A Theoretical Analysis of DeepWalk and Node2vec for Exact Recovery of Community Structures in Stochastic Blockmodels

Yichi Zhang and Minh Tang

***Abstract*—Random-walk-based network embedding algorithms like DeepWalk and node2vec are widely used to obtain euclidean representation of the nodes in a network prior to performing downstream inference tasks. However, despite their impressive empirical performance, there is a lack of theoretical results explaining their large-sample behavior. In this paper, we study node2vec and DeepWalk through the perspective of matrix factorization. In particular, we analyze these algorithms in the setting of community detection for stochastic blockmodel graphs (and their degree-corrected variants). By exploiting the row-wise uniform perturbation bound for leading singular vectors, we derive high-probability error bounds between the matrix factorization-based node2vec/DeepWalk embeddings and their true counterparts, uniformly over all node embeddings. Based on strong concentration results, we further show the *perfect* membership recovery by node2vec/DeepWalk, followed by K-means/medians algorithms. Specifically, as the network becomes sparser, our results guarantee that with large enough window size and vertex number, applying K-means/medians on the matrix factorization-based node2vec embeddings can, with high probability, correctly recover the memberships of all vertices in a network generated from the stochastic blockmodel (or its degree-corrected variants). The theoretical justifications are mirrored in the numerical experiments and real data applications, for both the original node2vec and its matrix factorization variant.**

***Index Terms*—Stochastic blockmodel, network embedding, perfect community recovery, node2vec, DeepWalk, matrix factorization.**

I. Introduction

GIVEN a network $\mathcal{G}$, a popular approach for analyzing $\mathcal{G}$ is to first map or embed its vertices into some low dimensional euclidean space and then apply machine learning and statistical inference procedures in this space. Through this embedding process, multiple tasks could be conducted on the network, such as community detection (e.g., [1], [2]), link prediction (e.g., [3]), node classification (e.g., [4], [5]) and network visualization (e.g., [6]). There has been a large and diverse collection of network embedding algorithms proposed in the literature, including those based on spectral embedding [7], [8], [9], multivariate statistical dimension reduction [10], [11], and neural network [12], [13], [14]. See [15], [16], [17] and [18] for recent surveys of network embedding and graph representation learning.

In recent years there has been significant interest in network embeddings based on random walks. The most well-known examples include DeepWalk [4] and node2vec [19]. These algorithms are computationally efficient and furthermore yield impressive empirical performance in many different scientific applications including recommendation systems [20], biomedical natural language processing [21], human protein identification [22], traffic prediction [23] and city road layout modeling [24]. Nevertheless, despite their wide-spread use, there is still a lack of theoretical results on their large-sample properties. In particular, it is unclear what the node embeddings represent as well as their behavior as the number of nodes increases.

Theoretical properties for DeepWalk, node2vec, and related algorithms had been studied previously in the computer science community. The focus here had been mostly on the convergence of the *entries* of the co-occurrence matrix as the lengths and/or number of random walks go to infinity. For example, motivated by the analysis in [25] for word2vec, the authors of [26], [27] showed that DeepWalk and node2vec using the skip-gram model with negative sampling is equivalent to factorizing a matrix whose entries are obtained by taking the entry-wise logarithm of a co-occurrence matrix, provided that the embedding dimension d is sufficiently large (possibly exceeding the number of nodes n). These authors also derived the limiting form of the entries of this matrix as the length of the random walks goes to infinity. These results were further extended in [28] to yield finite-sample concentration bounds for the co-occurrence entries. Note, however, that the above-cited works focused exclusively on the case of a fixed graph and thus do not provide results on the large sample behavior of these algorithms as n increases.

The statistical community, in contrast, had extensively studied the large-sample properties of graph embeddings based on matrix factorization. However the embedding algorithms considered are almost entirely based on singular value decomposition

Manuscript received 23 October 2022; revised 11 April 2023; accepted 20 October 2023. Date of publication 25 October 2023; date of current version 8 January 2024. The work of Minh Tang was supported by DMS under Grant 2210805. Recommended for acceptance by K. M. Lee. *(Corresponding author: Yichi Zhang.)*

The authors are with the Department of Statistics, North Carolina State University, Raleigh, NC 27606 USA (e-mail: yzhan239@ncsu.edu; minh_tang@ncsu.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2023.3327631, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2023.3327631

(SVD) of either the adjacency matrix or the Laplacian matrix and its normalized and/or regularized variants. For example, in the setting of the popular stochastic blockmodel random graphs, [7] and [8] derived consistency results for a truncated SVD of the normalized Laplacian matrix and the adjacency matrix. Subsequently [29], [30] strengthened these results by providing central limit theorems for the components of the eigenvectors of either the adjacency matrix or the normalized Laplacian matrix under the more general random dot product graphs model. As DeepWalk and node2vec are based on taking the entry-wise logarithm of a random-walk co-occurrence matrix, the techniques used in these cited results do not readily translate to this setting.

A. Contributions of the Current Paper

The current paper studies large-sample properties of random-walk-based embedding algorithms. We first present convergence results for the embeddings of DeepWalk and node2vec in the case of stochastic blockmodel graphs and their degree-corrected variant. We then show that running K-means or K-medians on the resulting embeddings is sufficient for *exact* recovery of the latent community assignments. Our theoretical results thus provide a bridge between previous results in the computer science community and their statistics counterparts.

We emphasize that our focus on stochastic blockmodel graphs is done purely for ease of exposition. Indeed, most of our results continue to hold for the more general inhomogeneous Erdős-Rényi (IER) random graphs model [31], [32], provided that the edge probabilities are sufficiently homogeneous, i.e., the minimum and maximum values for the edge probabilities are of the same order (possibly converging to 0) as n increases; recall that IER is one of the most general models for edge independent random graphs. In particular, we can show that the co-occurrence matrices constructed from the sampled networks is uniformly close (entrywise) to that for the true but unknown edge probabilities matrices. However, as IER random graphs need not possess low-dimensional structure (even when n increases), it is not clear what the embeddings obtained from these co-occurrence matrices represent. See Section VI for further discussion.

We now outline our approach. The original node2vec and DeepWalk algorithms are based on optimizing a non-convex skip-gram model using stochastic gradient descent (SGD); this optimization problem has multiple local minimum and the obtained embeddings can thus be numerically unstable (see e.g., [33]). We instead consider, for each embedding dimension d, the optimal low-rank approximation of an observed transformed co-occurrence matrix similar to that used in [25], [28], and recently [27], [34]. We first show that the entries of the co-occurrence matrix computed using the observed adjacency matrix is *uniformly* close to the entries of the co-occurrence matrix computed using the true but unknown edge probabilities matrix. This uniform bound implies that the entry-wise logarithm of the two co-occurrence matrices are also *uniformly* close and thus, with high probability, the co-occurrence matrix constructed using the observed graph is well-defined. In the case of stochastic blockmodel graphs the true edge probabilities matrix give rise to a (transformed) co-occurrence matrix with rank at most K where K is the number of blocks and thus for stochastic blockmodel graphs with $K \ll n$ blocks. By leveraging both classical (e.g., the celebrated Davis-Kahan theorem [35]) as well as recent results on matrix perturbations in the $2 \to \infty$ norm (e.g., [36], [37]), we show that the truncated low-rank representation of both matrices are *uniformly* close, i.e., the embeddings of the observed graph is, up to orthogonal transformation, approximately the same as that for the true edge probabilities matrix. Therefore, by running K-means or K-medians on the embeddings of the observed graph, we can with high probability recover the latent community structure for *every* node.

TABLE I
TABLE OF NOTATION

Notation	Description
$\mathcal{G} = (\mathbf{V}, \mathbf{E})$	graph with n vertices; $\mathbf{V} = \{v_i\}_{i=1}^{n}$ and $\mathbf{E} = \{e_{ii'}\}_{i,i'=1}^{n}$ are the vertice and edge sets, respectively
$\mathcal{N}(v_i)$	set of nodes $v_{i'}$ in $\mathcal{G}$ adjacent to v_i
$\mathbf{A} = [a_{ii'}]$	$n \times n$ adjacency matrix of $\mathcal{G}$
$\mathbf{P} = [p_{ii'}]$	$n \times n$ edge probability matrix of $\mathcal{G}$
d_i	observed degree of node v_i in $\mathcal{G}$, i.e., $d_i = \sum_{i'=1}^{n} a_{ii'}$
p_i	expected degree of node v_i in $\mathcal{G}$, i.e., $p_i = \sum_{i'=1}^{n} p_{ii'}$
$\mathbf{D_A}$	diagonal matrix with diagonal entries $d_1, d_2, \ldots, d_n$.
$\mathbf{D_P}$	diagonal matrix with diagonal entries $p_1, \ldots, p_n$.
$\hat{\mathbf{W}} = \mathbf{A}\mathbf{D}_{\mathbf{A}}^{-1}$	1-step transition matrix for a random walk on $\mathcal{G}$
$\mathbf{W} = \mathbf{P}\mathbf{D}_{\mathbf{P}}^{-1}$	counterpart of $\hat{\mathbf{W}}$ based on the edge probability matrix $\mathbf{P}$
$\mathbf{0}_d$	d dimensional vector with all elements equal to 0
$\mathbf{1}_d$	d dimensional vector with all elements equal to 1
$\mathbb{O}_{d,d'}$	set of all $d \times d'$ matrices with orthonormal columns
$\mathbb{O}_d$	set of all $d \times d$ orthogonal matrices

Our paper is organized as follows. In Section II we give a brief introduction of node2vec [19] and DeepWalk [4], and describe the matrix factorization perspective for these algorithms. In particular DeepWalk can be treated as a special case of node2vec by setting the 2nd-order random-walk parameters (p, q) to be (1,1), which will be assumed in Section III for simplicity of theoretical analysis. In Section III we provide uniform entry-wise error bounds for the entries of the t-step random-walk transition matrix and their implications for community recovery. The theoretical results in Section III hold for both the dense and sparse regimes where the average degree grows linearly and sublinearly in the number of nodes, respectively. In Section IV we present simulations to corroborate our theoretical results. In Section V we apply node2vec to three real-world network datasets and show its remarkable practical performances. We conclude the paper in Section VI with a discussion of some open questions and potential improvements. All proofs of the stated results, associated technical lemmas, and additional numerical results are provided in the Supplementary File, available online.

B. Notation

We summarize in Table I some general notations that are used throughout this paper. Unless specified otherwise, all graphs $\mathcal{G}$ in this paper are assumed to be undirected, unweighted and loop-free. In the subsequent discussion we often assume that the upper triangular entries of adjacency matrix $\mathbf{A}$ are independent Bernoulli random variables with $\mathbb{E}[a_{ii'}] = p_{ii'}$ when $i' < i$, where $p_{ii'}$ is the ii'th entry in edge probability matrix $\mathbf{P}$. As $\mathbf{A}$ and $\mathbf{P}$ are symmetric, we also set $a_{ii'} = a_{i'i}$ and $p_{ii'} = p_{i'i}$ for $i' > i$.

We use $\|\cdot\|$, $\|\cdot\|_{\mathrm{F}}$, $\|\cdot\|_\infty$ and $\|\cdot\|_{\max}$ to denote the spectral norm, Frobenius norm, maximum absolute row sum, and maximum entry-wise value of a matrix, respectively. We also use $\|\cdot\|_{\max,\text{off}}$ and $\|\cdot\|_{\max,\text{diag}}$ to denote the maximum value for the off-diagonal and diagonal entries of a matrix, i.e., for a square matrix $\mathbf{M} = [m_{ii'}]$,

$$\|\mathbf{M}\|_{\max,\text{off}} = \max_{i \neq i'} |m_{ii'}|, \quad \|\mathbf{M}\|_{\max,\text{diag}} = \max_{i} |m_{ii}|. \qquad (1.1)$$

We use $|\cdot|$ to denote the absolute value of a real number as well as the cardinality of a finite set.

For two terms a and b, let $a \wedge b := \min\{a, b\}$. We write $a \precsim b$ (resp. $a \succsim b$) if there exists a constant c (resp. c') not depending on a and b such that $a \leq cb$ (resp. $a \geq c'b$). If $a \precsim b$ and $a \succsim b$ then $a \asymp b$. We say an event $\mathcal{A}$ depending on n happens with high probability (whp) if $\mathbb{P}(\mathcal{A}) \geq 1 - \mathcal{O}(n^{-a})$ for some constant $a > 3$. Finally, for random sequences A_n, B_n, we write $A_n = O_{\mathbb{P}}(B_n)$ if A_n/B_n is bounded whp and $A_n = o_{\mathbb{P}}(B_n)$ if $A_n/B_n \to 0$ whp. For a given positive integer K, we denote by $[K]$ the set $\{1, 2, \ldots, K\}$.

II. Summary of node2vec and SBM

In this section we first provide a brief overview of the node2vec algorithm. We then discuss the popular stochastic blockmodel (SBM) for random graphs. Finally we discuss a matrix factorization perspective to node2vec and show that, for a graph $\mathcal{G}$ generated from a stochastic blockmodel, this matrix factorization approach leads to a low-rank approximation of an elementwise non-linear transformation of the random walk transition matrix for $\mathcal{G}$.

A. Node2vec With Negative Sampling

First introduced in [19], node2vec is a computationally efficient and widely-used algorithm for network embedding. Motivated by the ideas behind word2vec for text documents [38], node2vec generates sequences of nodes using random walks which are then fed into a skip-gram model [39] to yield the node embeddings. The original skip-gram model is quite computationally demanding for large networks and hence, in practice, usually replaced by a skip-gram with negative sampling (SGNS). The resulting algorithm is summarized below.

1) *(Sampling Random Paths):* First generates r independent 2nd order random walks on $\mathcal{G}$ with each having a fixed length L. A 2nd order random walk of length L starting at v_i with parameters p and q is generated as follows. First let $v_1^{(i)} = v_i$. Next sample $v_2^{(i)}$ from $\mathcal{N}(v_1^{(i)})$ uniformly at random. Then for $3 \leq \ell \leq L$, sample $v_\ell^{(i)} \in \mathcal{N}\left(v_{\ell-1}^{(i)}\right)$ with probability,

$$\mathbb{P}(v_\ell^{(i)} = v_0) = \begin{cases} \frac{1}{p} J(v_0) & \text{if } v_0 = v_{\ell-2}^{(i)}, \\ J(v_0) & \text{if } v_0 \in \mathcal{N}(v_{\ell-2}^{(i)}), \\ \frac{1}{q} J(v_0) & \text{if } v_0 \notin \mathcal{N}(v_{\ell-2}^{(i)}), \end{cases}$$

where $J(v_0)$ is given by

$$\frac{1}{J(v_0)} = p^{-1} + \left|\mathcal{N}\left(v_{\ell-2}^{(i)}\right) \cap \mathcal{N}\left(v_{\ell-1}^{(i)}\right)\right| + q^{-1}\left|\mathcal{N}\left(v_{\ell-2}^{(i)}\right)^{\mathrm{c}} \cap \mathcal{N}\left(v_{\ell-1}^{(i)}\right)\right| \qquad (2.1)$$

The form of $J(v_0)$ allows for $v_\ell^{(i)}$ to have possibly unbalanced probabilities of reaching three different types of nodes in the neighborhood of $v_{\ell-1}^{(i)}$, namely (1) the previous node $v_{\ell-2}^{(i)}$; (2) nodes belonging to both the neighborhoods of $v_{\ell-2}^{(i)}$ and $v_{\ell-1}^{(i)}$; (3) nodes belonging only to the neighborhood of $v_{\ell-1}^{(i)}$ but not the neighborhood of $v_{\ell-2}^{(i)}$. The parameters $p > 0$ and $q > 0$ provide weights for these three different types of nodes and hence control the speed at which the random walk leaves the neighborhood of the original node v_i. In this paper, we assume that the starting vertex v_i of any random walk is sampled according to a stationary distribution $\mathbf{S} = (S_1, \ldots, S_n)$ on $\mathcal{G}$ with

$$\mathbb{P}\left(\text{Starting Vertex is } v_i\right) = S_i = \frac{d_i}{2|\mathbf{E}|} \qquad (2.2)$$

for all $v_i \in \mathbf{V}$. For a given $i \in [n]$ we denote by r_i the number of random walks starting from v_i, $\boldsymbol{\ell}_j^{(i)}$ as the jth random walk starting from v_i and $\mathcal{L}_i = \{\boldsymbol{\ell}_j^{(i)}, j \in [r_i]\}$ as the set of all random walks starting from v_i.

Remark 1: We consider only the case of $p = q = 1$ for our theoretical analysis. The choice $p = q = 1$ is the default setting for node2vec as suggested in the original paper [19] and leads to a sampling scheme equivalent to that of DeepWalk [4]; the subsequent analysis thus also applies to DeepWalk.

2) *(Calculating $\mathbf{C}$):* Borrowing ideas from word2vec [38], node2vec creates a $n \times n$ node-context matrix $\mathbf{C} = [C_{ii'}]_{n \times n}$ whose ii'th entry records the number of times the pair $(v_i, v_{i'})$ appears among all random paths in $\bigcup_{i=1}^n \mathcal{L}_i$. More specifically, for a given window size (t_L, t_U), $C_{ii'}$ is the number of times that $(v_i, v_{i'})$ appears within a sequence

$$\ldots, v_i, \underbrace{\ldots\ldots}_{t-1\text{vertices}}, v_{i'}, \ldots \quad \text{or}$$

$$\ldots, v_{i'}, \underbrace{\ldots\ldots}_{t-1\text{vertices}}, v_i, \ldots \qquad (2.3)$$

among all random paths in $\bigcup_{i=1}^n \mathcal{L}_i$; here t is any integer satisfying $t_L \leq t \leq t_U \leq L - 1$

Remark 2: The original node2vec algorithm fixed $t_L = 1$ while in this paper, we allow for varying t_L for a more flexible theoretical analysis. In Section III we show that different values for (t_L, t_U) could lead to different convergence rates for the embedding and furthermore appropriate values for (t_L, t_U) depend intrinsically on the sparsity of the network.

3) *(Skip-gram model with negative sampling):* Given the $n \times n$ matrix $\mathbf{C}$ and an embedding dimension d, node2vec uses the SGNS model to learn the node embedding matrix $\mathbf{F} \in \mathbb{R}^{n\times d}$ and the context embedding matrix $\mathbf{F}' \in \mathbb{R}^{n\times d}$. The ith row of $\mathbf{F}$ is the d-dimensional embedding vector of node v_i. In slight contrast to the original node2vec, in this paper we do not require the constraint $\mathbf{F} = \mathbf{F}'$. The objective function of SGNS model for a given $\mathbf{C}$ is defined as

$$\mathrm{g}(\mathbf{F}, \mathbf{F}') = \sum_{ij} C_{ij} \left[\log\left\{\sigma(\boldsymbol{f}_i^\top \boldsymbol{f}'_j)\right\} + \kappa \mathbb{E}_{\boldsymbol{f}'_{\mathcal{N}} \sim \mathbf{P}_{\text{ns}}} \left[\log\left\{\sigma(-\boldsymbol{f}_i^\top \boldsymbol{f}'_{\mathcal{N}})\right\}\right]\right]. \quad (2.4)$$

Here σ is the logistic function, $\boldsymbol{f}_i$ (resp. $\boldsymbol{f}'_j$) is the ith (resp. jth) row of $\mathbf{F}$ (resp. $\mathbf{F}'$), κ is the ratio of negative to positive samples, and $\boldsymbol{f}'_{\mathcal{N}}$ is a negative sample generated from the empirical unigram distribution $\mathbf{P}_{\text{ns}}$, i.e., $\boldsymbol{f}'_{\mathcal{N}}$ is sampled from $\{\boldsymbol{f}'_j\}_{j=1}^n$ according to

$$\mathbf{P}_{\text{ns}}(\boldsymbol{f}'_{\mathcal{N}} = \boldsymbol{f}'_j) = \frac{\sum_{k=1}^n C_{jk}}{\sum_{k=1}^n \sum_{k'=1}^n C_{kk'}}.$$

The original node2vec algorithm solves for $(\hat{\mathbf{F}}, \hat{\mathbf{F}}')$ by minimizing (2.4) over $(\mathbf{F}, \mathbf{F}')$ using SGD. In this paper we use a matrix factorization approach, described in Section II-C, to find $(\hat{\mathbf{F}}, \hat{\mathbf{F}}')$.

B. Stochastic Blockmodel

The stochastic blockmodel (SBM) of [40] is one of the most popular generative models for network data. It often serves as a benchmark for evaluating community detection algorithms [41]. Our theoretical analysis of node2vec/DeepWalk is situated in the context of this model. We parametrize a K-blocks SBM in terms of two parameters $(\mathbf{B}, \mathbf{Z})$ where $\mathbf{B} = [b_{uu'}]$ is a symmetric matrix of blocks connectivity and $\mathbf{Z} \in \{0,1\}^{n\times K}$ is a matrix whose rows denote the block assignments for the nodes; we use $\tau(i) \in [K]$ to represent the community assignment for node i, i.e., the ith row of $\mathbf{Z}$ contains a single 1 in the $\tau(i)$th element and 0 everywhere else. Given $\mathbf{B}$ and $\mathbf{Z}$, the edges $a_{ii'}$ of $\mathcal{G}$ are *independent* Bernoulli random variables with $\mathbb{P}[a_{ii'} = 1] = B_{\tau(i),\tau(i')}$, i.e., the probability of connection between i and i' depends only on the communities assignment of i and i'. Denote by

$$\mathbf{P} = [p_{ii'}] = \mathbf{Z}\mathbf{B}\mathbf{Z}^\top \quad (2.5)$$

the matrix of edge probabilities. We denote a graph with adjacency matrix $\mathbf{A}$ sampled from a stochastic blockmodel as $\mathbf{A} \sim \mathrm{SBM}(\mathbf{B}, \mathbf{Z})$, and, for any stochastic blockmodel graph, we denote by n_k the number of vertices assigned to block k. We shall also assume, without loss of generality, that $\mathbf{Z}$ is ordered by blocks:

$$\mathbf{Z} := \begin{pmatrix} \mathbf{1}_{n_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{n_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{1}_{n_K} \end{pmatrix}. \quad (2.6)$$

In real-world applications the average degree of a network usually grows at a slower rate than $\Theta(n)$. To model this phenomenon we introduce a sparse parameter ρ_n that can vanish as $n \to \infty$. For ease of exposition, we use the following parametrization of $\mathbf{B}$ that is commonly used in the literature (see e.g., [42]).

Assumption 1: There exists a fixed $K \times K$ matrix $\mathbf{B}_0$ such that $\mathbf{B} = \rho_n \mathbf{B}_0$ with $\rho_n \gtrsim n^{-\beta}$ for some $\beta \in [0, 1)$.

The parameter ρ_n scales the edge probabilities in $\mathbf{B}$. As $\rho_n \gtrsim n^{-\beta}$, the average degree of the nodes in $\mathcal{G}$ grows at rate $n^{1-\beta}$ so that larger values of β lead to sparser network. It is well-known that, for sufficiently large n, if $\mathcal{G}$ satisfies Assumption 1 then $\mathcal{G}$ is connected with high probability (see e.g., Section 7.1 of [31]). Then $\mathbf{P} = \mathbf{Z}\mathbf{B}\mathbf{Z}^\top$ has a $K \times K$ block structure and thus has rank at most K.

C. Node2vec and Matrix Factorization

In general, for a fixed given embedding dimension $d < n$, minimization of the objective function in (2.4) leads to a non-convex optimization problem and the potential convergence of SGD into local minima makes the asymptotic analysis of $\hat{\mathbf{F}}$ quite complicated. Indeed, almost all existing results for non-convex optimization using gradient descent or SGD only guarantee convergence to a local minima provided that the initial estimate is sufficiently close to this local minima, see e.g., [43, Section 5] and [44]. We thus desire a different approach for finding $\hat{\mathbf{F}}$, namely one for which the form of $\hat{\mathbf{F}}$ is more readily apparent. One such approach is the use of matrix factorization. For example, in the context of word2vec embedding, [25] showed that minimization of (2.4) when $\mathbf{C}$ is a word-context matrix is equivalent to a matrix factorization problem on some *elementwise* non-linear transformation of $\mathbf{C}$ and that this transformation can be related to the notion of pointwise mutual information between the words. Motivated by this line of inquiry, we consider a formulation of node2vec wherein $\hat{\mathbf{F}}\hat{\mathbf{F}}'^\top$ is a low-rank approximation of some elementwise transformation $\tilde{\mathbf{M}}$ of $\hat{\mathbf{W}}$; recall that $\hat{\mathbf{W}}$ is the 1-step transition matrix for the canonical random walk on $\mathcal{G}$. We emphasize that this approach had been considered previously in [26] and recently by [27], [34]. The main contribution of our paper is in showing that this matrix factorization leads to consistent community recovery for stochastic blockmodel graphs.

We now describe the matrix $\tilde{\mathbf{M}}$. In the context of the word2vec algorithm, [25] showed that there exists some embedding dimension d such that the minimizer of (2.4) over $\mathbf{F} \in \mathbb{R}^{n\times d}$ and $\mathbf{F}' \in \mathbb{R}^{n\times d}$ satisfies

$$\hat{\mathbf{F}}\hat{\mathbf{F}}'^\top = \tilde{\mathbf{M}}(\mathbf{C}, \kappa) := \left[\log \frac{C_{ij}\left(\sum_{ij} C_{ij}\right)}{\kappa\left(\sum_i C_{ij}\right)\left(\sum_j C_{ij}\right)}\right]_{n\times n} \quad (2.7)$$

Using the same idea for our analysis of node2vec, we first fixed n and show that if the number of sampled random paths increases then $\tilde{\mathbf{M}}(\mathbf{C}, k)$ converges, elementwise, to a limiting matrix $\tilde{\mathbf{M}}_0$ defined below. Note that the entries of $\tilde{\mathbf{M}}_0$ can be interpreted as point-wise mutual information (PMI) between the nodes.

Theorem 1: Let n be fixed but arbitrary. Suppose $\mathcal{G}$ is a connected graph on n vertices and t_U is large enough such that the entries of $\sum_{t=t_L}^{t_U} \hat{\mathbf{W}}^t$ are all positive. Applying the node2vec sampling strategy introduced in Section II-A on $\mathcal{G}$ we have

$$\tilde{\mathbf{M}}(\mathbf{C},\kappa) \xrightarrow{\text{a.s.}} \tilde{\mathbf{M}}_0(\mathcal{G}, t_L, t_U, \kappa, L) := \log\left\{\frac{2|\mathbf{A}|}{\kappa\gamma}\sum_{t=t_L}^{t_U}(L-t)\mathbf{D}_{\mathbf{A}}^{-1}\hat{\mathbf{W}}^t\right\} \tag{2.8}$$

as the number of random paths $r = \sum_{i=1}^n r_i \to \infty$; recalling that $\hat{\mathbf{W}} = \mathbf{A}\mathbf{D}_{\mathbf{A}}^{-1}$. The convergence of $\tilde{\mathbf{M}}(\mathbf{C},\kappa)$ to $\tilde{\mathbf{M}}_0$ is element-wise and uniform over all entries of $\tilde{\mathbf{M}}(\mathbf{C},\kappa)$. Here $|\mathbf{A}|$ denotes the sum of the entries in $\mathbf{A}$ and the constant γ is defined as

$$\gamma := \frac{1}{2}(L - t_L - t_U)(t_U - t_L + 1).$$

To reduce notation clutter, we will henceforth drop the dependency of $\tilde{\mathbf{M}}_0$ on the parameters $\mathcal{G}, t_L, t_U, \kappa, L$. As the value of r is chosen purely for computational expediency, i.e., smaller values of r require sampling fewer random walks, we will thus take the conceptual view that $r \to \infty$ so that $\tilde{\mathbf{M}}(\mathbf{C},\kappa) \to \tilde{\mathbf{M}}_0$; note that $\tilde{\mathbf{M}}_0$ can be constructed explicitly from $\mathbf{A}$ without needing to sample any random walk. Combining (2.7) and Theorem 1, we have that, for any fixed n, there exists an embedding dimension d such that for $r \to \infty$, the matrices $\hat{\mathbf{F}}$ and $\hat{\mathbf{F}}'$ are exact factors for factorizing $\tilde{\mathbf{M}}_0$. Note that $\mathbf{D}_{\mathbf{A}}^{-1}\hat{\mathbf{W}}^t$ is symmetric for any $t \geq 1$ and hence $\tilde{\mathbf{M}}_0$ is *symmetric*.

In practice, one usually chooses $d \ll n$ to reduce the noise in the embeddings as well as combat the curse of dimensionality in downstream inference. Obviously, if $d < n$ then exact factors $(\hat{\mathbf{F}}, \hat{\mathbf{F}}')$ for factorizing $\tilde{\mathbf{M}}_0$ might no longer exist (see e.g., [25]). The requirement that $\hat{\mathbf{F}}\hat{\mathbf{F}}'^{\top} = \tilde{\mathbf{M}}_0$ is, however, both misleading and unnecessary. Indeed, as the observed graph is but a single *noisy* sample generated from some true but unobserved edge probabilities matrix $\mathbf{P}$, what we really want to recover is the factorization induced by $\mathbf{P}$. More specifically, replacing $\hat{\mathbf{W}}^t$ and $|\mathbf{A}|$ with $\mathbf{W}^t$ and $|\mathbf{P}|$ in $\tilde{\mathbf{M}}_0$, we define

$$\mathbf{M}_0 = \log\left\{\frac{2|\mathbf{P}|}{\kappa\gamma}\sum_{t=t_L}^{t_U}(L-t)\mathbf{D}_{\mathbf{P}}^{-1}\mathbf{W}^t\right\} \tag{2.9}$$

as the underlying-truth counterpart of $\tilde{\mathbf{M}}_0$; note that, similar to $\tilde{\mathbf{M}}_0$, we had dropped the parameters associated with $\mathbf{M}_0$ for simplicity of notations. Under the SBM setting, the true signal matrices $\mathbf{P}$ and $\mathbf{M}_0$ are both low-rank and hence an embedding dimension of $d = \text{rk}(\mathbf{M}_0) \ll n$ is sufficient to recover the factorization induced by $\mathbf{M}_0$.

To be more precise, recall from (2.6) that for stochastic blockmodel graphs, the matrix $\mathbf{P}$ has a $K \times K$ block structure. Thus both $\mathbf{W}^t$ and $\mathbf{D}_{\mathbf{P}}^{-1}\mathbf{W}^t$ also have $K \times K$ block structures. Equation (2.9) then implies that $\mathbf{M}_0$ also has a $K \times K$ block structure and hence $\text{rank}(\mathbf{M}_0) \leq K$. Most importantly, the $K \times K$ block structure of $\mathbf{M}_0$ is also sufficient for recovering the community structure in $\mathcal{G}$. We will show in Section III that the relative error, in the *row-wise maximum* norm, between $\tilde{\mathbf{M}}_0$ and $\mathbf{M}_0$ converges to 0 as $n \to \infty$. This convergence, together with results for perturbation of eigenspaces, implies the existence of an embedding dimension $d \leq K$ for which the $n \times d$ matrices $\hat{\mathbf{F}}$ and $\hat{\mathbf{F}}'$ obtained by factorizing $\tilde{\mathbf{M}}_0$ lead to exact recovery of the community structure in $\mathcal{G}$.

Remark 3: If $\mathbf{P}$ does not arise from a stochastic blockmodel graph then $\mathbf{M}_0$ need not have a low-rank structure. Nevertheless, we can still consider a rank-d approximation to $\mathbf{M}_0$ for some $d < \text{rk}(\mathbf{M}_0)$. Furthermore, as we will clarify in Section VI, the bound for $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\max}$ in Section III also holds for general edge independent random graphs, provided that the entries of $\mathbf{P}$ is reasonably homogeneous. Hence $\tilde{\mathbf{M}}_0$ has an approximate low-rank structure if and only if $\mathbf{M}_0$ also has an approximate low-rank structure.

In summary, motivated by the low-rank structure of $\mathbf{M}_0$ in the case of SBM graphs, we view the matrix factorization approach for node2vec as finding the best rank $d < n$ approximation $\hat{\boldsymbol{\mathcal{F}}} \cdot \hat{\boldsymbol{\mathcal{F}}'}^{\top}$ to $\tilde{\mathbf{M}}_0$ under Frobenius norm, i.e.,

$$\left(\hat{\boldsymbol{\mathcal{F}}}, \hat{\boldsymbol{\mathcal{F}}'}\right) = \underset{(\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{F}}')\in\mathbb{R}^{n\times d}.\mathbb{R}^{n\times d}}{\text{argmin}} \left\|\tilde{\mathbf{M}}_0 - \boldsymbol{\mathcal{F}} \cdot \boldsymbol{\mathcal{F}'}^{\top}\right\|_{\text{F}}. \tag{2.10}$$

The minimizer of (2.10) is obtained by truncating the SVD of $\tilde{\mathbf{M}}_0$. More specifically, let

$$\tilde{\mathbf{M}}_0 = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^{\top} \tag{2.11}$$

with a decreasing order of singular values in $\hat{\mathbf{\Sigma}}$. Then for a given $d \leq \text{rk}(\mathbf{M}_0)$, let

$$\hat{\boldsymbol{\mathcal{F}}} = \hat{\mathbf{U}}_d,\ \hat{\boldsymbol{\mathcal{F}}}' = \hat{\mathbf{V}}_d\hat{\mathbf{\Sigma}}_d \tag{2.12}$$

where $\hat{\mathbf{U}}_d \in \mathbb{R}^{n\times d}, \hat{\mathbf{V}}_d \in \mathbb{R}^{n\times d}$ are the first d columns of $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$, respectively, and $\hat{\mathbf{\Sigma}}_d \in \mathbb{R}^{d\times d}$ is the diagonal matrix containing the d largest singular values in $\hat{\mathbf{\Sigma}}$.

Remark 4: The appropriate embedding dimension d for factorizing $\tilde{\mathbf{M}}_0$ depends on knowing $\text{rank}(\mathbf{M}_0)$. but the convergence of $\tilde{\mathbf{M}}_0$ to that of $\mathbf{M}_0$ does not require knowing $\text{rank}(\mathbf{M}_0)$. For ease of exposition we will assume that $\text{rank}(\mathbf{M}_0)$ is known; in practice it can be estimated consistently using an eigenvalue thresholding procedure provided that $\mathbf{M}_0$ has a low-rank structure. Finally, in the context of SBM graphs and their degree-corrected variant, community recovery using $\hat{\mathbf{F}}$ also depends on knowing K. For simplicity, we also assume that K is known, noting that consistent estimates for K are provided in [45], [46].

III. Theoretical Analysis

A. Entry-Wise Concentration of $\hat{\mathbf{W}}^t$ and $\tilde{\mathbf{M}}_0$

Recall that $\hat{\boldsymbol{\mathcal{F}}}$ is obtained from the eigendecomposition of $\tilde{\mathbf{M}}_0$ while the true embedding is obtained from the eigendecomposition of $\mathbf{M}_0$ (see (2.9)). Therefore, before studying the community recovery using $\hat{\boldsymbol{\mathcal{F}}}$, we first study the convergence of $\tilde{\mathbf{M}}_0$ to $\mathbf{M}_0$. In particular, we derive concentration bounds for $\tilde{\mathbf{M}}_0 - \mathbf{M}_0$ in both Frobenius and infinity norms. These bounds are facilitated by the following Theorem 2 which provides a precise uniform bound for the entry-wise difference between the t-step transition matrix $\hat{\mathbf{W}}^t$ and $\mathbf{W}^t$ defined using the adjacency matrix $\mathbf{A}$ and the edge probabilities matrix $\mathbf{P}$, respectively.

Theorem 2: Let $\mathcal{G} \sim \text{SBM}(\mathbf{B}, \mathbf{Z})$ where $\mathbf{B}$ satisfies Assumption 1. We then have the following bounds.

1) *(Dense regime)* Suppose $\rho_n \asymp 1$. Then

$$\left\|\hat{\mathbf{W}} - \mathbf{W}\right\|_{\max} = \mathcal{O}_{\mathbb{P}}(n^{-1}), \tag{3.1}$$

$$\left\|\hat{\mathbf{W}}^2 - \mathbf{W}^2\right\|_{\max,\mathrm{diag}} = \mathcal{O}_{\mathbb{P}}(n^{-1}), \tag{3.2}$$

$$\left\|\hat{\mathbf{W}}^2 - \mathbf{W}^2\right\|_{\max,\mathrm{off}} = \mathcal{O}_{\mathbb{P}}\left(\frac{\log^{1/2} n}{n^{3/2}}\right), \tag{3.3}$$

Furthermore, for $t \geq 3$,

$$\left\|\hat{\mathbf{W}}^t - \mathbf{W}^t\right\|_{\max} = \mathcal{O}_{\mathbb{P}}\left(\frac{\log^{1/2} n}{n^{3/2}}\right), \tag{3.4}$$

2) *(Sparse regime)* Let $\rho_n \to 0$ with $\rho_n \gtrsim n^{-\beta}$ for some $\beta \in [0, 1)$. Then for $t \geq 4$ satisfying $\frac{t-3}{t-1} > \beta$ we have

$$\left\|\hat{\mathbf{W}}^t - \mathbf{W}^t\right\|_{\max} = \mathcal{O}_{\mathbb{P}}\left(\frac{\log^{1/2} n}{n^{3/2}\rho_n^{1/2}}\right). \tag{3.5}$$

In addition, if $0 \leq \beta < 1/2$ then

$$\left\|\hat{\mathbf{W}}^2 - \mathbf{W}^2\right\|_{\max,\mathrm{off}} = \mathcal{O}_{\mathbb{P}}\left(\frac{\log^{1/2} n}{n^{3/2}\rho_n}\right),$$

$$\left\|\hat{\mathbf{W}}^3 - \mathbf{W}^3\right\|_{\max} = \mathcal{O}_{\mathbb{P}}\left(\frac{\log^{1/2} n}{n^{3/2}\rho_n}\right). \tag{3.6}$$

Remark 5: Throughout this paper we assume that $t_L \geq 2$ instead of $t_L \geq 1$ as used in the original node2vec formulation. The rationale for this assumption is as follows. Recall the definition of $\tilde{\mathbf{M}}_0$ in (2.8). If we allow t to start from 1 in the sum $\sum_{t=t_L}^{t_U} (L-t) \cdot (\mathbf{D}_{\mathbf{A}}^{-1}\hat{\mathbf{W}}^t)$ then the term $\hat{\mathbf{W}}$ might lead to a convergence rate of $\tilde{\mathbf{M}}_0$ to $\mathbf{M}_0$ that is slower than that given in (3.7). For example in the dense regime (3.1) and (3.3) show that the entries of $\hat{\mathbf{W}} - \mathbf{W}$ are of larger magnitude than the entries of $\hat{\mathbf{W}}^t - \mathbf{W}^t$ for $t \geq 2$.

Before discussing the convergence rate of $\tilde{\mathbf{M}}_0$ to $\mathbf{M}_0$ we first find a value of t_U such that, for large values of n, $\tilde{\mathbf{M}}_0$ is well defined with high probability. We note that the entries of $\{\mathbf{W}^t\}_{t\geq 1}$ are uniformly of order $\Theta(n^{-1})$. Then, under the dense regime, $t = 2$ is sufficient to guarantee that all the off-diagonal entries of $\hat{\mathbf{W}}^t$ are uniformly of order $\Omega(n^{-1} - n^{-3/2}\log^{1/2} n) = \Omega(n^{-1})$ with high probability (c.f. (3.2)) while $t = 3$ is sufficient to guarantee that all entries of $\hat{\mathbf{W}}^t$ are of order $\Omega(n^{-1})$ with high probability (c.f. (3.3)). If we are under the sparse regime with $\beta < 1/2$ then these same values of $t \geq 2$ are still sufficient to guarantee that the entries of $\hat{\mathbf{W}}^t$ are of order $\Omega(n^{-1})$ (c.f. (3.5) and (3.6)). Finally, if we are under the sparse regime with $\beta \geq 1/2$ then choosing $t \geq 4$ with $\frac{t-3}{t-1} > \beta$ is sufficient to guarantee that the entries $\hat{\mathbf{W}}^t$ are uniformly of order $\Omega(n^{-1} - n^{-3/2}\rho_n^{-1/2}\log^{1/2} n) = \Omega(n^{-1})$ with high probability. Now recall that the matrix $\tilde{\mathbf{M}}_0$ is of the form

$$\log\left\{\frac{2|\mathbf{A}|}{\kappa\gamma}\sum_{t=t_L}^{t_U}(L-t)\mathbf{D}_{\mathbf{A}}^{-1}\hat{\mathbf{W}}^t\right\}$$

We therefore have, for $t_U \geq 3$ in the dense regime, $t_U \geq 2$ in the not too sparse regime of $\beta < 1/2$, or for $\frac{t_U-3}{t_U-1} > \beta$ in general, that the entries of the inner sum are bounded away from 0 with high probability. For the dense regime, the condition can be further relaxed to $t_U \geq 2$, as a dense graph has a diameter of 2 and thus all entries of $\hat{\mathbf{W}}^2$ are uniformly larger than 0 with high probability; see Theorem 10.10 in [31]. Therefore, with high probability, the elementwise logarithm is well-defined for all entries of $\tilde{\mathbf{M}}_0$. Given the existence of $\tilde{\mathbf{M}}_0$, the following result shows the convergence rate of $\tilde{\mathbf{M}}_0$ to $\mathbf{M}_0$.

Theorem 3: Suppose $\mathcal{G} \sim \mathrm{SBM}(\mathbf{B}, \boldsymbol{\Theta})$ satisfies Assumption 1, and $t_U \geq t_L \geq 2$ where t_L is chosen as described above. Then $\tilde{\mathbf{M}}_0$ is well-defined with high probability. Denote

$$\Delta = \max\{\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}, \|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\infty}\}.$$

We then have the following bounds.

1) *(Dense regime)* Let $\rho_n \asymp 1$. Then for $t_L \geq 2$ we have

$$\Delta = \mathcal{O}_{\mathbb{P}}\left(n^{1/2}\log^{1/2} n\right). \tag{3.7}$$

2) *(Sparse regime)* Let $\rho_n \to 0$ with $\rho_n \gtrsim n^{-\beta}$ for some $\beta \in [0, 1)$. Then for t_L satisfying $\frac{t_L-3}{t_L-1} > \beta$ we have

$$\Delta = \mathcal{O}_{\mathbb{P}}\left(n^{1/2}\rho_n^{-1/2}\log^{1/2} n\right). \tag{3.8}$$

In addition, if $0 \leq \beta < 1/2$ then for $t_L \geq 2$ we have

$$\Delta = \mathcal{O}_{\mathbb{P}}\left(n^{1/2}\rho_n^{-1}\log^{1/2} n\right). \tag{3.9}$$

In both regimes we have $\|\mathbf{M}_0\|_{\mathrm{F}} = \Theta(n)$ and $\|\mathbf{M}_0\|_{\infty} = \Theta(n)$.

In summary, as β increases (equivalently, as ρ_n decreases) so that the graph $\mathcal{G}$ becomes sparser, we could (1) still guarantee the existence of $\tilde{\mathbf{M}}_0$ when t_U is sufficiently large, and (2) control the convergence rate of $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}$ and $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\infty}$ relative to $\|\mathbf{M}_0\|_{\mathrm{F}}$ and $\|\mathbf{M}_0\|_{\infty}$, respectively, through increasing t_L.

B. Subspace Perturbations and Exact Recovery

Theorem 3 implies that $\tilde{\mathbf{M}}_0$ is close to $\mathbf{M}_0$ under both Frobenius and infinity norms, i.e., $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\star}/\|\mathbf{M}_0\|_{\star} = o_{\mathbb{P}}(1)$ for $\star \in \{F, \infty\}$ and sufficiently large n. Now, by (2.6), $\mathbf{M}_0$ has a $K \times K$ block structure and hence $\mathrm{rk}(\mathbf{M}_0) \leq K$. Furthermore the eigenvectors of $\mathbf{M}_0$ associated with its non-zero eigenvalues is sufficient for recovering the community assignments induced by $\mathbf{Z}$. The following result, which follows from bounds for $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\infty}$ given in Theorem 3 together with perturbations bounds for invariant subspaces using $2 \to \infty$ norm [36], shows that the embedding $\hat{\boldsymbol{\mathcal{F}}}$ given by the leading eigenvectors of $\tilde{\mathbf{M}}_0$ is *uniformly* close to that of the leading eigenvectors of $\mathbf{M}_0$. Therefore K-means or K-medians clustering on the rows of $\hat{\boldsymbol{\mathcal{F}}}$ will recover the community membership for *every* node, i.e, attain strong or exact recovery of $\mathbf{Z}$.

Theorem 4: Under the condition of Theorem 3, let $\hat{\mathbf{U}}\hat{\boldsymbol{\Sigma}}\hat{\mathbf{U}}^{\top}$ and $\mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^{\top}$ be the eigen-decomposition of $\tilde{\mathbf{M}}_0$ and $\mathbf{M}_0$, respectively. Let $d = \mathrm{rk}(\mathbf{M}_0)$ and note that $\mathbf{U}$ is a $n \times d$ matrix. Let $\hat{\boldsymbol{\mathcal{F}}} = \hat{\mathbf{U}}_d$ be the matrix formed by the columns of $\hat{\mathbf{U}}$ corresponding to the d largest-in-magnitude eigenvalues of $\tilde{\mathbf{M}}_0$. For a $n \times d$ matrix $\mathbf{Z}$ with rows $Z_1, Z_2, \dots, Z_n$ let $\|\mathbf{Z}\|_{2\to\infty}$ denote the maximum ℓ_2 norms of the $\{Z_i\}$, i.e.,

$$\|\mathbf{Z}\|_{2\to\infty} = \max_i \|Z_i\|_2.$$

We then have the following results.

i) *(Dense regime)* Let $\rho_n \asymp 1$. Then for $t_L \geq 2$ we have

$$\min_{\mathbf{T}\in\mathbb{O}_d} \left\|\hat{\boldsymbol{\mathcal{F}}}\mathbf{T} - \mathbf{U}\right\|_{\mathrm{F}} = \mathcal{O}_{\mathbb{P}}\left(\frac{\log^{1/2} n}{n^{1/2}}\right)$$

$$\min_{\mathbf{T}\in\mathbb{O}_d} \left\|\hat{\boldsymbol{\mathcal{F}}}\mathbf{T} - \mathbf{U}\right\|_{2\to\infty} = \mathcal{O}_{\mathbb{P}}\left(\frac{\log^{1/2} n}{n}\right). \tag{3.10}$$

ii) *(Sparse regime)* Let $\rho_n \to 0$ with $\rho_n \gtrsim n^{-\beta}$ for some $\beta \in [0, 1/2)$. If $t_L \geq 2$, we have

$$\min_{\mathbf{T}\in\mathbb{O}_d} \left\|\hat{\boldsymbol{\mathcal{F}}}\mathbf{T} - \mathbf{U}\right\|_{\mathrm{F}} = \mathcal{O}_{\mathbb{P}}\left(\frac{\log^{1/2} n}{n^{1/2}\rho_n}\right)$$

$$\min_{\mathbf{T}\in\mathbb{O}_d} \left\|\hat{\boldsymbol{\mathcal{F}}}\mathbf{T} - \mathbf{U}\right\|_{2\to\infty} = \mathcal{O}_{\mathbb{P}}\left(\frac{\log^{1/2} n}{n\rho_n}\right) \tag{3.11}$$

iii) *(Sparse regime)* Let $\rho_n \to 0$ with $\rho_n \gtrsim n^{-\beta}$ for some $\beta \in [0, 1)$. If $\frac{t_L - 3}{t_L - 1} > \beta$, we have,

$$\min_{\mathbf{T}\in\mathbb{O}_d} \left\|\hat{\boldsymbol{\mathcal{F}}}\mathbf{T} - \mathbf{U}\right\|_{\mathrm{F}} = \mathcal{O}_{\mathbb{P}}\left(\frac{\log^{1/2} n}{(n\rho_n)^{1/2}}\right)$$

$$\min_{\mathbf{T}\in\mathbb{O}_d} \left\|\hat{\boldsymbol{\mathcal{F}}}\mathbf{T} - \mathbf{U}\right\|_{2\to\infty} = \mathcal{O}_{\mathbb{P}}\left(\frac{\log^{1/2} n}{n\rho_n^{1/2}}\right). \tag{3.12}$$

Given the above convergence rates, clustering the rows of $\hat{\boldsymbol{\mathcal{F}}}$ using either K-means or K-medians will, with high probablity, recover the membership of *every* node in $\mathcal{G}$.

Remark 6: Settings (ii) and (iii) in Theorem 4 both consider the sparse regime but setting (ii) focuses on the case where $\rho_n = \omega(n^{-1/2})$ and exact recovery is achieved whenever $t_L \geq 2$ while setting (iii) considers the more general scenario of $\rho_n = \omega(n^{-\beta})$ for any fixed but arbitrary $\beta < 1$. We note that for ease of exposition we had imposed $\frac{t_L - 3}{t_L - 1} > \beta$ for setting (iii) but this condition can be relaxed to

$$\frac{t_L - 2}{t_L} > \beta, \tag{3.13}$$

under which we still have $\tilde{\mathbf{M}}_0$ is well-defined with high probability, and have a more complicated bound of

$$\min_{\mathbf{T}\in\mathbb{O}_d} \left\|\hat{\boldsymbol{\mathcal{F}}}\mathbf{T} - \mathbf{U}\right\|_{2\to\infty} \precsim \mathcal{O}_{\mathbb{P}}\left\{\frac{\log^{1/2} n}{n^{3/2}\rho_n^{1/2}} + (n\rho_n)^{-t_L/2}\right\}$$

(see (B.71)). The above bound is still sufficient to guarantee that running K-means or K-medians on the rows of $\hat{\boldsymbol{\mathcal{F}}}$ will recover the membership of every node in $\mathcal{G}$ with high probability; see Section B.4 in the Supplementary File, available online, for a rigorous proof.

A recent preprint [34] which appeared on arXiv after the first version of our paper also studied community recovery using SVD-based DeepWalk/node2vec and they have a similar requirement for t_L as (3.13); see (3.1) in [34]. For comparison we note that [34] only derived the convergence rate of $\hat{\boldsymbol{\mathcal{F}}}$ under Frobenius norm, and thereby prove a *weak* recovery result which allows at most $o(n^{1/2})$ nodes to be misclassified. In contrast the max-norm concentration of $\hat{\mathbf{W}}^t$ in Theorem 2 helps us derive a $2 \to \infty$ norm convergence for $\hat{\boldsymbol{\mathcal{F}}}$, based on which we achieved the much stronger exact recovery (i.e., there are no mis-classified nodes). Finally we conjecture that (3.13) for t_L is sufficient but not necessary. Our simulation results in Section IV agree with this conjecture and we leave its verification for future work.

Remark 7 (Extension to DCSBM): The exact recovery results in Theorem 4 can also be extended to the case of degree-corrected SBM graphs [47], [48], [49]. Recall that the edge probabilities for a DCSBM is $\mathbf{P} = \boldsymbol{\Theta}\mathbf{Z}\mathbf{B}\mathbf{Z}^{\top}\boldsymbol{\Theta}$ where $\boldsymbol{\Theta} = \mathrm{diag}(\theta_1, \dots, \theta_n)$ is the diagonal matrix containing the degree-correction parameters. DCSBM allows heterogeneous edge probabilities within each community and thus yields a more flexible model in comparison with SBM. Section A.4 and A.5 in the Supplementary File, available online, demonstrates how to extend the technical derivations for Theorem 4 to the DCSBM case provided that the $\{\theta_i\}$ are sufficiently homogeneous, i.e., that $\max_i \theta_i / \min_i \theta_i = \mathcal{O}(1)$.

IV. Simulation

We now present numerical experiments for the matrix factorization perspective of node2vec/DeepWalk. These experiments complement our theoretical results in Section III and illustrate the interplay between the sparsity of the graphs, the choice of window sizes, and their combined effects on the nodes embedding.

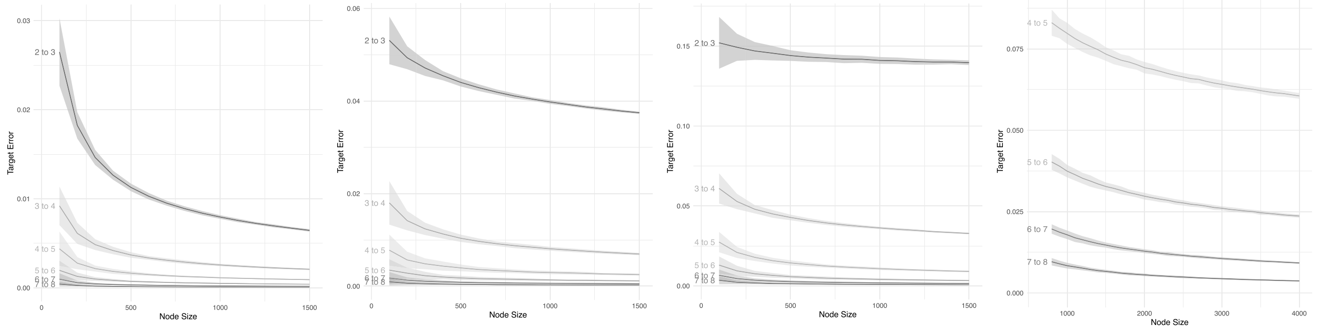
A. Error Bounds for $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}$

We first compare the large-sample empirical behavior of $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}$ against the theoretical bounds given in Theorem 3. We shall simulate undirected graphs generated from a 2-blocks SBM with parameters

$$\mathbf{B}(\rho_n) := \begin{pmatrix} 0.8\rho_n & 0.3\rho_n \\ 0.3\rho_n & 0.8\rho_n \end{pmatrix}, \quad \boldsymbol{\pi} = (0.4, 0.6), \tag{4.1}$$

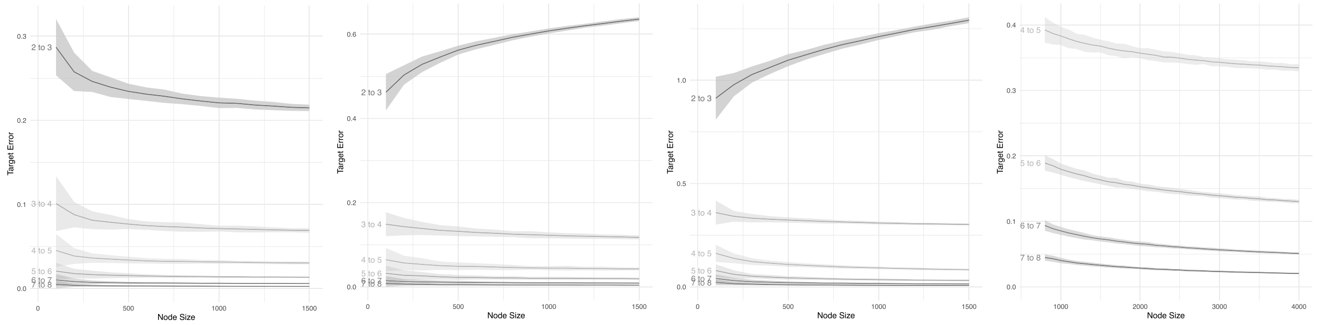
and sparsity $\rho_n \in \{1, 3n^{-1/3}, 3n^{-1/2}, 3n^{-2/3}\}$. While this two blocks setting is quite simple, it nevertheless displays the effect of the sparsity ρ_n and the window size (t_L, t_U) on the upper bound for $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}$.

For each value of n and sparsity ρ_n we run 100 independent replications where, in each replicate, we generate $\mathcal{G} \sim \mathrm{SBM}(\mathbf{B}(\rho_n), \boldsymbol{\Theta}_n)$, and calculate $\tilde{\mathbf{M}}_0$ for different choices of (t_L, t_U). In particular, we consider two types of window size, namely $t_U = t_L + 1$ and $t_U = t_L + 3$. While $t_U = t_L + 1$ is not commonly used in practice, for simulation purpose this choice clearly show the effects of the random walks' length t on the error $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}$. In contrast, the choice $t_U = t_L + 3$ is more realistic but also partially obfuscates the effect of t on



(a) $\rho_n = 1$ (b) $\rho_n = 3n^{-1/3}$ (c) $\rho_n = 3n^{-1/2}$ (d) $\rho_n = 15n^{-2/3}$

Fig. 1. Sample means and 95% empirical confidence intervals for $\varepsilon_1\left(\tilde{\mathbf{M}}_0\right)$ based on 100 Monte Carlo replicates under different settings of n, ρ_n and (t_L, t_U), with $t_U - t_L = 1$. The X-axis represents different n and Y-axis represents the relative error. The labels on the left-hand side of curves/empirical confidence bands denote different choices of values for (t_L, t_U).



(a) $\rho_n = 1$ (b) $\rho_n = 3n^{-1/3}$ (c) $\rho_n = 3n^{-1/2}$ (d) $\rho_n = 15n^{-2/3}$

Fig. 2. Sample means and 95% empirical confidence intervals for $\varepsilon_2\left(\tilde{\mathbf{M}}_0\right)$ based on 100 Monte Carlo replicates under different settings of n, ρ_n and (t_L, t_U), with $t_U - t_L = 1$. The X-axis represents different n and Y-axis represents the relative error. The labels on the left-hand side of curves/empirical confidence bands denote different choices of values for (t_L, t_U).

$\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}$. Recall that, from the discussion prior to Theorem 3, sparser values of ρ_n require larger values of t_U to guarantee that $\tilde{\mathbf{M}}_0$ is well-defined. The choices for $(\rho_n, n, (t_L, t_U))$ in the simulations are summarized below.

- If $\rho_n \geq 3n^{-1/2}$ then $n \in \{100, 200, 300, \ldots 1500\}$. We chose $2 \leq t_L \leq 7$ when $t_U = t_L + 1$ and chose $2 \leq t_L \leq 5$ when $t_U = t_L + 3$.
- If $\rho_n = 3n^{-2/3}$ then $n \in \{800, 900, \ldots, 4000\}$. We chose $4 \leq t_L \leq 7$ when $t_U = t_L + 1$ and $3 \leq t_L \leq 5$ when $t_U = t_L + 3$.

We calculate two relative error criteria for the generated $\tilde{\mathbf{M}}_0$, namely

$$\varepsilon_1(\tilde{\mathbf{M}}_0) = \frac{\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}}{\|\mathbf{M}_0\|_{\mathrm{F}}} \text{ and } \varepsilon_2(\tilde{\mathbf{M}}_0) = \frac{\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}}{n^{1/2}\rho_n^{-1/2}\log^{1/2} n}.$$

We expect that, as n increases, the first criterion converges to 0 while the second criterion remains bounded.

The results of our experiments are presented in Figs. 1 and 2 in the main text and Figs. D1 and D2 in the Supplementary File, available online. More specifically, Figs. 1 and D1 show the mean and 95% empirical confidence intervals of the empirical errors $\varepsilon_1(\tilde{\mathbf{M}}_0)$ across 100 Monte Carlo replicates under different simulation settings, where $t_U - t_L$ is set to 1 and 3, respectively. Similarly, Figs. 2 and D2 illustrate the empirical errors $\varepsilon_2(\tilde{\mathbf{M}}_0)$.

Relative Error 1 $(\varepsilon_1(\tilde{\mathbf{M}}_0))$*:* We first confirm the convergence of $\varepsilon_1(\tilde{\mathbf{M}}_0)$ to 0. Figs. 1 and D1 show the means and 95% confidence intervals for $\varepsilon_1(\tilde{\mathbf{M}}_0)$ based on 100 Monte Carlo replicates for different values of $\rho_n, (t_L, t_U)$. These figures indicate the following general patterns as predicted by the theoretical results in Theorem 3.

- The error $\varepsilon_1(\tilde{\mathbf{M}}_0)$ is smallest in the dense case and deteriorates as the sparsity factor ρ_n decreases.
- The error also depends on (t_L, t_U) with larger values of $t_U - t_L$ leading to smaller $\varepsilon_1(\tilde{\mathbf{M}}_0)$
- If the window size is too small, e.g., $(t_L, t_U) = (2, 3)$ or $(t_L, t_U) = (2, 5)$, then $\tilde{\mathbf{M}}_0$ is often times not well-defined.

Relative Error 2 $(\varepsilon_2(\tilde{\mathbf{M}}_0))$*:* Figs. 1 and D1 corroborate our theoretical results in Section III. Nevertheless, there are two additional questions we should consider. The first is whether or not the bound $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}} = \mathcal{O}_{\mathbb{P}}\left(n^{1/2}\rho_n^{-1/2}\log^{1/2} n\right)$ in Theorem 3 is tight and, if it is tight, the second is whether or not the condition $\frac{t_L-3}{t_L-1} > \beta$ is necessary to achieve this rate. Analogous to the previous two figures, Figs. 2 and D2 show the means and 95% empirical confidence intervals for the relative

TABLE II
PROPORTIONS OF TIMES THAT SGD-BASED AND SVD-BASED NODE2VEC VARIANTS PERFECTLY RECOVER ALL NODES' MEMBERSHIPS OVER 100 MONTE CARLO REPLICATES, UNDER DIFFERENT SETTINGS OF n, ρ_n AND t_U

n	SVD-based node2vec		Original node2vec	
	$t_U = 5$	$t_U = 8$	$t_U = 5$	$t_U = 8$
600	1.00	1.00	1.00	1.00
900	1.00	1.00	1.00	1.00
1500	1.00	1.00	1.00	1.00

n	SVD-based node2vec		original node2vec	
	$t_U = 5$	$t_U = 8$	$t_U = 5$	$t_U = 8$
600	0.44	0.42	0.01	0.05
900	0.62	0.63	0.07	0.11
1500	0.90	0.90	0.57	0.28

The graphs are generated from B(ρn) with sparsity ρn =3$n^{-1/3}$ (left table) and ρn =3$n^{-1/2}$ (right table).

error $\varepsilon_2(\tilde{\mathbf{M}}_0)$ over 100 Monte Carlo replicates for different values of ρ_n and (t_L, t_U). From these simulations we can answer the above questions as follows.

- If $\rho_n \gtrsim n^{-\beta}$ is such that $\beta \leq \frac{t_L-3}{t_L-1}$ then $\varepsilon_2(\tilde{\mathbf{M}}_0)$ appears to converge to a constant as n increases. There is thus evidence that the rate $n^{1/2}\rho_n^{-1/2}\log^{1/2} n$ for $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}$ is optimal. Nevertheless, if t_L is large relative to ρ_n, e.g., $\rho_n \in \{3n^{-1/3}, 3n^{-1/2}\}$ and $t_L \geq 6$, then $\varepsilon_2(\tilde{\mathbf{M}}_0)$ appears to converges to 0 which suggests that for a fixed β the error rate for $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}$ can be smaller than $n^{1/2}\rho_n^{-1/2}\log^{1/2} n$; this might be due to the convergence of $\hat{\mathbf{W}}^t$ and $\mathbf{W}^t$ towards the stationary distributions as t increases.
- For cases such as $(t_L, t_U) \in \{(3,4),(3,6)\}$ and $\rho_n = 3n^{-1/2}$ or $(t_L, t_U) \in \{(4,5),(3,6)\}$ and $\rho_n = 15n^{-2/3}$, the t_L's do not satisfy $\frac{t_L-3}{t_L-1} > \beta$. Nevertheless, $\varepsilon_2(\tilde{\mathbf{M}}_0)$ still appears to converge to a constant as n increases. This suggests that $\frac{t_L-3}{t_L-1} > \beta$ is sufficient but possibly not necessary for the bound in (3.8) to hold. On the other hand, for fixed n and ρ_n, the error $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}$ generally decreases as $t_U - t_L$ increases.
- Finally if $(t_L, t_U) \in \{(2,3),(2,5)\}$ and $\rho_n \in \{3n^{-1/3}, 3n^{-1/2}\}$ then $\varepsilon_2(\tilde{\mathbf{M}}_0)$ increases with n. This supports the claim in Theorem 3 of a phase transition for the error rate of $\|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|_{\mathrm{F}}$ as t_L increases.

In summary our simulation results support the conclusion of Theorem 3. In particular, the error rate in Theorem 3 is sharp and the condition $\frac{t_L-3}{t_L-1} > \beta$ is sufficient but perhaps not necessary.

B. Exact Recovery of Community Structure

Theorem 4 together with Remark 6 showed that $\hat{\boldsymbol{\mathcal{F}}}$ combined with K-means/medians can correctly recover the memberships of all nodes in a SBM with high probability. We demonstrate this result for two-block SBMs with block probabilities being either $\mathbf{B}(\rho_n)$ as given in (4.1) or

$$\mathbf{B}^{\natural}(\rho_n) := \begin{pmatrix} 0.3\rho_n & 0.8\rho_n \\ 0.8\rho_n & 0.3\rho_n \end{pmatrix}.$$

Note that $\mathbf{B}(\rho_n)$ and $\mathbf{B}^{\natural}(\rho_n)$ corresponds to an assortative and a dis-assortative structure, respectively. Given specific setting of $\mathbf{B}, n, \rho_n$, we randomly sample 100 graphs where each vertex is randomly assigned to one of the two blocks with equal probability and evaluate the membership recovery performances of the original node2vec [19] (based on SGD) and node2vec using matrix factorization (as described [26], [27], [34] and this paper) followed by clustering using K-means. We set the window sizes to $t_U \in \{5, 8\}$ and choose $\kappa = 5$ and $L = 200$. For the original node2vec we also set $t_L = 1$ as the default and $r_1 = \cdots = r_n = 200$, while for the SVD-based node2vec we set $t_L = t_U - 3$. We report in Tables II and III the proportions of times for the 100 simulated graphs that these two variants of the node2vec algorithm correctly recover the memberships of *all* nodes. More specifically, let the true and estimated community labels be denoted by $\{\tau(i)\}_{i=1}^n$ and $\{\hat{\tau}(i)\}_{i=1}^n$, respectively. The accuracy of $\hat{\tau}$ is defined as

$$\text{Accuracy}(\hat{\tau}) = \min_{\xi} \frac{\#\{i | \xi(\hat{\tau}(i)) \neq \tau(i)\}}{n} \qquad (4.2)$$

where the minimization is over all permutations ξ of $\{1, \ldots, K\}$. Thus $\hat{\tau}$ is an exact recovery if Accuracy$(\hat{\tau}) = 0$.

The numerical results in Tables II and III show that as n increases, both the original and SVD-based node2vec are more likely to perfectly recover memberships of all nodes in the graph, under all different settings of $\rho_n, \mathbf{B}, t_U$. Furthermore, the frequency of exact recovery for $\rho_n = 3n^{-1/3}$ is considerably higher than that for $\rho_n = 3n^{-1/2}$. This is consistent with the results in Theorem 4 as a smaller magnitude for ρ_n results in a slower convergence rate for $\hat{\boldsymbol{\mathcal{F}}}$ under both the Frobenius and $2 \to \infty$ norms. In addition, the exact recovery performance of SVD-based node2vec when $\rho_n \asymp n^{-1/2}$ and $(t_L, t_U) = (2, 5)$ suggests that the t_L threshold for Theorem 4 in (3.13) is possibly not sharp as $\frac{t_L-2}{t_L} = 0 < \beta = 1/2$. Finally we note that the SVD-based node2vec has better empirical performance than the original node2vec in these experiments as well as in the experiments for three-block SBMs and DCSBMs in Section IV-C. This is consistent with the discussion in Section II. Indeed, the entries of $\tilde{\mathbf{M}}_0$ are the limit of those for the original node2vec when the number of sampled paths $r \to \infty$ and furthermore $\tilde{\mathbf{M}}_0$ has an approximately low-rank structure as n increases. In other words, at least for SBM and DCSBM graphs, the original node2vec can be viewed as a computationally efficient approximation to the SVD-based embeddings of $\tilde{\mathbf{M}}_0$.

C. Embedding Performance

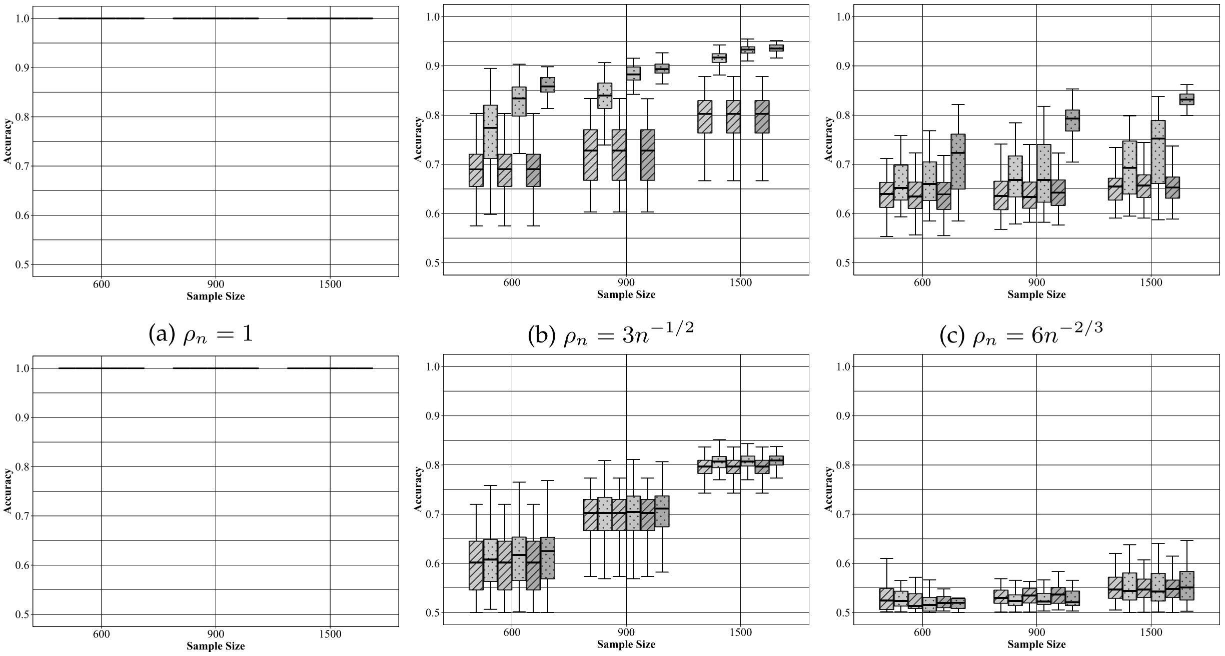
In this section we perform more numerical experiments to take a closer look at the finite-sample performance of community detection, using both the original and SVD-based node2vec embeddings. We consider both three-blocks SBM and three-blocks DCSBM according to the following parameter settings.

TABLE III
PROPORTIONS OF TIMES THAT SGD-BASED AND SVD-BASED NODE2VEC VARIANTS PERFECTLY RECOVER ALL NODES' MEMBERSHIPS OVER 100 MONTE CARLO REPLICATES, UNDER DIFFERENT SETTINGS OF n, ρ_n AND t_U

n	SVD-based node2vec		original Node2vec	
	$t_U = 5$	$t_U = 8$	$t_U = 5$	$t_U = 8$
600	1.00	1.00	1.00	0.50
900	1.00	1.00	1.00	0.95
1500	1.00	1.00	1.00	1.00

n	SVD-based node2vec		original Node2vec	
	$t_U = 5$	$t_U = 8$	$t_U = 5$	$t_U = 8$
600	0.24	0.30	0.00	0.00
900	0.67	0.69	0.00	0.00
1500	0.90	0.90	0.13	0.37

The graphs are generated from B(ρn) with sparsity $\rho n = 3n^{-1/3}$ (left table) and $\rho n = 3n^{-1/2}$ (right table).



(a) $\rho_n = 1$ (b) $\rho_n = 3n^{-1/2}$ (c) $\rho_n = 6n^{-2/3}$

(d) $\rho_n = 1$ (e) $\rho_n = 3n^{-1/2}$ (f) $\rho_n = 6n^{-2/3}$

Fig. 3. Community detection accuracy of node2vec followed by K-means for SBM graphs. The boxplots of the accuracy for each value of n, ρ_n and t_U are based on 100 Monte Carlo replications. Boxplots with the slash pattern (resp. dot pattern) summarized the results for the original (resp. SVD-based) node2vec. Different colors (yellow, green, blue) represent the algorithms implemented for different choices of $t_U \in \{5, 6, 8\}$. The first and second row plot the results when the block probabilities for the SBM is $\mathbf{B}_1$ and $\mathbf{B}_2$, respectively.

Stochastic Blockmodel: The three-blocks SBMs have block probabilities being either

$$\mathbf{B}_1 = \begin{pmatrix} 0.8 & 0.4 & 0.3 \\ 0.4 & 0.7 & 0.5 \\ 0.3 & 0.5 & 0.9 \end{pmatrix} \text{ or } \mathbf{B}_2 = \begin{pmatrix} 0.8 & 0.5 & 0.5 \\ 0.5 & 0.8 & 0.5 \\ 0.5 & 0.5 & 0.8 \end{pmatrix}, \tag{4.3}$$

and block assignment probabilities $\boldsymbol{\pi} = (0.3, 0.3, 0.4)$.

Degree-Corrected Stochastic Blockmodel: The DCSBM is a direct generalization of SBMs with the only difference being that each node i has a degree-correction parameter θ_i and that the probability of connection between nodes i and j is

$$p_{ij} = \theta_i \theta_j B_{\tau(i)\tau(j)}$$

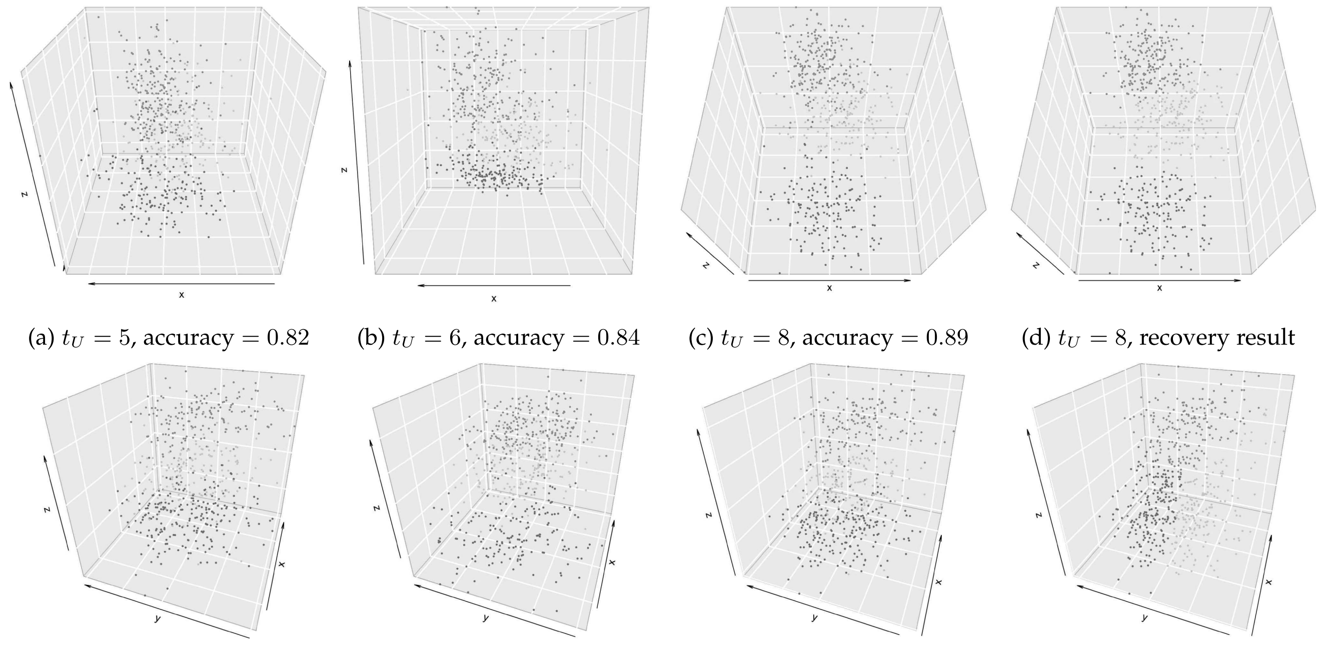
instead of $p_{ij} = B_{\tau(i)\tau(j)}$ as in the case of SBMs. For more on DCSBMs and their inference, see [47], [48], [49]. We generate the degree correction parameters θ_i as

$$\theta_i = |Z_i| + 1 - (2\pi)^{-1/2},\ Z_1, \ldots, Z_n \overset{\text{iid}}{\sim} \mathcal{N}(0, 0.25) \tag{4.4}$$

This procedure for generating θ_i is the same as that in [49].

For each value of n and ρ_n we perform 100 Monte Carolo replications where we generate graphs from the above SBM and DCSBM models and test both the original node2vec and the SVD-based node2vec with $t_U = 5, 6, 7$. Other settings of the node2vec algorithms are similar to Section IV-B and the accuracy is measured via (4.2). The results for the SBM graphs are presented in Fig. 3 of the main paper while those for the DCSBM graphs are presented in Fig. D3 of the Supplementary File, available online. We now summarize the main trend in these figures.

- We get exact recovery when $\rho_n = 1$ in both Fig. 3 and Fig. D3, thereby illustrating that the condition $t_L \geq 2$ in Theorem 4 is sufficient for exact recovery in the dense regime.
- When $\rho_n \to 0$ faster (i.e., the network is more sparse), we need a larger n to achieve the same level of accuracy. This is consistent with Theorem 4 as the convergence rate for $\hat{\boldsymbol{\mathcal{F}}}$ depends on $n\rho_n$.



(a) $t_U = 5$, accuracy $= 0.82$ (b) $t_U = 6$, accuracy $= 0.84$ (c) $t_U = 8$, accuracy $= 0.89$ (d) $t_U = 8$, recovery result

(e) $t_U = 5$, accuracy $= 0.58$ (f) $t_U = 6$, accuracy $= 0.57$ (g) $t_U = 8$, accuracy $= 0.60$ (h) $t_U = 8$, recovery result

Fig. 4. Visualizations of the SVD-based node2vec embeddings (first row) and original node2vec embeddings (second row) with different choices of t_U. The embeddings are for a single realization of a SBM graph on $n = 600$ vertices with block probabilities matrix $\mathbf{B}_1$ (see (4.3)), sparsity $\rho_n = 3n^{-1/2}$, and block assignment probabilities $\boldsymbol{\pi} = (0.3, 0.3, 0.4)$. The embeddings in panels (a)–(c) and (e)–(g) are colored using the true membership assignments while the embeddings in panels (d) and (h) are colored using the K-means clustering. Accuracy of the recovered memberships for the different embeddings followed by K-means clustering are also reported for panels (a)–(c) and (e)–(g).

- When $\mathbf{B} = \mathbf{B}_2$ the original node2vec and SVD-based node2vec have very similar accuracy and thus our theoretical analysis of SVD-based node2vec closely reflects the performance of the original node2vec.
- When $\mathbf{B} = \mathbf{B}_1$ the SVD-based node2vec has higher accuracy compared to the original node2vec. However, the embeddings generated by these algorithms are still quite similar. A plausible reason for why the original node2vec has lower accuracy is because the downstream K-means clustering is sub-optimal for these embeddings. We illustrate this by visualizing the embeddings for two realizations of the SBM and DCSBM graphs where we set $\rho_n = 3n^{-1/2}$, $n = 600$. These visualizations (see Fig. 4 in the main text and Fig. D4 in the Supplementary File, available online) provide us with the following intuitions: (i) the original and SVD-based node2vec variants yield similar embedding patterns; (ii) for SVD-based node2vec, increasing the window size could help separate nodes from different communities and thereby improve the community detection accuracy; (iii) although the embeddings appear similar, K-means clustering yields more accurate membership recovery for the SVD-based node2vec compared to the original SVD-based node2vec embeddings. For example, comparing panels (c) and (d) in Fig. 4 we see that K-means clustering recovers most of the membership assignments for embeddings from the SVD-based node2vec. In contrast, panels (g) and (h) in Fig. 4 show that K-means clustering is less accurate for embeddings from the original node2vec. Indeed, if we replace K-means with Gaussian mixtures model (GMM) [50], [51] in panels (g) and (h) of Fig. 4 we increase the clustering accuracy from 0.6 to 0.84 which is close to that of 0.89 for the SVD-based node2vec (see Fig. D5 of the Supplementary File, available online).

V. Applications to Real-World Networks

We test the membership recovery performance of node2vec on three real-world networks, namely, the Zachary's karate graph (henceforth, ZK) [52], political blogs graph (henceforth, PB) [53], and Wikipedia graph (henceforth, WIKI) [8]. In each of the three graphs, the memberships of all vertices have been assigned based on specific real-world meanings without missing. Both ZK and PB contain 2 communities, while WIKI contains 6 communities. ZK is connected with 34 vertices. By conventions [8], [47], we ignore the directions of edges and focus on the largest connected components of PB and WIKI, which contain 1222 and 1323 vertices, respectively. We refer interested readers to the references above for more detailed information about the three real-world network datasets.

For each network dataset, we embed the vertices into the K-dimensional Euclidian space through both the SVD-based and original node2vec, and then cluster the embeddings by K-means to estimate the memberships of each vertex; K is chosen as the exact number of memberships in each graph. We test three window sizes $t_U \in \{10, 15, 20\}$. Similar to Section IV, we set $t_L = t_U - 5$ for the SVD-based node2vec and $t_L = 1$ for

TABLE IV
UPPER TABLE REPORTS THE MEMBERSHIP RECOVERY ACCURACY OF DIFFERENT EMBEDDING METHODS ON THE ZK AND PB NETWORK DATASETS

Network	SVD-based node2vec			Original node2vec			ASE	LSE	ASE+SP
	$t_U = 10$	$t_U = 15$	$t_U = 20$	$t_U = 10$	$t_U = 15$	$t_U = 20$			
ZK	0.97	0.97	0.97	0.97	0.97	0.97	1.00	0.97	0.97
PB	0.96	0.95	0.95	0.96	0.95	0.95	0.64	0.51	0.95

Network	SVD-based node2vec			Original node2vec			ASE	LSE	ASE+SP
	$t_U = 10$	$t_U = 15$	$t_U = 20$	$t_U = 10$	$t_U = 15$	$t_U = 20$			
WIKI	0.09	0.09	0.08	0.09	0.10	0.10	0.04	0.08	0.07

The lower table reports the ARI of different embedding methods on the WIKI network dataset. ASE and LSE denote spectral clusterings using the truncated eigendecomposition of the adjacency and normalized Laplacian matrix [1], [7], [8], respectively. ASE+SP denote spectral clustering using the truncated eigendecomposition of the adjacency matrix together with a spherical projection step [54], [55].

the original node2vec by default. To measure the membership recovery performances, we calculate the accuracies between the estimated memberships and the real memberships for ZK and PB; see the definition of accuracy in (4.2). For WIKI, because the criterion of accuracy becomes computationally inflexible, we alternatively use the adjusted rand index (ARI). Similar to the accuracy, ARI $= 1$ indicates the estimated memberships perfectly recover the real memberships, while ARI $= 0$ indicates the estimated memberships are assigned randomly. We also compare performances of node2vec algorithms with other popular spectral embedding algorithms, including the spectral clustering based on adjacency and normalized Laplacian [1], [7], [8], and the spectral clustering with projection onto the sphere [54]; for all methods, we use K-means for the downstream clustering.

The recovery results are summarized in Table IV. The SVD-based and original node2vec algorithms have similar performances, which are generally better than or equivalently to other methods in all three datasets. In addition, we note the PB dataset is better modeled as a DCSBM [47]. Recall that, as shown in Remark 4, node2vec can theoretically attain exact recovery for DCSBMs and hence the high-accuracy of node2vec on the PB dataset is expected. Similarly, [54] shows a valid theoretical guarantee of the spectral clustering with a spherical projection, when applied to the DCSBM graph. This can also be verified by the high accuracy of ASE+SP on PB as shown in Table IV.

VI. DISCUSSION

In this paper we derive perturbation bounds and show exact recovery for the DeepWalk and node2vec (with $p = q = 1$) algorithms under the assumption that the observed graphs are instances of the stochastic blockmodel graphs. Our results are valid under both the dense and sparse regimes for sufficient large t_L and n. The simulation results corroborate our theoretical findings; in particular, they show that increasing the sample size and window size can improve the community detection accuracy for both sparse SBM and DCSBM graphs.

We emphasize that our paper only include real data analysis on simple graphs with a small number of nodes just to illustrate the agreement between our theoretical results and the empirical performance of DeepWalk/node2vec. This is intentional as DeepWalk and node2vec are widely-used algorithms with numerous papers demonstrating their uses for analyzing real graphs in diverse applications. In contrast, our paper is one of a few that addresses the theory underpinning these algorithms and is, to the best of our knowledge, the first paper to establish consistency and exact recovery for SBMs and DCSBMs using these random-walk-based embedding algorithms. Note that exact recovery for SBMs can also be achieved using other algorithms such as those based on semidefinite programming, variational Bayes, and spectral embedding; see [41], [56], [57] for a few examples.

There are several open questions for future research:

1) In this paper we only consider the case of $p = q = 1$ for node2vec embedding (recall that $p = q = 1$ is the default parameter values for node2vec). If $p \neq 1$ and/or $q \neq 1$ then the transformed co-occurrence matrix $\tilde{\mathbf{M}}_0$ can no longer be expressed in terms of the adjacency matrix $\mathbf{A}$ or the transition matrix $\hat{\mathbf{W}}^t$; this renders the theoretical analysis for general values of p and q substantially more involved. One potential approach to this problem is to consider, similar to the notion of the *non-backtracking matrix* in community detection for sparse SBM [58], a transition matrix associated with the edges of $\mathcal{G}$ as opposed to the transition matrix associated with the vertices in $\mathcal{G}$. Indeed, if $p \neq q$ then the transition probability from a vertex v to another vertex w depends also on the vertex, say u, preceding v in the random walk. i.e., the transition probability for (v, w) depends on the choice of (u, v).
2) In this paper we focus on error bounds (in Frobenius and infinity norms) of node2vec/DeepWalk embedding for stochastic blockmodel graphs and their degree-corrected variant. An important question is whether or not stronger limit results are available for these algorithms. For example spectral embeddings of stochastic blockmodel graphs obtained via eigendecompositions of either the adjacency or the normalized Laplacian matrices are well-approximated by mixtures of multivariate Gaussians; see [29], [30] for more precise statements of these results and their implications for statistical inference in networks. It is thus natural to inquire if normal approximations also holds node2vec/Deepwalk. We ran several

one-round simple simulations to visualize the embeddings of node2vec/DeepWalk when the graphs are sampled from a SBM with

$$\mathbf{B} = \begin{pmatrix} 0.42 & 0.42 \\ 0.42 & 0.5 \end{pmatrix} \text{ and } \boldsymbol{\pi} = (0.4, 0.6). \tag{6.1}$$

The results are summarized in Fig. D6 in the Supplementary File, available online. In particular, when n is large these embeddings are also well-approximated by a mixture of multivariate Gaussians. We leave the theoretical justification of this phenomenon for future work.

3) As we allude to in the introduction, for simplicity we only consider (degree-corrected) stochastic blockmodel graphs in this paper. For the more general inhomogeneous Erdős-Rényi random graphs model, we expect that Theorems 2 and 3 still hold, provided that the edge probabilities are sufficiently homogeneous, i.e., the minimum and maximum values for the edge probabilities values are of the same order as n increases. However, the error bounds in Theorem 4 might no longer apply since the entry-wise logarithmic transformation of the co-occurrence matrices can lead to the setting wherein $\mathbf{M}_0$ is no longer low-rank, e.g., the rank of $\mathbf{M}_0$ can be as large as n the number of vertices. Furthermore, even when $\mathbf{M}_0$ have an approximate low-rank structure, due to the logarithmic transformation there is still the question of how the embedding of $\mathbf{M}_0$ relates to the underlying latent structure in $\mathbf{P}$.

4) Finally, in this paper we mainly focus on the node2vec and DeepWalk embedding through matrix factorization (SVD-based node2vec), but also compare the SVD-based node2vec with the original node2vec in the numerical experiments. As we mentioned in the introduction the original node2vec algorithm uses (stochastic) gradient descent (GD/SGD) to optimize (2.4) and obtain the embedding. As (2.4) is non-convex there can be a large number of local-minima, thereby making the theoretical analysis intractable unless we assume that the initial estimates for GD/SGD are sufficiently close to the global minima; see e.g., [43], [44] for some examples of results relating the closeness of the initial estimates and the convergence rate of GD/SGD. One popular initialization scheme for GD/SGD is via spectral methods and thus we can consider using the SVD-based embedding $\hat{\boldsymbol{\mathcal{F}}}$ as a "warm-start" for (2.4). We leave the precise convergence analysis of the resulting GD/SGD iterations to the interested reader. We note, however, that while this is certainly an interesting technical problem, the practical benefits might be limited. Indeed, the theoretical results in Section III guaranteed perfect recovery using $\hat{\boldsymbol{\mathcal{F}}}$ while the empirical evaluations in Sections IV-B and IV-C suggest that clustering based on $\hat{\boldsymbol{\mathcal{F}}}$ is comparable or even better than that of the original node2vec. In other words, as the main objective is to recover the structure in $\mathbf{M}_0$ induced by $\mathbf{P}$, it is certainly possible that optimizing (2.4) does not lead to better inference performance due to the noise in using $\mathbf{A}$ as a replacement for $\mathbf{P}$.

References

[1] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, pp. 395–416, 2007.

[2] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 203–209.

[3] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, pp. 1019–1031, 2007.

[4] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.

[5] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[6] A. Theocharidis, S. Van Dongen, A. J. Enright, and T. C. Freeman, "Network visualization and analysis of gene expression data using biolayout express 3D," *Nat. Protoc.*, vol. 4, 2009, Art. no. 1535.

[7] K. Rohe, S. Chatterjee, and B. Yu, "Spectral clustering and the high-dimensional stochastic blockmodel," *Ann. Statist.*, vol. 39, pp. 1878–1915, 2011.

[8] D. L. Sussman, M. Tang, D. E. Fishkind, and C. E. Priebe, "A consistent adjacency spectral embedding for stochastic blockmodel graphs," *J. Amer. Statist. Assoc.*, vol. 107, pp. 1119–1128, 2012.

[9] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[10] S. L. Robinson and R. J. Bennett, "A typology of deviant workplace behaviors: A multidimensional scaling study," *Acad. Manage. J.*, vol. 38, pp. 555–572, 1995.

[11] J. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2005, pp. 1569–1576.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[13] L. Cai, J. Li, J. Wang, and S. Ji, "Line graph neural networks for link prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5103–5113, Sep. 2022.

[14] S. Gui et al., "PINE: Universal deep embedding for graph nodes via partial permutation invariant set functions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 770–782, Feb. 2022.

[15] H. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[16] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Eng. Bull.*, vol. 40, no. 3, pp. 52–74, Mar. 2017.

[17] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 833–852, May 2019.

[18] F. Chen, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, "Graph representation learning: A survey," *APSIPA Trans. Signal Inf. Process.*, vol. 9, 2020, Art. no. e15.

[19] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.

[20] E. Palumbo, G. Rizzo, R. Troncy, E. Baralis, M. Osella, and E. Ferro, "Knowledge graph embeddings with node2vec for item recommendation," in *Proc. Eur. Semantic Web Conf.*, Springer, 2018, pp. 117–120.

[21] Y. Zhang, Q. Chen, Z. Yang, H. Lin, and Z. Lu, "BioWordVec, improving biomedical word embeddings with subword information and mesh," *Sci. Data*, vol. 6, pp. 1–9, 2019.

[22] X. Zhang, L. Chen, Z.-H. Guo, and H. Liang, "Identification of human membrane protein types by incorporating network embedding methods," *IEEE Access*, vol. 7, pp. 140 794–140 805, 2019.

[23] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 1234–1241.

[24] H. Chu et al., "Neural turtle graphics for modeling city road layouts," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4522–4530.

[25] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2177–2185.

[26] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and node2vec," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, 2018, pp. 459–467.

[27] C. Lin, D. L. Sussman, and P. Ishwar, "Ergodic limits, relaxations, and geometric properties of random walk node embeddings," 2021, *arXiv:2109.04526*.

[28] J. Qiu, C. Wang, B. Liao, R. Peng, and J. Tang, "Concentration bounds for co-occurrence matrices of Markov chains," 2020, *arXiv: 2008.02464*.

[29] M. Tang and C. E. Priebe, "Limit theorems for eigenvectors of the normalized Laplacian for random graphs," *Ann. Statist.*, vol. 46, pp. 2360–2415, 2018.

[30] P. Rubin-Delanchy, J. Cape, M. Tang, and C. E. Priebe, "A statistical interpretation of spectral embedding: The generalised random dot product graph," *J. Roy. Statist. Soc., Ser. B*, vol. 84, pp. 1446–1473, 2022.

[31] B. Bollobás, *Random Graphs*. Cambridge, U.K.: Cambridge Univ. Press, 2001.

[32] P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent space approaches to social network analysis," *J. Amer. Statist. Assoc.*, vol. 97, pp. 1090–1098, 2002.

[33] C. Hacker and B. Rieck, "On the suprising behavior of node2vec," ICML Workshop on Topology, Algebra, and Machine Learning, 2022, *arXiv:2206.082525*.

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

[35] C. Davis and W. Kahan, "The rotation of eigenvectors by a pertubation. III," *SIAM J. Numer. Anal.*, vol. 7, pp. 1–46, 1970.

[36] J. Cape, M. Tang, and C. E. Priebe, "The two-to-infinity norm and singular subspace geometry with applications to high-dimensional statistics," *Ann. Statist.*, vol. 47, pp. 2405–2439, 2019.

[37] L. Lei, "Unified two-to-infinity eigenspace perturbation theory for symmetric random matrices," 2019, *arXiv: 1909.04798*.

[38] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[39] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[40] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Netw.*, vol. 5, pp. 109–137, 1983.

[41] E. Abbe, "Community detection and stochastic block models: Recent developments," *J. Mach. Learn. Res.*, vol. 18, pp. 6446–6531, 2017.

[42] P. J. Bickel and A. Chen, "A nonparametric view of network models and Newman–Girvan and other modularities," in *Proc. Nat. Acad. Sci. USA*, vol. 106, pp. 21 068–21 073, 2009.

[43] Y. Chi, Y. M. Lu, and Y. Chen, "Nonconvex optimization meets low-rank matrix factorization: An overview," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5239–5269, Oct. 2019.

[44] B. Fehrman, B. Gess, and A. Jentzen, "Convergence rates for the stochastic gradient descent method for non-convex objective function," *J. Mach. Learn. Res.*, vol. 21, pp. 5354–5401, 2020.

[45] P. Sarkar and P. J. Bickel, "Hypothesis testing for automated community detection in networks," *J. Roy. Statist. Assoc. Ser. B*, vol. 78, pp. 253–273, 2016.

[46] J. Lei, "A goodness-of-fit test for stochastic block models," *Ann. Statist.*, vol. 44, pp. 401–424, 2016.

[47] B. Karrer and M. E. Newman, "Stochastic blockmodels and community structure in networks," *Phys. Rev. E*, vol. 83, no. 1, 2011, Art. no. 016107.

[48] Y. Zhao, E. Levina, and J. Zhu, "Consistency of community detection in networks under degree-corrected stochastic block models," *Ann. Statist.*, vol. 40, pp. 2266–2292, 2012.

[49] C. Gao, Z. Ma, A. Y. Zhang, and H. H. Zhou, "Community detection in degree-corrected block models," *Ann. Statist.*, vol. 46, pp. 2153–2185, 2018.

[50] L. Scrucca, M. Fop, T. B. Murphy, and A. E. Raftery, "mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models," *R J.*, vol. 8, no. 1, 2016, Art. no. 289.

[51] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *J. Amer. Statist. Assoc.*, vol. 97, no. 458, pp. 611–631, 2002.

[52] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropological Res.*, vol. 33, no. 4, pp. 452–473, 1977.

[53] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 US election: Divided they blog," in *Proc. 3rd Int. Workshop Link Discov.*, 2005, pp. 36–43.

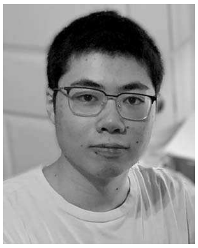
[54] A. Modell and P. Rubin-Delanchy, "Spectral clustering under degree heterogeneity: A case for the random walk Laplacian," 2021, *arXiv:2105.00987*.

[55] A. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 849–856.

[56] C. Gao, Z. Ma, A. Y. Zhang, and H. H. Zhou, "Achieving optimal misclassification proportion in stochastic blockmodels," *J. Mach. Learn. Res.*, vol. 18, pp. 1–45, 2017.

[57] V. Lyzinski, D. L. Sussman, M. Tang, A. Athreya, and C. E. Priebe, "Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding," *Electron. J. Statist.*, vol. 8, pp. 2905–2922, 2014.

[58] C. Bordenave, M. Lelarge, and L. Massoulié, "Non-backtracking spectrum of random graphs: Community detection and non-regular Ramanujan graphs," in *Proc. IEEE 56th Annu. Symp. Found. Comput. Sci.*, 2015, pp. 1347–1357.



Yichi Zhang received the BS degree in mathematics from Sichuan University, in 2018. He is currently working toward the PhD degree with the Department of Statistics, North Carolina State University. His research interests include statistical inference on random graphs, high-dimensional statistics, and causal inference.

Minh Tang received the BS degree from Assumption University, Thailand, in 2001, the MS degree from the University of Wisconsin, Milwaukee, in 2004, and the PhD degree from Indiana University, Bloomington, in 2010, all in computer science. He is currently an assistant professor with the Department of Statistics, North Carolina State University. His research interests include statistical pattern recognition, dimensionality reduction, and high-dimensional data analysis.