

Motivation

- Most Graph Neural Networks (GNNs) use 1-hop neighbors. Severe limitation, real graphs are noisy!
- Real graphs are usually **homophilic**: Neighbors are similar. Models already leverage this by averaging over neighbors. *Why not exploit this more systematically?*

→ Generate more informative neighborhood via **graph diffusion**:

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

$$\tilde{A} = A + I_n, \quad \tilde{D}_{ii} = \sum_j \tilde{A}_{ij}, \quad \tilde{T}_{\text{sym}} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$$

e.g. heat kernel, personalized PageRank (PPR), GCN ($\theta_1 = 1$)

Sparsify result → new sparse graph, computationally efficient!

Spectral analysis

Why does this work?

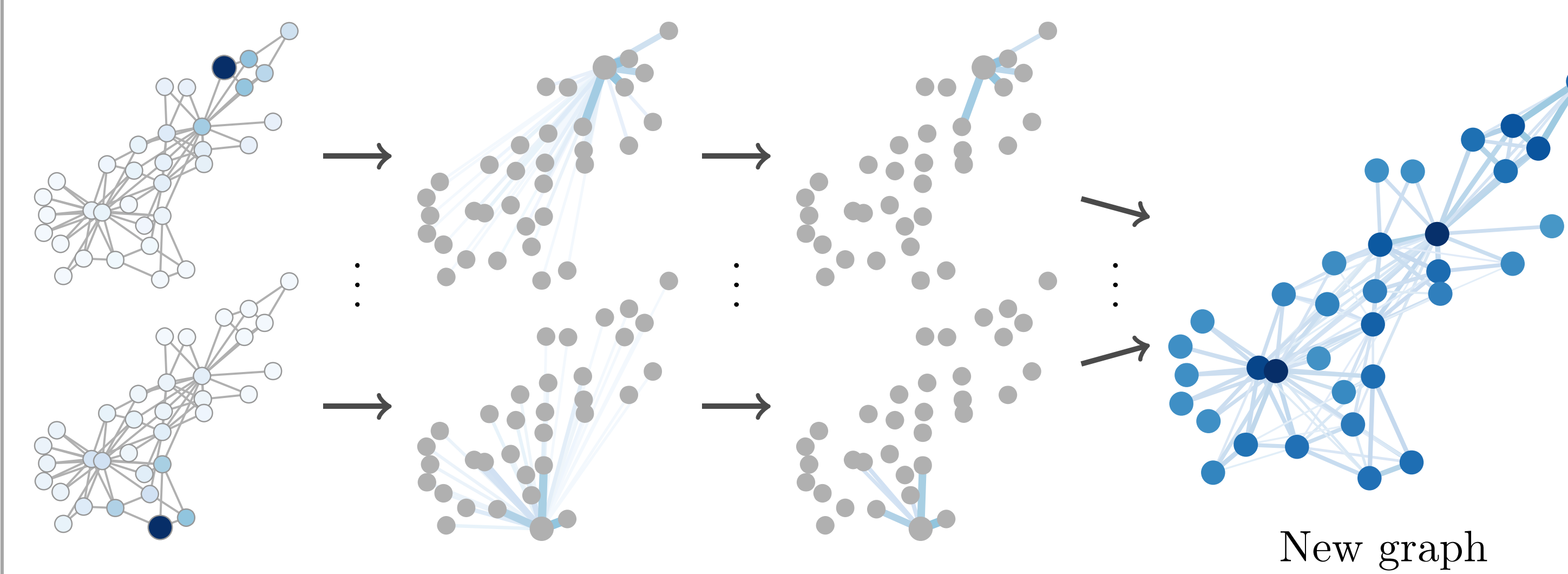
- Communities in graph correspond to eigenvectors: Low eigenvalue = large community.
- Using the adjacency matrix A corresponds to a **low-pass filter**.
- We are not limited to A ! Better filter? Graph diffusion.
→ Allows tuning the filter to the graph.

In fact, graph diffusion is *equivalent* to a polynomial filter:

$$g_{\xi}(L) = \sum_{j=0}^J \xi_j L^j, \quad \xi_j = \sum_{k=j}^{\infty} \binom{k}{j} (-1)^j \theta_k$$

Moreover, choosing proper θ_k guarantees localization.
→ sparsification possible, generalizes to unseen graphs

Graph Diffusion Convolution (GDC): Plug-and-play enhancement for graph-based models: GNNs, spectral clustering, ...

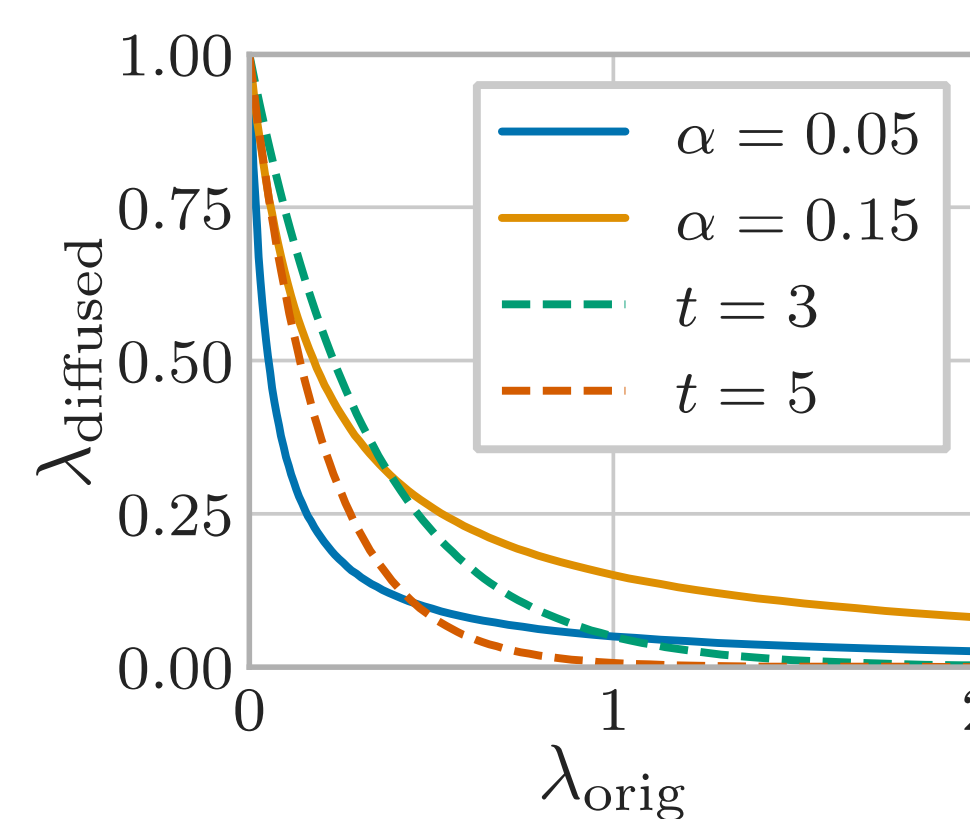


Graph diffusion Density defines edges Sparsify edges

Intuition: Denoising filter

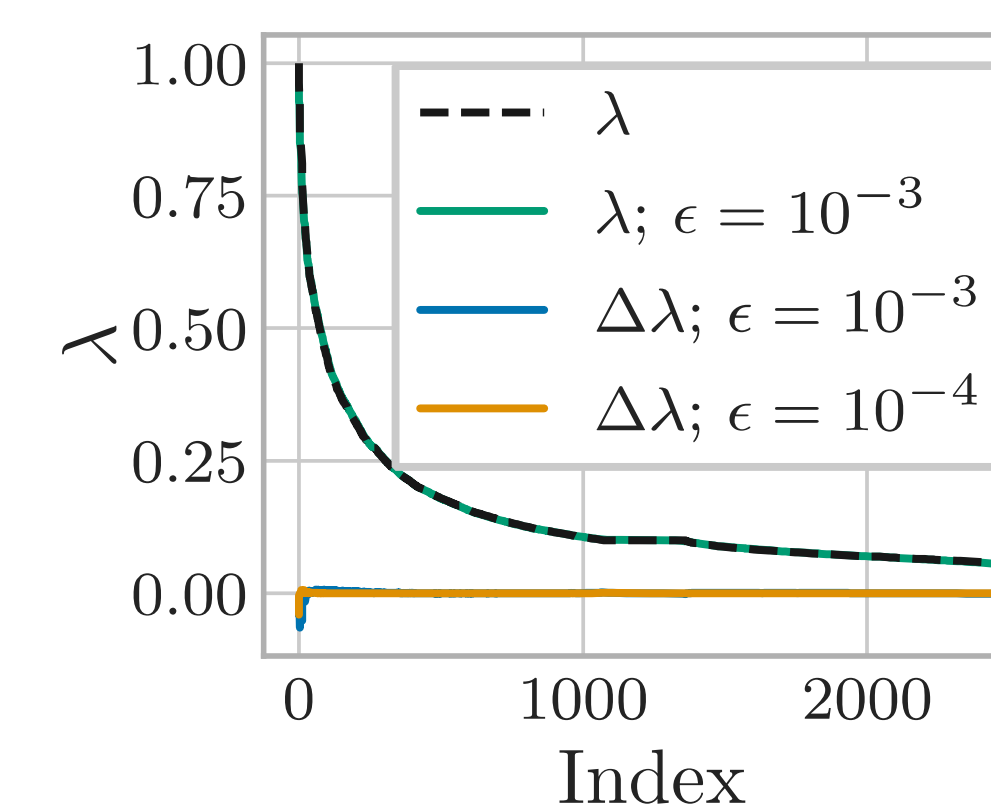
1. Graph diffusion

Low-pass filter



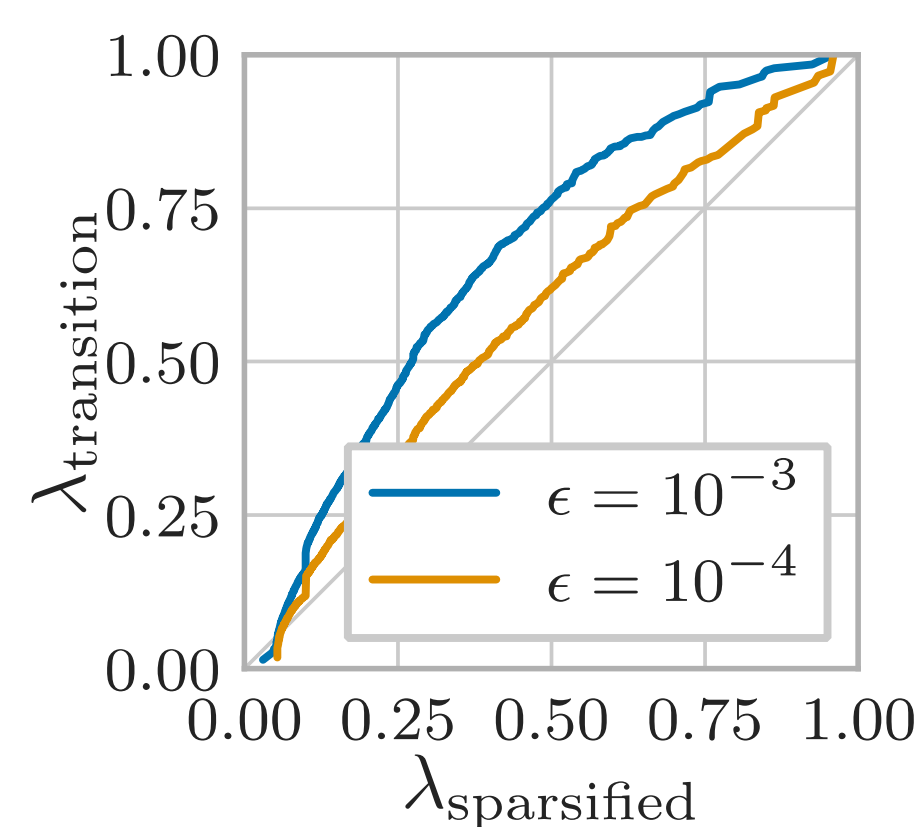
2. Sparsification

Almost no impact on eigenvalues



3. Transition matrix

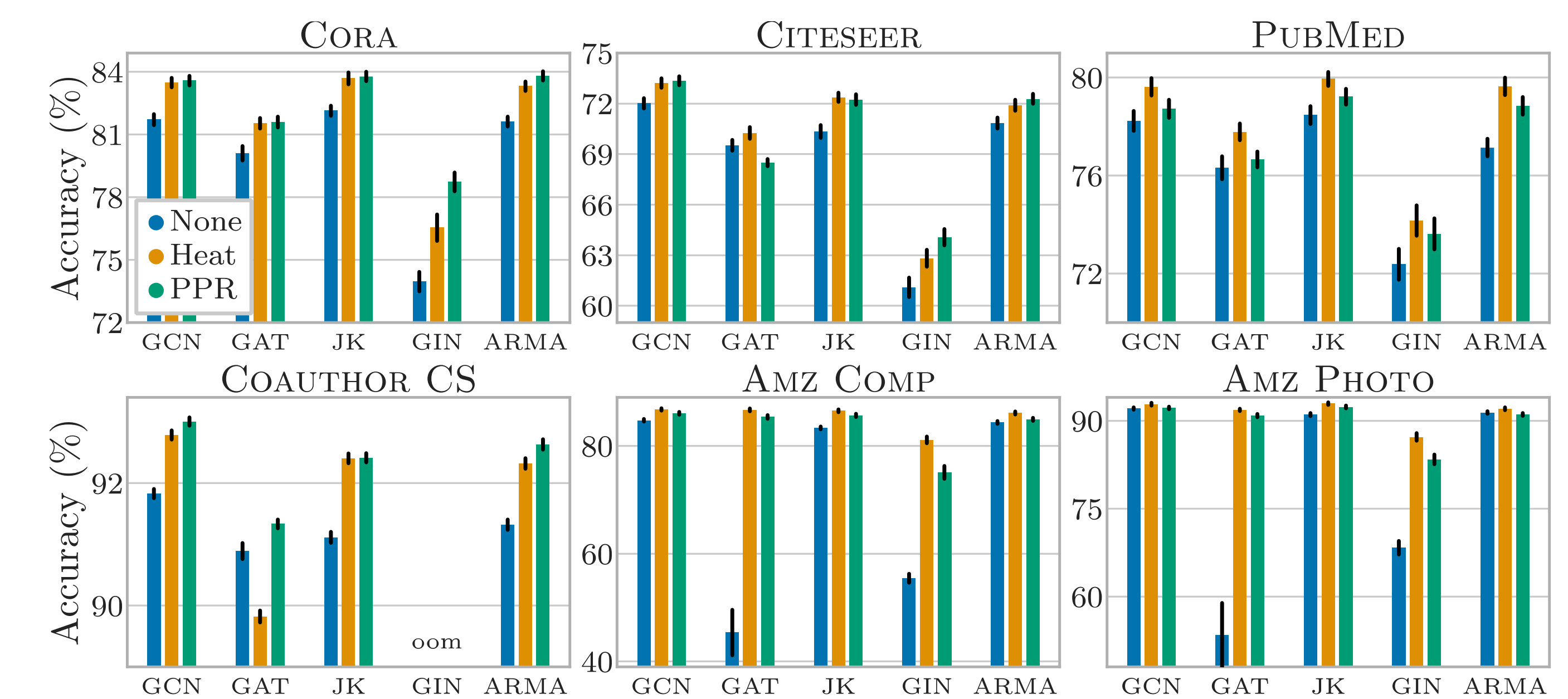
Amplifies medium eigenvalues



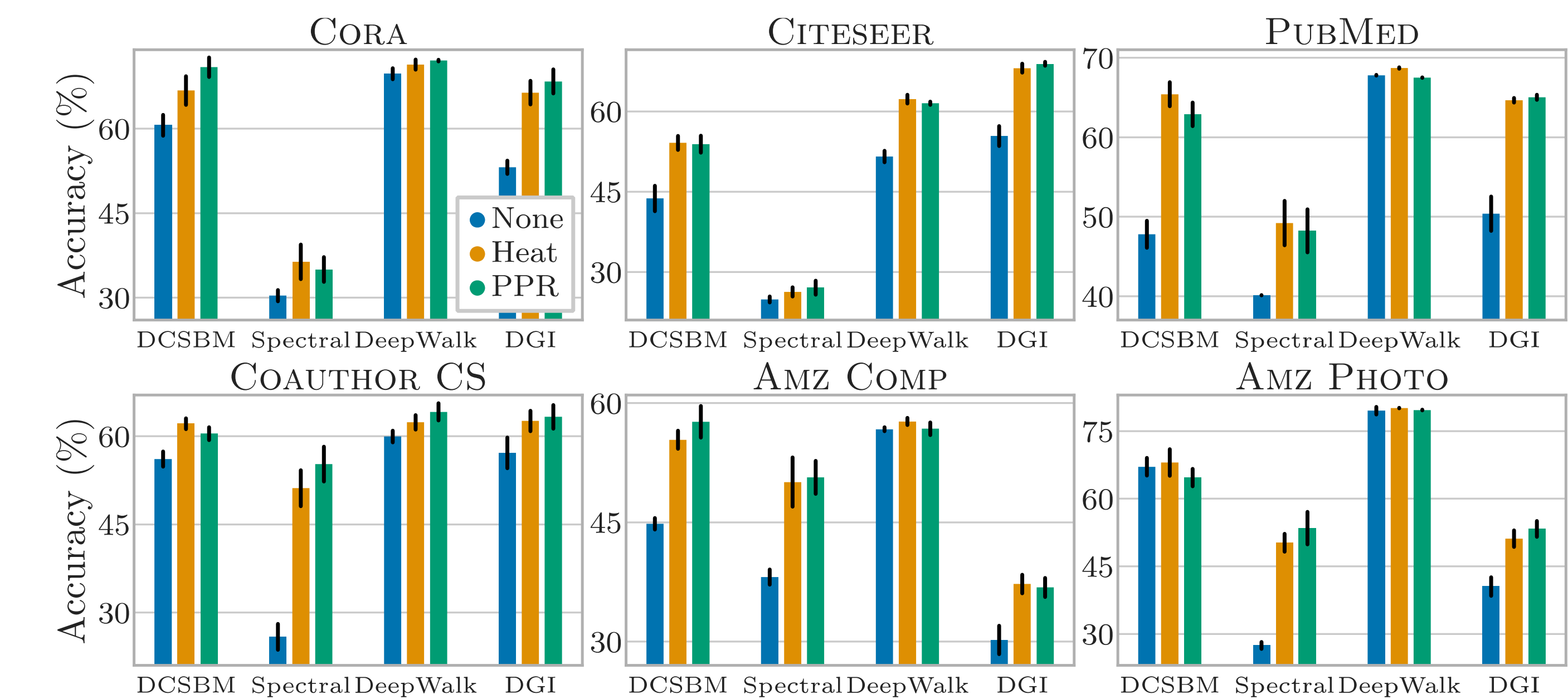
Consistent performance improvements

- Across 9 models and 6 datasets
- Every setting tuned individually, over 100,000 training runs

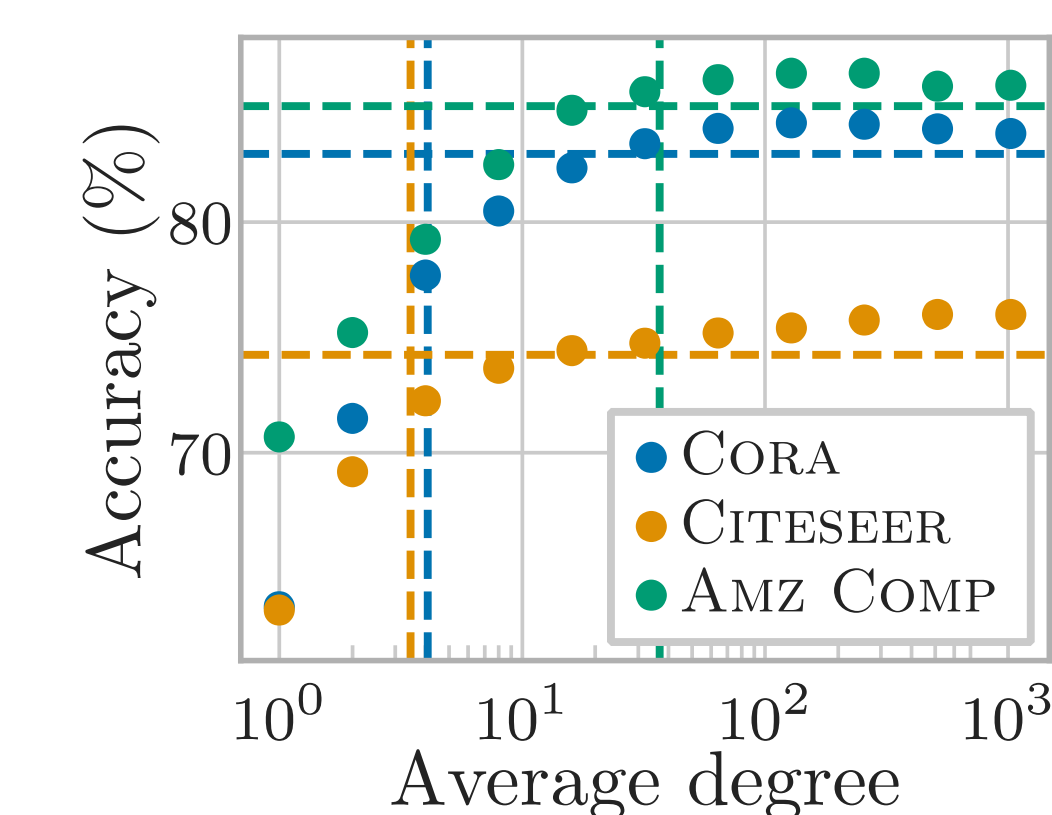
Semi-supervised classification (GNNs)



Unsupervised clustering



Similar graph density



Best for sparse labels

