Statistical Guarantees for Link Prediction using Graph Neural Networks

Alan Chung [1]   Amin Saberi [2]   Morgane Austern [1]

Abstract

This paper derives statistical guarantees for the performance of Graph Neural Networks (GNNs) in link prediction tasks on graphs generated by a graphon. We propose a linear GNN architecture (LG-GNN) that produces consistent estimators for the underlying edge probabilities. We establish a bound on the mean squared error and give guarantees on the ability of LG-GNN to detect high-probability edges. Our guarantees hold for both sparse and dense graphs. Finally, we demonstrate some of the shortcomings of the classical GCN architecture, as well as verify our results on real and synthetic datasets.

1. Introduction

Graph Neural Networks (GNNs) have emerged as a powerful tool for link prediction (Zhang & Chen, 2018; Zhang, 2022). A significant advantage of GNNs lies in their adaptability to different graph types. Traditional link prediction heuristics tend to presuppose network characteristics. For example, the common neighbors heuristic presumes that nodes that share many common neighbors are more likely to be connected, which is not necessarily true in biological networks (Kovács, 2019). In contrast, GNNs inherently learn predictive features through the training process, presenting a more flexible and adaptable method for link prediction.

This paper provides statistical guarantees for link prediction using GNNs, in graphs generated by the graphon model. A graphon is specified by a symmetric measurable kernel function $W : \Omega^2 \to [0, 1]$. A graph $G_n = (V_n, E_n)$ with the vertex set $V_n = \{1, 2, \cdots, n\}$ is sampled from W as follows: (i) each vertex $i \in V_n$ draws latent feature $(\omega_i) \overset{i.i.d}{\sim} \mu$ for some probability distribution μ on $\Omega \subset \mathbb{R}^q$; (ii) the edges of G_n are generated independently and with probability $W_{n,i,j} := \rho_n \cdot W(\omega_i, \omega_j)$, where $\rho_n \in (0, 1]$ is a constant called the sparsifying factor[1].

The graphon model includes various widely researched graph types, such as Erdos-Renyi, inhomogeneous random graphs, stochastic block models, degree-corrected block models, random exponential graphs, and geometric random graphs as special cases; see (Lovász, 2012) for a more detailed discussion.

This paper's key contribution is the analysis of a linear Graph Neural Network model (LG-GNN) which can provably estimate the edge probabilities in a graphon model. Specifically, we present a GNN-based algorithm that yields estimators, denoted as $\hat{p}_{i,j}$, that converge to true edge probabilities $\rho_n W(\omega_i, \omega_j)$. Crucially, these estimators have mean squared error converging to 0 at the rate $o_{n\to\infty}(\rho_n^2)$. To our knowledge, this work is the first to rigorously characterize the ability of GNNs to estimate the underlying edge probabilities in general graphon models.

The estimators $\hat{p}_{i,j}$ are constructed in two main steps. We first employ LG-GNN (Algorithm 1), to embed the vertices of G_n. Concretely, for each vertex $i \in [n]$, LG-GNN computes a set $\Lambda_i = \{\lambda_i^0, \lambda_i^1, \ldots, \lambda_i^L\}$, where $\lambda_i^k \in \mathbb{R}^{d_n}$ and d_n is the embedding dimension. Then, Λ_i, Λ_j are used to construct estimators $\hat{q}_{i,j}^{(k)}$ for the *moments* of W. We refer the reader to Section 4 for the formal definition of the moments. Intuitively, the kth moment $W_{n,i,j}^{(k)}$ represents the probability that there is a path of length k between vertices with latent features ω_i and ω_j in G_n.

The next step is to show that when the number of distinct nonzero eigenvalues of W, denoted m_W, is finite, then the edge probabilities $W_{n,i,j}$ can be written as a linear function of the moments $W_{n,i,j}^{(2:m_W+1)}$. This naturally motivates Algorithm 2, which learn the $W_{n,i,j}$'s from the moment estimators $\hat{q}_{i,j}^{(2:m_W+1)}$, using a constrained regression. The regression coefficients $\hat{\beta}^{n,m_W}$ are then used to produce estimators $\hat{p}_{i,j}$ for $W_{n,i,j}$.

The main result of the paper (stated in Theorem 4.4) presents the convergence rate of the mean square error of $\hat{\beta}^{n,m_W}$. It shows that if L, the number of message-passing layers in

[1]Department of Statistics, Harvard University [2]Department of Management Science & Engineering, Stanford University. Correspondence to: Alan Chung <alanchung@g.harvard.edu>.

---

[1]Throughout the paper, we will assume $(\omega_i) \sim \text{Unif}[0, 1]$. This

is without loss of generality. For any graphon $\tilde{W}$ with features in some arbitrary $\Omega \subset \mathbb{R}^q$ sampled from μ on Ω, there exists some graphon W with latent features drawn from $\text{Unif}[0, 1]$ so that the graphs generated from these two graphons are equivalent in law. See Remark 4 in (Davison & Austern, 2023) for more details.

LG-GNN, is at least $m_W - 1$, then the mean square error converges to 0. That implies that our estimators for the edge probabilities $W_{n,i,j}$ are consistent. For $L < m_W - 1$, the theorem provides the rate at which the mean square error decreases when L increases. The second main result, stated in Proposition 4.5, gives statistical guarantees on how well LG-GNN can detect in-community edges in a symmetric stochastic block model. A notable feature of this theorem is that the implied convergence rate is much faster than that of Theorem 4.4, which demonstrates mathematically that *ranking* high and low probability edges is easier than estimating the underlying probabilities of edges.

Finally, we would like to highlight two key aspects of the results of the paper. Firstly, our statistical guarantees for edge prediction are proven for scenarios when node features are absent and the initial node embeddings λ_i^0 are chosen at random. This underscores that effective link prediction can be achieved solely through the appropriate selection of GNN architecture, even in the absence of additional node data. The second point relates to graph sparsity: although graphons typically produce dense graphs, introducing the sparsity factor ρ_n results in vertex degrees of $O(\rho_n \cdot n)$, facilitating the exploration of sparse graphs. Our findings are pertinent for $\log(n)/n \ll \rho_n \leq 1$. Note that a sparsity of $\log(n)/n$ is the necessary threshold for connectivity (Spencer, 2001), highlighting the generality of our results.

While the primary focus of this paper is theoretical, we complement our theoretical analysis with experimental evaluations on real-world datasets (specifically, the Cora dataset) and graphs derived from random graph models. Our empirical observations reveal that in scenarios where node features are absent, LG-GNN exhibits performance comparable to the traditional Graph Convolutional Network (GCN) on simple random graphs, and surpasses GCN in more complex graphs sampled from graphons. Additionally, LG-GNN presents two further benefits: LG-GNN does not involve any parameter tuning (e.g., through the minimization of a loss function), resulting in significantly faster operation, and it avoids the common oversmoothing issues associated with the use of numerous message-passing layers.

1.1. Organization of the Paper

Section 2 discusses related works and introduces the motivation for our paper. Section 3 introduces our notation and presents an outline for our exposition. Section 4 presents our main results and Section 5 states a negative result for naive GNN architectures with random embedding initialization. Lastly, Section 6 discusses the issues of identifiability, and Section 7 presents our experimental results.

2. Related Works

Link prediction on graphs have a wide range of applications in domains ranging from social network analysis to drug discovery (Hasan & Zaki, 2011; Abbas et al., 2021). A survey of techniques and applications can be found in (Kumar et al., 2020; Martínez et al., 2016; Djihad Arrar, 2023).

Much of the existing theory on GNNs is regarding their expressive power. For example, (Xu et al., 2018; Morris et al., 2021) show that GNNs with deterministic node initializations have expressive power bounded by that of the 1-dimensional Weisfeiler-Lehman (WL) graph isomorphism test. Generalizations such as k-GNN (Morris et al., 2021) have been proposed to boost the expressive power higher in the WL-hierarchy. The Structural Message Passing GNN (SGNN) (Vignac et al., 2020) was also proposed and was shown to be universal on graphs with bounded degrees, and converges to continuous "c-SGNNs" (Keriven et al., 2021), which were also shown to be universal on many popular random graph models. Lastly, (Abboud et al., 2020) showed that GNNs that use random node initializations are universal, in that they can approximate any function defined on graphs with fixed order.

A recent wave of works focus on deriving statistical guarantees for graph representation algorithms. A common data-generating model for the graph is a graphon model (Lovász & Szegedy, 2006; Borgs et al., 2008; 2012). A large literature has been devoted to establishing guarantees for community detection on graphons such as the stochastic block model; see (Abbe, 2018) for an overview. For this task, spectral embedding methods have long been proposed (see (Deng et al., 2021; Ma et al., 2021) for some recent examples). Lately, statistical guarantees for modern random walk-based graph representation learning algorithms have also been obtained. Notably (Davison & Austern, 2021; Barot et al., 2021; Qiu et al., 2018; Zhang & Tang, 2021) characterize the asymptotic properties of the embedding vectors obtained by deepwalk, node2vec, and their successors and obtain statistical guarantees for downstream tasks such as edge prediction. Recently, some works also aim at obtaining learning guarantees for GNNs. Stability and transferability of certain untrained GNNs have been established in (Ruiz et al., 2021; Maskey et al., 2023; Ruiz et al., 2023; Keriven et al., 2020). For example (Keriven et al., 2020) shows that for relatively sparse graphons, the embedding produced by an untrained GNN will converge in L^2 to a limiting embedding that depends on the underlying graphon. They use this to study the stability of the obtained embeddings to small changes in the training distribution. Other works established generalization guarantees for GNNs. Those depend respectively on the number of parameters in the GNN (Maskey et al., 2022), or on the VC dimension and Radamecher complexity (Esser et al., 2021)

of the GNN.

Differently from those two lines of work, our paper studies when link prediction is possible using GNNs, establishes statistical guarantees for link prediction, and studies how the architecture of the GNN influences its performance. More similar to our paper is (Kawamoto et al., 2018), which exploits heuristic mean-field approximations to predict when community detection is possible using an untrained GNN. Note, however, that contrary to us, their results are not rigorous and the accuracy of their approximation is instead numerically evaluated. (Lu, 2021) formally established guarantees for in-sample community detection for two community SBMs with a GNN trained via coordinate descent. However, our work establishes learning guarantees for general graphons beyond two-community SBMs, both in the in-sample and out-sample settings. Moreover, the link prediction task we consider, while related to community detection, is still significantly different. (Baranwal et al., 2021) studies node classification for contextual SBMs and shows that an oracle GNN can significantly boost the performance of linear classifiers. Another related work (Magner et al., 2020) studies the capacity of GNN to distinguish different graphons when the number of layers grows at least as $L = \Omega(\log(n))$. Interestingly they find that GNN struggles in differentiating graphons whose expected degree sequence is not sufficiently heterogeneous, which unfortunately occurs for many graphon models, including the symmetric SBM. It is interesting to note that in Proposition 5.1 we will show that this is also the regime where the classical GCN fails to provide reliable edge probability prediction. Finally, some learning guarantees have also been derived for other graph models. Notably (Alimohammadi et al., 2023) studied the convergence of GraphSAGE and related GNN architectures under local graph convergence.

3. Notation and Preliminaries

In this section, we present our assumptions, some background regarding GNNs, and the link prediction goals that we focus on.

3.1. Assumptions

As mentioned in the introduction, the random graph $G_n = (V_n, E_n)$ with the vertex set $V_n = \{1, 2, \cdots, n\}$ is sampled from a graphon $W : [0,1]^2 \to [0,1]$, where each vertex $i \in V_n$ draws latent feature $(\omega_i) \overset{i.i.d}{\sim} \text{Unif}[0,1]$ and the edges are generated independently and with probability $W_{n,i,j} := \rho_n \cdot W(\omega_i, \omega_j)$. We let $A = (a_{ij})$ denote the adjacency matrix. When the graph and context are clear, we let $W_n := \rho_n W$, and let $W_{n,i,j} := \rho_n W(\omega_i, \omega_j)$. We

make the following three assumptions:

$$\log(n)/n \ll \rho_n \leq 1 \qquad (H_1)$$
$$\exists\, \delta_W > 0 \text{ s.t. } \delta_W \leq W(\cdot, \cdot) \leq 1 - \delta_W \qquad (H_2)$$
$$W \text{ is a Hölder-by-parts function} \qquad (H_3)$$

We refer the reader to Appendix B for a more detailed discussion.

3.2. Graph Neural Networks

An L-layer GNN, comprised of L processing layers, transforms graph data into numerical representations, or embeddings, of each each vertex. Concretely, a GNN associates each vertex $i \in [n]$ to some $\lambda_i^L \in \mathbb{R}^{d_n}$, where we call d_n the embedding dimension. The learned embeddings are then used for downstream tasks such as node prediction, graph classification or link prediction, as investigated in this paper.

A GNN computes the embeddings iteratively through message passing. We let λ_i^k denote the embedding produced for vertex i after k GNN iterations. As such, λ_i^0 denotes the initialization of the embedding for vertex u. The message passing layer can be expressed generally as

$$\lambda_i^{k+1} = \phi\left(\lambda_i^k, \bigoplus_{j \in N(i)} \psi(\lambda_i^k, \lambda_j^k, e_{ij})\right),$$

where $N(i)$ is the set of neighbors of vertex i, ϕ, ψ are continuous functions, e_{ij} is the feature of the edge (u, v), and $\bigoplus$ is some permutation-invariant aggregation operator, for example, the sum (Wu et al., 2022).

One classical architecture is the Graph Convolutional Network (GCN) (Kipf & Welling, 2017), whose update equation is given by

$$\lambda_i^k = \sigma\left(M_{k,0}\lambda_i^{k-1} + M_{k,1} \sum_{j \in N(i)} \frac{\lambda_j^{k-1}}{\sqrt{|N(i)| \cdot |N(j)|}}\right), \tag{1}$$

where $\sigma(\cdot)$ is a non-linear function and $M_{k,0}, M_{k,1} \in \mathbb{M}_{d_n \times d_n}(\mathbb{R})$ are matrices. These matrices are chosen by minimizing some empirical risk during a training process, typically through gradient descent.

In some settings, additional node features for each vertex are given, and the initialization λ_i^0 is chosen to incorporate this information. In this paper, we focus on the setting when no node features are present, and a natural way to initialize our embeddings λ_i^0 is at random. One of our key messages is that even without additional node information, link prediction is provably possible with a correct choice of GNN architecture.

3

3.3. Link Prediction

Given a graph $G_n = ([n], E_n)$ generated from a graphon, potential link prediction tasks are (a) to determine which of the non-edges are most likely to occur, or (b) to estimate the underlying probability of a particular edge (i, j) according to the graphon. Here, we make the careful distinction between two different link prediction evaluation tasks. One task is regarding the **ranking** of a set of test edges. Suppose a set of test edges $e_1, e_2, \ldots, e_k$ has underlying probabilities $p_{e_1} \geq p_{e_2} \geq \cdots \geq p_{e_k}$ according to W. The prediction algorithm assigns a predicted probability $\hat{p}_{e_i}$ for each edge e_i and is evaluated on how well it can extract the true ordering (e.g., the AUC-ROC metric).

Another link prediction task is to estimate the underlying probabilities of edges in a random graph model. For example, in a stochastic block model, a practitioner might wish to determine the underlying connection probabilities, as opposed to simply determine the ranking. We will refer to this task as **graphon estimation**. It is important to note that the latter task is generally more difficult.

We also distinguish between two link prediction settings, i.e., the **in-sample** and **out-sample** settings. In in-sample prediction, the aim is to discover potentially missing edges between two vertices $i, j \in [n]$ already present at training time. On the contrary, in out-of-sample prediction, the objective is to predict edges among vertices that were not present at training. If $\tilde{V}$ are the set of vertices not present at training, the goal is to use the trained GNN to predict edges (i, j) for $i, j \in \tilde{V}$, or (i, j) for $i \in V_{train}$ and $j \in \tilde{V}$.

4. Main Results

We introduce the *Linear Graphon Graph Neural Network*, or LG-GNN in Algorithm 1. The algorithm starts by assigning each node i a random feature $Z_i \sim \frac{1}{\sqrt{d_n}} \mathcal{N}(0, I_{d_n})$, where $d_n = \Omega(1/\rho_n)$ is the embedding dimension. The first message passing layer computes λ_i^0 by summing Z_j for all $j \in N(i)$, scaled by $\frac{1}{\sqrt{n}}$. The subsequent layers normalize the λ_j^k's by $1/n$ before adding them to λ_i^k. We show in Proposition D.3 and Lemma D.4 that this procedure essentially counts the number of paths between pairs of vertices. Specifically, $\mathbb{E}[\langle \lambda_i^{k_1}, \lambda_j^{k_2} \rangle | A, (\omega_\ell)]$ is a linear combination of the "empirical moments" of W (Equation (7)). The second stage of Algorithm 1 then recovers these empirical moments by decoupling the aforementioned linear equations. We refer the reader to Appendix D.2 for more details and intuition behind LG-GNN.

We note that the scaling of $1/\sqrt{n}$ in the first message passing layer is crucial in allowing the embedding vectors (λ_ℓ^k) to learn information about the latent features (ω_ℓ) asymptotically. We show in Proposition 5.1 that without this construc-

tion, the classical GCN is unable to produce meaningful emebddings with random feature initializations.

4.1. Statistical Guarantees for Moment Estimation

Define the kth moment of a sparsified graphon W_n as $W_n^{(k)}(x, y) :=$

$$\int_{[0,1]^{k-1}} W_n(x, t_1) W_n(t_1, t_2) \ldots W_n(t_{k-1}, y) \mathrm{d}t_{1:k-1},$$

which is the probability that there is a path of a length k between two vertices with latent features x, y, averaging over the latent features of the vertices in the path. As with the graphon itself, we denote $W_{n,i,j}^{(k)} := W_n^{(k)}(\omega_i, \omega_j)$. The following proposition shows that the estimators $\hat{q}_{i,j}^{(k)}$ are consistent estimators for these moments $W_{n,i,j}^{(k)}$.

Proposition 4.1. *Suppose that the graph $G_n = ([n], E_n)$ is generated according to a graphon $W_n = \rho_n W$. Suppose that assumptions (H_2) and (H_3) hold. Then, with probability at least $1 - 5/n - n \cdot \exp(-\delta_W \rho_n(n-1)/3)$, for all $2 \leq k \leq L + 2$,*

$$\left| \hat{q}_{i,j}^{(k)} - W_{n,i,j}^{(k)} \right| \leq \frac{\rho_n^{k-1}}{\sqrt{n-1}} \log(n)^k \left[3a_k \sqrt{\rho_n} + \frac{96a_{k-1}}{\sqrt{d_n}} \right], \tag{2}$$

where $a_k = C(8(k+2))^k k^{k+1} \sqrt{k!}$ and C is some absolute constant.

4.2. Edge Prediction Using the Moments of the Graphon

Proposition 4.1 relates the embeddings produced by LG-GNN to the underlying graph moments. We show in Theorem 4.4 that the $\hat{q}_{i,j}^{(k)}$s can be used to derive consistent estimators for the underlying edge probability $W_{n,i,j}$ between vertices i and j.

The key observation is that for any Hölder-by-parts graphon W, there exists some $m \in \mathbb{N} \cup \{\infty\}$ such that

$$W(x, y) = \sum_{i=1}^{m} \mu_i \phi_i(x) \phi_i(y) \qquad \forall x, y \in [0, 1]$$

for some sequence of eigenvalues (μ_i) with $|\mu_i| \leq 1$ and eigenfunctions (ϕ_i) orthonormal in $L^2([0, 1])$. This, coupled with the Cayley-Hamilton theorem (Hamilton, 1853), implies that W can be re-expressed as a linear combination of its moments. We will refer to the number of distinct nonzero eigenvalues of W as m_W, which we call the *distinct rank*.

Proposition 4.2. *Suppose that $W : [0, 1]^2 \to [0, 1]$ is a Hölder-by-parts graphon. Then, there exists a vector*

Algorithm 1 LG-GNN architecture

Input: a Graph $G_n = ([n], E_n)$; $L \geq 0$
Output: estimators $\hat{q}_{i,j}^{(k)}$ for the kth moments $W_{n,i,j}^{(k)}$.

Sample $(Z_i)_{i=1}^n \overset{iid}{\sim} \frac{1}{\sqrt{d_n}} \mathcal{N}(0, I_{d_n})$.
GNN Iteration:
for $i \in [n]$ **do**
$\quad \lambda_i^0 \leftarrow \frac{1}{\sqrt{n-1}} \sum_{\ell=1}^n a_{i\ell} Z_\ell$
end
for $k \in [L]$ **do**
$\quad$ **for** $i \in [n]$ **do**
$\quad\quad \lambda_i^k \leftarrow \lambda_i^{k-1} + \frac{1}{n-1} \sum_{\ell \leq n} a_{i\ell} \lambda_\ell^{k-1}$
$\quad$ **end**
end

Computing Estimators for $W_{n,i,j}^{(k)}$**:**
for $i \neq j$ **do**
$\quad \hat{q}_{i,j}^{(2)} := \langle \lambda_i^0, \lambda_j^0 \rangle$.
end
for $k \in \{3, 4, \ldots, L+2\}$ **do**
$\quad \hat{q}_{i,j}^{(k)} := \langle \lambda_i^{k-2}, \lambda_j^0 \rangle - \sum_{r=0}^{k-3} \binom{k-2}{r} \hat{q}_{i,j}^{(r+2)}$
end
Return: $\left\{ (\hat{q}_{ij}^{(2)}, \hat{q}_{ij}^{(3)}, \ldots, \hat{q}_{ij}^{(L+2)})_{i \neq j} \right\}$

$\beta^{*,m_W} = \left(\beta_1^{*,m_W}, \beta_2^{*,m_W} \ldots, \beta_{m_W}^{*,m_W} \right)$ *such that for all* $(x,y) \in [0,1]^2$,

$$W(x,y) = \sum_{i=1}^{m_W} \beta_i^{*,m_W} W^{(i+1)}(x,y). \tag{3}$$

The above suggests the following algorithm for edge prediction using the embedding produced by LG-GNN.

Algorithm 2 LG-GNN edge prediction algorithm

Input: Graph $G_n = ([n], E_n)$, search space $\mathcal{F}$, threshold β; $L \geq 0$.
Output: Set of predicted edges
Using Algorithm 1, compute $q_{i,j}^{(2:L+2)} := (\hat{q}_{i,j}^{(2)}, \ldots, q_{i,j}^{(L+2)})$ for every vertex i, j
Compute:

$$\hat{\beta}^{n,L+1} = \arg\min_{\beta \in \mathcal{F}} \sum_{i \neq j} \left(\left\langle \beta, \hat{q}_{i,j}^{(2:L+2)} \right\rangle - a_{i,j} \right)^2$$

Compute: $\hat{p}_{i,j} := \left\langle \hat{\beta}^{n,L+1}, \hat{q}_{i,j}^{(2:L+2)} \right\rangle$ for all i, j
Return: $\{ (i,j) \mid \hat{p}_{i,j} \geq \gamma \}$ the set of predicted edges.

Algorithm 2 estimates the edge probabilities by regressing the moment estimators $\hat{q}_{i,j}^{(2:L+2)}$ onto the a_{ij}'s. The coeffi-

cients of the regression are chosen through constrained optimization. This is necessary due to high multi-collinearity among the observations $\hat{q}_{i,j}^{(2:L+2)}$. Other methods to control the multi-collinearity include using Partial Least Squares (PLS) regression in Algorithm 2. This leads to an alternative algorithm presented in Algorithm 3, called PLSG-GNN, that is also evaluated in the experiments section.

Before stating our main theorem, we define a few quantities.
Definition 4.3 (MSE error). For any vector $\beta \in \mathbb{R}^k$, define the mean squared error

$$R_T(\beta) = \mathbb{E}\left[\left(\left\langle \beta, \hat{q}_{n+1,n+2}^{(2,\text{len}(\beta)+1)} \right\rangle - W_n(\omega_{n+1}, \omega_{n+2}) \right)^2 \right],$$

where the expectation is taken with respect to the randomness in $\omega_{n+1}, \omega_{n+2}$. We interpret $n+1, n+2$ as being two new vertices that were not present at training time. We also define the following quantity, used in the statement of Theorem 4.4:

$$R(\beta) = \mathbb{E}\left[\left(\left\langle \beta, W^{(2,\text{len}(\beta)+1)}(x,y) \right\rangle - W(x,y) \right)^2 \right].$$

For some set $\mathcal{F} \subset \mathbb{R}^k$, we can interpret $\arg\min_{\beta \in \mathcal{F}} R(\beta)$ as the "L^2 projection" of $W(x,y)$ onto the subspace spanned by $\langle \beta, W^{(2:k+1)}(x,y) \rangle$. In particular, if $\mathcal{F}$ contains a vector $\beta^{*,k}$ satisfying Equation (3), then $\arg\min_{\beta \in \mathcal{F}} R(\beta) = 0$. In the context of Algorithm 2, this suggests that we can obtain a consistent estimator for $W_{n,i,j}$. Hence, since Proposition 4.2 guarantees that such a $\beta^{*,k}$ exists when $m_W < \infty$, the intuition is that if both the search space and number of layers $\mathcal{F}, L$ are sufficiently large, then Algorithm 2 should produce estimators $\hat{p}_{i,j}$ that are consistent.

The following theorem shows that this intuition is indeed true. It states that $\hat{p}_{i,j}$ is a consistent estimator for the edge probability $W_{n,i,j}$ if the number of LG-GNN layers is large enough, and characterizes its convergence rate. We will show this when the search space $\mathcal{F}$ is a rectangle of the form $\mathcal{F} := \prod_{i=1}^{L+1} [-\frac{b_i}{\rho_n^i}, \frac{b_i}{\rho_n^i}] \subset \mathbb{R}^{L+1}$ for some $b_i > 0$. We discuss the implications of this result after its statement.

Theorem 4.4 (Main Theorem). *Let* $G_n = ([n], E_n)$ *be sampled from some graphon* $\rho_n W$, *where* W *satisfies* (H_2) *and* (H_3). *Take* $\hat{\beta}^{n,L+1}$ *to be the estimators given by Algorithm 2. Define* $\beta^{*,L+1} \in \arg\min_{\beta \in \mathcal{F}} R(\beta)$ *to be the population minimizer. Then, with probability at least* $1 - 5/n - n \cdot \exp(-\delta_W \rho_n (n-1)/3)$, *the MSE converges at rate*

$$R_T(\hat{\beta}^{n,L+1}) \leq R(\beta^{*,L+1}) + \tilde{O}\left(\frac{\kappa_1^2 \rho_n^2}{\sqrt{n}} \right)$$

$$+ \kappa_1 \kappa_2 \frac{\rho_n \cdot \log(n)^{L+1}}{\sqrt{n}} \left[\sqrt{\rho_n} + \frac{1}{\sqrt{d_n}} \right],$$

where $\kappa_1 = O((1 - \delta_W) \sum_{i=1}^{L+1} |b_i|(1 - \delta_W)^i)$ *and* $\kappa_2 = O(\sum_{i=1}^{L+1} |b_i|).$

We remark that when d_n increases quickly enough, the inequality in Theorem 4.4 implies that

$$R_T(\hat{\beta}^{n,L+1}) \leq R(\beta^{*,L+1}) + O\left(\frac{\log(n)^{L+1} \cdot \rho_n^{3/2}}{\sqrt{n}}\right).$$

In particular, when $\rho_n \gg \log(n)^{2L+2}/n$, and d_n increases fast enough, then $R_T(\hat{\beta}^{n,L+1}) \leq R(\beta^{*,L+1}) + o(\rho_n^2)$, i.e. the MSE decreases faster than the sparsity of the graph.

As mentioned in the discussion preceeding Theorem 4.4, if the search space S is large enough to contain some vector $\beta^{*,L+1}$ such that $W(x,y) = \sum_{i=1}^{L+1} \beta^{*,i} W^{(i+1)}(x,y)$ for all x, y, then $R(\beta^{*,L+1}) = 0$, and the MSE converges to 0. Notably, Proposition 4.2 implies that this search space exists for $L \geq m_W - 1$. In this sense, m_W captures the "complexity" of W, and each layer of LG-GNN extracts an additional order of complexity.

When $L = m_W - 1$, in order for the search space S to contain the β^{*,m_W} defined in Proposition 4.2, we require that $b_i > \beta_i^{*,m_W}$, where b_i is defined in Theorem 4.4. Considering the proof of Proposition 4.2, if $m_W < \infty$, then b_{m_W} is on the order of $\frac{1}{|\mu_1 \mu_2 ... \mu_{m_W}|}$, and hence the constant κ_2 in Proposition 4.2 is on this order as well. This dependence on the inverse of small eigenvalues is a statistical bottleneck; it turns out, however, that if we are concerned only with edge ranking, instead of graphon estimation, this dependence can be greatly reduced. This is outlined in Proposition 4.5.

If Algorithm 2 is used for predicting all the edges that have a probability of more than $\gamma > 0$ of existing, then the 0-1 loss will also go to zero. Indeed for almost every $\gamma > 0$ we have

$$\frac{1}{n^2} \sum_{i,j \leq n} \mathbb{I}(\hat{p}_{i,j} \geq \gamma) - \mathbb{I}(\rho_n W(\omega_i, \omega_j) \geq \gamma) \xrightarrow{p} 0.$$

Furthermore, if $L < m_W - 1$ is smaller than the number of distinct eigenvalues of W, then we have

$$R(\beta^{*,L+1}) \leq \sqrt{\sum_{s=1}^{m_W} \left[\sum_{r=L+1}^{m_W} \beta_r^{*,m_W} \left(\mu_s^{r+1} - \mu_s^{L+1}\right)\right]^2},$$

implying that the the $R(\beta^{*,L+1})$ in Theorem 4.4 decreases as L increases. This latter bound is proved in Lemma F.1.

4.3. Preserving Ranking in Link Prediction

Theorem 4.4 states that under general conditions, LG-GNN yields a consistent estimator for the underlying edge probability $W_n(\omega_i, \omega_j) = \rho_n W_{i,j}$. However, estimating edge

probabilities is strictly harder than discovering a set of high-probability edges. In practical applications, one often cares about *ranking* the underlying edges, i.e., whether an algorithm can assign higher probabilities to positive test edges than to negative ones. Metrics such as the AUC-ROC and Hits@k capture this notion. The following proposition characterizes the performance of LG-GNN in ranking edges in a k-community symmetric SBM. See Appendix A.3 for more details about SBMs.

Before stating the proposition, we define the following notation for a k-community symmetric SBM. Let $S_{in} = \{(i,j)|\text{vertices } i, j \text{ belong to the same community}\}$, $S_{out} = \{(k,\ell)|\text{vertices } k, \ell \text{ are in different communities}\}$, and define

$$E_{rank} := \left\{ \min_{(i,j) \in S_{in}} \hat{p}_{i,j} > \max_{(k,\ell) \in S_{out}} \hat{p}_{k,\ell} \right\}$$

to be the event that the predicted probabilities for all of the in-community edges are greater than all of the predicted probabilities for the across-community edges, i.e., LG-GNN achieves perfect ranking of the graph edges. We will prove that this event happens with high probability. See Proposition G.1 for the full proposition.

Proposition 4.5 (Informal). *Consider a k-community symmetric stochastic block model with parameters $p > q$ and sparsity factor ρ_n. Let $\mu_1 = \frac{p+(k-1)q}{k} > \frac{p-q}{k} = \mu_2$ be the eigenvalues of the associated graphon. Suppose that the search space $\mathcal{F}$ is such that $\{\beta \in \mathbb{R}^{L+1} | \ ||\beta||_{L^1} \leq (\mu_1 \rho_n)^{-1}\} \subseteq \mathcal{F}$.*

Produce probability estimators $\hat{p}_{i,j}$ for the probability of an edge between vertices i and j using Algorithm 1 and Algorithm 2 with parameters $L, \mathcal{F}$ where $L \geq 1$. Then, there exists a constant $A > 0$ such that when

$$\frac{\log(n)^{L+1}}{\rho_n \sqrt{n}} \left[\sqrt{\rho_n} + \frac{1}{\sqrt{d_n}}\right] \leq A\mu_2^3$$

holds, then with high probability, E_{rank} occurs, i.e., LG-GNN correctly predicts higher probability for all of the in-community edges than for cross-community edges.

Proposition 4.5 gives conditions under which LG-GNN achieves perfect ranking on a k-community symmetric SBM. One subtle but important point is the implied convergence rate. In Proposition 4.5, the size of the search space is required only to be on the order of $(\mu_1 \rho_n)^{-1}$. In the notation of Theorem 4.4, this means that the constant κ_2 is upper bounded by $1/\mu_1$, which indicates a much faster rate of converge than the rate that is required by Theorem 4.4 to define consistent estimators. This confirms the intuition that ranking is easier than graphon estimation, and in particular, should be less sensitive to small eigenvalues. Proposition 4.5 demonstrates the extent to which ranking is easier than graphon estimation.

5. Performance of the Classical GCN Architecture

As mentioned in Section 4, in the context of random node initializations, a naive choice of GNN architecture can cause learning to fail. In the following proposition, we demonstrate that for a large class of graphons, the Classical GCN architecture with random initializations results in embeddings that cannot be informative in out-of-sample graphon estimation. To make this formal, we assume that at training, only $n - m$ vertices are observable. We denote by $G_n|_{V_{n-m}}$ the induced subgraph with vertex set $V_{n-m} = \{1, \ldots, n - m\}$. And we consider graphons that are such that

$$\text{The function } W : x \to \int_0^1 W(x, y)dy \text{ is constant.} \quad (H_4)$$

Note that many graphons satisfy this assumption, including symmetric SBMs.

Proposition 5.1. *Suppose that the graph $G_n = ([n], E_n)$ is generated according to a graphon $W_n = \rho_n W$. Moreover assume that Assumptions (H_1), (H_2), (H_4) hold.*

Suppose that the initial embeddings $(\lambda_i^0) \overset{i.i.d}{\sim} \mu$ are so that each coordinate is generated i.i.d. from a $\frac{s^2}{\sqrt{d_n}}$ sub-Gaussian distribution. Assume that the subsequent embeddings (λ_i^ℓ) are computed iteratively according to Equation (1), where $\sigma(\cdot)$ is taken to be $1-$Lipschitz and where the weight matrices $(M_{k,0}, M_{k,1})$ are trained on $G_n|_{V_{n-m}}$ and satisfy $\|M_{k,0}\|_{op}, \|M_{k,1}\|_{op} \overset{a.s}{\leq} M$.

Then, there exist random variables μ_n^ℓ, $\ell \in [L]$, that are independent of $\omega_{n-m+1}, \ldots, \omega_n$ such that for a certain $\kappa > 0$ with probability at least $1 - \frac{2}{n} - 2ne^{-\frac{12\log(n)}{\rho_n}} - 2ne^{-\kappa d_n}$,

$$\sup_{\ell \leq L} \|\lambda_n^L - \mu_n^\ell\| \leq \frac{K}{\sqrt{\rho_n(n-1)}}, \quad (4)$$

where $K > 0$ is an absolute constant.

We show that this leads to suboptimal risk for graphon estimation. For simplicity we show this for dense graphons, e.g when $\rho_n = 1$.

Proposition 5.2. *Suppose that the conditions of Proposition 5.1 hold. Moreover assume that the graphon $W(\cdot, \cdot)$ is not constant and that $\rho_n = 1$ for all $n \in \mathbb{N}$. Then, there exists some constant $K > 0$ such that for any Lipchitz prediction rule $f(\cdot, \cdot)$, for all vertices $i \in [n]$, we have*

$$\mathbb{E}\Big(\big[W(\omega_i, \omega_n) - f(\lambda_i^L, \lambda_n^L) \big]^2 \Big) \geq K + o_n(1). \quad (5)$$

Proposition 5.1 and Proposition 5.2 imply that in the out-of-sample setting, the embeddings produced by Equation (1) with random node feature initializations will lead to suboptimal estimators for the edge probability $W(\omega_i, \omega_i)$.

A key feature of the proof of Proposition 5.1 is that $\sum_{u \in N(v)} \frac{\lambda_u^{k-1}}{\sqrt{|N(u)\|N(v)|}}$ concentrates to 0 very quickly with random node initializations. This demonstrates the importance of the (subtle) construction of the first round of message passing λ_u^0 in Algorithm 1. We also note that Proposition 5.2 doesn't necessarily imply that predicted probabilities $\hat{p}_{e_i}$ will be ineffective at ranking test edges, though we do see in the experiments that the performance is decreased for the out-of-sample case.

6. Identifiability and Relevance to Common Random Graph Models

We remark that a key feature of LG-GNN is that it uses the embedding vector λ_i^k produced at each layer. Indeed $\hat{p}_{i,j}$ depends on all of the terms $\{\langle \lambda_i^0, \lambda_j^0 \rangle, \langle \lambda_i^0, \lambda_j^1 \rangle, \ldots, \langle \lambda_i^0, \lambda_j^L \rangle\}$. This is in contrast to many classical ways of using GNNs for link prediction that depend only on $\langle \lambda_i^L, \lambda_j^L \rangle$. The following proposition shows that this construction is necessary to obtain consistent estimators.

Proposition 6.1. *For any $L \geq 0$, there exists a 2-community stochastic block model, such that for every continuous function $f : \mathbb{R} \to \mathbb{R}$ we have*

$$f(\langle \lambda_i^L, \lambda_j^L \rangle) \overset{p}{\nrightarrow} W(\omega_i, \omega_j).$$

This notably implies that

$$\liminf_{n, d_n \to \infty} \inf_{f \in c^0(\mathbb{R})} \mathbb{E}\Big(\big(f(\langle \lambda_i^L, \lambda_j^L \rangle) - W(\omega_i, \omega_j) \big)^2 \Big) > 0$$

To illustrate this, consider the following example for $L = 0$.

Example 6.2. *Consider an 2 community symmetric SBM with edge connection probability matrix $\begin{pmatrix} 1/2 & 1/4 \\ 1/4 & 3/4 \end{pmatrix}$. The matrix of second moments, that is, the matrix of probabilities of paths of length two between members of the two communities is $\begin{pmatrix} 5/32 & 5/32 \\ 5/32 & 5/16 \end{pmatrix}$. Hence for any continuous function f and every vertex i, j, k belonging respectively to communities 1 for i, j and 2 for k then we have that $f(\langle \omega_i^0, \omega_j^0 \rangle) \overset{p}{\to} f(5/32)$ and $f(\langle \omega_i^0, \omega_k^0 \rangle) \overset{p}{\to} f(5/32)$ have the same limit. Hence this implies that no consistent estimator of $W(\cdot, \cdot)$ can be built by using only $(\langle \lambda_i^0, \lambda_j^0 \rangle)$.*

While this result is for the specific case of Algorithm 1, which in particular contains no non-linearities, we anticipate that this general procedure of learning a function that maps a set of dot products $\{\langle \lambda_i^{k_1}, \lambda_j^{k_2} \rangle\}_{k_1, k_2}$ to a predicted probability, instead of just $\langle \lambda_i^L, \lambda_j^L \rangle$, can lead to better performance for practioners on various types of GNN architectures.

7. Experimental Results

We compare experimentally a GCN, LG-GNN, and PLSG-GNN. We perform experiments on the Cora dataset (McCallum et al., 2000) in the in-sample setting. We also show results for various random graph models. The results for random graphs below are in the out-sample setting, more results are in Appendix I. We report the AUC-ROC and Hits@k metric, and also a custom metric called the Probability Ratio@k, which is more suited to the random graph setting. We refer the reader to Appendix I for a more complete discussion.

LG-GNN and PLSG-GNN perform similarly to the classical GCN in settings with no node features and can outperform it on more complex graphons. One major advantage of LG-GNN/PLSG-GNN is that they do not require extensive tuning of hyperparameters (e.g., through minimizing a loss function) and hence run much faster and are easier to fit. For example, training the 4-layer GCN resulted in convergence issues, even on a wide set of learning rates.

7.1. Real Data: Cora Dataset

The following results are in the in-sample setting. We consider when (a) the GCN has access to node features (b) the GCN does not.

Table 1. GCN has no access to node features

Params	Model	Hits@50	Hits@100
layers=2	GCN	0.496 ± 0.025	0.633 ± 0.023
	LG-GNN	0.565 ± 0.012	0.637 ± 0.006
	PLSG-GNN	$\mathbf{0.591} \pm 0.014$	$\mathbf{0.646} \pm 0.013$
layers=4	GCN	0.539 ± 0.008	$\mathbf{0.665} \pm 0.007$
	LG-GNN	0.564 ± 0.005	0.620 ± 0.008
	PLSG-GNN	$\mathbf{0.578} \pm 0.014$	0.637 ± 0.013

Table 2. GCN has access to node features

Params	Model	Hits@50	Hits@100
layers=2	GCN	$\mathbf{0.753} \pm 0.019$	$\mathbf{0.898} \pm 0.021$
	LG-GNN	0.555 ± 0.027	0.603 ± 0.034
	PLSG-GNN	0.577 ± 0.033	0.626 ± 0.042
layers=4	GCN	$\mathbf{0.609} \pm 0.072$	$\mathbf{0.776} \pm 0.069$
	LG-GNN	0.560 ± 0.013	0.601 ± 0.012
	PLSG-GNN	0.574 ± 0.025	0.625 ± 0.024

7.2. Synthetic Dataset: Random Graph Models

7.2.1. 10-COMMUNITY SYMMETRIC SBM

The following are results for a 10-community stochastic block model with parameter matrix P that has randomly generated entries. The diagonal entries $P_{i,i}$

are generated as Unif$(0.5, 1)$, and $P_{i,j}$ is generated as Unif$(0, \min(P_{i,i}, P_{j,j}))$. The specific connection matrix that was used is in Appendix I.

Table 3. $\rho_n = 1$

Params	Model	P-Ratio@100	AUC-ROC
layers=2	GCN	0.709 ± 0.125	0.716 ± 0.019
	LG-GNN	0.883 ± 0.016	0.734 ± 0.005
	PLSG-GNN	$\mathbf{0.886} \pm 0.016$	$\mathbf{0.735} \pm 0.005$
layers=4	GCN	0.645 ± 0.025	0.578 ± 0.109
	LG-GNN	0.879 ± 0.011	$\mathbf{0.786} \pm 0.002$
	PLSG-GNN	$\mathbf{0.883} \pm 0.013$	0.732 ± 0.001

Table 4. $\rho_n = 1/\sqrt{n}$

Params	Model	P-Ratio@100	AUC-ROC
layers=2	GCN	0.344 ± 0.021	0.493 ± 0.004
	LG-GNN	0.580 ± 0.020	0.497 ± 0.009
	PLSG-GNN	$\mathbf{0.586} \pm 0.035$	$\mathbf{0.521} \pm 0.008$
layers=4	GCN	0.285 ± 0.016	0.486 ± 0.006
	LG-GNN	$\mathbf{0.589} \pm 0.016$	$\mathbf{0.532} \pm 0.003$
	PLSG-GNN	0.578 ± 0.013	0.508 ± 0.011

7.2.2. GEOMETRIC GRAPH

Each vertex i has latent feature X_i generated uniformly at random on $\mathbb{S}^{d-1}$, $d = 11$. Two vertices i and j are connected if $\langle X_i, X_j \rangle \geq t = 0.2$, corresponding to a connection probability ≈ 0.26. Higher sparsity is achieved by adjusting the threshold t.

Table 5. $\rho_n = 1$

Params	Model	P-Ratio@100	AUC-ROC
layers=2	GCN	$\mathbf{1.000} \pm 0.000$	0.873 ± 0.020
	LG-GNN	$\mathbf{1.000} \pm 0.000$	0.915 ± 0.007
	PLSG-GNN	0.997 ± 0.005	$\mathbf{0.917} \pm 0.010$
layers=4	GCN	0.813 ± 0.021	0.591 ± 0.016
	LG-GNN	$\mathbf{1.000} \pm 0.000$	0.956 ± 0.001
	PLSG-GNN	$\mathbf{1.000} \pm 0.000$	$\mathbf{0.958} \pm 0.001$

Table 6. $\rho_n = 1/\sqrt{n}$

Params	Model	P-Ratio@100	AUC-ROC
layers=2	GCN	0.333 ± 0.017	0.840 ± 0.008
	LG-GNN	$\mathbf{0.523} \pm 0.037$	0.818 ± 0.022
	PLSG-GNN	0.423 ± 0.054	$\mathbf{0.842} \pm 0.017$
layers=4	GCN	0.313 ± 0.021	$\mathbf{0.848} \pm 0.021$
	LG-GNN	$\mathbf{0.570} \pm 0.016$	0.823 ± 0.010
	PLSG-GNN	0.510 ± 0.014	0.843 ± 0.013

8. Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

9. Acknowledgements

References

Abbas, K., Abbasi, A., Dong, S., Niu, L., Yu, L., Chen, B., Cai, S.-M., and Hasan, Q. Application of network link prediction in drug discovery. *BMC bioinformatics*, 22: 1–21, 2021.

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

Abboud, R., Ceylan, I. I., Grohe, M., and Lukasiewicz, T. The surprising power of graph neural networks with random node initialization. *arXiv preprint arXiv:2010.01179*, 2020.

Abdi, H. Partial least squares regression and projection on latent structure regression (pls regression). *Wiley interdisciplinary reviews: computational statistics*, 2(1):97–106, 2010.

Alimohammadi, Y., Ruiz, L., and Saberi, A. A local graph limits perspective on sampling-based gnns. *arXiv preprint arXiv:2310.10953*, 2023.

Baranwal, A., Fountoulakis, K., and Jagannath, A. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. *arXiv preprint arXiv:2102.06966*, 2021.

Barot, A., Bhamidi, S., and Dhara, S. Community detection using low-dimensional network embedding algorithms. *arXiv preprint arXiv:2111.05267*, 2021.

Bellec, P. C. Concentration of quadratic forms under a bernstein moment assumption. *arXiv preprint arXiv:1901.08736*, 2019.

Borgs, C., Chayes, J. T., Lovász, L., Sós, V. T., and Vesztergombi, K. Convergent sequences of dense graphs i: Subgraph frequencies, metric properties and testing. *Advances in Mathematics*, 219(6):1801–1851, 2008.

Borgs, C., Chayes, J. T., Lovász, L., Sós, V. T., and Vesztergombi, K. Convergent sequences of dense graphs ii. multiway cuts and statistical physics. *Annals of Mathematics*, pp. 151–219, 2012.

Davison, A. and Austern, M. Asymptotics of network embeddings learned via subsampling. *arXiv preprint arXiv:2107.02363*, 2021.

Davison, A. and Austern, M. Asymptotics of network embeddings learned via subsampling, 2023.

Deng, S., Ling, S., and Strohmer, T. Strong consistency, graph laplacians, and the stochastic block model. *The Journal of Machine Learning Research*, 22(1):5210–5253, 2021.

Djihad Arrar, Nadjet Kamel, A. L. A comprehensive survey of link prediction methods. *The Journal of Supercomputing*, 80:3902–3942, 2023.

Esser, P., Chennuru Vankadara, L., and Ghoshdastidar, D. Learning theory can (sometimes) explain generalisation in graph neural networks. *Advances in Neural Information Processing Systems*, 34:27043–27056, 2021.

Fabian, M., Habala, P., Hajek, P., Santalucia, V., Pelant, J., and Zizler, V. *Functional Analysis and Infinite-Dimensional Geometry*. CMS Books in Mathematics. Springer New York, 2013. ISBN 9781475734805. URL https://books.google.com/books?id=TWLaBwAAQBAJ.

Goemans, L. M. Chernoff bounds, and some applications. https://math.mit.edu/~goemans/18310S15/chernoff-notes.pdf, 2015.

Hamilton, W. R. *Lectures on Quaternions: Containing a Systematic Statement of a New Mathematical Method; of which the Principles Were Communicated in 1843 to the Royal Irish Academy; and which Has Since Formed the Subject of Successive Courses of Lectures, Delivered in 1848 and Subsequent Years, in the Halls of Trinity College, Dublin: with Numerous Illustrative Diagrams, and with Some Geometrical and Physical Applications*. Hodges and Smith, 1853.

Hasan, M. A. and Zaki, M. J. A survey of link prediction in social networks. *Social network data analytics*, pp. 243–275, 2011.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical*

Association, 58(301):13–30, 1963. ISSN 01621459. URL http://www.jstor.org/stable/2282952.

Kawamoto, T., Tsubaki, M., and Obuchi, T. Mean-field theory of graph neural networks in graph partitioning. *Advances in Neural Information Processing Systems*, 31, 2018.

Keriven, N., Bietti, A., and Vaiter, S. Convergence and stability of graph convolutional networks on large random graphs. *Advances in Neural Information Processing Systems*, 33:21512–21523, 2020.

Keriven, N., Bietti, A., and Vaiter, S. On the universality of graph neural networks on large random graphs, 2021.

Kim, J. and Vu, V. Concentration of multivariate polynomials and applications. *Combinatorica, to appear*.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks, 2017.

Kovács, István A., e. a. Network-based prediction of protein interactions. *Nature Communiations*, 2019.

Kumar, A., Singh, S. S., Singh, K., and Biswas, B. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, 553:124289, 2020.

Lovász, L. *Large networks and graph limits*, volume 60. American Mathematical Soc., 2012.

Lovász, L. and Szegedy, B. Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B*, 96(6):933–957, 2006.

Lu, W. Learning guarantees for graph convolutional networks on the stochastic block model. In *International Conference on Learning Representations*, 2021.

Ma, S., Su, L., and Zhang, Y. Determining the number of communities in degree-corrected stochastic block models. *The Journal of Machine Learning Research*, 22(1):3217–3279, 2021.

Magner, A., Baranwal, M., and Hero, A. O. The power of graph convolutional networks to distinguish random graph models. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 2664–2669. IEEE, 2020.

Martínez, V., Berzal, F., and Cubero, J.-C. A survey of link prediction in complex networks. *ACM computing surveys (CSUR)*, 49(4):1–33, 2016.

Maskey, S., Levie, R., Lee, Y., and Kutyniok, G. Generalization analysis of message passing neural networks on large random graphs. *Advances in neural information processing systems*, 35:4805–4817, 2022.

Maskey, S., Levie, R., and Kutyniok, G. Transferability of graph neural networks: an extended graphon approach. *Applied and Computational Harmonic Analysis*, 63:48–83, 2023.

McCallum, A. K., Nigam, K., Rennie, J., and Seymore, K. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks, 2021.

Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., and Tang, J. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 459–467, 2018.

Ruiz, L., Wang, Z., and Ribeiro, A. Graphon and graph neural network stability. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5255–5259. IEEE, 2021.

Ruiz, L., Chamon, L. F., and Ribeiro, A. Transferability properties of graph neural networks. *IEEE Transactions on Signal Processing*, 2023.

Spencer, J. *The Strange Logic of Random Graphs*. Algorithms and Combinatorics. Springer Berlin Heidelberg, 2001. ISBN 9783540416548. URL https://books.google.com/books?id=u2c3LpjWs7EC.

Stein, E. and Shakarchi, R. *Real Analysis: Measure Theory, Integration, and Hilbert Spaces*. Princeton University Press, 2009. ISBN 9781400835560. URL https://books.google.com/books?id=2Sg3Vug65AsC.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Vignac, C., Loukas, A., and Frossard, P. Building powerful and equivariant graph neural networks with structural message-passing, 2020.

Wu, L., Cui, P., Pei, J., and Zhao, L. *Graph Neural Networks: Foundations, Frontiers, and Applications*. Springer Singapore, Singapore, 2022.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Zhang, M. Graph neural networks: Link prediction. In Wu, L., Cui, P., Pei, J., and Zhao, L. (eds.), *Graph Neural*

Networks: Foundations, Frontiers, and Applications, pp. 195–223. Springer Singapore, Singapore, 2022.

Zhang, M. and Chen, Y. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

Zhang, Y. and Tang, M. Consistency of random-walk based network embedding algorithms. *arXiv preprint arXiv:2101.07354*, 2021.

A. Notation and Preliminaries

We let $\| \cdot \|_p$ be the vector Euclidean norm. The L^p norm over the probability space will be denoted $\|X\|_{L^p} = (\mathbb{E}[|X|^p])^{1/p}$.

A.1. Graph Notation

We let $A = (a_{ij})_{i,j=1}^n$ be the adjacency matrix of the graph. Let $(\omega_i)_{i=1}^n$ be the latent features of the vertices, generated from $\mathrm{Unif}(0,1)$. Let W be the graphon, and let ρ_n be the sparsifying factor. We denote $W_n := \rho_n W$ (we will typically be concerned only with W_n, since that is the graphon from which the graph is generated). Let $N(i)$ be the set of neighbors of a vertex i, and hence $|N(i)|$ is the degree of i.

We define the kth moment of a graphon W_n to be the function from $[0,1]^2 \to [0,1]$ given by

$$W_n^{(k)}(x,y) := \int_{[0,1]^{k-1}} W_n(x,t_1)W_n(t_1,t_2)\ldots W_n(t_{k-1},y)\mathrm{d}t_1 \ldots \mathrm{d}t_{k-1}. \tag{6}$$

Heuristically, if one fixes two vertices v_x, v_y with latent features x, y, then this is the probability of a particular path of length k from v_x, v_y, when averaging over the possible latent features of the vertices in the path. Correspondingly, we define the empirical kth moment between two vertices i and j to be

$$\hat{W}_{n,i,j}^{(k)} = \frac{1}{(n-1)^{k-1}} \sum_{r_1,\ldots,r_{k-1} \leq n} a_{ir_1} a_{r_1 r_2} \ldots a_{r_{k-1}j}. \tag{7}$$

A.2. GNN Notation

We let λ_i^k be the embedding for the ith vertex produced by the GNN after the kth layer. The linear GNN architecture is given by Equation (58):

$$\lambda_i^k = M_{k,0}\lambda_i^{k-1} + M_{k,1}\frac{1}{n-1} \sum_{\ell \leq n} a_{i\ell}\lambda_\ell^{k-1}, \tag{8}$$

where $M_{k,0}$ and $M_{k,1}$ denote the weight matrices of the GNN at the kth layer. We remark that LG-GNN corresponds to $M_{k,0} = M_{k,1} = \mathbb{I}d_{d_n}$ being the identity matrix. Let

$$N_s^k := \sum_{\substack{r_1,\ldots,r_k \in \{0,1\} \\ \sum_{i=1}^k r_i = s}} M_{k,r_1} M_{k-1,r_2} \ldots M_{1,r_k}, \tag{9}$$

which is a quantity that shows up naturally in the GNN iteration. The classical GCN architecture we consider is also given in Equation (1).

A.3. Stochastic Block Model

We define the stochastic block model, as it is a running model in this paper.

A stochastic block model $\mathrm{SBM(n,P)}$ is parameterized by the number of vertices n in the graph and a connection matrix $P \in \mathbb{R}^{k \times k}$. Each vertex belongs in a particular *community*, labeled $\{1,2,\ldots,k\}$. We assign each vertex to belong to community j with probability p_j. In this paper, we choose $p_j = 1/k$ for all $j \in [k]$. Let c_i denote the community of the ith vertex. The graph is generated as follows. For each pair of vertices $i \neq j$, we connect them with an edge with probability P_{c_i,c_j}. We also denote the *symmetric* stochastic block model by $\mathrm{SSBM(n,p,q)}$. The SSBM is a stochastic block model with only two parameters: the parameter matrix P is so that $P_{ii} = p$, $P_{ij} = q$ if $i \neq j$.

The following lemma details how to represent a SBM using a graphon.

Lemma A.1. *Consider a stochastic block model* $\mathrm{SBM(n,P)}$. *Suppose that* $P \in \mathbb{R}^{k \times k}$ *is a symmetric matrix and that* P *has spectral decomposition* $P = \sum_{i=1}^k \lambda_i v_i v_i^T$, *where* $\|v_i\|_2 = 1$. *Let* $W : [0,1]^2 \to [0,1]$ *be the corresponding graphon.*

Then $\mathrm{SBM(n,P)}$ *can be represented by a graphon as follows.* $W(x,y) = P_{ij}$ *if* $x \in [(i-1)/k, i/k)$ *and* $y \in [(j-1)/k, j/k]$. *The eigenvalues of* W *are given by* $\mu_i := \lambda_i/k$ *with corresponding eigenfunctions* $\phi_i(x)$, *where* $\phi_i(x) = \sqrt{k}(v_i)_j$ *if* $x \in [(j-1)/k, j/k]$.

This further implies that the eigenfunctions $\phi_i(x)$ are bounded above pointwise by $\sqrt{k}$, in that $|\phi_i(x)| \leq \sqrt{k}$ for all $i \in [k]$ and $x \in [0,1]$, since $\|v_i\|_2 = 1$, implying that each entry of v_i has norm bounded above by 1.

This proof of this lemma is a simple verification of the properties. We note that the eigenfunctions are scaled by $\sqrt{k}$ because they integrate to 1 in $L^2([0,1])$.

A.4. PLSG-GNN Algorithm

We state the PLSG-GNN Algorithm, which is an analog of Algorithm 2 that uses Partial Least Squares Regression (PLS). Let $g : \mathbb{N}^2 \to \mathbb{N}$ be an enumeration of the pairs (i,j), $i \neq j$. In the algorithm below, let PLS denote the Partial Least Squares algorithm as introduced in (Abdi, 2010).

Algorithm 3 LG-GNN edge prediction algorithm

Input: Graph $G = (V, E)$, set S, threshold β; L.
Output: Set of predicted edges
Using Algorithm 1, compute $q_{i,j}^{(2,L+2)} := (\hat{q}_{i,j}^{(2)}, \ldots, q_{i,j}^{(L+2)})$ for every vertex i, j. Define the matrix $\hat{Q}$ and vector $\vec{a}$, as $\hat{Q}_{g(i,j)} := \left(q_{i,j}^{(2:L+2)} \right)$, $i < j$, and $\vec{a}_{g(i,j)} = a_{i,j}$, $i < j$.
Compute:

$$\hat{\beta}^{n,L+1} = \text{PLS}(\hat{Q}, \tilde{a}).$$

Compute: $\hat{p}_{i,j} := \left\langle \hat{\beta}^{n,L+1}, \hat{q}_{i,j}^{(2:L+2)} \right\rangle$ for all i, j
Return: $\{(i,j)|\,\hat{p}_{i,j} \geq \gamma\}$ the set of predicted edges.

B. Properties of Holder-by-Parts Graphons

Here, we discuss properties of symmetric, piecewise-Holder graphons. This section is largely from (Davison & Austern, 2023), Appendix H. Refer to that text for a more complete exposition; we just present the details most relevant to our needs.

Let μ be the Lebesgue measure. We define a partition $\mathcal{Q}$ of $[0,1]$ to be a finite collection of pairwise disjoint, connected sets whose union is $[0,1]$, such that , for all $Q \in \mathcal{Q}$, $\mu(\text{int}(Q)) > 0$ and $\mu(\text{cl}(Q)\backslash\text{int}(Q)) = 0$. This induces a partition $\mathcal{Q}^{\otimes 2} = \mathcal{Q} \otimes \mathcal{Q}$ of $[0,1]^2$. We say that a graphon W lies in the Holder class $\text{Holder}([0,1]^2, \beta, M, \mathcal{Q}^{\otimes 2})$ if W is (β, M) Holder continuous in each $Q_i \otimes Q_j \in \mathcal{Q}^{\otimes 2}$. All graphons in question in this paper are assumed to belong to this class.

A graphon W can be viewed as an operator between L^p spaces. In this paper, we focus on the case of $p = 2$. In particular, for a fixed Graphon w, one can define the Hilbert-Schmidt operator

$$T_W[f](x) := \int_0^1 W(x,y)f(y)dy.$$

Since W is symmetric, T is self-adjoint. Furthermore, because $W(\cdot, \cdot) \leq 1$, T_W is a compact operator, as in (Stein & Shakarchi, 2009) page 190. Hence, the spectral theorem (for example, in (Fabian et al., 2013), Theorem 7.46) states that there exists a sequence of eigenvalues $\mu_i \to 0$ and eigenvectors ϕ_i (that form an orthonormal basis of $L^2([0,1])$, such that

$$T_W[f] = \sum_{n=1}^{\infty} \mu_n \langle f, \phi_n \rangle \phi_n, \quad W(x,y) = \sum_{n=1}^{\infty} \mu_n \phi_n(x)\phi_n(y), \tag{10}$$

and $\sum_{n=1}^{\infty} \mu_n^2 < \infty$. We note also that $|\mu_i| \leq 1$. This is because if μ_i is an eigenvalue, then

$$\int_0^1 W(x,y)\phi_i(y)\mathrm{d}y = \mu_i\phi_i(x) \Rightarrow \mu_i^2\phi_i(x)^2 = \left(\int_0^1 W(x,y)\phi_i(y)\mathrm{d}y \right)^2 \leq 1,$$

where the last inequality is true because W is bounded by 1 and $\phi_i(y)^2$ integrates to 1. Then, since $\phi_i(x)^2$ also integrates to 1, this shows the result.

B.1. Linear Relationship Between Moments and W (Proof of Proposition 4.2)

Proof of Proposition 4.2. Suppose that $m_W < \infty$ is the number of distinct nonzero eigenvalues of W, and label them by $|\mu_1| \geq |\mu_2| \geq \cdots \geq |\mu_{m_W}|$. Recall that

$$W(x, y) = \sum_{i=1}^{m_W} \mu_i \phi_i(x) \phi_i(y).$$

We first prove via induction that

$$W^{(k)}(x, y) = \sum_{i=1}^{m_W} \mu_i^k \phi_i(x) \phi_i(y).$$

Assume that this is true for $k \in \{1, 2, \ldots, K\}$. Now we show that $W^{(K+1)}(x, y) = \sum_{i=1}^{m_W} \mu_i^{K+1} \phi_i(x) \phi_i(y)$. Because the ϕ_i are orthonormal in $L^2([0, 1])$, we can compute

$$
\begin{aligned}
W^{(K+1)}(x, y) &= \int_0^1 W^{(K)}(x, t) W(t, y) \mathrm{d}t \\
&= \int_0^1 \left(\sum_{i=1}^{m_W} \mu_i^K \phi_i(x) \phi_i(t) \right) \cdot \left(\sum_{i=1}^{m_W} \mu_i \phi_i(y) \phi_i(t) \right) \mathrm{d}t \\
&= \int_0^1 \sum_{i,j}^{m_W} \mu_i^K \mu_j \phi_i(x) \phi_j(y) \phi_i(t) \phi_j(t) \mathrm{d}t \\
&= \sum_{i=1}^{m_W} \mu_i^{K+1} \phi_i(x) \phi_i(y),
\end{aligned}
$$

where the last equality is due to the orthonormality of the ϕ_i in $L^2([0, 1])$. This completes the induction. We now argue that there is a linear relationship between $W(x, y)$ and $(W^{(2)}(x, y), \ldots, W^{(m_W+1)}(x, y))$, i.e., there exists some β^{*, m_W} such that

$$W(x, y) = \sum_{i=1}^{m_W} \beta_i^{*, m_W} W^{(i+1)}(x, y)$$

for all x, y. In light of the above discussion, we can observe that the vector $\beta^{*, m_W} = \left(\beta_1^{*, m_W}, \beta_2^{*, m_W} \ldots, \beta_{m_W}^{*, m_W} \right)$ is simply the solution (if it exists) to the system of equations

$$
\begin{pmatrix}
\mu_1^2 & \mu_1^3 & \cdots & \mu_1^{m_W+1} \\
\mu_2^2 & \mu_2^3 & \cdots & \mu_2^{m_W+1} \\
\vdots & \vdots & \ddots & \vdots \\
\mu_{m_W}^2 & \mu_{m_W}^3 & \cdots & \mu_{m_W}^{m_W+1}
\end{pmatrix}
\begin{pmatrix}
\beta_1 \\
\beta_2 \\
\vdots \\
\beta_{m_W}
\end{pmatrix}
=
\begin{pmatrix}
\mu_1 \\
\mu_2 \\
\vdots \\
\mu_{m_W}
\end{pmatrix}
$$

To observe that a solution indeed exists, it suffices to observe that the matrix on the LHS is of full rank, i.e., has nonzero determinant. To see this, we note that the ith row is a multiple of $v_i := (1, \mu_i, \ldots, \mu_i^{m_W-1})$. We note that the matrix whose ith row is v_i is a Vandermonde matrix, which has nonzero determinant if all of the variables are distinct. Then, since multiplying each row by a constant changes the determinant only by a multiplicative factor, this suffices for the proof. $\square$

C. Proof of Proposition 5.1 and Proposition 5.2

We let A denote the adjacency matrix. In the proof, for random variables X and Borel sets B, we might write quantities of the form $\mathbb{P}(X \in B|A)$. This denotes a conditional probability, where we condition on the realization of the graph. A notation we also use is $X|A \sim \mathrm{Dist}$, which denotes the conditional distribution of a random variable X, conditioned on the realization of the graph. We first state the following Lemma, used in the proof of Proposition 5.1.

Lemma C.1. *Suppose that $G_n = ([n], E_n)$ is generated from the graphon $W_n(\cdot, \cdot) = \rho_n W(\cdot, \cdot)$. Write $W(\omega_i, \cdot) := \int_0^1 W(\omega_i, x) dx$. Then we have that*

$$\mathbb{P}\left(\sup_{i \leq n} \frac{1}{n-1} \big| |N(i)| - \rho_n W(\omega_i, \cdot) \big| \geq \rho_n t \right) \leq 2n \left(e^{-\frac{(n-1)\rho_n t^2}{3}} + e^{-2(n-1)t^2} \right) \tag{11}$$

Proof of Lemma C.1. We first show the above result for a fixed vertex i (without loss of generality, let $i = n$), and then conclude the proof through a union bound. We first state

Lemma C.2 ((Goemans, 2015), Theorem 4). *Let $X = \sum_{i=1}^{n} X_i$, where $X_i \sim Bern(p_i)$, and all the X_i are independent. Let $\mu = \mathbb{E}[X] = \sum_{i=1}^{n} d_n$. Then*

$$\mathbb{P}(|X - \mathbb{E}[X]| \geq \delta\mu) \leq 2\exp\left(-\mu\delta^2/3\right)$$

for all $\delta > 0$.

Now suppose that the latent feature ω_n is fixed. For any vertex $j \neq n$, we have

$$\mathbb{P}(a_{jn} = 1|\omega_n) = \int_0^1 W_n(\omega_n, x)dx \tag{12}$$

$$= \rho_n W(\omega_n, \cdot). \tag{13}$$

Recall that $|N(n)| = \sum_{j \neq n} a_{jn}$ hence $\mathbb{E}(|N(n)|\big|\omega_n) = \rho_n W(\omega_n, \cdot)$. We show that $|N(n)|$ concentrates around $\rho_n W(\omega_n, \cdot)$. In this goal, remark that $|N(n)|\big|(\omega_i)$ is distributed as a sum of independent Bernoulli random variables with probabilities $\rho_n W(\omega_i, \omega_n)$. Therefore, according to Lemma C.2, for all $t \in (0,1)$ we have

$$\mathbb{P}\left(\frac{1}{n-1}\Big||N(n)| - \mathbb{E}(|N(n)|\big|(\omega_i))\Big| \geq \rho_n t\right) \leq 2e^{-\frac{(n-1)\rho_n t^2}{3}}. \tag{14}$$

Moreover, we remark that conditionally on ω_n, the random variables $(W(\omega_n, \omega_j))_{j \neq n}$ are i.i.d. Therefore, according to Hoeffding's inequality, we remark that for all $t > 0$, we have

$$\mathbb{P}\left(\frac{1}{n-1}\Big|\mathbb{E}(|N(n)|\big|(\omega_i)) - \mathbb{E}(|N(n)|\big|\omega_n)\Big| \geq \rho_n t\right) \leq 2\exp\left(-2(n-1)t^2\right). \tag{15}$$

Using the union bound this directly implies that for all $t \in (0,1)$ we have

$$\mathbb{P}\left(\sup_{i \leq n} \frac{1}{n-1}\Big||N(i)| - \rho_n W(\omega_i, \cdot)\Big| \geq \rho_n t\right) \leq 2n\left(e^{-\frac{(n-1)\rho_n t^2}{3}} + e^{-2(n-1)t^2}\right) \tag{16}$$

$\square$

Proof of Proposition 5.1. The proof proceeds through induction. Let $A > 0$ be a constant such that $\sqrt{\frac{A\log(n)}{\rho_n(n-1)}} \leq \delta_W/2$. We denote the event

$$E := \left\{\sup_{i \in [n]} \Big||N(i)| - (n-1)\rho_n W(\omega_i, \cdot)\Big| \geq \sqrt{\rho_n(n-1)}\sqrt{A\log(n)}\right\}.$$

We remark that according to Lemma C.1 we have $\mathbb{P}(E^c) \leq 2n\left(e^{-\frac{A\log(n)}{3}} + e^{-\frac{2A\log(n)}{\rho_n}}\right)$. For the remainder of the proof we will work under the event E. Note that when E holds this also implies that

$$\inf_{i \in [n]} |N(i)| \geq \frac{1}{2}\rho_n \delta_W(n-1) \tag{17}$$

For ease of notation we define $\xi := \frac{2\sqrt{2}(s \wedge 1)M}{\sqrt{\delta_W}}$ and write

$$\epsilon(n,k) := \frac{\xi}{\sqrt{\rho_n(n-1)}}\left\{1 + \left((2M)^{k-1} - 1\right)\left(1 + \frac{2M^L(s \wedge 1)}{4\delta_W}\sqrt{1 + \frac{\sqrt{A\log(n)}}{\sqrt{\rho_n(n-1)}}}\sqrt{A\log(n)}\left(\frac{1 + \sqrt{\frac{A\log(n)}{(n-1)\rho_n}}}{\overline{W}}\right)^{L-2}\right)\right\}$$

We will then show that there is a constant $\kappa > 0$ so that, conditional on E holding, with a probability of at least $1 - 2ne^{-\kappa d_n}$ there exists embedding vectors (μ_i^k) that are independent from $\omega_{n-m+1:n}$ such that for every $k \leq L$ we have

$$\sup_{i \leq n} \|\lambda_i^k - \mu_i^k\|_2 \leq \epsilon(n,k).$$

To do so we proceed by induction. Firstly, since $\sigma(\cdot)$ is Lipschitz, we observe that for all $i \leq n$ that we have

$$\left\| \lambda_i^1 - \sigma\left(M_{1,0}\lambda_i^0\right) \right\|_2 \leq \left\| M_{1,1} \sum_{\ell \leq n} \frac{a_{i\ell}\lambda_\ell^{k-1}}{\sqrt{|N(i)\|N(\ell)|}} \right\|_2, \tag{18}$$

which we will show is bounded by $O\left(\sqrt{\sum_{\ell \in N(i)} \frac{1}{|N(i)\|N(\ell)|}}\right)$ with high probability. Using the hypothesis that $\|M_{1,1}\|_{\mathrm{op}} \overset{a.s}{\leq} M$, we note that

$$\left\| M_{1,1} \sum_{\ell \leq n} \frac{a_{i\ell}\lambda_\ell^{k-1}}{\sqrt{|N(i)\|N(\ell)|}} \right\|_2 \leq \|M_{1,1}\|_{op} \left\| \sum_{\ell \leq n} \frac{a_{i\ell}\lambda_\ell^{k-1}}{\sqrt{|N(i)\|N(\ell)|}} \right\|_2 \tag{19}$$

$$\leq M \left\| \sum_{\ell \leq n} \frac{a_{i\ell}\lambda_\ell^{k-1}}{\sqrt{|N(i)\|N(\ell)|}} \right\|_2. \tag{20}$$

To bound this last quantity, we note that conditioned on G_n, we have that $\sum_{\ell \leq n} \frac{a_{i\ell}\lambda_\ell^0}{\sqrt{|N(\ell)|}}$ is a $\sqrt{\sum_{\ell \in N(i)} \frac{s^2}{d_n|N(i)\|N(\ell)|}}$-sub-Gaussian vector with i.i.d entries. We will therefore use the following lemma

Lemma C.3. *Suppose that $X \in \mathbb{R}^{d_n}$ is a $\eta/\sqrt{d_n}$ sub-Gaussian vector with i.i.d coordinates. There exists some universal constant $\kappa > 0$ such that*

$$\mathbb{P}\left(\left| \|X\|_2 - \mathbb{E}(\|X\|_2) \geq t\right)\right| \leq 2\exp\left(-\frac{\kappa d_n t^2}{\eta^2}\right) \tag{21}$$

Proof of Lemma C.3. This is a direct consequence of Theorem 3.1.1 from (Vershynin, 2018).

$\square$

We remark that

$$\mathbb{E}\left(\left\| \sum_{\ell \leq n} \frac{a_{i\ell}\lambda_\ell^0}{\sqrt{|N(i)\|N(\ell)|}} \right\|_2 \Big| G_n\right) \leq \sqrt{\sum_{\ell \in N(i)} \frac{s^2}{|N(i)\|N(\ell)|}}.$$

Therefore we obtain that there exists a universal constant $\kappa > 0$ such that

$$\mathbb{P}\left(\left\| \sum_{\ell \leq n} \frac{a_{i\ell}\lambda_\ell^0}{\sqrt{|N(i)\|N(\ell)|}} \right\|_2 - \sqrt{\sum_{\ell \in N(i)} \frac{s^2}{|N(i)|N(\ell)|}} \geq t \Big| G_n\right) \leq 2\exp\left(-\kappa t^2 d_n \left(\sum_{\ell \in N(i)} \frac{s^2}{|N(i)\|N(\ell)|}\right)^{-1}\right), \tag{22}$$

and from this, setting $t = \sqrt{\sum_{\ell \in N(i)} \frac{s^2}{|N(i)\|N(\ell)|}}$, we can deduce that with probability at least $1 - 2\exp\left(-\kappa d_n\right)$,

$$\|\lambda_i^1 - \sigma\left(M_{1,0}\lambda_i^0\right)\|_2 \leq M \left\| \sum_{\ell \leq n} \frac{a_{i\ell}\lambda_\ell^0}{\sqrt{|N(i)\|N(\ell)|}} \right\|_2 \tag{23}$$

$$\leq 2(s \wedge 1)M \sqrt{\sum_{\ell \in N(i)} \frac{1}{|N(i)\|N(\ell)|}} \tag{24}$$

$$\overset{(a)}{\leq} \frac{2\sqrt{2}(s \wedge 1)M}{\sqrt{(n-1)\delta_W \rho_n}} \leq \frac{\xi}{\sqrt{(n-1)\rho_n}}. \tag{25}$$

where to get (a) we used the fact that under E we have that $\inf_{l \neq n}|N(l)| \geq \frac{\rho_n \delta_W(n-1)}{2}$.

We denote

$$\mu_i^1 := \sigma(M_{k,0}\lambda_i^0)$$

and remark that the random variables (μ_i^1) are independent from $(\omega_j)_{j=n-m+1}^n$, since $((\lambda_i^0)_i, M_{1,0}, M_{1,1}))$ are assumed to be independent from $(\omega_j)_{j=n-m+1:n}$. We remark in addition that for all i we have $\|\mu_i^1\|_2 \leq M\|\lambda_i^0\|_2$. We know that λ_i^0 is a $s/\sqrt{d_n}$ sub-Gaussian vector with i.i.d coordinates. Therefore by using Lemma C.3 again, we obtain that there exists $\tilde{k} > 0$ such that with probability of at least $1 - 2ne^{-2\tilde{\kappa}d_n}$ we have

$$\sup_{i \leq n} \|\mu_i^1\|_2 \leq 2M(s \wedge 1).$$

Denote the event

$$\tilde{E}_1 := \left\{ \sup_{i \in [n]} \|\lambda_i^1 - \mu_i^1\|_2 \leq \epsilon(n, 1) \ \& \ \sup_{i \leq n} \|\mu_i^1\|_2 \leq 2M(s \wedge 1) \right\}.$$

Taking a union bound over all vertices, we know that $\tilde{E}_1$ holds, conditionally on E holding, with a probability of at least $1 - 2n\exp(-\kappa d_n) - 2n\exp(-\tilde{\kappa}d_n)$. We now suppose that both $\tilde{E}_1$ and $\tilde{E}$ hold. Suppose that for some $1 < k < L$ the following event is true: for all $r \leq k$, there exists some set of vectors $(\mu_i^r)_{i \in [n]}$ independent of the latent features $(\omega_i)_{i=n-m+1}^n$ such that

$$\sup_{i \in [n]} \|\lambda_i^r - \mu_i^r\|_2 \leq \epsilon(n, r), \qquad \sup_{i \leq n} \|\mu_i^r\|_2 \leq 2(s \wedge 1)M^r \left(\frac{1 + \sqrt{\frac{A\log(n)}{(n-1)\rho_n}}}{\overline{W}} \right)^{r-1} \tag{26}$$

We will show that the same statement holds for $k + 1$. In this goal, we denote by $\tilde{E}_k$ the event

$$\tilde{E}_k := \left\{ \sup_{i \in [n]} \|\lambda_i^r - \mu_i^r\|_2 \leq \epsilon(n, r), \ \& \ \sup_{i \leq n} \|\mu_i^r\|_2 \leq 2(s \wedge 1)M^r \left(\frac{1 + \sqrt{\frac{A\log(n)}{(n-1)\rho_n}}}{\overline{W}} \right)^{r-1} \qquad \forall r \leq k \right\}.$$

For ease of notation, for each i, write $v_i^k = \lambda_i^k - \mu_i^k$. We write $\lambda_i^k = \mu_i^k + v_i^k$, where the norm of v_i^k is bounded, under the event $\tilde{E}^k$, by $\epsilon(n, k)$. Furthermore, we note that

$$\lambda_i^{k+1} = \sigma \left(M_{k+1,0}\lambda_i^k + M_{k+1,1} \sum_{\ell \leq n} \frac{a_{i\ell}\lambda_\ell^k}{\sqrt{|N(i)||N(\ell)|}} \right) \tag{27}$$

$$= \sigma \left(M_{k+1,0}\mu_i^k + M_{k+1,1} \sum_{\ell \leq n} \frac{a_{i\ell}\mu_\ell^k}{\sqrt{|N(i)||N(\ell)|}} + M_{k+1,0}v_i^k \right. \tag{28}$$

$$\left. + M_{k+1,1} \sum_{\ell \leq n} \frac{a_{i\ell}v_\ell^k}{\sqrt{|N(i)||N(\ell)|}} \right) \tag{29}$$

Under the event $\tilde{E}^k$ we have

$$\left\| M_{k+1,0}v_i^k + M_{k+1,1} \sum_{\ell \leq n} \frac{a_{i\ell}v_\ell^k}{\sqrt{|N(i)||N(\ell)|}} \right\|_2 \leq 2M\epsilon(n, k).$$

As $\sigma(\cdot)$ is Lipschitz, this implies that

$$\sup_{i \leq n} \left\| \lambda_i^{k+1} - \sigma \left(M_{k+1,0}\mu_i^k + M_{k+1,1} \sum_{\ell \leq n} \frac{a_{i\ell}\mu_\ell^k}{\sqrt{|N(i)||N(\ell)|}} \right) \right\|_2 \tag{30}$$

$$\leq 2M\epsilon(n, k). \tag{31}$$

Moreover we also remark that as E and $\tilde{E}^k$ holds we have

$$\sup_{i \leq n} \left\| \lambda_i^{k+1} - \sigma\Big(M_{k+1,0}\mu_i^k + M_{k+1,1}\sum_{\ell \leq n} \frac{a_{i\ell}\mu_\ell^k}{(n-1)\rho_n\sqrt{W(\omega_i,\cdot)W(\omega_\ell,\cdot)}}\Big)\right\|_2 \tag{32}$$

$$\leq M \sup_{i \leq n} \|\mu_i^k\|_2 \sum_{\ell \leq n} a_{i\ell}\left| \frac{1}{\sqrt{|N(i)\|N(\ell)|}} - \frac{1}{(n-1)\rho_n\sqrt{W(\omega_i,\cdot)W(\omega_\ell,\cdot)}}\right| \tag{33}$$

$$\leq \Big(\frac{1+\sqrt{\frac{A\log(n)}{(n-1)\rho_n}}}{\overline{W}}\Big)^{k-1} \frac{\sqrt{2}(s \wedge 1)M^{k+1}}{\sqrt{\delta_W}\delta_W}\sqrt{1+\frac{\sqrt{A\log(n)}}{\sqrt{\rho_n(n-1)}}}\frac{\sqrt{A\log(n)}}{\sqrt{\rho_n(n-1)}} \tag{34}$$

$$\leq \Big(\frac{1+\sqrt{\frac{A\log(n)}{(n-1)\rho_n}}}{\overline{W}}\Big)^{L-2} \frac{\sqrt{2}(s \wedge 1)M^L}{\sqrt{\delta_W}\delta_W}\sqrt{1+\frac{\sqrt{A\log(n)}}{\sqrt{\rho_n(n-1)}}}\frac{\sqrt{A\log(n)}}{\sqrt{\rho_n(n-1)}}. \tag{35}$$

Note however that we have assumed that $W(x,\cdot) = \overline{W}$ is a constant function. This therefore implies that $\sigma\Big(M_{k+1,0}\mu_i^k + M_{k+1,1}\sum_{\ell \leq n} \frac{a_{i\ell}\mu_\ell^k}{(n-1)\rho_n\overline{W}}\Big)$ is independent from $\omega_{n-m+1,n}$. Defining

$$\mu_i^{k+1} := \sigma\Big(M_{k+1,0}\mu_i^k + M_{k+1,1}\sum_{\ell \leq n} \frac{a_{i\ell}\mu_\ell^k}{(n-1)\rho_n\overline{W}}\Big), \tag{36}$$

we have that

$$\sup_{i \in [n]} \|\lambda_i^{k+1} - \mu_i^{k+1}\|_2 \leq \epsilon(n, k+1). \tag{37}$$

Moreover we note that

$$\|\mu_i^{k+1}\|_2 \leq 2M \sup_{i \leq n} \|\mu_i^k\|_2\Big(1 + \sum_{\ell \leq n}\frac{a_{i\ell}\mu_\ell^k}{(n-1)\rho_n\overline{W}}\Big) \tag{38}$$

$$\leq 2M \sup_{i \leq n} \|\mu_i^k\|_2\Big(1 + \frac{|N(i)|}{(n-1)\rho_n\overline{W}}\Big) \tag{39}$$

$$\overset{(a)}{\leq} 2M \sup_{i \leq n} \|\mu_i^k\|_2 \frac{1 + \sqrt{A\log(n)/((n-1)\rho_n)}}{2\overline{W}} \tag{40}$$

where to get (a) we used the fact that we assumed that $\tilde{E}$ holds. Hence we obtain that

$$\sup_{i \leq n} \|\mu_i^{k+1}\|_2 \leq 2M(s \wedge 1)\Big(\frac{2M + 2M\sqrt{A\log(n)/((n-1)\rho_n)}}{2\overline{W}}\Big)^k.$$

Hence if $\tilde{E}^1$ and $\tilde{E}$ hold this implies that $\tilde{E}^{k+1}$ and $\tilde{E}$ hold which completes the induction. We hence have that

$$P\Big(\sup_{i \leq n}\sup_{i \in [n]} \|\lambda_i^r - \mu_i^r\|_2 \leq \epsilon(n, r),\ \forall r \leq L\Big) \geq 1 - 2n\big(e^{-\frac{A\log(n)}{3}} + e^{-\frac{2A\log(n)}{\rho_n}} + e^{-\kappa d_n} + e^{-\tilde{\kappa}d_n}\big)$$

Choosing $A = 6$ yields the desired result. $\qquad\square$

We then prove Proposition 5.2

Proof. Suppose that $f : \mathbb{R}^2 \to \mathbb{R}$ is Lipchtiz with respect to the Euclidean distance in $\mathbb{R}^2$. Using Proposition 5.1, we know that there exists $\kappa > 0$ and embeddings (μ_j^L) that are independent from $\omega_{n-m+1:n}$ such that with a probability of at least $1 - \frac{2}{n} - \frac{2}{n^{11}} - 2ne^{-\kappa d_n}$ we have,

$$\sup_{\ell \leq L} \|\lambda_n^L - \mu_n^\ell\|_2 \leq \frac{K}{\sqrt{n}}, \tag{41}$$

18

where $K > 0$ is an absolute constant. As f is assumed to be a Lipchitz function we obtain that

$$\left| f(\lambda_n^L, \lambda_i^L) - f(\mu_n^L, \mu_i^L) \right| \leq \frac{2K}{\sqrt{n}}.$$

Now denote the event $E_n := \{ f(\lambda_i^L, \lambda_n^L) \geq 2 \}$ and define $\tilde{E}_n := \{ |f(\lambda_n^L, \lambda_i^L) - f(\mu_n^L, \mu_i^L)| \leq \frac{2K}{\sqrt{n}} \}$. We will obtain two different bounds respectively when

- $P(E_n) \geq \frac{1}{3} \mathbb{E}\left(\left[W(\omega_i, \omega_n) - W(\omega_i, \cdot) \right]^2 \right)$

- $P(E_n) < \frac{1}{3} \mathbb{E}\left(\left[W(\omega_i, \omega_n) - W(\omega_i, \cdot) \right]^2 \right).$

Firstly, if $P(E_n) \geq \frac{1}{3} \mathbb{E}\left(\left[W(\omega_i, \omega_n) - W(\omega_i, \cdot) \right]^2 \right)$ we remark that

$$\mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\lambda_i^L, \lambda_n^L) \right]^2 \right) \geq \mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\lambda_i^L, \lambda_n^L) \right]^2 \mathbb{I}(E_n) \right) \tag{42}$$

$$\geq P(E_n) > \frac{1}{3} \mathbb{E}\left(\left[W(\omega_i, \omega_n) - W(\omega_i, \cdot) \right]^2 \right). \tag{43}$$

Now assume instead $P(E_n) < \frac{1}{3} \mathbb{E}\left(\left[W(\omega_i, \omega_n) - W(\omega_i, \cdot) \right]^2 \right)$. This implies that

$$\mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\lambda_i^L, \lambda_n^L) \right]^2 \right) - \mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\mu_i^L, \mu_n^L) \right]^2 \mathbb{I}(E_n^c \cap \tilde{E}_n^c) \right) \tag{44}$$

$$= \mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\lambda_i^L, \lambda_n^L) \right]^2 \mathbb{I}(E_n) \right) + \mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\lambda_i^L, \lambda_n^L) \right]^2 \mathbb{I}(E_n^c \cap \tilde{E}_n) \right) \tag{45}$$

$$+ \mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\lambda_i^L, \lambda_n^L) \right]^2 \mathbb{I}(E_n^c \cap \tilde{E}_n^c) \right) - \mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\mu_i^L, \mu_n^L) \right]^2 \mathbb{I}(E_n^c \cap \tilde{E}_n^c) \right) \tag{46}$$

$$\geq -\left| \mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\lambda_i^L, \lambda_n^L) \right] \left[f(\mu_i^L, \mu_n^L) - f(\lambda_i^L, \lambda_n^L) \right] \mathbb{I}(E_n^c \cap \tilde{E}_n^c) \right) \right| \tag{47}$$

$$- \left| \mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\mu_i^L, \mu_n^L) \right] \left[f(\mu_i^L, \mu_n^L) - f(\lambda_i^L, \lambda_n^L) \right] (E_n^c \cap \tilde{E}_n^c) \right) \right| \tag{48}$$

$$\overset{(a)}{\geq} -\frac{2K}{\sqrt{n}} \left(2 + 2 + \frac{2K}{\sqrt{n}} \right) P(E_n^C \cap \tilde{E}_n^c) \tag{49}$$

where to get (a) we used the fact that under $E_n^c \cap \tilde{E}_n^c$ we have

$$|W(\omega_i, \omega_n) - f(\mu_i^L, \mu_n^L)| \leq 2 + \frac{2K}{\sqrt{n}}$$

and

$$|W(\omega_i, \omega_n) - f(\lambda_i^L, \lambda_n^L)| \leq 2.$$

Hence we obtain that

$$\mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\lambda_i^L, \lambda_n^L) \right]^2 \right) \geq \mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\mu_i^L, \mu_n^L) \right]^2 \mathbb{I}(E_n^c \cap \tilde{E}_n^c) \right) + o_n(1).$$

However as (μ_j^L) are independent from ω_n we have that $f(\mu_i^L, \mu_n^L)$ is independent from ω_n. Hence if we write $W(x, \cdot) = \int_0^1 W(x, y) dy$ we obtain that

$$\mathbb{E}\left(\left[W(\omega_i, \omega_n) - f(\mu_i^L, \mu_n^L) \right]^2 \mathbb{I}(E_n^c \cap \tilde{E}_n^c) \right) \tag{50}$$

$$= \mathbb{E}\left(\left[W(\omega_i, \omega_n) - W(\omega_i, \cdot) \right]^2 \mathbb{I}(E_n^c \cap \tilde{E}_n^c) \right) + \mathbb{E}\left(\left[f(\mu_i^L, \mu_n^L) - W(\omega_i, \cdot) \right]^2 \mathbb{I}(E_n^c \cap \tilde{E}_n^c) \right) \tag{51}$$

$$\geq \mathbb{E}\left(\left[W(\omega_i, \omega_n) - W(\omega_i, \cdot) \right]^2 \mathbb{I}(E_n^c \cap \tilde{E}_n^c) \right) \tag{52}$$

$$\geq -P(E_n) - P(\tilde{E}_n) + \mathbb{E}\left(\left[W(\omega_i, \omega_n) - W(\omega_i, \cdot) \right]^2 \right). \tag{53}$$

Now we have assumed that $P(E) \to 0$ and we know that $P(\tilde{E}) \to 0$. Hence we obtain that

$$\mathbb{E}\Big(\big[W(\omega_i, \omega_n) - f(\lambda_i^L, \lambda_n^L) \big]^2 \Big) \geq \frac{2}{3} \mathbb{E}\Big(\big[W(\omega_i, \omega_n) - W(\omega_i, \cdot) \big]^2 \Big) + o_n(1).$$

Now we have assumed that $W(\cdot, \cdot)$ is not the constant graphon but H_3 assumes that $x \to W(x, \cdot)$ is a constant function. Hence by choosing

$$K := \frac{1}{3} \mathbb{E}\Big(\big[W(\omega_i, \omega_n) - W(\omega_i, \cdot) \big]^2 \Big) > 0$$

we obtain that

$$\mathbb{E}\Big(\big[W(\omega_i, \omega_n) - f(\lambda_i^L, \lambda_n^L) \big]^2 \Big) \geq K + o_n(1).$$

$\square$

D. Proof of Proposition 4.1

We proceed in two main steps. The first step is to establish a high-probability bound for $\big| \hat{W}_{n,i,j}^{(k)} - W_{n,i,j}^{(k)} \big|$. This bound is then used to establish a bound on $|\hat{q}_{i,j}^{(k)} - W_{n,i,j}^{(k)}|$. The main goal is to prove Proposition D.8, which is a restatement of Proposition 4.1.

We will do these steps separately in the below subsections.

D.1. Proof of Proposition 4.1, Part 1

The goal of this subsection is to prove the following lemma.

Lemma D.1. *With probability at least $1 - 3/n$, we have that for all $2 \leq k \leq L + 2$,*

$$\max_{i \neq j} \big| \hat{W}_{n,i,j}^{(k)} - W_{n,i,j}^{(k)} \big| \leq 3a_k \frac{\rho_n^{k-1/2}}{\sqrt{n-1}} \log(n)^k,$$

where $a_k = C\sqrt{2}(8(k+2))^k k^{k+1} \sqrt{k!}/\sqrt{B}$, where B, C are some absolute positive constants.

We proceed in three steps. We first establish a high probability bound for $\big| \hat{W}_{n,i,j}^{(k)} - \mathbb{E}[\hat{W}_{n,i,j}^{(k)}|(\omega_\ell)] \big|$. Then, we establish a high probability bound for $\big| \mathbb{E}[\hat{W}_{n,i,j}^{(k)}|(\omega_\ell)] - \mathbb{E}[\hat{W}_{n,i,j}^{(k)}|\omega_i, \omega_j] \big|$. We then bound $\big| \mathbb{E}[\hat{W}_{n,i,j}^{(k)}|\omega_i, \omega_j] - W_{n,i,j}^{(k)} \big|$.

D.1.1. BOUNDING $\big| \hat{W}_{n,i,j}^{(k)} - \mathbb{E}[\hat{W}_{n,i,j}^{(k)}|(\omega_\ell)] \big|$

We use the following

Lemma D.2 ((Kim & Vu)). *Let (ξ_i) be a sequence of independent Bernouilli random variables. Let N be an integer and $P : \mathbb{R}^N \to \mathbb{R}$ be a polynomial of degree k. For a subset $A \subset [|N|]^k$ we write by $\partial_A P$ the partial derivative of P with respect to the indexes A. Define $\mu_1 = \max_{|A| \geq 1} \mathbb{E}[\partial_A P((\xi_i)_{i \leq N})]$ and let $\mu_0 = \max_{|A| \geq 0} \mathbb{E}[\partial_A P((\xi_i)_{i \leq N})]$. Then,*

$$\mathbb{P}\big(|P((\xi_i)_{i \leq N}) - \mathbb{E}[P((\xi_i)_{i \leq N})|(\omega_l)]| > a_k \sqrt{\mu_0 \mu_1} \lambda^k \big) \leq G \cdot \exp(-\lambda + (k-1)\log(N)), \tag{54}$$

where $a_k = 8^k \sqrt{k!}$, and G is an absolute constant.

We will apply Lemma D.2 to obtain the desired result. We first fix $i \neq j$. In this goal we set $N := n(n-1)/2$ and define P to be the following polynomial:

$$P((a_{k,l})_{k \neq l \leq n}) := \hat{W}_{n,i,j}^{(k)} = \frac{1}{(n-1)^{k-1}} \sum_{r_1, \dots, r_k} a_{i,r_i} \dots a_{r_{k-1}, j}.$$

We remark that conditionally on the features (ω_l), the random variables $(a_{k,l})$ are independent Bernouili random variables. We note that our goal is to give a high probability bound on the difference between $P((a_{k,l})_{k \neq l \leq n})$ and its expectation $\mathbb{E}[P((a_{k,l})_{k \neq l \leq n})|(\omega_l)]$. We first bound $\mathbb{E}[\partial_A P((a_{k,l})_{k \neq l \leq n})]$. We note that this is maximized when A contains only

one element. This is because when differentiating by $a_{i,j}$, all of the terms that do not include this edge vanish, hence differentiating by more $a_{i,j}$ will cause more edges to vanish.

Furthermore, $\mathbb{E}[(\partial/\partial a_{s,t})P((a_{k,l})_{k\neq l\leq n})]$ is maximized by choosing (s,t) to be an edge that appears most often, such that as many terms as possible are preserved. Because the endpoints are fixed as i, j, it suffices to bound the desired quantity for $(s,t) = (i,1)$ (without loss of generality, assume $i \neq 1$; note the choice of 1 was arbitrary). For each string $a_{i,r_1}a_{r_1,r_2}\ldots a_{r_{k-1},j}$, if it contains $a_{i,1}$, then the number of terms in the string will be lowered by 1 upon differentiation (otherwise it equals 0 identically), hence after differentiating, the maximum number of terms in the string is $k-1$.

We now upper-bound the number of strings $a_{i,r_1}a_{r_1,r_2}\ldots a_{r_{k-1},j}$ that contain a_{i1} and also have exactly t distinct edges.

1. **Case 1:** $r_1 = 1$. Then, there are $k-2$ free indices remaining. However, since there are t distinct edges, that means $k-t$ edges are repeated (appear at least more than once). Note that each repeated edge removes one free index. Hence, the remaining number of degrees of freedom is $t-2 \vee 0$.

2. **Case 2:** $r_1 \neq 1$. Then, since the edge $(i,1)$ needs to appear in the sequence, there are at most k locations for it to appear, and then 2 ways to orient it (it can either be $(i,1)$ or $(1,i)$). So, there are $2k$ ways to choose the edge $(i,1)$, and then there remain $t-3 \vee 0$ ways degrees of freedom remaining.

Combining the two cases, there are at most $(n-1)^{t-2\vee 0} + 2k(n-1)^{t-3\vee 0} \leq 2(n-1)^{t-2\vee 0}$ ways to choose the set of indices $\{r_1, r_2, \ldots, r_{k-1}\}$. Then, there are at most $(k-1)^{k-1}$ ways to choose the values of $r_1, r_2, \ldots, r_{k-1}$ among this set, which is upper bounded by k^k. Hence, the number of configurations with exactly t distinct edges is upper bounded by $2k^k(n-1)^{t-2\vee 0}$. Hence, we can bound

$$\mathbb{E}[(\partial/\partial a_{s,t})P((a_{k,l})_{k\neq l\leq n})|(\omega_l)] \leq \frac{2k^k}{(n-1)^{k-1}}\sum_{t=1}^{k}(n-1)^{t-2\vee 0}\rho_n^{t-1}$$

$$\leq 2k^k\sum_{t=1}^{k}\frac{\rho_n^{t-1}}{(n-1)^{k-t+1}}$$

$$\leq 2k^{k+1}\frac{\rho_n^{k-1}}{n-1},$$

where the last inequality follows if $\rho_n > \frac{1}{n-1}$. We now bound $\mathbb{E}[P((a_{k,l})_{k\neq l\leq n})|(\omega_l)]$. We first upper-bound the number of paths from i to j of length k with exactly ℓ distinct edges. For convenience, denote $r_0 = i$ and $r_k = j$. Firstly, we note that if there are exactly ℓ distinct edges, then $|\{r_0, r_1, r_2, \ldots, r_{k-1}, r_k\}| \leq \ell+1$. Since $r_0 = i, r_k = j$, there are at most $\binom{n-1}{\ell-1}$ ways to choose a superset in which $\{r_1, r_2, \ldots, r_{k-1}\}$ lies. Then, there are at most $(\ell-1)^{k-1} \leq k^k$ ways to choose the indices $r_1, r_2, \ldots, r_{k-1}$ among this set. Hence, there are most $(n-1)^{\ell-1}k^k$ paths of length k with exactly ℓ distinct edges from i to j. Hence,

$$\mathbb{E}[P((a_{k,l})_{k\neq l\leq n})|(\omega_l)] \leq \frac{1}{(n-1)^{k-1}}\sum_{\ell=1}^{k}(n-1)^{\ell-1}k^k \cdot \rho_n^{\ell}$$

$$\leq k^k\sum_{\ell=1}^{k}\frac{\rho_n^{\ell}}{(n-1)^{k-\ell}}$$

$$\leq k^{k+1}\rho_n^k.$$

Now we apply Lemma D.2 to obtain

$$\mathbb{P}\left(|P((\xi_i)_{i\leq N}) - \mathbb{E}[P((\xi_i)_{i\leq N})|(\omega_l)]| > b_k\frac{\rho_n^{k-1/2}}{\sqrt{n-1}}\lambda^k\right) = G \cdot \exp(-\lambda + (\mathrm{k}-1)\log(\mathrm{N}))$$

$$\leq G \cdot \exp(-\lambda + (\mathrm{k}-1)\log(\mathrm{n}))$$

for some absolute constant G, where $b_k = \sqrt{2}8^k k^{k+1}\sqrt{k!}$. Choosing $\lambda = \log(G) + (k+2)\log(n)$, and union bounding

over all $i \neq j$ and $2 \leq k \leq L + 2$, we have that with probability at least $1 - 1/n$, for all $2 \leq k \leq L + 2$,

$$\max_{i \neq j} \left| \hat{W}_{n,i,j}^{(k)} - \mathbb{E}[\hat{W}_{n,i,j}^{(k)} | (\omega_l)] \right| \leq a_k \frac{\rho_n^{k-1/2}}{\sqrt{n-1}} \log(n)^k.$$

where $a_k = C\sqrt{2}(8(k+2))^k k^{k+1} \sqrt{k!}/\sqrt{B}$, where B is from the constant in the Big O factor, and C is some constant.

D.1.2. STEP 2: BOUNDING $\left| \mathbb{E}[\hat{W}_{n,i,j}^{(k)} | (\omega_\ell)] - \mathbb{E}[\hat{W}_{n,i,j}^{(k)} | \omega_i, \omega_j] \right|$

We now bound $\left| \mathbb{E}[\hat{W}_{n,i,j}^{(k)} | (\omega_\ell)] - \mathbb{E}[\hat{W}_{n,i,j}^{(k)} | \omega_i, \omega_j] \right|$ using McDiarmid's Inequality. For ease of notation, we assume WLOG that $i = 1$ and $j = 2$, and denote $r_0 = i$ and $r_k = j$. To use McDiarmid's inequality; we first bound the maximum deviation in altering one of the coordinates. WLOG we alter the nth coordinate ω_n and bound

$$\left| \mathbb{E}[\hat{W}_{n,1,2}^{(k)} | (\omega_\ell)_{\ell \neq n}, \omega_n] - \mathbb{E}[\hat{W}_{n,1,2}^{(k)} | (\omega_\ell)_{\ell \neq n}, \omega_n'] \right|.$$

Recalling the definition

$$\hat{W}_{n,1,2}^{(k)} = \frac{1}{(n-1)^{k-1}} \sum_{r_1, r_2, \ldots, r_{k-1}} a_{1,r_1} a_{r_1, r_2} \ldots a_{r_{k-1}, 2},$$

denote $B_{(r_s)} = \mathbb{E}[a_{1,r_1} a_{r_1, r_2} \ldots a_{r_{k-1}, 2} | (\omega_\ell)_{\ell \neq n}, \omega_n] - \mathbb{E}[a_{1,r_1} a_{r_1, r_2} \ldots a_{r_{k-1}, 2} | (\omega_\ell)_{\ell \neq n}, \omega_n']$. We first bound each $B_{(r_s)}$ individually over different choices of the indices (r_s). We note that if none of the $r_s = n$, then $B_{(r_s)} = 0$. Hence, we need consider only the terms in the summation for which at least one of the r_s equals n.

If (r_s) corresponds to a path with exactly $k - t$ distinct edges, then $|B_{(r_s)}| \leq \rho_n^{k-t}$. We upper bound the number of paths of length k that have exactly $k - t$ distinct edges. Note that $t \leq k - 2$, since our we are considering terms such that there exist r_s that equal $1, 2, n$, so there cannot be less than two distinct edges. We note that if there are exactly $k - t$ distinct edges, then the number of distinct numbers among the set $\{r_0, r_1, \ldots, r_k\}$ is at most $k + 1 - t$. Because $r_0 = 1$ and $r_k = 2$, and n must be one of the r_s must equal n, there are at most $\binom{n-1}{k-2-t}$ ways to choose the remaining vertices. Then, the number of ways to choose the values of r_s among these $k + 1 - t$ options is bounded by $(k + 1 - t)^{k-1}$. Hence, the total number of options is upper bounded by $\binom{n-1}{k-t-2}(k + 1 - t)^{k-1} \leq (n-1)^{k-t-2}(k+1)^{k-1}$. Lastly, we note that $t \in \{0, 1, \ldots, k - 1\}$. Hence, the constant in the exponential bound of McDiarmid's Inequality is given by

$$
\begin{aligned}
\frac{1}{(n-1)^{k-1}} \sum_{t=0}^{k-2} \rho_n^{2(k-t)}(n-1)^{k-t-2}(k+1)^{k-1} &= \frac{\rho_n^{2k}(k+1)^{k-1}}{n-1} \sum_{t=0}^{k-2} \left(\frac{1}{n\rho_n^2}\right)^t \\
&= \frac{\rho_n^{2k}(k+1)^{k-1}}{n-1} \frac{1 - (1/n\rho_n^2)^{k-1}}{1 - (1/n\rho_n^2)} \\
&\leq \frac{\rho_n^{2k}(k+1)^{k-1}}{n-1} \frac{1 - (1/n\rho_n^2)^{k-1}}{1 - (1/n\rho_n^2)} \\
&\leq 4 \frac{\rho_n^{2k} k^k}{n}
\end{aligned}
\tag{55}
$$

if $n\rho_n^2 \geq \frac{1}{10}$, since then $\frac{1-(1/n\rho_n^2)^{k-1}}{1-(1/n\rho_n^2)} \leq \frac{10}{9}$, and $(k+1)^{k-1} \leq 2k^k$ for all $k \geq 2$. Then, the McDiarmid Inequality states that

$$\mathbb{P}\left(|\mathbb{E}[\hat{W}_{n,i,j}^{(k)} | (\omega_\ell)] - \mathbb{E}[\hat{W}_{n,i,j}^{(k)} | \omega_i, \omega_j]| \geq t \right) \leq 2\exp\left(-2t^2 \frac{n}{\rho_n^{2k} k^k} \right) \tag{56}$$

Hence, choosing $t = \frac{\sqrt{k^k \rho_n^{2k}}}{\sqrt{n}} \sqrt{2\log(n)}$ and union bounding over $i \neq j$, $2 \leq k \leq L + 2$, we have that with probability at least $1 - 2/n$, for all $2 \leq k \leq L + 2$,

$$\max_{i \neq j} |\mathbb{E}[\hat{W}_{n,i,j}^{(k)} | (\omega_\ell)] - \mathbb{E}[\hat{W}_{n,i,j}^{(k)} | \omega_i, \omega_j]| \geq \sqrt{\frac{k^k \rho_n^{2k}}{n}} \sqrt{2\log(n)} \tag{57}$$

D.1.3. STEP 3: BOUNDING $\big|\mathbb{E}[\hat{W}_{n,i,j}^{(k)}|\omega_i, \omega_j] - W_{n,i,j}^{(k)}\big|$

Recall

$$\hat{W}_{n,i,j}^{(k)} = \frac{1}{(n-1)^{k-1}} \sum_{r_1,\ldots,r_{k-1}} a_{ir_1} a_{r_1 r_2} \ldots a_{r_{k-1}j}$$

We see that

$$\mathbb{E}[\hat{W}_{n,i,j}^{(k)}|\omega_i, \omega_j] = \frac{1}{(n-1)^k} \sum_{\ell=1}^{k} W_{n,i,j}^{(\ell)} \cdot (\# \text{ paths with } \ell \text{ distinct edges})$$

Firstly, we claim that the number of paths of length k starting from vertex i to j that have no repeated edges is lower bounded by $(n-2)(n-3)\ldots(n-k)$. This is simply because if no vertex is passed through twice along the path, then there cannot exist repeated edges. There are $n-2$ choices for r_1, then $n-3$ choices for r_2, etc., which shows this assertion. This implies that the number of paths with k distinct edges is $(n-1)^{k-1} + P_k$, where $|P_k| = O(k^2 n^{k-2})$. Note that this also implies that the number of paths of length k is of order $O(k^2 n^{k-2})$. Hence, we can write

$$\mathbb{E}[\hat{W}_{n,i,j}^{(k)}|\omega_i, \omega_j] = W_{n,i,j}^{(k)} + \frac{1}{(n-1)^{k-1}} P_k \cdot W_{n,i,j}^{(k)} + \frac{1}{(n-1)^{k-1}} \sum_{\ell=2}^{k-1} W_{n,i,j}^{(\ell)} (\# \text{ paths with } \ell \text{ distinct edges})$$

$$\Rightarrow |\mathbb{E}[\hat{W}_{n,i,j}^{(k)}|\omega_i, \omega_j] - W_{n,i,j}^{(k)}| \le \frac{1}{(n-1)^{k-1}} |P_k| \cdot W_{n,i,j}^{(k)} + \left| \frac{1}{(n-1)^{k-1}} \sum_{\ell=2}^{k-1} W_{n,i,j}^{(\ell)} (\# \text{ paths with } \ell \text{ distinct edges}) \right|$$

To proceed with the triangle inequality, we first upper-bound the number of paths from i to j of length k with exactly ℓ distinct edges. For convenience, denote $r_0 = i$ and $r_k = j$. Firstly, we note that if there are exactly ℓ distinct edges, then $|\{r_0, r_1, r_2, \ldots, r_{k-1}, r_k\}| \le \ell + 1$. Since $r_0 = i, r_k = j$, there are at most $\binom{n-1}{\ell-1}$ ways to choose a superset in which $\{r_1, r_2, \ldots, r_{k-1}\}$ lies. Then, there are at most $(\ell-1)^{k-1} \le k^k$ ways to choose the indices $r_1, r_2, \ldots, r_{k-1}$ among this set. Hence, there are most $(n-1)^{\ell-1} k^k$ paths of length k with exactly ℓ distinct edges from i to j. Hence,

$$|\mathbb{E}[\hat{W}_{n,i,j}^{(k)}|\omega_i, \omega_j] - W_{n,i,j}^{(k)}| \le O\left(\frac{k^2}{n}\right) \rho_n^k + \sum_{\ell=2}^{k-1} \frac{1}{(n-1)^{k-\ell}} \rho_n^\ell k^k$$

$$= k^k O\left(\frac{\rho_n^{k-1}}{n} + \frac{\rho_n^{k-2}}{n^2} + \cdots + \frac{\rho_n}{n^{k-1}}\right)$$

$$= O\left(k^{k+1} \frac{\rho_n^{k-1}}{n}\right),$$

where this last line is true because $\rho_n > \frac{1}{n}$.

D.1.4. STEP 4: COMBINING THE BOUNDS

Combining the three steps and using the triangle inequality, we have that with probability at least $1 - 3/n$, we have that for all $2 \le k \le L + 2$,

$$\max_{i \ne j} |\hat{W}_{n,i,j}^{(k)} - W_{n,i,j}^{(k)}| \le \sqrt{\frac{k^k \rho_n^{2k}}{n}} \sqrt{2\log(n)} + a_k \frac{\rho_n^{k-1/2}}{\sqrt{n-1}} \log(n)^k + O\left(k^{k+1} \frac{\rho_n^{k-1}}{n}\right)$$

$$\le 3a_k \frac{\rho_n^{k-1/2}}{\sqrt{n-1}} \log(n)^k,$$

for sufficiently large n, as we note that the second term is the dominating one when $\rho_n > 1/n$. This suffices for the proof of Lemma D.1.

D.2. Proof of Proposition 4.1, Part 2

The main goal of this subsection is to prove Proposition D.8. Before proving that, we first present general properties of the GNN embedding vectors our proposed algorithm produces (where we consider a more general version of our proposed GNN in which the weight matrices are not the identity). Uninterested readers can skip directly to Proposition D.8 to see the main result, and those who interested in more details can continue to read the exposition below.

In this appendix, we consider a version of our proposed GNN architecture with general weight matrices, given by

$$\lambda_i^k = M_{k,0}\lambda_i^{k-1} + M_{k,1}\frac{1}{n-1}\sum_{\ell \leq n} a_{i\ell}\lambda_\ell^{k-1}, \tag{58}$$

where $M_{k,0}, M_{k,1}$ are matrices that can be freely chosen. Note also that $a_{ii} = 0$, and hence the normalization by $n-1$. As proposed in Algorithm 1, we initialize the embeddings by first sampling $(Z_i) \overset{iid}{\sim} \frac{1}{\sqrt{d_n}}\mathcal{N}(0, I_{d_n})$, and then computing the first layer through

$$\lambda_i^0 = \frac{1}{\sqrt{n-1}}\sum_{\ell=1}^n a_{i\ell}Z_\ell. \tag{59}$$

We compute a total of L GNN iterations and for all vertices i, produce the sequence $\lambda_i^0, \lambda_i^1, \ldots, \lambda_i^L$.

In this appendix, we prove Proposition D.8 in a series of steps:

1. We first give a general formula for λ_i^k, and then demonstrate that $\mathbb{E}[\langle \lambda_i^{k_1}, \lambda_j^{k_2} \rangle]$ is a linear combination of the empirical moments of the graphon $\hat{W}_{n,i,j}^{(k)}$. This is done in Lemma D.4.

2. We then show in Lemma D.5 that $\hat{q}_{i,j}^{(k)}$ can be written in the simpler form

$$\hat{q}_{i,j}^{(k)} = \left\langle \frac{1}{\sqrt{n-1}}\sum_{\ell \leq n} a_{j,\ell}Z_\ell, \frac{1}{\sqrt{n-1}}\sum_{\ell \leq n} \hat{W}_{n,i,\ell}^{(k-1)}Z_\ell \right\rangle.$$

3. We then use the above observation to establish a concentration result for $\hat{q}_{i,j}^{(k)}$ in Proposition D.8.

D.3. Formula for the Embedding Vectors

Recall the definition from Equation (9)

$$N_s^k := \sum_{\substack{r_1,\ldots,r_k \in \{0,1\} \\ \sum_{i=1}^k r_i = s}} M_{k,r_1}M_{k-1,r_2}\ldots M_{1,r_k}. \tag{60}$$

For example, $N_0^3 = M_{3,0}M_{2,0}M_{1,0}$ and $N_1^3 = M_{3,0}M_{2,1}M_{1,0} + M_{3,0}M_{2,0}M_{1,1} + M_{3,1}M_{2,0}M_{1,0}$. Then,

Proposition D.3. *Consider the GNN Architecture defined in Algorithm 1, and recall the definition of the empirical moment between vertices i and j,*

$$\hat{W}_{n,i,j}^{(k)} = \frac{1}{(n-1)^{k-1}}\sum_{r_1,\ldots,r_{k-1} \leq n} a_{ir_1}a_{r_1r_2}\ldots a_{r_{k-1}j}$$

as in Equation (7). Then for $k \geq 0$, we have

$$\lambda_i^k = \frac{1}{\sqrt{n-1}}\sum_{\ell \leq n} \left(\sum_{q=0}^k N_q^k \cdot \hat{W}_{n,i,\ell}^{q+1}\right) Z_\ell. \tag{61}$$

Proof of Proposition D.3. We proceed through induction. The induction base case of $k = 0$ is satisfied by definition of λ_i^0. Now, suppose for induction that, for $k = K$,

$$\lambda_i^K = \frac{1}{\sqrt{n-1}}\sum_{\ell \leq n} \left(\sum_{q=0}^K N_q^K \cdot \hat{W}_{n,i,\ell}^{q+1}\right) Z_\ell.$$

We use the definition of our GNN iteration to compute λ_i^{K+1}. In particular, we observe that λ_i^{K+1} will be a linear combination of the Z_ℓ, where the coefficient of Z_ℓ is given by

$$
M_{K+1,0}\frac{1}{\sqrt{n-1}}\left(\sum_{q=0}^{K} N_q^K \cdot \hat{W}_{n,i,\ell}^{q+1}\right) + M_{K+1,1}\frac{1}{n-1}\sum_{r\leq n} a_{ir}\left(\frac{1}{\sqrt{n-1}}\sum_{q=0}^{K} N_q^K \cdot \hat{W}_{n,r,\ell}^{q+1}\right)
$$

$$
=\frac{1}{\sqrt{n-1}}M_{K+1,0}N_0^K \cdot \hat{W}_{n,i,\ell}^1
$$

$$
+\frac{1}{\sqrt{n-1}}\left(\sum_{q=1}^{K}\left(M_{K+1,0}N_q^K \cdot \hat{W}_{n,i,\ell}^{q+1} + \frac{1}{n-1}M_{K+1,1}N_{q-1}^K \cdot \hat{W}_{n,r,\ell}^q\sum_{r\leq n} a_{ir}\right)\right)
$$

$$
+\frac{1}{\sqrt{n-1}}\left(\frac{1}{n-1}M_{K+1,1}N_q^K \cdot \hat{W}_{n,r,\ell}^{K+1}\sum_{r\leq n} a_{ir}\right). \tag{62}
$$

To arrive at the desired result, we first make a few observations. Firstly, we note that

$$
\frac{1}{n-1}\hat{W}_{n,r,\ell}^q\sum_{r\leq n} a_{ir} = \frac{1}{(n-1)^q}\left(\sum_{r_1,\ldots,r_{q-1}\leq n} a_{rr_1}a_{r_1r_2}\ldots a_{r_{q-1}\ell}\right)\sum_{r\leq n} a_{ir}
$$

$$
=\frac{1}{(n-1)^q}\sum_{r_1,r_2,\ldots,r_q} a_{ir_1}a_{r_1r_2}\ldots a_{r_q\ell}
$$

$$
=\hat{W}_{n,i,\ell}^{q+1}, \tag{63}
$$

which allows us to simplify the analogous quantities in the last two terms. To simplify the second term, we use the definition of N_q^K and note that

$$
M_{K+1,0}N_q^K + M_{K+1,1}N_{q-1}^K = N_q^{K+1}.
$$

To see this, we note that

$$
M_{K+1,0}N_q^K + M_{K+1,1}N_{q-1}^K = M_{K+1,0}\sum_{\substack{r_1,\ldots,r_K\in\{0,1\}\\ \sum_{i=1}^K r_i=q}} M_{K,r_1}M_{K-1,r_2}\ldots M_{1,r_K}
$$

$$
+ M_{K+1,1}\sum_{\substack{r_1,\ldots,r_K\in\{0,1\}\\ \sum_{i=1}^K r_i=q-1}} M_{K,r_1}M_{K-1,r_2}\ldots M_{1,r_K}
$$

$$
=\sum_{\substack{r_1,\ldots,r_K,r_{K+1}\in\{0,1\}\\ \sum_{i=1}^K r_i=q}} M_{K+1,r_1}M_{K,r_2}M_{K-1,r_2}\ldots M_{1,r_{K+1}}
$$

$$
= N_q^{K+1},
$$

which allows us to simplify the second term. Finally, we see that the coefficient of Z_ℓ is given by

$$
\frac{1}{\sqrt{n-1}}N_0^{K+1} \cdot \hat{W}_{n,i,\ell}^1 + \frac{1}{\sqrt{n-1}}\sum_{q=1}^{K}\left(N_q^{K+1} \cdot \hat{W}_{n,i,\ell}^{q+1}\right) + \frac{1}{\sqrt{n-1}}N_q^{K+1} \cdot \hat{W}_{n,i,l}^{K+2}
$$

$$
=\frac{1}{\sqrt{n-1}}\sum_{\ell\leq n}\sum_{q=0}^{K+1} N_q^K \cdot \hat{W}_{n,i,\ell}^{q+1}.
$$

Hence, we obtain that

$$
\lambda_i^{K+1} = \left(\frac{1}{\sqrt{n-1}}\sum_{\ell\leq n}\sum_{q=0}^{K+1} N_q^K \cdot \hat{W}_{n,i,\ell}^{q+1}\right) Z_\ell, \tag{64}
$$

as desired. $\qquad\square$

D.4. Expectation of Dot Products and their Concentration

The following lemma shows that the expectation of the dot products of the embedding vectors, conditional on the graph, is a linear combination of the empirical moments $\hat{W}_{i,j}^k$.

Lemma D.4. *Suppose that λ_i^k are produced through Algorithm 1. Then, conditional on the latent features $(\omega_i)_{i=1}^n$ and the adjacency matrix A, we have*

$$\mathbb{E}\left[\langle \lambda_i^{k_1}, \lambda_j^{k_2}\rangle | A, (\omega_i)_{i=1}^n\right] = \frac{1}{d_n} \sum_{q_1=0}^{k_1} \sum_{q_2=0}^{k_2} \text{Tr}\left(\left(N_{q_1}^{k_1}\right)^T N_{q_2}^{k_2}\right) \hat{W}_{n,i,j}^{q_1+q_2+2}. \tag{65}$$

Proof of Lemma D.4. Firstly, we note that if $W \sim \mathcal{N}(0, I_k)$, then $\mathbb{E}[W^T A W] = \text{Tr}(A)$. Then, we can compute

$$\mathbb{E}\left[\langle \lambda_i^{k_1}, \lambda_j^{k_2}\rangle | A, (\omega_i)_{i=1}^n\right] = \frac{1}{d_n} \sum_{\ell \leq n} \text{Tr}\left(\left(\frac{1}{\sqrt{n-1}} \sum_{q=0}^{k_1} N_q^{k_1} \cdot \hat{W}_{n,i,\ell}^{q+1}(A)\right)^T \left(\frac{1}{\sqrt{n-1}} \sum_{q=0}^{k_2} N_q^{k_2} \cdot \hat{W}_{n,j,\ell}^{q+1}(A)\right)\right) \tag{66}$$

$$= \frac{1}{d_n} \sum_{q_1=0}^{k_1} \sum_{q_2=0}^{k_2} \text{Tr}\left(\left(N_{q_1}^{k_1}\right)^T N_{q_2}^{k_2} \frac{1}{n-1} \sum_{\ell \leq n} \hat{W}_{n,i,\ell}^{q_1+1}(A) \hat{W}_{n,j,\ell}^{q_2+1}(A)\right) \tag{67}$$

$$= \frac{1}{d_n} \sum_{q_1=0}^{k_1} \sum_{q_2=0}^{k_2} \text{Tr}\left(\left(N_{q_1}^{k_1}\right)^T N_{q_2}^{k_2}\right) \hat{W}_{n,i,j}^{q_1+q_2+2}. \tag{68}$$

$\square$

Now that these properties of the embedding vectors have been shown, we now return to the setting of our algorithm, where the weight matrices $M_{k,i}$ are chosen to be the identity. We now prove that the algorithm to produce estimators $\hat{q}_{ij}^{(k)}$ for $W_{ij}^{(k)}$ in Algorithm 1 is asymptotically consistent, and we establish the convergence rate. For the reader's convenience, we rewrite the algorithm below. The following lemma explains the intuition as to why we expect $\hat{q}_{i,j}^{(k)}$ to be an estimator for $\hat{W}_{i,j}^{(k)}$.

Algorithm 4 GNN Architecture and Estimators for the Graphon Moments

---

Input: a Graph $G = (V, E)$; $n := |V|$.
Output: estimators $\hat{q}_{ij}$ for the edge probability W_{ij}.

Computing Estimators for $W_{ij}^{(k)}$:
for $i \neq j$ **do**
$\quad \hat{q}_{i,j}^{(2)} := \langle \lambda_i^0, \lambda_j^0\rangle.$
end
for $k \in \{3, 4, \ldots, L+2\}$ **do**
$\quad \hat{q}_{i,j}^{(k)} := \langle \lambda_i^{k-2}, \lambda_j^0\rangle - \sum_{r=0}^{k-3} \binom{k-2}{r} \hat{q}_{r+2}$
end
Return: $\left\{(\hat{q}_{ij}^{(2)}, \hat{q}_{ij}^{(3)}, \ldots, \hat{q}_{ij}^{(L+2)})_{i \neq j}\right\}$

Lemma D.5. *As in Algorithm 1, define (with the weight matrices $M_{k,i} = I_{d_n}$)*

$$\hat{q}_{i,j}^{(2)} := \langle \lambda_i^0, \lambda_j^0\rangle, \quad \hat{q}_{i,j}^{(k)} := \langle \lambda_i^{k-2}, \lambda_j^0\rangle - \sum_{r=0}^{k-3} \binom{k-2}{r} \hat{q}_{r+2}.$$

Then

$$\hat{q}_{i,j}^{(k)} = \left\langle \frac{1}{\sqrt{n-1}} \sum_{\ell \leq n} a_{j,\ell} Z_\ell, \frac{1}{\sqrt{n-1}} \sum_{\ell \leq n} \hat{W}_{n,i,\ell}^{(k-1)} Z_\ell \right\rangle.$$

Under the heuristic that $Z_{\ell_1}^T Z_{\ell_2} \approx \mathbb{I}(\ell_1 = \ell_2)$, then we see that $\hat{q}_{i,j}^{(k)} \approx \hat{W}_{n,i,\ell}^{(k)}$.

Proof of Lemma D.5. We first show that we can write

$$\hat{q}_{i,j}^{(k)} = \left\langle \lambda_j^0, \sum_{r=0}^{k-2} \binom{k-2}{k-2-r}(-1)^{k-2-r} \cdot \lambda_i^r \right\rangle. \tag{69}$$

Note equivalently this can be written as

$$\hat{q}_{i,j}^{(k)} = \left\langle \lambda_j^0, \sum_{r=0}^{k-2} \binom{k-2}{r}(-1)^r \cdot \lambda_i^{k-2-r} \right\rangle.$$

We show this using induction. Assume this is true for all $k \leq K$ for some K. We can compute $\hat{q}_{i,j}^{(K+1)}$ using the formula in Algorithm 1. Using the definition of $\hat{q}_{i,j}^{(K+1)}$ we can compute that the coefficient of λ_i^a in $\hat{q}_{i,j}^{(K+1)}$ is given by

$$-\sum_{r=a}^{K-2} \binom{K-1}{r}\binom{r}{r-a}(-1)^{r-a} = -\sum_{r=a}^{K-2} \binom{K-1}{K-r-1}\binom{r}{a}(-1)^{r-a}.$$

To compute this, we first argue that

$$\sum_{r=a}^{K-1} \binom{K-1}{K-r-1}\binom{r}{a}(-1)^r = 0.$$

We use generating functions. We note that $\binom{K-1}{r}(-1)^r$ is the coefficient of x^{K-r-1} in the expansion of $(1-x)^{K-1}$. Then, we note that $\binom{r}{a}$ is the coefficient of x^{r-a} in the expansion of $\frac{1}{(1-x)^{a+1}}$. Hence, this summation simply represents the coefficient of x^{K-a-1} in the expansion of $(1-x)^{K-a-2}$. However, since $(1-x)^{K-a-2}$ is a degree $K-a-2$ polynomial, the coefficient is simply 0. Hence, this implies that

$$-\sum_{r=a}^{K-2} \binom{K-1}{K-r-1}\binom{r}{a}(-1)^{r-a} = (-1)^{K-1-a}\binom{K-1}{a}.$$

Thus, we have shown that the coefficient of λ_i^a in $\hat{q}_{i,j}^{(K+1)}$ is of the desired form, which suffices to prove Equation (69). Now, continuing with the proof, we recall that Proposition D.3 states that

$$\lambda_i^k = \frac{1}{\sqrt{n-1}} \sum_{\ell \leq n} \left(\sum_{q=0}^k \binom{k}{q} \cdot \hat{W}_{n,i,\ell}^{(q+1)} \right) Z_\ell,$$

so

$$\hat{q}_{i,j}^{(k)} = \left\langle \lambda_j^0, \sum_{r=0}^{k-2} \binom{k-2}{r}(-1)^r \cdot \frac{1}{\sqrt{n-1}} \sum_{\ell \leq n} \left(\sum_{q=0}^{k-2-r} \binom{k-2-r}{q} \cdot \hat{W}_{n,i,\ell}^{(q+1)} Z_\ell \right) \right\rangle.$$

We analyze the second term in the dot product more closely. The coefficient of Z_ℓ in the second term is equal to (ignoring the factor of $1/\sqrt{n-1}$ for now)

$$\sum_{r=0}^{k-2} \sum_{q=0}^{k-2-r} (-1)^r \binom{k-2}{r}\binom{k-2-r}{q} \hat{W}_{n,i,\ell}^{(q+1)}$$

$$= \sum_{q=0}^{k-2} \sum_{r=0}^{k-2-r} (-1)^r \binom{k-2}{r}\binom{k-2-r}{q} \hat{W}_{n,i,\ell}^{(q+1)}$$

$$= \sum_{q=0}^{k-2} \hat{W}_{n,i,\ell}^{(q+1)} \sum_{r=0}^{k-2-q} (-1)^r \binom{k-2}{r}\binom{k-2-r}{q}.$$

Hence it suffices to argue that $\sum_{r=0}^{k-2-q}(-1)^r\binom{k-2}{r}\binom{k-2-r}{q} = 1$ if $q = k-2$, and 0 otherwise. We argue this in Lemma D.6. Assuming that this is true, then we see that

$$
\begin{aligned}
\hat{q}_{i,j}^{(k)} &= \left\langle \lambda_j^0, \sum_{r=0}^{k-2}\binom{k-2}{r}(-1)^r \cdot \frac{1}{\sqrt{n-1}}\sum_{\ell \leq n}\left(\sum_{q=0}^{k-2-r}\binom{k-2-r}{q}\cdot \hat{W}_{n,i,\ell}^{(q+1)}Z_\ell\right)\right\rangle \\
&= \left\langle \lambda_j^0, \frac{1}{\sqrt{n-1}}\sum_{\ell \leq n}\hat{W}_{n,i,\ell}^{(q+1)}Z_\ell\right\rangle \\
&= \left\langle \frac{1}{\sqrt{n-1}}\sum_{\ell \leq n}a_{j\ell}Z_\ell, \frac{1}{\sqrt{n-1}}\sum_{\ell \leq n}\hat{W}_{n,i,\ell}^{(q+1)}Z_\ell\right\rangle,
\end{aligned}
$$

as desired. To conclude the proof, we present and prove Lemma D.6.

Lemma D.6. *Let $k \geq 0$ be an integer. Then*

$$
\sum_{r=0}^{k-q}(-1)^r\binom{k}{r}\binom{k-r}{q} = \begin{cases}0 & q < k \\ 1 & q = k\end{cases}.
$$

Proof. Consider the formal series

$$
(1+x)^k = \sum_{s=0}^{k}\binom{k}{s}x^s, \quad \frac{1}{(x+1)^{q+1}} = \sum_{s=q}^{\infty}(-1)^{s-q}\binom{s}{q}x^{s-q}.
$$

Multiplying these two series, we notice that the desired quantity $\sum_{r=0}^{k-q}(-1)^r\binom{k}{r}\binom{k-r}{q}$ is exactly the coefficient of x^{k-q} in the product of the two series, which is $(1+x)^{k-q-1}$. However, x^{k-q} is a monomial of degree $k-q$, and hence has coefficient 0 in $(1+x)^{k-q-1}$, which has degree $(1+x)^{k-q-1}$ when $q < k$. The notable exception is when $k = q$, and then the coefficient (of the constant term) in $(1+x)^{-1}$ is exactly equal to 1. This suffices for the proof. $\square$

$\square$

We now establish the main concentration result, Proposition D.8. Before doing so, we first state the following lemma.

Lemma D.7. *Let $\xi = (\xi_1, \ldots, \xi_n)$, and let $\xi_1, \ldots, \xi_n$ be independent, zero-mean normal random variables with for all $i = 1, 2, \ldots n$, $\mathbb{E}[\xi_i^2] = \sigma_i^2$. Let $D = Diag(\sigma_1, \ldots, \sigma_n)$. Let B be any $n \times n$ real matrix. Then for all $\epsilon > 0$,*

$$
\mathbb{P}\left(\left|\xi^T B\xi - \mathbb{E}[\xi^T B\xi]\right| > \epsilon\right) \leq \exp\left(-\min\left(\frac{\epsilon}{4\|DBD\|_F}, \frac{\epsilon^2}{16\|DBD\|_F^2}\right)\right) \tag{70}
$$

Proof of Lemma D.7. We adapt Proposition 1 from (Bellec, 2019), which states the following.

Let $\xi = (\xi_1, \ldots, \xi_n)$, and let $\xi_1, \ldots, \xi_n$ be independent, zero-mean normal random variables with for all $i = 1, 2, \ldots n$, $\mathbb{E}[\xi_i^2] = \sigma_i^2$. Let $D = Diag(\sigma_1, \ldots, \sigma_n)$. Let B be any $n \times n$ real matrix. Then for any $x > 0$,

$$
\mathbb{P}\left(\left|\xi^T B\xi - \mathbb{E}[\xi^T B\xi]\right| > 2\|DBD\|_F\sqrt{x} + 2\|DBD\|_2 x\right) \leq \exp(-x).
$$

To adapt this proposition into the form in Lemma D.7, we firstly note that $\|X\|_2 \leq \|X\|_F$, so

$$
2\|DBD\|_F\sqrt{x} + 2\|DBD\|_2 x \leq 2\|DBD\|_F(\sqrt{x} + x) \tag{71}
$$

$$
\leq 4\|DBD\|_F \cdot \max(\sqrt{x}, x) \tag{72}
$$

Then

$$
\mathbb{P}\left(\left|\xi^T B\xi - \mathbb{E}[\xi^T B\xi]\right| > 4\|DBD\|_F \cdot \max(\sqrt{x}, x)\right) \tag{73}
$$

$$
\leq \mathbb{P}\left(\left|\xi^T B\xi - \mathbb{E}[\xi^T B\xi]\right| > 2\|DBD\|_F(\sqrt{x} + x)\right) \tag{74}
$$

$$
\leq \mathbb{P}\left(\left|\xi^T B\xi - \mathbb{E}[\xi^T B\xi]\right| > 2\|DBD\|_F\sqrt{x} + 2\|DBD\|_2 x\right) \tag{75}
$$

$$
\leq \exp(-\mathrm{x}), \tag{76}
$$

which implies that

$$\mathbb{P}\left(\left|\xi^T B \xi - \mathbb{E}[\xi^T B \xi]\right| > \epsilon\right) \leq \exp\left(-\min\left(\frac{\epsilon}{4\|DBD\|_F}, \frac{\epsilon^2}{16\|DBD\|_F^2}\right)\right), \tag{77}$$

as desired. $\qquad\square$

Proposition D.8 (Proposition 4.1). *Suppose that $L \leq n$ and that (H_2) holds. Then, conditional on A and $(\omega_i)_{i=1}^n$, with probability at least $1 - 5/n - n \cdot \exp\left(-\delta_W \rho_n(n-1)/3\right)$ we have that for all $2 \leq k \leq L+2$,*

$$\sup_{i \neq j \in [n]} \left|\hat{q}_{i,j}^{(k)} - W_{n,i,j}^{(k)}\right| \leq \frac{\rho_n^{k-1}}{\sqrt{n-1}} \log(n)^k \left[3a_k \sqrt{\rho_n} + \frac{96a_{k-1}}{\sqrt{d_n}}\right], \tag{78}$$

where $a_k = C(8(k+2))^k k^{k+1} \sqrt{k!}$, where C is some absolute positive constant.

We first introduce the following lemma:

Lemma D.9. *Suppose that the graphon W satisfies condition H_2, and suppose the sparsity factor is ρ_n. Then,*

$$\mathbb{P}\left(\max_{i \in [n]} \frac{1}{n-1} \sum_{\substack{j \leq n \\ j \neq i}} a_{ij} \geq \rho_n(1+\delta)\Big|(\omega_i)_{i=1}^n\right) \leq n \cdot \exp\left(-\frac{\delta^2}{2+\delta} \sum_{\substack{j \leq n \\ j \neq i}} \rho_n W(\omega_i, \omega_j)\right).$$

Choosing $\delta = 1$ yields that with probability at least $1 - n \cdot \exp\left(-\frac{\delta_W}{3}\rho_n(n-1)\right)$, conditional on $(\omega_i)_{i=1}^n$,

$$\max_{i \in [n]} \frac{1}{n-1} \sum_{\substack{j \leq n \\ j \neq i}} a_{ij} < 2\rho_n.$$

Summing over all i, this implies that

Proof of Lemma D.9. We use the following lemma about sums of independent Bernoulli random variables:

Lemma D.10 ((Goemans, 2015), Theorem 4). *Let $X = \sum_{i=1}^n X_i$, where $X_i \sim Bern(p_i)$, and all the X_i are independent. Let $\mu = \mathbb{E}[X] = \sum_{i=1}^n p_i$. Then*

$$\mathbb{P}(X \geq (1+\delta)\mu) \leq \exp\left(-\frac{\delta^2}{2+\delta}\mu\right)$$

for all $\delta > 0$.

Fix i. We note that the random variables $(a_{ij})_{j \neq i}$ are independent conditioned on the $(\omega_r)_{r=1}^n$. Using these variables directly in this lemma above yields

$$\mathbb{P}\left(\sum_{\substack{j \leq n \\ j \neq i}} a_{ij} \geq (1+\delta) \sum_{\substack{j \leq n \\ j \neq i}} \rho_n W(\omega_i, \omega_j)\right) \leq \exp\left(-\frac{\delta^2}{2+\delta} \sum_{\substack{j \leq n \\ j \neq i}} \rho_n W(\omega_i, \omega_j)\right).$$

Then, noting that $\delta_W \leq W(\cdot, \cdot) \leq 1$, and substiting $\delta = 1$, we obtain

$$\mathbb{P}\left(\frac{1}{n-1} \sum_{\substack{j \leq n \\ j \neq i}} a_{ij} < 2\rho_n\right) \geq 1 - \exp\left(-\frac{\delta_W}{3}\rho_n(n-1)\right).$$

A union bound over all $i \in [n]$ concludes the proof.

$\qquad\square$

Proof of Proposition D.8. In the remainder of this proof, we condition on the event in Lemma D.1, which is that

$$\max_{i \neq j} \left| \hat{W}_{n,i,j}^{(k)} - W_{n,i,j}^{(k)} \right| \leq 3 a_k \frac{\rho_n^{k-1/2}}{\sqrt{n-1}} \log(n)^k.$$

This contributes the probability of $3/n$. For simplicity of notation, denote $B_{n,k} := 3 a_k \frac{\rho_n^{k-1/2}}{\sqrt{n-1}} \log(n)^k$. We also condition on the event in Lemma D.9, which contributes the probability of $n \cdot \exp(-\delta_W \rho_n (n-1)/3)$.

Fix some $i \neq j$. We prove the claim for this particular choice of i, j, and then union bound over all pairs at the end of the proof. Recall that Lemma D.5 states that

$$\hat{q}_{i,j}^{(k)} = \left\langle \frac{1}{\sqrt{n-1}} \sum_{\ell \leq n} a_{j,\ell} Z_\ell, \frac{1}{\sqrt{n-1}} \sum_{\ell \leq n} \hat{W}_{n,i,\ell}^{(k-1)} Z_\ell \right\rangle.$$

We first note that because $Z_\ell \sim \frac{1}{\sqrt{d_n}} \mathcal{N}(0, I_{d_n})$, we have that

$$\mathbb{E}_{(Z_\ell)} \left[\hat{q}_{i,j}^{(k)} \big| A, (\omega_i) \right] = \frac{1}{n-1} \sum_{\ell \leq n} a_{j,\ell} \hat{W}_{n,i,\ell}^{(k-1)} = \hat{W}_{n,i,\ell}^{(k)},$$

where this is the expectation is over the randomness in the Gaussian vectors (Z_ℓ). Hence, to show the desired result, it suffices just to show the concentration of a quadratic form of Gaussian vectors. Concretely, writing $Z = (Z_1, Z_2, \ldots, Z_n)$, we can write $\langle \lambda_i^k, \lambda_j^0 \rangle = Z^T C Z$, where

$$C = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} \end{pmatrix}$$

and

$$\begin{aligned} C_{m_1 m_2} &= \left(\frac{1}{\sqrt{n-1}} a_{j,m_2} \right) \cdot \left(\frac{1}{\sqrt{n-1}} \hat{W}_{n,i,m_1}^{(k-1)} \right) I_{d_n} \\ &= \frac{a_{j,m_2} \hat{W}_{n,i,m_1}^{(k-1)}}{n-1} \cdot I_{d_n}. \end{aligned}$$

To show the concentration of the quadratic form $Z^T C Z$, we employ Lemma D.7 to do this. In order to apply Lemma D.7, we first bound the Frobenius norm of C. Noting that

$$\|C\|_F = \sqrt{\sum_{m_1, m_2 \leq n} \|C_{m_1 m_2}\|_F^2},$$

we can write

$$\begin{aligned} \|C\|_F &= \sqrt{\frac{d_n}{(n-1)^2} \sum_{m_1, m_2 \leq n} a_{j,m_2} \left(\hat{W}_{n,i,m_1}^{(k-1)} \right)^2} \\ &= \sqrt{\frac{d_n}{(n-1)^2} \Big(\underbrace{\sum_{m_2 \leq n} a_{j,m_2}}_{\leq 2\rho_n} \Big) \cdot \Big(\sum_{m_1 \leq n} \left(\hat{W}_{n,i,m_1}^{(k-1)} \right)^2 \Big)} \\ &\leq \sqrt{3 \rho_n d_n B_{n,k-1}^2} \\ &= B_{n,k-1} \sqrt{3 \rho_n d_n}. \end{aligned}$$

For ease of notation, we will write $\|C\|_F \leq F B_{n,k-1} \sqrt{\rho_n} \sqrt{d_n}$ for some constant $F \leq \sqrt{3}$. We now use Lemma D.7. Noting that each element of Z is an independent $N(0, 1/d_n)$ random variable, then Lemma D.7 states that when $\frac{\epsilon \sqrt{d_n}}{4FB_{n,k-1}\sqrt{\rho_n}} > 1$, we have

$$\mathbb{P}\left(\left|\hat{q}_{i,j}^{(k)} - \hat{W}_{n,i,j}^{(k)}\right| > \epsilon \,\middle|\, A, (\omega_i)_{i=1}^n\right) \leq 2\exp\left(-\frac{\epsilon\sqrt{d_n}}{4FB_{n,k-1}\sqrt{\rho_n}}\right).$$

Choose $\epsilon = \frac{4FB_{n,k-1}\sqrt{\rho_n}}{\sqrt{d_n}}t$. Then

$$\mathbb{P}\left(\left|\hat{q}_{i,j}^{(k)} - \hat{W}_{n,i,j}^{(k)}\right| > \frac{4FB_{n,k-1}\sqrt{\rho_n}}{\sqrt{d_n}}t \,\middle|\, A, (\omega_i)_{i=1}^n\right) \leq 2\exp\left(-t\right).$$

Now, union bounding over all $k \in \{2, 3, \ldots, L+2\}$ and $i < j$, $i,j \in [n]$, we have that with probability at least $1 - 2\exp(-t + 3\log(\mathrm{n}))$, for all $2 \leq k \leq L+2$, (assuming $L \leq n-1$), conditional on A, $(\omega_i)_{i=1}^n$,

$$\left|\hat{q}_{i,j}^{(k)} - \hat{W}_{n,i,j}^{(k)}\right| \leq \frac{4FB_{n,k-1}\sqrt{\rho_n}}{\sqrt{d_n}}t.$$

Taking $t = 4\log(n)$, we have that with probability $1 - 2/n$, for all $2 \leq k \leq L+2$, conditional on A, $(\omega_i)_{i=1}^n$, and using that $F \leq \sqrt{3}$,

$$\left|\hat{q}_{i,j}^{(k)} - \hat{W}_{n,i,j}^{(k)}\right| \leq \frac{32B_{n,k-1}\sqrt{\rho_n}}{\sqrt{d_n}}\log(n).$$

Now, recall that

$$\max_{i \neq j}\left|\hat{W}_{n,i,j}^{(k)} - W_{n,i,j}^{(k)}\right| \leq 3a_k \frac{\rho_n^{k-1/2}}{\sqrt{n-1}}\log(n)^k = B_{n,k}.$$

Hence, the triangle inequality implies that for all $2 \leq k \leq L+2$, with probability at least $1 - 5/n - n \cdot \exp(-\delta_W \rho_n(n-1)/3)$,

$$\begin{aligned}
\left|\hat{q}_{i,j}^{(k)} - W_{n,i,j}^{(k)}\right| &\leq B_{n,k} + \frac{32B_{n,k-1}\sqrt{\rho_n}}{\sqrt{d_n}}\log(n) \\
&\leq 3a_k\frac{\rho_n^{k-1/2}}{\sqrt{n-1}}\log(n)^k + \frac{32}{\sqrt{d_n}} \cdot 3a_{k-1}\frac{\rho_n^{k-1}}{\sqrt{n-1}}\log(n)^k \\
&= \frac{\rho_n^{k-1}}{\sqrt{n-1}}\log(n)^k\left[3a_k\sqrt{\rho_n} + \frac{96a_{k-1}}{\sqrt{d_n}}\right],
\end{aligned}$$

as desired. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

E. Proof of Theorem 4.4

In this section, we prove Theorem 4.4. We first review some notation used below.

Define the vectors

$$W_n^{(2,k)}(x,y) := \left(W_n^{(2)}(x,y), \ldots, W_n^{(k)}(x,y)\right)$$
$$\hat{q}_{ij}^{(2,k)} = \left(\hat{q}_{ij}^{(2)}, \hat{q}_{ij}^{(3)}, \ldots, \hat{q}_{ij}^{(k)}\right),$$

and recall that $W_{n,i,j}$ denotes $W_n(\omega_i, \omega_j)$. Define

$$\begin{aligned}
r(n, d_n, m) &:= \max_{2 \leq k \leq m+1} \rho_n^{-(k-1)}\left(\frac{\rho_n^{k-1}}{\sqrt{n-1}}\log(n)^k\left[3a_k\sqrt{\rho_n} + \frac{96a_{k-1}}{\sqrt{d_n}}\right]\right) \\
&= \max_{2 \leq k \leq m+1}\left(\frac{\log(n)^k}{\sqrt{n-1}}\left[3a_k\sqrt{\rho_n} + \frac{96a_{k-1}}{\sqrt{d_n}}\right]\right)
\end{aligned}$$

We note that when $\rho_n \gg \log(n)^{2(m+1)}/n$, $r(n, d_n, m) = o(\rho_n)$. The term in the parentheses in the first equation is simply the bound on $\left|\hat{q}_{i,j}^{(k)} - W_{n,i,j}^{(k)}\right|$ presented in Proposition 4.1. $r(n, d_n, m)$ will be a natural quantity that appears later in this section.

We also define

$$R(\beta) = \mathbb{E}\left[\left(\left\langle \beta, W_n^{(2,1+\mathrm{len}(\beta))}(x,y)\right\rangle - W_n(x,y)\right)^2\right]$$

where the expectation is over $x, y \sim \mathrm{Unif}(0,1)$, and define the empirical risk $R_n(\beta)$ as

$$R_n(\beta) = \frac{2}{n(n-1)}\sum_{i<j}^n \left(\left\langle \beta, \hat{q}_{ij}^{(2,1+\mathrm{len}(\beta))}\right\rangle - a_{ij}\right)^2.$$

We also define the out-of-sample test error as

$$R_T(\beta) = \mathbb{E}\left[\left(\left\langle \beta, \hat{q}_{n+1,n+2}^{(2,1+\mathrm{len}(\beta))}\right\rangle - W_n(\omega_{n+1},\omega_{n+2})\right)^2\right]. \tag{79}$$

The following proposition is the main component of Theorem 4.4.

Proposition E.1 (Theorem 4.4). *Let $\mathcal{F} = \prod_{i=1}^k [-a_i, a_i]$ be a subset of $\mathbb{R}^k$, where $a_i = b_i/\rho_n^i$ for some $b_i > 0$. Let k be a positive integer and define*

$$\hat{\beta}^{n,k} := \arg\min_{\beta \in \mathcal{F}} R_n(\beta), \quad \beta^{*,k} := \arg\min_{\beta \in \mathcal{F}} R(\beta).$$

Define $D = \sum_{i=1}^k |b_i|$. Then with probability at least $1 - 5/n - n \cdot \exp(-\delta_{\mathrm{W}}\rho_n(n-1)/3) - \delta$,

$$R_T(\hat{\beta}^{n,k}) \le R(\beta^{*,k}) + 6D\rho_n \cdot r(n, d_n, k)(T+2) + 3D^2 r(n, d_n, k)^2 + \tilde{O}\left(\frac{\rho_n^2(T+1)^2}{\sqrt{n}}\right)$$

where $T = (1 - \delta_W)\sum_{r=1}^k b_r(1-\delta_W)^r$ and the $\tilde{O}$ constant depends on $\sqrt{\log(1/\delta)}$.

Proof of Theorem 4.4. We write

$$R_T(\hat{\beta}^{n,k}) \le R(\beta^{*,k}) + |R(\hat{\beta}^{n,k}) - R(\beta^{*,k})| + |R_T(\hat{\beta}^{n,k}) - R(\hat{\beta}^{n,k})|. \tag{80}$$

We first bound

$$R(\hat{\beta}^{n,k}) - R(\beta^{*,k}) = \left[R(\hat{\beta}^{n,k}) - R_n(\hat{\beta}^{n,k})\right] + \left[R_n(\hat{\beta}^{n,k}) - R_n(\hat{\beta}^{*,k})\right] + \left[R_n(\hat{\beta}^{*,k}) - R(\hat{\beta}^{*,k})\right].$$

We note that the LHS is ≥ 0 by definition of $\beta^{*,k}$. We note that the second term on the RHS is ≤ 0 by definition of $\hat{\beta}^{n,k}$. Hence, it follows that

$$|R(\hat{\beta}^{n,k}) - R(\beta^{*,k})| \le \left|\left[R(\hat{\beta}^{n,k}) - R_n(\hat{\beta}^{n,k})\right] + \left[R_n(\hat{\beta}^{*,k}) - R(\hat{\beta}^{*,k})\right]\right|.$$

Lemma E.2 states that

$$R_n(\beta) - R(\beta) = \frac{2}{n(n-1)}\sum_{i<j}(a_{ij} - (W_{n,ij})^2) + S_2(\beta) + S_3(\beta) + K_n(\beta) - \mathbb{E}[K_n(\beta)],$$

so hence

$$\begin{aligned}
|R(\hat{\beta}^{n,k}) - R(\beta^{*,k})| &\le |\left[R(\hat{\beta}^{n,k}) - R_n(\hat{\beta}^{n,k})\right] + \left[R_n(\hat{\beta}^{*,k}) - R(\hat{\beta}^{*,k})\right]| \\
&\le |S_2(\hat{\beta}^{n,k}) + S_3(\hat{\beta}^{n,k}) + K_n(\hat{\beta}^{n,k}) - \mathbb{E}[K_n(\hat{\beta}^{n,k})]| \\
&\quad + |S_2(\hat{\beta}^{*,k}) + S_3(\hat{\beta}^{*,k}) + K_n(\hat{\beta}^{*,k}) - \mathbb{E}[K_n(\hat{\beta}^{*,k})]| \\
&\le 4D\rho_n \cdot r(n, d_n, k) \cdot (T+2) + 2D^2 r(n, d_n, k)^2 + \tilde{O}\left(\frac{\rho_n^2(T+1)^2}{\sqrt{n}}\right), \tag{81}
\end{aligned}$$

where the last inequality follows from Lemma E.2. We now bound $|R_T(\hat\beta^{n,k}) - R(\hat\beta^{n,k})|$. For any $\beta \in S$, write

$$
\begin{aligned}
R_T(\beta) &= \mathbb{E}\left[\left(\left\langle \beta, \hat{q}_{n+1,n+2}^{(2,k+1)}\right\rangle - W_n(\omega_{n+1}, \omega_{n+2})\right)^2\right] \\
&= \mathbb{E}\left[\left(\left\langle \beta, W_{n,n+1,n+2}^{(2,k+1)}\right\rangle + \left\langle \beta, \hat{q}_{n+1,n+2}^{(2,k+1)} - W_{n,n+1,n+2}^{(2,k+1)}\right\rangle - W_{n,n+1,n+2}\right)^2\right] \\
&= \underbrace{\mathbb{E}\left[\left(\left\langle \beta, W_{n,n+1,n+2}^{(2,k+1)}\right\rangle - W_{n,n+1,n+2}\right)^2\right]}_{R(\beta)} \\
&\quad + \mathbb{E}\left[2\cdot\left(\left\langle \beta, W_{n,n+1,n+2}^{(2,k+1)}\right\rangle - W_{n,n+1,n+2}\right)\cdot\left\langle\beta, \hat{q}_{n+1,n+2}^{(2,k+1)} - W_{n,n+1,n+2}^{(2,k+1)}\right\rangle\right] \\
&\quad + \mathbb{E}\left[\left\langle\beta, \hat{q}_{n+1,n+2}^{(2,k+1)} - W_{n,n+1,n+2}^{(2,k+1)}\right\rangle^2\right],
\end{aligned}
$$

which implies that (using similar arguments as in the proof of Lemma F.4),

$$
|R_T(\beta) - R(\beta)| \le 2D\rho_n(T+1)r(n,d_n,k) + D^2 r(n,d_n,k)^2 \tag{82}
$$

Substituting Equation (81) and Equation (82) into Equation (80) yields the desired result.

$\square$

The following lemma is used directly in the above proof of Theorem 4.4. We state it and prove it below.

Lemma E.2. *Let $\mathcal{F} = \prod_{i=1}^{k}[-a_i, a_i]$ be a subset of $\mathbb{R}^k$, where $a_i = b_i/\rho_n^i$ for some $b_i > 0$. Let $\beta \in S$ be arbitrary. Define $D = \sum_{i=1}^{k}|b_i|$. Then*

$$
R_n(\beta) - R(\beta) - \frac{2}{n(n-1)}\sum_{i<j}(a_{ij} - (W_{n,ij})^2) = S_2(\beta) + S_3(\beta) + K_n(\beta) - \mathbb{E}[K_n(\beta)].
$$

Furthermore, employing Lemma F.2, Lemma F.4, and Lemma F.5 implies that with probability at least $1 - 5/n - n \cdot \exp(-\delta_W\rho_n(n-1)/3) - \delta$,

$$
\left|R_n(\beta) - R(\beta) - \frac{2}{n(n-1)}\sum_{i<j}(a_{ij} - (W_{n,ij})^2)\right| \le 2D\rho_n\cdot r(n,d_n,k)\cdot(T+2) + D^2\cdot r(n,d_n,k)^2 + \tilde{O}\left(\frac{\rho_n^2(T+1)^2}{\sqrt{n}}\right),
$$

where $T = (1 - \delta_W)\sum_{r=1}^{k}b_r(1 - \delta_W)^r$ and the $\tilde{O}$ constant depends on $\sqrt{\log(1/\delta)}$. We note that this is the probability at which this lemma holds, since Lemma F.2, Lemma F.4, and Lemma F.5 all condition on the same events, so the probabilities in their respective statements do not add.

Proof of Lemma E.2. Let $\beta \in S$ be arbitrary. Consider

$$
\begin{aligned}
R_n(\beta) &= \frac{2}{n(n-1)} \sum_{i<j} \left(\left\langle \beta, \hat{q}_{ij}^{(2,k+1)} \right\rangle - a_{ij} \right)^2 \\
&= \frac{2}{n(n-1)} \sum_{i<j} \left(\left\langle \beta, W_{n,ij}^{(2,k+1)} + \hat{q}_{ij}^{(2,k+1)} - W_{n,ij}^{(2,k+1)} \right\rangle - a_{ij} \right)^2 \\
&= \underbrace{\frac{2}{n(n-1)} \sum_{i<j} \left(\left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle - a_{ij} \right)^2}_{S_1(\beta)} \\
&\quad + \underbrace{\frac{2}{n(n-1)} \sum_{i<j} \left[2\langle \beta, \hat{q}_{ij}^{(2,k+1)} - W_{n,ij}^{(2,k+1)} \rangle (\langle \beta, W_{n,ij}^{(2,k+1)} \rangle - a_{ij}) \right]}_{S_2(\beta)} \\
&\quad + \underbrace{\frac{2}{n(n-1)} \sum_{i<j} \langle \beta, \hat{q}_{ij}^{(2,k+1)} - W_{n,ij}^{(2,k+1)} \rangle^2}_{S_3(\beta)}
\end{aligned}
$$

We analyze these three terms successively. We first rewrite $S_1(\beta)$ as

$$
\frac{2}{n(n-1)} \sum_{i<j} \left(\left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle^2 - 2a_{ij} \left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle + a_{ij} + (W_{n,ij})^2 - (W_{n,ij})^2 \right)
$$
$$
= \frac{2}{n(n-1)} \sum_{i<j} \left[\left(\left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle^2 - 2a_{ij} \left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle + (W_{n,ij})^2 \right) + \left(a_{ij} - (W_{n,ij})^2 \right) \right]
$$

We observe that

$$
\underbrace{\frac{2}{n(n-1)} \sum_{i<j} \left(\left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle^2 - 2a_{ij} \left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle + (W_{n,ij})^2 \right)}_{K_n(\beta)}
$$

has expectation

$$
R(\beta) = \mathbb{E}\left[\left(\left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle - W_{n,ij} \right)^2 \right]. \tag{83}
$$

Hence, we can write

$$
R_n(\beta) = K_n(\beta) + S_2(\beta) + S_3(\beta) + \frac{2}{n(n-1)} \sum_{i<j} (a_{ij} - (W_{n,ij})^2), \tag{84}
$$

and, we obtain that

$$
R_n(\beta) - R(\beta) - \frac{2}{n(n-1)} \sum_{i<j} (a_{ij} - (W_{n,ij})^2) = S_2(\beta) + S_3(\beta) + K_n(\beta) - \mathbb{E}[K_n(\beta)] \tag{85}
$$

The result then follows by invoking Lemma F.2, Lemma F.4, and Lemma F.5. □

F. Proofs of Lemma F.1, Lemma F.2, Lemma F.4, and Lemma F.5

This section presents Lemma F.1, which is used in Theorem 4.4, and its proof. We also present Lemma F.2, Lemma F.4, and Lemma F.5, which are used in the proof of Theorem 4.4.

F.1. Proof of Lemma F.1

Lemma F.1. *Suppose that W has finite distinct rank m_W, and let $\beta^{*,m_W} \in \mathbb{R}^{m_W}$ so that*

$$W(x,y) = \sum_{r=1}^{m_W} \beta_r^{*,m_W} W^{(r+1)}(x,y).$$

Let $v = (v_1, v_2, \ldots, v_k)$ denote the vector that minimizes

$$\left\| W(x,y) - \sum_{r=1}^{k} v_r W^{(r+1)}(x,y) \right\|_{L^2}.$$

Then

$$\left\| W(x,y) - \sum_{r=1}^{k} v_r W^{(r+1)}(x,y) \right\|_{L^2} \leq \sqrt{ \sum_{s=1}^{m_W} \left[\sum_{r=k}^{m_W} \beta_r^{*,m_W} \left(\mu_s^{r+1} - \mu_s^{k+1} \right) \right]^2 }$$

Proof of Lemma F.1. By definition of v being a minimizer of $\left\| W(x,y) - \sum_{r=1}^{k} v_r W^{(r+1)}(x,y) \right\|_{L^2}$, this quantity would be bounded by the error incurred if we replace v with the vector $w = \left(\beta_1^{*,m_W}, \beta_2^{*,m_W}, \ldots, \beta_{k-1}^{*,m_W}, \sum_{s=k}^{m_W} \beta_s^{*,m_W} \right)$. This would yield

$$\| W(x,y) - \sum_{r=1}^{k} w_r W^{(r+1)}(x,y) \|_2$$

$$= \| \sum_{r=1}^{m_W} \beta_r^{*,m_W} W^{(r+1)}(x,y) - \sum_{r=1}^{k} w_r W^{(r+1)}(x,y) \|_2$$

$$= \| \sum_{r=1}^{m_W} \beta_r^{*,m_W} W^{(r+1)}(x,y) - \sum_{r=1}^{k-1} \beta_r^{*,m_W} W^{(r+1)}(x,y) - \left(\sum_{s=k}^{m_W} \beta_s^{*,m_W} \right) W^{(k+1)}(x,y) \|_2$$

$$= \| \sum_{r=k}^{m_W} \beta_r^{*,m_W} W^{(r+1)}(x,y) - \left(\sum_{s=k}^{m_W} \beta_s^{*,m_W} \right) W^{(k+1)}(x,y) \|_2$$

$$= \| \sum_{r=k}^{m_W} \beta_r^{*,m_W} \left(W^{(r+1)}(x,y) - W^{(k+1)}(x,y) \right) \|_2$$

$$= \| \sum_{r=k}^{m_W} \beta_r^{*,m_W} \left(\sum_{s=1}^{m_W} \left(\mu_s^{r+1} - \mu_s^{k+1} \right) \phi_s(x)\phi_s(y) \right) \|_2$$

$$= \| \sum_{s=1}^{m_W} \left(\sum_{r=k}^{m_W} \beta_r^{*,m_W} \left(\mu_s^{r+1} - \mu_s^{k+1} \right) \right) \phi_s(x)\phi_s(y) \|_2$$

$$= \sqrt{ \sum_{s=1}^{m_W} \left[\sum_{r=k}^{m_W} \beta_r^{*,m_W} \left(\mu_s^{r+1} - \mu_s^{k+1} \right) \right]^2 }$$

$\square$

F.2. Proof of Lemma F.2

Lemma F.2. *Let $\mathcal{F} = \prod_{i=1}^{k} [-a_i, a_i]$ be a subset of $\mathbb{R}^k$, where $a_i = b_i / \rho_n^i$ for some $b_i > 0$. Define $D = \sum_{i=1}^{k} |b_i|$, and define*

$$K_n(\beta) := \frac{2}{n(n-1)} \sum_{i<j} \left(\left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle^2 - 2a_{ij} \left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle + (W_{n,ij})^2 \right).$$

Let

$$T = (1 - \delta_W) \sum_{r=1}^{k} b_r (1 - \delta_W)^r.$$

Then with probability at least $1 - 5/n - \exp(-\delta_W \rho_n (n-1)/3) - \delta$, we have

$$\sup_{\beta \in \mathcal{F}} |K_n(\beta) - \mathbb{E}[K_n(\beta)]| = \tilde{O}\left(\frac{\rho_n^2 (T+1)^2}{\sqrt{n}} \right), \tag{86}$$

where $\tilde{O}$ hides logarithmic factors.

Proof of Lemma F.2. In the proof of this lemma, we are inherently conditioning on all of the events that the proof of Proposition D.8 conditions on. Specifically, we are conditioning on the event that

$$\sup_{i \neq j \in [n]} \left| \hat{q}_{i,j}^{(k)} - W_{n,i,j}^{(k)} \right| \leq \frac{\rho_n^{k-1}}{\sqrt{n-1}} \log(n)^k \left[3a_k \sqrt{\rho_n} + \frac{96 a_{k-1}}{\sqrt{d_n}} \right],$$

and that

$$\max_{i \in [n]} \frac{1}{n-1} \sum_{\substack{j \leq n \\ j \neq i}} a_{ij} < 2\rho_n \Rightarrow \frac{2}{n(n-1)} \sum_{i<j} a_{ij} < 2\rho_n.$$

Firstly, we note that $|\langle \beta, W^{(2,k+1)}(x,y) \rangle| \leq \rho_n T$. See the proof of Lemma F.4 for a more detailed calculation.

We use an ϵ-net argument to obtain the desired uniform concentration result over the entire space. We first bound the cardinality of an ϵ-net needed to cover S, where the covering sets are ϵ-balls in the L^1 norm in $\mathbb{R}^{m_W}$. We then establish a high-probability bound for the quantity $|K_n(\beta_0) - \mathbb{E}[K_n(\beta_0)]|$ using a concentration inequality for U-statistics, for a fixed β_0. Then, the continuity of $K_n(\beta)$ will yield a bound for $|K_n(\beta) - \mathbb{E}[K_n(\beta)]|$ for all β in the same ϵ-ball as β_0. We then take a union bound over all balls in the ϵ-net to arrive at the conclusion.

We note that a hypercube with side length $2\epsilon/k$ centered at some x is contained in the L^1 ϵ-ball centered at x, so bounding the cardinality of a covering with hypercubes of side length $2\epsilon/k$ would also bound the cardinality of a covering with L^1 ϵ-balls. To determine this cardinality, we can simply consider the construction of tiling S (which is a hyper-rectangle) with hypercubes simply by packing the cubes side-to-side. Hence, we obtain an ϵ-net of size bounded by

$$\prod_{i=1}^{k} 2 \frac{b_i}{\rho_n^i} \frac{k}{2\epsilon} = \frac{1}{\rho_n^{k(k+1)/2}} \left(\frac{k}{\epsilon} \right)^k \prod_{i=1}^{k} b_i. \tag{87}$$

Now, we bound $|K_n(\beta) - \mathbb{E}[K_n(\beta)]|$. In this goal we define

$$K_n^1(\beta) := \frac{2}{n(n-1)} \sum_{i<j} \left(\left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle^2 - 2 W_{n,i,j} \left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle + (W_{n,ij})^2 \right)$$

and

$$K_n^2(\beta) := -\frac{4}{n(n-1)} \sum_{i<j} \left(a_{ij} \left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle - W_{n,i,j} \left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle \right)$$

We remark that $K_n(\beta) = K_n^1(\beta) + K_n^2(\beta)$. Using the triangle inequality, we notice that it is enough to show concentration of $K_n^1(\beta)$ and $K_n^2(\beta)$ around their respective expectations.

We first remark that $\mathbb{E}(K_n^2(\beta)) = 0$ and show concentration $K_n^2(\beta)$ around its expectation. In this goal, notice that conditional on (ω_i), the random variables $(a_{i,j})$ are i.i.d Bernoulli random variables. Moreover we notice that conditionally on the features (ω_i) we have that $P((a_{i,j})) = -\frac{4}{n(n-1)} \sum_{i<j} a_{i,j} \left\langle \beta, W_{n,ij}^{(2,k+1)} \right\rangle$ is a polynomial of degree one of the Bernoulli random variables $(a_{i,j})$. Hence, we use Lemma D.2. We note that $E[P((a_{i,j}))] \leq 2T\rho_n^2$, and the first derivative with respect to $a_{1,2}$ is $\frac{\partial}{\partial a_{1,2}} P((a_{i,j})) = -\frac{4}{n(n-1)} \left\langle \beta, W_{n,1,2}^{(2,k+1)} \right\rangle \leq \frac{4\rho_n T}{n(n-1)}$. Then, for all $\lambda > 0$ we have

Think it should be

$$P\left(\frac{4}{n(n-1)}\left|\sum_{i<j}^{n} a_{i,j}\left\langle\beta, W_{n,ij}^{(2,k+1)}\right\rangle - \mathbb{E}\left[a_{i,j}\left\langle\beta, W_{n,ij}^{(2,k+1)}\right\rangle|(\omega_i)\right]\right| \geq \frac{2\sqrt{2}a_1\lambda}{\sqrt{n(n-1)}}T\rho_n^{3/2}\right) \leq 2G\cdot\exp\left(-\lambda\right),$$

(88)

where G is some constant from Lemma D.2. Moreover we notice that $\mathbb{E}\left[a_{i,j}\left\langle\beta, W_{n,ij}^{(2,k+1)}\right\rangle|(\omega_i)\right] = W_{n,i,j}\left\langle\beta, W_{n,ij}^{(2,k+1)}\right\rangle$. Therefore, we obtain that

$$P\left(\left|K_n^2(\beta)\right| \geq \frac{2\sqrt{2}a_1\lambda}{\sqrt{n(n-1)}}\rho_n^{3/2}T\right) \leq 2G\cdot\exp\left(-\lambda\right)$$

(89)

Then, we derive a concentration bound for $K_n^1(\beta)$, for a fixed vector $\beta \in S$. The randomness in $K_n^1(\beta)$ term comes from the latent features ω_i and we observe that it is a U-Statistic with two variables. To bound the desired quantity, we use the following

Lemma F.3 (Equation (5.7) from (Hoeffding, 1963)). *Let $X_1, X_2, \ldots, X_N$ be independent random variables. For $r \leq n$, consider a random variable of the form*

$$U = \frac{1}{n(n-1)\ldots(n-r+1)}\sum_{i_1\neq i_2\neq\ldots\neq i_r} g(X_{i_1}, \ldots, X_{i_r}).$$

Then if $a \leq g(x_1, x_2, \ldots, x_r) \leq b$, it follows that

$$\mathbb{P}(|U - \mathbb{E}[U]| \geq t) \leq e^{-2\lfloor n/r\rfloor t^2/(b-a)^2}$$

To use this quantity, we first bound $K_n(\beta)$. Using Equation (108), we have

$$\left|\left\langle\beta, W_{n,ij}^{(2,k+1)}\right\rangle^2 - 2W_{n,i,j}\left\langle\beta, W_{n,ij}^{(2,k+1)}\right\rangle + (W_{n,ij})^2\right|$$

(90)

$$\leq \rho_n^2(T^2 + 2T + 1)$$

(91)

$$= \rho_n^2(T+1)^2$$

(92)

Hence, for a fixed β_0, we have that

$$\mathbb{P}\left(\left|K_n^1(\beta_0) - \mathbb{E}[K_n^1(\beta_0)]\right| \geq t\right) \leq 2\exp\left(\frac{-\lfloor\frac{n}{2}\rfloor t^2}{2\rho_n^4(T+1)^4}\right).$$

(93)

We now use continuity to argue that $|K_n(\beta) - \mathbb{E}[K_n(\beta)]|$ is bounded for all β in the ϵ-ball containing β_0. we derive a bound on $|(K_n(\beta_1) - \mathbb{E}[K_n(\beta_1)]) - (K_n(\beta_2) - \mathbb{E}[K_n(\beta_2)])|$ for when $\|\beta_1 - \beta_2\|_1 \leq \epsilon$. We can use the triangle inequality and bound $|K_n(\beta_1) - K_n(\beta_2)|$ and $|\mathbb{E}[K_n(\beta_1)] - \mathbb{E}[K_n(\beta_2)]|$ separately.

We can write

$$|\mathbb{E}[K_n(\beta_1)] - \mathbb{E}[K_n(\beta_2)]| = \left|\mathbb{E}\left[\left(\left\langle\beta_1, W_{n,ij}^{(2,k+1)}\right\rangle - \rho_n W_{ij}\right)^2\right] - \mathbb{E}\left[\left(\left\langle\beta_2, W_{n,ij}^{(2,k+1)}\right\rangle - \rho_n W_{ij}\right)^2\right]\right|$$

(94)

$$= \left|\mathbb{E}\left[\left\langle\beta_1 - \beta_2, W_{n,ij}^{(2,k+1)}\right\rangle\left(\left\langle\beta_1 + \beta_2, W_{n,ij}^{(2,k+1)}\right\rangle - 2\rho_n W_{ij}\right)\right]\right|$$

(95)

$$\leq \rho_n^3\epsilon\cdot 2(T+1)$$

(96)

where the $\rho_n^2\epsilon$ term is from the first term: the L^1 norm of $\beta_1 - \beta_2$ is bounded by ϵ, and each entry in the vector $W_{n,ij}^{(2,k+1)}$ is bounded by ρ_n^2. The factor of $\rho_n(T+1)$ is using Equation (108).

37

In a similar way, we can bound $|K_n(\beta_1) - K_n(\beta_2)|$. We first bound the quantity

$$\left| \frac{2}{n(n-1)} \sum_{i<j} \left(\left\langle \beta_1, W_{n,ij}^{(2,k+1)} \right\rangle^2 - \left\langle \beta_2, W_{n,ij}^{(2,k+1)} \right\rangle^2 \right) \right| \tag{97}$$

$$= \left| \frac{2}{n(n-1)} \sum_{i<j} \left\langle \beta_1 - \beta_2, W_{n,ij}^{(2,k+1)} \right\rangle \cdot \left\langle \beta_1 + \beta_2, W_{n,ij}^{(2,k+1)} \right\rangle \right| \tag{98}$$

$$\leq \rho_n^3 \epsilon \cdot 2T \tag{99}$$

Then we can bound

$$\left| \frac{2}{n(n-1)} \sum_{i<j} \left(2a_{ij} \left\langle \beta_1, W_{n,ij}^{(2,k+1)} \right\rangle - 2a_{ij} \left\langle \beta_2, W_{n,ij}^{(2,k+1)} \right\rangle \right) \right| \tag{100}$$

$$\leq \frac{2}{n(n-1)} \sum_{i<j} \left| 2a_{ij} \left\langle \beta_1 - \beta_2, W_{n,ij}^{(2,k+1)} \right\rangle \right| \tag{101}$$

$$\leq \max_{i<j} \left| \left\langle \beta_1 - \beta_2, W_{n,ij}^{(2,k+1)} \right\rangle \right| \cdot \frac{2}{n(n-1)} \sum_{i<j} |2a_{ij}| \tag{102}$$

$$\leq 4\epsilon \rho_n^3, \tag{103}$$

where the last inequality comes from conditioning on the event mentioned at the beginning of the proof. From here, we can see that

$$|(K_n(\beta_1) - \mathbb{E}[K_n(\beta_1)]) - (K_n(\beta_2) - \mathbb{E}[K_n(\beta_2)])| \leq \rho_n^3 \epsilon (4T + 6) \tag{104}$$

This implies that

$$\mathbb{P}\left(\sup_{\beta \in S_\epsilon} |K_n(\beta) - \mathbb{E}[K_n(\beta)]| \geq t + \frac{2\sqrt{2}a_1\lambda}{\sqrt{n(n-1)}} T\rho_n^{3/2} + \rho_n^3 \epsilon(4T+6) \right) \tag{105}$$

$$\leq 2 \cdot \text{card}(S_\epsilon) \exp\left(\frac{-\lfloor \frac{n}{2} \rfloor t^2}{2\rho_n^4(T+1)^4} \right) - 2G\exp(-\lambda) \tag{106}$$

Choosing

$$t = \rho_n^2(T+1)^2 \sqrt{\frac{1}{4\lfloor \frac{n}{2} \rfloor} \log\left(\frac{4 \cdot \text{card}(S_\epsilon)}{\delta} \right)}, \quad \text{and} \quad \lambda = \log\left(\frac{4G}{\delta} \right),$$

we have that with probability $1 - \delta$,

$$\sup_{\beta \in \mathcal{F}} |K_n(\beta) - \mathbb{E}[K_n(\beta)]| \leq \rho_n^2(T+1)^2 \sqrt{\frac{1}{4\lfloor \frac{n}{2} \rfloor} \log\left(\frac{4 \cdot \text{card}(S_\epsilon)}{\delta} \right)} + \frac{2\sqrt{2}a_1\lambda}{\sqrt{n(n-1)}} T\rho_n^{3/2} \log(4G/\delta) + \rho_n^3 \epsilon(4T+6).$$

Choose $\epsilon = \frac{1}{\sqrt{n}}$. Recall that

$$\text{card}(S_\epsilon) \leq \frac{1}{\rho_n^{k(k+1)/2}} \left(\frac{k}{\epsilon} \right)^k \prod_{i=1}^{k} b_i$$

$$\Rightarrow \log(\text{card}(S_\epsilon)) \leq k\log(k) + k\log(1/\epsilon) + \frac{k(k+1)}{2}\log(1/\rho_n) + \log\left(\prod_{i=1}^{k} b_i \right).$$

Then, we conclude that with probability at least $1 - \delta$,

$$\sup_{\beta \in \mathcal{F}} |K_n(\beta) - \mathbb{E}[K_n(\beta)]| \leq \tilde{O}\left(\frac{\rho_n^2(T+1)^2}{\sqrt{n}} \right),$$

38

where the Big-O constant depends on $\sqrt{\log(1/\delta)}$.

$\square$

Lemma F.4. *Let* $\mathcal{F} = \prod_{i=1}^{k}[-a_i, a_i]$ *be a subset of* $\mathbb{R}^k$, *where* $a_i = b_i/\rho_n^i$ *for some* $b_i > 0$. *Define* $D = \sum_{i=1}^{k}|b_i|$, *and for* $\beta \in \mathcal{F}$, *define*

$$S_2(\beta) := \frac{2}{n(n-1)} \sum_{i<j} \left[2\langle \beta, \hat{q}_{ij}^{(2,k+1)} - W_{n,ij}^{(2,k+1)} \rangle (\langle \beta, W_{n,ij}^{(2,k+1)} \rangle - a_{ij}) \right].$$

Then with probability at least $1 - 5/n - n \cdot \exp(-\delta_W \rho_n(n-1)/3)$,

$$S_2(c) \leq 2D\rho_n \cdot r(n, d_n, k) \cdot (T+2), \tag{107}$$

where $T = (1 - \delta_W) \sum_{r=1}^{k} b_r (1 - \delta_W)^r$.

Proof of Lemma F.4. In the proof of this lemma, we are inherently conditioning on all of the events that the proof of Proposition D.8 conditions on. Specifically, we are conditioning on the event that

$$\sup_{i \neq j \in [n]} \left| \hat{q}_{i,j}^{(k)} - W_{n,i,j}^{(k)} \right| \leq \frac{\rho_n^{k-1}}{\sqrt{n-1}} \log(n)^k \left[3a_k\sqrt{\rho_n} + \frac{96a_{k-1}}{\sqrt{d_n}} \right],$$

and that

$$\max_{i \in [n]} \frac{1}{n-1} \sum_{\substack{j \leq n \\ j \neq i}} a_{ij} < 2\rho_n \Rightarrow \frac{2}{n(n-1)} \sum_{i<j} a_{ij} < 2\rho_n.$$

We first consider $S_2(\beta)$ for some arbitrary $\beta \in S$.

$$S_2(\beta) \leq \frac{2}{n(n-1)} \left| \sum_{i<j}^{n} 2\langle \beta, \hat{q}_{ij}^{(2,k+1)} - W_{n,ij}^{(2,k+1)} \rangle \left(\langle \beta, W_{n,ij}^{(2,k+1)} \rangle - a_{ij} \right) \right|$$

$$\leq 2 \cdot \frac{2}{n(n-1)} \sum_{i<j}^{n} \left| \langle \beta, \hat{q}_{ij}^{(2,k+1)} - W_{n,ij}^{(2,k+1)} \rangle \left(\langle \beta, W_{n,ij}^{(2,k+1)} \rangle - a_{ij} \right) \right|$$

$$= 2 \cdot \frac{2}{n(n-1)} \sum_{i<j}^{n} \left| \langle \beta, \hat{q}_{ij}^{(2,k+1)} - W_{n,ij}^{(2,k+1)} \rangle \right| \left| \left(\langle \beta, W_{n,ij}^{(2,k+1)} \rangle - a_{ij} \right) \right|$$

$$\leq 2D \cdot r(n, d_n, k) \cdot \frac{2}{n(n-1)} \sum_{i<j}^{n} \left| \langle \beta, W_{n,ij}^{(2,k+1)} \rangle - a_{ij} \right|$$

$$\leq 2D \cdot r(n, d_n, k) \cdot \frac{2}{n(n-1)} \sum_{i<j}^{n} \left(\left| \langle \beta, W_{n,ij}^{(2,k+1)} \rangle \right| + a_{ij} \right)$$

$$\leq 2D \cdot r(n, d_n, k) \cdot \frac{2}{n(n-1)} \left(\sum_{i<j}^{n} \left| \langle \beta, W_{n,ij}^{(2,k+1)} \rangle \right| + \sum_{i<j}^{n} a_{ij} \right)$$

We write

$$
\begin{aligned}
|\langle \beta, W_{n,ij}^{(2,k+1)} \rangle| &\leq \left| \sum_{r=1}^{m_W} \beta_r \cdot \rho_n^{r+1} W_{i,j}^{(r+1)} \right| \\
&\leq \left| \sum_{r=1}^{m_W} \beta_r \cdot \rho_n^{r+1} (1 - \delta_W)^{r+1} \right| \\
&\leq \sum_{r=1}^{m_W} \frac{b_r}{\rho_n^r} \cdot \rho_n^{r+1} (1 - \delta_W)^{r+1} \\
&\leq \rho_n \, (1 - \delta_W) \underbrace{\sum_{r=1}^{m_W} b_r (1 - \delta_W)^r}_{T}
\end{aligned}
\tag{108}
$$

This yields the bound

$$
S_2(\beta) \leq 2D\rho_n \cdot r(n, d_n, k) \cdot (T + 2),
$$

where $T = (1 - \delta_W) \sum_{r=1}^{m_W} b_r (1 - \delta_W)^r$. $\qquad\square$

Lemma F.5. *Let $\mathcal{F} = \prod_{i=1}^{k}[-a_i, a_i]$ be a subset of $\mathbb{R}^k$, where $a_i = b_i/\rho_n^i$ for some $b_i > 0$. Define $D = \sum_{i=1}^{k} |b_i|$, and for $\beta \in \mathcal{F}$, define*

$$
S_3(\beta) := \frac{2}{n(n-1)} \sum_{i<j} \langle \beta, \hat{q}_{ij}^{(2,k+1)} - W_{n,ij}^{(2,k+1)} \rangle^2.
$$

Then with probability at least $1 - 5/n - n \cdot \exp(-\delta_W \rho_n (n-1)/3)$,

$$
S_3(\beta) \leq D^2 \cdot r(n, d_n, k)^2.
$$

Proof of Lemma F.5. We bound this term as in the proof of Lemma F.4. It directly follows that

$$
\frac{2}{n(n-1)} \sum_{i<j} \langle \beta, \hat{q}_{ij}^{(2,k+1)} - W_{n,ij}^{(2,k+1)} \rangle^2 \leq D^2 \cdot r(n, d_n, k)^2
$$

$\qquad\square$

G. Proof of Proposition 4.5

In this section, we state a formal version of Proposition 4.5 and provide the proof.

Proposition G.1 (Proposition 4.5, Formal). *Consider a k-community symmetric stochastic block model (see Appendix A.3 for the definition) with parameters $p > q$ and sparsity factor ρ_n, which has eigenvalues $\mu_1 = \frac{p+(k-1)q}{k} > \frac{p-q}{k} = \mu_2$. Fix some $L \geq 1$ and define $\mathcal{F} = \{\beta \in \mathbb{R}^{L+1} | \, \|\beta\|_{L^1} \leq (\mu_1 \rho_n)^{-1}\}$.*

Produce probability estimators $\hat{p}_{i,j}$ for the probability of an edge between vertices i and j using Algorithm 1 and Algorithm 2. Let $r(n, d_n, L+1)$ be defined as in Theorem 4.4. Suppose that n, d_n satisfy

$$
\frac{4\mu_2}{k} \frac{r(n, d_n, L+1)}{\rho_n} + \frac{4}{k^2\mu_1} \left(\frac{r(n, d_n, L+1)}{\rho_n} \right)^2 + \frac{4}{(k-1)} \frac{r(n, d_n, L+1)}{\rho_n} (T+2)
$$

$$
+ \frac{2}{(k-1)\mu_1} \left(\frac{r(n, d_n, L+1)}{\rho_n} \right)^2 + \frac{A\mu_1}{\sqrt{n}} \frac{\log(n)}{k-1} \leq \mu_2^3,
$$

where A is a constant that depends on $\sqrt{\log(1/\delta)}$, for some positive constant $\delta > 0$.

Let $S_{in} = \{(i,j) | i, j \text{ belong to the same community}\}$ and $S_{out} = \{(k, ell) | k, lk, \ell \text{ belong to different communities}\}$. Then, with probability at least $1 - 5/n - \exp(-\delta_W \rho_n \cdot (n-1)) - \delta$, the following event occurs:

$$
\left\{ \min_{(i,j) \in S_{in}} \hat{p}_{i,j} > \max_{(k,\ell) \in S_{out}} \hat{p}_{k,\ell} \right\}
$$

Proof of Proposition 4.5. Consider a k-community symmetric SBM with connection matrix P, where $P_{i,i} = p$ for all $i \in k$ and $P_{i,j} = q$ for all $i \neq j$. We first write the eigen-decomposition of this matrix; there are two eigenvalues and we write an orthogonal basis for their eigenspaces.

$$\lambda_1 = p + q(k-1) : \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \right\}, \lambda_2 = p - q : \left\{ \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ -2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \\ -3 \\ \vdots \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ -(k-1) \end{pmatrix} \right\}.$$

According to Lemma A.1, the eigenvalues of the corresponding graphon W representation are $\frac{p-q}{k}$ and $\frac{p+q(k-1)}{k}$. Call the eigenvalues μ_1 and μ_2, and let $\mu_1 > \mu_2$ without loss of generality. The eigenfunctions ϕ_i of W are also given by Lemma A.1, and are essentially scaled versions of the above eigenvectors. For the remainder of this proof, we assume that the graph was generated from $\rho_n W$ for some sparsity factor ρ_n.

Define $p_n = \rho_n p$, $q_n = \rho_n q$. Using that $W_n(x,y) = \sum_r (\rho_n \mu_r) \phi_r(x) \phi_r(y)$, we see that

$$p_n = \rho_n \mu_1 + \rho_n \mu_2 \underbrace{\left[\sum_{r=1}^{k-1} \frac{k}{r(r+1)} \right]}_{C_1}, \quad q_n = \rho_n \mu_1 + \rho_n \mu_2 \underbrace{\left[-\frac{k}{2} + \sum_{r=2}^{k-1} \frac{k}{r(r+1)} \right]}_{C_2}.$$

Now suppose that a graph $G = (V, E) = ([n], E)$ is generated from W_n. We demonstrate that under the conditions mentioned in Proposition 4.5, Algorithm 1 and Algorithm 2 results in probability predictions $\hat{p}_{i,j}$ so that $\hat{p}_{i,j} > \hat{p}_{k,\ell}$ when $c_i = c_j$ and $c_k \leq c_\ell$. In other words, the probability predictions for all of the intra (within) community edges are higher than the probability predictions for all of the inter (across) community edges.

As in the proposition statement, define $\mathcal{F} = \{\beta \in \mathbb{R}^{L+1} | \, ||\beta||_{L^1} \leq (\mu_1 \rho_n)^{-1}\}$, and suppose that $L \geq 1$ is the number of layers that are computed. In other words, LG-GNN computes the set of embeddings $\{\lambda_i^0, \lambda_i^1, \dots, \lambda_i^L\}$ for all i, and for each pair of vertices i, j, it computes moment estimators $\left\{ \hat{q}_{i,j}^{(2)}, \hat{q}_{i,j}^{(3)}, \dots, \hat{q}_{i,j}^{(L+2)} \right\}$.

Then, in Algorithm 2, we solve the optimization problem

$$\hat{\beta}^{n,L+1} = \arg\min_{\beta \in \mathcal{F}} \sum_{i<j} \left(a_{ij} - \left\langle \beta, \hat{q}_{i,j}^{(2,3,\dots,L+2)} \right\rangle \right)^2.$$

For $i = 1, 2$ and any $\beta \in \mathbb{R}^{L+1}$, define $\hat{\mu}_{n,i}(\beta) = \sum_{r=1}^m \beta_r (\rho_n \mu_i)^{r+1}$, where the subscript n makes implicit that there is dependence on ρ_n. Defining

$$R(\beta) = \mathbb{E}\left[\left(\left\langle \beta, W_n^{(2,L+2)}(x,y) \right\rangle - W_n(x,y) \right)^2 \right],$$

we note that for any fixed $\beta \in \mathcal{F}$, we have that

$$R(\beta) = (\rho_n \mu_1 - \hat{\mu}_{n,1}(\beta))^2 + (k-1) \cdot (\rho_n \mu_2 - \hat{\mu}_{n,2}(\beta))^2.$$

Let ω_i, ω_j be the latent features of two vertices that both correspond to being in community 1, and let ω_k, ω_ℓ be the latent features of two vertices that correspond to being in communities 1 and 2, respectively. Now, suppose that for the edges (i,j) and (k,ℓ), LG-GNN assigns them predicted probabilities $\hat{p}_{i,j} = \left\langle \hat{\beta}^{n,L+1}, \hat{q}_{i,j}^{(2,\dots,L+2)} \right\rangle$, $\hat{p}_{k,\ell} = \left\langle \hat{\beta}^{n,L+1}, \hat{q}_{k,\ell}^{(2,\dots,L+2)} \right\rangle$, respectively, and suppose that $\hat{p}_{k,\ell} > \hat{p}_{i,j}$. Consider

$$\hat{p}_{i,j} = \left\langle \hat{\beta}^{n,L+1}, \hat{q}_{i,j}^{(2,\dots,L+2)} \right\rangle = \left\langle \hat{\beta}^{n,L+1}, W_{n,i,j}^{(2,\dots,L+2)} \right\rangle + \left\langle \hat{\beta}^{n,L+1}, \hat{q}_{i,j}^{(2,\dots,L+2)} - W_{n,i,j}^{(2,\dots,L+2)} \right\rangle$$

$$= \hat{\mu}_{n,1}(\hat{\beta}^{n,L+1}) + C_1 \hat{\mu}_{n,2}(\hat{\beta}^{n,L+1}) + \left\langle \hat{\beta}^{n,L+1}, \hat{q}_{i,j}^{(2,\dots,L+2)} - W_{n,i,j}^{(2,\dots,L+2)} \right\rangle,$$

where we simplified the first term this way in the last equality by using the form of the eigenvectors ϕ_r, and noting that ω_i, ω_j both correspond to vertices in community 1. In a similar way, we have that

$$\hat{p}_{k,\ell} = \left\langle \hat{\beta}^{n,L+1}, \hat{q}_{k,\ell}^{(2,\ldots,L+2)} \right\rangle = \left\langle \hat{\beta}^{n,L+1}, W_{n,k,\ell}^{(2,\ldots,L+2)} \right\rangle + \left\langle \hat{\beta}^{n,L+1}, \hat{q}_{k,\ell}^{(2,\ldots,L+2)} - W_{n,k,\ell}^{(2,\ldots,L+2)} \right\rangle$$

$$= \hat{\mu}_{n,1}(\hat{\beta}^{n,L+1}) + C_2 \hat{\mu}_{n,2}(\hat{\beta}^{n,L+1}) + \left\langle \hat{\beta}^{n,L+1}, \hat{q}_{k,\ell}^{(2,\ldots,L+2)} - W_{n,k,\ell}^{(2,\ldots,L+2)} \right\rangle,$$

Hence, if $\hat{p}_{k,\ell} > \hat{p}_{i,j}$, then noting that $C_1 - C_2 = k$, we have

$$k \hat{\mu}_2(\hat{\beta}^{n,L+1}) < \left\langle \hat{\beta}^{n,L+1}, \hat{q}_{k,\ell}^{(2,\ldots,L+2)} - W_{n,k,\ell}^{(2,\ldots,L+2)} \right\rangle - \left\langle \hat{\beta}^{n,L+1}, \hat{q}_{i,j}^{(2,\ldots,L+2)} - W_{n,i,j}^{(2,\ldots,L+2)} \right\rangle$$

We note that Proposition 4.1 states that with probability at least $1 - 5/n - n \cdot \exp(-\delta_W \rho_n(n-1)/3) - \delta$, we have that for all $2 \le k \le L+2$,

$$\left| \hat{q}_{i,j}^{(k)} - W_{n,i,j}^{(k)} \right| \le \frac{\rho_n^{k-1}}{\sqrt{n-1}} \log(n)^k \left[3a_k \sqrt{\rho_n} + \frac{96 a_{k-1}}{\sqrt{d_n}} \right],$$

for some constants a_k, and also that the conclusion from Lemma F.2 holds. We will be conditioning on these events for the remainder of the proof. We also note that

$$r(n, d_n, L+2) := \max_{2 \le k \le L+2} \rho_n^{-(k-1)} \left(\frac{\rho_n^{k-1}}{\sqrt{n-1}} \log(n)^k \left[3a_k \sqrt{\rho_n} + \frac{96 a_{k-1}}{\sqrt{d_n}} \right] \right)$$

$$= \max_{2 \le k \le L+2} \frac{\log(n)^k}{\sqrt{n-1}} \left[3a_k \sqrt{\rho_n} + \frac{96 a_{k-1}}{\sqrt{d_n}} \right]$$

$$= o(\rho_n) \quad \text{if } \rho_n \gg \rho_n^{2(L+2)} n$$

Using these definitions, noting that $\left\| \hat{\beta}^{n,L+1} \right\|_{L^1} \le \frac{1}{\mu_1 \rho_n}$, we have that

$$\hat{\mu}_{n,2}(\hat{\beta}^{n,L+1}) < \frac{2}{k \mu_1} r(n, d_n, L+1).$$

Define $\beta_0 = (1/(\mu_1 \rho_n), 0, 0, \ldots, 0) \in \mathbb{R}^m$ to have $1/\mu_1$ as the first component, and 0 everywhere else. Now, we note that Lemma E.2 states, with the same probability above, that

$$R(\beta_0) - R(\hat{\beta}^{n,L+1}) = R_n(\beta_0) - R_n(\hat{\beta}^{n,L+1}) + P,$$

where

$$|P| \le \frac{4}{\mu_1} \rho_n \cdot r(n, d_n, L+1)(T+2) + \frac{2}{\mu_1^2} \cdot r(n, d_n, L+1)^2 + A \frac{\rho_n^2}{\sqrt{n}} \log(n),$$

where $T = \frac{p^2}{\mu_1}$ and A is some constant that depends on $\sqrt{\log(1/\delta)}$. We also note that $R_n(\beta_0) - R_n(\hat{\beta}^{n,L+1}) \ge 0$ because $\hat{\beta}^{n,L+1}$ is the minimizer of the empirical risk. This implies that $R(\beta_0) \ge R(\hat{\beta}^{n,L+1}) + P$. So, noting that $R(\beta_0) = \rho_n^2 (k-1) \left(\mu_2 - \frac{\mu_2^2}{\mu_1} \right)^2$, and noting that $\frac{\mu_2}{\mu_1} < 1$,

$$\rho_n^2 (k-1) \left(\mu_2 - \frac{\mu_2^2}{\mu_1} \right)^2 \ge (\rho_n \mu_1 - \hat{\mu}_{n,1}(\hat{\beta}^{n,L+1}))^2 + (k-1)(\rho_n \mu_2 - \hat{\mu}_{n,2}(\hat{\beta}^{n,L+1}))^2 + P$$

$$\Rightarrow \rho_n^2 (k-1) \left(\mu_2 - \frac{\mu_2^2}{\mu_1} \right)^2 \ge (k-1)(\rho_n \mu_2 - \hat{\mu}_{n,2}(\hat{\beta}^{n,L+1}))^2 + P$$

$$\Rightarrow 0 < \rho_n^2 \frac{\mu_2^3}{\mu_1} \left[2 - \frac{\mu_2}{\mu_1} \right] \le (2\rho_n \mu_2 \hat{\mu}_{n,2}(\hat{\beta}^{n,L+1}) + \hat{\mu}_{n,2}(\hat{\beta}^{n,L+1})^2) - \frac{P}{k-1}.$$

However, this is a contradiction when $|(2\rho_n \mu_2 \hat{\mu}_{n,2}(\hat{\beta}^{n,L+1}) - \hat{\mu}_{n,2}(\hat{\beta}^{n,L+1})^2) - \frac{P}{k-1}| < \rho_n^2 \frac{\mu_2^3}{\mu_1} \left[2 - \frac{\mu_2}{\mu_1} \right]$. Consider the bounds

$$|P| \le \frac{4}{\mu_1} \rho_n \cdot r(n, d_n, L+1)(T+2) + \frac{2}{\mu_1^2} \cdot r(n, d_n, L+1)^2 + A \frac{\rho_n^2}{\sqrt{n}} \log(n)$$

Hence, when

$$\frac{4\rho_n\mu_2}{k\mu_1}r(n, d_n, L+1) + \frac{4}{k^2\mu_1^2}r(n, d_n, L+1)^2 + \frac{4\rho_n}{(k-1)\mu_1}r(n, d_n, L+1)(T+2)$$

$$+ \frac{2}{(k-1)\mu_1^2}r(n, d_n, L+1)^2 + A\frac{\rho_n^2}{\sqrt{n}}\frac{\log(n)}{k-1} \leq \rho_n^2\frac{\mu_2^3}{\mu_1},$$

the result follows. Dividing both sides by ρ_n^2, and multiplying by μ_1, the above is equivalent to

$$\frac{4\mu_2}{k}\frac{r(n, d_n, L+1)}{\rho_n} + \frac{4}{k^2\mu_1}\left(\frac{r(n, d_n, L+1)}{\rho_n}\right)^2 + \frac{4}{(k-1)}\frac{r(n, d_n, L+1)}{\rho_n}(T+2)$$

$$+ \frac{2}{(k-1)\mu_1}\left(\frac{r(n, d_n, L+1)}{\rho_n}\right)^2 + \frac{A\mu_1}{\sqrt{n}}\frac{\log(n)}{k-1} \leq \mu_2^3.$$

The result follows.

$\square$

H. Proof of Proposition 6.1

Note that in this proof, we assume that the sparisity factor $\rho_n = 1$. Consider a 2-community stochastic block model (see Appendix A.3 for more details) parameterized by the matrix $\begin{pmatrix} p & r \\ r & q \end{pmatrix}$. The eigenvalues and eigenvectors are given by

$$\lambda_1 = \frac{1}{2}(p + q + A), \quad v_1 = \begin{pmatrix} \frac{p-q+A}{2r} \\ 1 \end{pmatrix}, \quad \lambda_2 = \frac{1}{2}(p + q - A), \quad v_2 = \begin{pmatrix} \frac{p-q-A}{2r} \\ 1 \end{pmatrix},$$

where $A = \sqrt{(p-q)^2 + 4r^2}$. Then, recall that Lemma A.1 states that the eigenvalues of the graphon representation W of this SBM has eigenvalues $\mu_i = \frac{1}{2}\lambda_i$. We also use the eigenfunctions ϕ_i for W as written in Lemma A.1. We recall that Lemma D.4 states that for all $L \geq 0$, we have

$$\mathbb{E}\left[\langle\lambda_i^L, \lambda_j^L\rangle | A, (\omega_i)_{i=1}^n\right] = \sum_{q_1=0}^{L}\sum_{q_2=0}^{L}\binom{L}{q_1}\binom{L}{q_2}\hat{W}_{n,i,j}^{(q_1+q_2+2)}$$

Then, the proof of Proposition D.8 implies that for all i, j,

$$\Rightarrow \langle\lambda_i^L, \lambda_j^L\rangle \xrightarrow{p} \sum_{q_1=0}^{L}\sum_{q_2=0}^{L}\binom{L}{q_1}\binom{L}{q_2}W_{n,i,j}^{(q_1+q_2+2)}$$

In this proof, we let c_i denote the community of vertex i and let S_j denote all of the vertices in community j. In the graphon reprentation of this 2-community SBM, if ω_i is the latent feature for vertex i, then $\omega_i \in [0, 1/2)$ if and only if vertex i belongs to community 1, and $\omega_i \in [1/2, 1]$ if and only if vertex i belongs to community 2. To reflect this and simplify notation, we let $W_{n,S_i,S_j} := W_n(\omega_a, \omega_b)$, where ω_a and ω_b are any $\omega \in [0, 1]$ so that correspond to the appropriate communities. For example, $W_{n,S_1,S_1} = W_n(1/4, 1/4) = p$, which is the probability that two vertices in community 1 are connected. We write the above as

$$\sup_{k \in S_i, \ell \in S_j}\left|\langle\lambda_k^L, \lambda_\ell^L\rangle - \sum_{q_1=0}^{L}\sum_{q_2=0}^{L}\binom{L}{q_1}\binom{L}{q_2}W_{n,S_i,S_j}^{(q_1+q_2+2)}\right| \xrightarrow{p} 0$$

In the remainder of the proof, we choose parameters $p, q, r \in [0, 1]$ so that $\sum_{q_1=0}^{k_1}\sum_{q_2=0}^{k_2}\binom{k_1}{q_1}\binom{k_2}{q_2}W_{n,S_2,S_2}^{(q_1+q_2+2)} = \sum_{q_1=0}^{k_1}\sum_{q_2=0}^{k_2}\binom{k_1}{q_1}\binom{k_2}{q_2}W_{n,S_1,S_2}^{(q_1+q_2+2)}$, but $W_{n,S_2,S_2} \neq W_{n,S_1,S_2}$ (this last equality indicates that the connection probability between two vertices in community 2 is different than the connection probability between a vertex in community 1 and a vertex in community 2). This would suffice for the proof, since the continuous mapping theorem would imply that for any continuous function f,

$$\sup_{k \in S_1, \ell \in S_2} \left| f\left(\langle \lambda_k^L, \lambda_\ell^L \rangle\right) - f\left(\sum_{q_1=0}^{L} \sum_{q_2=0}^{L} \binom{L}{q_1}\binom{L}{q_2} W_{n,S_1,S_2}^{(q_1+q_2+2)}\right) \right| \xrightarrow{P} 0$$

$$\sup_{k \in S_2, \ell \in S_2} \left| f\left(\langle \lambda_k^L, \lambda_\ell^L \rangle\right) - f\left(\sum_{q_1=0}^{L} \sum_{q_2=0}^{L} \binom{L}{q_1}\binom{L}{q_2} W_{n,S_2,S_2}^{(q_1+q_2+2)}\right) \right| \xrightarrow{P} 0,$$

and we note that $W_{n,S_2,S_2} \neq W_{n,S_1,S_2}$. With this in mind, consider

$$\sum_{q_1=0}^{L} \sum_{q_2=0}^{L} \binom{L}{q_1}\binom{L}{q_2} W_{n,S_i,S_j}^{(q_1+q_2+2)} = \sum_{q_1=0}^{L} \sum_{q_2=0}^{L} \binom{L}{q_1}\binom{L}{q_2} \left(\sum_{r=1}^{2} \mu_r^{q_1+q_2+2} \phi_r(S_i)\phi_r(S_j) \right)$$

$$= \sum_{k=0}^{2L} \sum_{r=1}^{2} \binom{2L}{k} \mu_r^{k+2} \phi_r(S_i)\phi_r(S_j)$$

$$= \sum_{r=1}^{2} \mu_r^2 (\mu_r + 1)^{2L} \phi_r(S_i)\phi_r(S_j)$$

Substituting in the forms of the eigenvectors ϕ_1, ϕ_2, it suffices to show that there exist values of p, q, r, with $q \neq r$, so that

$$\mu_1^2(\mu_1 + 1)^{2L} + \mu_2^2(\mu_2 + 1)^{2L} = \frac{p - q + A}{2r}\mu_1^2(\mu_1 + 1)^{2L} + \frac{p - q - A}{2r}\mu_2^2(\mu_2 + 1)^{2L}$$

$$\Leftrightarrow \mu_2^2(\mu_2 + 1)^{2L}(2r - (p - q) + A) = \mu_1^2(\mu_1 + 1)^{2L}(-2r + (p - q) + A),$$

which would suffice for the proof. Recall that $A = \sqrt{(p-q)^2 + 4r^2}$, $\mu_1 = \frac{1}{4}(p + q + A)$, and $\mu_2 = \frac{1}{4}(p + q - A)$. Choosing $q = 0$ and substituting these values in, we obtain that the above is equivalent to

$$\underbrace{\left(p - \sqrt{p^2 + 4r^2}\right)^2 \left(\frac{1}{4}(p - \sqrt{p^2 + 4r^2}) + 1\right)^{2L} (2r - p + \sqrt{p^2 + 4r^2})}_{(1)}$$

$$\underbrace{- \left(p + \sqrt{p^2 + 4r^2}\right)^2 \left(\frac{1}{4}(p + \sqrt{p^2 + 4r^2}) + 1\right)^{2L} (p - 2r + \sqrt{p^2 + 4r^2})}_{(2)} = 0$$

We show that there exist values p and r, $r \neq 0$, so that there exists a root for some $p, r \in (0, 1)$. We will fix $p = \epsilon \ll 1$ to be some small number to be decided later. ϵ might depend on L. Since this function is continuous in all variables, we use the intermediate value theorem (by varying r) to deduce that there exists a root for some sufficiently small p. Firstly, we observe that $\lim_{r \downarrow 0}(1) = 0$. On the other hand, $\lim_{r \downarrow 0}(2) = (2p)^3(1 + p/2)^{2L} > 0$. This implies that $(1) - (2) < 0$ for sufficiently small r. Then, it suffices to argue that $(1) - (2) > 0$ for $r = 1$, as then the intermediate value theorem implies the desired result.

Let $p = \epsilon$. Taylor's theorem implies that $\sqrt{\epsilon^2 + 4} = 2 + \frac{\epsilon^2}{4} + O(\epsilon^4)$. Hence, we can write (1) as

$$(1) = \left(2 - \epsilon + O(\epsilon^2)\right)^2 \left(\frac{1}{2} + \frac{1}{4}\epsilon + O(\epsilon^2)\right)^{2L} (4 - \epsilon + O(\epsilon^2)).$$

We can also write

$$(2) = (2 + \epsilon + O(\epsilon^2)) \left(\frac{3}{2} + \frac{1}{4}\epsilon + O(\epsilon^2)\right)^{2L} (\epsilon + O(\epsilon^2)).$$

We note that the first two terms in both (1) and (2) are of constant order (the constants are difference, but both of constant order). However, the third term in (1) is of constant order, but the third term in (2) is of order ϵ. This implies that for sufficiently small ϵ, $(1) - (2) < 0$. This suffices to imply that for this small enough ϵ, there is some $r > 0$ so that $(1) - (2)$ has a root. This suffices for the proof.

I. Experiments

We present three different sets of results. The first is for real-data (the Cora dataset), and we use the in-sample train/test splitting scheme for this. The second are in-sample experiments for a variety of random graph models, and the third set are out-sample experiments for these random graph models. For clarity, we explain and define the metrics we are using in the experiments. For the real-data experiments, we consider only the in-sample setting, while for the random graph experiments, we consider both the in-sample and out-of-sample settings.

One point of clarification is that our link prediction procedure is not simply to guess a particular edge to be a positive edge if its predicted probability is over 0.5. If the graphon $W(\cdot, \cdot) < 0.5$, then this doens't make sense because the LG-GNN estimates the underlying probability of edges.

Instead, we concern ourselves more with the *ranking* of the edges and choose evaluation metrics to reflect that. Concretely, we evaluate our algorithms on whether they are able to assign higher probabilities to edges with higher underlying probabilities (and in the real-data case, whether they are able to assign higher probabilities to the positive edges than to the negative edges in the testing set).

To reflect this, the principle metrics we use are the AUC-ROC, Hits@k (as is standard for link prediction tasks in the Stanford Open Graph Benchmark) for the real data experiments. For the random graphs, introduce a new metric called the *Probability Ratio@k*, defined below, which is inspired by the Hits@K metric.

I.1. Train/Test Splits

We first describe our train/test split procedures.

I.1.1. IN-SAMPLE (RANDOM GRAPH)

We generate a graph $G = (V, E)$ with $V = [n]$ (n vertices). Let N be the set of non-edges. Concretely, $N = \{(i, j), i \neq j \in [n] | (i, j) \notin E_n\}$. We then split the edges into a train, validation, and testing set as follows.

For each edge $e \in E$, we remove it from the graph (independently from all the other edges) with probability $p = 0.2$. The edges that are not removed are labelled E_{train}. The set E_{train} will be the set of positive training edges. Among the edges that were removed, half of those will be the set of positive validation edges and the remaining will be the set of positive test edges. Call these E_{val} and E_{test}, respectively. During training, message passing only occurs along the edges in E_{train}.

Now, we select the negative edges among the set of edges $N \cup E_{test}$. Specifically, $1 - p$ fraction of these edges will be the negative training edges, $p/2$ fraction will be the negative validation edges, and the remaining $p/2$ fraction will be the negative testing edges.

It is important to pick the negative training edges from the set $N \cup E_{test}$, as opposed to simply from N. If the negative training edges were sampled only from N, then this would give implicit information about where the edges are in the graph. The model should not have access to which edges are in E_{test} vs in N a priori; if the negative training edges that were given to it are only from N, then it would implicitly know that the edges in E_{test} are less likely to be negative edges. Indeed, when we trained the GCN on a 2-community SBM with parameters 80 and 20 in the setting in which the negative training edges were sampled only from N, it was able to estimate the underlying parameters 80 and 20 almost perfectly, which should be impossible if it only had access to a graph with edges removed.

I.1.2. IN-SAMPLE (REAL DATA)

For real data, we use the same train/test split procedure as described above. However, during training and testing, we do not use the entire set of negative edges. This is because the graph is very sparse, and hence there are many more negative edges than positive edges. This makes link prediction difficult and causes the training procedure to be erratic.

I.1.3. OUT-SAMPLE

For each random graph model, we generate a graph $G = (V, E)$ with $V = [n]$ (n vertices). We partition $V = V_1 \cup V_2$, where V_1 contains a random $1 - p = 0.8$-fraction of the original set of vertices, and V_2 contains the remaining p fraction. Let G_1 be the subgraph induced by V_1 (i.e., the set of all positive and negative edges with both endpoints in V_1). Let E_1 be the set of edges that have both endpoints in V_1. Let E_2 be the set of edges that have at least one vertex in V_2, and let N_2 be

the set of negative edges with at least one vertex in V_2.

We pick a random $1 - p$ fraction of the positive and negative edges from G_1 to be the training positive and negative edges, and the remaining p fraction to be the validation edges. Then, we pick p fraction of the positive edges in E_2 and p fraction of the negative edges in N_2 to be the testing edges. The remaining edges in E_2 and N_2 we will refer to as message-passing edges.

We first train the models on the positive + validation edges. Then once the model is trained, we compute the embedding vectors by running message passing on the set of train + message passing edges. Finally, we do edge prediction on the testing edges.

I.2. Definition of Probability Ratio

Let $P(e)$ be the underlying probability of an edge $e = (i, j)$. In the graphon model, $P((i, j)) = W_n(\omega_i, \omega_j)$, and note that we have access to these values. Given a set of edges $E = \{e_i = (v_{i,1}, v_{i,2})\}_{i=1}^{|E|}$, we say that a link prediction algorithm ranks the edges as $e_{i_1} > e_{i_2} > \cdots > e_{i_{|E|}}$ if $\hat{p}_{e_{i_1}} > \hat{p}_{e_{i_2}} > \cdots > \hat{p}_{e_{i_{|E|}}}$, where $\hat{p}_e$ is the probability that the algorithm predicts for the edge e. Given some edge ranking as above, define the total predicted probability as

$$P_{pred,k} := \sum_{r=1}^{k} P(e_{i_r})$$

and the maximum probability as

$$P_{max,k} := \max_{e_1 \neq e_2 \neq \ldots \neq e_k \in E} \sum_{r=1}^{k} P(e_r).$$

In other words, the $P_{max,k}$ is the sum of the probabilities of the top k most likely edges in E. Then, the probability ratio is defined as $P_{pred,k}/P_{max,k}$.

In essence, the Probability Ratio@k captures what fraction of the top k probabilities a link prediction algorithm can capture. For example, suppose that there are three testing edges e_1, e_2, e_3 with underlying probabilities $0.8, 0.5, 0.2$, respectively. Suppose that some edge prediction algorithm ranks the edges as $e_1 > e_3 > e_2$. Then the Probability Ratio@2 is equal to $\frac{0.8+0.2}{0.8+0.5} \approx 0.77$.

I.3. Real-Data: Cora

For the dataset, we perform a train/test split using the StellarGraph edge splitter, which randomly removes positive edges while ensuring that the resulting graph remains connected. For the negative training edges, we sample an equal number of negative edges as positive edges to train on. For the negative testing edges, we sample an equal number of negative edges as positive testing edges.

I.3.1. RESULTS WITHOUT NODE FEATURES

Parameter Set	Model	Cross Entropy	Hits@50	Hits@100
	GCN	**0.645** $\pm$ 0.043	0.496 $\pm$ 0.025	0.633 $\pm$ 0.023
layers=2	LG-GNN	2.953 $\pm$ 0.013	0.565 $\pm$ 0.012	0.637 $\pm$ 0.006
	PLSG-GNN	0.679 $\pm$ 0.012	**0.591** $\pm$ 0.014	**0.646** $\pm$ 0.013
	GCN	0.689 $\pm$ 0.002	0.539 $\pm$ 0.008	0.665 $\pm$ 0.007
layers=4	LG-GNN	2.682 $\pm$ 0.010	0.564 $\pm$ 0.005	0.620 $\pm$ 0.008
	PLSG-GNN	**0.660** $\pm$ 0.030	**0.578** $\pm$ 0.014	0.637 $\pm$ 0.013

Table 7. GCN does not have access to node features

I.3.2. RESULTS WITH NODE FEATURES (GCN HAS ACCESS TO NODE FEATURES)

Table 8. GCN has access to node features

Parameter Set	Model	Cross Entropy	Hits@50	Hits@100
	GCN	**0.487** $\pm$ 0.003	**0.753** $\pm$ 0.019	**0.898** $\pm$ 0.021
layers=2	LG-GNN	3.034 $\pm$ 0.285	0.555 $\pm$ 0.027	0.603 $\pm$ 0.034
	PLSG-GNN	0.679 $\pm$ 0.027	0.577 $\pm$ 0.033	0.626 $\pm$ 0.042
	GCN	**0.661** $\pm$ 0.041	**0.609** $\pm$ 0.072	**0.776** $\pm$ 0.069
layers=4	LG-GNN	2.711 $\pm$ 0.213	0.560 $\pm$ 0.013	0.601 $\pm$ 0.012
	PLSG-GNN	0.677 $\pm$ 0.019	0.574 $\pm$ 0.025	0.625 $\pm$ 0.024

Figure 1 shows histograms of the predicted probabilities by each of the algorithms (with the two cases of the GCN having access or not having access to the node features). This is to give a visual demonstrate as to what PLSG-GNN is doing. There is a "low probability" hump around 0.25, but then smaller peaks of high-probability predictions. The humps clearly separate the edges in regimes of how connected they are and show clearly the properties of the graph topology.

Figure 1. Plot of the predicted probabilities by the PLS Regression (row 1), GCN without node features (row 2), and GCN with node features (row 3). The left column shows for 2 layers, the right column shows for 4 layers.

J. Experiments (In-Sample)

J.1. 6SSBM (80-20)

6-community symmetric stochastic block model with connection probabilities 0.8 and 0.2.

Table 9. Symmetric Stochastic Block Model with Connection Probabilities 0.8, 0.2

Parameters	Model	Cross Entropy	Prob Ratio @ 100	Prob Ratio @ 500	AUC ROC
rho=1, layers=2	GCN	0.587 ± 0.010	$\mathbf{1.000} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	$\mathbf{0.697} \pm 0.002$
	LG-GNN	$\mathbf{0.563} \pm 0.001$	$\mathbf{1.000} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	0.677 ± 0.003
	PLSG-GNN	0.583 ± 0.004	$\mathbf{1.000} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	0.673 ± 0.004
rho=1, layers=4	GCN	0.693 ± 0.000	0.973 ± 0.009	0.978 ± 0.006	0.640 ± 0.029
	LG-GNN	$\mathbf{0.532} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	$\mathbf{0.697} \pm 0.001$
	PLSG-GNN	0.579 ± 0.002	$\mathbf{1.000} \pm 0.000$	0.999 ± 0.001	0.680 ± 0.002
rho=1/sqrt(n), layers=2	GCN	0.693 ± 0.000	0.388 ± 0.058	0.408 ± 0.006	$\mathbf{0.503} \pm 0.007$
	LG-GNN	$\mathbf{0.090} \pm 0.001$	$\mathbf{0.458} \pm 0.035$	$\mathbf{0.450} \pm 0.006$	0.503 ± 0.007
	PLSG-GNN	0.091 ± 0.002	0.398 ± 0.014	0.393 ± 0.010	0.491 ± 0.007
rho=1/sqrt(n), layers=4	GCN	0.693 ± 0.000	0.383 ± 0.009	0.376 ± 0.004	0.499 ± 0.007
	LG-GNN	$\mathbf{0.088} \pm 0.000$	$\mathbf{0.430} \pm 0.016$	$\mathbf{0.439} \pm 0.013$	0.505 ± 0.004
	PLSG-GNN	0.089 ± 0.001	0.388 ± 0.009	0.388 ± 0.012	$\mathbf{0.507} \pm 0.001$
rho=log(n)/n, layers=2	GCN	0.693 ± 0.000	$\mathbf{0.382} \pm 0.009$	0.376 ± 0.012	0.458 ± 0.014
	LG-GNN	0.021 ± 0.001	0.367 ± 0.004	$\mathbf{0.389} \pm 0.002$	0.517 ± 0.006
	PLSG-GNN	$\mathbf{0.019} \pm 0.000$	0.370 ± 0.027	0.374 ± 0.012	$\mathbf{0.521} \pm 0.004$
rho=log(n)/n, layers=4	GCN	0.693 ± 0.000	0.405 ± 0.007	0.380 ± 0.005	0.496 ± 0.010
	LG-GNN	0.021 ± 0.000	$\mathbf{0.470} \pm 0.007$	$\mathbf{0.410} \pm 0.003$	0.505 ± 0.017
	PLSG-GNN	$\mathbf{0.019} \pm 0.000$	0.380 ± 0.004	0.375 ± 0.007	$\mathbf{0.513} \pm 0.010$

J.2. 6SSBM (55-45)

6-community symmetric stochastic block model with edge connection probabilities 0.55 and 0.45.

Table 10. Symmetric Stochastic Block Model with Connection Probabilities 0.55, 0.45

Parameters	Model	Cross Entropy	Prob Ratio @ 100	Prob Ratio @ 500	AUC ROC
rho=1, layers=2	GCN	0.693 ± 0.000	$\mathbf{0.859} \pm 0.004$	$\mathbf{0.852} \pm 0.001$	0.500 ± 0.001
	LG-GNN	0.695 ± 0.000	0.849 ± 0.006	0.849 ± 0.003	$\mathbf{0.500} \pm 0.002$
	PLSG-GNN	$\mathbf{0.693} \pm 0.000$	0.849 ± 0.006	0.849 ± 0.003	0.500 ± 0.002
rho=1, layers=4	GCN	0.693 ± 0.000	0.847 ± 0.000	0.846 ± 0.003	0.500 ± 0.000
	LG-GNN	0.695 ± 0.000	0.848 ± 0.005	0.850 ± 0.002	0.500 ± 0.001
	PLSG-GNN	$\mathbf{0.693} \pm 0.000$	$\mathbf{0.853} \pm 0.005$	$\mathbf{0.851} \pm 0.002$	$\mathbf{0.501} \pm 0.001$
rho=1/sqrt(n), layers=2	GCN	0.693 ± 0.000	0.847 ± 0.005	0.848 ± 0.001	0.502 ± 0.002
	LG-GNN	$\mathbf{0.130} \pm 0.000$	$\mathbf{0.852} \pm 0.001$	$\mathbf{0.849} \pm 0.000$	0.503 ± 0.006
	PLSG-GNN	0.130 ± 0.001	0.850 ± 0.011	0.848 ± 0.004	$\mathbf{0.506} \pm 0.003$
rho=1/sqrt(n), layers=4	GCN	0.693 ± 0.000	0.848 ± 0.002	0.847 ± 0.002	0.502 ± 0.006
	LG-GNN	$\mathbf{0.121} \pm 0.001$	$\mathbf{0.849} \pm 0.007$	0.849 ± 0.001	$\mathbf{0.502} \pm 0.010$
	PLSG-GNN	0.131 ± 0.001	0.848 ± 0.005	$\mathbf{0.852} \pm 0.002$	0.495 ± 0.004
rho=log(n)/n, layers=2	GCN	0.693 ± 0.000	$\mathbf{0.853} \pm 0.002$	0.850 ± 0.002	0.489 ± 0.000
	LG-GNN	0.031 ± 0.000	0.845 ± 0.007	$\mathbf{0.850} \pm 0.003$	$\mathbf{0.508} \pm 0.008$
	PLSG-GNN	$\mathbf{0.030} \pm 0.001$	0.852 ± 0.005	0.850 ± 0.002	0.496 ± 0.006
rho=log(n)/n, layers=4	GCN	0.693 ± 0.000	0.851 ± 0.004	0.848 ± 0.002	0.487 ± 0.004
	LG-GNN	0.031 ± 0.001	0.847 ± 0.001	0.848 ± 0.003	0.492 ± 0.005
	PLSG-GNN	$\mathbf{0.030} \pm 0.001$	$\mathbf{0.856} \pm 0.007$	$\mathbf{0.852} \pm 0.003$	$\mathbf{0.508} \pm 0.017$

J.3. 10 SBM

10-community stochastic block model with parameter matrix P that has randomly generated entries. The diagonal entries $P_{i,i}$ are generated as $\text{Unif}(0.5, 1)$, and $P_{i,j}$ is generated as $\text{Unif}(0, \min(P_{i,i}, P_{j,j}))$. The connection matrix is

$$
\begin{pmatrix}
0.9949 & 0.3084 & 0.4553 & 0.3747 & 0.6187 & 0.0052 & 0.2626 & 0.5787 & 0.4540 & 0.6768 \\
0.3084 & 0.8309 & 0.6851 & 0.0571 & 0.5225 & 0.3345 & 0.1279 & 0.0197 & 0.7063 & 0.7795 \\
0.4553 & 0.6851 & 0.7854 & 0.1000 & 0.7726 & 0.1882 & 0.1736 & 0.6723 & 0.3278 & 0.6033 \\
0.3747 & 0.0571 & 0.1000 & 0.6160 & 0.1168 & 0.0965 & 0.0021 & 0.1856 & 0.3248 & 0.4507 \\
0.6187 & 0.5225 & 0.7726 & 0.1168 & 0.8614 & 0.5492 & 0.1098 & 0.4278 & 0.6386 & 0.1171 \\
0.0052 & 0.3345 & 0.1882 & 0.0965 & 0.5492 & 0.6623 & 0.4277 & 0.0070 & 0.1145 & 0.2878 \\
0.2626 & 0.1279 & 0.1736 & 0.0021 & 0.1098 & 0.4277 & 0.5528 & 0.2016 & 0.5466 & 0.0410 \\
0.5787 & 0.0197 & 0.6723 & 0.1856 & 0.4278 & 0.0070 & 0.2016 & 0.8805 & 0.5233 & 0.0777 \\
0.4540 & 0.7063 & 0.3278 & 0.3248 & 0.6386 & 0.1145 & 0.5466 & 0.5233 & 0.9510 & 0.4890 \\
0.6768 & 0.7795 & 0.6033 & 0.4507 & 0.1171 & 0.2878 & 0.0410 & 0.0777 & 0.4890 & 0.8526
\end{pmatrix}
$$

Table 11. 10-community SBM with randomly generated parameters

Parameters	Model	Cross Entropy	Prob Ratio @ 100	Prob Ratio @ 500	AUC ROC
rho=1, layers=2	GCN	0.599 ± 0.001	0.878 ± 0.007	$\mathbf{0.872} \pm 0.007$	$\mathbf{0.764} \pm 0.001$
	LG-GNN	0.588 ± 0.001	0.908 ± 0.008	0.867 ± 0.009	0.726 ± 0.002
	PLSG-GNN	$\mathbf{0.588} \pm 0.001$	$\mathbf{0.909} \pm 0.007$	0.867 ± 0.006	0.727 ± 0.001
rho=1, layers=4	GCN	0.677 ± 0.003	0.737 ± 0.094	0.758 ± 0.106	0.672 ± 0.011
	LG-GNN	$\mathbf{0.562} \pm 0.008$	0.868 ± 0.042	$\mathbf{0.868} \pm 0.037$	$\mathbf{0.780} \pm 0.002$
	PLSG-GNN	0.588 ± 0.001	$\mathbf{0.896} \pm 0.038$	0.858 ± 0.023	0.728 ± 0.001
rho=1/sqrt(n), layers=2	GCN	0.693 ± 0.000	0.288 ± 0.022	0.315 ± 0.008	0.505 ± 0.003
	LG-GNN	0.111 ± 0.002	0.561 ± 0.015	0.546 ± 0.023	0.515 ± 0.003
	PLSG-GNN	$\mathbf{0.110} \pm 0.001$	$\mathbf{0.610} \pm 0.008$	$\mathbf{0.577} \pm 0.005$	$\mathbf{0.520} \pm 0.005$
rho=1/sqrt(n), layers=4	GCN	0.693 ± 0.000	0.298 ± 0.048	0.312 ± 0.043	0.512 ± 0.014
	LG-GNN	$\mathbf{0.105} \pm 0.003$	0.584 ± 0.036	0.564 ± 0.011	0.516 ± 0.008
	PLSG-GNN	0.110 ± 0.002	$\mathbf{0.589} \pm 0.022$	$\mathbf{0.564} \pm 0.003$	$\mathbf{0.517} \pm 0.011$
rho=log(n)/n, layers=2	GCN	0.693 ± 0.000	0.300 ± 0.017	0.307 ± 0.014	0.478 ± 0.019
	LG-GNN	0.026 ± 0.000	0.486 ± 0.010	$\mathbf{0.494} \pm 0.005$	$\mathbf{0.525} \pm 0.009$
	PLSG-GNN	$\mathbf{0.024} \pm 0.000$	$\mathbf{0.493} \pm 0.012$	0.490 ± 0.004	0.514 ± 0.018
rho=log(n)/n, layers=4	GCN	0.693 ± 0.000	0.312 ± 0.013	0.303 ± 0.010	$\mathbf{0.517} \pm 0.019$
	LG-GNN	0.026 ± 0.002	$\mathbf{0.498} \pm 0.004$	0.494 ± 0.007	0.517 ± 0.014
	PLSG-GNN	$\mathbf{0.025} \pm 0.001$	0.496 ± 0.008	$\mathbf{0.501} \pm 0.006$	0.514 ± 0.013

J.4. Geometric Graph

Each vertex i has a latent feature X_i generated uniformly at random on $\mathbb{S}^{d-1}$, $d = 11$. Two vertices i and j are connected if $\langle X_i, X_j \rangle \geq t = 0.2$, corresponding to a connection probability ≈ 0.26. Higher sparsity is achieved by adjusting t.

Table 12. Geometric Graph with threshold 0.2 (corresponding to a connection probability of ≈ 0.26)

Parameters	Model	Cross Entropy	Prob Ratio @ 100	Prob Ratio @ 500	AUC ROC
rho=1, layers=2	GCN	0.537 ± 0.012	$\mathbf{1.000} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	0.886 ± 0.015
	LG-GNN	0.354 ± 0.005	$\mathbf{1.000} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	0.916 ± 0.004
	PLSG-GNN	$\mathbf{0.343} \pm 0.006$	$\mathbf{1.000} \pm 0.000$	0.996 ± 0.003	$\mathbf{0.918} \pm 0.005$
rho=1, layers=4	GCN	0.693 ± 0.000	0.900 ± 0.127	0.767 ± 0.075	0.759 ± 0.039
	LG-GNN	0.305 ± 0.002	$\mathbf{1.000} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	0.950 ± 0.002
	PLSG-GNN	$\mathbf{0.301} \pm 0.002$	$\mathbf{1.000} \pm 0.000$	0.999 ± 0.001	$\mathbf{0.956} \pm 0.002$
rho=1/sqrt(n), layers=2	GCN	0.693 ± 0.000	0.333 ± 0.062	0.232 ± 0.030	$\mathbf{0.848} \pm 0.007$
	LG-GNN	0.046 ± 0.002	$\mathbf{0.637} \pm 0.059$	$\mathbf{0.379} \pm 0.020$	0.822 ± 0.012
	PLSG-GNN	$\mathbf{0.046} \pm 0.002$	0.453 ± 0.076	0.293 ± 0.035	0.844 ± 0.012
rho=1/sqrt(n), layers=4	GCN	0.693 ± 0.000	0.410 ± 0.016	0.275 ± 0.021	$\mathbf{0.883} \pm 0.003$
	LG-GNN	$\mathbf{0.045} \pm 0.001$	$\mathbf{0.637} \pm 0.054$	$\mathbf{0.377} \pm 0.024$	0.827 ± 0.003
	PLSG-GNN	0.045 ± 0.001	0.530 ± 0.079	0.345 ± 0.012	0.848 ± 0.003
rho=log(n)/n, layers=2	GCN	0.693 ± 0.000	0.003 ± 0.005	0.019 ± 0.005	$\mathbf{0.624} \pm 0.018$
	LG-GNN	0.019 ± 0.001	$\mathbf{0.163} \pm 0.021$	$\mathbf{0.247} \pm 0.005$	0.607 ± 0.019
	PLSG-GNN	$\mathbf{0.019} \pm 0.000$	0.100 ± 0.024	0.097 ± 0.014	0.611 ± 0.009
rho=log(n)/n, layers=4	GCN	0.693 ± 0.000	0.003 ± 0.005	0.011 ± 0.009	0.608 ± 0.018
	LG-GNN	0.019 ± 0.001	$\mathbf{0.170} \pm 0.022$	$\mathbf{0.237} \pm 0.029$	0.609 ± 0.032
	PLSG-GNN	$\mathbf{0.018} \pm 0.001$	0.133 ± 0.019	0.154 ± 0.023	$\mathbf{0.634} \pm 0.022$

K. Out-Sample Experiments

K.1. 6SSBM (80-20)

6-community symmetric stochastic block model with connection probabilities 0.8 and 0.2.

Table 13. Symmetric Stochastic Block Model with Connection Probabilities 0.8, 0.2

Parameters	Model	Cross Entropy	Prob Ratio @ 100	Prob Ratio @ 500	AUC ROC
rho=1, layers=2	GCN	0.610 ± 0.022	$\mathbf{1.000} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	$\mathbf{0.699} \pm 0.001$
	LG-GNN	$\mathbf{0.569} \pm 0.004$	0.998 ± 0.004	0.998 ± 0.001	0.682 ± 0.002
	PLSG-GNN	0.730 ± 0.184	0.998 ± 0.004	0.999 ± 0.001	0.677 ± 0.002
rho=1, layers=4	GCN	0.693 ± 0.000	0.623 ± 0.106	0.581 ± 0.090	0.520 ± 0.004
	LG-GNN	$\mathbf{0.545} \pm 0.004$	$\mathbf{1.000} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	$\mathbf{0.698} \pm 0.001$
	PLSG-GNN	0.585 ± 0.014	$\mathbf{1.000} \pm 0.000$	0.997 ± 0.003	0.680 ± 0.001
rho=1/sqrt(n), layers=2	GCN	0.693 ± 0.000	0.448 ± 0.028	0.423 ± 0.005	0.502 ± 0.008
	LG-GNN	0.092 ± 0.002	$\mathbf{0.450} \pm 0.031$	$\mathbf{0.436} \pm 0.013$	$\mathbf{0.508} \pm 0.004$
	PLSG-GNN	$\mathbf{0.091} \pm 0.002$	0.405 ± 0.004	0.387 ± 0.003	0.503 ± 0.003
rho=1/sqrt(n), layers=4	GCN	0.693 ± 0.000	0.415 ± 0.006	0.387 ± 0.004	$\mathbf{0.506} \pm 0.006$
	LG-GNN	$\mathbf{0.088} \pm 0.002$	$\mathbf{0.460} \pm 0.011$	$\mathbf{0.436} \pm 0.008$	0.501 ± 0.014
	PLSG-GNN	0.089 ± 0.001	0.390 ± 0.019	0.382 ± 0.011	0.496 ± 0.007
rho=log(n)/n, layers=2	GCN	0.693 ± 0.000	0.382 ± 0.018	0.371 ± 0.008	0.488 ± 0.010
	LG-GNN	0.021 ± 0.001	0.355 ± 0.024	0.371 ± 0.012	$\mathbf{0.510} \pm 0.019$
	PLSG-GNN	$\mathbf{0.019} \pm 0.001$	$\mathbf{0.387} \pm 0.049$	$\mathbf{0.377} \pm 0.015$	0.491 ± 0.030
rho=log(n)/n, layers=4	GCN	0.694 ± 0.000	0.377 ± 0.022	0.379 ± 0.014	0.506 ± 0.015
	LG-GNN	0.021 ± 0.000	$\mathbf{0.460} \pm 0.054$	$\mathbf{0.384} \pm 0.011$	0.499 ± 0.015
	PLSG-GNN	$\mathbf{0.018} \pm 0.000$	0.395 ± 0.013	0.376 ± 0.021	$\mathbf{0.511} \pm 0.011$

K.2. 6SSBM (55-45)

6-community symmetric stochastic block model with edge connection probabilities 0.55 and 0.45.

Table 14. Symmetric Stochastic Block Model with Connection Probabilities 0.55, 0.45

Parameters	Model	Cross Entropy	Prob Ratio @ 100	Prob Ratio @ 500	AUC ROC
rho=1, layers=2	GCN	**0.693** $\pm$ 0.000	**0.848** $\pm$ 0.006	**0.850** $\pm$ 0.002	**0.500** $\pm$ 0.002
	LG-GNN	0.698 $\pm$ 0.002	0.845 $\pm$ 0.005	0.846 $\pm$ 0.003	0.500 $\pm$ 0.001
	PLSG-GNN	0.694 $\pm$ 0.001	0.844 $\pm$ 0.003	0.846 $\pm$ 0.003	0.500 $\pm$ 0.001
rho=1, layers=4	GCN	**0.693** $\pm$ 0.000	**0.848** $\pm$ 0.004	**0.850** $\pm$ 0.001	0.498 $\pm$ 0.001
	LG-GNN	0.702 $\pm$ 0.001	0.847 $\pm$ 0.012	0.848 $\pm$ 0.004	0.499 $\pm$ 0.001
	PLSG-GNN	0.695 $\pm$ 0.000	0.844 $\pm$ 0.011	0.850 $\pm$ 0.004	**0.499** $\pm$ 0.001
rho=1/sqrt(n), layers=2	GCN	0.693 $\pm$ 0.000	0.851 $\pm$ 0.004	0.850 $\pm$ 0.002	0.496 $\pm$ 0.004
	LG-GNN	0.131 $\pm$ 0.003	**0.859** $\pm$ 0.010	**0.851** $\pm$ 0.005	0.505 $\pm$ 0.011
	PLSG-GNN	**0.131** $\pm$ 0.002	0.844 $\pm$ 0.007	0.850 $\pm$ 0.001	**0.505** $\pm$ 0.003
rho=1/sqrt(n), layers=4	GCN	0.693 $\pm$ 0.000	0.842 $\pm$ 0.008	0.847 $\pm$ 0.001	**0.500** $\pm$ 0.009
	LG-GNN	**0.123** $\pm$ 0.001	0.850 $\pm$ 0.003	**0.849** $\pm$ 0.001	0.497 $\pm$ 0.016
	PLSG-GNN	0.130 $\pm$ 0.002	**0.852** $\pm$ 0.008	0.848 $\pm$ 0.001	0.498 $\pm$ 0.012
rho=log(n)/n, layers=2	GCN	0.693 $\pm$ 0.000	0.844 $\pm$ 0.007	**0.850** $\pm$ 0.001	0.488 $\pm$ 0.031
	LG-GNN	0.030 $\pm$ 0.000	0.842 $\pm$ 0.002	0.846 $\pm$ 0.005	0.484 $\pm$ 0.011
	PLSG-GNN	**0.029** $\pm$ 0.001	**0.851** $\pm$ 0.001	0.849 $\pm$ 0.002	**0.505** $\pm$ 0.008
rho=log(n)/n, layers=4	GCN	0.693 $\pm$ 0.000	**0.851** $\pm$ 0.004	**0.847** $\pm$ 0.002	0.493 $\pm$ 0.024
	LG-GNN	0.030 $\pm$ 0.002	0.844 $\pm$ 0.005	0.845 $\pm$ 0.003	0.488 $\pm$ 0.015
	PLSG-GNN	**0.028** $\pm$ 0.000	0.845 $\pm$ 0.009	0.846 $\pm$ 0.004	**0.504** $\pm$ 0.002

K.3. 10 SBM

10-community stochastic block model with parameter matrix P that has randomly generated entries. The diagonal entries $P_{i,i}$ are generated as $\text{Unif}(0.5, 1)$, and $P_{i,j}$ is generated as $\text{Unif}(0, \min(P_{i,i}, P_{j,j}))$. The connection matrix is

$$\begin{pmatrix}
0.9949 & 0.3084 & 0.4553 & 0.3747 & 0.6187 & 0.0052 & 0.2626 & 0.5787 & 0.4540 & 0.6768 \\
0.3084 & 0.8309 & 0.6851 & 0.0571 & 0.5225 & 0.3345 & 0.1279 & 0.0197 & 0.7063 & 0.7795 \\
0.4553 & 0.6851 & 0.7854 & 0.1000 & 0.7726 & 0.1882 & 0.1736 & 0.6723 & 0.3278 & 0.6033 \\
0.3747 & 0.0571 & 0.1000 & 0.6160 & 0.1168 & 0.0965 & 0.0021 & 0.1856 & 0.3248 & 0.4507 \\
0.6187 & 0.5225 & 0.7726 & 0.1168 & 0.8614 & 0.5492 & 0.1098 & 0.4278 & 0.6386 & 0.1171 \\
0.0052 & 0.3345 & 0.1882 & 0.0965 & 0.5492 & 0.6623 & 0.4277 & 0.0070 & 0.1145 & 0.2878 \\
0.2626 & 0.1279 & 0.1736 & 0.0021 & 0.1098 & 0.4277 & 0.5528 & 0.2016 & 0.5466 & 0.0410 \\
0.5787 & 0.0197 & 0.6723 & 0.1856 & 0.4278 & 0.0070 & 0.2016 & 0.8805 & 0.5233 & 0.0777 \\
0.4540 & 0.7063 & 0.3278 & 0.3248 & 0.6386 & 0.1145 & 0.5466 & 0.5233 & 0.9510 & 0.4890 \\
0.6768 & 0.7795 & 0.6033 & 0.4507 & 0.1171 & 0.2878 & 0.0410 & 0.0777 & 0.4890 & 0.8526
\end{pmatrix}$$

Table 15. 10-community SBM with randomly generated parameters

Parameters	Model	Cross Entropy	Prob Ratio @ 100	Prob Ratio @ 500	AUC ROC
rho=1, layers=2	GCN	0.635 ± 0.014	0.709 ± 0.125	0.726 ± 0.108	0.716 ± 0.019
	LG-GNN	0.586 ± 0.004	0.883 ± 0.016	0.843 ± 0.014	0.734 ± 0.005
	PLSG-GNN	$\mathbf{0.586} \pm 0.004$	$\mathbf{0.886} \pm 0.016$	$\mathbf{0.844} \pm 0.013$	$\mathbf{0.735} \pm 0.005$
rho=1, layers=4	GCN	0.801 ± 0.193	0.645 ± 0.025	0.633 ± 0.027	0.578 ± 0.109
	LG-GNN	$\mathbf{0.564} \pm 0.011$	0.879 ± 0.011	$\mathbf{0.886} \pm 0.004$	$\mathbf{0.786} \pm 0.002$
	PLSG-GNN	0.592 ± 0.004	$\mathbf{0.883} \pm 0.013$	0.836 ± 0.015	0.732 ± 0.001
rho=1/sqrt(n), layers=2	GCN	0.693 ± 0.000	0.344 ± 0.021	0.318 ± 0.013	0.493 ± 0.004
	LG-GNN	0.115 ± 0.002	0.580 ± 0.020	0.557 ± 0.007	0.497 ± 0.009
	PLSG-GNN	$\mathbf{0.112} \pm 0.004$	$\mathbf{0.586} \pm 0.035$	$\mathbf{0.561} \pm 0.001$	$\mathbf{0.521} \pm 0.008$
rho=1/sqrt(n), layers=4	GCN	0.693 ± 0.000	0.285 ± 0.016	0.275 ± 0.006	0.486 ± 0.006
	LG-GNN	$\mathbf{0.105} \pm 0.000$	$\mathbf{0.589} \pm 0.016$	$\mathbf{0.563} \pm 0.003$	$\mathbf{0.532} \pm 0.003$
	PLSG-GNN	0.111 ± 0.002	0.578 ± 0.013	0.544 ± 0.009	0.508 ± 0.011
rho=log(n)/n, layers=2	GCN	0.693 ± 0.000	0.312 ± 0.011	0.316 ± 0.006	0.503 ± 0.017
	LG-GNN	0.026 ± 0.000	$\mathbf{0.528} \pm 0.029$	$\mathbf{0.504} \pm 0.006$	0.506 ± 0.015
	PLSG-GNN	$\mathbf{0.023} \pm 0.002$	0.511 ± 0.017	0.501 ± 0.013	$\mathbf{0.519} \pm 0.002$
rho=log(n)/n, layers=4	GCN	0.693 ± 0.000	0.304 ± 0.027	0.304 ± 0.015	$\mathbf{0.518} \pm 0.005$
	LG-GNN	0.026 ± 0.000	0.498 ± 0.017	0.486 ± 0.015	0.500 ± 0.013
	PLSG-GNN	$\mathbf{0.024} \pm 0.000$	$\mathbf{0.546} \pm 0.018$	$\mathbf{0.505} \pm 0.018$	0.498 ± 0.016

K.4. Geometric Graph

We generate points uniformly on $\mathbb{S}^d$ and connect two points if $\langle X_i, X_j \rangle \geq t$. For the following experiment, we chose $d = 11$ and $t = 0.3$. This corresponds to a probability of about 0.15.

Table 16. Geometric Graph with threshold 0.2 (corresponding to a connection probability of ≈ 0.26)

Parameters	Model	Cross Entropy	Prob Ratio @ 100	Prob Ratio @ 500	AUC ROC
rho=1, layers=2	GCN	0.573 ± 0.015	$\mathbf{1.000} \pm 0.000$	0.996 ± 0.002	0.873 ± 0.020
	LG-GNN	0.358 ± 0.009	$\mathbf{1.000} \pm 0.000$	$\mathbf{0.999} \pm 0.001$	0.915 ± 0.007
	PLSG-GNN	$\mathbf{0.350} \pm 0.013$	0.997 ± 0.005	0.999 ± 0.002	$\mathbf{0.917} \pm 0.010$
rho=1, layers=4	GCN	0.693 ± 0.000	0.813 ± 0.021	0.733 ± 0.079	0.591 ± 0.016
	LG-GNN	0.303 ± 0.004	$\mathbf{1.000} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	0.956 ± 0.001
	PLSG-GNN	$\mathbf{0.298} \pm 0.004$	$\mathbf{1.000} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	$\mathbf{0.958} \pm 0.001$
rho=1/sqrt(n), layers=2	GCN	0.693 ± 0.000	0.333 ± 0.017	0.216 ± 0.017	0.840 ± 0.008
	LG-GNN	0.046 ± 0.003	$\mathbf{0.523} \pm 0.037$	$\mathbf{0.311} \pm 0.021$	0.818 ± 0.022
	PLSG-GNN	$\mathbf{0.045} \pm 0.002$	0.423 ± 0.054	0.244 ± 0.020	$\mathbf{0.842} \pm 0.017$
rho=1/sqrt(n), layers=4	GCN	0.693 ± 0.000	0.313 ± 0.021	0.207 ± 0.013	$\mathbf{0.848} \pm 0.021$
	LG-GNN	$\mathbf{0.045} \pm 0.001$	$\mathbf{0.570} \pm 0.016$	$\mathbf{0.311} \pm 0.010$	0.823 ± 0.010
	PLSG-GNN	0.045 ± 0.001	0.510 ± 0.014	0.289 ± 0.003	0.843 ± 0.013
rho=log(n)/n, layers=2	GCN	0.693 ± 0.000	0.003 ± 0.005	0.012 ± 0.004	0.610 ± 0.026
	LG-GNN	0.018 ± 0.000	$\mathbf{0.210} \pm 0.029$	$\mathbf{0.276} \pm 0.005$	0.616 ± 0.018
	PLSG-GNN	$\mathbf{0.018} \pm 0.001$	0.063 ± 0.037	0.123 ± 0.033	$\mathbf{0.631} \pm 0.027$
rho=log(n)/n, layers=4	GCN	0.693 ± 0.000	0.007 ± 0.009	0.035 ± 0.007	$\mathbf{0.607} \pm 0.002$
	LG-GNN	0.019 ± 0.000	$\mathbf{0.147} \pm 0.012$	$\mathbf{0.191} \pm 0.017$	0.569 ± 0.015
	PLSG-GNN	$\mathbf{0.018} \pm 0.000$	0.107 ± 0.012	0.143 ± 0.018	0.607 ± 0.010