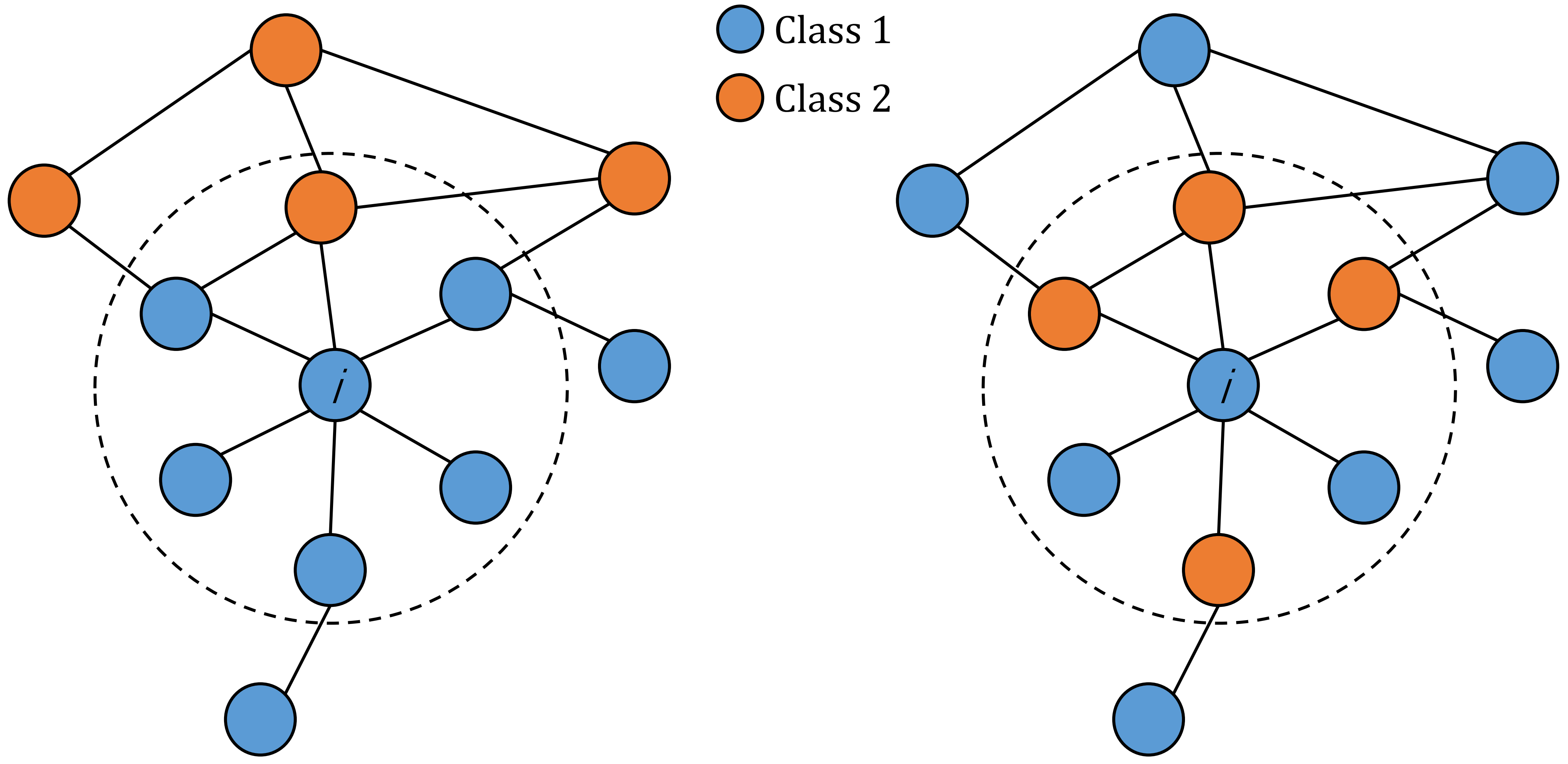
The labels of linked nodes in a graph are the same or consistent.

Problem: The assumption does not hold for heterophilic graphs!

Low-Pass Filter Type GNNs

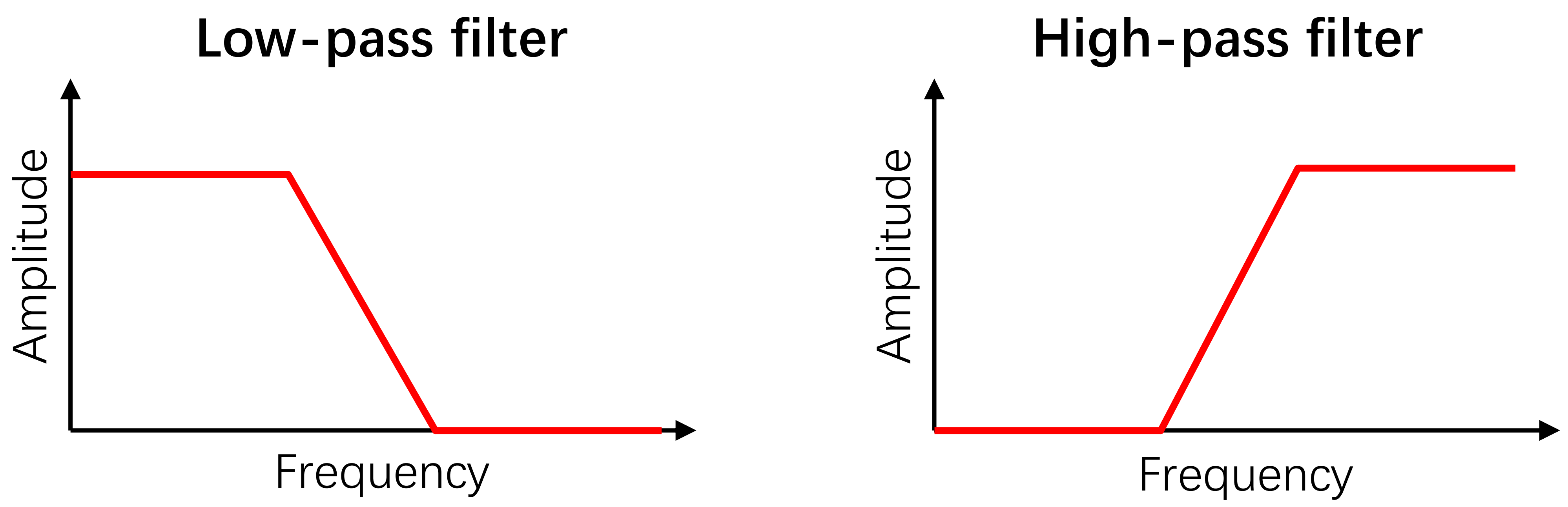
Some GNNs (e.g., GCN, SGC, APPNP) work as low-pass filters (The spectral range of Laplacian is [0, 2]).

Most GNNs fail on heterophilic graphs!



Homophilic graph

Heterophilic graph



Introduction — Motivation

Homophily Assumption of GNNs:

The labels of linked nodes in a graph are the same or consistent.

Problem: The assumption does not hold for heterophilic graphs!

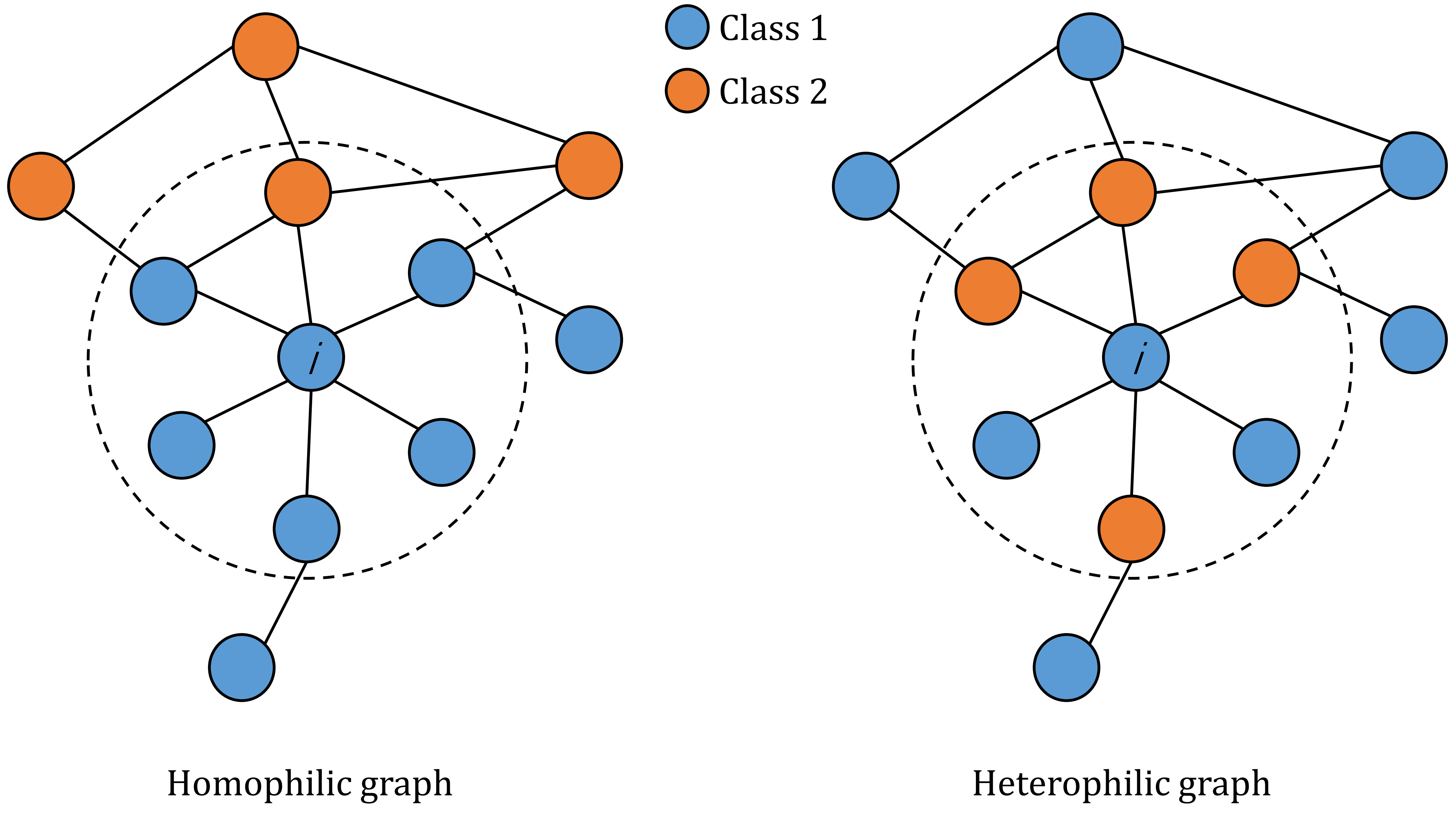
Low-Pass Filter Type GNNs

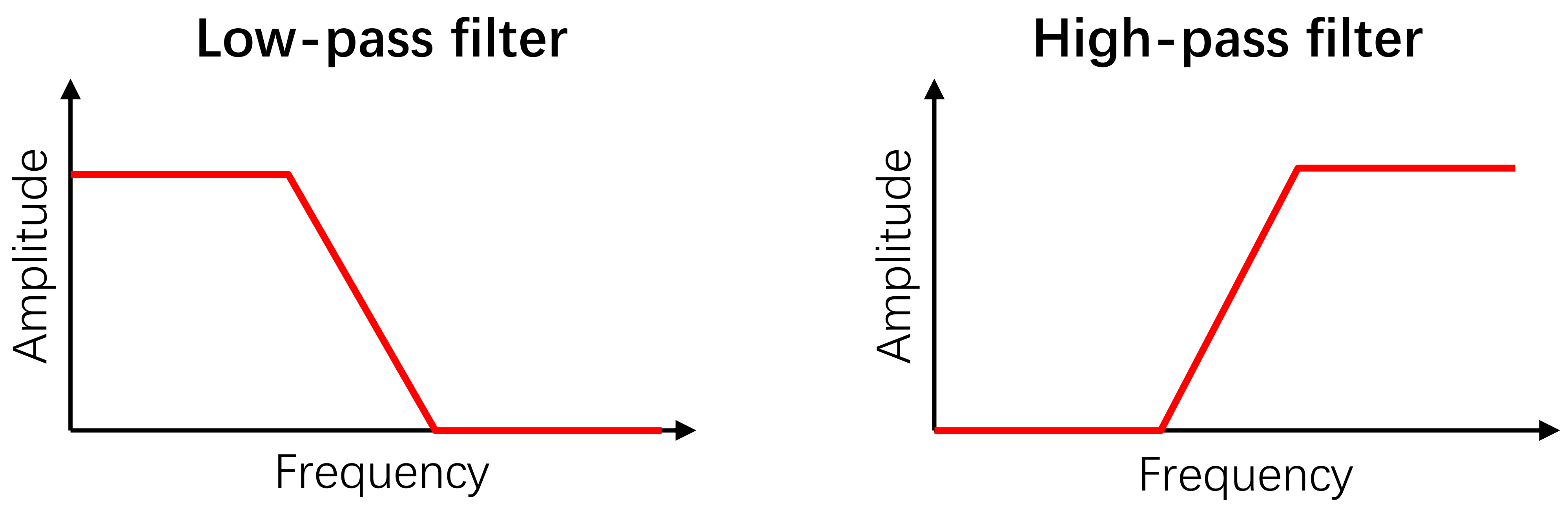
Some GNNs (e.g., GCN, SGC, APPNP) work as low-pass filters (The spectral range of Laplacian is [0, 2]).

Most GNNs fail on heterophilic graphs!

Idea: We could design new message passing schemes & GNN architectures based on p-Laplacian:

- $p \geq 2, 0 \leq \lambda \leq 2^{p-1}$;
- $1 < p < 2, 0 \leq \lambda \leq 2^{p-1}\sqrt{N_k}$;
- $p = 1, 0 \leq \lambda \leq \sqrt{N_{\min}}$.





Preliminary

Local variation

Graph gradient

$(\nabla f)([i,j])$ measures the variation of f on **edge** $[i,j]$:

$$(\nabla f)([i,j]) := \sqrt{\frac{W_{i,j}}{D_{j,j}}} f(j) - \sqrt{\frac{W_{i,j}}{D_{i,i}}} f(i)$$

Graph divergence

$(\mathrm{div} g)(i)$ measures the net outflow of g on **node** i:

$$(\mathrm{div} g)(i) = \sum_{j=1}^{N} \sqrt{\frac{W_{i,j}}{D_{i,i}}} \left(g([i,j]) - g([j,i])\right)$$

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$: A graph

$f : \mathcal{V} \rightarrow \mathbb{R}$: A function defined on $\mathcal{V}$

$g : \mathcal{E} \rightarrow \mathbb{R}$: A function defined on $\mathcal{E}$

$\mathbf{W}$: The adjacency matrix

$\mathbf{D}$: The degree matrix

Preliminary

Tencent AI Lab · ICML International Conference On Machine Learning

Local variation

Graph gradient

$(\nabla f)([i,j])$ measures the variation of f on **edge** $[i,j]$:

$$(\nabla f)([i,j]) := \sqrt{\frac{W_{i,j}}{D_{j,j}}} f(j) - \sqrt{\frac{W_{i,j}}{D_{i,i}}} f(i)$$

Graph divergence

$(\mathrm{div} g)(i)$ measures the net outflow of g on **node** i:

$$(\mathrm{div} g)(i) = \sum_{j=1}^{N} \sqrt{\frac{W_{i,j}}{D_{i,i}}} \left(g([i,j]) - g([j,i])\right)$$

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$: A graph

$f : \mathcal{V} \to \mathbb{R}$: A function defined on $\mathcal{V}$

$g : \mathcal{E} \to \mathbb{R}$: A function defined on $\mathcal{E}$

$\mathbf{W}$: The adjacency matrix

$\mathbf{D}$: The degree matrix

Graph p-Laplacian

$\Delta_p : \mathcal{F}_{\mathcal{V}} \to \mathcal{F}_{\mathcal{V}}$ is an operator defined by:

$$\Delta_p f := -\frac{1}{2} \mathrm{div}(\|\nabla f\|^{p-2} \nabla f)$$

When $p = 2$, $\Delta_2 = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$.

Preliminary

Tencent AI Lab ICML International Conference On Machine Learning

Local variation

Graph gradient

$(\nabla f)([i,j])$ measures the variation of f on **edge** $[i,j]$:

$$(\nabla f)([i,j]) := \sqrt{\frac{W_{i,j}}{D_{j,j}}} f(j) - \sqrt{\frac{W_{i,j}}{D_{i,i}}} f(i)$$

Graph divergence

$(\mathrm{div} g)(i)$ measures the net outflow of g on **node** i:

$$(\mathrm{div} g)(i) = \sum_{j=1}^{N} \sqrt{\frac{W_{i,j}}{D_{i,i}}} \left(g([i,j]) - g([j,i])\right)$$

Global variation

$\mathcal{S}_p(f)$ measures the overall variation of f on **graph** $\mathcal{G}$:

$$\mathcal{S}_p(f) := \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \|(\nabla f)([i,j])\|^p = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \left\| \sqrt{\frac{W_{i,j}}{D_{j,j}}} f(j) - \sqrt{\frac{W_{i,j}}{D_{i,i}}} f(i) \right\|^p$$

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$: A graph

$f : \mathcal{V} \to \mathbb{R}$: A function defined on $\mathcal{V}$

$g : \mathcal{E} \to \mathbb{R}$: A function defined on $\mathcal{E}$

$\mathbf{W}$: The adjacency matrix

$\mathbf{D}$: The degree matrix

Graph p-Laplacian

$\Delta_p : \mathcal{F}_{\mathcal{V}} \to \mathcal{F}_{\mathcal{V}}$ is an operator defined by:

$$\Delta_p f := -\frac{1}{2}\mathrm{div}(\|\nabla f\|^{p-2} \nabla f)$$

When $p = 2$, $\Delta_2 = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$.

Preliminary

Tencent AI Lab | ICML International Conference On Machine Learning

Local variation

Graph gradient

$(\nabla f)([i,j])$ measures the variation of f on **edge** $[i,j]$:

$$(\nabla f)([i,j]) := \sqrt{\frac{W_{i,j}}{D_{j,j}}} f(j) - \sqrt{\frac{W_{i,j}}{D_{i,i}}} f(i)$$

Graph divergence

$(\mathrm{div} g)(i)$ measures the net outflow of g on **node** i:

$$(\mathrm{div} g)(i) = \sum_{j=1}^{N} \sqrt{\frac{W_{i,j}}{D_{i,i}}} \left(g([i,j]) - g([j,i])\right)$$

Global variation

$\mathcal{S}_p(f)$ measures the overall variation of f on **graph** $\mathcal{G}$:

$$\mathcal{S}_p(f) := \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \|(\nabla f)([i,j])\|^p = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \left\| \sqrt{\frac{W_{i,j}}{D_{j,j}}} f(j) - \sqrt{\frac{W_{i,j}}{D_{i,i}}} f(i) \right\|^p$$

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$: A graph
$f : \mathcal{V} \to \mathbb{R}$: A function defined on $\mathcal{V}$
$g : \mathcal{E} \to \mathbb{R}$: A function defined on $\mathcal{E}$
$\mathbf{W}$: The adjacency matrix
$\mathbf{D}$: The degree matrix

Graph p-Laplacian

$\Delta_p : \mathcal{F}_{\mathcal{V}} \to \mathcal{F}_{\mathcal{V}}$ is an operator defined by:

$$\Delta_p f := -\frac{1}{2}\mathrm{div}(\|\nabla f\|^{p-2} \nabla f)$$

When $p = 2$, $\Delta_2 = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$.

The connection between $\mathcal{S}_p(f)$ and Δ_p :

$$\left.\frac{\partial \mathcal{S}_p(f)}{\partial f}\right|_i = p\Delta_p f(i)$$

Methodology— p-Laplacian regularization framework

Two assumptions:

1. Low variation of $\mathbf{F}$ over $\mathcal{G}$ w.r.t p;
2. The noise magnitude in $\mathbf{X}$ is small.

p-Laplacian regularization framework

$$\mathcal{L}_p(\mathbf{F}) := \min_{\mathbf{F}} \ \mathcal{S}_p(\mathbf{F}) + \mu \sum_{i=1}^{N} \|\mathbf{F}_{i,:} - \mathbf{X}_{i,:}\|^2$$

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$: A graph

$f : \mathcal{V} \to \mathbb{R}^c$: A c channel signal function

$\mathbf{X} \in \mathbb{R}^{N \times c}$: The node feature matrix

$\mathbf{F} \in \mathbb{R}^{N \times c}$: The node embedding matrix

μ : The hyperparameter

Methodology— p-Laplacian regularization framework

Tencent AI Lab

ICML International Conference On Machine Learning

Two assumptions:

1. Low variation of $\mathbf{F}$ over $\mathcal{G}$ w.r.t p;
2. The noise magnitude in $\mathbf{X}$ is small.

p-Laplacian regularization framework

$$\mathcal{L}_p(\mathbf{F}) := \min_{\mathbf{F}} \ \mathcal{S}_p(\mathbf{F}) + \mu \sum_{i=1}^{N} \|\mathbf{F}_{i,:} - \mathbf{X}_{i,:}\|^2$$

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$: A graph

$f : \mathcal{V} \to \mathbb{R}^c$: A c channel signal function

$\mathbf{X} \in \mathbb{R}^{N \times c}$: The node feature matrix

$\mathbf{F} \in \mathbb{R}^{N \times c}$: The node embedding matrix

μ : The hyperparameter

When $p = 2$

- **Closed-form** solution:

$$\mathbf{F}^* = \mu(\Delta_2 + \mu \mathbf{I}_N)^{-1}\mathbf{X}.$$

- Iteration algorithm (is guaranteed to converge):

$$\mathbf{F}^{(k+1)} = \alpha \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}\mathbf{F}^{(k)} + \beta \mathbf{X},$$

$$\alpha = \frac{1}{1+\mu}, \beta = \frac{\mu}{1+\mu}.$$

- $\mu(\Delta_2 + \mu \mathbf{I}_N)^{-1}$ = personalized PageRank matrix [1].

Methodology— p-Laplacian regularization framework

Two assumptions:

1. Low variation of $\mathbf{F}$ over $\mathcal{G}$ w.r.t p;
2. The noise magnitude in $\mathbf{X}$ is small.

p-Laplacian regularization framework

$$\mathcal{L}_p(\mathbf{F}) := \min_{\mathbf{F}} \ \mathcal{S}_p(\mathbf{F}) + \mu \sum_{i=1}^{N} \|\mathbf{F}_{i,:} - \mathbf{X}_{i,:}\|^2$$

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$: A graph

$f : \mathcal{V} \to \mathbb{R}^c$: A c channel signal function

$\mathbf{X} \in \mathbb{R}^{N \times c}$: The node feature matrix

$\mathbf{F} \in \mathbb{R}^{N \times c}$: The node embedding matrix

μ : The hyperparameter

When $p = 2$

- **Closed-form** solution:

$$\mathbf{F}^* = \mu(\Delta_2 + \mu \mathbf{I}_N)^{-1}\mathbf{X}.$$

- Iteration algorithm (is guaranteed to converge):

$$\mathbf{F}^{(k+1)} = \alpha \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}\mathbf{F}^{(k)} + \beta \mathbf{X},$$

$$\alpha = \frac{1}{1+\mu}, \beta = \frac{\mu}{1+\mu}.$$

- $\mu(\Delta_2 + \mu \mathbf{I}_N)^{-1}$ = personalized PageRank matrix [1].

When $p \geq 1$

- **Not** aware of any closed-form solution.
- Iteration algorithm:

$$\mathbf{F}^{(k+1)}_{i,:} = \alpha^{(k)}_{i,i} \sum_{j=1}^{N} \frac{M^{(k)}_{i,j}}{\sqrt{D_{i,i}D_{j,j}}} \mathbf{F}^{(k)}_{j,:} + \beta^{(k)}_{i,i} \mathbf{X}_{i,:}$$

$$M^{(k)}_{i,j} = W_{i,j} \left\| \sqrt{\frac{W_{i,j}}{D_{i,i}}} \mathbf{F}^{(k)}_{i,:} - \sqrt{\frac{W_{i,j}}{D_{j,j}}} \mathbf{F}^{(k)}_{j,:} \right\|^{p-2}$$

$$\alpha^{(k)}_{i,i} = 1 \Big/ \left(\sum_{j=1}^{N} \frac{M^{(k)}_{i,j}}{D_{i,i}} + \frac{2\mu}{p} \right), \quad \beta^{(k)}_{i,i} = \frac{2\mu}{p} \alpha^{(k)}_{i,i}$$

Methodology— p-Laplacian message passing & ^{p}GNN Architecture

When $p \geq 1$

- Iteration algorithm:

$$\mathbf{F}_{i,:}^{(k+1)} = \alpha_{i,i}^{(k)} \sum_{j=1}^{N} \frac{M_{i,j}^{(k)}}{\sqrt{D_{i,i} D_{j,j}}} \mathbf{F}_{j,:}^{(k)} + \beta_{i,i}^{(k)} \mathbf{X}_{i,:}$$

$$M_{i,j}^{(k)} = W_{i,j} \left\| \sqrt{\frac{W_{i,j}}{D_{i,i}}} \mathbf{F}_{i,:}^{(k)} - \sqrt{\frac{W_{i,j}}{D_{j,j}}} \mathbf{F}_{j,:}^{(k)} \right\|^{p-2}$$

$$\alpha_{i,i}^{(k)} = 1 / \left(\sum_{j=1}^{N} \frac{M_{i,j}^{(k)}}{D_{i,i}} + \frac{2\mu}{p} \right), \quad \beta_{i,i}^{(k)} = \frac{2\mu}{p} \alpha_{i,i}^{(k)}$$

Methodology— p-Laplacian message passing & ^{p}GNN Architecture

p-Laplacian Message Passing

$$\mathbf{F}^{(k+1)} = \alpha^{(k)}\mathbf{D}^{-1/2}\mathbf{M}^{(k)}\mathbf{D}^{-1/2}\mathbf{F}^{(k)} + \beta^{(k)}\mathbf{X}$$

$$M_{i,j}^{(k)} = W_{i,j}\left\|\sqrt{\frac{W_{i,j}}{D_{i,i}}}\mathbf{F}_{i,:}^{(k)} - \sqrt{\frac{W_{i,j}}{D_{j,j}}}\mathbf{F}_{j,:}^{(k)}\right\|^{p-2}$$

$$\alpha_{i,i}^{(k)} = 1\Big/\left(\sum_{j=1}^{N}\frac{M_{i,j}^{(k)}}{D_{i,i}} + \frac{2\mu}{p}\right), \quad \beta_{i,i}^{(k)} = \frac{2\mu}{p}\alpha_{i,i}^{(k)}$$

When $p \geq 1$

- Iteration algorithm:

$$\mathbf{F}_{i,:}^{(k+1)} = \alpha_{i,i}^{(k)}\sum_{j=1}^{N}\frac{M_{i,j}^{(k)}}{\sqrt{D_{i,i}D_{j,j}}}\mathbf{F}_{j,:}^{(k)} + \beta_{i,i}^{(k)}\mathbf{X}_{i,:}$$

$$M_{i,j}^{(k)} = W_{i,j}\left\|\sqrt{\frac{W_{i,j}}{D_{i,i}}}\mathbf{F}_{i,:}^{(k)} - \sqrt{\frac{W_{i,j}}{D_{j,j}}}\mathbf{F}_{j,:}^{(k)}\right\|^{p-2}$$

$$\alpha_{i,i}^{(k)} = 1\Big/\left(\sum_{j=1}^{N}\frac{M_{i,j}^{(k)}}{D_{i,i}} + \frac{2\mu}{p}\right), \quad \beta_{i,i}^{(k)} = \frac{2\mu}{p}\alpha_{i,i}^{(k)}$$

Methodology— p-Laplacian message passing & ^{p}GNN Architecture

p-Laplacian Message Passing

$$\mathbf{F}^{(k+1)} = \alpha^{(k)}\mathbf{D}^{-1/2}\mathbf{M}^{(k)}\mathbf{D}^{-1/2}\mathbf{F}^{(k)} + \beta^{(k)}\mathbf{X}$$

$$M_{i,j}^{(k)} = W_{i,j}\left\|\sqrt{\frac{W_{i,j}}{D_{i,i}}}\mathbf{F}_{i,:}^{(k)} - \sqrt{\frac{W_{i,j}}{D_{j,j}}}\mathbf{F}_{j,:}^{(k)}\right\|^{p-2}$$

$$\alpha_{i,i}^{(k)} = 1/\left(\sum_{j=1}^{N}\frac{M_{i,j}^{(k)}}{D_{i,i}} + \frac{2\mu}{p}\right), \quad \beta_{i,i}^{(k)} = \frac{2\mu}{p}\alpha_{i,i}^{(k)}$$

When $p \geq 1$

- Iteration algorithm:

$$\mathbf{F}_{i,:}^{(k+1)} = \alpha_{i,i}^{(k)}\sum_{j=1}^{N}\frac{M_{i,j}^{(k)}}{\sqrt{D_{i,i}D_{j,j}}}\mathbf{F}_{j,:}^{(k)} + \beta_{i,i}^{(k)}\mathbf{X}_{i,:}$$

$$M_{i,j}^{(k)} = W_{i,j}\left\|\sqrt{\frac{W_{i,j}}{D_{i,i}}}\mathbf{F}_{i,:}^{(k)} - \sqrt{\frac{W_{i,j}}{D_{j,j}}}\mathbf{F}_{j,:}^{(k)}\right\|^{p-2}$$

$$\alpha_{i,i}^{(k)} = 1/\left(\sum_{j=1}^{N}\frac{M_{i,j}^{(k)}}{D_{i,i}} + \frac{2\mu}{p}\right), \quad \beta_{i,i}^{(k)} = \frac{2\mu}{p}\alpha_{i,i}^{(k)}$$

p-Laplacian message passing is guaranteed to converge with proper chosen μ by Theorem 2.

Methodology— p-Laplacian message passing & ^{p}GNN Architecture

Tencent AI Lab ICML International Conference On Machine Learning

p-Laplacian Message Passing

$$\mathbf{F}^{(k+1)} = \alpha^{(k)}\mathbf{D}^{-1/2}\mathbf{M}^{(k)}\mathbf{D}^{-1/2}\mathbf{F}^{(k)} + \beta^{(k)}\mathbf{X}$$

$$M_{i,j}^{(k)} = W_{i,j}\left\|\sqrt{\frac{W_{i,j}}{D_{i,i}}}\mathbf{F}_{i,:}^{(k)} - \sqrt{\frac{W_{i,j}}{D_{j,j}}}\mathbf{F}_{j,:}^{(k)}\right\|^{p-2}$$

$$\alpha_{i,i}^{(k)} = 1/\left(\sum_{j=1}^{N}\frac{M_{i,j}^{(k)}}{D_{i,i}} + \frac{2\mu}{p}\right), \quad \beta_{i,i}^{(k)} = \frac{2\mu}{p}\alpha_{i,i}^{(k)}$$

When $p \geq 1$

- Iteration algorithm:

$$\mathbf{F}_{i,:}^{(k+1)} = \alpha_{i,i}^{(k)}\sum_{j=1}^{N}\frac{M_{i,j}^{(k)}}{\sqrt{D_{i,i}D_{j,j}}}\mathbf{F}_{j,:}^{(k)} + \beta_{i,i}^{(k)}\mathbf{X}_{i,:}$$

$$M_{i,j}^{(k)} = W_{i,j}\left\|\sqrt{\frac{W_{i,j}}{D_{i,i}}}\mathbf{F}_{i,:}^{(k)} - \sqrt{\frac{W_{i,j}}{D_{j,j}}}\mathbf{F}_{j,:}^{(k)}\right\|^{p-2}$$

$$\alpha_{i,i}^{(k)} = 1/\left(\sum_{j=1}^{N}\frac{M_{i,j}^{(k)}}{D_{i,i}} + \frac{2\mu}{p}\right), \quad \beta_{i,i}^{(k)} = \frac{2\mu}{p}\alpha_{i,i}^{(k)}$$

p-Laplacian message passing is guaranteed to converge with proper chosen μ by Theorem 2.

^{p}GNN Architecture

$$\mathbf{F}^{(0)} = \mathrm{ReLU}(\mathbf{X}\Theta^{(1)})$$

$$\mathbf{F}^{(k+1)} = \alpha^{(k)}\mathbf{D}^{-1/2}\mathbf{M}^{(k)}\mathbf{D}^{-1/2}\mathbf{F}^{(k)} + \beta^{(k)}\mathbf{F}^{(0)}, k = 0, 1, \ldots, K-1$$

$$\mathbf{Z} = \mathrm{softmax}(\mathbf{F}^{(K)}\Theta^{(2)})$$

$\mathbf{X} \in \mathbb{R}^{N\times c}$: The node feature matrix

$\mathbf{F} \in \mathbb{R}^{N\times c}$: The node embedding matrix

$\mathbf{\Theta}^{(1)}, \mathbf{\Theta}^{(2)}$: The 1st, 2nd layer NN parameters

K : The number of iterations

Spectral Analysis— p-Laplacian based graph convolution

p-Laplacian based graph convolution

$$g_\theta \star \mathbf{X} = \Phi_p(\mathbf{U})\hat{g}_\theta(\Lambda)\Phi_p(\mathbf{U})^\top \mathbf{X}$$

Spectral Analysis— p-Laplacian based graph convolution

p-Laplacian based graph convolution

$$g_\theta \star \mathbf{X} = \Phi_p(\mathbf{U})\hat{g}_\theta(\Lambda)\Phi_p(\mathbf{U})^\top \mathbf{X}$$

The polynomial approximation

$$g_\theta \star \mathbf{X} \approx \sum_{k=0}^{K-1} \theta_k \Phi_p(\mathbf{U})\Lambda^k \Phi_p(\mathbf{U})^\top \mathbf{X} = \sum_{k=0}^{K-1} \theta_k \Delta_p^k \mathbf{X}$$

Spectral Analysis— p-Laplacian based graph convolution

Tencent AI Lab
ICML International Conference On Machine Learning

p-Laplacian based graph convolution

$$g_\theta \star \mathbf{X} = \Phi_p(\mathbf{U})\hat{g}_\theta(\Lambda)\Phi_p(\mathbf{U})^\top \mathbf{X}$$

The polynomial approximation

$$g_\theta \star \mathbf{X} \approx \sum_{k=0}^{K-1} \theta_k \Phi_p(\mathbf{U})\Lambda^k \Phi_p(\mathbf{U})^\top \mathbf{X} = \sum_{k=0}^{K-1} \theta_k \Delta_p^k \mathbf{X}$$

Theorem 3

p-Laplacian message passing is a polynomial filter defined on the spectral domain of p-Laplacian!

$$g_\theta \star \mathbf{X} \approx \sum_{k=0}^{K-1} \theta_k \Delta_p^k \mathbf{X} \approx p\text{-Laplacian message passing}$$

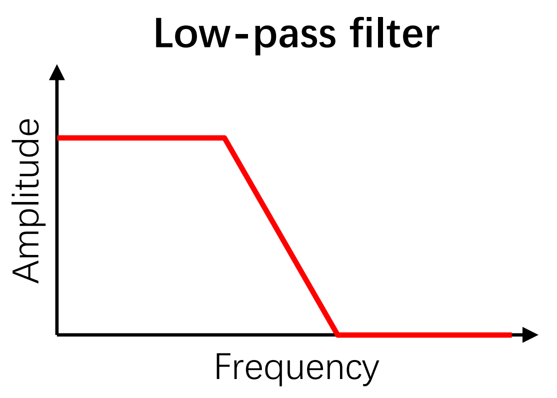
Spectral Analysis— Low-Pass & Low-High-Pass Filters

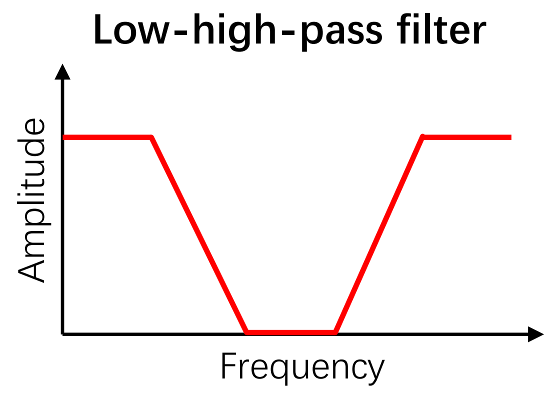
Spectral analysis of p-Laplacian message passing

It adaptively acts as low-pass & low-high-pass filters on node i in terms of its graph gradients $\|\nabla f(i)\|$.

Proposition 1. *Given a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ with node embeddings $\mathbf{F}$ and the p-Laplacian Δ_p with its p-eigenvectors $\{\mathbf{u}^{(l)}\}_{l=0,1,\ldots,N-1}$ and the p-eigenvalues $\{\lambda_l\}_{l=0,1,\ldots,N-1}$. Let $g_p(\lambda_{i-1}) := \alpha_{i,i} \sum_j D_{i,i}^{-1/2} M_{i,j} D_{j,j}^{-1/2}$ for $i \in [N]$ be the filters defined on the spectral domain of Δ_p, where $M_{i,j} = W_{i,j} \|\nabla f([i,j])\|^{p-2}$, $(\nabla f)([i,j])$ is the graph gradient of the edge between node i and j and $\|\nabla f(i)\|$ is the norm of graph gradient at i. N_i denotes the number of edges connected to i, $N_{min} = \min\{N_j\}_{j \in [N]}$, and $k = \arg\max_j(\{|u_j^{(l)}|/\sqrt{D_{l,l}}\}_{j\in[N]; l=0,\ldots,N-1})$, then*

1. *When $p = 2$, $g_p(\lambda_{i-1})$ works as low-high-pass filters.*
2. *When $p > 2$, if $\|\nabla f(i)\| \leq 2^{(p-1)/(p-2)}$, $g_p(\lambda_{i-1})$ works as low-high-pass filters on node i and $g_p(\lambda_{i-1})$ works as low-pass filters on i when $\|\nabla f(i)\| \geq 2^{(p-1)/(p-2)}$.*
3. *When $1 \leq p < 2$, if $0 \leq \|\nabla f(i)\| \leq 2(2\sqrt{N_k})^{1/(p-2)}$, $g_p(\lambda_{i-1})$ works as low-pass filters on node i and $g_p(\lambda_{i-1})$ works as low-high-pass filters on i when $\|\nabla f(i)\| \geq 2\left(2\sqrt{N_k}\right)^{1/(p-2)}$. Specifically, when $p = 1$, N_k can be replaced by N_{min}.*





	Low-pass filter	Low-high-pass filter
$1 \leq p < 2$	$\|\nabla f(i)\| < 2(2\sqrt{N_k})^{1/(p-2)}$	$\|\nabla f(i)\| \geq 2(2\sqrt{N_k})^{1/(p-2)}$
$p = 2$	$-$	Always
$p > 2$	$\|\nabla f(i)\| > 2^{(p-1)/(p-2)}$	$\|\nabla f(i)\| \leq 2^{(p-1)/(p-2)}$

Empirical Studies —semi-supervised node classification

Tencent AI Lab

ICML International Conference On Machine Learning

Table 2: Statistics of datasets.

Dataset	#Class	#Feature	#Node	#Edge	Training	Validation	Testing	$\mathcal{H}(\mathcal{G})$
Cora	7	1433	2708	5278	2.5%	2.5%	95%	0.825
CiteSeer	6	3703	3327	4552	2.5%	2.5%	95%	0.717
PubMed	3	500	19717	44324	2.5%	2.5%	95%	0.792
Computers	10	767	13381	245778	2.5%	2.5%	95%	0.802
Photo	8	745	7487	119043	2.5%	2.5%	95%	0.849
CS	15	6805	18333	81894	2.5%	2.5%	95%	0.832
Physics	5	8415	34493	247962	2.5%	2.5%	95%	0.915
Chameleon	5	2325	2277	31371	60%	20%	20%	0.247
Squirrel	5	2089	5201	198353	60%	20%	20%	0.216
Actor	5	932	7600	26659	60%	20%	20%	0.221
Wisconsin	5	251	499	1703	60%	20%	20%	0.150
Texas	5	1703	183	279	60%	20%	20%	0.097
Cornell	5	1703	183	277	60%	20%	20%	0.386

Baselines:

- MLP
- GCN
- SGC
- GAT
- JKNet
- APPNP
- GPRGNN

Empirical Studies —semi-supervised node classification

Table 2: Statistics of datasets.

Dataset	#Class	#Feature	#Node	#Edge	Training	Validation	Testing	$\mathcal{H}(\mathcal{G})$
Cora	7	1433	2708	5278	2.5%	2.5%	95%	0.825
CiteSeer	6	3703	3327	4552	2.5%	2.5%	95%	0.717
PubMed	3	500	19717	44324	2.5%	2.5%	95%	0.792
Computers	10	767	13381	245778	2.5%	2.5%	95%	0.802
Photo	8	745	7487	119043	2.5%	2.5%	95%	0.849
CS	15	6805	18333	81894	2.5%	2.5%	95%	0.832
Physics	5	8415	34493	247962	2.5%	2.5%	95%	0.915
Chameleon	5	2325	2277	31371	60%	20%	20%	0.247
Squirrel	5	2089	5201	198353	60%	20%	20%	0.216
Actor	5	932	7600	26659	60%	20%	20%	0.221
Wisconsin	5	251	499	1703	60%	20%	20%	0.150
Texas	5	1703	183	279	60%	20%	20%	0.097
Cornell	5	1703	183	277	60%	20%	20%	0.386

Baselines:

- MLP
- GCN
- SGC
- GAT
- JKNet
- APPNP
- GPRGNN

Table 1: Heterophilious results. Averaged accuracy (%) for 100 runs. Best results outlined in bold and the results within 95% confidence interval of the best results are outlined in underlined bold.

Method	Chameleon	Squirrel	Actor	Wisconsin	Texas	Cornell
MLP	$48.02_{\pm 1.72}$	**33.80**$_{\pm 1.05}$	$39.68_{\pm 1.43}$	$93.56_{\pm 3.14}$	$79.50_{\pm 10.62}$	$80.30_{\pm 11.38}$
GCN	$34.54_{\pm 2.78}$	$25.28_{\pm 1.55}$	$31.28_{\pm 2.04}$	$61.93_{\pm 3.00}$	$56.54_{\pm 17.02}$	$51.36_{\pm 4.59}$
SGC	$34.76_{\pm 4.55}$	$25.49_{\pm 1.63}$	$30.98_{\pm 3.80}$	$66.94_{\pm 2.58}$	$59.99_{\pm 9.95}$	$44.39_{\pm 5.88}$
GAT	$45.16_{\pm 2.10}$	$31.41_{\pm 0.98}$	$34.11_{\pm 1.28}$	$65.64_{\pm 6.29}$	$56.41_{\pm 13.01}$	$43.94_{\pm 7.33}$
JKNet	$33.28_{\pm 3.59}$	$25.82_{\pm 1.58}$	$29.77_{\pm 2.61}$	$61.08_{\pm 3.71}$	$59.65_{\pm 12.62}$	$55.34_{\pm 4.43}$
APPNP	$36.18_{\pm 2.81}$	$26.85_{\pm 1.48}$	$31.26_{\pm 2.52}$	$64.59_{\pm 3.49}$	$82.90_{\pm 5.08}$	$66.47_{\pm 9.34}$
GPRGNN	$43.67_{\pm 2.27}$	$31.27_{\pm 1.76}$	$36.63_{\pm 1.22}$	$88.54_{\pm 4.94}$	$80.74_{\pm 6.76}$	$78.95_{\pm 8.52}$
$^{1.0}$GNN	**48.86**$_{\pm 1.95}$	**33.75**$_{\pm 1.50}$	**40.62**$_{\pm 1.25}$	**95.37**$_{\pm 2.06}$	$84.06_{\pm 7.41}$	**82.16**$_{\pm 8.62}$
$^{1.5}$GNN	**48.74**$_{\pm 1.62}$	$33.33_{\pm 1.45}$	**40.35**$_{\pm 1.35}$	**95.24**$_{\pm 2.01}$	$84.46_{\pm 7.79}$	$78.47_{\pm 6.87}$
$^{2.0}$GNN	**48.77**$_{\pm 1.87}$	$33.60_{\pm 1.47}$	$40.07_{\pm 1.17}$	$91.15_{\pm 2.76}$	**87.96**$_{\pm 6.27}$	$72.04_{\pm 8.22}$
$^{2.5}$GNN	**48.80**$_{\pm 1.77}$	**33.79**$_{\pm 1.45}$	$39.80_{\pm 1.31}$	$87.08_{\pm 2.69}$	$83.01_{\pm 6.80}$	$70.31_{\pm 8.84}$

Table 3: Results on homophilic benchmark datasets. Averaged accuracy (%) for 100 runs. Best results are outlined in bold and the results within 95% confidence interval of the best results are outlined in underlined bold. OOM denotes out of memory.

Method	Cora	CiteSeer	PubMed	Computers	Photo	CS	Physics
MLP	$43.47_{\pm 3.82}$	$46.95_{\pm 2.15}$	$78.95_{\pm 0.49}$	$66.11_{\pm 2.70}$	$76.44_{\pm 2.83}$	$86.24_{\pm 1.43}$	$92.58_{\pm 0.83}$
GCN	$76.23_{\pm 0.79}$	$62.43_{\pm 0.81}$	$83.72_{\pm 0.27}$	$84.17_{\pm 0.59}$	$90.46_{\pm 0.48}$	$90.33_{\pm 0.36}$	$94.46_{\pm 0.08}$
SGC	$77.19_{\pm 1.47}$	**64.10**$_{\pm 1.36}$	$79.26_{\pm 0.69}$	$84.32_{\pm 0.59}$	$89.81_{\pm 0.57}$	$91.06_{\pm 0.05}$	OOM
GAT	$75.62_{\pm 1.01}$	$61.28_{\pm 1.09}$	$83.60_{\pm 0.22}$	$82.72_{\pm 1.29}$	$90.48_{\pm 0.57}$	$89.96_{\pm 0.27}$	$93.96_{\pm 0.21}$
JKNet	$77.19_{\pm 0.98}$	$63.32_{\pm 0.95}$	$82.54_{\pm 0.43}$	$79.94_{\pm 2.47}$	$88.29_{\pm 1.64}$	$89.69_{\pm 0.66}$	$93.92_{\pm 0.32}$
APPNP	**79.58**$_{\pm 0.59}$	$63.02_{\pm 1.10}$	**84.80**$_{\pm 0.22}$	$83.32_{\pm 1.11}$	$90.42_{\pm 0.53}$	$91.54_{\pm 0.24}$	**94.93**$_{\pm 0.06}$
GPRGNN	$76.10_{\pm 1.30}$	$61.60_{\pm 1.69}$	$83.16_{\pm 0.84}$	$82.78_{\pm 1.87}$	$89.81_{\pm 0.66}$	$90.59_{\pm 0.38}$	**94.72**$_{\pm 0.16}$
$^{1.0}$GNN	$77.59_{\pm 0.69}$	$63.19_{\pm 0.98}$	$83.21_{\pm 0.30}$	$84.46_{\pm 0.89}$	**90.69**$_{\pm 0.66}$	$91.46_{\pm 0.50}$	**94.72**$_{\pm 0.37}$
$^{1.5}$GNN	$78.86_{\pm 0.75}$	**63.80**$_{\pm 0.79}$	$83.65_{\pm 0.17}$	**85.03**$_{\pm 0.90}$	**90.91**$_{\pm 0.50}$	**92.12**$_{\pm 0.40}$	**94.90**$_{\pm 0.16}$
$^{2.0}$GNN	$78.93_{\pm 0.60}$	**63.65**$_{\pm 1.08}$	$84.19_{\pm 0.22}$	$84.39_{\pm 0.85}$	$90.40_{\pm 0.63}$	**92.28**$_{\pm 0.47}$	**94.93**$_{\pm 0.14}$
$^{2.5}$GNN	$78.87_{\pm 0.57}$	$63.28_{\pm 0.97}$	**84.45**$_{\pm 0.18}$	$83.85_{\pm 0.87}$	$89.82_{\pm 0.64}$	$91.94_{\pm 0.40}$	**94.87**$_{\pm 0.11}$

Emprical Studies — Semi-supervised node classification

Tencent AI Lab

ICML International Conference On Machine Learning

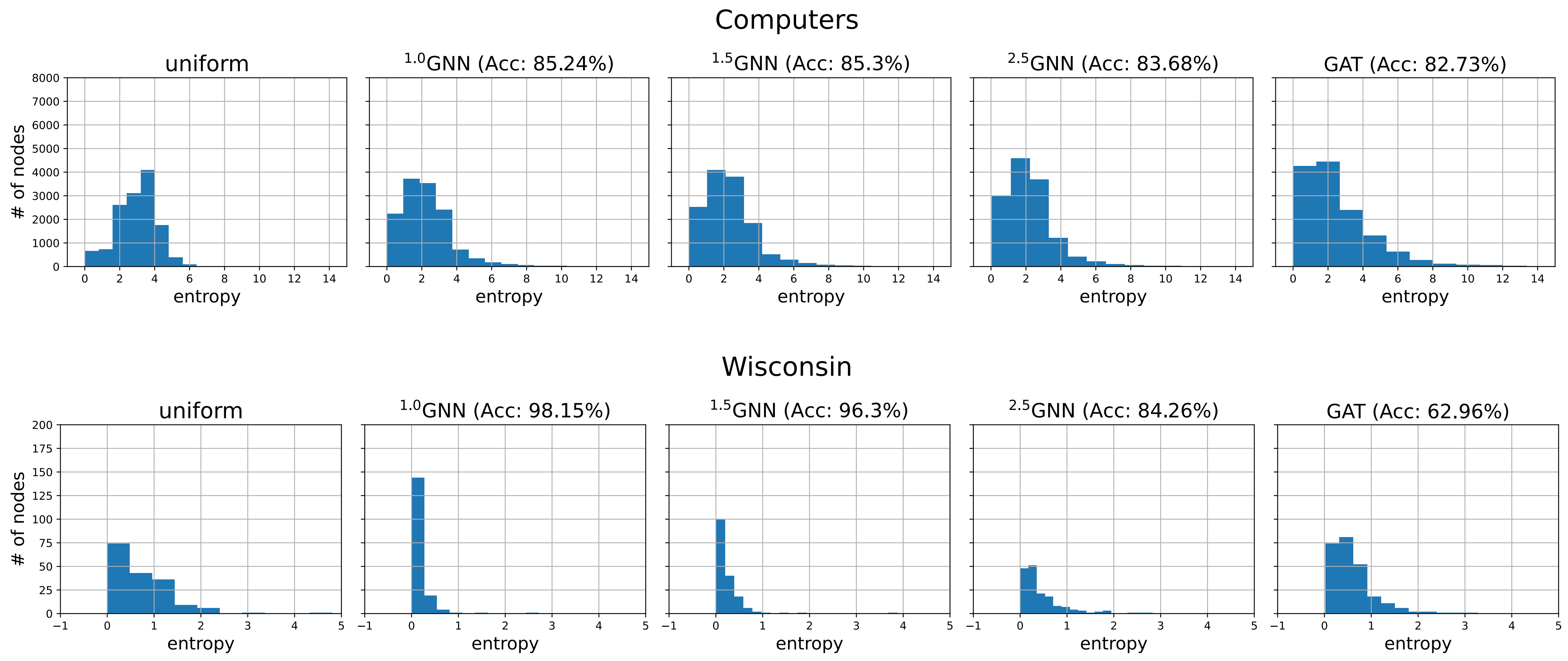


Fig. 1: Aggregation weight entropy distribution of graphs.

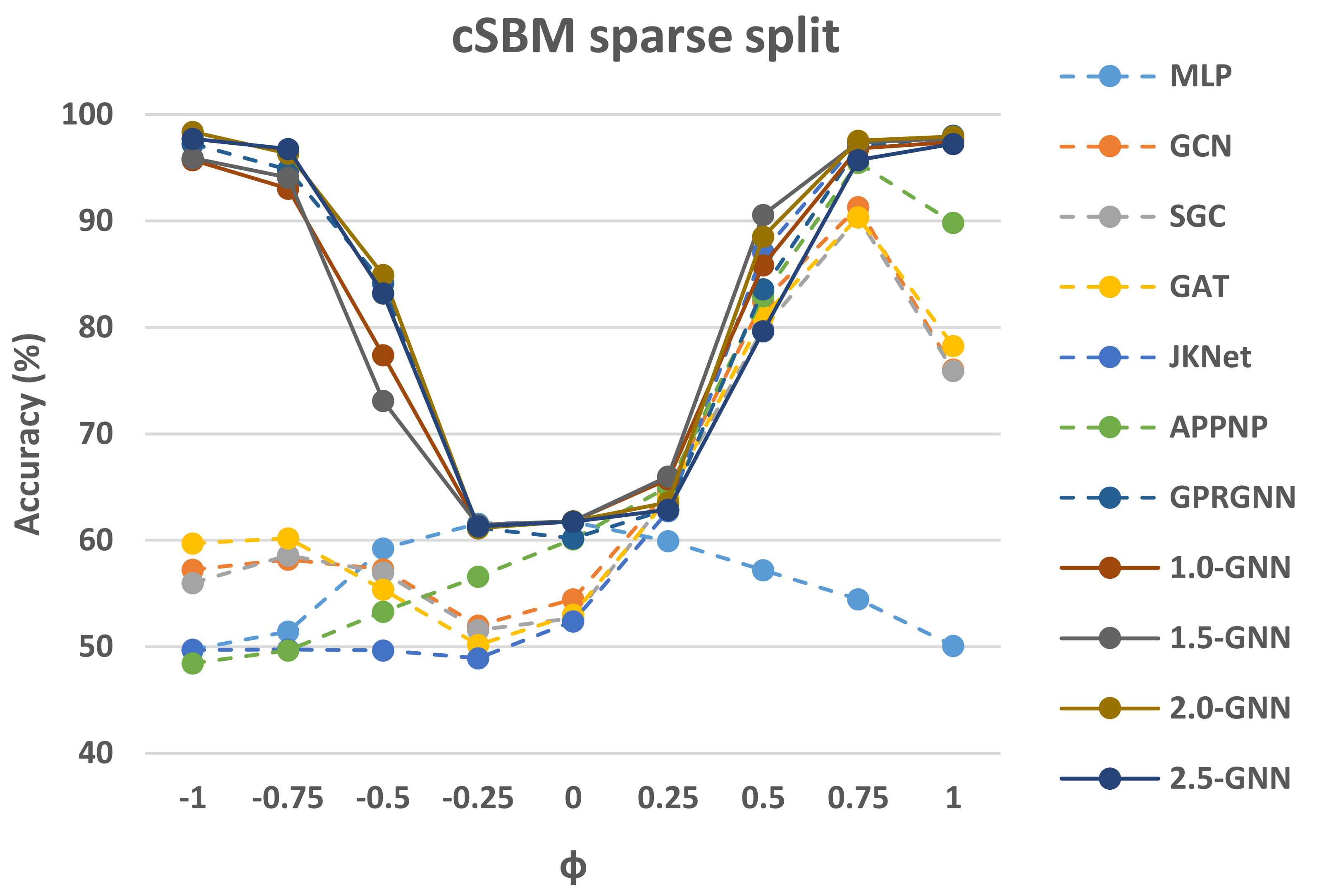


Fig. 2: Averaged accuracy on cSBM datasets.

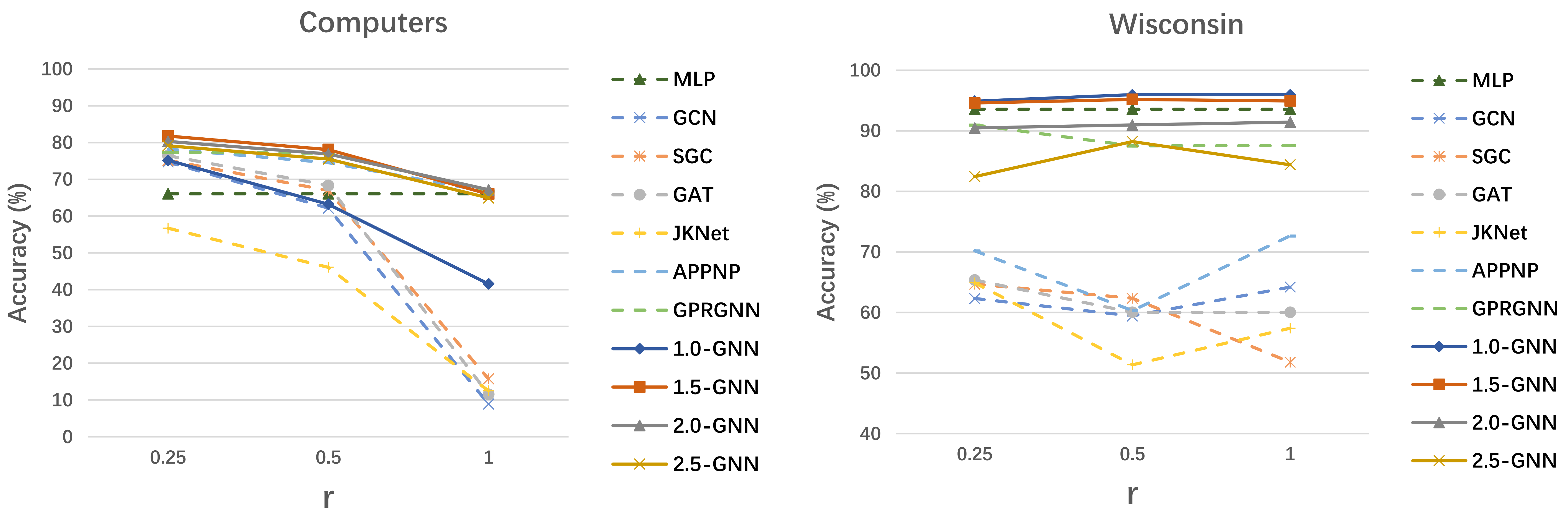


Fig. 1: Averaged accuracy on graphs with noisy edges.

Thank you!

Tencent AI Lab

ICML
International Conference On Machine Learning