Learning Laplacian Positional Encodings

Michael Ito[1], Jiong Zhu[1], Dexiong Chen[2], Danai Koutra[1], Jenna Wiens[1]
[1]University of Michigan Computer Science and Engineering
[2]Max Planck Institute of Biochemistry Department of Machine Learning and Systems Biology
Contact: mbito@umich.edu

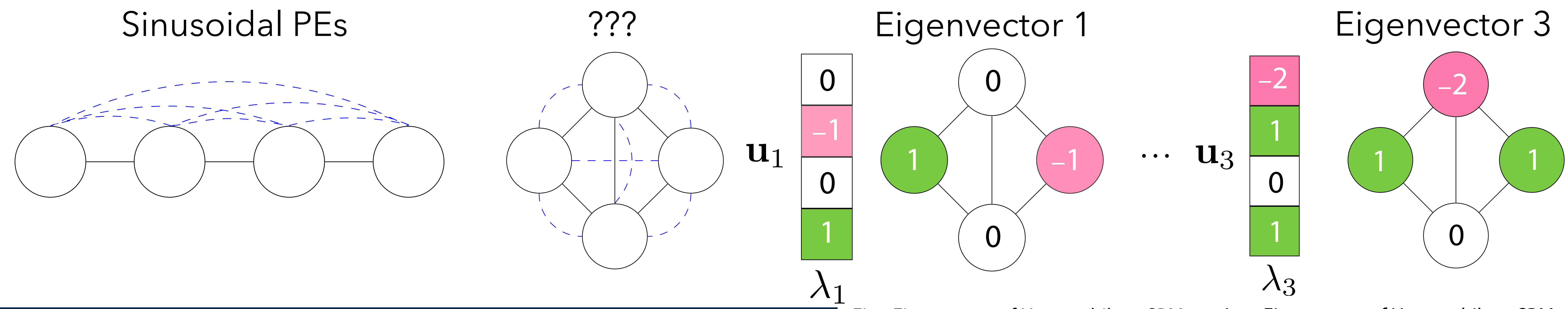
Paper | Code

1. Motivation

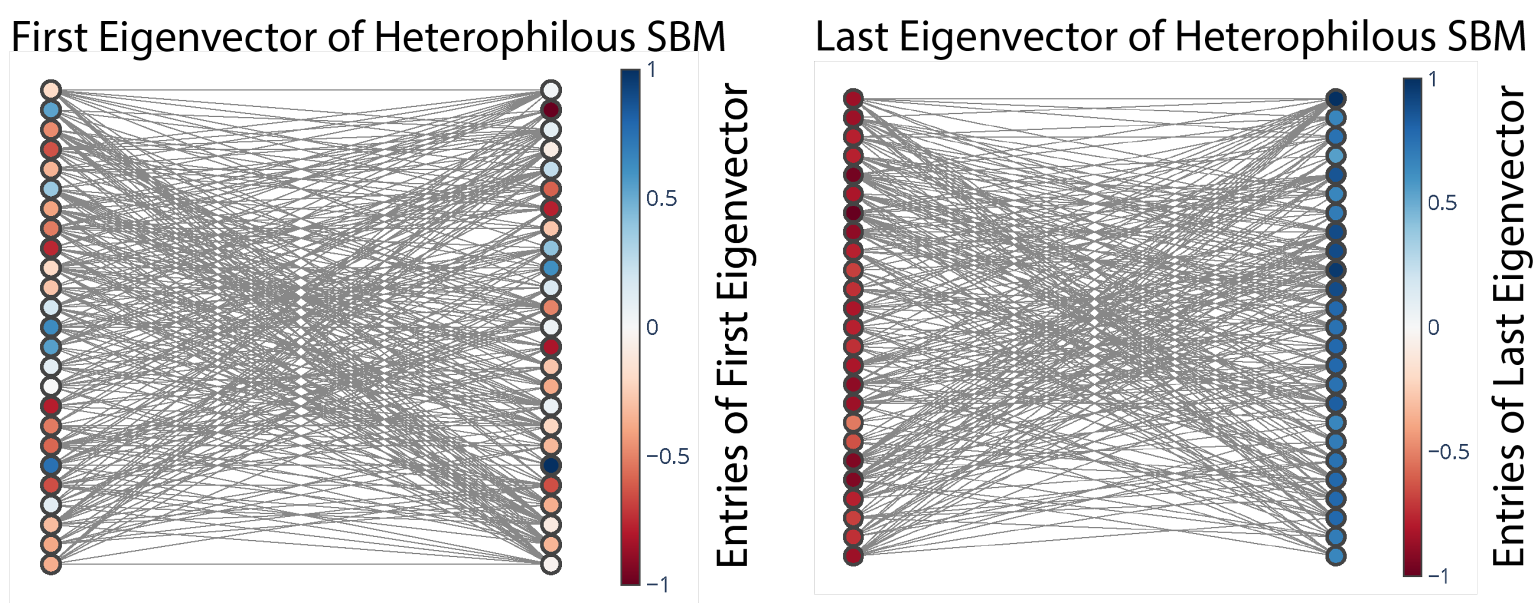
Transformers have led to tremendous progress in NLP and CV. How can we design transformers *for arbitrary graphs*?

Laplacian PEs, the **first k** eigenvectors of the Laplacian, *generalize* sin/cos to graphs

$$L = D - A \rightarrow L = U^\top \Lambda U, \quad P = U[:, :k]$$



While LPEs are beneficial for **homophilous graphs**, we show they are not beneficial for **heterophilous graphs.**



2. Learnable Laplacian PEs (LLPEs)

Intuition for learning LPEs: *Leverage the* ***full*** *eigenvector matrix* **U** *along with their corresponding eigenvalues* **Λ**

We learn mapping h: $[0, 2] \rightarrow \mathbb{R}$ where $h(\lambda_i)$ is eigenvector i's importance

$$\mathbf{P}_{\text{LLPE}} = \mathbf{U}\mathbf{W}_{\text{LLPE}}, \quad \mathbf{W}_{\text{LLPE}} = \begin{pmatrix} h(\lambda_0; \boldsymbol{\theta}_0) & \cdots & h(\lambda_0; \boldsymbol{\theta}_d) \\ \vdots & \ddots & \vdots \\ h(\lambda_n; \boldsymbol{\theta}_0) & \cdots & h(\lambda_n; \boldsymbol{\theta}_d) \end{pmatrix},$$

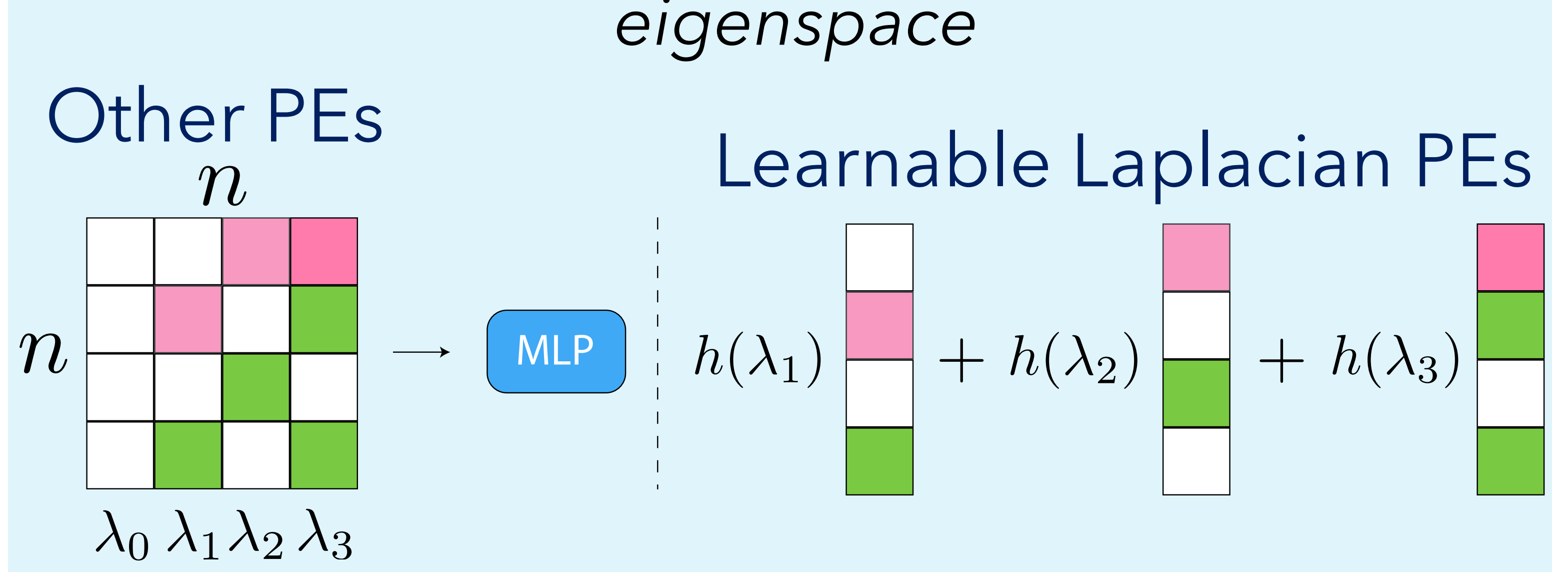
where $\boldsymbol{\theta}_j$ parametrizes h. We then set h as a truncated **Chebyshev series**,

$$h(\lambda_i; \boldsymbol{\theta}_j) = \sum_{m=0}^{M} \boldsymbol{\theta}_j[m] \cdot T_m(\tilde{\lambda}_i), \quad T_m(\tilde{\lambda}_i) = \cos(m \cdot \arccos(\tilde{\lambda}_i))$$

where $\boldsymbol{\theta}_j$ are learnable Chebyshev weights.

LLPEs Exhibit Tighter Generalization

LLPEs operate on the *significantly smaller Laplacian spectrum*, as opposed the *full eigenspace*



LLPEs Can Approximate Graph Distances

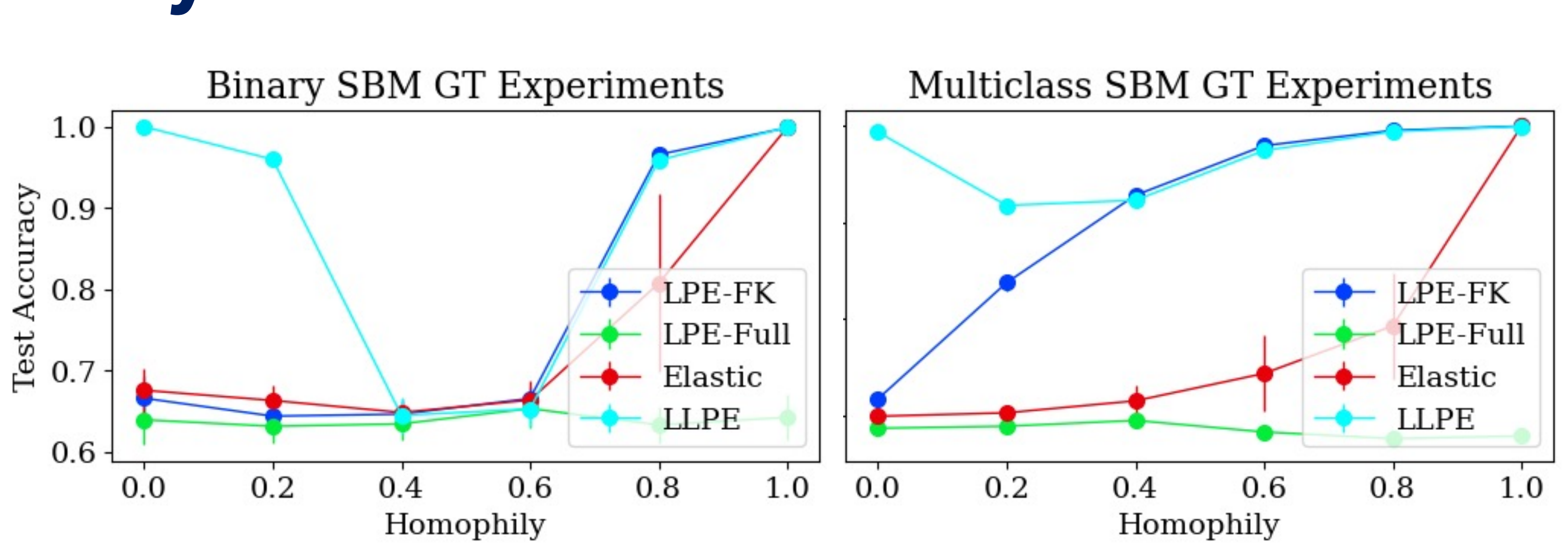
LLPE can approximate *distances on graphs*, including random walks, heat kernels, and diffusion --- formally, distances defined

$$f_r(i, j)^2 = \sum_{k=1}^{n} r(\lambda_k)(\mathbf{u}_k[i] - \mathbf{u}_k[j])^2.$$

where i and j are nodes and r: $[0, 2] \rightarrow \mathbb{R}^+$.

3. Experimental Results

LLPE identifies relevant graph structure on synthetic SBMs



	Tolokers	Cora-Full	Computers	Cora
# Nodes $\vert V\vert$	11.7K	19.7K	13.7K	2.7K
# Edges $\vert E\vert$	519K	126.8K	491.7K	5.4K
Homophily	0.17	0.50	0.70	0.75
ElasticPE	73.23 ± 2.90	57.92 ± 1.39	85.28 ± 0.86	74.67 ± 1.68
SAN-PE	78.42 ± 1.15	60.25 ± 0.60	85.36 ± 0.55	73.16 ± 1.40
SignNet	73.96 ± 0.86	60.28 ± 0.59	85.09 ± 0.68	72.66 ± 2.28
RWSE	74.09 ± 0.69	60.07 ± 0.87	85.05 ± 0.83	74.27 ± 2.28
LLPE (ours)	**80.85 ± 0.83**	**61.02 ± 0.60**	**87.83 ± 0.45**	**80.83 ± 1.33**

Across a range of homophily LLPEs outperform existing PEs on complex real graphs

We propose a **new learnable Laplacian position encoding** that helps capture graph structure in both homophilous and heterophilous settings by leveraging the **full spectrum of the graph Laplacian**, representing a significant step in developing data-driven PEs that capture complex graph structures

Funding: This work was supported by a graduate fellowship from the Department of Energy DE-SC0023112