Fitting Low-rank Models on Egocentrically Sampled Partial Networks

Ga Ming Angus Chan
University of Virginia

Tianxi Li
University of Virginia

Abstract

The statistical modeling of random networks has been widely used to uncover interaction mechanisms in complex systems and to predict unobserved links in real-world networks. In many applications, network connections are collected via egocentric sampling: a subset of nodes is sampled first, after which all links involving this subset are recorded; all other information is missing. Compared with the assumption of "uniformly missing at random", egocentrically sampled partial networks require specially designed modeling strategies. Current statistical methods are either computationally infeasible or based on intuitive designs without theoretical justification. Here, we propose an approach to fit general low-rank models for egocentrically sampled networks, which include several popular network models. This method is based on graph spectral properties and is computationally efficient for large-scale networks. It results in consistent recovery of missing subnetworks due to egocentric sampling for sparse networks. To our knowledge, this method offers the first theoretical guarantee for egocentric partial network estimation in the scope of low-rank models. We evaluate the technique on several synthetic and real-world networks and show that it delivers competitive performance in link prediction tasks.

1 INTRODUCTION

Massive network data that capture complicated dynamics and interactions in human society, the economy, ecosystems, and biology are now available (Goldenberg et al., 2010; Newman, 2018). The past 15 years have witnessed substantial progress in random network models within the statistics field. Associated efforts have provided countless model options to analyze network data with well-established theories (Bickel and Chen, 2009; Gao and Ma, 2021; Hoff et al., 2002; Rohe et al., 2011). Drawing insights from complex networks is fundamental to many scientific challenges (Kolaczyk and Csárdi, 2014; Newman, 2018), such as understanding community structures (Karrer and Newman, 2011), predicting new links (Liben-Nowell and Kleinberg, 2007; Zhao et al., 2017), and modeling peer effects in downstream tasks (Le and Li, 2020).

Missing data is a commonly encountered issue in data analysis (Little and Rubin, 2019) and plagues network problems, especially in social networks. Network connections are frequently obtained through surveys or sampling processes. One application of the network model, link prediction, is intrinsically embedded in missing data scenarios (Kumar et al., 2020; Martínez et al., 2016). Moreover, the missingness in network problems often exhibits unique patterns and calls for specialized modeling strategies.

We consider the missingness from *egocentric sampling* in this paper. Egocentric sampling is a widely used mechanism for acquiring network data (Alatas et al., 2016; Ali and Dwyer, 2009; Arnaboldi et al., 2013; Bandiera and Rasul, 2006). Under this approach, a subset of subjects is randomly sampled and all their connections are recorded. Any connections between subjects outside the sample are missing. Handcock and Gile (2010) designs a model based on the exponential random graph model (ERGM), which can handle egocentrically missing data. However, their model fitting is not computationally feasible for moderately sized networks in general, though its computation can be improved in certain settings (Krivitsky and Morris, 2017). As we will show, this class of models is too restrictive to make effective link predictions. Li et al. (2023) introduced an algorithm motivated by the CUR decomposition in computational mathematics (Drineas et al., 2006), which is computationally efficient and demonstrates strong empirical performance on link prediction tasks. However, the underlying statistical model fitted by their method is unclear. Theoretical guarantees are not available for the aforementioned methods. The only family with known theoretical model-fitting correctness is the stochastic block model family, implicitly available from Chen and Lei (2018), includ-

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

ing Chandrasekhar and Lewis (2011) as a special case.

This paper proposes a method to estimate general low-rank random network models based on egocentrically sampled partial networks. Our technique can consistently estimate network models, even for sparse networks whose average node degree is sublinear in sample size. To our knowledge, this approach is the first to feature theoretical guarantees for general low-rank models on egocentrically sampled partial networks. Our results cover many special cases, most of which previously had no known model-fitting theory; examples include the setting of Li et al. (2023), along with the random dot product model (Young and Scheinerman, 2007) and its generalization (Rubin-Delanchy et al., 2017). As an unexpected byproduct, our method provides a new insight into the method of Li et al. (2023): while Li et al. (2023) motivated their algorithm by a CUR format, their method indeed fits a general low-rank structure. Additionally, our algorithm is based on the spectral decomposition of the partial network adjacency matrix, a technique that is extremely efficient for computation on large-scale networks. We empirically demonstrate that our approach displays competitive performance in dealing with link prediction problems.

2 METHODOLOGY

2.1 Setup

Notations. We will use bold font capital letters, such as $\mathbf{A}$, to denote a matrix. $\mathbf{A}_{kk'}$ will be used to denote a submatrix of $\mathbf{A}$ (to be defined later), while the element at the ith row and jth column of $\mathbf{A}$ will be denoted by A_{ij}. Let $\mathbf{A}^T$ and $\mathbf{A}^+$ be the transpose and Moore-Penrose inverse of $\mathbf{A}$, respectively. Furthermore, $\|\mathbf{A}\|_F$ is the Frobenius norm of $\mathbf{A}$. For any positive integer n, we define $[n] = \{1, 2, \cdots, n\}$.

Let N be the total number of nodes in the full network, indexed by $i = 1, \cdots, n$. We can represent the network by its adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$, where $A_{ij} = 1$ if and only if nodes i and j are connected in the network. We consider undirected and unweighted networks for presentation simplicity. In this case, we have $\mathbf{A}^T = \mathbf{A}$. In Section 2.4, we briefly discuss how to extend our method to handle more general networks. We will study the statistical properties under the so-called "inhomogeneous Erdös-Renyi framework". Specifically, we assume there exists a probability matrix $P \in [0,1]^{N\times N}$ such that

$$A_{ij} \sim \text{Bernoulli}(P_{ij}), \quad i < j \quad \text{independently}.$$

This framework is arguably one of the most prominently applied for random network modeling. Under it, the assumed network structures are incorporated into matrix P. We assume a low-rank model for our study and define $K = \text{rank}(P) \ll N$. The family of low-rank models includes many prevalent random network models such as the stochastic block model (SBM), the degree-corrected block model (DCBM) and their generalizations (Airoldi et al., 2008; Holland et al., 1983; Jin et al., 2017; Karrer and Newman, 2011; Li et al., 2022; Sengupta and Chen, 2018) as well as the random dot product model (RDPG) and its variants (Rubin-Delanchy et al., 2017; Young and Scheinerman, 2007). Indeed, as studied by Chatterjee (2015), most popular random network models are approximately low-rank.

When the network is only partially observed from egocentric sampling, suppose there are n nodes whose connections are fully observed. Without loss of generality, we can assume that these nodes are the first n rows and columns in the adjacency matrix $\mathbf{A}$. Consider the following block partition:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix},$$

where $\mathbf{A}_{11} \in \{0,1\}^{n\times n}$, $\mathbf{A}_{12} \in \{0,1\}^{n\times(N-n)}$, $\mathbf{A}_{21} = \mathbf{A}_{12}^\intercal$, and $\mathbf{A}_{22} \in \{0,1\}^{(N-n)\times(N-n)}$. Note that $\mathbf{A}_{12} = \mathbf{A}_{21}^T$. In our problem, $\mathbf{A}_{11}$, $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$ are observed while $\mathbf{A}_{22}$ (in red) is missing. We can partition $\mathbf{P}$ in the same way

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{pmatrix}.$$

The goal of model fitting is to recover $\mathbf{P}$ from the observed blocks $\mathbf{A}_{11}$, $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$. The core challenge is to recover $\mathbf{P}_{22}$ for which no observations are available.

2.2 Low-rank Estimation (LE) Algorithm

A natural approach to the current problem is to use certain types of low-rank matrix completion (Candès and Plan, 2010; Plan and Vershynin, 2011). However, as shown in Li et al. (2023), such methods can be slow and suffer from poor accuracy due to the egocentric missing pattern. Li et al. (2023) employed an intuitive CUR decomposition based on the missing structure of $\mathbf{A}$ that efficiently computes a low-rank imputation with good empirical performance. Yet, the exact reason for this technique's success is unclear. We take a more principled approach by leveraging the low-rank structure precisely, leading to superior performance over Li et al. (2023) while also providing theoretical guarantees.

Specifically, our algorithm is motivated by a self-consistency property studied by Owen and Perry (2009) for low-rank matrices.

Lemma 2.1 (Owen and Perry (2009)) *For any $p \times q$ matrix $\mathbf{M}$ with the partition*

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{pmatrix},$$

Suppose $\mathrm{rank}(\mathbf{M}_{11}) = \mathrm{rank}(\mathbf{M})$*, we have*

$$\mathbf{M}_{11} = \mathbf{M}_{12}\mathbf{M}_{22}^{+}\mathbf{M}_{21}.$$

As per this lemma, suppose $\mathrm{rank}(\mathbf{P}_{11}) = K$. We can then exactly compute $\mathbf{P}_{22}$ by $\mathbf{P}_{11}$ and $\mathbf{P}_{12}$. In practice, when we do not observe $\mathbf{P}_{11}$ and $\mathbf{P}_{12}$, it seems logical to take $\mathbf{A}_{11}$ and $\mathbf{A}_{12}$ as the "plug-in" estimators of $\mathbf{P}_{11}$ and $\mathbf{P}_{12}$. Yet this naive approach will not work well for two reasons. First, due to the binary nature, the Moore-Penrose inverse operator on $\mathbf{A}_{11}$ is too noisy to be a good approximation. Second, directly using $\mathbf{A}_{11}$ ignores the requirement of $\mathrm{rank}(\mathbf{P}_{11}) = K$ needed in Lemma 2.1. An additional step is thus necessary to resolve the two issues simultaneously. In brief, we first take the optimal rank-K approximation of $\mathbf{A}_{11}$ as a smoothing step and then take the Moore-Penrose of the smoothed estimator. The full procedure is described in Algorithm 1.

Algorithm 1 (LE imputation for the missing network) *Given egocentrically sampled submatrices* $\mathbf{A}_{11}$*,* $\mathbf{A}_{12}$*, and the rank* K*, complete the following steps:*

1. *Take the singular value decomposition of* $\mathbf{A}_{11} = \mathbf{U}\mathbf{D}\mathbf{V}^T$*, where* $\mathbf{D}$ *is the diagonal matrix containing the singular values of* $\mathbf{A}_{11}$ *in non-increasing order, and* $\mathbf{U}$ *and* $\mathbf{V}$ *are orthogonal matrices with each column being a singular vector.*
2. *Compute* $\tilde{\mathbf{P}}_{11} = \mathbf{U}_{(K)}\mathbf{D}_{(K)}\mathbf{V}_{(K)}^T$*, in which* $\mathbf{U}_{(K)}$ *and* $\mathbf{V}_{(K)}$ *are the matrices of the first* K *columns of* $\mathbf{U}$ *and* $\mathbf{V}$*, respectively, and* $\mathbf{D}_{(K)}$ *is the diagonal matrix of only the first* K *singular values.*
3. *Set* $\hat{\mathbf{P}}_{22} = \mathbf{A}_{12}^T\tilde{\mathbf{P}}_{11}^{+}\mathbf{A}_{12}$.
4. *(Optional) If a strict constraint of all entries within* $[0, 1]$ *is desired, truncate all values of* $\hat{\mathbf{P}}_{22}$ *to* $[0, 1]$.
5. *Return* $\hat{\mathbf{P}}_{22}$.

2.3 Connections and New Insights to Li et al. (2023)

Algorithm 1 turns out to be closely connected to the subspace estimation (SE) method of Li et al. (2023). In particular, suppose we replace the SVD of Steps 1-2 in Algorithm 1 by the rank-K approximation of $\mathbf{A}_{\mathrm{obs}} = (\mathbf{A}_{11}, \mathbf{A}_{12})$ and then take the resulting first n columns as $\hat{\mathbf{P}}_{11}$. Using $\hat{\mathbf{P}}_{11}$ instead of $\tilde{\mathbf{P}}_{11}$ in other calculations would lead to the SE method. This connection is a bit surprising given that the SE method is originally designed for matrices with CUR decomposition (see their Theorem 1), a strict subset of low-rank matrices. Our current connection, therefore, reveals a new interpretation for the SE method – it is fitting general low-rank models instead of CUR-form models. As we will show, the LE method comes with theoretical guarantees for its performance. However, our analysis could not be extended to the SE method. This is because the column extraction operation of $\hat{\mathbf{P}}_{11}$ complicates the perturbation analysis. Our intuition is that such a step, though it uses more information from data (by taking $\mathbf{A}_{\mathrm{obs}}$), also requires a strong signal-to-noise ratio. Our empirical experiments support this conjecture (see Section 4). We leave the theoretical analysis of the SE estimation for future work.

2.4 Other Considerations

Model tuning by cross-validation. The LE algorithm takes the rank K as a tuning parameter. In practice, we can tune a proper K via cross-validation as indicated in Li et al. (2023). Specifically, we randomly hold out ρ proportion of the fully observed nodes, denoted by $V \subset [n]$ for validation. The remaining nodes $[n] \setminus V$ are treated as a smaller egocentrically sampled partial network, on which the LE algorithm is applied with a sequence of K. We then evaluate the link prediction performance of the LE algorithm with each K value on the partial network associated with the hold-out set V. The K value that achieves the highest area under the receiver operating characteristic (ROC) curve (i.e., AUC) is returned. This procedure is repeated multiple times, and we use the rounded average of the selected ranks as the final rank.

Full matrix recovery. Algorithm 1 only outputs the estimated $\hat{\mathbf{P}}_{22}$. This is because imputing the missing subnetwork would be the most widely needed step in analyzing partial network data, and it would be the unique contribution of our paper. Estimation of the full matrix $\mathbf{P}$, as a global model estimation, is also available. The other components $\mathbf{P}_{11}$ and $\mathbf{P}_{12}$ can be estimated together with the low-rank smoothing of the observed component $\mathbf{A}_{\mathrm{obs}}$, by taking its rank-$K$ truncated SVD. Suppose the resulting matrix is $\tilde{\mathbf{P}}_{\mathrm{obs}}$. We can take its last $N - n$ columns $\tilde{\mathbf{P}}_{\mathrm{obs},(n+1):N}$ as $\hat{\mathbf{P}}_{12}$ and the first n column as $\hat{\mathbf{P}}_{11}$. Then due to symmetric properties, we set $\hat{\mathbf{P}}_{21} = \hat{\mathbf{P}}_{12}^T$. We can also force symmetry on $\hat{\mathbf{P}}_{11}$ by taking $\frac{1}{2}(\hat{\mathbf{P}}_{11} + \hat{\mathbf{P}}_{11})^T$ as the estimator of $\mathbf{P}_{11}$. Similar to Step 4 of Algorithm 1, we can truncate the values of all estimators to $[0, 1]$ as an optional step. Our theoretical properties hold in both cases. Combined with the output $\hat{\mathbf{P}}_{22}$ of Algorithm 1, the final estimation of the full matrix $\mathbf{P}$ is given by

$$\hat{\mathbf{P}} = \begin{pmatrix} \hat{\mathbf{P}}_{11} & \hat{\mathbf{P}}_{12} \\ \hat{\mathbf{P}}_{21} & \hat{\mathbf{P}}_{22} \end{pmatrix}. \tag{1}$$

As we will show in the next section, $\hat{\mathbf{P}}$ would be a consistent estimator of $\mathbf{P}$. However, in practice, $\mathbf{P}$ may not be an exactly rank-K matrix. We do not think it is necessary to force the rank of $\hat{\mathbf{P}}$ in most applications. However, if needed, one can force the rank value with the post-processing of SVD truncation.

Extension to directed and weighted networks. While we focus on undirected and unweighted networks, the exten-

sion to directed and weighted networks can be naturally made. Notice that Lemma 2.1 holds for general asymmetric matrices. $\mathbf{A}$ and $\mathbf{P}$ are no longer symmetric for a directed network. And the inhomogeneous Erdös-Renyi model is still well defined, with all entries A_{ij} being independently generated according to P_{ij}. Algorithm 1 can be exactly applied. Our theoretical results can be similarly derived.

3 THEORETICAL PROPERTIES OF THE LE ESTIMATOR

As mentioned, one major contribution of our work is the theoretical guarantee for general low-rank network model recovery from the partial network, which is not previously available. We now proceed to introduce our theoretical results. We first introduce additional notations and our regularity assumptions. Define $p^* = \max_{ij} P_{ij}$. For any matrix $\mathbf{M}$, denote its kth largest singular value by $\sigma_k(\mathbf{M})$. For two sequences a_n and b_n, we will write $a_n \lesssim b_n$ if there exists a constant C such that $a_n \leq Cb_n$. Correspondingly, we write $b_n \gtrsim a_n$ if $a_n \lesssim b_n$. Moreover, if $a_n \lesssim b_n$ and $b_n \lesssim a_n$, we write $a_n \sim b_n$.

We make the following assumptions.

Assumption A1 (Low-rank recoverable) *The rank of the model satisfies* $\mathrm{rank}(\mathbf{P}_{11}) = \mathrm{rank}(\mathbf{P}) = K$. *In particular,* $K \lesssim \sqrt{\log n}$.

Assumption A2 (Well-conditioned model)

$$np^* \sim \sigma_K(\mathbf{P}_{11}) \leq \sigma_1(\mathbf{P}_{11}) \sim np^*$$
$$Np^* \sim \sigma_K(\mathbf{P}) \leq \sigma_1(\mathbf{P}) \sim Np^*$$

Assumption A1 is strictly needed to ensure the validity of the low-rank recovery on the population matrix $\mathbf{P}$ by Lemma 2.1. In contrast, assumption A2 can be relaxed for better generality. However, we keep it in the current form for conciseness and interpretability of our error bound.

Remark 1 *Both A1 and A2 are indeed motivated by the general sparse graphon model of Bickel and Chen (2009). The sparse graphon framework is arguably the most popular way to generate the* $\mathbf{P}$ *in the inhomogeneous Erdös-Renyi model (Gao and Ma, 2021; Klopp et al., 2017; Lin et al., 2020; Mukherjee et al., 2017). In particular, it is not difficult to see that under the sparse graphon mechanism, if the true graphon does generate a low-rank* $\mathbf{P}$ *and the observed nodes are uniformly sampled in the egocentric sampling, both A1 and A2 hold almost surely. In particular, the assumptions hold for the SBM, RDPG and their variants.*

Remark 2 *Meanwhile, we also want to emphasize that we do not assume that observed subjects form a random sample out of all nodes. So our theory is still applicable in situations of nonuniform sampling or even informative sampling, as long as A2 holds. This property is crucial for real-world applications. We will provide evidence supporting this claim in Section 4.*

Theorem 3.1 *Under assumptions A1 and A2, further assume that* $np^* \gtrsim \log(n)$ *and* $K \lesssim \sqrt{\log n}$. *If* $\hat{\mathbf{P}}_{22}$ *is the estimator of* $\mathbf{P}_{22}$ *produced by Algorithm 1, we have*

$$\|\hat{\mathbf{P}}_{22} - \mathbf{P}_{22}\|_F \lesssim \sqrt{K}\left(\left(\frac{N}{n}\right)^{3/2}\sqrt{KNp^*} + \frac{N^2}{n^2}\log n\right) \tag{2}$$

with probability tending to 1.

The requirement for K can be further relaxed with a more complicated error bound which we will not pursue. For illustration, consider the following two special cases

1. Suppose n and N are in the same order ($n \sim N$), we can see that the error bound on the missing network is in the order of $K\sqrt{np^*} + \sqrt{K}\log n$. Since $\|\mathbf{P}_{22}\|_F \sim n\sqrt{p^*}$, we know that $\|\hat{\mathbf{P}}_{22} - \mathbf{P}_{22}\|_F / \|\mathbf{P}_{22}\|_F \to 0$ and the estimation consistency is guaranteed under the current assumptions.
2. Suppose K is bounded and $np^* = \log^2 n$. Then the error bound is the order of $\frac{N^2}{n^2}\log n$. So $\|\hat{\mathbf{P}}_{22} - \mathbf{P}_{22}\|_F / \|\mathbf{P}_{22}\|_F \to 0$ as long as $n \gg N^{4/5}$. Therefore, though our method allows the sampling proportion to be vanishing, the decaying rate has to be slow.

The error bound for the full matrix recovery is a straightforward extension of Theorem 3.1 as follows.

Corollary 1 *Under the assumptions of Theorem 3.1, for the full matrix estimator* $\hat{\mathbf{P}}$ *defined in* (1)*, we have*

$$\|\hat{\mathbf{P}} - \mathbf{P}\|_F \lesssim \sqrt{K}\left(\left(\frac{N}{n}\right)^{3/2}\sqrt{KNp^*} + \frac{N^2}{n^2}\log n\right)$$

with high probability.

Corollary 1 still has the same order error bound as Theorem 3.1. This indicates that the major estimation error for the full matrix $\mathbf{P}$ is still on the unobserved component $\mathbf{P}_{22}$. Because the components $\mathbf{P}_{11}$ and $\mathbf{P}_{12}$ are observed (with noises) and the model is low-rank. The recovery of these terms would be easy. The imputation of the unobserved subnetwork is the most challenging component of the problem, and it is our emphasized contribution. Similar to the previous illustration, the corollary indicates that if $N \sim n$, for example, the full matrix estimation is consistent.

4 SYNTHETIC EXPERIMENTS

In this section, we evaluate the proposed LE method in link prediction tasks to predict the unobserved subnetwork under several synthetic network models. Our implementation

is based on R. We include all the egocentric-suitable methods mentioned in Section 1 as benchmarks as follows.

1. Subspace estimation method (SE) (Li et al., 2023). This method is based on the CUR decomposition of a matrix, as a special case of the low-rank model. The approach is effective for link prediction with computational scalability. It does not come with theory.

2. LE+. This method is computed as the average of the LE and SE estimations. This simple hybrid will be adaptive to better one of the two methods according to individual scenarios.

3. The neighborhood smoothing method (NS) for graphon estimation (Zhang et al., 2017). This method was originally studied under piecewise-smooth dense graphon models (Bickel and Chen, 2009), and we adapt it according to the egocentric sampling. The piecewise-smooth graphon model family overlaps with some low-rank models, but the two model classes are not nested. Its theory in egocentrically sampled and sparse networks is unclear. The computation is polynomial in N but not well suited to networks with more than a few thousand nodes. We use the Python implementation provided from Li et al. (2023) for it.

4. ERGM (Handcock and Gile, 2010). We use the model version with egocentric sampling and geometrically weighted edgewise shared partnerships introduced by Hunter (2007). This ERGM configuration works best in our evaluation. The model is implemented in the R package *ergm* (Handcock et al., 2019). Note that the ERGM does not follow the inhomogeneous Erdös-Renyi framework in general. This method is the slowest among all the benchmarks, and it does not come with a known theoretical guarantee.

5. SBM model fitting. This approach is adapted to egocentric sampling given the model fitting strategy of Chen and Lei (2018) based on spectral clustering (Lei and Rinaldo, 2015). It is implemented in the R package *randnet* (Li et al., 2021). The model fitting is known to be consistent if the true model is the SBM. This model has a high computation speed similar to that of LE (the proposed method) and SE.

6. DCBM model fitting. Similar to the SBM, the model fitting strategy of Chen and Lei (2018) is applied based on the R package *randnet*.

4.1 Experiment Setup

Following Li et al. (2023), synthetic networks with dimension $N = 500$ are generated under four network models.

1. The SBM: In this model, we first randomly assign N nodes to K groups or communities. Let $\mathbf{Z} \in \{0,1\}^{N\times K}$ be a community label matrix, where $Z_{ik} = 1$ if and only if the node i belongs to the kth community. With a matrix symmetric matrix $\mathbf{B} \in [0,1]^{K\times K}$ as the "community-connection" matrix, the probability matrix is $\mathbf{P} = \mathbf{Z}\mathbf{B}\mathbf{Z}^T$. We set $K = 5$ in our experiments. This model represents an important benchmark scenario because it is strictly low-rank ($K = 5$) and we do have a model-based method to fit it consistently via the SBM fitting. Therefore, LE's performance under this model gives a measure of its adaptivity to specific models.

2. The DCBM: The out-in ratio (β), i.e., the ratio of between-block to within-block edges, is set to be the ratio of the average between-block edge probability to within-block edge probability as in the "community-connection" matrix B in the SBM. Similarly, the number of communities $K = 5$. The degree parameters follow the power-law distribution with $\alpha = 0.1$ and lower bound 1.

3. The RDPG (product model). Under this model, we first generate $Z_i \in [0,1]^K$ for each node $i \in [N]$. Each coordinate of Z_i is generated independently from $\mathrm{Beta}(0.5, 1)$, following the procedure of Athreya et al. (2017). The connection probability between nodes i and j is given by $P_{ij} = Z_i^\top Z_j$. This model provides a more general low-rank scenario compared to the SBM, for which no known consistency guarantees were previously available.

4. The latent space model (distance model). For each node $i \in [N]$, we generate its latent vector $Z_i \sim N(0, I_5)$. Then the connection probability between nodes i and j is set by $P_{ij} = [1+\exp(\|Z_i - Z_j\|)]^{-1}$ where $\|Z_i - Z_j\|$ is the Euclidean distance between Z_i and Z_j. The $\mathbf{P}$ of the distance model is not low-rank. This model thus serves to test the potential to approximate full-rank models.

The network generated from the above models may not be sparse. Therefore, after deriving the probability matrix $\mathbf{P}$, we further scale it to control the average expected degree of the generated networks. We focus on networks in our experiments with expected degrees of 20 and 50.

We evaluate the methods based on the mean squared error (MSE) on the unobserved probability matrix

$$\mathrm{MSE} = \|\hat{\mathbf{P}}_{22} - \mathbf{P}_{22}\|_F^2/(N-n)^2.$$

Additionally, the performance is also measured by the timing of model fitting, reflecting the computational efficiency. Since most of the methods involve tuning procedures for which the configuration and preference can vary across users, we do not include the tuning procedure in timing evaluation and only focus on the model-fitting procedure.

We are also aware that the different methods are implemented differently: NS is implemented in Python; LE, SE, SBM and DCBM are implemented in R; the major component of ERGM fitting is implemented in C and called by R functions. Overall, we believe the comparison of NS, SE, LE, and SBM is fair, while the ERGM, if implemented similarly, would be even more inferior in speed comparison. All experiments are independently repeated 100 times.

4.2 Evaluation under Missing Completely At Random

For egocentric missingness with missing-completely-at-random (MCAR), we randomly sample $\rho = \{0.1, 0.2, 0.5, 0.9\}$ of the nodes as observed while the rest as missing. We will investigate the performance of all methods in configurations of different synthetic models, expected degrees and ρ.

4.2.1 Performance Evaluation

Link Prediction Accuracy. Table 1 displays the prediction errors under the SBM, DCBM, product, and distance models, respectively, with various levels of sampling proportions and sparsity levels.

Under the SBM model, the SBM model-fitting is generally the best method since it fits the correct model. LE, SE, LE+ and NS all have similar performances and are inferior to the SBM. LE is generally more accurate than SE, and its advantage increases with the sampling proportion ρ. The only case when SE is better is when ρ is very low (0.1) while the network is dense (50). This coincides with our theoretical conjecture. LE+ can be better or similar to the better one of the two.

Under the DCBM model, LE+ is generally the best method, while LE and SE have similar performance. The comparison between LE, SE and LE+ remains similar. A surprising result is that DCBM model fitting does not produce the best result and is inferior to SBM model fitting under low-degree sparse networks.

Under the product and distance models, LE and LE+ are essentially the best ones. These experiments demonstrate the effectiveness of the proposed estimator to fit general low-rank structures and even beyond that.

One interesting fact may be that SBM performs well under sparse networks (Deg.$= 20$) regardless of the generating model. This may be explained by its parsimony, reducing the variance in estimation when little information is available from the observed data.

Timing Comparison. Table 2 summarizes the average computing time of all the methods under evaluation. The ERGM is by far the slowest and generally not applicable for networks with over a few hundred nodes. NS is feasible but is still slower than the other methods. The comparison between LE, SE, SBM and DCBM can be different across network configurations, but overall, they are all comparably efficient in speed.

Table 1: MSE of Link Prediction Performance on Synthetic Networks (10^{-3}).

Model	(ρ,Deg.)	LE	SE	LE+	NS	ERGM	SBM	DCBM
SBM	(0.1,20)	4.81 (0.139)	5.33 (0.228)	**4.19** (0.101)	7.79 (0.098)	9.15 (0.056)	6.43 (0.467)	8.16 (0.37)
	(0.1,50)	14.4 (0.246)	12.5 (0.236)	11.2 (0.149)	17.7 (0.285)	43.1 (0.1)	**6.71** (0.161)	13.8 (0.168)
	(0.2,20)	2.87 (0.043)	2.88 (0.049)	2.44 (0.031)	4.51 (0.037)	9.23 (0.036)	**2.41** (0.043)	3.26 (0.046)
	(0.2,50)	7.44 (0.105)	7.6 (0.087)	6.96 (0.069)	10.2 (0.165)	42.8 (0.069)	**3.03** (0.054)	8.25 (0.133)
	(0.5,20)	1.35 (0.016)	1.52 (0.017)	1.35 (0.014)	2.04 (0.021)	9.8 (0.026)	**0.763** (0.02)	1.96 (0.021)
	(0.5,50)	3.54 (0.027)	4.02 (0.05)	3.64 (0.035)	5.1 (0.087)	43.4 (0.129)	**0.823** (0.039)	3.4 (0.077)
	(0.9,20)	1.05 (0.018)	1.43 (0.041)	1.27 (0.033)	1.24 (0.021)	10.5 (0.108)	**0.548** (0.027)	1.81 (0.047)
	(0.9,50)	2.18 (0.026)	3.82 (0.08)	2.87 (0.044)	1.78 (0.035)	45.5 (0.561)	**0.584** (0.043)	2.6 (0.078)
DCBM	(0.1,20)	7.2 (0.14)	8.38 (0.231)	**6.21** (0.138)	8.46 (0.084)	10.2 (0.051)	9.84 (0.67)	10.4 (0.495)
	(0.1,50)	20.4 (0.229)	16.4 (0.208)	14.7 (0.131)	21.9 (0.172)	50.2 (0.112)	**13.6** (0.183)	16.3 (0.225)
	(0.2,20)	4.37 (0.043)	4.01 (0.045)	**3.43** (0.028)	5.06 (0.04)	10.3 (0.034)	3.67 (0.065)	4.23 (0.043)
	(0.2,50)	9.12 (0.139)	8.67 (0.076)	8.1 (0.079)	14.6 (0.135)	50.3 (0.055)	**7.51** (0.183)	7.91 (0.152)
	(0.5,20)	**1.68** (0.022)	1.98 (0.024)	1.69 (0.018)	2.76 (0.023)	11.1 (0.032)	1.43 (0.032)	1.98 (0.027)
	(0.5,50)	3.39 (0.028)	3.69 (0.035)	3.46 (0.036)	8.93 (0.107)	50.6 (0.175)	5.95 (0.093)	**3.08** (0.148)
	(0.9,20)	**1.04** (0.023)	1.44 (0.043)	1.23 (0.028)	1.93 (0.051)	13.1 (0.16)	1.21 (0.06)	1.43 (0.07)
	(0.9,50)	**2.25** (0.052)	2.65 (0.061)	2.53 (0.051)	4.68 (0.196)	54.9 (0.72)	7.73 (0.356)	2.82 (0.098)
product	(0.1,20)	3 (0.136)	3.16 (0.125)	**2.37** (0.085)	6.61 (0.079)	7.15 (0.058)	8.48 (0.646)	6.96 (0.297)
	(0.1,50)	5.63 (0.049)	5.65 (0.084)	**5.12** (0.049)	13.6 (0.126)	33.1 (0.181)	6.49 (0.213)	6.75 (0.104)
	(0.2,20)	1.17 (0.011)	1.23 (0.019)	**1.06** (0.011)	3.44 (0.025)	7.37 (0.041)	1.77 (0.105)	2.03 (0.049)
	(0.2,50)	3.33 (0.02)	3.39 (0.033)	**3.24** (0.024)	7.84 (0.046)	33.1 (0.103)	4.7 (0.029)	4.16 (0.031)
	(0.5,20)	**0.582** (0.005)	0.67 (0.01)	0.592 (0.006)	1.54 (0.01)	7.44 (0.029)	0.855 (0.01)	0.968 (0.013)
	(0.5,50)	**1.99** (0.013)	2.09 (0.021)	**2** (0.017)	4.74 (0.038)	33.2 (0.065)	3.86 (0.021)	3.06 (0.023)
	(0.9,20)	**0.413** (0.007)	0.665 (0.021)	0.478 (0.008)	1.09 (0.014)	7.44 (0.072)	0.988 (0.027)	1.03 (0.033)
	(0.9,50)	**1.6** (0.024)	2 (0.041)	1.73 (0.026)	2.9 (0.042)	33.8 (0.275)	3.82 (0.062)	3.07 (0.055)
distance	(0.1,20)	3.85 (0.137)	4.53 (0.213)	**3.38** (0.124)	7.2 (0.086)	7.82 (0.061)	9.01 (0.651)	8.32 (0.378)
	(0.1,50)	9.87 (0.078)	9.62 (0.092)	**9.08** (0.06)	17.5 (0.124)	36.4 (0.15)	10.1 (0.203)	10.7 (0.124)
	(0.2,20)	1.87 (0.017)	1.86 (0.02)	**1.73** (0.014)	4.19 (0.03)	7.86 (0.041)	2.22 (0.05)	2.61 (0.035)
	(0.2,50)	7.49 (0.028)	7.45 (0.045)	**7.18** (0.026)	11.6 (0.051)	36.7 (0.093)	8.47 (0.03)	8.3 (0.044)
	(0.5,20)	1.34 (0.005)	1.38 (0.008)	**1.31** (0.006)	2.26 (0.011)	8.05 (0.027)	1.53 (0.01)	1.73 (0.012)
	(0.5,50)	6.22 (0.026)	6.34 (0.026)	**5.8** (0.02)	8.72 (0.041)	37.7 (0.054)	8.17 (0.031)	7.35 (0.029)
	(0.9,20)	**2.13** (0.011)	2.23 (0.02)	2.18 (0.016)	2.78 (0.016)	9.69 (0.07)	2.53 (0.017)	2.61 (0.017)
	(0.9,50)	9.49 (0.049)	9.5 (0.054)	**9.1** (0.047)	12.7 (0.051)	46.1 (0.266)	13.4 (0.073)	12 (0.062)

Summary. Overall, the LE method renders competitive accuracy in all settings even when the model is full-rank (but can be approximated by low-rank structures). The closely related SE method is inferior to LE, except in low-sampling-high-density cases. LE+ can generally achieve adaptive performance by combining LE and SE.

Table 2: Timing Comparison between Benchmark Methods on Synthetic Networks (In Milliseconds).

Model	(ρ,Deg.)	LE	SE	LE+	NS	ERGM	SBM	DCBM
SBM	(0.1,20)	7.09 (2.5)	9.39 (2.97)	9.32 (1.71)	134 (0.176)	27000 (147)	36.9 (1.1)	46 (2)
	(0.1,50)	3.78 (0.108)	7.55 (2.4)	11.2 (2.42)	133 (0.0811)	35000 (248)	13.2 (0.387)	31.3 (0.744)
	(0.2,20)	10.9 (3.07)	7.55 (1.63)	12 (0.144)	150 (0.0346)	21400 (127)	37.1 (1.1)	49.4 (2.09)
	(0.2,50)	9.66 (2.62)	11.8 (3.08)	15.2 (1.65)	151 (0.0678)	30100 (183)	13.6 (1.57)	31.7 (0.913)
	(0.5,20)	24.3 (2.9)	28.6 (4.05)	46.9 (2.24)	197 (0.0479)	11400 (114)	19.1 (1.52)	52.2 (1.14)
	(0.5,50)	26.8 (2.98)	25.6 (3.08)	50.8 (1.99)	197 (0.0921)	21400 (168)	19.8 (2.19)	25.6 (1.63)
	(0.9,20)	78.2 (3.38)	72.6 (2.88)	197 (3.84)	307 (0.706)	8080 (88.4)	25.2 (1.59)	79.2 (1.44)
	(0.9,50)	72.6 (1.76)	79.2 (3.67)	212 (4.77)	311 (0.109)	18700 (193)	28.8 (0.69)	36.5 (0.915)
DCBM	(0.1,20)	7.56 (2.53)	4.75 (0.0665)	11.9 (2.63)	139 (0.261)	26700 (133)	46.5 (2.04)	48.9 (0.981)
	(0.1,50)	10.1 (3.31)	5.81 (1.73)	9.23 (1.77)	138 (0.109)	35600 (216)	18.1 (0.385)	34.5 (0.811)
	(0.2,20)	8.16 (1.73)	6.71 (0.141)	18.1 (2.92)	148 (0.219)	21200 (129)	43.7 (1.52)	52.1 (2.2)
	(0.2,50)	5.75 (0.115)	7.57 (1.76)	16.6 (2.9)	147 (0.166)	30500 (207)	17.4 (0.281)	36.6 (1.68)
	(0.5,20)	20.9 (2.78)	24.8 (3.23)	44.6 (1.67)	185 (0.257)	11200 (121)	21.2 (0.3)	49.5 (1.96)
	(0.5,50)	20.9 (2.87)	18 (0.136)	51.2 (3.81)	185 (0.202)	21600 (162)	26 (1.52)	27.8 (0.732)
	(0.9,20)	67.3 (2.44)	68.9 (2.52)	185 (3)	251 (0.485)	8180 (91.3)	22 (1.58)	68.5 (2.99)
	(0.9,50)	68.9 (1.31)	72.5 (2.97)	196 (4.17)	249 (0.254)	18900 (175)	64.4 (1.82)	43.4 (2.38)
product	(0.1,20)	8.57 (3.03)	5.33 (1.57)	7.22 (0.0859)	136 (0.158)	26900 (143)	48.6 (1.07)	55.8 (2.09)
	(0.1,50)	3.25 (0.0948)	8.62 (2.98)	12.1 (2.94)	137 (0.178)	35700 (264)	37.2 (0.875)	42.7 (3.35)
	(0.2,20)	10.5 (3.08)	7.26 (1.68)	11.3 (0.146)	147 (0.186)	21600 (127)	54.6 (1.39)	59.9 (2.33)
	(0.2,50)	6.7 (1.71)	8.75 (2.43)	14.6 (2.55)	147 (0.12)	30800 (227)	54 (0.635)	42.2 (0.906)
	(0.5,20)	23.1 (2.96)	26.2 (3.81)	44.9 (2.36)	180 (0.431)	11100 (108)	63.5 (1.77)	74.2 (2.31)
	(0.5,50)	19.5 (1.74)	17.9 (1.65)	44.9 (2.38)	180 (0.303)	21900 (172)	80.8 (0.668)	69.2 (2.95)
	(0.9,20)	66.6 (2.65)	65.8 (2.81)	188 (3.16)	247 (0.393)	7690 (94.3)	102 (1.98)	98.6 (2.67)
	(0.9,50)	66.3 (2.28)	64.2 (2.1)	187 (2.96)	246 (0.318)	19200 (170)	93.6 (0.613)	76.6 (2.34)
distance	(0.1,20)	4.85 (1.47)	5.9 (1.74)	7.03 (0.164)	138 (0.249)	27000 (133)	55 (2.12)	55.1 (0.827)
	(0.1,50)	3.22 (0.0602)	8.28 (2.66)	7.65 (1.49)	137 (0.095)	35600 (226)	42.8 (0.901)	36.7 (0.814)
	(0.2,20)	5.14 (0.0914)	7.49 (1.68)	16.2 (2.93)	147 (0.229)	21500 (142)	64.9 (2.75)	58 (2.06)
	(0.2,50)	5.05 (0.0567)	5.45 (0.0868)	16.2 (3.1)	148 (0.147)	30600 (240)	61.4 (0.788)	44.3 (0.976)
	(0.5,20)	18 (0.129)	18.7 (1.61)	46.4 (3.67)	183 (0.37)	11100 (103)	69 (2.52)	68.5 (1.26)
	(0.5,50)	22.5 (2.56)	20.4 (2.27)	45.1 (3.47)	183 (0.295)	21800 (166)	91.1 (0.851)	67.9 (2.07)
	(0.9,20)	68.3 (2.65)	67.1 (0.687)	164 (5.2)	250 (0.561)	7700 (91.2)	110 (2.73)	96.5 (1.53)
	(0.9,50)	68.9 (2.47)	74.4 (3.76)	158 (2.9)	246 (0.283)	19100 (174)	108 (0.75)	72.6 (1.94)

4.3 Evaluation under Missing Not At Random

All methods are also evaluated under the missing-not-at-random (MNAR) scenario, where the sampling probability of a node depends on its degree (in $\mathbf{A}$). Two settings are evaluated: the positive setting ($+$), where nodes with higher degrees are more likely to be sampled; and the negative setting ($-$), where nodes with lower degrees are more likely to be sampled. Based on the adjacency matrix A, the nodes are divided into three groups by the descending order of their node degrees by proportions $33\%, 34\%$ and 33%. The three groups are sampled at $\delta_+ = \{1.5, 1, 0.5\}$ or $\delta_- = \{0.5, 1, 1.5\}$ respectively of $\rho = \{0.2, 0.5\}$. For example, the first group under $\rho = 0.2$ and the positive setting is sampled at $\tilde{\rho} = \delta_+ \rho = 1.5 \times 0.2 = 0.3$.

Tables 3–4 display the MSE of the prediction error under the four generating models, with varying sampling proportions and sparsity levels for both positive and negative settings. The result is consistent with the findings under uniform sampling. Under the SBM model, SBM still has the best general performance. Under other settings, LE+ is generally the best method while LE and SE have comparable performance. Timing comparison is not included as it is similar to that of uniform sampling.

Table 3: MSE of Link Prediction Performance on Synthetic Networks, Positive MNAR Setting (10^{-3}).

Model	(ρ,Deg.)	LE	SE	LE+	NS	ERGM	SBM	DCBM
SBM	(0.2,20)	2.05 (0.023)	2.14 (0.023)	1.89 (0.016)	3.48 (0.026)	10.4 (0.032)	**1.72** (0.026)	2.54 (0.03)
	(0.2,50)	6.41 (0.083)	6.11 (0.061)	5.85 (0.048)	6.68 (0.062)	41.7 (0.13)	**2.49** (0.035)	6.37 (0.117)
	(0.5,20)	1.12 (0.014)	1.06 (0.01)	1.04 (0.009)	1.37 (0.008)	9.66 (0.033)	**0.549** (0.009)	1.14 (0.012)
	(0.5,50)	3.88 (0.015)	4.19 (0.045)	3.78 (0.023)	3.82 (0.028)	37 (0.082)	**1.26** (0.038)	5.94 (0.12)
DCBM	(0.2,20)	3.6 (0.036)	3.39 (0.038)	**2.9** (0.025)	4.4 (0.024)	11.4 (0.032)	2.97 (0.032)	3.63 (0.045)
	(0.2,50)	7.26 (0.103)	7.2 (0.055)	6.92 (0.048)	12 (0.087)	51.3 (0.069)	6.61 (0.123)	**6.24** (0.131)
	(0.5,20)	1.12 (0.014)	1.26 (0.012)	1.13 (0.01)	2 (0.016)	11.1 (0.039)	**1.07** (0.019)	1.16 (0.02)
	(0.5,50)	2.4 (0.018)	2.6 (0.024)	2.45 (0.021)	6.53 (0.068)	44.9 (0.105)	4.5 (0.049)	**2.17** (0.125)
product	(0.2,20)	0.93 (0.006)	0.961 (0.012)	**0.87** (0.008)	2.91 (0.021)	8.94 (0.032)	1.39 (0.05)	1.46 (0.023)
	(0.2,50)	2.8 (0.013)	2.83 (0.026)	**2.71** (0.015)	6.33 (0.021)	36.9 (0.073)	4.77 (0.033)	3.9 (0.037)
	(0.5,20)	**0.392** (0.003)	0.43 (0.005)	**0.392** (0.003)	1 (0.005)	8.39 (0.028)	0.702 (0.006)	0.58 (0.006)
	(0.5,50)	**1.4** (0.007)	1.46 (0.013)	**1.4** (0.008)	3.19 (0.019)	35.2 (0.066)	3.4 (0.022)	2.16 (0.015)
distance	(0.2,20)	1.58 (0.008)	1.57 (0.011)	**1.51** (0.007)	3.52 (0.022)	9.35 (0.028)	1.93 (0.048)	2.07 (0.022)
	(0.2,50)	6.8 (0.02)	6.74 (0.026)	**6.54** (0.017)	9.92 (0.02)	40.2 (0.073)	8.26 (0.026)	7.7 (0.044)
	(0.5,20)	**1.13** (0.003)	1.15 (0.006)	**1.12** (0.003)	1.73 (0.004)	8.94 (0.029)	1.31 (0.006)	1.3 (0.006)
	(0.5,50)	5.5 (0.017)	5.44 (0.016)	**5.17** (0.016)	7.35 (0.02)	38.7 (0.067)	7.08 (0.019)	6.15 (0.015)

Summary. Overall, no extreme change in performance is observed across the tested missingness mechanisms. This indicates that non-uniform sampling does not severely affect our method LE and any benchmark methods, supporting our theoretical claim that our method is applicable for non-uniform sampling as long as A1 holds.

5 LINK PREDICTION IN REAL-WORLD NETWORKS

In this section, we evaluate our approach to link prediction on real-world networks. We consider three examples: two social networks and one airline traffic network. We wish to demonstrate the merits and limitations of our approach. In the first two examples, our method outperforms other benchmarks, indicating that the low-rank model assumption is reasonable. In the third example, the NS method works better. As such, the network may not be

Table 4: MSE of Link Prediction Performance on Synthetic Networks, Negative MNAR Setting (10^{-3}).

Model	(ρ,Deg.)	LE	SE	LE+	NS	ERGM	SBM	DCBM
SBM	(0.2,20)	4.13 (0.039)	4.1 (0.083)	**3.37** (0.044)	5.59 (0.03)	8.16 (0.02)	3.78 (0.109)	4.26 (0.069)
	(0.2,50)	11.3 (0.198)	9.5 (0.095)	9.1 (0.093)	18 (0.139)	42.5 (0.056)	**4.85** (0.121)	9.89 (0.158)
	(0.5,20)	2.55 (0.03)	2.75 (0.041)	2.33 (0.027)	4.39 (0.036)	10.1 (0.025)	**1.75** (0.048)	3.08 (0.028)
	(0.5,50)	5.49 (0.057)	6.23 (0.069)	5.59 (0.048)	18.8 (0.183)	53.9 (0.089)	**0.638** (0.047)	5.46 (0.124)
DCBM	(0.2,20)	5.02 (0.04)	4.72 (0.052)	**3.95** (0.029)	5.85 (0.03)	9.33 (0.022)	4.23 (0.072)	5.01 (0.071)
	(0.2,50)	11.2 (0.144)	9.95 (0.093)	9.41 (0.075)	18 (0.1)	49.2 (0.054)	**9.35** (0.199)	9.64 (0.214)
	(0.5,20)	2.49 (0.027)	2.87 (0.029)	2.38 (0.02)	4.02 (0.026)	11.3 (0.032)	**2.21** (0.039)	3.04 (0.048)
	(0.5,50)	**4.7** (0.034)	5.32 (0.05)	4.79 (0.035)	15.4 (0.118)	57.9 (0.163)	9.66 (0.106)	5.13 (0.27)
product	(0.2,20)	1.5 (0.012)	1.54 (0.025)	**1.35** (0.017)	3.92 (0.025)	5.82 (0.022)	3.24 (0.217)	2.61 (0.064)
	(0.2,50)	4.2 (0.022)	4.03 (0.03)	**3.85** (0.024)	9.84 (0.034)	29 (0.057)	5.14 (0.051)	4.89 (0.046)
	(0.5,20)	0.866 (0.006)	1 (0.013)	**0.861** (0.008)	2.24 (0.011)	6.46 (0.021)	1.33 (0.029)	1.49 (0.015)
	(0.5,50)	2.78 (0.015)	2.88 (0.025)	**2.76** (0.018)	8.58 (0.033)	32.1 (0.056)	5.32 (0.036)	4.16 (0.03)
distance	(0.2,20)	2.3 (0.023)	2.37 (0.044)	**2.06** (0.018)	4.53 (0.026)	6.48 (0.021)	3.31 (0.146)	3.3 (0.068)
	(0.2,50)	8.33 (0.029)	8.27 (0.047)	**7.94** (0.025)	13.5 (0.034)	32.9 (0.051)	9.01 (0.043)	9.13 (0.046)
	(0.5,20)	1.7 (0.008)	1.79 (0.014)	**1.68** (0.009)	3.01 (0.009)	7.34 (0.021)	1.98 (0.017)	2.31 (0.017)
	(0.5,50)	7.55 (0.03)	7.74 (0.035)	**6.96** (0.026)	13.3 (0.037)	37.5 (0.05)	10.5 (0.038)	9.08 (0.028)

well-approximated with a low-rank structure, whereas the graphon structure underlying the NS method might be more suitable.

5.1 Data Information

The first network is the Enron email network of Priebe et al. (2005) between 184 employees of the Enron company; edges indicate employees' email communication. The second network is a faculty friendship network between 81 faculty members at a UK university (Nepusz et al., 2008). The last network contains 755 airports in the United States, based on the U.S. Bureau of Transportation Statistics, where two airports are connected if there is a direct passenger flight between them. For simplicity, we ignore edge directions and weights. The average node density, betweenness centrality, and closeness centrality of each node are shown in Figure 1. Overall, the airport network exhibits much stronger heterogeneity in topological features. The nodes are, on average, much less connected and much less central, while special hub airports provide highly dense and central connections. Such strong topological heterogeneity suggests that the low-rank model may not be able to approximate this network well, as might occur in real-world situations (Seshadhri et al., 2020). As demonstrated in the link prediction evaluation, this topological variation does lead to a preference for different link prediction strategies.

5.2 Performance Evaluation

Contrary to our synthetic experiments, it is impossible to evaluate the MSE of the missing probability matrix. Instead, we specifically evaluate the methods based on their link prediction accuracy on the unobserved entries ($\mathbf{A}_{22}$). Because the entries in $\mathbf{A}_{22}$ are binary, for a given threshold on values of $\hat{\mathbf{P}}_{22}$, we get

$$\text{TPR} = \frac{\#\{\text{Correctly predicted edges}\}}{\#\{\text{Total existing edges}\}}$$
$$\text{FPR} = \frac{\#\{\text{Incorrectly predicted edges}\}}{\#\{\text{Total existing non-edges}\}}.$$

Varying the threshold and plotting the true positive rate (TPR) against the false positive rate (FPR) produces a ROC curve. We evaluate the link prediction accuracy using the area under the ROC curves (AUC) as our metric for link prediction performance. ROC curve is a commonly used performance metric in link prediction problems (Liben-Nowell and Kleinberg, 2007; Zhao et al., 2017).

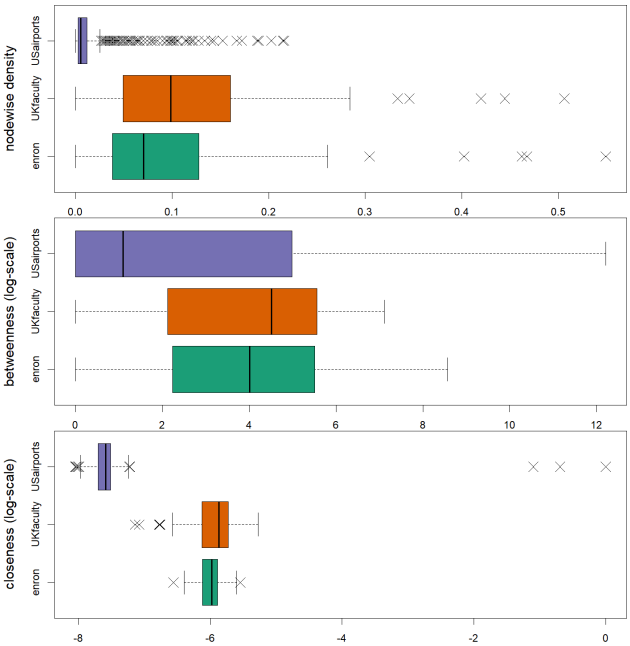

Figure 1: Attributes of Real Networks; Node Degree (Left), Betweenness (Middle), and Closeness (Right).

5.2.1 Evaluation under MAR

The same sampling mechanism as for synthetic networks is applied. Results for the three networks are summarized in Table 5. For the Enron and UK faculty networks, LE+ has the best prediction performance among all methods. Meanwhile, LE has comparable performance to LE+ and is better than other methods. The low-rank model is thus suitable for these two networks. The NS method also delivers reasonably good performance thanks to its generality. For

the U.S. airport network, under low sampling proportions $\rho = \{0.1, 0.2\}$, LE, SE, LE+, NS and DCBM are similar and are inferior to the SBM method. Under higher sampling proportions $\rho = \{0.5, 0.9\}$, the NS method has the best performance, while the other benchmark methods perform similarly. This pattern implies that all low-rank methods perform no better than a block approximation while missing additional structures which the NS method might integrate. Such an observation conveys that low-rank models may not be feasible for this airport network. Also, this is consistent with the synthetic experiments where SBM performs the best under very low sampling proportions due to its simplicity. The missing values for the UK faculty network under the sampling proportion $\rho = 0.1$ is not produced because it has a small network size ($N = 81$) inappropriate for such a low value of ρ.

Table 5: Predictive AUC on Three Networks.

Dataset	ρ	LE	SE	LE+	NS	ERGM	SBM	DCBM
enron	0.1	0.7 (0.004)	0.699 (0.005)	**0.724** (0.005)	0.634 (0.004)	0.508 (0.003)	0.663 (0.003)	0.68 (0.004)
	0.2	0.784 (0.003)	0.774 (0.003)	**0.803** (0.004)	0.718 (0.004)	0.508 (0.003)	0.713 (0.004)	0.741 (0.004)
	0.5	0.874 (0.003)	0.85 (0.002)	**0.882** (0.002)	0.824 (0.003)	0.527 (0.004)	0.774 (0.003)	0.788 (0.003)
	0.9	0.903 (0.005)	0.882 (0.005)	**0.909** (0.004)	0.845 (0.006)	0.64 (0.009)	0.805 (0.009)	0.828 (0.007)
UK faculty	0.2	0.728 (0.007)	0.701 (0.006)	**0.731** (0.009)	0.631 (0.007)	0.51 (0.002)	0.657 (0.008)	0.667 (0.006)
	0.5	0.837 (0.004)	0.792 (0.005)	**0.842** (0.004)	0.77 (0.005)	0.516 (0.003)	0.724 (0.007)	0.739 (0.007)
	0.9	**0.845** (0.012)	0.768 (0.016)	**0.846** (0.012)	0.785 (0.016)	0.606 (0.014)	0.733 (0.02)	0.747 (0.018)
US airports	0.1	0.76 (0.006)	0.776 (0.006)	0.775 (0.006)	0.742 (0.005)	0.502 (¡0.001)	**0.835** (0.002)	0.798 (0.003)
	0.2	0.822 (0.003)	0.82 (0.004)	0.837 (0.003)	0.817 (0.002)	0.505 (¡0.001)	**0.859** (0.002)	0.837 (0.002)
	0.5	0.878 (0.003)	0.89 (0.004)	**0.893** (0.003)	**0.893** (0.002)	0.527 (0.001)	**0.893** (0.002)	0.876 (0.002)
	0.9	0.89 (0.006)	0.91 (0.005)	0.92 (0.005)	**0.931** (0.005)	0.535 (0.004)	0.903 (0.007)	0.884 (0.008)

5.2.2 Evaluation under MNAR

Real network results again suggest that our method is applicable for non-uniform sampling as long as assumption A1 holds. Results for the three networks are summarized in Tables 6–7. Similar to the synthetic experiments, when compared with the uniform-sampling setting, the positive setting, which puts more weight on high-degree nodes, improves the performance of all methods and the negative setting does the opposite.

Table 6: Predictive AUC on Three Networks, Positive MNAR Setting.

Dataset	ρ	LE	SE	LE+	NS	ERGM	SBM	DCBM
enron	0.2	0.803 (0.003)	0.792 (0.003)	**0.822** (0.003)	0.743 (0.003)	0.514 (0.003)	0.728 (0.003)	0.748 (0.003)
	0.5	0.85 (0.004)	0.838 (0.005)	**0.864** (0.004)	0.802 (0.003)	0.538 (0.004)	0.776 (0.005)	0.746 (0.005)
UK faculty	0.2	0.747 (0.004)	0.732 (0.006)	**0.774** (0.004)	0.657 (0.005)	0.512 (0.002)	0.662 (0.008)	0.686 (0.006)
	0.5	**0.802** (0.008)	0.777 (0.008)	0.801 (0.007)	0.732 (0.006)	0.524 (0.004)	0.736 (0.006)	0.733 (0.008)
US airports	0.2	0.833 (0.003)	0.836 (0.005)	0.84 (0.004)	0.832 (0.002)	0.508 (<0.001)	**0.866** (0.002)	0.842 (0.002)
	0.5	0.833 (0.003)	**0.849** (0.003)	0.845 (0.003)	0.835 (0.003)	0.526 (0.001)	0.843 (0.004)	0.787 (0.006)

Table 7: Predictive AUC on Three Networks, Negative MNAR Setting.

Dataset	ρ	LE	SE	LE+	NS	ERGM	SBM	DCBM
enron	0.2	0.752 (0.004)	0.749 (0.003)	**0.781** (0.003)	0.699 (0.003)	0.501 (0.003)	0.697 (0.003)	0.723 (0.003)
	0.5	0.855 (0.003)	0.827 (0.002)	**0.867** (0.002)	0.807 (0.003)	0.514 (0.003)	0.753 (0.003)	0.777 (0.002)
UK faculty	0.2	0.675 (0.008)	0.651 (0.009)	**0.709** (0.007)	0.61 (0.007)	0.513 (0.002)	0.633 (0.008)	0.636 (0.006)
	0.5	**0.837** (0.004)	0.763 (0.005)	0.833 (0.004)	0.768 (0.005)	0.526 (0.003)	0.703 (0.007)	0.737 (0.006)
US airports	0.2	0.773 (0.006)	0.784 (0.007)	0.788 (0.006)	0.743 (0.006)	0.503 (<0.001)	**0.851** (0.002)	0.817 (0.003)
	0.5	0.871 (0.002)	0.887 (0.003)	**0.892** (0.003)	0.882 (0.002)	0.513 (0.001)	0.883 (0.002)	0.879 (0.002)

6 DISCUSSION

We have introduced an LE algorithm to fit low-rank models on egocentrically sampled partial networks. The approach is computationally efficient and presents theoretical guarantees for its correctness. Our technique is the first known consistent method for general low-rank models under egocentric sampling, and it does not require "missing completely at random" assumptions. It can accurately predict missing links when the true model is low-rank or can be approximated by low-rank structures. However, we want to stress that one may have difficulty determining whether a partial network is from a low-rank model in practice; no single mode of link prediction works well in all cases. A practically preferable approach involves combining methods to achieve more adaptive performance in various situations, as studied by Ghasemian et al. (2020); Li and Le (2021); Peixoto (2018); Yao et al. (2021). Notably, even in this ensemble setting, having a strong individual method that works well in numerous situations is still necessary. We believe the proposed LE approach greatly contributes to potential candidates as one such model.

The proposed method can be extended in several directions. One limitation of our theory is the strict low-rank assumption; theoretical properties for approximately low-rank models are thus far unknown. A more general theory in these scenarios would largely expand the method's scope. As another example, if a sequence of evolving networks is observed (subject to egocentric missingness), a critical but open question concerns how to fit a dynamic network model compatible with this missingness. We will leave this and other investigations for future work.

There are several applications in which we can potentially embed the current method. For example, graph learning methods such as GNN (Zhou et al., 2020) take networks as input, and our method can help to handle the semi-supervised prediction setting when the training data are only partially observed. Another application is to use our theory to study the privacy-preserving algorithms for network data (Hehir et al., 2021).

Acknowledgements

T. Li is supported in part by the NSF grant DMS-2015298 and the University of Virginia 3-Caveliers award. G. Chan (as a student at the University of Virginia) is also supported in part by the NSF grant DMS-2015298.

References

Airoldi, E. M., Blei, D., Fienberg, S., and Xing, E. (2008). Mixed membership stochastic blockmodels. *Advances in neural information processing systems*, 21.

Alatas, V., Banerjee, A., Chandrasekhar, A. G., Hanna, R., and Olken, B. A. (2016). Network structure and the aggregation of information: Theory and evidence from indonesia. *American Economic Review*, 106(7):1663–1704.

Ali, M. M. and Dwyer, D. S. (2009). Estimating peer effects in adolescent smoking behavior: a longitudinal analysis. *Journal of Adolescent Health*, 45(4):402–408.

Arnaboldi, V., Guazzini, A., and Passarella, A. (2013). Egocentric online social networks: Analysis of key features and prediction of tie strength in facebook. *Computer Communications*, 36(10-11):1130–1144.

Athreya, A., Fishkind, D. E., Tang, M., Priebe, C. E., Park, Y., Vogelstein, J. T., Levin, K., Lyzinski, V., and Qin, Y. (2017). Statistical inference on random dot product graphs: a survey. *The Journal of Machine Learning Research*, 18(1):8393–8484.

Bandiera, O. and Rasul, I. (2006). Social networks and technology adoption in northern mozambique. *The Economic Journal*, 116(514):869–902.

Bickel, P. J. and Chen, A. (2009). A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073.

Candès, E. J. and Plan, Y. (2010). Matrix completion with noise. *Proceedings of the IEEE*.

Chandrasekhar, A. and Lewis, R. (2011). Econometrics of sampled networks. *Unpublished manuscript, MIT.[422]*.

Chatterjee, S. (2015). Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214.

Chen, K. and Lei, J. (2018). Network cross-validation for determining the number of communities in network data. *Journal of the American Statistical Association*, 113(521):241–251.

Drineas, P., Kannan, R., and Mahoney, M. (2006). Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206.

Gao, C. and Ma, Z. (2021). Minimax rates in network analysis: Graphon estimation, community detection and hypothesis testing. *Statistical Science*, 36(1):16–33.

Ghasemian, A., Hosseinmardi, H., Galstyan, A., Airoldi, E. M., and Clauset, A. (2020). Stacking models for nearly optimal link prediction in complex networks. *Proceedings of the National Academy of Sciences*, 117(38):23393–23400.

Goldenberg, A., Zheng, A. X., Fienberg, S. E., Airoldi, E. M., et al. (2010). A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2(2):129–233.

Handcock, M. S. and Gile, K. J. (2010). Modeling social networks from sampled data. *The Annals of Applied Statistics*, 4(1):5.

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., Krivitsky, P. N., and Morris, M. (2019). *ergm: Fit, Simulate and Diagnose Exponential-Family Models for Networks*. The Statnet Project. R package version 3.10.4.

Hehir, J., Slavkovic, A., and Niu, X. (2021). Consistent spectral clustering of network block models under local differential privacy. *arXiv preprint arXiv:2105.12615*.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

Hunter, D. R. (2007). Curved exponential family models for social networks. *Social networks*, 29(2):216–230.

Jin, J., Ke, Z. T., and Luo, S. (2017). Estimating network memberships by simplex vertex hunting. *arXiv preprint arXiv:1708.07852*.

Karrer, B. and Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107.

Klopp, O., Tsybakov, A. B., and Verzelen, N. (2017). Oracle inequalities for network models and sparse graphon estimation. *The Annals of Statistics*, 45(1):316–354.

Kolaczyk, E. D. and Csárdi, G. (2014). *Statistical analysis of network data with R*, volume 65. Springer.

Krivitsky, P. N. and Morris, M. (2017). Inference for social network models from egocentrically sampled data, with application to understanding persistent racial disparities in hiv prevalence in the us. *The annals of applied statistics*, 11(1):427.

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

Le, C. M. and Li, T. (2020). Linear regression and its inference on noisy network-linked data. *arXiv preprint arXiv:2007.00803*.

Lei, J. and Rinaldo, A. (2015). Consistency of spectral

clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237.

Li, T. and Le, C. M. (2021). Network estimation by mixing: Adaptivity and more. *arXiv preprint arXiv:2106.02803*.

Li, T., Lei, L., Bhattacharyya, S., Van den Berge, K., Sarkar, P., Bickel, P. J., and Levina, E. (2022). Hierarchical community detection by recursive partitioning. *Journal of the American Statistical Association*, 117(538):951–968.

Li, T., Levina, E., Zhu, J., and Le, C. M. (2021). *randnet: Random Network Model Estimation, Selection and Parameter Tuning. R package version 0.4*.

Li, T., Wu, Y.-J., Levina, E., and Zhu, J. (2023). Link prediction for egocentrically sampled networks. *Journal of Computational and Graphical Statistics*, 0(0):1–24.

Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031.

Lin, Q., Lunde, R., and Sarkar, P. (2020). On the theoretical properties of the network jackknife. In *International Conference on Machine Learning*, pages 6105–6115. PMLR.

Little, R. J. and Rubin, D. B. (2019). *Statistical analysis with missing data*, volume 793. John Wiley & Sons.

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

Mukherjee, S. S., Sarkar, P., and Lin, L. (2017). On clustering network-valued data. *Advances in neural information processing systems*, 30.

Nepusz, T., Petróczi, A., Négyessy, L., and Bazsó, F. (2008). Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E*, 77(1):016107.

Newman, M. (2018). *Networks*. Oxford university press.

Owen, A. B. and Perry, P. O. (2009). Bi-cross-validation of the svd and the nonnegative matrix factorization. *The annals of applied statistics*, 3(2):564–594.

Peixoto, T. P. (2018). Reconstructing networks with unknown and heterogeneous errors. *Physical Review X*, 8(4):041011.

Plan, Y. and Vershynin, R. (2011). One-bit compressed sensing by linear programming. *arXiv preprint*, 2(1103909):1–18.

Priebe, C. E., Conroy, J. M., Marchette, D. J., and Park, Y. (2005). Scan statistics on enron graphs. *Computational & Mathematical Organization Theory*, 11(3):229–247.

Rohe, K., Chatterjee, S., and Yu, B. (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915.

Rubin-Delanchy, P., Cape, J., Tang, M., and Priebe, C. E. (2017). A statistical interpretation of spectral embedding: the generalised random dot product graph. *arXiv preprint arXiv:1709.05506*.

Sengupta, S. and Chen, Y. (2018). A block model for node popularity in networks with community structure. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(2):365–386.

Seshadhri, C., Sharma, A., Stolman, A., and Goel, A. (2020). The impossibility of low-rank representations for triangle-rich complex networks. *Proceedings of the National Academy of Sciences*, 117(11):5631–5637.

Yao, Y., Pirš, G., Vehtari, A., and Gelman, A. (2021). Bayesian hierarchical stacking: Some models are (somewhere) useful. *Bayesian Analysis*, 1(1):1–29.

Young, S. J. and Scheinerman, E. R. (2007). Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 138–149. Springer.

Zhang, Y., Levina, E., and Zhu, J. (2017). Estimating network edge probabilities by neighbourhood smoothing. *Biometrika*, 104(4):771–783.

Zhao, Y., Wu, Y.-J., Levina, E., and Zhu, J. (2017). Link prediction for partially observed networks. *Journal of Computational and Graphical Statistics*, 26(3):725–733.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI open*, 1:57–81.

Supplementary Materials of "Fitting low-rank models on egocentrically sampled partial networks"

1 Proofs

In this section, we present the derivation of the error bound in the main paper. In our analysis, we will say some event happens with high probability if it happens with probability tending to 1 as $N \to \infty$. We use C and c as generic universal constants that may vary case by case. Let $\mathbf{P}, \mathbf{A}, \hat{\mathbf{P}}$ be as defined as in the main paper, we introduce the following notations:

- $\|\cdot\|$ denotes the spectral norm
- $\|\cdot\|_F$ denotes the Frobenius norm
- $p^* = \max_{ij} p_{ij}$
- $\lambda_k(\mathbf{M})$ is the k-th largest eigenvalue of the matrix M
- $\sigma_k(\mathbf{M})$ is the k-th largest singular value of the matrix M

Lemma 1.1 (Owen and Perry (2009))**.** *For any $p \times q$ matrix $\mathbf{M}$ with the partition*

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{pmatrix},$$

Suppose $\operatorname{rank}(\mathbf{M}_{11}) = \operatorname{rank}(\mathbf{M})$*, we have*

$$\mathbf{M}_{11} = \mathbf{M}_{12}\mathbf{M}_{22}^{+}\mathbf{M}_{21}.$$

Lemma 1.2 (Lei and Rinaldo (2015))**.** *Let $\mathbf{P}$ be the probability under the inhomogeneous Erdös-Renyi model and $\mathbf{A}$ be the adjacency matrix from $\mathbf{P}$. Assume that $np^* \geq c \log n$ for some constant $c > 0$. There exists a constant C such that*

$$\|\mathbf{A} - \mathbf{P}\| \leq C\sqrt{np^*} \tag{1}$$

with high probability.

Lemma 1.3 (Yu, Wang, and Samworth (2015))**.** *Given a symmetric matrix $\mathbf{P}$. Suppose $\operatorname{rank}(\mathbf{P}) = K$ and let its eigendecomposition be $\mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$, where $\mathbf{\Sigma} = \operatorname{diag}(\lambda_1, \cdots, \lambda_K)$ contains all the eigenvalues in nonincreasing order. For another symmetric matrix $\mathbf{A}$ in the same dimension, suppose its rank K eigendecomposition is given by $\tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{U}}^T$. There exists an orthogonal matrix $\mathbf{O} \in \mathbb{R}^{K \times K}$ such that*

$$\|\tilde{\mathbf{U}}\mathbf{O} - \mathbf{U}\|_F \leq \frac{3\sqrt{K}||A - P||}{\lambda_K}. \tag{2}$$

Lemma 1.4 (Athreya et al. (2017))**.** *Let $\mathbf{P}$ be the probability under the inhomogeneous Erdös-Renyi model with $\operatorname{rank}(\mathbf{P}) = K$ and $\mathbf{A}$ be the adjacency matrix from $\mathbf{P}$. With the notations of Lemma 1.3, and the same orthogonal matrix $\mathbf{O}$, we have*

$$\|\mathbf{O}\mathbf{\Sigma} - \tilde{\mathbf{\Sigma}}\mathbf{O}\|_F \leq C(K^2 + \log n) \tag{3}$$

with high probability.

Assumption A1 (Low-rank recoverable)**.** *The rank of the model satisifes $\operatorname{rank}(\mathbf{P}_{11}) = \operatorname{rank}(\mathbf{P}) = K$.*

Assumption A2 (Well-conditioned model)**.** *There exists a constant $\psi > 0$ such that*

$$\frac{1}{\psi} \cdot np^* \leq \sigma_K(\mathbf{P}_{11}) \leq \sigma_1(\mathbf{P}_{11}) \leq \psi \cdot np^*$$

$$\frac{1}{\psi} \cdot Np^* \leq \sigma_K(\mathbf{P}) \leq \sigma_1(\mathbf{P}) \leq \psi \cdot Np^*$$

Assumption A1 is strictly needed to ensure the validity of the low-rank recovery on the population matrix $\mathbf{P}$ by Lemma 1.1. In contrast, assumptions A2 can be relaxed for better generality. However, we keep it in the current form for conciseness and interpretability of our error bound. Moreover, both A1 and A2 are indeed motivated by the general sparse graphon model of Bickel and Chen (2009) and are easy to hold when the egocentric sampling is done randomly on the nodes under many low-rank models. In particular, we have the following proposition

Theorem 1.5. *Under assumptions A1 and A2, further assume that $np^* > c \log n$ and $K \leq c\sqrt{\log n}$ for some constant $c > 0$, we have*

$$\|\hat{\mathbf{P}}_{22} - \mathbf{P}_{22}\|_F \leq C\sqrt{K}\left(\left(\frac{N}{n}\right)^{3/2}\sqrt{KNp^*} + \frac{N^2}{n^2}\log n\right) \tag{4}$$

for some constant $C > 0$ with high probability.

For illustration, consider the following two special cases

1. Suppose n and N are in the same order, we can see that the error bound on the missing network is in the order of $K\sqrt{np^*} + \sqrt{K}\log n$. Since $\|\mathbf{P}_{22}\|_F \sim n\sqrt{p^*}$, we know that $\|\hat{\mathbf{P}}_{22} - \mathbf{P}_{22}\|_F / \|\mathbf{P}_{22}\|_F \to 0$ and the estimation consistency is guaranteed under the current assumptions.

2. Suppose K is bounded and $np^* = \log^2 n$. Then the error bound is in the order of $\frac{N^2}{n^2}\log n$. So $\|\hat{\mathbf{P}}_{22} - \mathbf{P}_{22}\|_F / \|\mathbf{P}_{22}\|_F \to 0$ as long as $n \gg N^{4/5}$. Therefore, though our method allows the sampling proportion to be vanishing, the decaying rate has to be slow.

Proof of Theorem 1.5. Consider the prediction error of the probability matrix $\hat{\mathbf{P}}_{22} - \mathbf{P}_{22}$. We start with the spectral norm bound. By using the subproductivity of the spectral norm and triangular inequality, we have

$$\begin{aligned}
\|\hat{\mathbf{P}}_{22} - \mathbf{P}_{22}\| =& \|\mathbf{A}_{21}\tilde{\mathbf{P}}_{11}^{+}\mathbf{A}_{12} - \mathbf{P}_{21}\mathbf{P}_{11}^{+}\mathbf{P}_{12}\| \\
\leq& \|(\mathbf{A}_{21} - \mathbf{P}_{21})\mathbf{P}_{11}^{+}\mathbf{P}_{12}\| + \|\mathbf{A}_{21}(\tilde{\mathbf{P}}_{11}^{+}\mathbf{A}_{12} - \mathbf{P}_{11}^{+}\mathbf{P}_{12})\| \\
\leq& \|\mathbf{A}_{21} - \mathbf{P}_{21}\|\|\mathbf{P}_{11}^{+}\|\|\mathbf{P}_{12}\| + \|\mathbf{A}_{21}\|\|\tilde{\mathbf{P}}_{11}^{+}\mathbf{A}_{12} - \mathbf{P}_{11}^{+}\mathbf{P}_{12}\| \\
\leq& \|\mathbf{A}_{21} - \mathbf{P}_{21}\|\|\mathbf{P}_{11}^{+}\|\|\mathbf{P}_{12}\| + \|\mathbf{A}_{21}\|\|\tilde{\mathbf{P}}_{11}^{+}\mathbf{A}_{12} - \mathbf{P}_{11}^{+}\mathbf{A}_{12}\| + \|\mathbf{A}_{21}\|\|\mathbf{P}_{11}^{+}\mathbf{A}_{12} - \mathbf{P}_{11}^{+}\mathbf{P}_{12}\| \\
\leq& \|\mathbf{A}_{21} - \mathbf{P}_{21}\|\|\mathbf{P}_{11}^{+}\|\|\mathbf{P}_{12}\| + \|\mathbf{A}_{21}\|\|\mathbf{P}_{11}^{+}\|\|\mathbf{A}_{12} - \mathbf{P}_{12}\| + \|\mathbf{A}_{21}\|^2\|\tilde{\mathbf{P}}_{11}^{+} - \mathbf{P}_{11}^{+}\| \\
\leq& \|\mathbf{A}_{12} - \mathbf{P}_{12}\|\|\mathbf{P}_{11}^{+}\|\left(2\|\mathbf{P}_{12}\| + \|\mathbf{A}_{12} - \mathbf{P}_{12}\|\right) + \left(\|\mathbf{P}_{21}\| + \|\mathbf{A}_{12} - \mathbf{P}_{12}\|\right)^2\|\tilde{\mathbf{P}}_{11}^{+} - \mathbf{P}_{11}^{+}\| \qquad (5) \\
=& \mathcal{I} + \mathcal{II}. \qquad (6)
\end{aligned}$$

Denote the eigendecompositions up to K of $\mathbf{P}_{11}$ and $\mathbf{A}_{11}$ by $\mathbf{P}_{11} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$ and $\mathbf{A}_{11} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{U}}^T$ respectively. Note that since $\mathrm{rank}(\mathbf{P}) = K$, the eigendecomposition of $\mathbf{P}$ is exact. Note that, since the singular values match the eigenvalues up to their signs, we have $\mathbf{P}_{11}^{+} = \mathbf{U}\mathbf{\Sigma}^{-1}\mathbf{U}^T$ and $\tilde{\mathbf{P}}_{11}^{+} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}^{-1}\mathbf{U}^T$. We try to control the terms separately.

We want to control the concentration of each component of the $\mathbf{A}$ matrix partition. In particular, we are taking the joint event of Lemma 1.4, and Lemma 1.2 for $\mathbf{A}$ and $\mathbf{A}_{11}$. Notice that here $np^* > c\log n$ indicates that $Np^* > c\log N$ due to the monotonicity of $\log n/n$. Under this condition, therefore, we have $\|\mathbf{A}_{11} - \mathbf{P}_{11}\| \leq C\sqrt{np^*}$ and $\|\mathbf{A}_{21} - \mathbf{P}_{21}\| \leq \|\mathbf{A} - \mathbf{P}\| \leq C\sqrt{Np^*}$. Under this event, we also have

$$|\lambda_k(\mathbf{A}_{11}) - \lambda_k(\mathbf{P}_{11})| \leq \|\mathbf{A}_{11} - \mathbf{P}_{11}\| \leq \sqrt{np^*}, 1 \leq k \leq K.$$

Therefore, due to the assumption that $\lambda_K(\mathbf{P}_{11}) \geq \psi np^*$ and $np^* \geq c\log n$, for sufficiently large n, we have

$$|\lambda_K(\mathbf{A}_{11})| \geq \frac{1}{2}|\lambda_K(\mathbf{P}_{11})|.$$

Upper bound of term $\mathcal{I}$.

$$\|\mathbf{P}_{11}^{+}\| = \frac{1}{|\lambda_K(\mathbf{P}_{11})|} \leq \psi \cdot \frac{1}{np^*}.$$

Also notice that $\mathbf{P}_{12}$ is a submatrix of $\mathbf{P}$ so $\|\mathbf{P}_{12}\| \leq \psi N p^*$. So we have

$$\mathcal{I} = \|\mathbf{A}_{12} - \mathbf{P}_{12}\|\|\mathbf{P}_{11}^{+}\| \left(2\|\mathbf{P}_{12}\| + \|\mathbf{A}_{12} - \mathbf{P}_{12}\|\right) \leq C\frac{N}{n}\sqrt{Np^*}.$$

Upper bound of term $\mathcal{II}$. Let $\mathbf{O} \in \mathbb{R}^{K\times K}$ be an orthogonal matrix in Lemmas 1.3 and 1.4. Consider the term $\|\tilde{\mathbf{P}}_{11}^{+} - \mathbf{P}_{11}^{+}\|$:

$$\begin{aligned}
\|\tilde{\mathbf{P}}_{11}^{+} - \mathbf{P}_{11}^{+}\| =& \|\tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^{T} - \mathbf{U}\mathbf{\Sigma}^{-1}\mathbf{U}^{T}\| \\
=& \|\tilde{\mathbf{U}}\mathbf{O}\mathbf{O}^{\top}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^{\top} - \mathbf{U}\mathbf{\Sigma}^{-1}\mathbf{U}^{\top}\| \\
\leq& \|\tilde{\mathbf{U}}\mathbf{O} - \mathbf{U}\|\|\mathbf{O}^{\top}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^{\top}\| + \|\mathbf{U}\mathbf{O}^{\top}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^{\top} - \mathbf{U}\mathbf{\Sigma}^{-1}\mathbf{U}^{\top}\| \\
\leq& \|\tilde{\mathbf{U}}\mathbf{O} - \mathbf{U}\|\|\mathbf{O}^{\top}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^{\top}\| + \|\mathbf{U}\mathbf{O}^{\top}\tilde{\mathbf{\Sigma}}^{-1}\mathbf{O}\mathbf{O}^{\top}\tilde{\mathbf{U}}^{\top} - \mathbf{U}\mathbf{O}^{\top}\tilde{\mathbf{\Sigma}}^{-1}\mathbf{O}\mathbf{U}^{\top}\| \\
&+ \|\mathbf{U}\mathbf{O}^{\top}\tilde{\mathbf{\Sigma}}^{-1}\mathbf{O}\mathbf{U}^{\top} - \mathbf{U}\mathbf{\Sigma}^{-1}\mathbf{U}^{\top}\| \\
\leq& \|\tilde{\mathbf{U}}\mathbf{O} - \mathbf{U}\|\|\tilde{\mathbf{\Sigma}}^{-1}\| + \|\mathbf{U}\mathbf{O}^{\top}\tilde{\mathbf{\Sigma}}^{-1}\mathbf{O}\|\|\mathbf{O}^{\top}\tilde{\mathbf{U}}^{\top} - \mathbf{U}^{\top}\| + \|\mathbf{U}\|\|\mathbf{O}^{\top}\tilde{\mathbf{\Sigma}}^{-1}\mathbf{O} - \mathbf{\Sigma}^{-1}\|\|\mathbf{U}\| \\
\leq& 2\|\tilde{\mathbf{U}}\mathbf{O} - \mathbf{U}\|\|\tilde{\mathbf{\Sigma}}^{-1}\| + \|\mathbf{O}^{\top}\tilde{\mathbf{\Sigma}}^{-1}\mathbf{O} - \mathbf{\Sigma}^{-1}\| \\
\leq& 2\|\tilde{\mathbf{U}}\mathbf{O} - \mathbf{U}\|\|\tilde{\mathbf{\Sigma}}^{-1}\| + \|\tilde{\mathbf{\Sigma}}^{-1}\mathbf{O} - \mathbf{O}\mathbf{\Sigma}^{-1}\| \\
\leq& 2\|\tilde{\mathbf{U}}\mathbf{O} - \mathbf{U}\|\|\tilde{\mathbf{\Sigma}}^{-1}\| + \|\tilde{\mathbf{\Sigma}}^{-1}\|\|\mathbf{\Sigma}^{-1}\|\|\mathbf{O}\mathbf{\Sigma} - \tilde{\mathbf{\Sigma}}\mathbf{O}\| \\
=& \|\tilde{\mathbf{\Sigma}}^{-1}\|\left(2\|\tilde{\mathbf{U}}\mathbf{O} - \mathbf{U}\| + \|\mathbf{\Sigma}^{-1}\|\|\mathbf{O}\mathbf{\Sigma} - \tilde{\mathbf{\Sigma}}\mathbf{O}\|\right) \\
\leq& [\frac{1}{2}\sigma_K(\mathbf{P}_{11})]^{-1}\Big\{6\sqrt{\frac{K}{np^*}} + [\sigma_K(\mathbf{P}_{11})]^{-1}C\log(n)\Big\} \\
\leq& C\frac{1}{np^*}(\sqrt{\frac{K}{np^*}} + \frac{\log n}{np^*}).
\end{aligned}$$

Therefore, we have

$$\mathcal{II} = \left(\|\mathbf{P}_{21}\| + \|\mathbf{A}_{12} - \mathbf{P}_{12}\|\right)^2 \|\tilde{\mathbf{P}}_{11}^{+} - \mathbf{P}_{11}^{+}\| \leq C\left(\left(\frac{N}{n}\right)^{3/2}\sqrt{KNp^*} + \frac{N^2}{n^2}\log n\right).$$

Note that this bound for $\mathcal{II}$ dominates that for $\mathcal{I}$. Substituting both bounds for $\mathcal{I}$ and $\mathcal{II}$ into (5) leads to

$$\|\hat{\mathbf{P}}_{22} - \mathbf{P}_{22}\| \leq C\left(\left(\frac{N}{n}\right)^{3/2}\sqrt{KNp^*} + \frac{N^2}{n^2}\log n\right). \tag{7}$$

Finally, notice that $\mathrm{rank}(\hat{\mathbf{P}}_{22})$ and $\mathrm{rank}(\mathbf{P}_{22}) = K$, which indicates that $\mathrm{rank}(\hat{\mathbf{P}}_{22} - \mathbf{P}_{22}) \leq 2K$. So we have the Frobenius norm bound

$$\|\hat{\mathbf{P}}_{22} - \mathbf{P}_{22}\|_F \leq C\sqrt{K}\left(\left(\frac{N}{n}\right)^{3/2}\sqrt{KNp^*} + \frac{N^2}{n^2}\log n\right). \tag{8}$$

Finally, notice that if even the truncation to $[0, 1]$ is applied, this process would not increase the error at all entries of $\hat{\mathbf{P}}_{22}$, so the error bound still holds. □

The error bound for the full matrix recovery is a straightforward extension of Theorem 1.5.

Corollary 1. *Under the assumptions of Theorem 1.5, for the full matrix estimator $\hat{\mathbf{P}}$, we have*

$$\|\hat{\mathbf{P}} - \mathbf{P}\|_F \leq C\sqrt{K}\left(\left(\frac{N}{n}\right)^{3/2}\sqrt{KNp^*} + \frac{N^2}{n^2}\log n\right)$$

with high probability.

Proof of Corollary 1.

$$\begin{aligned}\|\hat{\mathbf{P}}-\mathbf{P}\|_F^2 &= \|\hat{\mathbf{P}}_{11}-\mathbf{P}_{11}\|_F^2 + 2\|\hat{\mathbf{P}}_{12}-\mathbf{P}_{12}\|_F^2 + \|\hat{\mathbf{P}}_{22}-\mathbf{P}_{22}\|_F^2 \\ &\le 2\|\hat{\mathbf{P}}_{11}-\mathbf{P}_{11}\|_F^2 + 2\|\hat{\mathbf{P}}_{12}-\mathbf{P}_{12}\|_F^2 + \|\hat{\mathbf{P}}_{22}-\mathbf{P}_{22}\|_F^2 \\ &\le 2\|\hat{\mathbf{P}}_{\text{obs}}-\mathbf{P}_{\text{obs}}\|_F^2 + \|\hat{\mathbf{P}}_{22}-\mathbf{P}_{22}\|_F^2 \\ &\le 2K\|\hat{\mathbf{P}}_{\text{obs}}-\mathbf{P}_{\text{obs}}\|^2 + \|\hat{\mathbf{P}}_{22}-\mathbf{P}_{22}\|_F^2.\end{aligned}$$

For the first term, under the same high probability event of Theorem 1.5, we have

$$\begin{aligned}\|\hat{\mathbf{P}}_{\text{obs}}-\mathbf{P}_{\text{obs}}\| &\le \|\hat{\mathbf{P}}_{\text{obs}}-\mathbf{A}_{\text{obs}}\| + \|\mathbf{P}_{\text{obs}}-\mathbf{A}_{\text{obs}}\| \\ &\le \sigma_K(\mathbf{A}_{\text{obs}}) + \|\mathbf{P}_{\text{obs}}-\mathbf{A}_{\text{obs}}\| \\ &\le \|\mathbf{P}_{\text{obs}}-\mathbf{A}_{\text{obs}}\| + \sigma_K(\mathbf{P}_{\text{obs}}) + \|\mathbf{P}_{\text{obs}}-\mathbf{A}_{\text{obs}}\| \\ &\le 2\|\mathbf{P}-\mathbf{A}\| \\ &\le C\sqrt{Np^*}.\end{aligned}$$

Combining this result with Theorem 1.5, we have

$$\|\hat{\mathbf{P}}-\mathbf{P}\|_F^2 \le C\left(KNp^* + K^2\left(\frac{N}{n}\right)^3 Np^* + K\left(\frac{N}{n}\right)^4 \log^2 n\right) \le C'\left(K^2\left(\frac{N}{n}\right)^3 Np^* + K\left(\frac{N}{n}\right)^4 \log^2 n\right)$$

So we have

$$\|\hat{\mathbf{P}}-\mathbf{P}\|_F \le C\sqrt{K}\left(\left(\frac{N}{n}\right)^{3/2}\sqrt{KNp^*} + \frac{N^2}{n^2}\log n\right)$$

under the event. From the proof, it can also be seen that the major error for the full matrix estimation is still on the unobserved component $\hat{\mathbf{P}}_{22}$.

Finally, notice that if even the truncation to $[0,1]$ is applied, this process would not increase the error at all entries of $\hat{\mathbf{P}}$, so the error bound still holds.

□

References

Athreya, A., Fishkind, D. E., Tang, M., Priebe, C. E., Park, Y., Vogelstein, J. T., … Qin, Y. (2017). Statistical inference on random dot product graphs: a survey. *The Journal of Machine Learning Research*, *18*(1), 8393–8484.

Bickel, P. J., & Chen, A. (2009). A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, *106*(50), 21068–21073.

Lei, J., & Rinaldo, A. (2015). Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, *43*(1), 215–237.

Owen, A. B., & Perry, P. O. (2009). Bi-cross-validation of the svd and the nonnegative matrix factorization. *The annals of applied statistics*, *3*(2), 564–594.

Yu, Y., Wang, T., & Samworth, R. J. (2015). A useful variant of the davis–kahan theorem for statisticians. *Biometrika*, *102*(2), 315–323.