Learning Laplacian Positional Encodings for Heterophilous Graphs

Michael Ito[1] **Jiong Zhu**[1] **Dexiong Chen**[2] **Danai Koutra**[1] **Jenna Wiens**[1]
University of Michigan[1], Max Planck Institute of Biochemistry[2]

Abstract

In this work, we theoretically demonstrate that current graph positional encodings (PEs) are not beneficial and could potentially hurt performance in tasks involving heterophilous graphs, where nodes that are close tend to have different labels. This limitation is critical as many real-world networks exhibit heterophily, and even highly homophilous graphs can contain local regions of strong heterophily. To address this limitation, we propose Learnable Laplacian Positional Encodings (LLPE), a new PE that leverages the full spectrum of the graph Laplacian, enabling them to capture graph structure on both homophilous and heterophilous graphs. Theoretically, we prove LLPE's ability to approximate a general class of graph distances and demonstrate its generalization properties. Empirically, our evaluation on 12 benchmarks demonstrates that LLPE improves accuracy across a variety of GNNs, including graph transformers, by up to 35% and 14% on synthetic and real-world graphs, respectively. Going forward, our work represents a significant step towards developing PEs that effectively capture complex structures in heterophilous graphs.

1 INTRODUCTION

In node classification, graph positional encodings (PEs) improve the discriminative performance of graph neural networks (GNNs) by injecting them with valuable positional information, allowing them to better capture the positions of nodes within the graph (You et al., 2019; Dwivedi et al., 2022; Rampášek et al., 2022). In particular, PEs aim to encode distance such that if two nodes are close in graph distance, their PEs should also be close. While it has been shown that PEs are beneficial for homophilous graphs, where nodes of the same label tend to be close, we find that they are not as beneficial in heterophilous graphs, where nodes that are close tend to have different labels. To demonstrate this intuition, consider a homophilous graph of two clusters where the cluster assignment is the node label (Figure 1). Here, Laplacian PEs that encode closeness by leveraging the eigenvectors of the Laplacian are beneficial since nodes that are close have the same label. Now consider a heterophilous bipartite graph where node labels are the sets that do not connect to each other. Here, nodes that are close have opposing labels, and thus Laplacian PEs no longer capture the relevant structural information.

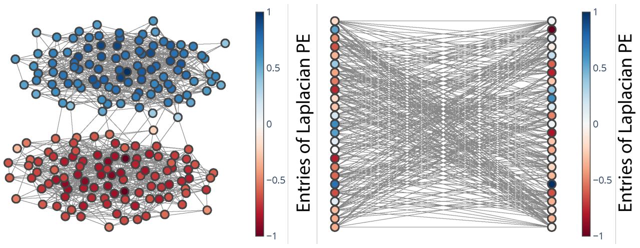


Figure 1: LPEs on homophilous/heterophilous SBMs.

Notably, real-world graphs exhibit a spectrum of homophily-heterophily, and while many real-world graphs exhibit strong homophily, there are also many graphs that exhibit strong heterophily such as protein networks where different amino acids form connections and transaction networks where fraudsters are more likely to connect to accomplices (Zhu et al., 2020). Furthermore, we find that even if a graph is highly homophilous globally, it may still contain local regions that are highly heterophilous as seen in some social networks, where users can form diverse connections (Newman, 2018; Loveland et al., 2023). These observations have led to significant research efforts in understanding the impact of heterophily on GNNs and designing models that can generalize to both homophilous and heterophilous graphs (Zhu et al., 2020; Chien et al., 2021; Bo et al., 2021; Zhu et al., 2021;

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

Yan et al., 2022; Ma et al., 2022; Luan et al., 2024; Zhu et al., 2024; Ito et al., 2025). In this work, given the spectrum of homophily-heterophily in real data, we investigate the *impact of PEs on node classification in both homophilous and heterophilous graph settings*. While orthogonal to the design of heterophilous GNNs, our work can improve these models by augmenting them with our proposed PEs.

In our theoretical analysis, we focus primarily on Laplacian PEs (LPEs) due to their importance and prevalence in GNN designs (Rampášek et al., 2022; Dwivedi and Bresson, 2021; Kreuzer et al., 2021; Lim et al., 2022; Wang et al., 2022). Our analysis reveals that LPEs can fail to capture relevant structures on heterophilous graphs since they do not include the full spectrum of the Laplacian. While other learnable PEs such as RWSE (Dwivedi et al., 2022) and SAN-PE (Kreuzer et al., 2021) can be extended to the full spectrum, empirically we find that they often struggle with high dimensional and/or noisy eigenspaces and as a result can fail to capture relevant structures (i.e., identify the relevant eigenvectors of the Laplacian). To address this gap, we introduce Learnable Laplacian Positional Encodings (LLPEs), a new PE that leverages the relevant eigenvectors and eigenvalues of the graph Laplacian. We demonstrate LLPE's theoretical expressivity by showing its ability to approximate a general class of graph distances, enabling it to capture relevant graph structure on arbitrary graphs. We further prove that it exhibits better statistical generalization properties compared to other designs. Empirically, we demonstrate the effectiveness of LLPE by showing that it improves performance for a variety of GNNs, including graph transformers, in comparison to other PEs across 12 homophilous and heterophilous benchmarks. In summary, we make the following contributions.

- **Identification of Limitations.** We show that LPEs and popular learnable PEs can fail to capture relevant graph structure in heterophilous graphs since they do not scale to high dimensions.
- **New Position Encodings for Homophily and Heterophily.** We introduce Learnable Laplacian Positional Encodings that capture relevant structure on homophilous and heterophilous graphs by leveraging the full spectrum of the Laplacian.
- **Theoretical Insights.** We show theoretically that LLPE can approximate important notions of graph distance and that it exhibits the best statistical generalization among other designs.
- **Empirical Analysis.** Empirically, we show that LLPE improves performance over current popular encodings on 12 node classification benchmarks.

2 BACKGROUND

In this section, we first provide background on node classification since it is the main task we consider throughout our work. We then provide an overview of message-passing GNNs and their relationship to homophily and heterophily. Next, we provide background on graph PEs, which are key components for many GNNs in both node and graph classification. Lastly, we provide background on graph transformers, a GNN architecture that has achieved high performance across a variety of graph benchmarks due to the design of effective PEs (Rampášek et al., 2022; Dwivedi and Bresson, 2021; Kreuzer et al., 2021; Chen et al., 2022).

2.1 Node Classification

A graph is defined $G = (V, \mathbf{A}, \mathbf{X}, \mathbf{y})$, where V is the node set, $\mathbf{A} \in \mathbb{R}^{|V|\times|V|}$ is the adjacency matrix, $\mathbf{X} \in \mathbb{R}^{|V|\times d}$ is the node feature matrix, and $\mathbf{y} \in \mathbb{R}^{|V|}$ is node label vector. For node $i \in V$, we denote its feature vector and label as $\mathbf{x}_i$ and y_i, respectively. Given a single G with a random sample of node representations $\mathbf{X}_{\text{tr}} = \{\mathbf{x}_0, \ldots, \mathbf{x}_{n_{\text{tr}}}\}$ and their labels $\mathbf{y}_{\text{tr}} = \{y_0, \ldots, y_{n_{\text{tr}}}\}$, the node classification task is to learn a classifier f such that the expected loss $\mathbb{E}[\mathscr{L}(f(\mathbf{x}, \mathbf{A}), y)]$ is minimized, where $\mathscr{L}$ is some loss function. Given f, one can then label the remaining nodes in G. In our work, we focus on the transductive setting where no new nodes are added at inference.

2.2 Homophily, GNNs and Graph PEs

Graph Homophily. The homophily ratio is the probability that a node forms an edge with another node with the same label, and a graph is considered homophilous if it has a high homophily ratio and heterophilous otherwise (Zhu et al., 2020).

Message-Passing GNNs. Many GNNs follow a message-passing scheme, where each layer updates each node's representation by aggregating the representations of its immediate neighbors (Gilmer et al., 2017). Most message-passing GNNs differ in the aggregation step, and popular choices include the symmetric normalized mean (Kipf and Welling, 2017), attention (Veličković et al., 2018), and sum pooling (Xu et al., 2019). Intuitively, when many nodes are connected to other nodes with the same label, message-passing is beneficial, and different classes will be well-separated in feature space after message passing. On the other hand, when many nodes are connected to nodes of different labels, message-passing may be detrimental and result in significant overlap between different classes in feature space. Notions of homophily attempt to capture this phenomenon (Mironov and Prokhorenkova, 2024).

In recent years, new GNN designs have been proposed in order to make these models effective for heterophilous graphs, without compromising performance over homophilous graphs, including separation of ego- and neighbor- embeddings (Zhu et al., 2020), learning from neighbors at various distances (Abu-El-Haija et al., 2019), learned degree corrections (Yan et al., 2022), and more (Chien et al., 2021; Bo et al., 2021; Zhu et al., 2021; Yan et al., 2022; Ma et al., 2022; Luan et al., 2024). Despite these advances, GNNs are known to exhibit various limitations, as they lack the ability to encode the position of a node within a graph (You et al., 2019). Thus, they are commonly augmented with PEs (Kreuzer et al., 2021).

Transformer-based GNNs. Graph transformers (GTs) are a new class of GNN, replacing message-passing in favor of global attention (Chen et al., 2022; Ma et al., 2023). Due to their departure from message-passing, GTs are believed to overcome many of the limitations of message-passing and as a result have achieved high performance on a variety of benchmarks (Kreuzer et al., 2021; Rampášek et al., 2022). Moreover, a crucial component of GTs is the choice of PE since it is the main component of the architecture that captures graph structure (Dwivedi and Bresson, 2021).

Graph Positional Encodings. In the context of learning from graphs, PEs are typically added or concatenated with node features. In our theoretical analysis, we focus on Laplacian positional encoding (LPE) defined as the k eigenvectors of the graph Laplacian that correspond to its k smallest eigenvalues. LPEs have been shown to improve GNNs both theoretically and empirically by encoding useful notions of node positional information.

3 ANALYSIS OF LPEs

In this section, we explore LPEs in capturing relevant graph structure on stochastic block models (SBMs) (Holland et al., 1983). In our analysis, we show that LPEs do capture the relevant graph structure in homophilous SBMs but fail to do so in heterophilous SBMs. Instead, the eigenvectors corresponding to the largest eigenvalues capture the relevant graph structure. For the remainder of our analysis, we assume that the eigenvalues are sorted from smallest to largest, and the corresponding eigenvectors are sorted similarly. Our analysis both highlights the limitations of LPEs and sheds light on potential solutions to these limitations.

3.1 Community Detection on SBMs

We consider $G(n, k, p, q)$, a multiclass SBM, where the set of n vertices are divided into k communities of size $\frac{n}{k}$. Edges between nodes in the same community are sampled with probability p, while edges between nodes in different communities are sampled with probability q. The community detection task is to recover the community labels for all nodes given a single realization of G. If LPEs can recover the communities, they effectively capture the relevant graph structure since the communities provide the best notion of node position within the SBM. Importantly, an SBM is homophilous when $p > q$ and heterophilous otherwise. Thus, by specifying conditions on p and q, we can analyze LPEs along different homophilous and heterophilous graphs.

3.2 Laplacian Encodings on Multiclass SBMs

To put our theoretical results into context, we first restate the following well-known result from the network community detection literature that states in a homophilous multiclass SBM, the first k eigenvectors of the graph Laplacian recovers the node communities up to a small error (Abbe, 2018).

Theorem 3.1 (Abbe (2018))**.** *Let* $\mathbf{A}$ *and* $\mathbf{L}$ *be the adjacency and Laplacian matrix drawn from the stochastic block model* $G(n, k, p, q)$*. Assume* $p \gg q$ *and* $min(d_i) \geq Cln(n)$ *where* C *is an appropriately large constant. Then, with high probability, the nonzero entries along the rows of the* ***first nontrivial*** $k-1$ ***eigenvectors*** *of* $\mathbf{L}$ *correctly recover the true communities up to an orthogonal transformation with at most* $\mathcal{O}(k^{\frac{3}{2}})$ *misclassified nodes.*

Since nearby nodes have the same label, LPEs capture relevant graph structure in homophilous SBMs. We now pose the same question for heterophilous SBMs where nearby nodes have different labels. In our next theorem, we demonstrate that on heterophilous SBMs the first k eigenvectors no longer recover the communities, and instead, the last k eigenvectors do.

Theorem 3.2. *Let* $\mathbf{A}$ *and* $\mathbf{L}$ *be the adjacency and Laplacian matrix drawn from* $G(n, k, p, q)$*. Assume* $q \gg p$ *and* $min(d_i) \geq Cln(n)$ *where* C *is an appropriately large constant. Then, with high probability, the nonzero entries along the rows of the* ***last*** $k-1$ ***eigenvectors*** *of* $\mathbf{L}$ *correctly recover the true communities up to an orthogonal transformation with at most* $\mathcal{O}(k^{\frac{3}{2}})$ *misclassified nodes. Moreover, the first nontrivial* k *eigenvectors* ***do not*** *recover the true communities.*

We prove Theorem 3.2 in Appendix A.2. Since on heterophilous SBMs, nodes that are close have opposing labels, LPEs do not capture the relevant structure and in order to do so the last $k-1$ eigenvectors need to be leveraged. This result captures the same intuition as designs in heterophilous GNNs (Zhu et al., 2020) aiming to capture high frequency components of the graph by leveraging intermediate GNN representations.

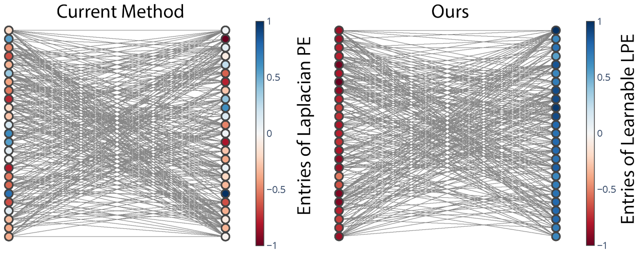


Figure 2: LPEs and LLPEs on heterophilous SBMs.

4 LEARNABLE LAPLACIAN ENCODINGS

Our investigation of LPEs finds that community structure in heterophilous graphs is not captured by the first k eigenvectors, but rather the last $k-1$. Since graphs can be homophilous or heterophilous, we propose *learning* which parts of the spectrum are important. In this section, we first introduce Learnable Laplacian Position Encodings (LLPE), a new PE that leverages the full spectrum of the graph Laplacian. We demonstrate LLPE's theoretical expressivity by showing it captures relevant graph structure on both homophilous and heterophilous graphs as well as general notions of graph distance for arbitrary graphs. We next show that LLPE exhibits the best statistical generalization among other design choices, justifying our designs. We then discuss an extension of LLPE to large graphs when the full eigendecomposition is infeasible.

4.1 Learnable Laplacian Position Encodings (LLPE)

Given the eigendecomposition of the graph Laplacian, $\mathbf{L} = \mathbf{U}^\top \mathbf{\Lambda} \mathbf{U}$, LPEs can be represented as follows: $\mathbf{P}_{\text{LPE}} = \mathbf{U}_k \mathbf{W}$, where $\mathbf{U}_k$ are the first k eigenvectors and $\mathbf{W}$ is a learnable linear projection matrix. As shown in Theorem 3.2 the first k eigenvectors may not capture heterophilous graph structure since heterophily may be captured in other parts of the spectrum. Current approaches apply MLPs or transformers to only the first k eigenvectors, and one simple approach is to extend these PEs to include the full eigenvector matrix $\mathbf{U}$ rather than only the first k eigenvectors. However, we find that this approach does not work well in practice since intuitively we expect that only a small number of eigenvectors are relevant, and the dimension of $\mathbf{U}$ can be very large especially for large graphs. We further note that LPEs do not explicitly leverage eigenvalue information. We construct LLPE to address these limitations.

The key insight in the design of LLPE is to leverage the full eigenvector matrix $\mathbf{U}$ along with the corresponding eigenvalues $\mathbf{\Lambda}$. By leveraging the eigenvalue information, we can learn which eigenvectors are important. More specifically, we learn a mapping $h : [0, 2] \to \mathbb{R}$ where $h(\lambda_i)$ represents eigenvector i's importance in the PE. Formally, LLPE can be defined as:

$$\mathbf{P}_{\text{LLPE}} = \mathbf{U}\mathbf{W}_{\text{LLPE}}, \text{ where} \tag{1}$$

$$\mathbf{W}_{\text{LLPE}} = \begin{pmatrix} h(\lambda_1; \boldsymbol{\theta}_1) & \cdots & h(\lambda_1; \boldsymbol{\theta}_d) \\ \vdots & \ddots & \vdots \\ h(\lambda_n; \boldsymbol{\theta}_1) & \cdots & h(\lambda_n; \boldsymbol{\theta}_d) \end{pmatrix}, \tag{2}$$

where $\mathbf{W}_{\text{LLPE}} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{\theta}_j$ is a weight vector that parametrizes h. Inspired by their use in spectral filtering (Hammond et al., 2011), we set h to be a truncated Chebyshev series. Thus, $h(\lambda_i; \boldsymbol{\theta}_j)$ has the form

$$h(\lambda_i; \boldsymbol{\theta}_j) = \sum_{m=0}^{M} \boldsymbol{\theta}_j[m] \cdot T_m(\tilde{\lambda}_i), \text{ where} \tag{3}$$

$$T_m(\tilde{\lambda}_i) = \cos(m \cdot \arccos(\tilde{\lambda}_i)), \tag{4}$$

where $\tilde{\lambda}$ are the normalized eigenvalues, M is a tunable hyperparameter, T_m is a Chebyshev polynomial of order m, and $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_d \in \mathbb{R}^M$ are learnable Chebyshev coefficients. To learn these parameters, we update them via backpropagation when training the GNN. We additionally regularize the l^1 and l^2 norms of the columns of $\mathbf{W}_{\text{LLPE}}$ during training to encourage sparse outputs. Figure 2 demonstrates an overview of LLPE's improvements on community detection over LPE on binary heterophilous SBMs.

4.2 Theoretical Expressivity of LLPE

In this section, we demonstrate LLPE's theoretical expressivity. We first show that LLPE can capture relevant structure on homophilous and heterophilous SBMs. We next demonstrate that LLPE can approximate general notions of graph distances, including ones defined with random walks, heat kernels, and diffusion maps. This shows that LLPE can recover relevant structure on arbitrary graphs. In the following proposition, we state the approximation result for SBMs.

Proposition 4.1. *Let* $\mathbf{A}$ *and* $\mathbf{L}$ *be the adjacency and Laplacian matrix drawn from* $G(n, k, p, q)$*. If* $p \gg q$ *or* $q \gg p$ *and* $min(d_i) \geq Cln(n)$ *where* C *is an appropriately large constant, then LLPE can correctly recover the true communities up to an orthogonal transformation with at most* $\mathcal{O}(k^{\frac{3}{2}})$ *misclassified nodes.*

We prove proposition 4.1 in Appendix B.2. Since LLPE leverages the full eigenvector matrix $\mathbf{U}$ and its corresponding eigenvalues $\mathbf{\Lambda}$, it can capture the graph structure on both homophilous and heterophilous SBMs. We next show that LLPE can further capture relevant notions of structure for arbitrary graphs. We begin by introducing a general class of functions on graphs.

Definition 4.2. Let G be an arbitrary graph. Define $f_r : V \times V \to \mathbb{R}$ as a function on G with respect to $r : [0, 2] \to \mathbb{R}^+$ such that $f_r(i, j)$ has the following form,

$$f_r(i,j)^2 = \sum_{k=1}^{n} r(\lambda_k)(\mathbf{u}_k[i] - \mathbf{u}_k[j])^2. \tag{5}$$

When r is monotonically decreasing in λ, Definition 4.2 generalizes a variety of distances on graphs. For example when $r(\lambda_k) = \frac{1}{\lambda_k^2}$, f_r is the commute time (Lovász, 1993), where $f_r(i, j)$ is the expected number of steps in a random walk starting at node i travelling to node j then travelling back to node i. When $r(\lambda_k) = e^{-2t\lambda_k}$ for some parameter t, f_r is the diffusion distance (Coifman and Lafon, 2006), where $f_r(i, j)$ is the number of paths of length t from node i to node j. Finally, when $r(\lambda_k) = \frac{1}{\lambda_k}$, f_r is the Biharmonic distance (Lipman et al., 2010) proposed to improve upon the commute time and diffusion distance.

In the cases of the commute time, diffusion distance, and Biharmonic distance, r acts as a low pass filter that amplifies the first eigenvectors and attenuates the last ones. As a result, nodes that are close will have small $f_r(i, j)$, while nodes that are far away will have large $f_r(i, j)$, effectively capturing a notion of distance. For homophilous graphs, these notions capture the relevant graph structure. On the other hand, if r is monotonically increasing in λ, then r acts as a high pass filter, and nodes far away will have small $f_r(i, j)$, while nodes that are close will have large $f_r(i, j)$. In this case, $f_r(i, j)$ captures the relevant graph structure on heterophilous graphs.

LLPE can approximate functions on graphs of the form of Definition 4.2 for any r, demonstrating that LLPE can capture relevant notions of graph structure. We prove Theorem 4.3 in Appendix B.2.

Theorem 4.3. *Let G be an arbitrary graph and f_r be a function on G of the form in Definition 4.2 for some r. LLPE can recover f_r such that for any nodes i and j, the l^2 distance between LLPE's encoding for nodes i and j approximates $f_r(i, j)$.*

4.3 Generalization Properties of LLPE

In this section, we discuss the statistical generalization of LLPE. Our main result derives the upper and lower bounds of LLPE's Rademacher complexity, indicating that generalization depends on the norms of the Chebyshev coefficients. Our analysis justifies our choice of h. In particular, defining h as a truncated Chebyshev series obtains better generalization over other approximating polynomials and existing PEs.

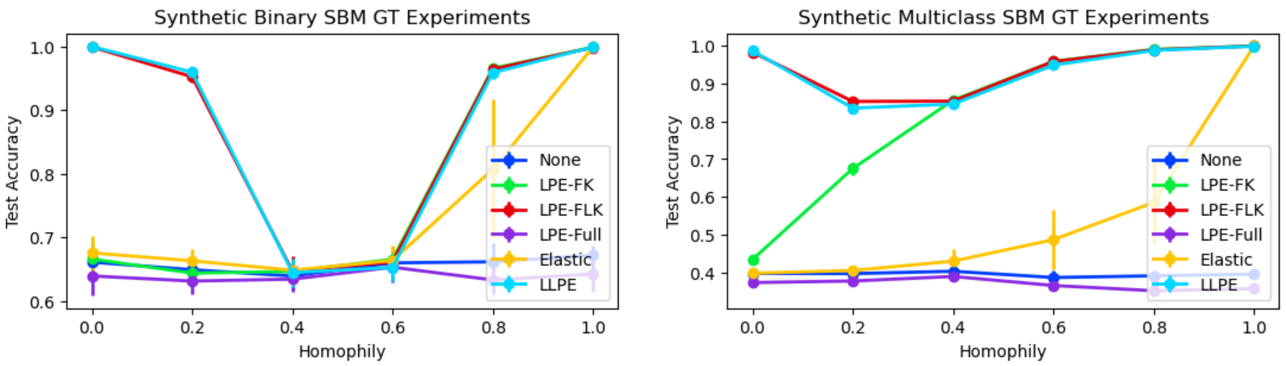
Theorem 4.4. *Let $\mathcal{H}_{LLPE} = \{\tilde{\lambda} \to \sum_{m=1}^{M} \theta_m \cdot \tilde{T}_m(\tilde{\lambda}) : \boldsymbol{\theta} \in \mathbb{R}^M, ||\boldsymbol{\theta}||_2 \leq C_{LLPE}\}$, where C_{LLPE} is some constant greater than 0, $\tilde{\lambda}$ denotes the normalized eigenvalues, and $\tilde{T}_m$ is the normalized Chebyshev polynomial of order m. Then, the empirical Rademacher complexity of the hypothesis class $\mathcal{H}_{LLPE}$ for a sample $S = (\lambda_1, \ldots, \lambda_n)$ admits the following upper and lower bounds:*

$$\frac{C_{LLPE}}{\sqrt{2n}} \leq \hat{\mathfrak{R}}_S(\mathcal{H}_{LLPE}) \leq \frac{\sqrt{2}C_{LLPE}}{\sqrt{n}} \tag{6}$$

We prove Theorem 4.4 in Appendix B.3. Theorem 4.4 tells us that the empirical Rademacher complexity scales with the upper bound of the l^2 norm of the Chebyshev coefficients. Thus, the bounds do not depend *explicitly* on the order M of the Chebsyshev series, but rather implicitly. As a result, LLPE obtains high expressivity and good generalization with large M, so long as the norm of the Chebyshev coefficients remains small. To prove this result, we leverage the fact that Chebyshev polynomials of order m obtain the minimum l^∞ norm among all other polynomials of order m. Importantly, the extension of other learnable PEs that rely on applying an MLP or transformer to the full eigenvector matrix $\mathbf{U} \in \mathbb{R}^{n \times n}$ result in model weights with dimension scaling in n, the number of nodes, which can be very large for large graphs. The dimension of LLPE's Chebyshev coefficients do not depend on n, but rather rely on M the number of terms in the sum (which is typically much smaller than n). This serves as a form of implicit regularization for large graphs, and provides theoretical insight into why LLPE is able to learn relevant eigenvectors in high-dimensional noisy graphs while other learnable PEs cannot.

4.4 Extension of LLPE to Large Graphs

Our discussion of LLPE thus far has assumed access to the full eigendecomposition of the Laplacian costing $\mathcal{O}(n^3)$. For small and medium-sized graphs, this is feasible to compute, but for large ones, it is not. We thus need a cheaper alternative for large graphs. When dealing with large graphs, motivated by our analysis of LPEs, instead of using all eigenvectors and eigenvalues, we propose to use the first and last k. These eigenvectors and eigenvalues can be obtained using the Arnoldi iteration algorithm (Arnoldi, 1951) with time complexity $\mathcal{O}(k^2 n)$, assuming a sparse graph. Thus, for large sparse graphs, the eigenvectors and eigenvalues at the ends of the spectrum are feasible to obtain. We empirically demonstrate the algorithm's efficiency in Appendix C.2. Furthermore, in the context of large graphs we demonstrate there is not a significant performance difference between LLPE and its approximate version for large graphs (Appendix C.4).



(a) Binary SBM Experiment

(b) Multiclass SBM Experiment

Figure 3: Mean and standard deviations (error bars) of all model-PE combinations on the synthetic SBMs. LLPE performs well across both high homophily and high heterophily, while LPE-FK does not.

5 EXPERIMENTAL RESULTS

We evaluate LLPE on homophilous and heterophilous benchmarks, aiming to answer the following research questions:

- **RQ1**: In a simple synthetic setting, where the data generation process and relevant eigenvectors are known, does LLPE and other PE baselines capture the relevant graph structure?
- **RQ2a**: On complex real-world small and medium graphs of varying homophily and heterophily, to what extent does LLPE and other PE baselines improve GNN performance?
- **RQ2b**: On real-world large graphs, does the approximate version of LLPE scale and does it improve GNN performance?

Implementation Details. To obtain the eigenvectors and eigenvalues of the Laplacian, we utilize a fast implementation of Arnoldi iteration readily available in SciPy and optimized with sparse matrix and vector operations. We compute the Chebyshev polynomials using efficient tensor operations in Pytorch. Finally, LLPE is trained in an end to end fashion with the GNN via gradient descent.

5.1 RQ1: Synthetic Data Experiments

Data Generation. We generate datasets according to binary and multiclass SBMs. The binary SBMs have $n = 2000$ nodes while the multiclass SBMs have $n = 5000$ nodes and $k = 5$ communities. For both SBMs, we generate multiple graphs by varying the homophily ratio. Given an SBM, node labels are determined by the node's community, and node features are generated by sampling a vector of Gaussian distributed features. We present the remaining details in Appendix D[1].

Base Model and PE. We test the performance of a GT (full) (Dwivedi and Bresson, 2021) with the PEs: (1) No PE, (2) LPE-FK which uses the first k eigenvectors, (3) LPE-FLK which uses the first and last k eigenvectors, (4) LPE-Full which leverages all the eigenvectors, (5) ElasticPE (Liu et al., 2023), and (6) our proposed encodings, LLPE. All PEs are concatenated to node features.

Training and Evaluation. We train all model-PE combinations by minimizing the negative log-likelihood on the train set. We select the best performing model across hyperparameters on the validation set for evaluation on the test set. We report the mean ± standard deviation of the accuracy on the test set across 10 random splits (60%/20%/20%), following Pei et al. (2020).

Results. In Figure 3, we present results on the synthetic experiments. On homophilous SBMs, PEs that leverage the first k eigenvectors perform well, while on heterophilous SBMs, PEs that leverage the last k eigenvectors perform well. Since LLPE learns the relevant eigenvectors, it captures structure in both homophilous and heterophilous SBMs. We note that LPE-FLK performs identically to LLPE since the first and last k eigenvectors are all that is necessary to capture the relevant structure on the SBMs. To test the approach on more complex structures, we conducted an additional experiment in Appendix C.1 on synthetic graphs generated with preferential attachment (Zhu et al., 2020). On the more complex graphs, LPE-FLK is no longer able to capture the relevant structure, while LLPE is able to capture the relevant structure in these settings.

[1] Code can be found at:
https://github.com/MLD3/LearningLaplacianPEs

Table 1: Mean ± standard deviation of model-PE test accuracy across 10 random splits on the small benchmarks. We highlight in **green** the best performing model-PE combination. We additionally count the number of test splits LLPE outperforms LPE-FK's performance indicated by (·/10).

Model	**PE/SE**	**Texas** **h = 0.00**	**Cornell** **h = 0.15**	**Cora-ML** **h = 0.74**	**Cora** **h = 0.75**	**Photo** **h = 0.76**	**Avg.** **Rank**
MLP	No PE	82.64 ± 6.80	75.19 ± 5.84	73.05 ± 2.29	66.47 ± 2.60	90.19 ± 1.42	2.8
	LPE-FK	78.93 ± 6.50	77.33 ± 5.23	71.57 ± 3.48	64.59 ± 3.58	89.43 ± 0.80	4.2
	LPE-FLK	81.60 ± 8.96	74.13 ± 5.12	72.12 ± 3.85	63.93 ± 2.46	90.22 ± 0.73	4.0
	LPE-Full	79.22 ± 7.52	74.91 ± 5.90	70.45 ± 5.40	62.30 ± 2.08	88.77 ± 0.98	6.4
	ElasticPE	80.29 ± 6.01	75.16 ± 5.07	80.39 ± 3.08	63.26 ± 2.50	88.07 ± 0.98	4.8
	RWSE	81.05 ± 5.81	75.46 ± 7.10	70.58 ± 3.34	62.40 ± 2.59	89.22 ± 1.63	4.8
	LLPE (ours)	**84.82 ± 6.05 (8/10)**	**77.60 ± 4.61 (2/10)**	**80.99 ± 1.67 (10/10)**	**78.62 ± 1.68 (10/10)**	**92.61 ± 0.74 (10/10)**	**1.0**
SAGE	No PE	80.29 ± 7.13	73.34 ± 3.51	87.92 ± 1.30	87.61 ± 1.20	**95.32 ± 0.38**	**2.8**
	LPE-FK	80.29 ± 6.60	72.81 ± 3.24	87.49 ± 1.43	86.86 ± 0.82	95.12 ± 0.38	5.0
	LPE-FLK	78.70 ± 7.96	73.08 ± 4.43	87.24 ± 1.50	86.84 ± 1.14	95.21 ± 0.51	5.8
	LPE-Full	78.95 ± 6.63	74.42 ± 5.08	87.24 ± 1.82	87.08 ± 0.94	95.00 ± 0.42	5.2
	ElasticPE	80.57 ± 7.74	**77.59 ± 3.48**	87.64 ± 1.12	87.26 ± 0.91	95.04 ± 0.52	3.2
	RWSE	81.36 ± 6.79	73.85 ± 4.34	87.62 ± 1.57	87.10 ± 0.70	95.21 ± 0.51	3.0
	LLPE (ours)	**83.99 ± 5.12 (6/10)**	72.28 ± 6.21 (4/10)	**88.17 ± 1.52 (7/10)**	**88.28 ± 1.01 (10/10)**	95.12 ± 0.43 (5/10)	3.0
GT (full)	No PE	84.52 ± 5.97	75.70 ± 7.25	78.52 ± 1.35	73.86 ± 2.05	91.82 ± 0.58	5.8
	LPE-FK	84.80 ± 4.92	76.50 ± 8.85	78.73 ± 1.81	73.99 ± 2.32	92.00 ± 0.63	4.8
	LPE-FLK	**85.61 ± 5.44**	78.66 ± 5.61	78.00 ± 2.28	73.82 ± 2.28	91.61 ± 0.54	4.8
	LPE-Full	85.33 ± 6.18	78.89 ± 6.43	77.66 ± 1.79	70.84 ± 2.18	91.53 ± 0.43	6.4
	ElasticPE	85.06 ± 4.81	78.11 ± 5.90	**85.48 ± 1.15**	74.67 ± 1.68	91.70 ± 0.50	3.6
	SAN-PE	82.15 ± 6.20	78.40 ± 5.77	78.40 ± 1.36	73.16 ± 1.40	91.19 ± 0.45	6.8
	SignNet	84.82 ± 6.48	77.30 ± 6.57	78.15 ± 2.05	72.66 ± 2.46	91.70 ± 0.60	6.4
	RWSE	85.60 ± 9.16	**79.18 ± 6.27**	78.10 ± 1.54	74.27 ± 2.28	91.55 ± 0.77	4.0
	LLPE (ours)	85.34 ± 6.44 (4/10)	78.39 ± 5.94 (6/10)	84.50 ± 1.25 (10/10)	**80.83 ± 1.33 (10/10)**	**94.34 ± 0.53 (10/10)**	**2.4**

Table 2: Mean ± standard deviation of model-PE test accuracy (AUROC for Tolokers) across 10 random splits on the medium benchmarks. We follow the same conventions as in Table 1.

Model	**PE/SE**	**Amazon-ratings** **h = 0.12**	**Tolokers** **h = 0.17**	**Cora-full** **h = 0.50**	**Computers** **h = 0.70**	**Avg.** **Rank**
SAGE	No PE	43.84 ± 0.64	82.64 ± 0.79	**68.68 ± 0.63**	91.02 ± 0.40	2.75
	LPE-FK	43.88 ± 1.17	82.56 ± 0.55	67.91 ± 0.78	91.02 ± 0.44	4.00
	ElasticPE	42.84 ± 0.39	76.89 ± 2.52	66.87 ± 1.01	90.05 ± 0.53	6.50
	SAN-PE	42.83 ± 0.78	80.37 ± 1.55	66.86 ± 0.61	90.97 ± 0.45	6.50
	SignNet	43.48 ± 0.97	82.41 ± 0.54	68.24 ± 0.46	91.04 ± 0.44	3.75
	RWSE	43.97 ± 1.21	82.43 ± 0.54	68.10 ± 0.73	91.06 ± 0.45	3.00
	LLPE (ours)	**45.56 ± 1.14 (10/10)**	**83.46 ± 0.69 (10/10)**	68.12 ± 0.68 (5/10)	**91.08 ± 0.29 (6/10)**	**1.50**
GT (full)	No PE	38.92 ± 0.60	74.04 ± 0.92	60.71 ± 0.70	85.13 ± 0.87	3.75
	LPE-FK	38.59 ± 0.59	74.21 ± 0.63	59.33 ± 0.94	85.19 ± 0.74	4.75
	ElasticPE	30.08 ± 5.48	73.23 ± 2.90	57.92 ± 1.39	85.28 ± 0.86	6.00
	SAN-PE	38.93 ± 0.60	78.42 ± 1.15	60.25 ± 0.60	85.36 ± 0.55	2.50
	SignNet	38.61 ± 0.51	73.96 ± 0.86	60.28 ± 0.59	85.09 ± 0.68	5.00
	RWSE	38.82 ± 0.52	74.09 ± 0.69	60.07 ± 0.87	85.05 ± 0.83	5.00
	LLPE (ours)	**39.80 ± 0.62 (10/10)**	**80.85 ± 0.83 (10/10)**	**61.02 ± 0.60 (9/10)**	**87.83 ± 0.45 (10/10)**	**1.00**

Table 3: Mean ± standard deviation of test accuracy (AUROC for Questions) on the large benchmarks.

Model	**PE/SE**	**Penn94** **h = 0.03**	**Questions** **h = 0.08**
SAGE	No PE	72.54 ± 0.51	74.77 ± 1.16
	LPE-FK	72.71 ± 0.70	74.94 ± 1.18
	LLPE (large)	**72.79 ± 0.45 (5/10)**	**77.84 ± 1.29 (10/10)**

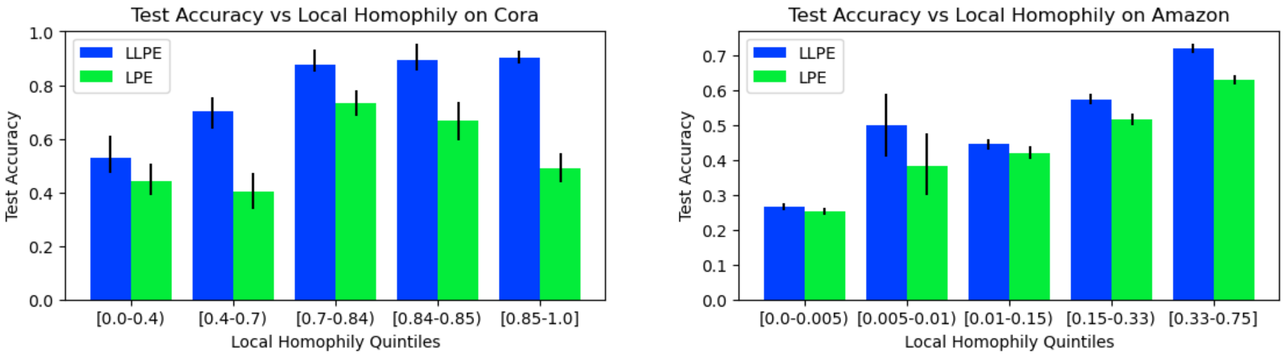
5.2 RQ2a/b: Real-world Data Experiments

Below, we describe our base models and PEs tested, training and evaluation procedure, and results for our real-world experiments.

Datasets. We evaluate on 12 homophilous and heterophilous node classification datasets, dividing them into small benchmarks ranging from $300 - 8000$ nodes (Yang et al., 2016; Bojchevski and Günnemann, 2018; Shchur et al., 2018; Pei et al., 2020), medium benchmarks ranging from $10000 - 25000$ nodes (Platonov et al., 2023), and large benchmarks ranging from $40000 - 50000$ nodes (Lim et al., 2021). For each dataset we report the class homophily (Lim et al., 2021).

Base Models and PEs. We test the following base models: (1) MLPs, (2) GTs (Dwivedi and Bresson, 2021), and (3) GraphSage (Hamilton et al., 2017). We then test the following PEs: (a) No PE, (b) LPE-FK (Dwivedi and Bresson, 2021), (c) LPE-FLK, (d) LPE-Full, (e) ElasticPE (Liu et al., 2023), (f) SignNet (Lim et al., 2022), (g) SAN-PE (Kreuzer et al., 2021), (h) RWSE (Dwivedi et al., 2022), and (i) our proposed encodings, LLPE.

Training and Evaluation. We follow the same training and evaluation procedure in Section 5.1. Importantly, in order to make fair comparisons, we evaluate each PE across a wide range of k eigenvectors and



(a) Accuracy for local homophily quintiles in Cora.

(b) Accuracy for local homophily quintiles in Amazon.

Figure 4: Performance of GTs with LPE-FK and LLPE across local node homophily quintiles on Cora and Amazon-ratings. Error bars are based on 95% bootstrapped confidence intervals.

eigenvalues, letting $k \in [8, \text{all}]$ during hyperparameter selection, following Kreuzer et al. (2021); Lim et al. (2022). Generally, we find that most PEs perform best under smaller choices of k as opposed to larger ones. We present the remaining details in Appendix D.

Results. Across small graph datasets, LLPE obtains the best average rank of 2.13, while the second best ranked PE obtains an average rank of 3.80 (Table 1). Moreover, for 7/15 model and dataset combinations, LLPE outperforms LPE-FK 10/10 times across the test splits. In our medium datasets, we find similar results and LLPE obtains the best average rank of 1.25 in comparison to the second best rank of 3.25 (Table 2). For 5/8 model-PE and dataset combinations, LLPE outperforms LPE-FK 10/10 times. In our large datasets, the approximate version of LLPE for large graphs outperforms LPE-FK and No-PE on both benchmarks (Table 3).

In our real-world experiments, we find that LPE-FLK does not capture the relevant structure, since real-world graphs require leveraging more intricate combinations of the eigenvectors. In many cases, learnable PEs such as SignNet and SAN-PE perform no better than LPE-FK such as in Amazon-ratings, Tolokers, and Cora, suggesting that existing learnable PEs may not identify the relevant eigenvectors in graphs of 1000s of nodes. In contrast, when given all eigenvectors where n is as large as 25,000 LLPE does identify relevant eigenvectors as it consistently obtains higher performance compared to LPE-FK across many datasets. On large graphs of size 50,000, LLPE (large) is also able to identify relevant eigenvectors, improving performance over the two baselines.

Although LLPE obtains the best ranking across datasets, it exhibits limitations. In particular, LLPE requires tuning multiple hyperparameters, including the order M of the Chebyshev polynomials and the number of eigenvectors k on large datasets. We find that performance can degrade under certain hyperparameter selections. In particular, our approach is sensitive to small choices of M, leading to low approximation capabilities, and small choices of k where too few eigenvectors are included in the PE (Appendix C).

5.3 Sensitivity Analysis of LLPE

Interestingly, on small datasets LLPE's performance gains are larger for homophilous graphs. We investigate this result and find that LLPE's gains on homophilous graphs can be attributed to higher performance in local regions of high heterophily within the graph. In Figure 4, we measure the performance of GT with LLPE and LPE-FK on Cora and Amazon across quintiles of local node homophily, homophily measured at the node level. Although Cora has high global homophily, it also contains a large portion of nodes that exhibit low local homophily (Figure 4(a)). On these nodes, LLPE leads to large performance improvements in comparison to LPE-FK, thus demonstrating that LLPE can lead to performance improvements on graphs that are globally homophilous yet also contain local regions of heterophily. In Figure 4(b), we analyze Amazon, a heterophilous benchmark. Here, we find that most nodes exhibit low local homophily, and LLPE leads to performance improvements across these nodes, resulting in an increase in overall performance on Amazon.

6 RELATED WORK

In past work, researchers have analyzed the first k eigenvectors of the Laplacian in the context of SBMs, focusing on regularization (Le et al., 2015, 2017; Guédon and Vershynin, 2016), while others have analyzed the largest k eigenvectors of the adjacency matrix (Rohe et al., 2011; Joseph and Yu, 2016). Generally, spectral clustering is applied to these eigenvectors to extract communities from homophilous graphs. In contrast, we analyze the last k eigenvectors of the Laplacian in Section 3, demonstrating the connection between these eigenvectors and heterophily. We further propose a new PE leveraging the Laplacian's full spectrum rather than focusing on a subset.

Researchers have proposed many PEs that aim to improve LPEs with the addition of learnable components (see Appendix C.5 for more details). However, all of these approaches focus on the first k eigevenvectors since they assume homophilous graph structure. In contrast, we do not make such assumption and thus propose learning which eigenvectors are most relevant. While existing learnable PEs can be extended to the full spectrum of the Laplacian, we find that they do not capture relevant structure when provided the full spectrum since they are designed for small graphs containing 10-100 nodes and learning relevant eigenvectors is more challenging in large graphs of 1,000-50,000 nodes due to the high dimension of the eigenspace and increased noise. LLPE addresses this issue by leveraging learnable Chebyshev polynomials whose weights do not scale in n, resulting in better generalization.

Spectral GNNs (Defferrard et al., 2016; He et al., 2021; Chien et al., 2021; Wang and Zhang, 2022) also leverage polynomial processing of eigenvalues similar to LLPE. More specifically, given the eigendecomposition of the Laplacian, $\mathbf{L} = \mathbf{U}^\top \mathbf{\Lambda} \mathbf{U}$ and graph signal $\mathbf{x}$, the spectral GNN operation is $\mathbf{U}^\top g(\mathbf{\Lambda})\mathbf{U}\mathbf{x}$, where $g(\mathbf{\Lambda})$ are the processed eigenvalues that filter $\mathbf{x}$ in the spectral domain enabled by the graph Fourier transform and its inverse, $\mathbf{U}\mathbf{x}$ and $\mathbf{U}^\top \hat{\mathbf{x}}$, respectively. In contrast, LLPE enhances GNN performance by directly leveraging Laplacian eigenvectors and eigenvalues as learnable positional encodings. It learns linear combinations of eigenvectors by defining $\mathbf{U}\mathbf{W}$, where $\mathbf{W}$ is a matrix of learnable weights obtained by transformations on the eigenvalues. LLPE is then added to GNNs like message-passing neural networks or graph transformers as additional node positional information. Similar to how our work can improve heterophilous GNNs, our work can thus improve spectral GNNs by augmenting them with LLPE.

7 CONCLUSION

We present the first analysis of PEs on heterophilous benchmarks in node classification. Specifically, we demonstrate the limitations of popular PEs in capturing heterophily and propose a new PE, Learnable LPE. We demonstrate theoretically that LLPE captures homophily and heterophily by leveraging the full spectrum of the Laplacian. In our empirical analysis, we demonstrate that LLPE improves performance for a variety of GNNs across 12 node classification benchmarks. Our results indicate that many popular PEs do not capture heterophily, and thus going forward our work represents a significant step in developing data-driven PEs that capture complex structure on heterophilous graphs or heterophilous regions within the more prevalent class of homophilous graph datasets.

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0023112. It was also partially supported by National Science Foundation under Grant No. IIS 2212143, and in part by an award from the Carl Friedrich von Siemens from the Alexander von Humboldt Foundation. We also thank the anonymous reviewers and members of the MLD3 lab for their valuable feedback.

References

Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *International conference on machine learning*, 2019.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=wTTjnvGphYj`.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, 2022.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Advances in Neural Information Processing Systems*, 2020.

Mark Newman. *Networks*. Oxford university press, 2018.

Donald Loveland, Jiong Zhu, Mark Heimann, Benjamin Fish, Michael T. Schaub, and Danai Koutra. On performance discrepancies across local homophily levels in graph neural networks. In *Learning on Graphs Conference (LOG)*, volume 231 of *Proceedings of Machine Learning Research*. PMLR, 2023.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. *International conference on machine learning*, 2021.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. Graph neural networks with heterophily. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In *2022 IEEE International Conference on Data Mining*, 2022.

Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? In *International Conference on Learning Representations*, 2022.

Sitao Luan, Chenqing Hua, Qincheng Lu, Liheng Ma, Lirong Wu, Xinyu Wang, Minkai Xu, Xiao-Wen Chang, Doina Precup, Rex Ying, Stan Z. Li, Jian Tang, Guy Wolf, and Stefanie Jegelka. The heterophilic graph learning handbook: Benchmarks, models, theoretical analysis, applications and challenges, 2024. URL https://arxiv.org/abs/2407.09618.

Jiong Zhu, Gaotang Li, Yao-An Yang, Jing Zhu, Xuehao Cui, and Danai Koutra. On the impact of feature heterophily on link prediction with graph neural networks. In *Advances in Neural Information Processing Systems*, 2024.

Michael Ito, Danai Koutra, and Jenna Wiens. Understanding gnns and homophily in dynamic node classification. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. PMLR, 2025.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. In *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems*, 2021.

Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. In *International Conference on Learning Representations*, 2022.

Haorui Wang, Haoteng Yin, Muhan Zhang, and Pan Li. Equivariant and stable positional encoding for more powerful graph neural networks. In *International Conference on Learning Representations*, 2022.

Dexiong Chen, Leslie O'Bray, and Karsten Borgwardt. Structure-aware transformer for graph representation learning. In *International Conference on Machine Learning*, 2022.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, 2017.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Mikhail Mironov and Liudmila Prokhorenkova. Revisiting graph homophily measures. In *Learning on Graphs Conference (LOG)*, volume 269 of *Proceedings of Machine Learning Research*. PMLR, 2024.

Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *International Conference on Machine Learning*, 2019.

Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *International Conference on Machine Learning*, 2023.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Emmanuel Abbe. Community detection and stochastic block models: recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

László Lovász. Random walks on graphs. *Combinatorics, Paul erdos is eighty*, 2(1-46):4, 1993.

Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1): 5–30, 2006.

Yaron Lipman, Raif M Rustamov, and Thomas A Funkhouser. Biharmonic distance. *ACM Transactions on Graphics (TOG)*, 29(3):1–11, 2010.

Walter Edwin Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of applied mathematics*, 9(1): 17–29, 1951.

Renming Liu, Semih Cantürk, Olivier Lapointe-Gagné, Vincent Létourneau, Guy Wolf, Dominique Beaini, and Ladislav Rampášek. Graph positional and structural encoder. In *International Conference on Machine Learning*, 2023.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, 2016.

Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *International Conference on Learning Representations*, 2018.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. In *Advances in Neural Information Processing Systems Relational Representation Learning Workshop*, 2018.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: are we really making progress? In *International Conference on Learning Representations*, 2023.

Derek Lim, Felix Matthew Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Prasad Bhalerao, and Ser-Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *Advances in Neural Information Processing Systems*, 2021.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 2017.

Can M Le, Elizaveta Levina, and Roman Vershynin. Sparse random graphs: regularization and concentration of the laplacian. *arXiv preprint arXiv:1502.03049*, 2015.

Can M Le, Elizaveta Levina, and Roman Vershynin. Concentration and regularization of random graphs. *Random Structures & Algorithms*, 51(3):538–561, 2017.

Olivier Guédon and Roman Vershynin. Community detection in sparse networks via grothendieck's inequality. *Probability Theory and Related Fields*, 165 (3):1025–1049, 2016.

Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. 2011.

Antony Joseph and Bin Yu. Impact of regularization on spectral clustering. 2016.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, 2016.

Mingguo He, Zhewei Wei, Hongteng Xu, et al. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In *Advances in Neural Information Processing Systems*, 2021.

Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In *International Conference on Machine Learning*, 2022.

Gilbert W Stewart and Ji-guang Sun. Matrix perturbation theory. *(No Title)*, 1990.

Vladimir Rakocevic and Harald K Wimmer. A variational characterization of canonical angles between subspaces. *Journal of Geometry*, 78(1-2):122–124, 2003.

Yi Yu, Tengyao Wang, and Richard J Samworth. A useful variant of the davis–kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015.

Roberto Imbuzeiro Oliveira. Concentration of the adjacency matrix and of the laplacian in random graphs with independent edges. *arXiv preprint arXiv:0911.0600*, 2009.

Theodore J Rivlin. *Chebyshev polynomials.* Courier Dover Publications, 2020.

John C Mason and David C Handscomb. *Chebyshev polynomials.* Chapman and Hall/CRC, 2002.

Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. On the rademacher complexity of linear hypothesis sets. *arXiv preprint arXiv:2007.11045*, 2020.

Shouheng Li, Dongwoo Kim, and Qing Wang. Restructuring graph for higher homophily via adaptive spectral clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, 2021.

Jinyoung Park, Seongjun Yun, Hyeonjin Park, Jaewoo Kang, Jisu Jeong, Kyung-Min Kim, Jung-Woo Ha, and Hyunwoo J Kim. Deformable graph transformer. *arXiv preprint*, 2022.

Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. In *International Conference on Learning Representations*, 2023.

Dong Li, Biqing Qi, Junqi Gao, Huan Xiong, Bin Gu, and Xinquan Chen. Mpformer: Advancing graph modeling through heterophily relationship-based position encoding. *arXiv preprint*, 2024.

Erlin Pan and Zhao Kang. Beyond homophily: Reconstructing structure for graph-agnostic clustering. In *International Conference on Machine Learning*, 2023.

Adrián Arnaiz-Rodríguez, Ahmed Begga, Francisco Escolano, and Nuria Oliver. Diffwire: Inductive graph rewiring via the lov\'asz bound. In *Learning on Graphs*, 2022.

Ameya Velingker, Ali Sinop, Ira Ktena, Petar Veličković, and Sreenivas Gollapudi. Affinity-aware graph networks. In *Advances in Neural Information Processing Systems*, 2023.

Bohang Zhang, Shengjie Luo, Liwei Wang, and Di He. Rethinking the expressive power of gnns via graph biconnectivity. In *International Conference on Learning Representations*, 2023.

Qincheng Lu, Jiaqi Zhu, Sitao Luan, and Xiao-Wen Chang. Representation learning on heterophilic graph with directional neighborhood attention. *arXiv preprint*, 2024.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes**
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes**
   (b) Complete proofs of all theoretical results. **Yes**
   (c) Clear explanations of any assumptions. **Yes**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **No**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **Yes**
   (b) The license information of the assets, if applicable. **Not Applicable**
   (c) New assets either in the supplemental material or as a URL, if applicable. **Not Applicable**
   (d) Information about consent from data providers/curators. **Not Applicable**
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. **Not Applicable**
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable**
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable**

Supplementary Material

Contents

A PROOFS FOR LAPLACIAN POSITIONAL ENCODINGS

A.1 Background, Definitions, and Lemmas on Perturbation Theory and Graph Laplacians

Perturbation theory analyzes how a function changes when its input is subject to perturbations. For our purposes, the functions of interest are the eigenvalues and eigenvectors of the graph Laplacian. Formally, our setup will focus on the expected graph Laplacian $\mathbb{E}[\mathbf{L}]$ according to a graph model and the observed graph Laplacian $\mathbf{L} = \mathbb{E}[\mathbf{L}] + \mathbf{E}$ that we observe from data. Intuitively, $\mathbb{E}[\mathbf{L}]$ is the informative component of $\mathbf{L}$ since we can determine its eigenstructure while $\mathbf{E}$ is the noise or error in $\mathbf{L}$. Our main goal will be to quantify the distance between $\mathbf{L}$'s eigenvalues and eigenvectors in comparison to $\mathbb{E}[\mathbf{L}]$'s eigenvalues and eigenvectors.

When quantifying the distance between eigenvectors corresponding to simple eigenvalues, it is straightforward to compute distances, and we can use typical metrics defined on vectors. However, when generalizing to distances between eigenspaces it becomes much more challenging. In order to demonstrate this, consider the following example from Stewart and Sun (1990). We define matrices $\mathbf{A}$ and its perturbations $\mathbf{A}_1$ and $\mathbf{A}_2$ for some $\epsilon > 0$,

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{A}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1+\epsilon & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} 1 & \epsilon/2 & 0 \\ \epsilon/2 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{7}$$

First, notice that $\mathbf{A}$ has no unique eigenvectors corresponding to its nonzero eigenvalue. Any vector lying in the span of the standard basis vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ is an eigenvector of $\mathbf{A}$. Thus, depending on the choice of eigenvectors for $\mathbf{A}$, the perturbation bound could be large or small. Second, notice that $\mathbf{A}_1$ has the same eigenvectors of $\mathbf{A}$, while $\mathbf{A}_2$'s eigenvectors, $(1, 1, 0)$ and $(1, -1, 0)$, are very different in comparison to $\mathbf{A}$. Thus, depending on the nature of the perturbation, different eigenstructures may arise from $\mathbf{A}$. Despite the differences from the two perturbations and the choice of eigenvectors for $\mathbf{A}$ however, the eigenvectors of small perturbations of $\mathbf{A}$ will always span a space close to the *subspace* of the original eigenvectors. Since the subspace spanned by the eigenvectors is more stable than the eigenvectors themselves, we focus on bounding the distances between the subspaces. Specifically, we introduce notions of angles and distance on the set of l dimensional subspaces of $\mathbb{R}^n$.

Definition A.1. Let $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^n$ be two l dimensional subspaces, and let the columns of $\mathbf{X}$ and $\mathbf{Y}$ form orthonormal bases for $\mathcal{X}$ and $\mathcal{Y}$. The unitarily invariant metric ρ is defined, $\rho(\mathcal{X}, \mathcal{Y}) = \inf_{\mathbf{Q} \in O(l)} ||\mathbf{X} - \mathbf{Y}\mathbf{Q}||_F$, where $O(l)$ denotes the set of all $l \times l$ orthogonal matrices and $||\cdot||_F$ denotes the Frobenius norm.

We note two important properties of ρ. First, if $\mathcal{X} = \mathcal{Y}$, then there exists some unitary matrix $\mathbf{Q}$ such that $\mathbf{X} = \mathbf{Q}\mathbf{Y}$ and thus $\rho(\mathcal{X}, \mathcal{Y}) = 0$. The second is that ρ is *unitarily invariant.* Formally, for all $\mathbf{U} \in O(l)$, $\rho(\mathbf{U}\mathcal{X}, \mathbf{U}\mathcal{Y}) = \rho(\mathcal{X}, \mathcal{Y})$. That is, rotations do not change the distance between $\mathcal{X}$ and $\mathcal{Y}$.

In our analysis, it is also essential to generalize the notion of angles between vectors to angles between subspaces. We use the following variational definition from Rakocevic and Wimmer (2003).

Definition A.2. Let $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^n$ be two l dimensional subspaces, and let the columns of $\mathbf{X}$ and $\mathbf{Y}$ form orthonormal bases for $\mathcal{X}$ and $\mathcal{Y}$. Let $\mathcal{P}_{\mathcal{X}}$ and $\mathcal{P}_{\mathcal{Y}}$ be the orthogonal projectors onto $\mathcal{X}$ and $\mathcal{Y}$ and let $\frac{\pi}{2} \geq \sigma_0(\mathcal{P}_{\mathcal{X}}\mathcal{P}_{\mathcal{Y}}) \geq \cdots \geq \sigma_l(\mathcal{P}_{\mathcal{X}}\mathcal{P}_{\mathcal{Y}}) \geq 0$ be the singular values of $\mathcal{P}_{\mathcal{X}}\mathcal{P}_{\mathcal{Y}}$ ordered by decreasing magnitude. Then, the canonical angles between $\mathcal{X}$ and $\mathcal{Y}$ are the inverse cosine of the singular values, $\theta_k = \arccos(\sigma_k(\mathcal{P}_{\mathcal{X}}\mathcal{P}_{\mathcal{Y}}))$.

The following lemma elucidates the relationship between the metric ρ and the canonical angles,

Lemma A.3 (Stewart and Sun (1990))**.** *Let $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^n$ be two l dimensional subspaces, and let the columns of $\mathbf{X}$ and $\mathbf{Y}$ form orthonormal bases for $\mathcal{X}$ and $\mathcal{Y}$. Let ρ be as defined in Definition A.1 and let $\Theta = diag(\theta_0, \ldots, \theta_l)$ where θ_i is defined in Definition A.2. Then, $\rho(\mathcal{X}, \mathcal{Y}) \leq \sqrt{2}\,||sin\Theta||_F$*

Now, we restate a variant of the Davis-Kahan theorem which bounds the canonical angles between the eigenspaces of two Hermitian matrices in terms of the distance between the two matrices (Yu et al., 2015).

Theorem A.4 (Yu et al. (2015), sinΘ theorem)**.** *Let $\mathbf{X}$ and $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times n}$ be Hermitian matrices with eigenvalues $\lambda_1, \ldots, \lambda_n$ and $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n$ respectively. Fix $1 \leq r \leq s \leq p$, let $d = s - r + 1$ and $\mathbf{V} = (\mathbf{v}_r, \mathbf{v}_{r+1}, \ldots, \mathbf{v}_s) \in \mathbb{R}^{n \times d}$ and let $\tilde{\mathbf{V}} = (\tilde{\mathbf{v}}_r, \tilde{\mathbf{v}}_{r+1}, \ldots, \tilde{\mathbf{v}}_s) \in \mathbb{R}^{n \times d}$ have orthonormal columns satisfying $\mathbf{X}\mathbf{v}_j = \lambda_j \mathbf{v}_j$ and $\tilde{\mathbf{X}}\tilde{\mathbf{v}}_j = \tilde{\lambda}_j \tilde{\mathbf{v}}_j$ for $j = r, r+1, \ldots, s$. Let Θ be the diagonal matrix of canonical angles between $\mathbf{V}$ and $\tilde{\mathbf{V}}$. Assume $d \ll n$, then,*

$$||sin\Theta||_F \leq \frac{2d^{1/2}\left|\left|\mathbf{X} - \tilde{\mathbf{X}}\right|\right|}{min\{\lambda_{r-1} - \lambda_r, \lambda_s - \lambda_{s-1}\}} \tag{8}$$

A key result that we leverage in our analysis is the concentration of the Laplacian (Oliveira, 2009). Specifically, let $\mathbb{E}[\mathbf{L}]$ and $\mathbf{L}$ be the expected and observed graph Laplacians of graph G, respectively. We say the Laplacian concentrates about its expectation if $\mathbb{E}[\mathbf{L}]$ is close to $\mathbf{L}$ as measured by the operator norm $||\mathbb{E}[\mathbf{L}] - \mathbf{L}||$. Since $||\cdot||$ is the operator norm, concentration results of this nature imply a tight control of the eigenvectors of the Laplacian up to perturbations as given by the Davis-Kahan theorem. We will specifically be interested in stochastic block model graphs as defined by $G(n, p, q, k)$, where n is the number of nodes, k is the number of communities, p is the probability of two nodes from the same community forming an edge, and q is the probability of two nodes from different communities forming an edge. For simplicity, we assume the dense regime where we let the minimum node degree be $\min(d_i) = C\ln(n)$ for some constant C. This assumption is necessary for the concentration of the *unregularized* Laplacian since $\min(d_i) < C\ln(n)$ implies $||\mathbb{E}[\mathbf{L}] - \mathbf{L}|| \geq 1$. Note that concentration can be extended to sparse Laplacians with proper regularization as shown in (Le et al., 2015, 2017). Below, we formally restate the concentration of the graph Laplacian in the dense regime,

Theorem A.5 (Oliveira (2009), Concentration of the Laplacian)**.** *Let $G(n, p, q, k)$ be a stochastic block model, $\mathbb{E}[\mathbf{L}]$ the expected Laplacian of G, and $\mathbf{L}$ the observed Laplacian of G. Let $C \geq 0$ be a constant independent of n, p, q and k. If $min(d_i) \geq Cln(n)$, then for $n^{-c} \leq \delta \leq 1/2$,*

$$\mathbb{P}\left(||\mathbb{E}[\mathbf{L}] - \mathbf{L}|| \leq 14\sqrt{\frac{ln(4n/\delta)}{min(d_i)}} \right) \geq 1 - \delta \tag{9}$$

A.2 Laplacian Positional Encodings

With the tools introduced on perturbation theory and the concentration results for the Laplacian in the dense regime, we can prove our results for Laplacian positional encodings. We begin proving our results for heterophilous block models and show that our results and derivations can be extended to homophilous block models in a straightforward fashion. We begin with results on binary SBMs and later generalize our results to multiclass SBMs as in the main paper.

Proposition A.6. *Let $\mathbf{A}$ and $\mathbf{L}$ be the Adjacency and Laplacian matrix drawn from the stochastic block model $G(n, 2, p, q)$. Assume $q \gg p$ and $min(d_i) \geq Cln(n)$ where C is an appropriately large constant. Then, with high probability, the signs of the entries of the **last eigenvector** of $\mathbf{L}$ correctly recover the true communities up $\mathcal{O}(1)$ of misclassified nodes. Moreover, the first nontrivial eigenvector **does not** recover the true communities.*

Proof. We first decompose $\mathbf{L}$ into its signal and error components, $\mathbf{L} = \mathbb{E}[\mathbf{L}] + \mathbf{E}$, where $\mathbb{E}[\mathbf{L}]$ is the expected graph Laplacian and $\mathbf{E}$ is the remaining residual term. Since $\mathbf{L}$ is drawn from a stochastic block model with parameters $n, 2, p, q$, we can exactly solve for the characteristic polynomial, eigenvalues, and eigenvectors of its expectation. We express $\mathbb{E}[\mathbf{L}]$ as,

$$\mathbb{E}[\mathbf{L}] = I - \mathbb{E}[\mathbf{D}]^{1/2}\mathbb{E}[\mathbf{A}]\mathbb{E}[\mathbf{D}]^{1/2} \tag{10}$$

where $\mathbb{E}[\mathbf{A}]$ is the expected Adjacency and $\mathbb{E}[\mathbf{D}]$ is the expected diagonal degree matrix. If we order all nodes according to the community they belong to, the components in Eq. 21 can be written as,

$$\mathbb{E}[\mathbf{A}] = \begin{pmatrix} \mathbf{P_A} & \mathbf{Q_A} \\ \mathbf{Q_A} & \mathbf{P_A} \end{pmatrix}, \quad \mathbb{E}[\mathbf{D}] = \begin{pmatrix} \frac{pn}{2} + \frac{qn}{2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{pn}{2} + \frac{qn}{2} \end{pmatrix} \tag{11}$$

where $\mathbb{E}[\mathbf{A}]$ is written in block form with $\mathbf{P_A}, \mathbf{Q_A} \in \mathbb{R}^{\frac{n}{2} \times \frac{n}{2}}$ where $\mathbf{P_A} = \mathbf{1}_{\frac{n}{2}}\mathbf{1}_{\frac{n}{2}}^{\top} \cdot p$ and $\mathbf{Q_A} = \mathbf{1}_{\frac{n}{2}}\mathbf{1}_{\frac{n}{2}}^{\top} \cdot q$ where $\mathbf{1}_{\frac{n}{2}} \in \mathbb{R}^{\frac{n}{2}}$ is a $\frac{n}{2}$-dimensional vector of all ones and $\mathbb{E}[\mathbf{D}]$ is written as a diagonal matrix with $\mathbb{E}[\mathbf{D}] = I \cdot (\frac{pn}{2} + \frac{qn}{2})$. Now, $\mathbb{E}[\mathbf{L}]$ can be written,

$$\mathbb{E}[\mathbf{L}] = \begin{pmatrix} \mathbf{P_L} & \mathbf{Q_L} \\ \mathbf{Q_L} & \mathbf{P_L} \end{pmatrix}, \tag{12}$$

where again we write $\mathbb{E}[\mathbf{L}]$ in block form where $\mathbf{P_L}, \mathbf{Q_L} \in \mathbb{R}^{\frac{n}{2}\times\frac{n}{2}}$ and are defined,

$$\mathbf{P_L} = \begin{pmatrix} 1-\frac{p}{\frac{n}{2}(p+q)} & -\frac{p}{\frac{n}{2}(p+q)} & \cdots & -\frac{p}{\frac{n}{2}(p+q)} \\ -\frac{p}{\frac{n}{2}(p+q)} & 1-\frac{p}{\frac{n}{2}(p+q)} & \cdots & -\frac{p}{\frac{n}{2}(p+q)} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{p}{\frac{n}{2}(p+q)} & -\frac{p}{\frac{n}{2}(p+q)} & \cdots & 1-\frac{p}{\frac{n}{2}(p+q)} \end{pmatrix}, \quad \mathbf{Q_L} = \begin{pmatrix} -\frac{q}{\frac{n}{2}(p+q)} & \cdots & -\frac{q}{\frac{n}{2}(p+q)} \\ \vdots & \ddots & \vdots \\ -\frac{q}{\frac{n}{2}(p+q)} & \cdots & -\frac{q}{\frac{n}{2}(p+q)} \end{pmatrix}. \tag{13}$$

Having derived the exact form of $\mathbb{E}[\mathbf{L}]$, we can derive its characteristic polynomial $p_{\mathbb{E}[\mathbf{L}]}(\lambda)$,

$$p_{\mathbb{E}[\mathbf{L}]}(\lambda) = \frac{\lambda(\lambda-1)^{n-2}(p\lambda + q(\lambda-2))}{p+q}. \tag{14}$$

Equation 14 tells us that the eigenvalues of $\mathbb{E}[\mathbf{L}]$ are $\lambda = 0, 1, \frac{2q}{p+q}$ with multiplicities $1, n-2, 1$. By assumption, $q \gg p$, implying $\frac{2q}{p+q} > 1$, where the last eigenvector corresponds to the eigenvalue $\frac{2q}{p+q}$. The eigenvector of $\frac{2q}{p+q}$ can then be written as the vector $\mathbf{v} \in \mathbb{R}^{n\times 1}$,

$$\mathbf{v}^\top = \begin{pmatrix} -1 & \cdots & -1 & +1 & \cdots & +1 \end{pmatrix} \tag{15}$$

where the first $\frac{n}{2}$ entries of $\mathbf{v}$ are -1 and the last $\frac{n}{2}$ entries of $\mathbf{v}$ are $+1$. Let $\tilde{\mathbf{v}}$ denote the last eigenvector of $\mathbf{L}$. Combining Theorem A.4, Theorem A.5, and Lemma A.3 we have with probability $1-\delta$,

$$\begin{aligned} \inf_{q\in\{-1,+1\}} ||\mathbf{v} - q\tilde{\mathbf{v}}||_2 &\leq \frac{2\,||\mathbb{E}[\mathbf{L}] - \mathbf{L}||}{\min\{\lambda_{r-1}-\lambda_r, \lambda_s - \lambda_{s-1}\}} & (16) \\ &\leq \frac{28\sqrt{\frac{\ln(4n/\delta)}{d_{\min}}}}{\frac{2q}{p+q}-1} & (17) \\ &= \frac{28\sqrt{\frac{\ln(4n/\delta)}{d_{\min}}}(p+q)}{q-p} & (18) \\ &= \mathcal{O}(1) & (19) \end{aligned}$$

Thus, with high probability the last eigenvector of the observed Laplacian $\mathbf{L}$ recovers the communities up to the sign of its entries with $\mathcal{O}(1)$ misclassified nodes. In order to prove our second claim, we note eigenvalue $\lambda = 1$ of $\mathbb{E}[\mathbf{L}]$ has a multiplicty of $n-2$ and its eigenvectors have the form,

$$\tilde{\mathbf{v}}^\top \in \{\begin{pmatrix} -1 & +1 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}, \cdots, \begin{pmatrix} 0 & \cdots & 0 & -1 & 0 & \cdots & 0 & +1 \end{pmatrix}\}, \tag{20}$$

where $\mathbf{v}$ is an all zeros vector other than one index set to -1 and another other set to 1. Assuming we have ordered the nodes according to their community, the index with entry -1 is the index of the first node in either the first or second community while the index with entry 1 is any other index of a node in that community. Since $\tilde{\mathbf{v}}$ is a vector of all zeros other than two other entries, any $k-1$ of them cannot recover the true communities for all n nodes. □

Theorem A.7. *Let* $\mathbf{A}$ *and* $\mathbf{L}$ *be the Adjacency and Laplacian matrix drawn from the stochastic block model* $G(n,k,p,q)$*. Assume* $q \gg p$ *and* $d_{min} \geq Cln(n)$ *where* C *is an appropriately large constant. Then, with high probability, the nonzero entries along the rows of the* ***last*** $k-1$ ***eigenvectors*** *of* $\mathbf{L}$ *correctly recover the true communities up to an orthogonal transformation with at most* $\mathcal{O}(k^{\frac{3}{2}})$ *misclassified nodes. Moreover, the first nontrivial* k *eigenvectors* ***do not*** *recover the true communities.*

Proof. We first decompose $\mathbf{L}$ into its signal and error components, $\mathbf{L} = \mathbb{E}[\mathbf{L}] + E$, where $\mathbb{E}[\mathbf{L}]$ is the expected graph Laplacian and E is the remaining residual term. Since $\mathbf{L}$ is drawn from a stochastic block model with parameters n, k, p, q, we can exactly solve for the characteristic polynomial, eigenvalues, and eigenvectors of its expectation. We express $\mathbb{E}[\mathbf{L}]$ as,

$$\mathbb{E}[\mathbf{L}] = I - \mathbb{E}[\mathbf{D}]^{1/2}\mathbb{E}[\mathbf{A}]\mathbb{E}[\mathbf{D}]^{1/2} \tag{21}$$

where $\mathbb{E}[\mathbf{A}]$ is the expected Adjacency and $\mathbb{E}[\mathbf{D}]$ is the expected diagonal degree matrix. If we order all nodes according to the community they belong to, the components in Eq. 21 can be written as,

$$\mathbb{E}[\mathbf{A}] = \begin{pmatrix} \mathbf{P_A} & \mathbf{Q_A} & \cdots & \mathbf{Q_A} \\ \mathbf{Q_A} & \mathbf{P_A} & \cdots & \mathbf{Q_A} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Q_A} & \mathbf{Q_A} & \cdots & \mathbf{P_A} \end{pmatrix}, \quad \mathbb{E}[\mathbf{D}] = \begin{pmatrix} \frac{pn}{k} + q(n - \frac{n}{k}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{pn}{k} + q(n - \frac{n}{k}) \end{pmatrix} \tag{22}$$

where $\mathbb{E}[\mathbf{A}]$ is written in block form with $\mathbf{P_A}, \mathbf{Q_A} \in \mathbb{R}^{\frac{n}{k} \times \frac{n}{k}}$ where $\mathbf{P_A} = \mathbf{1}_k\mathbf{1}_k^\top \cdot p$ and $\mathbf{Q_A} = \mathbf{1}_k\mathbf{1}_k^\top \cdot q$ where $\mathbf{1}_k \in \mathbb{R}^k$ is a k-dimensional vector of all ones and $\mathbb{E}[\mathbf{D}]$ is written as a diagonal matrix with $\mathbb{E}[\mathbf{D}] = I \cdot (\frac{pn}{k} + q(n - \frac{n}{k}))$. Now, $\mathbb{E}[\mathbf{L}]$ can be written,

$$\mathbb{E}[\mathbf{L}] = \begin{pmatrix} \mathbf{P_L} & \mathbf{Q_L} & \cdots & \mathbf{Q_L} \\ \mathbf{Q_L} & \mathbf{P_L} & \cdots & \mathbf{Q_L} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Q_L} & \mathbf{Q_L} & \cdots & \mathbf{P_L} \end{pmatrix}, \tag{23}$$

where again we write $\mathbb{E}[\mathbf{L}]$ in block form where $\mathbf{P_L}, \mathbf{Q_L} \in \mathbb{R}^{\frac{n}{k} \times \frac{n}{k}}$ and are defined,

$$\mathbf{P_L} = \begin{pmatrix} 1 - \frac{p}{\frac{np}{k} + q(n - \frac{n}{k})} & -\frac{p}{\frac{np}{k} + q(n - \frac{n}{k})} & \cdots & -\frac{p}{\frac{np}{k} + q(n - \frac{n}{k})} \\ -\frac{p}{\frac{np}{k} + q(n - \frac{n}{k})} & 1 - \frac{p}{\frac{np}{k} + q(n - \frac{n}{k})} & \cdots & -\frac{p}{\frac{np}{k} + q(n - \frac{n}{k})} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{p}{\frac{np}{k} + q(n - \frac{n}{k})} & -\frac{p}{\frac{np}{k} + q(n - \frac{n}{k})} & \cdots & 1 - \frac{p}{\frac{np}{k} + q(n - \frac{n}{k})} \end{pmatrix}, \text{ and} \tag{24}$$

$$\mathbf{Q_L} = \begin{pmatrix} -\frac{q}{\frac{np}{k} + q(n - \frac{n}{k})} & \cdots & -\frac{q}{\frac{np}{k} + q(n - \frac{n}{k})} \\ \vdots & \ddots & \vdots \\ -\frac{q}{\frac{np}{k} + q(n - \frac{n}{k})} & \cdots & -\frac{q}{\frac{np}{k} + q(n - \frac{n}{k})} \end{pmatrix}. \tag{25}$$

Having derived the exact form of $\mathbb{E}[\mathbf{L}]$, we can derive its characteristic polynomial $p_{\mathbb{E}[\mathbf{L}]}(\lambda)$,

$$p_{\mathbb{E}[\mathbf{L}]}(\lambda) = \frac{\lambda(\lambda - 1)^{n-k}(p\lambda + q((k-1)\lambda - k))^{k-1}}{(p + (k-1)q)^{k-1}}. \tag{26}$$

Eq. 26 tells us that the eigenvalues of $\mathbb{E}[\mathbf{L}]$ are $\lambda = 0, 1, \frac{kq}{p+(k-1)q}$ with multiplicities $1, n-k, k-1$. By assumption, $q > p$, implying $\frac{kq}{p+(k-1)q} > 1$, where the last $k-1$ eigenvectors correspond to the eigenvalue $\frac{kq}{p+(k-1)q}$. The eigenvectors of $\frac{kq}{p+(k-1)q}$ can then be written as the columns of the matrix $\mathbf{V} \in \mathbb{R}^{n \times k-1}$,

$$\mathbf{V} = \begin{pmatrix} -\mathbf{1}_k & -\mathbf{1}_k & \cdots & -\mathbf{1}_k \\ \mathbf{1}_k & \mathbf{0}_k & \cdots & \mathbf{0}_k \\ \mathbf{0}_k & \mathbf{1}_k & \cdots & \mathbf{0}_k \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{0}_k & \mathbf{0}_k & \cdots & \mathbf{1}_k \end{pmatrix} \tag{27}$$

where $\mathbf{0}_k, \mathbf{1}_k \in \mathbb{R}^k$ are k-dimensional vector of all zeros and ones, respectively. Let $\tilde{\mathbf{V}}$ denote the last $k-1$

eigenvectors of $\mathbf{L}$. Combining Theorem A.4, Theorem A.5, and Lemma A.3 we have with probability $1-\delta$,

$$\begin{aligned}
\inf_{\mathbf{Q}\in O(l)}||\mathbf{V}-\tilde{\mathbf{V}}\mathbf{Q}||_F &\leq \frac{2k^{1/2}\,||\mathbb{E}[\mathbf{L}]-\mathbf{L}||}{\min\{\lambda_{r-1}-\lambda_r, \lambda_s-\lambda_{s-1}\}} && (28)\\
&\leq \frac{28k^{1/2}\sqrt{\frac{\ln(4n/\delta)}{d_{\min}}}}{\frac{kq}{p+(k-1)q}-1} && (29)\\
&= \frac{28k^{1/2}\sqrt{\frac{\ln(4n/\delta)}{d_{\min}}}(p+q(k-1))}{q-p} && (30)\\
&= \mathcal{O}(k^{\frac{3}{2}}) && (31)
\end{aligned}$$

Thus, with high probability the last $k-1$ eigenvectors of the observed Laplacian $\mathbf{L}$ recover the communities up to the nonzero entries of its rows with at most $\mathcal{O}(k^{\frac{3}{2}})$ misclassified nodes. In order to prove our second claim, we note eigenvalue $\lambda=1$ of $\mathbb{E}[\mathbf{L}]$ has a multiplicty of $n-k$ and its eigenvectors have the form,

$$\tilde{\mathbf{v}}^{\top} \in \{\begin{pmatrix}-1 & +1 & 0 & \cdots & 0 & 0 & \cdots & 0\end{pmatrix}, \ \cdots, \begin{pmatrix}0 & \cdots & 0 & -1 & 0 & \cdots & 0 & +1\end{pmatrix}\}, \tag{32}$$

where $\mathbf{v}$ is an all zeros vector other than one index set to -1 and another other set to 1. Assuming we have ordered the nodes according to their community, the index with entry -1 is the index of the first node in some community k while the index with entry 1 is any other index of a node in community k. Since $\mathbf{v}$ is a vector of all zeros other than two other entries, any $k-1$ of them cannot recover the true communities for all n nodes. □

Our results on the heterophilous SBMs can be extended to homophilous SBMs since the derivation for the expected Laplacian, characteristic polynomial, eigenvalues, and eigenvectors remain the same. In fact, the only difference is with the assumption $p \gg q$, and here, the eigenvector(s) of interest lie at the start of the spectrum rather than at the end. We demonstrate this phenomenon below.

Proposition A.8 (Le et al. (2015, 2017))**.** *Let* $\mathbf{A}$ *and* $\mathbf{L}$ *be the Adjacency and Laplacian matrix drawn from the stochastic block model* $G(n,2,p,q)$*. Assume* $p \gg q$ *and* $min(d_i) \geq Cln(n)$ *where* C *is an appropriately large constant. Then, with high probability, the signs of the entries of the* ***first nontrivial eigenvector*** *of* $\mathbf{L}$ *correctly recovers the true communities up* $\mathcal{O}(1)$ *of misclassified nodes.*

Proof. Since in the derivation for $\mathbb{E}[\mathbf{L}]$ and $p_{\mathbb{E}[\mathbf{L}]}(\lambda)$ in Proposition A.6 we did not rely on the assumption that $q \gg p$ and we are assuming a block model defined as $G(n,2,p,q)$, $\mathbb{E}[\mathbf{L}]$ and $p_{\mathbb{E}[\mathbf{L}]}(\lambda)$ are the same as in Proposition A.6. Again, the eigenvalues of $\mathbb{E}[\mathbf{L}]$ are $\lambda = 0, 1, \frac{2q}{p+q}$. Now, by assumption $p \gg q$, implying $\frac{2q}{p+q} < 1$, and as a result the *first nontrivial eigenvector* corresponds to eigenvalue $\frac{2q}{p+q}$. The proof follows similarly as in Proposition A.6 except here we apply Theorems A.4, Theorem A.5, and Lemma A.3 to the first nontrivial eigenvector rather than the last eigenvector, and we arrive at our conclusion. □

Theorem A.9. *Let* $\mathbf{A}$ *and* $\mathbf{L}$ *be the Adjacency and Laplacian matrix drawn from the stochastic block model* $G(n,k,p,q)$*. Assume* $p \gg q$ *and* $min(d_i) \geq Cln(n)$ *where* C *is an appropriately large constant. Then, with high probability, the nonzero entries along the rows of the* ***first nontrivial*** $k-1$ ***eigenvectors*** *of* $\mathbf{L}$ *correctly recovers the true communities up to an orthogonal transformation with at most* $\mathcal{O}(k^{\frac{3}{2}})$ *misclassified nodes.*

Proof. Since in the derivation for $\mathbb{E}[\mathbf{L}]$ and $p_{\mathbb{E}[\mathbf{L}]}(\lambda)$ in Theorem 3.2 we did not rely on the assumption that $q \gg p$ and we are assuming a block model defined as $G(n,k,p,q)$, $\mathbb{E}[\mathbf{L}]$ and $p_{\mathbb{E}[\mathbf{L}]}(\lambda)$ are the same as in Theorem 3.2. Again, the eigenvalues of $\mathbb{E}[\mathbf{L}]$ are $\lambda = 0, 1, \frac{kq}{p+(k-1)q}$. Now, by assumption $p \gg q$, implying $\frac{kq}{p+(k-1)q} < 1$, and as a result the *first nontrivial* $k-1$ *eigenvectors* correspond to eigenvalue $\frac{kq}{p+(k-1)q}$. The proof follows similarly as in Theorem 3.2 except here we apply Theorems A.4, Theorem A.5, and Lemma A.3 to the first nontrivial $k-1$ eigenvectors rather than the last $k-1$ eigenvectors, and we arrive at our conclusion. □

B PROOFS FOR LEARNABLE LAPLACIAN POSITIONAL ENCODINGS

B.1 Background, Definitions, and Lemmas on Chebyshev Polynomials

Here we provide the background on approximation theory, Chebyshev Polynomials, and the approximation power of Chebyshev series necessary to prove our theorems for LLPE. We first introduce relevant notions from approximation theory, beginning with the l^∞ norm, allowing us to measure the closeness of an arbitrary f to g. We next review the definition of the Chebyshev polynomial, Chebyshev series, and the approximation power of truncated Chebyshev series. Further discussions on Chebyshev polynomials can be found in Rivlin (2020); Mason and Handscomb (2002).

Definition B.1. The l^∞ norm of function f on interval $[a, b]$ is defined,

$$||f||_\infty = \max_{a \leq x \leq b} |f(x)| \tag{33}$$

If for some prescribed ϵ, $||f - g|| \leq \epsilon$, we say that g uniformly approximates f. Moreover, if for some g^*, $||f - g^*|| \leq ||f - g||$ for all g, we say that g^* is a best approximation to f.

The Chebyshev polynomial $T_m(x)$ of the first kind is a polynomial of degree m defined as,

$$T_m(x) = \cos(m \cdot \arccos(x)) \tag{34}$$

when $0 \leq \arccos(x) \leq \pi$. One crucial property of the Chebyshev polynomial of degree m is that $\tilde{T}_m$ the normalized Chebyshev polynomial such that its leading coefficient is 1 has the minimum l^∞ norm among all polynomials of degree m. The following theorem states this formally,

Theorem B.2 (Rivlin (2020))**.** *Let $p_m \in \mathscr{P}_m$ be a polynomial of degree m. Then,*

$$||p_m||_\infty \geq \left|\left|\tilde{T}_m\right|\right|_\infty = \begin{cases} 2^{1-m}, & m > 0 \\ 1, & m = 0 \end{cases} \tag{35}$$

We later use Theorem B.2 to prove optimal statistical generalization for LLPE among all other choices of approximating polynomials. Now, for any function f, there exists its Chebyshev series denoted as,

$$f(x) \sim \sum_{m=0}^{\infty} a_m T_m(x), \quad \text{where } a_k = \frac{2}{\pi} \int_{-1}^{1} f(x) T_k(x) \frac{\partial x}{\sqrt{1 - x^2}} \tag{36}$$

If f is continuous, its Chebyshev series is pointwise convergent to f. Moreover, we can obtain uniform convergence if we place stricter assumptions on f. In particular, if f satisfies the Dini-Lipschitz condition such that

$$\lim_{m \to \infty} \log(m) \omega\left(\frac{1}{m}\right) = 0, \tag{37}$$

where $\omega(\delta) = \sup_{x_1 - x_2 \leq \delta} |f(x_1) - f(x_2)|$, then its Chebyshev series is uniformly convergent. The above discussion tell us that the Chebyshev series approximates functions that satisfy certain continuity and Lipschitz constraints. However, in practice, we are limited to partial sums of the Chebyshev series, and thus we will be mainly concerned with the approximating power of the partial sums of degree M. We denote the Mth partial sum of the Chebyshev series of f as,

$$s_M(f; x) = s_M(x) = \sum_{m=0}^{M} a_m T_m(x). \tag{38}$$

First we will discuss the relationship between the error of the best approximating polynomial of degree M to f and the error of the Mth partial sum of the Chebyshev series to f.

Theorem B.3 (Rivlin (2020))**.** *Let f be continuous and define,*

$$S_M(f) = ||f - s_M(f)|| , \quad E_M(f) = ||f - p_M^*|| , \tag{39}$$

where p_M^* is the best uniform approximating polynomial of degree M to f. Then,

$$E_M(f) \leq S_M(f) < \left(4 + \frac{4}{\pi^2} log(n)\right) E_M(f) \tag{40}$$

Theorem B.3 tells us that the error of $s_M(f)$ doesn't deviate by *too much* from the error of the best approximating polynomial of degree M, p_M^*. More specifically, the error grows on the order of $\mathcal{O}(\log(M))$, and as result, the Mth partial sum of the Chebyshev series is a *near-best* approximation of the function f. The following theorem will be our main tool used in proving the approximation theorems for LLPE and states that partial sums of the Chebyshev series converge exponentially if f is continuously differentiable,

Theorem B.4 (Mason and Handscomb (2002))**.** *If f has $d+1$ continuous derivatives on $[-1, +1]$, then $||f(x) - s_M(f)|| = O(n^{-d})$.*

B.2 Theoretical Expressivity of LLPE

Definition B.5. Let G be an arbitrary graph. Define $f_r : V \times V \to \mathbb{R}$ as a function on G with respect to $r : [0, 2] \to \mathbb{R}^+$ such that $f_r(i, j)$ has the following form,

$$f_r(i,j)^2 = \sum_{k=1}^{n} r(\lambda_k)(\mathbf{u}_k[i] - \mathbf{u}_k[j])^2. \tag{41}$$

Theorem B.6. *Let G be an arbitrary graph and d_r be a function on G of the form in Definition 4.2 for some r. LLPE can recover d_r such that for any nodes i and j, the l^2 distance between LLPE's encoding for nodes i and j approximates $d_r(i, j)$.*

Proof. Let $\epsilon > 0$. Formally, we aim to show,

$$\left| ||\mathbf{P}_{\text{LLPE}}[i, :] - \mathbf{P}_{\text{LLPE}}[j, :]||_2^2 - d_r(i,j)^2 \right| \leq \epsilon \tag{42}$$

To begin, let $d = n$ such that $\mathbf{P}_{\text{LLPE}} = \mathbf{U}\mathbf{W}_{\text{LLPE}} \in \mathbb{R}^{n \times n}$. Here, column j of $\mathbf{W}_{\text{LLPE}}$ corresponds to a Chebyshev series parameterized by $\boldsymbol{\theta}_j \in \mathbb{R}^M$. Now, the key in proving the approximation is choose M large enough such that $h(\lambda; \boldsymbol{\theta}_j)$ can approximate functions of the form,

$$f_j(\lambda) = r(\lambda_j)e^{-(\lambda - \lambda_j)^2 C_{\max}}, \tag{43}$$

where $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ and $C_{\max}$ is a large constant. Let us discuss function f_j. Essentially, f_j is a function that activates at λ_j with value $r(\lambda_j)$ and is 0 for all other eigenvalues. Indeed if we choose $C_{\max}$ to be large enough, we find that only values within a range $[\lambda_j - \delta, \lambda + \delta]$ for some small $\delta > 0$ have nonzero output, while values outside this range are zero. Notice further that f_j has infinitely many derivatives, and thus if we choose a large enough M, by Theorem B.4 we have $||f(\lambda) - h(\lambda; \boldsymbol{\theta}_j)|| < \epsilon$ for some prescribed ϵ. In our case, we choose a large enough M such that $||f(\lambda) - h(\lambda; \boldsymbol{\theta}_j)|| < \epsilon / n \cdot \max_k(\mathbf{u}_k[i] - \mathbf{u}_k[j])^2$. Having established that $h(\lambda; \boldsymbol{\theta}_j)$ can approximate $f_j(\lambda)$, $\mathbf{P}_{\text{LLPE}} = \mathbf{U}\mathbf{W}_{\text{LLPE}} \approx \mathbf{U}r(\boldsymbol{\Lambda})$. This gives us,

$$\left| ||\mathbf{P}_{\text{LLPE}}[i, :] - \mathbf{P}_{\text{LLPE}}[j, :]||_2^2 - d_r(i,j)^2 \right| \tag{44}$$

$$\leq \sum_{k=1}^{n} (r(\lambda_k) + \frac{\epsilon}{n \cdot \max_k(\mathbf{u}_k[i] - \mathbf{u}_k[j])^2})(\mathbf{u}_k[i] - \mathbf{u}_k[j])^2 - \sum_{k=1}^{n} (r(\lambda_k)(\mathbf{u}_k[i] - \mathbf{u}_k[j])^2 \tag{45}$$

$$\leq \sum_{k=1}^{n} (\frac{\epsilon}{n \cdot \max_k(\mathbf{u}_k[i] - \mathbf{u}_k[j])^2})(\mathbf{u}_k[i] - \mathbf{u}_k[j])^2 \leq \epsilon \tag{46}$$

□

Proposition B.7. *Let* $\mathbf{A}$ *and* $\mathbf{L}$ *be the Adjacency and Laplacian matrix drawn from the stochastic block model* $G(n, k, p, q)$*. If* $p \gg q$ *or* $q \gg p$ *and* $min(d_i) \geq Cln(n)$ *where* C *is an appropriately large constant, then LLPE can correctly recover the true communities up to an orthogonal transformation with at most* $\mathcal{O}(k^{\frac{3}{2}})$ *misclassified nodes.*

Proof. First assume $p \gg q$ and let $0 < \epsilon < \mathcal{O}(k^{\frac{3}{2}})$. Formally, we aim to show,

$$||\mathbf{P}_{\text{LLPE}} - \mathbf{U}_k||_F \leq \epsilon. \tag{47}$$

To begin, let $d = k$. The key in proving the approximation is to choose M large enough such that $h(\lambda; \boldsymbol{\theta}_j)$ can approximate functions of the form,

$$f_j(\lambda) = e^{-(\lambda - \lambda_j)^2 C_{\max}} \tag{48}$$

Here, f_j is a function that activates at λ_j with value $+1$ and is zero for all other eigenvalues. Since again f_j has infinitely many derivatives, we can choose a large enough M such that by Theorem B.4 we have $||f(\lambda) - h(\lambda; \boldsymbol{\theta}_j)|| < \epsilon$ for some prescribed ϵ/k. Having established that $h(\lambda; \boldsymbol{\theta}_j)$ can approximate $f_j(\lambda)$, $\mathbf{P}_{\text{LLPE}} \approx \mathbf{U}_k$, thus proving the result for $p \gg q$. We can similarly prove the result assuming $q \gg p$, by defining f_j as follows,

$$f_j(\lambda) = e^{-(\lambda - \lambda_{n-j})^2 C_{\max}} \tag{49}$$

where instead f_j is a function that activates at λ_{n-j} with value $+1$ and is zero elsewhere. This allows us to effectively obtain, $\mathbf{P}_{\text{LLPE}} \approx \mathbf{U}_{-k:}$, where $\mathbf{U}_{-k:}$ is the last k eigenvectors of the graph Laplacian, thus proving the result for $p \gg q$.

□

B.3 Generalization Properties of LLPE

We provide the relevant background on Rademacher complexity found in Awasthi et al. (2020). Let $\mathcal{H}$ be a hypothesis class where $h : \mathbb{R}^d \to \mathbb{R}$ when $h \in \mathcal{H}$. The empirical Rademacher complexity of $\mathcal{H}$ given sample $S = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is defined,

$$\hat{\mathfrak{R}}(\mathcal{H}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\text{sup}_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{n} \sigma_i h(\mathbf{x}_i)\right] \tag{50}$$

where $\boldsymbol{\sigma}$ is a vector of Rademacher variables drawn uniformly from values in $\{-1, +1\}$. For a family of functions $\mathcal{F}$ where $f : \mathbb{R}^d \to [0, 1]$, we have the following result: for any $\delta > 0$, with probability $1 - \delta$ over the randomness of $S \sim D$, the following inequality holds for all $f \in \mathcal{F}$:

$$\mathbb{E}_{x \sim D}[f(x)] - \mathbb{E}_{x \sim S}[f(x)] \leq 2\hat{\mathfrak{R}}(\mathcal{F}) + \sqrt{\frac{\log(\frac{1}{\delta})}{2m}} \tag{51}$$

Awasthi et al. (2020) considers the hypothesis set of linear predictors with weight bounded in l^p norm with $\mathcal{H}_p = \{\mathbf{x} \to \mathbf{x} \cdot \mathbf{w} : ||\mathbf{w}||_p \leq W\}$ and proves the following upper and lower bounds on the empirical Rademacher complexity of $\mathcal{H}_2$.

Theorem B.8 (Awasthi et al. (2020))**.** *Let* $\mathcal{H}_2 = \{\mathbf{x} \to \mathbf{x} \cdot \mathbf{w} : ||\mathbf{w}||_2 \leq W\}$ *be a hypothesis class of linear predictors defined over* $\mathbb{R}^d$ *with bounded weight in* l^2 *norm. Then, the empirical Rademacher complexity of* $\mathcal{H}_p$ *for sample* $S = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ *admits the following upper and lower bounds,*

$$\frac{W}{\sqrt{2n}} ||\mathbf{X}||_F \leq \hat{\mathfrak{R}}_S(\mathcal{H}_2) \leq \frac{W}{n} ||\mathbf{X}||_F \tag{52}$$

where $\mathbf{X}$ *is the* $n \times d$ *feature matrix and* $||\cdot||_F$ *is the Frobenius norm.*

Theorem B.9. *Let* $\mathcal{H}_{LLPE} = \{\tilde{\lambda} \to \sum_{m=1}^{M} \theta_m \cdot \tilde{T}_m(\tilde{\lambda}) : \boldsymbol{\theta} \in \mathbb{R}^M, ||\boldsymbol{\theta}||_2 \leq C_{LLPE}\}$*, where* C_{LLPE} *is some constant greater than 0,* $\tilde{\lambda}$ *is the rescaled eigenvalues that lie in* $[-1, +1]$*, and* $\tilde{T}_m$ *is the normalized Chebyshev polynomial.*

Then, the empirical Rademacher complexity of the hypothesis class $\mathcal{H}_{LLPE}$ for a sample $S = (\lambda_0, \dots, \lambda_n)$ admits the following upper and lower bounds,

$$\frac{C_{LLPE}}{\sqrt{2n}} \leq \hat{\mathfrak{R}}_S(\mathcal{H}_{LLPE}) \leq \frac{\sqrt{2}C_{LLPE}}{\sqrt{n}} \tag{53}$$

Proof. The key in proving Theorem 4.4 is to recognize that $\mathcal{H}_{\text{LLPE}}$ is equivalent to a linear hypothesis class with weight $\boldsymbol{\theta}$ and feature matrix $\tilde{\mathbf{T}} \in \mathbb{R}^{n \times M}$, where $\tilde{\mathbf{T}}$ is a matrix of normalized Chebyshev polynomial outputs. Once we characterize the Frobenius norm of $\tilde{\mathbf{T}}$, we can apply Theorem B.8 to obtain our upper and lower bounds. To begin, we can represent the Frobenius norm of $\tilde{\mathbf{T}}$ as follows,

$$\left|\left|\tilde{\mathbf{T}}\right|\right|_F = \left|\left|\begin{pmatrix} \tilde{T}_0(\lambda_0) & \cdots & \tilde{T}_M(\lambda_0) \\ \vdots & \ddots & \vdots \\ \tilde{T}_0(\lambda_n) & \cdots & \tilde{T}_M(\lambda_n) \end{pmatrix}\right|\right|_F \tag{54}$$

Notice that the jth column of $\tilde{\mathbf{T}}$ corresponds to the vector of the Chebyshev outputs on the eigenvalues where the Chebyshev polynomial is of order j. We can then upper bound the Frobenius norm of $\tilde{\mathbf{T}}$ by replacing each column with the vector of the l^∞ norms of the Chebyshev polynomials as follows,

$$\left|\left|\begin{pmatrix} \tilde{T}_0(\lambda_0) & \cdots & \tilde{T}_M(\lambda_0) \\ \vdots & \ddots & \vdots \\ \tilde{T}_0(\lambda_n) & \cdots & \tilde{T}_M(\lambda_n) \end{pmatrix}\right|\right|_F \leq \left|\left|\begin{pmatrix} \left|\left|\tilde{T}_0\right|\right|_\infty & \cdots & \left|\left|\tilde{T}_M\right|\right|_\infty \\ \vdots & \ddots & \vdots \\ \left|\left|\tilde{T}_0\right|\right|_\infty & \cdots & \left|\left|\tilde{T}_M\right|\right|_\infty \end{pmatrix}\right|\right|_F \tag{55}$$

We can now apply Theorem B.2 to the upper bound on the Frobenius norm of $\tilde{\mathbf{T}}$, where we obtain,

$$\left|\left|\begin{pmatrix} \left|\left|\tilde{T}_0\right|\right|_\infty & \cdots & \left|\left|\tilde{T}_M\right|\right|_\infty \\ \vdots & \ddots & \vdots \\ \left|\left|\tilde{T}_0\right|\right|_\infty & \cdots & \left|\left|\tilde{T}_M\right|\right|_\infty \end{pmatrix}\right|\right|_F = \left|\left|\begin{pmatrix} 1 & \cdots & 2^{1-m} \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 2^{1-m} \end{pmatrix}\right|\right|_F \tag{56}$$

Now, we can proceed to derive an upper bound on the Frobenius norm of $\tilde{\mathbf{T}}$ as follows,

$$\left|\left|\tilde{\mathbf{T}}\right|\right|_F \leq \sqrt{\sum_{j=1}^{M}\sum_{i=1}^{n}\left|\tilde{\mathbf{T}}[i,j]\right|^2} \tag{57}$$

$$\leq \sqrt{\sum_{j=1}^{M} n(2^{1-j})^2} \tag{58}$$

$$\leq \sqrt{n}\sqrt{\sum_{j=0}^{\infty} 2^{-j}} \tag{59}$$

$$= \sqrt{2n} \tag{60}$$

where Equation 59 follows since $(2^{1-j})^2 \leq 2^{1-j}$ for all $j \geq 0$ and Equation 60 follows since $\sqrt{\sum_{j=0}^{\infty} 2^{-j}}$ is a geometric series. Having derived an upper bound on the Frobenius norm of $\tilde{\mathbf{T}}$, applying Theorem B.8 to $\mathcal{H}_{\text{LLPE}}$ proves the upper bound of the result. We prove the lower bound of the result by lower bounding the Frobenius norm of $\tilde{\mathbf{T}}$. To do so, we replace each column with the vector of the minimum absolute value of the Chebyshev polynomials as follows,

$$\left|\left|\begin{pmatrix} \tilde{T}_0(\lambda_0) & \cdots & \tilde{T}_M(\lambda_0) \\ \vdots & \ddots & \vdots \\ \tilde{T}_0(\lambda_n) & \cdots & \tilde{T}_M(\lambda_n) \end{pmatrix}\right|\right|_F \geq \left|\left|\begin{pmatrix} \min_\lambda(\left|\tilde{T}_0(\lambda)\right|) & \cdots & \min_\lambda(\left|\tilde{T}_M(\lambda)\right|) \\ \vdots & \ddots & \vdots \\ \min_\lambda(\left|\tilde{T}_0(\lambda)\right|) & \cdots & \min_\lambda(\left|\tilde{T}_M(\lambda)\right|) \end{pmatrix}\right|\right|_F \tag{61}$$

The Chebyshev polynomial $\tilde{T}_0$ is defined as the constant function equal to one, while each of the Chebyshev polynomials of order $m > 0$ have exactly $m + 1$ zeros. Thus, the lower bound of the Frobenius norm of $\tilde{\mathbf{T}}$ can be expressed as the following,

$$\left\|\begin{pmatrix} \min_\lambda(\left|\tilde{T}_0(\lambda)\right|) & \cdots & \min_\lambda(\left|\tilde{T}_M(\lambda)\right|) \\ \vdots & \ddots & \vdots \\ \min_\lambda(\left|\tilde{T}_0(\lambda)\right|) & \cdots & \min_\lambda(\left|\tilde{T}_M(\lambda)\right|) \end{pmatrix}\right\|_F = \left\|\begin{pmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix}\right\|_F \tag{62}$$

The lower bound of the Frobenius norm of $\tilde{\mathbf{T}}$ then becomes,

$$\left\|\tilde{\mathbf{T}}\right\|_F \geq \sqrt{n} \tag{63}$$

Lastly, applying Theorem B.8 to $\mathcal{H}_{\text{LLPE}}$ proves the lower bound of the result.

□

C ADDITIONAL THEORY, EMPIRICAL RESULTS AND DISCUSSIONS

C.1 Comparing LLPE and LPE-FLK with Complex Synthetic Graphs

We conducted an additional experiment on synthetic graphs of varying homophily levels using a modified preferential attachment process introduced by Zhu et al. (2020) (Figure 5). Here, nodes are added one by one, and the probability that a new node u of class i forms an edge with existing node v of class j is proportional to both the class compatibility between classes i and j, $H_{i,j}$, and the degree of node v. As a result, the degree distribution for the resulting graphs follows a powerlaw, and the homophily can be controlled by the compatibility matrix H. We train and evaluate a GT equipped with LPE-FLK and LLPE and present results below. The results demonstrate that GT with LPE-FLK obtains the same performance across different homophilies and is unable to capture the relevant graph structure in high or low homophily settings on the more complex powerlaw graph. On the other hand, the performance of GT with LLPE significantly increases when homophily is either very high or very low, indicating it is able to capture the relevant graph structure and identify the relevant eigenvectors in complex graph structures.

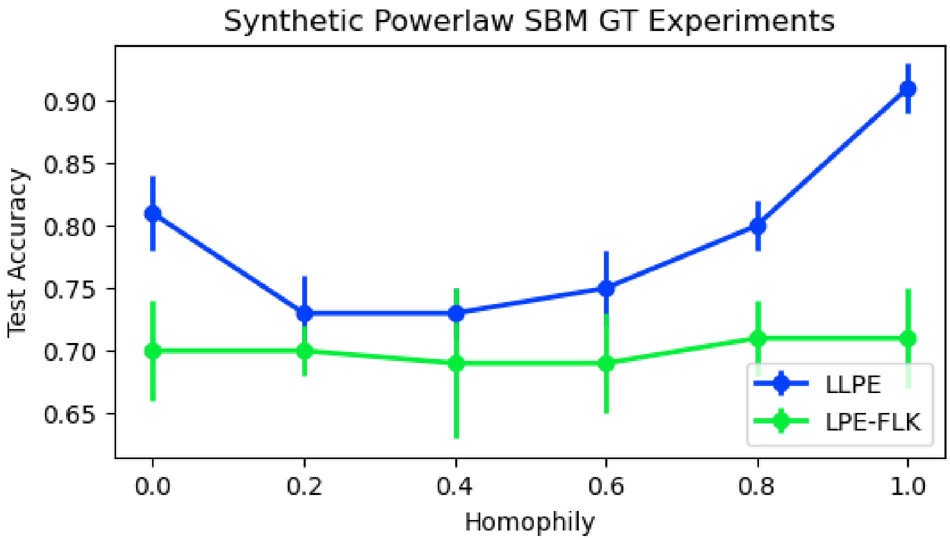
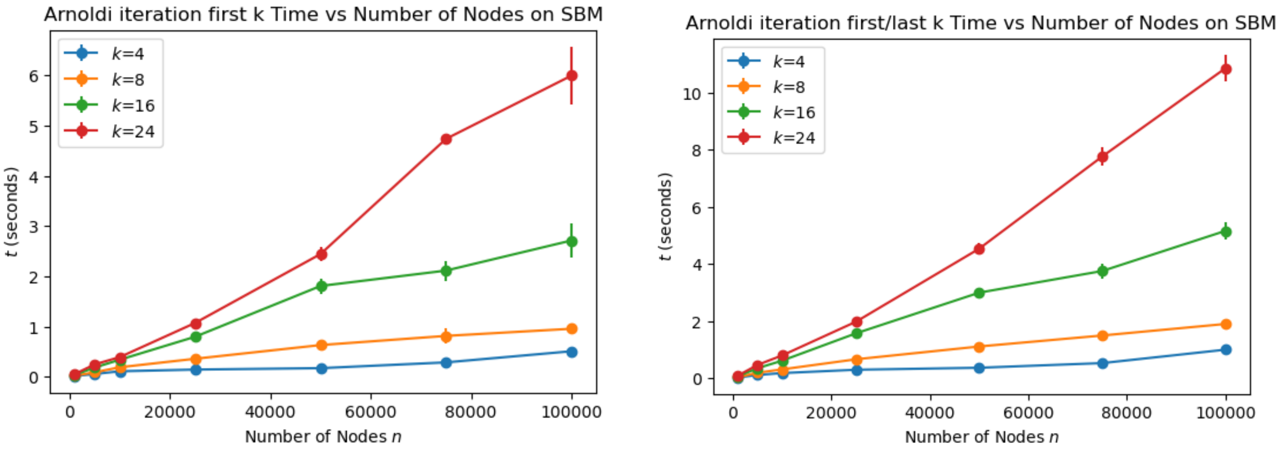
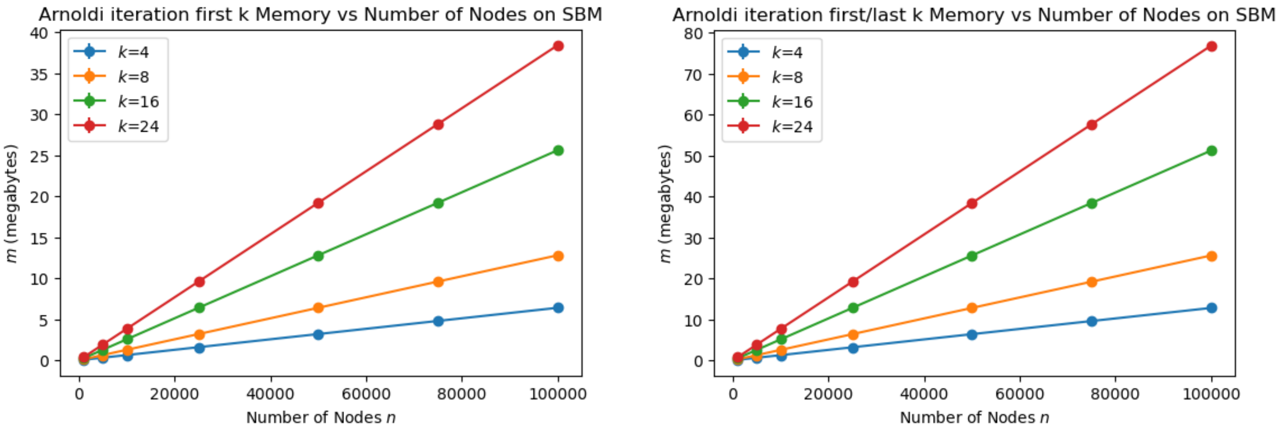


Figure 5: Mean and standard deviations (error bars) of all model-PE combinations on the synthetic SBMs. LLPE performs well across both high homophily and high heterophily, while LPE-FK does not.



(a) Time complexity in obtaining first k eigenvectors (b) Time complexity in obtaining first and last k eigenvectors

Figure 6: Arnoldi-iteration time complexity along SBMs varying n and k.



(a) Space complexity obtaining first k eigenvectors (b) Space complexity obtaining first and last k eigenvectors

Figure 7: Arnoldi-iteration space complexity along SBMs varying n and k.

C.2 Computational Efficiency of LLPE

We demonstrate the computational efficiency of the large version of LLPE by measuring both time and space complexity of Arnoldi iteration in obtaining the first and last k eigenvectors on an SBM of sizes up to 100,000 nodes. We utilize a fast implementation of Arnoldi iteration readily available in SciPy optimized with sparse matrix and vector operations. We conduct our experiment on an Intel ® Xeon ® CPU E5-2620 measuring the runtime and memory usage of the algorithm as we increase n, the size of the graph and k, the number of eigenvectors obtained (Figures 6 and 7). Figure 6 demonstrates that obtaining the first and last k eigenvectors of graphs with 1000s of nodes is very fast, requiring only seconds, while obtaining the first and last k eigenvectors on graphs of size 100,000 nodes requires tens of seconds. Figure 7 demonstrates that the memory requirements during Arnoldi-iteration is also efficient, requiring tens of megabytes on graphs of size 100,000 nodes.

C.3 Sensitivity to Order M of LLPE

In order to test the sensitivity of LLPE to choices of M, we plot the test performance of LLPE across different choices of M on Cora's giant component in Figure 8. Figure 8 tells us that as long as M is large enough, LLPE obtains high test accuracy. Interestingly, the sensitivity analysis agrees with both our theoretical results on the approximation capabilities and statistical generalization for LLPE. In particular, Proposition 4.1 and Theorem 4.3 indicate that M needs to be large enough in order for LLPE to obtain good approximation capabilities. Similarly,

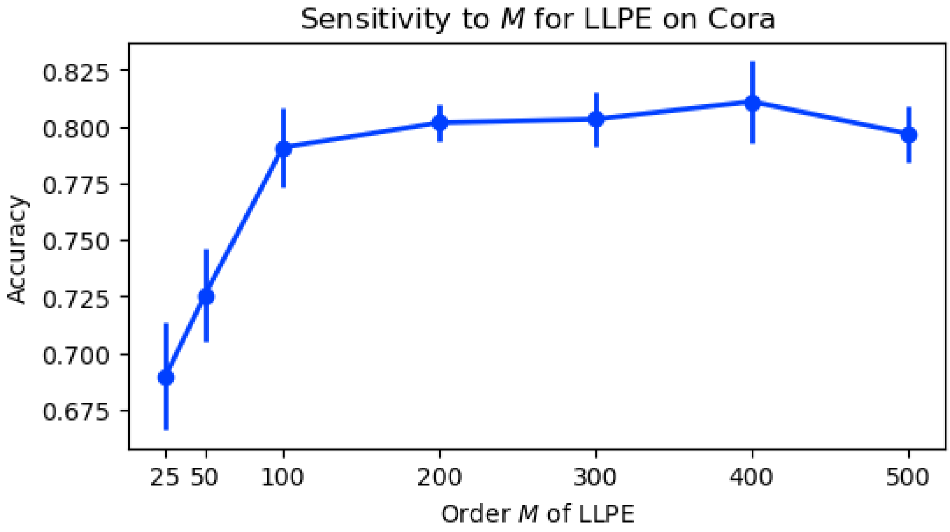


Figure 8: Sensitivity analysis to the order M of LLPE on Cora.

we find empirically that when M is too small as indicated when $M \in [25, 50]$ performance degrades. On the other hand, Theorem 4.4 tells us that LLPE's statistical generalization does not depend explicitly on the order M of LLPE. As a result, we can select a large M to guarantee high expressivity, while not harming generalization. Indeed, we find empirically that for choices of M as large as $M = 500$, LLPE maintains a high test accuracy.

C.4 Sensitivity to Number of Eigenvectors k for Large LLPE

In order to test the sensitivity of the large version of LLPE to choices of k, we plot the performance of LLPE across different choices of k on Cora's giant component in Figure 9. Figure 9 tells us that if k is too small such as $k \in [8, 32]$, then performance degrades. On the other hand, once k exceeds this range, performance is stable across choices of k. Although performance is stable across choice of $k > 32$, we find that the best performing choice of k lies at $k = 64$ rather than $k = 1024$. This result indicates that we can potentially improve LLPE by searching across choices of k from the start and ends of the spectrum. However, this design introduces another hyperparameter, requiring a careful search across the spectrum of the Laplacian. It is also prone to missing eigenvectors lying in the middle of the spectrum, where previous work has shown these eigenvectors can be relevant (Li et al., 2023). We thus leave this idea for exploration in future work.

Interestingly, the sensitivity to the choice of k is different on LLPE in comparison to LPE. In LPE, the best choice of k tends to be small where $k \in [8, 32]$, while for LLPE the best choice of k tends to be large where $k \in [64, 1024]$. In fact for LPE, we observe in our empirical results that no choices of k greater than 32 obtain the best performance among models trained with LPE. We hypothesize that these behaviors are due to the ability of LLPE to learn which eigenvectors are important and inability of LPE to learn which eigenvectors are important. Since LLPE learns which eigenvectors are important, providing them with only a small subset of the eigenvectors reduces performance. On the other hand, LPE cannot learn which eigenvectors are important, and thus providing to them more eigenvectors reduces performance.

C.5 Additional related work on PEs

RWSE (Dwivedi et al., 2022) and SAN-PE (Kreuzer et al., 2021) focus on improving GNN expressivity for graph classification by using learned MLPs or Set Transformers based on the first k eigenvectors. SignNet (Lim et al., 2022) addresses LPE's sign ambiguity, and PEG (Wang et al., 2022) designs rotation and reflection equivariant LPEs.

PEs other than LPE rely on random walks and node distances such as RWSE, the diagonals of the m-step random walk matrix (Dwivedi et al., 2022), pair-wise shortest path node distances (Ying et al., 2021), and shortest paths between nodes and anchor nodes (You et al., 2019). Park et al. (2022) propose applying MLPs to the matrix of

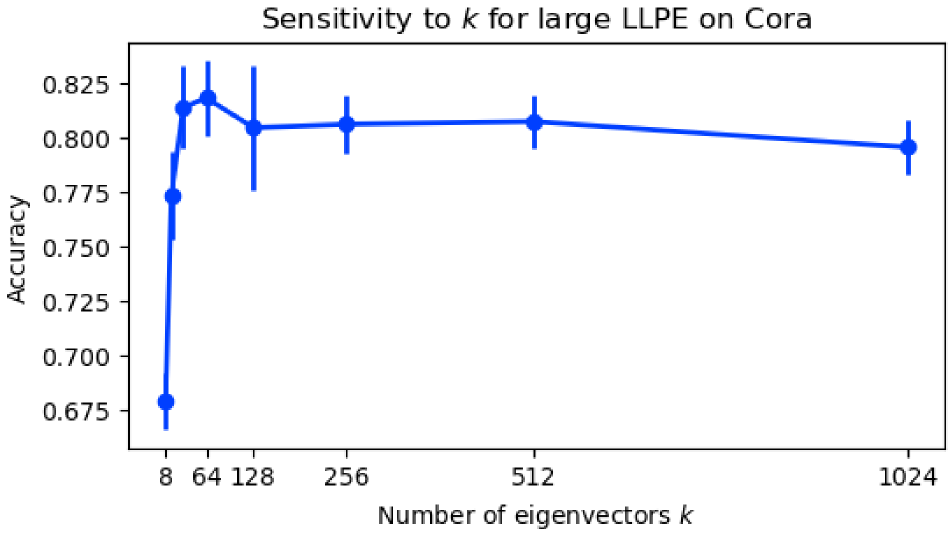


Figure 9: Sensitivity analysis to the number of eigenvectors k of large LLPE on Cora's giant component.

Katz indices, a weighted sum of powers of the adjacency matrix. This design is similar to RWSE (Dwivedi et al., 2022) where they propose to apply MLPs to the random walk matrix. Bo et al. (2023) propose Specformer, a new spectral GNN, that applies sine and cosine PEs to the eigenvalues, feeds them to a transformer and MLP, and finally uses the processed eigenvalues as spectral filters in a spectral GNN. Li et al. (2024) propose a PE that applies message-passing and sine and cosine PEs to node representations. This design is similar to the work of structure-aware transformers (Chen et al., 2022). Lastly, Pan and Kang (2023) propose high and low-pass filters for graph clustering. In contrast to the above works, we design a new position encoding that operates directly on the full matrix of eigenvectors without the use of neighborhood information or message passing.

Arnaiz-Rodríguez et al. (2022) propose a PE that can predict commute times and the Fiedler vector given node features. The PE is then shown to improve performance on small heterophilous benchmarks. Velingker et al. (2023) propose PEs based on random walks such as effective resistance, commute, and hitting times. This work is also similar to RWSE (Dwivedi et al., 2022). Zhang et al. (2023) propose to include graph distances as PEs in order to solve graph biconnectivity. Lastly, Lu et al. (2024) propose a new class of Laplacian matrices that reduce diffusion distance. In contrast to the above PEs, which focus on non-learnable random walk measures, we propose LLPE, a learnable PE that extends Laplacian PEs to capture homophily and heterophily.

D EXPERIMENTAL DETAILS

D.1 Synthetic Experimental Details

In our synthetic experiments, we utilize binary and multiclass SBMs. For each dataset, we set the average degree of each node as $d = 10$ and generate 5 different SBMs according to the homophily ratios $h = [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]$. For example, when the homophily ratio is 0.8, each node has on average 8 neighbors in their own community and 2 neighbors in other communities. Node labels are the node's community. For the binary SBM, we sample 10 independent Gaussian distributed node features with mean $\mathbf{y}[i] \cdot \mu$ and variance σ^2. For the multiclass SBM, we sample a multivariate Gaussian distributed vector with mean $\mu \cdot \text{one-hot}(\mathbf{y}[i])$ and covariance Σ. For all model and PE combinations, we follow the same hyperparameter search as described in the real-world experiments.

D.2 Real-world Experimental Details

We use small, medium, and large datasets from (Yang et al., 2016; Bojchevski and Günnemann, 2018; Shchur et al., 2018; Pei et al., 2020; Platonov et al., 2023; Lim et al., 2021). We test the following base models: MLP, a feedforward neural network, SAGE (Hamilton et al., 2017), a message-passing neural network, GT (full) (Dwivedi and Bresson, 2021), a graph transformer that leverages global attention. For each model, we search across learning rates $\eta \in [0.05, 0.01, 0.005, 0.001]$, optimizer SGD with momentum and Adam, dropouts in [0.0, 0.2, 0.5], size of hidden dimension in [64, 128], and number of layers in [1, 2]. For the GT, we additionally search across layer norms in [0.0, 0.0001]. For all models, we train with full batch mode with early stopping set to 200. We select the best performing model on the validation set for evaluation. In addition to LLPE, we test all models with the following PEs.

1. LPE-FK: The first k nontrivial eigenvectors of the graph Laplacian as defined in Dwivedi and Bresson (2021) with a learnable linear projection matrix.
2. LPE-FLK: The first and last k eigenvectors of the graph Laplacian with a learnable linear projection matrix.
3. LPE-Full: All eigenvectors of the graph Laplacian with a learnable linear projection matrix.
4. Elastic-PE: The matrix of electrostatic potentials as defined in Liu et al. (2023) with a learnable linear projection matrix.
5. SignNet: SignNet with DeepSets as defined in Lim et al. (2022).
6. SAN-PE: SAN-PE as defined in Kreuzer et al. (2021).
7. RWSE: The diagonals of the m-step random walk matrix as defined in (Dwivedi et al., 2022).

For LPE-FK, LPE-FLK, Elastic-PE, SignNet, and SAN-PE, we search across the number of eigenvector and eigenvalue pairs $k \in [8, 16, 32, 64, 128, 256, 512]$. For all PEs, we find that typically the best performing k lies in the range [8, 32]. For RWSE, we search across the range $m \in [8, 16, 32, 64]$. For LLPE, we search across $M \in [64, 128]$, l^1 regularization in [0.001, 0.0001, 0.0], and set $d = 128$. For all model and PE combinations, we linearly project the node features and PE representations into the same space, then concatenate them before passing the entire representation to the base model as described in Dwivedi and Bresson (2021). We implement and train our models on a GeForce GTX 1080.