Electronic Journal of Statistics
Vol. 19 (2025) 745–791
ISSN: 1935-7524
https://doi.org/10.1214/25-EJS2347

Bootstrapping networks with latent space structure

Keith Levin[1] **and Elizaveta Levina**[2]

[1] *Department of Statistics, University of Wisconsin–Madison, e-mail:* `kdlevin@wisc.edu`

[2] *Department of Statistics, University of Michigan, e-mail:* `elevina@umich.edu`

Abstract: A core problem in statistical network analysis is to develop network analogues of classical techniques. The problem of bootstrapping network data presents a particular challenge, since one typically observes a single network rather than a sample. Here we propose two methods for obtaining bootstrap samples for networks drawn from latent space models. The first method generates bootstrap replicates of network statistics that can be represented as U-statistics in the latent positions, and avoids actually constructing new bootstrapped networks. Many network quantities can be represented as U-statistics, including average degree and subgraph counts, but other equally popular summaries, such as clustering coefficients, are not expressible as U-statistics. Our second bootstrapping method generates replicates of whole networks, and thus can be used for bootstrapping more general network functions. Under the assumption of a random dot product graph, a type of latent space network model, we show consistency of the proposed bootstrap methods. We give motivating examples throughout and demonstrate the effectiveness of our methods on both synthetic and real data and show that they improve upon prior methods.

MSC2020 subject classifications: Primary 62F40, 05C82; secondary 62G09.
Keywords and phrases: Bootstrap, networks, U-statistics.

Received January 2024.

1. Introduction

1.1. Motivation: network data and bootstrapping

As networks become ever more common in the sciences, there is a pressing need for network analogues of classical statistical methods. Such analogues have been developed for tasks such as two-sample testing [24, 61, 60], changepoint detection [65] and community estimation (i.e., the network analogue of clustering; see [45, 1]) to name a few. Among these fundamental classical tasks is the bootstrap [21], which allows one to make inferences about a population distribution by resampling from an observed i.i.d. sample as though it were the population itself. Unfortunately, since one typically observes only one network, naïve resampling is insufficient. While resampling methods for other dependent data such as time

---

arXiv: 1907.10821

series (see, e.g., [32]) exist, and methods for the related task of network cross-validation have been developed (see [40] and citations therein), the network bootstrap remains underexplored.

We propose two bootstrap methods for networks, one for generating bootstrap samples of network statistics with a particular structure, and another for generating bootstrap replicates of whole networks. The latter method can also be used to bootstrap network statistics, but it comes with a higher computational cost. The topic of bootstrapping network data is still relatively new, and nearly all papers on the topic [9, 25, 44] focus on the problem of bootstrapping subgraph counts of networks generated from graphons [42]. For a particular subgraph of interest on p vertices, [9] generate bootstrap replicates of its counts by sampling from the set of connected p-node subgraphs of the observed network on $n > p$ nodes, using the algorithm introduced in [66]. Following similar lines, [25] consider two related methods for generating bootstrap samples of subgraph counts from an observed network. The first resamples from what the authors term the "empirical graphon", which amounts to resampling vertices with replacement from the observed network. The second method draws samples from a stochastic block model fit to the observed network, since such a model can approximate any graphon using sufficiently many communities [50]. Chang and co-authors [16] considered the problem of estimating subgraph densities under a different setting, where there exists a true underlying network, but one observes a noisy version in which edges have been added or removed at random. The authors presented a bootstrap method for constructing confidence intervals for the subgraph densities of the true underlying network.

Recent independent work by Lunde and Sarkar [44], concurrent with this work, presents a bootstrapping procedure based on sampling induced subgraphs of the observed network. They showed that their bootstrap is valid for any network statistic obeying a central limit theorem under appropriate scaling and subgraph sampling operations, under a condition that is nontrivial to verify in general. Our results in this paper can be adapted to show that this condition is satisfied by the U-statistics we consider. However, since their bootstrap operates by sampling subgraphs of the observed network, it cannot take advantage of computational speedups available to our method (see Section 3). More recent work, which appeared during revision of this manuscript, has applied some of these computational speedups in a similar way [41]. We note also recent work by [71], which uses Edgeworth expansions to produce confidence intervals for subgraph counts.

There are many other network quantities of interest beyond subgraph counts, and little is known about bootstrapping these more general quantities. Under network latent space models [29], many such quantities of interest can be expressed as U-statistics in the latent positions. We show in Theorem 3.3 that under the fairly general latent space model known as the random dot product graph [69], these U-statistics can be bootstrapped by first estimating the latent positions of the vertices and then bootstrapping a plug-in version of the quantity of interest using known techniques for bootstrapping U- and V-statistics [3, 30, 15]. Beyond U-statistics, there are settings where we may require bootstrap samples of whole networks. For example, we may be interested in bootstrapping network statistics that do not admit a latent space U-statistic representation. Here we seek to use a single observed network to generate bootstrap network samples with (approximately) the same distribution as the observed network. We show in Theorem 4.4 that this is possible under the random dot product graph, which includes many common network models as special cases, including the stochastic block model and its variants [29, 2]. Previous work has considered generating parametric bootstrap network samples from the stochastic block model [10, 11, 34]. Shalizi and Asta [56] proved that in latent space models, generating network samples based on maximum likelihood estimates of the latent positions yields consistent bootstrap samples, but obtaining these maximum likelihood estimates is, under most latent space models, computationally infeasible. Lei [36] studied a latent space model for exchangeable random graphs. While that paper does not consider bootstrapping, it contains a result analogous to our Theorem 4.4 for a weaker notion of graph convergence. To the best of our knowledge, the present paper is the first to establish rigorous theoretical results for generating whole-network samples under the random dot product graph. An early version of our whole-network resampling approach was suggested by Athreya and co-authors [4], albeit without any theoretical guarantees. Further, while we restrict our attention here to the random dot product graph for the sake of concreteness and notational simplicity, the basic ideas of this paper are applicable to general latent space models, so long as the latent positions can be estimated at a suitable rate.

1.2. Latent space models and U-statistics

As a simple example of how latent space models interact nicely with certain network statistics, consider the triangle density, defined as

$$\hat{P}(K_3) = \binom{n}{3}^{-1} \sum_{1 \le i < j < k \le n} A_{ij} A_{jk} A_{ki}, \tag{1}$$

where $A \in \{0,1\}^{n \times n}$ is a symmetric adjacency matrix. Subgraph densities such as this play a role for node-exchangeable random graphs (i.e., those generated from graphons) that is analogous to the role of moments for Euclidean data [12, 42, 47]. As a result, obtaining confidence intervals for expected subgraph densities such as $\mathbb{E}\hat{P}(K_3)$ based on a single observed graph is of statistical import.

Under a latent space model, A is generated by first drawing latent positions $X_1, X_2, \ldots, X_n$ i.i.d. from some distribution F on a set $\mathcal{X}$ with a symmetric link function $\kappa : \mathcal{X} \times \mathcal{X} \to [0,1]$. The entries of A are drawn independently conditional on the latent positions, with $A_{ij} \sim \operatorname{Bern}(\kappa(X_i, X_j))$, and the conditional expectation of the triangle density is

$$\mathbb{E}[\hat{P}(K_3) \mid X_1, \ldots, X_n] = \binom{n}{3}^{-1} \sum_{1 \le i < j < k \le n} \kappa(X_i, X_j)\kappa(X_j, X_k)\kappa(X_k, X_i),$$

which is a U-statistic with kernel $h(x,y,z) = \kappa(x,y)\kappa(y,z)\kappa(z,x)$. In Section 3, we show that many network quantities can be written similarly, including all subgraph densities. For any such U-statistic with kernel h, if we had access to the latent positions, we could apply existing techniques for bootstrapping U-statistics [13, 3, 30] to generate bootstrap replicates of

$$U_n = U_n(h) = \binom{n}{m}^{-1} \sum_{1 \le i_1 < i_2 < \cdots < i_m \le n} h(X_{i_1}, X_{i_2}, \dots, X_{i_m}).$$

In practice, we do not observe the latent positions and must instead estimate them from the observed adjacency matrix A. Supposing for now that we had estimates $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_n$ based on A, a sensible approach would be to bootstrap the quantity

$$\hat{U}_n = \hat{U}_n(h) = \binom{n}{m}^{-1} \sum_{1 \le i_1 < i_2 < \cdots < i_m \le n} h(\hat{X}_{i_1}, \hat{X}_{i_2}, \dots, \hat{X}_{i_m}).$$

Under suitable smoothness conditions on h and provided that the true latent positions can be sufficiently accurately estimated, we may reasonably expect $\hat{U}_n$ to be a good approximation to U_n and that, furthermore, bootstrap techniques applied to $\hat{U}_n$ instead of U_n will still produce (approximately) equivalent bootstrap samples.

1.3. *Nonparametric network bootstrap samples*

The classical bootstrap [21] is based on sampling with replacement from a sample $X_1, X_2, \dots, X_n$ as though it were the population itself. There is no straightforward analogue for network data, both due to dependence among the edges and because we typically observe only one network rather than an i.i.d. sample. Fortunately, the structure of latent space models provides a way forward. The latent positions $X_1, X_2, \dots, X_n$ are drawn i.i.d. from F, and thus a bootstrap sample $X_1^*, X_2^*, \dots, X_n^*$ drawn i.i.d. from the empirical distribution $F_n = n^{-1}\sum_{i=1}^n \delta_{X_i}$ can be thought of as being approximately drawn from F. Once again, if we had access to the latent positions, it would be natural to generate a bootstrap replica A^* of the adjacency matrix by drawing from F_n as though it were the latent position distribution. That is, we draw $X_1^*, X_2^*, \dots, X_n^*$ i.i.d. from F_n and generate $A_{ij}^* \sim \text{Bern}(\kappa(X_i^*, X_j^*))$. Since we do not observe the latent positions $X_1, X_2, \dots, X_n$, a natural approach is to produce estimates $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_n$ from A and use their empirical distribution $\hat{F}_n$ in place of F_n. Then we may generate bootstrap network samples $\hat{A}^*$ by drawing from a latent space model with latent position distribution $\hat{F}_n$. Once again, provided that the estimates approximate the true latent positions well, we may expect $\hat{F}_n$ to be a good approximation to F_n, which is in turn a good approximation to F. We may then expect that the distribution of $\hat{A}^*$ approximates the distribution of A. Indeed, we will prove below that $\hat{A}^*$ converges to A in a suitably-defined Wasserstein metric. Defining a Wasserstein distance on graphs requires a notion of distance on graphs. There are a few such distances in the literature. In Section 4, we consider a distance that we call the "graph matching distance" and show that under the random dot product graph, the network resampling scheme outlined above produces bootstrap samples $\hat{A}^*$ that converge to the distribution of A in the Wasserstein distance defined with respect to the graph matching distance.

1.4. Roadmap

The rest of this paper is structured as follows. In Section 2, we briefly review the necessary background related to the random dot product graph and U-statistics. Section 3 presents our method and theoretical results for bootstrapping network U-statistics, and Section 4 covers our method for generating bootstrap samples of whole networks. Section 5 gives a brief experimental demonstration of both of these methods. We conclude in Section 6 with a brief discussion and directions for future work.

2. Background and preliminaries

2.1. Notation

For a positive integer n, $[n]$ denotes the set $\{1, 2, \ldots, n\}$ and S_n denotes the set of permutations of $[n]$. For integers $m < n$, $\mathcal{C}_m^n$ denotes all ordered m-tuples of distinct elements of $[n]$,

$$\mathcal{C}_m^n = \{(i_1, i_2, \ldots, i_m) \in [n]^m : 1 \le i_1 < i_2 < \cdots < i_m \le n\}.$$

For a vector x, $\|x\|$ denotes its Euclidean norm. For a matrix M, $\|M\|$ denotes its spectral norm, $\|M\|_F$ its Frobenius norm, and $\|M\|_1 = \sum |M_{ij}|$. Throughout, C is a positive constant, not depending on network size, whose value may change from line to line or within the same line.

2.2. The random dot product graph

The random dot product graph [69, 4] is a latent space network model in which the latent positions are points in Euclidean space, and edge probabilities are given by inner products of the latent positions.

Definition 2.1 (Random dot product graph)**.** Let F on $\mathbb{R}^d$ be a d-dimensional inner product distribution, meaning that $0 \le x^T y \le 1$ for all $x, y \in \operatorname{supp} F$. Let $X_1, X_2, \ldots, X_n$ be drawn i.i.d. from F and arranged as the rows of $X \in \mathbb{R}^{n \times d}$. Conditional on X, generate the symmetric adjacency matrix $A \in \mathbb{R}^{n \times n}$ by independently drawing $A_{i,j} \sim \mathrm{Bern}(X_i^T X_j)$ for all $1 \le i < j \le n$. We say that A is a random dot product graph with latent position distribution F, and write $(A, X) \sim \mathrm{RDPG}(F, n)$.

We assume without loss of generality that the second moment matrix of F, $\mathbb{E}X_1X_1^T \in \mathbb{R}^{d\times d}$, has full rank, since otherwise we may consider a lower-dimensional model in which it has full rank. Since any orthogonal transformation of the latent positions preserves inner products and yields the same distribution for A, we can only hope to recover the latent positions up to an orthogonal transformation. Non-identifiability of this type is unavoidable in latent space models; see [56]. Under the random dot product graph, the latent positions can be estimated (up to this non-identifiability) via spectral methods. One such method is the "adjacency spectral embedding" [58].

Definition 2.2 (Adjacency spectral embedding)**.** Let $\hat{S} \in \mathbb{R}^{d\times d}$ be the diagonal matrix formed by the top d largest-magnitude eigenvalues of the adjacency matrix A and let $\hat{U} \in \mathbb{R}^{n\times d}$ be the matrix with the corresponding eigenvectors as its columns. The adjacency spectral embedding of A is defined as $\mathrm{ASE}(A,d) = \hat{U}\hat{S}^{1/2} \in \mathbb{R}^{n\times d}$.

The rows of $\hat{X} = \mathrm{ASE}(A,d)$ are estimates of the latent positions, with a guaranteed convergence rate. This convergence suggests that the plug-in procedure sketched in Section 1 may succeed in the limit. We will formalize this intuition in Section 3.

Lemma 2.3 ([45], Lemma 5)**.** *Let* $(A,X) \sim \mathrm{RDPG}(F,n)$*. Then with probability at least* $1 - Cn^{-2}$ *there exists an orthogonal matrix* $Q \in \mathbb{R}^{d\times d}$ *such that*

$$\max_{i\in[n]} \|Q^T\hat{X}_i - X_i\| \le \frac{C\log n}{\sqrt{n}}.$$

As specified in Definition 2.1, the random dot product graph generates only networks with positive definite expected adjacency matrices. This restriction can be removed by replacing the inner product $X_i^TX_j$ with $X_i^TI_{p,q}X_j$, where $I_{p,q}$ is a diagonal matrix with p ones and q negative ones on its diagonal [54]. The graph root distribution [36] further generalizes this model to allow for infinite-dimensional latent positions. In both models, an eigenvalue truncation similar to Definition 2.2 yields estimation akin to Lemma 2.3. While both of these generalizations are useful, we restrict our attention to the random dot product graph for the sake of brevity and notational simplicity, noting that our results can be extended with minimal additional assumptions to these two more general models. In essence, all that is required is that the latent positions can be estimated at a rate like $O(n^{-1/2})$, up to logarithmic factors. We note that care must be taken in extending our results to the graph root distribution: estimation in that model relies on truncating infinite-dimensional latent positions to finitely many dimensions, which would force us to allow the U-statistic kernel to change in n. To avoid this, we must think of the estimated latent positions in the graph root distribution as being infinite-dimensional latent positions with all entries after the truncation dimension set to 0.

As defined here, the random dot product graph produces dense networks, in the sense that the number of edges grows as $\Omega(n^2)$. This is a basic property of node-exchangeable models [42], but it is at odds with many networks observed in the real world, where sparsity is the norm: the number of edges is observed to grow much slower than the quadratic number of *possible* edges. The solution typically pursued in the literature is to scale the edge probabilities by a sparsity parameter $\rho_n \in (0,1)$ obeying $\rho_n \to 0$ as $n \to \infty$. That is, we replace the latent positions $X_1, X_2, \dots, X_n$ with scaled versions $\sqrt{\rho_n} X_1, \sqrt{\rho_n} X_2, \dots, \sqrt{\rho_n} X_n$. Provided that this sparsity parameter does not decrease to zero too quickly, this poses no real complication for the bootstrapping methods proposed here, but it does complicate the notation and setup. As such, we consider bootstrapping under the dense random dot product graph in the next few sections below. We return to the matter of incorporating sparsity in Section 3.4.

2.3. Bootstrapping U-statistics

Given a measurable function $h : \mathcal{X}^m \to \mathbb{R}$ symmetric in its arguments and a sample $X_1, X_2, \dots, X_n$ drawn i.i.d. from a distribution F on $\mathcal{X}$, the U-statistic with kernel h is given by

$$U_n = U_n(h) = \frac{1}{\binom{n}{m}} \sum_{\mathbf{i} \in \mathcal{C}_m^n} h(X_{i_1}, X_{i_2}, \dots, X_{i_m}). \tag{2}$$

The study of quantities of this form dates to [28]. See Chapter 5 of [55] for a more thorough overview. Consider a quantity of interest $\theta = \theta(F) = \mathbb{E} h(X_1, X_2, \dots, X_m)$, where the expectation is taken with respect to $X_1, X_2, \dots, X_m \overset{\text{i.i.d.}}{\sim} F$. A classic result states that if $X_1, X_2, \dots, X_n \overset{\text{i.i.d.}}{\sim} F$, then provided the kernel h is non-degenerate with respect to F (i.e., $h_1(z) = \mathbb{E} h(z, X_2, X_3, \dots, X_m)$ has $\operatorname{Var} h_1(X_1) > 0$), $\sqrt{n}(U_n - \theta(F))$ converges in distribution to a mean-0 normal with variance $m^2 \zeta_1$, where

$$\zeta_1 = \mathbb{E} \left(\mathbb{E}[h(X_1, X_2, \dots, X_m) \mid X_1] - \theta \right)^2. \tag{3}$$

ζ_1 is typically unknown, but can be estimated via the bootstrap [13, 3, 30, 15]. [13] showed that for non-degenerate h, if $\mathbb{E} h(X_1, X_1, \dots, X_1) < \infty$, one can draw $X_1^*, X_2^*, \dots, X_n^* \overset{\text{i.i.d.}}{\sim} F_n$ and consider

$$\binom{n}{m}^{-1} \sum_{\mathbf{i} \in \mathcal{C}_m^n} h(X_{i_1}^*, X_{i_2}^*, \dots, X_{i_m}^*).$$

As observed by [13], resampling from F_n fails when the kernel h is degenerate, but a weighted bootstrap can be used instead [3, 30]. We do not consider degenerate kernels here, but weighted bootstrap schemes can also yield computational speedups [3]. See also [48, 20] for more recent work on bootstrapping array-like data.

Following [15], consider the quantity

$$U_n^* = \binom{n}{m}^{-1} \sum_{\mathbf{i} \in \mathcal{C}_m^n} \mathbb{W}_{\mathbf{i}} \, h(X_{i_1}, X_{i_2}, \dots, X_{i_m}), \tag{4}$$

where $\mathbb{W} \in \mathbb{R}^{\mathcal{C}_m^n}$ is a vector of random weights. Taking $\mathbb{W}_{\mathbf{i}} = \prod_{k=1}^m W_{i_k}$ with $W \sim \text{Multinomial}(n, n^{-1})$, one recovers the "Efron-weighted" bootstrap [3, 30]. Taking

$$\mathbb{W}_{\mathbf{i}} = \frac{W_{i_1} + \cdots + W_{i_m}}{m\sqrt{1 - 1/n}},$$

one obtains the additive bootstrap discussed at length in Chapter 4 of [15]. Under suitable conditions on the weight vector $\mathbb{W}$, the quantity in Equation (4) converges in distribution to the same $N(0, m^2\zeta_1)$ limit as U_n. The specific conditions on $\mathbb{W}$ vary, but provided those conditions are met, U_n^* is distributionally consistent in the sense that the asymptotic distribution of the bootstrap sample $\sqrt{n}(U_n^* - \theta)$ matches that of $\sqrt{n}(U_n - \theta)$.

3. Bootstrapping latent position U-statistics

3.1. Network U-statistics: examples

Our aim in this section is to obtain bootstrap samples of a U-statistic $U_n = U_n(h)$, which is a function of the latent positions $X_1, X_2, \dots, X_n$. The obstacle is that we observe the adjacency matrix A and not the latent positions themselves. We will show below that using the estimates $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_n$ in place of the true latent positions results in a U-statistic that converges to U_n almost surely. Further, bootstrapping the resulting plug-in U-statistic yields asymptotically equivalent bootstrap samples to those we would obtain by following the schemes in Section 2.3 if the latent positions were observed. Before presenting these convergence results, we highlight a few examples of network quantities that are expressible as latent position U-statistics.

Example 1 (Average Degree)**.** Consider the (normalized) average degree,

$$\bar{d}(A) = \frac{1}{n}\sum_{i=1}^n \frac{d_i}{n-1} = \frac{1}{n}\sum_{i=1}^n \frac{\sum_{j \neq i} A_{i,j}}{n-1} .$$

Under the random dot product graph, its conditional expectation is

$$\mathbb{E}[\bar{d}(A) \mid X] = \mathbb{E}\frac{1}{n(n-1)}\sum_{i=1}^n \sum_{j \neq i} \mathbb{E}[A_{i,j} \mid X_i, X_j] = 2\binom{n}{2}^{-1} \sum_{i<j} X_i^T X_j,$$

which is a U-statistic with kernel $h(x, y) = 2x^T y$.

Example 2 (Subgraph Counts)**.** Let R and G be graphs on m and n vertices, respectively, with $m \leq n$. Numbering the vertices of G arbitrarily, for $\mathbf{i} = (i_1, i_2, \dots, i_m) \in \mathcal{C}_m^n$, let $G[\mathbf{i}]$ denote the m-vertex subgraph of G induced by $i_1, i_2, \dots, i_m$ and consider the quantity

$$\hat{P}(R) = \frac{1}{\binom{n}{m}} \sum_{\mathbf{i} \in \mathcal{C}_m^n} \mathbb{I}\{G[\mathbf{i}] \simeq R\}, \tag{5}$$

where we write $H \simeq R$ to denote that graphs H and R are isomorphic. $\hat{P}(R)$ thus measures the (empirical) proportion of times that R appears as a subgraph out of the total number of possible subgraphs on m vertices. Letting $B \in \mathbb{R}^{m \times m}$ be the adjacency matrix of graph R, we can write

$$\mathbb{E}[\hat{P}(R) \mid X] = \frac{1}{\binom{n}{m}} \sum_{\mathbf{i} \in \mathcal{C}_m^n} \sum_{\tau \in S_m} \frac{\prod_{1 \le k < \ell \le m} (X_{i_{\tau(k)}}^T X_{i_{\tau(\ell)}})^{B_{k\ell}} (1 - X_{i_{\tau(k)}}^T X_{i_{\tau(\ell)}})^{1-B_{k\ell}}}{N(R)}$$

where $N(R)$ denotes the number of graphs isomorphic to R. From this, we see that $\mathbb{E}[\hat{P}(R) \mid X]$ is a U-statistic with kernel

$$h_R(x_1, x_2, \dots, x_m) = \frac{1}{N(R)} \sum_{\tau \in S_m} \prod_{1 \le k < \ell \le m} (x_{i_{\tau(k)}}^T x_{i_{\tau(\ell)}})^{B_{k\ell}} (1 - x_{i_{\tau(k)}}^T x_{i_{\tau(\ell)}})^{1-B_{k\ell}}.$$

Example 3 (Maximum Mean Discrepancy)**.** The maximum mean discrepancy [26] is a test statistic for nonparametric two-sample hypothesis testing. Given $\lambda \in [0, 1]$ and two distributions F_1 and F_2, let $X_1, X_2, \dots, X_n$ be drawn i.i.d. from the mixture $\lambda F_1 + (1 - \lambda) F_2$, and let $Y_i = 1$ if $X_i \sim F_1$ and $Y_i = 0$ otherwise, for $i \in [n]$. Letting $I_1 = \{i : Y_i = 1\}$ and $n_1 = |I_1|$ and defining I_2 and n_2 analogously, the maximum mean discrepancy is given by

$$M_n = \sum_{i,j \in I_1 \text{ distinct}} \frac{\kappa(X_i, X_j)}{n_1(n_1 - 1)} + \sum_{i,j \in I_2 \text{ distinct}} \frac{\kappa(X_i, X_j)}{n_2(n_2 - 1)} - \sum_{i \in I_1} \sum_{j \in I_2} \frac{\kappa(X_i, X_j)}{n_1 n_2},$$

where κ is the kernel of a reproducing kernel Hilbert space. By definition, M_n is a U-statistic in $X_1, X_2, \dots, X_n$ with kernel $h((x_i, y_i), (x_j, y_j)) = (-1)^{y_i - y_j} \kappa(x_i, x_j)$. Suppose that F_1 and F_2 are such that $F = \lambda F_1 + (1 - \lambda) F_2$ is a d-dimensional inner product distribution, with $(A, X) \sim \mathrm{RDPG}(F, n)$, and suppose that we observe the indicators $Y_1, Y_2, \dots, Y_n$. A natural approach to testing the hypothesis that $F_1 = F_2$ is to form a test statistic from the (estimated) latent positions,

$$\hat{M}_n = \sum_{i,j \in I_1 \text{ distinct}} \frac{\kappa(\hat{X}_i, \hat{X}_j)}{n_1(n_1 - 1)} + \sum_{i,j \in I_2 \text{ distinct}} \frac{\kappa(\hat{X}_i, \hat{X}_j)}{n_2(n_2 - 1)} - \sum_{i \in I_1} \sum_{j \in I_2} \frac{\kappa(\hat{X}_i, \hat{X}_j)}{n_1 n_2},$$

where $\hat{X} = \mathrm{ASE}(A, d)$. A slight variant of this statistic was considered by Tang and co-authors for the purpose of two-network hypothesis testing [61].

Example 4 (Degree Moments)**.** The degree distribution of a graph carries important information about graph structure. Measures such as the variance of the degrees,

$$V_d(A) = n^{-2} \sum_{i,j} \left(\frac{d_i - d_j}{n} \right)^2,$$

where $d_i = \sum_k A_{i,k}$ is the degree of the i-th vertex, provide a useful summary of vertex behavior. Rearranging the sum, we have

$$\mathbb{E}[V_d(A) \mid X] = n^{-4} \sum_{i,j,k,\ell} \mathbb{E}\left[A_{i,k}(A_{i,\ell} - A_{j,\ell}) \mid X\right] = n^{-4} \sum_{i,j,k,\ell} X_i^T X_k (X_i - X_j)^T X_\ell,$$

which is a V-statistic [55] in the latent positions after appropriate symmetrization. Similar results can be shown for other central moments of the degree distribution.

A number of other network quantities are expressible similarly, either under a different latent geometry or after appropriate rescaling by some network-dependent quantity that converges almost surely to a parameter depending only on the latent position distribution. Examples include measures of assortative mixing by degrees [49], energy statistics [59, 33] and the Randić connectivity index [52].

Remark 1 (Specifying the Parameter of Interest)**.** Among the examples above, we have seen statistics of the form $t : \{0,1\}^{n\times n} \to \mathbb{R}$ (e.g., the average degree), where $\mathbb{E}[t(A) \mid X]$ is expressible as a U-statistic in the latent positions. We stress, however, that the target of inference here and in what follows is $\mathbb{E}[t(A)]$, rather than the conditional expectation $\mathbb{E}[t(A) \mid X]$ or, say, $t(\mathbb{E}A)$. Indeed, it is precisely because $\mathbb{E}[t(A) \mid X]$ is expressible as a U-statistic in the latent positions that we are able to construct a confidence interval for $\mathbb{E}h(X_1, X_2, \dots, X_m) = \mathbb{E}t(A)$.

3.2. Consistency of network U-statistic bootstrap

Having seen how U-statistics arise in network analysis, we return to our setting where the latent positions $X_1, X_2, \dots, X_n$ must be estimated from the matrix A. Letting $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_n \in \mathbb{R}^d$ be the rows of $\hat{X} = \operatorname{ASE}(A, d)$, we consider the plug-in U-statistic,

$$\hat{U}_n = \binom{n}{m}^{-1} \sum_{1\le i_1<i_2<\cdots<i_m\le n} h(\hat{X}_{i_1}, \hat{X}_{i_2}, \dots, \hat{X}_{i_m}). \tag{6}$$

If this quantity is to resemble U_n, h must be invariant to the non-identifiability inherent to the random dot product graph, and thus we make the following assumption.

Assumption 1. Let $\mathbb{O}_d$ denote the set of all d-by-d orthogonal matrices. The kernel h satisfies

$$h(X_1, X_2, \dots, X_n) = h(QX_1, QX_2, \dots, QX_n) \quad \text{for all} \quad Q \in \mathbb{O}_d.$$

The main results of this section state that for suitably smooth kernel functions, the plug-in estimate in Equation (6) and bootstrap samples formed from it are asymptotically equivalent to the U-statistic formed from the true latent positions $X_1, X_2, \dots, X_n \overset{\text{i.i.d.}}{\sim} F$. The following assumption makes this notion of smoothness precise.

Assumption 2. Let $\nabla^2 h : \mathbb{R}^{md} \to \mathbb{R}^{md\times md}$ be the Hessian of kernel h. $\nabla^2 h$ is continuous on the closure of $\operatorname{supp} F$ and there exists a neighborhood $\mathcal{S} \subseteq (\mathbb{R}^d)^m$ of $\operatorname{supp} F$ with

$$\sup\{\|\nabla^2 h(Z_1, Z_2, \dots, Z_m)\| : (Z_1, Z_2, \dots, Z_m) \in \mathcal{S}\} < \infty.$$

We note that with the exception of the maximum mean discrepancy, the rotation-invariance and homogeneity properties in Assumption 1 are trivially obeyed by all our example statistics in Section 3.1, by virtue of being based on inner products of latent positions. Whether or not the maximum mean discrepancy obeys these properties boils down to properties of the function κ chosen by the practitioner. We note that most commonly-chosen kernels that are monotone functions of distances (e.g., Gaussian radial basis functions), will obey these conditions. The bounded Hessian property in Assumption 2 is slightly more complicated: it depends on the interplay of the U-statistic kernel h and the latent position distribution F. We note, however, that if the U-statistic kernel h obeys Assumption 1, then so long as F is well-behaved (i.e., its support is bounded away from singularities of $\nabla^2 h$), all our example statistics in Section 3.1 obey Assumption 2.

The following theorem shows that Assumptions 1 and 2 are sufficient to ensure that the plug-in U-statistic $\hat{U}_n$ recovers U_n asymptotically. The proof is given in the Appendix.

Theorem 3.1. *Let F be a d-dimensional inner product distribution and suppose that $h : (\mathbb{R}^d)^m \to \mathbb{R}$ is a symmetric kernel satisfying Assumptions 1 and 2. Suppose $(A, X) \sim \mathrm{RDPG}(F, n)$ and let $\hat{X} = \mathrm{ASE}(A, d)$. Let U_n and $\hat{U}_n$ be the U-statistics based on, respectively, the true latent positions X and their estimates $\hat{X}$. Then $\sqrt{n}(\hat{U}_n - U_n) \to 0$ almost surely.*

From the fact that U_n converges almost surely to the population parameter $\theta = \theta(F) = \mathbb{E}U_n$ ([55] Theorem 5.4 A), $\hat{U}_n$ is a strongly consistent estimate of θ. Further, by Slutsky's Theorem, $\hat{U}_n$ has the same distributional limit as U_n, provided that the underlying U-statistic kernel is non-degenerate.

Corollary 3.2. *Under the setting of Theorem 3.1, provided that h is non-degenerate with respect to F, $\hat{U}_n$ satisfies $\hat{U}_n \to \mathbb{E}U_n = \theta(F)$ almost surely and $\sqrt{n}(\hat{U}_n - \theta) \to \mathcal{N}(0, m^2\zeta_1)$ in law, where ζ_1 is defined in Equation (3).*

We note that convergence to a normal random variable in this corollary (and several of our results to be discussed below) requires that the kernel h be non-degenerate with respect to F. That is, the function $h_1 : \mathbb{R}^d \to \mathbb{R}$ given by $h_1(z) = \mathbb{E}h(z, X_2, \ldots, X_m)$ is such that $\operatorname{Var} h_1(X_1) > 0$. This is a standard assumption when dealing with U-statistics (see, e.g., [55]). As an example of how degeneracy arises in the RDPG, consider the special case of the Erdős-Rényi random graph with edge probability p. That is, an RDPG in which the latent position distribution is a point mass at $\sqrt{p} \in [0, 1]$. For our statistic, consider the edge density (i.e., the subgraph density for the complete graph on two nodes)

$$\hat{\rho}_n = \binom{n}{2}^{-1} \sum_{i<j} A_{ij}.$$

Conditional on the latent positions,

$$\mathbb{E}[\hat{\rho} \mid X] = \binom{n}{2}^{-1} \sum_{i<j} X_i X_j,$$

so the kernel function is $h(x_1, x_2) = x_1 x_2$, and $h_1(x) = \mathbb{E} x X_2 = x\sqrt{p}$. Since $X_1 = X_2 = \sqrt{p}$ almost surely under this model, we have

$$\operatorname{Var} h_1(X_1) = p \operatorname{Var} X_1 = 0.$$

This degeneracy disappears once we allow F to be a distribution more complex than a point mass. More generally, we stress that degeneracy is a property that arises as an interplay between the kernel h and the latent position distribution F. With the exception of very particular, highly structured choices of h and F, degeneracy is very much the exception, not the norm.

Theorem 3.1 and its corollary establish the appropriate convergence of the (estimated) latent position U-statistic $\hat{U}_n$, but our main goal is the more delicate task of obtaining bootstrap samples to approximate the sampling distribution of U_n. If we knew the true latent positions, any number of techniques for bootstrapping U-statistics would work. The idea is thus to construct, instead of a bootstrap sample U_n^* as in (4), a plug-in version

$$\hat{U}_n^* = \binom{n}{m}^{-1} \sum_{\mathbf{i} \in \mathcal{C}_m^n} \mathbb{W}_{\mathbf{i}} \; h(\hat{X}_{i_1}, \hat{X}_{i_2}, \ldots, \hat{X}_{i_m}),$$

where again $\mathbb{W} \in \mathbb{R}^{\mathcal{C}_m^n}$ is a vector of random weights independent of the observed network. We will assume that these are adjacency spectral embedding estimates, but we stress that similar results can be obtained under any estimation scheme that recovers the latent positions at a suitably fast rate (see, e.g., [67, 68]). As mentioned in Section 2.3, the specific conditions on $\mathbb{W}$ needed to ensure the distributional consistency of U_n^* vary, but for our plug-in scheme to work, we require the following additional growth condition.

Assumption 3. The weight vector $\mathbb{W}$ satisfies

$$\max_{\mathbf{i} \in \mathcal{C}_m^n} |\mathbb{W}_{\mathbf{i}}| = o\left(\frac{\sqrt{n}}{\log^2 n} \right).$$

With this assumption, we have the following theorem for the plug-in U-statistic bootstrap. The proof is given in the Appendix.

Theorem 3.3. *Let F be a d-dimensional inner product distribution and suppose that $h : (\mathbb{R}^d)^m \to \mathbb{R}$ is a symmetric kernel non-degenerate with respect to F and satisfying Assumptions 1 and 2. Let $(A, X) \sim \mathrm{RDPG}(F, n)$ and $\hat{X} = \mathrm{ASE}(A, d)$. Let U_n^* be the weighted bootstrap U-statistic defined in Equation (4) and let $\hat{U}_n^*$ be its plug-in version. Then,*

1. *If U_n^* is distributionally consistent and the weight vector $\mathbb{W} \in \mathbb{R}^{\mathcal{C}_m^n}$ satisfies Assumption 3, then*
$$\sqrt{n}(\hat{U}_n^* - \hat{U}_n) \to \mathcal{N}(0, m^2\zeta_1) \qquad \text{in law,}$$
where ζ_1 is as defined in Equation (3),
2. *For the unweighted adjacency spectral embedding bootstrap*
$$\hat{V}_n^* = \binom{n}{m}^{-1} \sum_{\mathbf{i} \in \mathcal{C}_m^n} h(\hat{X}_{i_1}^*, \hat{X}_{i_2}^*, \dots, \hat{X}_{i_m}^*),$$
we have
$$\sqrt{n}(\hat{V}_n^* - \hat{U}_n) \to \mathcal{N}(0, m^2\zeta_1) \qquad \text{in law.}$$

Remark 2 (The degenerate case)**.** The reader familiar with U-statistics may wonder what can be said when the kernel h is degenerate with respect to F (see, e.g., [55], Chapter 5). It is known that if h is r-degenerate, then $n^{r/2}(U_n - \theta)$ converges to a nondegenerate limiting distribution, which is not, in general, normal [55, 3]. A result analogous to Theorem 3.3 for this case, unfortunately, does not appear feasible, since the concentration of U_n about the true parameter θ is of a smaller order than the concentration of the estimates $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_n$ about the true latent positions [45]. That is, the estimation error in Lemma 2.3 does not vanish fast enough to yield convergence of $\hat{U}_n^*$ to U_n^* in probability.

3.3. Computational concerns

Both U-statistics and the bootstrap are well known to be computationally intensive. As a result, a naïve implementation of our positional U-statistic resampling scheme would be of little practical utility for large n. To alleviate this computational burden, we use an additive weighted bootstrapping procedure, discussed at length in Chapter 4 of [15]. Consider a U-statistic with kernel h taking m arguments. Having generated a vector of weights $W \in \mathbb{R}^n$, we form the weight vector $\mathbb{W} \in \mathbb{R}^{\mathcal{C}_m^n}$ by setting $\mathbb{W}_{\mathbf{i}} = \sum_{k=1}^m W_{i_k}/m$ for each $\mathbf{i} = (i_1, i_2, \dots, i_m) \in \mathcal{C}_m^n$. While a number of choices for the distribution of W are possible (see [7] for a general discussion; for U-statistics specifically, see [30, 15]), we take $W \sim \text{Multinomial}(n, n^{-1})$ in our experiments in Section 5 for simplicity. A concentration inequality applied entry-wise to W followed by a union bound is enough to ensure that $\max_{\mathbf{i} \in \mathcal{C}_m^n} |\mathbb{W}_{\mathbf{i}}| \le C \log^m n$, so that Assumption 3 is satisfied. The additive structure of $\mathbb{W}$ enables a speedup in computing bootstrap replicates of U_n. We construct, for each $i \in [n]$, the quantity
$$\tilde{U}_{ni} = \binom{n-1}{m-1}^{-1} \sum_{\mathbf{i} \in \mathcal{C}_m^n \,:\, i \in \mathbf{i}} h(X_{i_1}, X_{i_2}, \dots, X_{i_m}). \tag{7}$$

Recalling our definition of U_n^* from Equation (4), it is simple to verify that
$$U_n^* = \frac{1}{n} \sum_{i=1}^n W_i \tilde{U}_{ni}.$$

As discussed in [15], this enables generation of many bootstrap samples after only a single instance of $O(n^m)$ computation time to construct the $\tilde{U}_{ni}$, rather than $O(Bn^m)$ computation time to generate B bootstrap samples under a more naïve implementation.

Unfortunately, the $O(n^m)$ time required to compute the quantity in Equation (7) for all $i \in [n]$ may still be quite substantial, particularly if m is larger than 2. We can further reduce the computational cost by making use of incomplete U-statistics [14, 19], replacing the average in (7) with a Monte Carlo estimate, drawing for each $i \in [n]$ a uniform random sample of size M with replacement from the set $\{\mathbf{i} \in \mathcal{C}_m^n : i \in \mathbf{i}\}$. With this modification, our method for obtaining B bootstrap samples for a U-statistic in the latent positions requires a single low-rank spectral decomposition followed by $O((M+B)n)$ sampling operations. Thus our bootstrap is far less computationally demanding than existing algorithms for generating bootstrap samples of subgraph counts, which require extensive sampling and counting operations.

3.4. Sparse networks

Our results above hold under the assumption that F does not depend on n, with the result that the expected node degrees grow linearly in n, which is unrealistic in many applications. A natural fix is to rescale the expectation of A by a sparsity factor ρ_n, so that $\mathbb{E}[A \mid X] = \rho_n XX^T$, and let ρ_n decrease in n. For identifiability, we assume that if X_1, X_2 are i.i.d. draws from F, then $\mathbb{E}X_1^T X_2 = 1$, and for all suitably large n, $x, y \in \operatorname{supp} F$ implies $0 \le \rho_n x^T y \le 1$. Analogous assumptions are common in the literature [12, 9, 25]. This is equivalent to rescaling the latent positions by $\sqrt{\rho_n}$. Under this scaling, $\hat{U}_n$ now estimates $\mathbb{E}h(\sqrt{\rho_n}X_1, \dots, \sqrt{\rho_n}X_m)$ rather than $\mathbb{E}h(X_1, \dots, X_m)$, and thus we must specify how the kernel h behaves with respect to scaling of its arguments. A consistency result analogous to Theorem 3.1 can be established under this setting with an additional homogeneity assumption on h. A proof can be found in the Appendix.

Theorem 3.4. *In the sparse setting just described, let h be a non-degenerate kernel satisfying Assumptions 1 and 2, with $\operatorname{supp} F$ replaced with the convex hull of $\{0\} \cup \operatorname{supp} F$. Suppose in addition that there exists $r \ge 1$ such that for all $\alpha \ge 0$ and all $x_1, x_2, \dots, x_m \in \operatorname{supp} F$,*

$$h(\alpha x_1, \dots, \alpha x_m) = \alpha^r h(x_1, \dots, x_m). \tag{8}$$

Define the estimator $\hat{\rho}_n = \binom{n}{2}^{-1} \sum_{i<j} A_{ij}$. Then

$$\frac{\sqrt{n}(\hat{U}_n - U_n)}{\hat{\rho}_n^r} \to 0 \qquad \textit{in probability.}$$

The condition in Equation (8) is satisfied by the average degree, subgraph count and degree moment examples presented at the beginning of this section.

Whether or not the maximum mean discrepancy obeys this condition depends on the kernel κ. Focusing on subgraph counts, for a graph R on m vertices with edge set $E = E(R)$, we have the U-statistic kernel

$$h_R(x_1, x_2, \ldots, x_m) = \frac{1}{N(R)} \sum_{\tau \in S_m} \prod_{\{i,j\} \in E} x_{\tau(i)}^T x_{\tau(j)} \prod_{\{i,j\} \notin E} (1 - x_{\tau(i)}^T x_{\tau(j)}),$$

where $N(R)$ is the number of graphs isomorphic to R. Thus,

$$h(\sqrt{\rho_n} x_1, \sqrt{\rho_n} x_2, \ldots, \sqrt{\rho_n} x_m) = \rho_n^{|E|} (1 - \rho_n)^{\binom{m}{2} - |E|} h(x_1, x_2, \ldots, x_m).$$

Rescaling by $\rho_n^{-|E|}$ yields the normalized subgraph density considered in [12],

$$\tilde{P}(R) = \rho_n^{-|E|} \mathbb{E} h_R(\sqrt{\rho_n} X_1, \sqrt{\rho_n} X_2, \ldots, \sqrt{\rho_n} X_m)$$

Bootstrapping this quantity is the focus of both [9] and [25]. Note that estimating $\tilde{P}(R)$ requires estimating both ρ_n and the subgraph density

$$\mathbb{E} h_R(\sqrt{\rho_n} X_1, \sqrt{\rho_n} X_2, \ldots, \sqrt{\rho_n} X_m).$$

Letting $\hat{\rho}_n = \binom{n}{2}^{-1} \sum_{i<j} A_{ij}$ again, we have the following distributional result under suitable structural assumptions on R (see [12, 9] for previous use and discussion of these assumptions) and suitable lower-bound on the sparsity ρ_n (required to ensure that suitably accurate estimation of the latent positions is possible). A proof of this result can be found in the Appendix.

Theorem 3.5. *Under the sparse setting described above, let R be a graph on m vertices, either acyclic or equal to a cycle on m vertices. Provided that $n\rho_n = \omega(\log n)$,*

$$\sqrt{n}\, \frac{\hat{U}_n^* - \hat{U}_n}{\hat{\rho}^r} \to \mathcal{N}(0, \sigma^2) \text{ in law,}$$

where σ^2 depends on R and F. An analogous result holds for the unweighted bootstrap $\hat{V}_n^*$.

Remark 3. Work by Zhang and Xia [71], focused on Edgeworth expansions for subgraph counts, observed a phenomenon in which sparsity acts as a "smoother": sparsity allows one to avoid concerns about, e.g., non-lattice conditions on F that typically arise in Edgeworth expansions. The reader may wonder if a similar phenomenon applies here. In short, no obvious analogue of such a "smoothing" effect exists for our setting. The key structural concern in our setting relates to non-degeneracy, which arises from an interplay between the latent position distribution F and the U-statistic kernel h, and it is not clear that sparsity provides any way to avoid these issues.

4. Generating whole-network bootstrap samples

In this section, we turn to the more general case of bootstrapping network quantities that cannot be expressed as U-statistics in the latent positions. We start with two examples.

Example 5 (Global Clustering Coefficient)**.** For a graph G with adjacency matrix A, the global clustering coefficient is given by

$$C(G) = \frac{3\sum_{1\le i<j<k\le n} A_{ij}A_{jk}A_{ki}}{\sum_{1\le i<j<k\le n}(A_{ij}A_{jk} + A_{jk}A_{ki} + A_{ki}A_{ij})}.$$

Letting L_3 denote the chain on three vertices and K_3 denote the triangle on 3 vertices, we can write $C(G)$ as a ratio of subgraph counts, $C(G) = F_{K_3}(G)/F_{L_3}(G)$. A quantity similar to this appeared in [9], where the authors constructed a confidence interval via the delta method and properties of subgraph counts.

Example 6 (Average Shortest Path Length)**.** For a graph G on n vertices with adjacency matrix A, the shortest path distance between vertices i and j, which we denote $d_A(i,j)$, is the length of the shortest path connecting vertices i and j in G. We take $d_A(i,j) = \infty$ if i and j are in different connected components. The average shortest path distance,

$$\bar{d}_A = \binom{n}{2}^{-1} \sum_{i<j} d_A(i,j) \tag{9}$$

provides a natural measure of the extent to which a graph exhibits small-world behavior. While Equation (9) looks superficially like a U-statistic, this is not the case, since d_A is a function that itself depends on the data, rather than being a fixed kernel.

For quantities such as these that cannot be directly represented as U-statistics, a natural approach to generating bootstrap samples would be to generate random adjacency matrices $\hat{A}^*$ having similar distribution to A and compute the statistic on those replicates. This approach requires, at a minimum, that the bootstrapped network $\hat{A}^*$ be similar in distribution to A, which in turn requires a notion of distance on networks. A well-known example of such a distance is the cut metric [42]. It is especially well-suited to node-exchangeable random graphs because it metrizes convergence of subgraph densities ([42], Theorem 11.3). A resulting drawback of the cut metric is that it captures only the local information of small subgraphs, and ignores larger network structures. A first step toward a more global network distance is given by the graph matching distance, which measures the fraction of edges that differ between two graphs after their vertices have been aligned to minimize the number of such edge discrepancies.

Definition 4.1 (Graph matching distance)**.** Let G_1, G_2 be two graphs each on n vertices, with adjacency matrices $A_1, A_2 \in \mathbb{R}^{n\times n}$. The graph matching distance is defined as

$$d_{\mathrm{GM}}(A_1, A_2) = \min_{P\in\Pi_n} \binom{n}{2}^{-1} \frac{\|A_1 - PA_2P^T\|_1}{2}, \tag{10}$$

where Π_n denotes the set of all n-by-n permutation matrices.

Two sequences of networks converge in this distance if they are, asymptotically, isomorphic to one another up to a vanishing fraction of their edges.

We note that this distance has appeared in the literature under a number of different names (see [8] and citations therein). The graph matching distance is, up to a constant depending on choice of normalization, an upper bound on the cut metric, which is also based on a computationally hard optimization problem. This upper bound is immediate from the fact that the cut norm of a matrix M is upper bounded by $\|M\|_1/n^2$ (see [42] Chapter 8).

With this network distance in hand, we define a Wasserstein distance between graphs analogously to the well-known Wasserstein distance between Euclidean random variables.

Definition 4.2. Let $p \geq 1$ and let A_1 and A_2 be the adjacency matrices of two random graphs both on n vertices, and let $\Gamma(A_1, A_2)$ denote the set of all couplings of A_1 and A_2. The Wasserstein p-distance between A_1 and A_2 is given by

$$W_p^p(A_1, A_2) = \inf_{\nu \in \Gamma(A_1, A_2)} \int d_{\mathrm{GM}}^p(A_1, A_2) d\nu.$$

The following lemma shows that, essentially, the Wasserstein distance between two random dot product graphs is bounded by the Wasserstein distance between their respective latent position distributions (up to an orthogonal transformation). Since the latter Wasserstein distance must account for the orthogonal rotation, we define the Wasserstein p-distance between two d-dimensional inner product distributions F_1, F_2, for all $p \geq 1$, as

$$\mathring{d}_p(F_1, F_2) = \min_{Q \in \mathbb{O}_d} d_p(F_1, F_2 \circ Q). \tag{11}$$

The lemma below will be the main technical tool required to show that the bootstrapped $\hat{A}^*$ described above converges to A in the graph matching Wasserstein distance. A similar result was shown by [36], in a different context and for the weaker cut metric instead of the graph matching distance. A proof of the lemma can be found in the Appendix.

Lemma 4.3. *Let F_1, F_2 be d-dimensional inner product distributions with $A_1 \sim \mathrm{RDPG}(F_1, n)$ and $A_2 \sim \mathrm{RDPG}(F_2, n)$. Then*

$$W_p^p(A_1, A_2) \leq 2\mathring{d}_1(F_1, F_2)$$

Recall the procedure for generating network bootstrap replicates outlined in Section 1.3. Given A, we obtain latent position estimates $\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n$ via the adjacency spectral embedding. Letting $\hat{F}_n$ be the empirical distribution of these estimates, we draw $(\hat{A}^*, \hat{X}^*) \sim \mathrm{RDPG}(\hat{F}_n, n)$. The convergence of the estimates to the true latent positions ensures that $\hat{F}_n$ approximates F_n (up to orthogonal transformation). Since the empirical distribution approximates the population distribution, F_n is close to F. Lemma 4.3 thus suggests that $\hat{A}^*$ will be distributionally similar to A. The following theorem makes this precise. Proof details are given in the Appendix.

Theorem 4.4. *Let* $(A, X) \sim \mathrm{RDPG}(F, n)$ *where* F *is a* d*-dimensional inner product distribution. Letting* $\hat{F}_n$ *denote the empirical distribution of the estimates* $\hat{X} = \mathrm{ASE}(A, d)$*, generate* $\hat{A}^* \sim \mathrm{RDPG}(\hat{F}_n, n)$*. If* $(H, Z) \sim \mathrm{RDPG}(F, n)$ *is an independent copy of* (A, X)*,*

$$W_p^p(\hat{A}^*, H) = O\left((n^{-1/2} + n^{-1/d}) \log n\right) .$$

Since the graph matching distance upper bounds the cut metric, which metrizes convergence of subgraph densities, Theorem 4.4 implies that the subgraph densities of $\hat{A}^*$ converge almost surely to the same limit as those of H. We would also like these counts to have the same distributional limits after appropriate rescaling. Unfortunately, this distributional limit is more delicate. While Theorem 4.4 ensures that the bootstrap replicate $\hat{A}^*$ converges in the Wasserstein metric to the correct target distribution, this is not in itself sufficient to ensure that some network statistic of interest, say $t(\hat{A}^*)$, converges to the same distribution as $t(H)$. Provided $\mathbb{E}t(H)$ is finite, it is sufficient that the network statistic in question be continuous with respect to our network Wasserstein metric, in the sense that $W_1(\hat{A}^*, H) \to 0$ implies that (by slight abuse of notation), $d_1(t(\hat{A}^*), t(H)) \to 0$. This continuity is immediate for subgraph counts, since the graph matching distance upper bounds the cut metric and convergence in the cut metric implies convergence of subgraph densities [42]. However, proving this continuity even for the $\sqrt{n}$-scaled subgraph densities considered in Section 3 using the techniques of Theorem 4.4 does not appear feasible. The coupling argument used in the proof fails under the $\sqrt{n}$-scaling needed to ensure a non-degenerate limit for subgraph densities (see, e.g., [12] Theorem 1), because of the parametric rate (up to logarithmic factors) in estimating the latent positions. On the other hand, our experiments in Section 5 suggest that the whole-network bootstrap does indeed succeed for subgraph densities, and thus we suspect that a more careful analysis will show that ($\sqrt{n}$-scaled) subgraph counts are continuous with respect to W_1. An expansion of each m-tuple of vertices about its expectation, similar to the analysis in [71], would be a natural starting point for such an alalysis. Looking beyond subgraph densities, our experiments suggest that the whole-network bootstrap produces asymptotically valid bootstrap samples for the Fiedler value (see Section 5.3). We anticipate that convergence of these statistics or more complicated linear spectral statistics (see, e.g., [5, 31]) should hold. Roughly speaking, one would simply need to bound the spectral or Frobenius norm in terms of the graph matching distance, but this approach depends on first establishing the appropriate asymptotic scaling for the Fiedler value under the RDPG, which remains an open problem. All told, our theoretical results above may be used to motivate the whole-network bootstrap as a tool for building confidence intervals for network statistics more complicated than, for example, subgraph counts. Nonetheless, there remains a theoretical gap in rigorously establishing the asymptotic behavior of these network statistics under this bootstrapping scheme that warrants future study.

5. Experiments

5.1. Overview of bootstrap methods under comparison

We now turn to a brief demonstration of the performance of the methods introduced in Sections 3 and 4 on both simulated and real data. Before we present these results, we address a few points related to implementation and model selection. Firstly, we note that our theoretical results in Sections 3 and 4 assume that the latent space dimension d is known. Of course, in practice one must estimate the dimension, and this model selection problem has received much attention [72, 23, 27, 18]. In the experiments that follow, we use the method of [18] to select the embedding dimension. We note that a range of other methods and heuristics exist for choosing the embedding dimension (see, e.g., [72, 27, 40]), and that choosing the embedding dimension remains an area of active research in network analysis. "Folklore" results state that it is better to choose the embedding dimension too high rather than too low. That is, it is better to overestimate d then to underestimate it. A brief exploration of the effects of model misspecification can be found in the Appendix. These experiments lend support to the "folklore" on dimension selection.

Our results are written for the adjacency spectral embedding, but this embedding may yield estimates that have inner products outside $[0, 1]$, Additional issues arise from the diagonal entries of A, which are negligible for the purpose of our asymptotic results in Sections 3 and 4, but tend to introduce a bias to the adjacency spectral embedding for finite-size networks. Further, one may worry that the adjacency spectral embedding is not asymptotically efficient [68], leading to overly wide confidence intervals (though this is not the case in the stochastic blockmodel; see [62]). To contend with these concerns, we consider latent position estimates based on the one-step estimator of [68], which corrects the adjacency spectral embedding to account for the Bernoulli edge model. In practice, we find that this one-step estimator and the adjacency spectral embedding perform quite similarly, with the one-step estimator performing slightly better than the adjacency spectral embedding in sparser networks, at the cost of occasional numerical instability, and we report results throughout using latent position estimates based on this one-step estimator.

In addition to the two methods presented in this paper, we consider confidence intervals produced by the empirical graphon introduced by [25]. Under this method, one produces a bootstrap replicate adjacency matrix A^* by drawing $Z_1, Z_2, \dots, Z_n$ i.i.d. from the uniform distribution on $[n]$, and take $A^*_{ij} = A_{Z_i,Z_j}$. A drawback of this scheme is that since the diagonal elements of A are equal to 0, resampled vertices k, ℓ with $Z_k = Z_\ell$ are precluded from forming an edge. For the purposes of estimating subgraph counts, this can be adjusted for, as demonstrated by the results in [25], but we will see in Section 5.3 that this causes inaccuracy when generating whole networks. We expect that correcting for this deficiency is possible, but it is not trivial and we do not pursue the question here. Where applicable, we also apply a whole-network bootstrap method based on fitting a stochastic blockmodel to the observed network, then generating a network according to the estimated blockmodel parameters. We use spectral clustering of the one-step estimator embeddings followed by K-means, with K chosen according to the method of [18]. Unfortunately, when applied to models other than the stochastic blockmodel, this procedure performed quite poorly, producing coverage rates well below 80%, and thus we do not include it in plots here. Experiments applying our methods to the stochastic blockmodel can be found in the Appendix, where our methods are competitive with this stochastic blockmodel resampler.

The computational cost of the subgraph count bootstrap methods of [9] precluded a thorough comparison. Small-scale experiments suggest that both are broadly competitive with our methods, albeit at a much higher computational cost. The methods presented in [44, 57, 71], all of which appeared after submission of this work, are also applicable to the problem of estimating subgraph densities. In small-scale experiments, we found that all three of these methods performed comparably to our methods for both edge density and triangle density. Given that [71, 57] are not, strictly speaking, bootstrap methods, and that [44] require computation of a scaling factor, we leave a more thorough comparison of these various methods to future work.

5.2. Simulated data: subgraph densities

We begin with an application of our network bootstrap methods to the problem of estimating subgraph densities. In particular, we consider the problem of obtaining a confidence interval for the triangle subgraph density $P(K_3)$ in a random dot product graph. In the Appendix, we also consider the edge density, which illustrates particularly clearly the issue with the empirical graphon. We consider networks generated from a random dot product graph whose latent position distribution is a rescaled mixture of Dirichlet distributions. To generate a latent position in $\mathbb{R}^d$, we draw from a d-component mixture, in which the k-th component is given by a Dirichlet distribution with parameter vector in which the k-th entry is 1.5 and all other entries are 1. We then scale the resulting latent position by a $\text{Beta}(d-1,1)$ random variable.

For each of our methods under comparison, we estimate the variance based on 30 bootstrap samples and use this estimate to construct a 90% confidence interval for the triangle density. We consider networks with latent dimension $d = 2, 5, 10$ and number of vertices ranging from 100 to 2000. Figure 1 shows the coverage rates obtained. Each data point is the empirical coverage rate of 500 Monte Carlo replicates. All three methods perform largely comparably, with performance degrading in the presence of higher-dimensional latent positions, though we see that in the case of latent dimension 5 or 10, performance improves as the network size increases. In the $d = 10$ case, while all three methods are overly conservative, the empirical graphon improves more slowly as a function of n compared to our methods.

Figure 2 shows the average runtime in seconds of these three methods, as a function of n for the $d = 10$ case of the triangle density experiment just described. The left-hand plot shows the full runtime of each of the three methods,

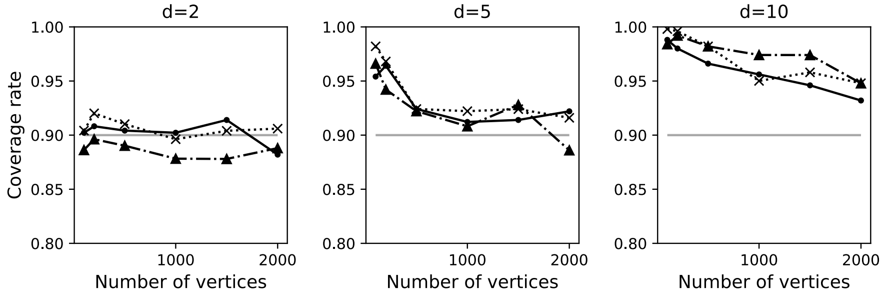

Fig 1. *Coverage rate for the triangle density as a function of the number of vertices for the whole-network (crosses, dotted line), U-statistic (circles, solid line) and empirical graphon (triangles, dotted dash).*

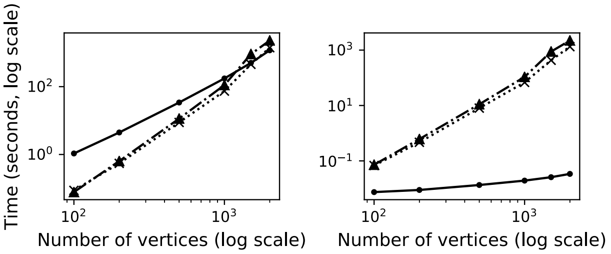

Fig 2. *Runtime in seconds as a function of the number of vertices for the whole-network (crosses, dotted line), U-statistic (circles, solid line) and empirical graphon (triangles, dotted dash). The left-hand plot shows the full time, including preprocessing. The right-hand plot shows only the time required to produce bootstrap samples, ignoring preprocessing time.*

incorporating both preprocessing time and resampling time. This preprocessing time includes, for example, the time to compute leading eigenvectors and to compute $\tilde{U}_{ni}$ as in Equation (7). The right-hand plot shows the runtime required only to generate bootstrap replicates, once all preprocessing has been finished. Examining the left-hand plot, we see that our whole-network method and the empirical graphon require more or less identical runtimes, and both scale cubically with the number of vertices This is unsurprising; both methods require generating whole networks and then computing the triangle density on the resulting network. The empirical graphon is consistently slightly slower than the whole-network method at larger network sizes, likely due to the sporadic memory access pattern required when constructing each sample. The U-statistic bootstrap is consistently slower than the other two methods, but we note that its runtime appears to scale quadratically with the number of vertices, rather than cubically. This is due to our use of incomplete U-statistics to estimate $\tilde{U}_{ni}$ rather than computing it exactly, where we have chosen to estimate each of these $O(n)$ quantities based on n Monte Carlo samples each. Turning our attention to the right-hand plot in Figure 2, we see that once this preprocessing step is completed, the U-statistic network resampling procedure is orders of magnitude faster than the whole-network and empirical graphon methods. This is an important and useful property of this method, since the slower runtime seen in the left-hand plot can be further ameliorated by parallelizing the estimation of $\tilde{U}_{ni}$ for $i \in [n]$.

5.3. Simulated data: more complicated network functions

We now illustrate the broader utility of our whole-network bootstrap method discussed in Section 4. Recall that our U-statistic method, like most other existing network bootstrap methods, is designed first and foremost for bootstrapping subgraph counts. Of course, not all network statistics of interest can be represented as subgraph counts (or, more generally, as U-statistics in the latent positions). Here, we consider two such network statistics that cannot be represented as subgraph counts, and thus serve to illustrate the utility of our whole-network bootstrap method for constructing confidence intervals for these quantities. We first apply this method to the problem of estimating the expectation of the the Fiedler value.

The Fiedler value [22] of a network is given by the second smallest eigenvalue of the combinatorial Laplacian, $L = D - A$, where D is the degree matrix, the diagonal matrix having $D_{ii} = \sum_j A_{ij}$. We generate networks of size $n = 100, 200, 500, 1000$ from a random dot product graph with latent position distribution given by a Dirichlet with parameter vector $(0.02, 0.01, \ldots, 0.01) \in \mathbb{R}^d$ for latent space dimension $d = 2, 5, 10$. We aim to produce a 90% confidence interval for the expectation of the Fiedler value under this distribution. To the best of our knowledge, there is no closed-form expression for our inferential target. As such, we estimate the expected Fiedler value based on $10,000$ Monte Carlo replicates for each experimental condition. We compare our whole-network bootstrap against the empirical graphon method discussed in Section 5.2. All methods use 30 bootstrap replicates. The results of this experiment are described in Figure 3. Each data point captures the empirical coverage rate of 500 independent Monte Carlo trials. In the $d = 2$ setting, the empirical graphon provides slightly more accurate confidence intervals, but in the higher-dimensional settings of $d = 5, 10$, our whole-network bootstrap achieves the nominal coverage rate on much smaller networks than the empirical graphon, with the two methods becoming comparable for larger networks.

To further illustrate our whole-network bootstrap method, we consider another network statistic that is not expressible as a latent U-statistic: the expected average shortest path length,

$$\mathbb{E}[\bar{d}_A \mid \bar{d}_A < \infty], \tag{12}$$

where we condition on the event that the graph is connected to avoid the situation where $\mathbb{E}[\bar{d}_A]$ is infinite. For comparison, we consider the empirical graphon as well as a parametric bootstrap procedure, which performs estimation over a much smaller space of models compared to the RDPG-based resampling scheme and the empirical graphon, and can thus serve as a gold standard when its underlying model is true. Once again, since we are not aware of any closed-form

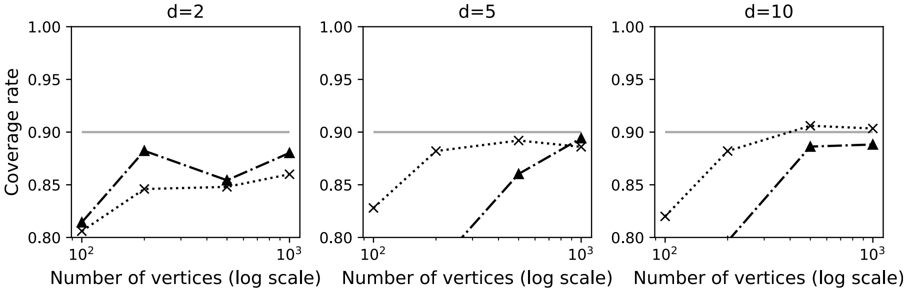

FIG 3. *Coverage rate for the Fiedler value as a function of the number of vertices for the whole-network (crosses, dotted line) and empirical graphon (triangles, dotted dash) bootstrap methods.*

expression for our inferential target, we estimate the quantity in Equation (12) based on $10,000$ Monte Carlo replicates for each of our experimental conditions.

We generate A from a random dot product graph with latent position distribution given by $\text{Beta}(2,3)$ for $n = 50, 100, 200, 500, 1000, 2000, 3000$. We discard and regenerate A in the event that the resulting graph is not connected, since we are interested in the average shortest path in A conditional on that average being finite. Given A, we estimate the latent positions again using the one-step estimator [68]. Letting $\hat{F}_n$ denote the empirical distribution of these estimates, we then draw bootstrap replicates of $(\hat{A}^*, \hat{X}^*) \sim \text{RDPG}(\hat{F}_n, n)$, computing the average shortest path of each iterate, resampling in the event that a sample $\hat{A}^*$ is not connected. For the empirical graphon, we resample from the same original observed network A. For the parametric bootstrap, we fit the parameters of a beta distribution to latent position estimates produced by the one-step estimator. Letting $(\hat{a}, \hat{b})$ denote the estimated parameters of the Beta distribution, we draw the $n \times n$ adjacency matrix $\hat{A}^*$ from $\text{RDPG}(\text{Beta}(\hat{a}, \hat{b}), n)$.

For each method, we generate 30 bootstrap samples, discarding and regenerating samples as needed to ensure connected networks. We construct a 90% confidence interval centered on the observed $\bar{d}_A$ via a normal approximation. The results are summarized in Figure 4. Each data point is the empirical coverage rate of 500 independent Monte Carlo trials. Our whole-network bootstrap and its parametric counterpart are both near the nominal coverage rate at all network sizes, while the empirical graphon is overly conservative for $n < 1000$.

5.4. Example: social network comparison

We now apply our novel bootstrap methods to a collection of social networks derived from Facebook [64, 53] at eight U.S. colleges and universities. We selected eight institutions pre hoc to be representative of different overlapping styles of school: selective non-Ivy (Johns Hopkins, Massachusetts Institute of Technology), engineering-focused (Massachusetts Institute of Technology, Rice), women's colleges (Simmons, Smith, Wellesley), small liberal arts (Oberlin, Trinity, Smith, Wellesley). The networks range in size from about 1500 to 6500

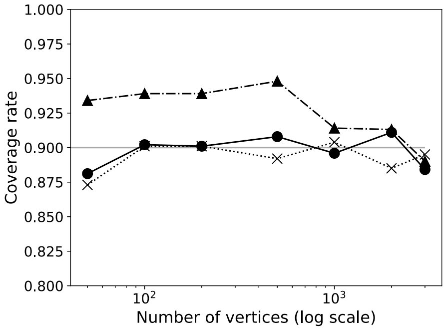

Fig 4. *Coverage rate for the average shortest path length as a function of the number of vertices for the whole-network (crosses, dotted line), parametric bootstrap (circles, solid line) and empirical graphon (triangles, dotted dash).*

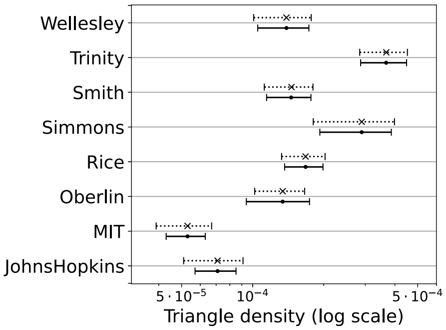

Fig 5. 90% *confidence intervals for the triangle density of social networks for eight different schools, using both our whole-network (crosses, dotted line) and U-statistic (circles, solid line) bootstrap methods.*

vertices. For each network, we used both our U-statistic and whole-network bootstrap methods to produce 90% confidence intervals for the triangle density. The results are summarized in Figure 5. Our two methods are largely in agreement, with no obvious trend of one producing narrower or wider intervals than the other. While there is no ground truth against which to verify our results, a few interesting patterns suggest that our confidence intervals are capturing true underlying similarities and differences in the social dynamics of these schools. For example, Massachusetts Institute of Technology and Johns Hopkins, both urban, selective non-Ivy schools, have very similar triangle densities, with confidence intervals that suggest they are distinct from the other six institutions. Smith and Wellesley, both small women-only liberal arts colleges, have very similar triangle densities. The similarity of Oberlin to these two institutions is unsurprising, given that it is a similarly-sized liberal arts college. One surprising result is that Rice University, an engineering-focused school similar in size to Massachusetts Institute of Technology, is more in line with the small liberal arts colleges.

6. Discussion and conclusion

We have presented two methods for bootstrapping network data under latent space models. For network quantities expressible as U-statistics in the latent positions, our results in Section 3 show that plugging in latent position estimates and applying existing bootstrap methods for U-statistics yields a distributionally consistent resampling procedure. Experimental evidence in Section 5.2 supports this claim. By design, our resampling scheme is able to take advantage of computational speedups for bootstrapping U-statistics, and thus provides a substantial computational improvement over existing approaches to bootstrapping subgraph counts, which require expensive combinatorial enumeration. We have also proposed a method to resample whole networks by first estimating the latent positions and then drawing bootstrap samples from the empirical distribution of these estimates and using these draws to generate a network. We have shown that under the random dot product graph model, networks produced in this way converge in a suitably defined Wasserstein distance to the network from which they are built.

Directions for future work are many. As alluded to in the paper, the core ideas presented here apply more broadly than the random dot product graph. Our results in this paper can be extended trivially to the generalized random dot product graph [54] and graph root distributions [36], but the basic ideas should work for any latent space model, as long as the latent positions can be accurately estimated. We leave an exploration of the precise analogues of our results for other latent space models to future work, along with investigating the extent to which the smoothness conditions required by the results of Section 3 might be relaxed.

As discussed at the end of Section 4, convergence under the Wasserstein network distance does not necessarily imply convergence of other network statistics such as $\sqrt{n}$-scaled subgraph densities or the average shortest path length. It is possible that a stronger notion of distance is needed to ensure such convergences, one that eschews the local perspective of the cut metric and graph matching distance in favor of more global measures of graph similarity. A distance between networks that considers path lengths or k-hop neighborhoods of individual vertices might better capture the global properties necessary to ensure this convergence. A number of works [17, 8, 63] present possible starting points for this line of work to better understand how notions of network distance interact with network statistics.

Appendix A: Additional experimental results

A.1. Additional simulated data: edge density

As in the first simulated data experiments in the main text, we consider a random dot product graph with d-dimensional latent space, but this time seek to estimate the edge density, $\mathbb{E}\binom{n}{2}^{-1}\sum A_{ij}$. We again take as our latent position distribution a rescaled mixture of Dirichlet distributions, with the k-th of the d mixture components given by a Dirichlet distribution whose parameter vector has k-th entry 1.5 and all other entries are 1. We then scale the latent positions by independent $\mathrm{Beta}(d-1,1)$ random variables.

Our goal is to produce a 90% confidence interval for this expectation. We consider networks with latent dimension $d = 2, 5, 10$ and number of vertices ranging from 100 to 1000. Once again, we compare our U-statistic bootstrap and our whole-network bootstrap against the empirical graphon. Figure 6 describes the outcome of this experiment. Each data point is the empirical coverage rate of 500 Monte Carlo replicates. Here we see more clearly the issue with the empirical graphon: the sampled network contains fewer edges than it should, resulting in erratic performance visible in the $d = 2$ and $d = 5$ settings. Our full-network method exhibits similar behavior, to a lesser extent. We observe that our U-statistic-based bootstrap encounters no such problem, matching or improving upon both other methods.

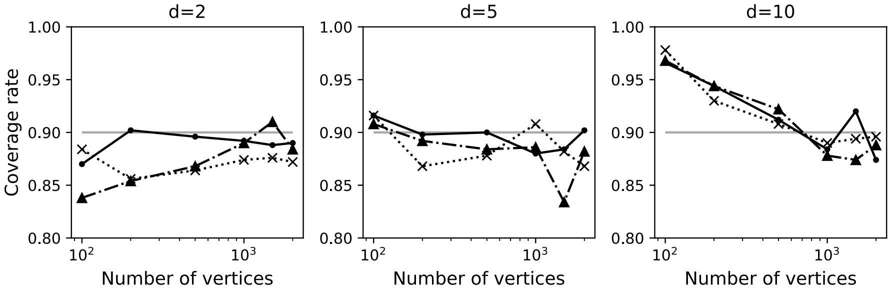

Fig 6. *Coverage rate for the edge density as a function of number of vertices for the whole-network (crosses, dotted line), U-statistic (circles, solid line) and empirical graphon (triangles, dotted dash).*

A.2. Effect of model selection and misspecification

As discussed in the main text, we have used the method of [18] to select the embedding dimension for our bootstrap methods in our experiments. Nonetheless, one may wonder how incorrectly choosing this dimension may influence subsequent inference. In addition, while we mentioned in the main text that we use the one-step estimator of [68] throughout, one may wonder how this method compares with the adjacency spectral embedding. To explore both of these points, we generated networks as in the simulated data triangle density experiments in the main text, with $n = 1000$ vertices and latent space dimension 10. We then applied our U-statistic and whole-network bootstrap methods, using estimates obtained from both the adjacency spectral embedding and the one-step estimator, varying the embedding dimension between 2 and 20. Figure 7 summarizes the results of this experiment. Each data point is the empirical coverage rate of 500 independent Monte Carlo trials. The plot shows performance of both

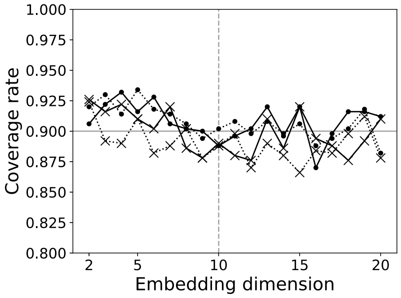

FIG 7. *Coverage rate as a function of the embedding dimension for the U-statistic (solid line) and whole-network (dotted line) bootstrap methods based on the adjacency spectral embedding (crosses) and one-step estimator (circles). The true latent space dimension* $d = 10$ *is indicated by a vertical dashed line.*

our U-statistic and whole-network bootstrap methods, using both the adjacency spectral embedding and the one-step estimator. We see that performance is quite similar across the board for both methods and using both estimators. Choosing the embedding dimension too low seems to introduce slight over-estimatation of the variance, but the coverage rate of both of our methods is otherwise quite stable with respect to choice of dimension. Of course, increasing this embedding dimension too far does eventually lead to a degradation in performance, but we found that performance was stable with dimension chosen incorrectly as large as about 50.

Beyond dimension selection, another possible source of model misspecification arises from the fact, mentioned in the introduction to the main text, that the random dot product graph generalizes the widely used stochastic blockmodel, and is a special case of the graphon. As such, it is natural to ask how the random dot product graph and bootstrap methods based thereon performs when applied to these related models.

First, we consider how our methods perform when applied to a network generated from a stochastic blockmodel. Since the stochastic blockmodel is a special case of the random dot product graph, we expect our methods to perform well. Still, we expect them to be out-performed by a method specialized to the stochastic blockmodel. We consider a method that fits a stochastic blockmodel to the observed network and generates new networks according to the fitted parameters. We estimate the parameters according to spectral clustering, whereby we first estimate latent positions using the one-step estimator of [68] to estimate embeddings, then assign vertices to communities according to K-means clustering. We note that this is broadly similar to the histogram network bootstrap introduced in [25].

We continue with the task of estimating the triangle density, this time generating networks from a stochastic blockmodel, in which each vertex is equally likely to belong to any community. Pairs of vertices in the same community form an edge with probability 0.4, and vertices in different communities form

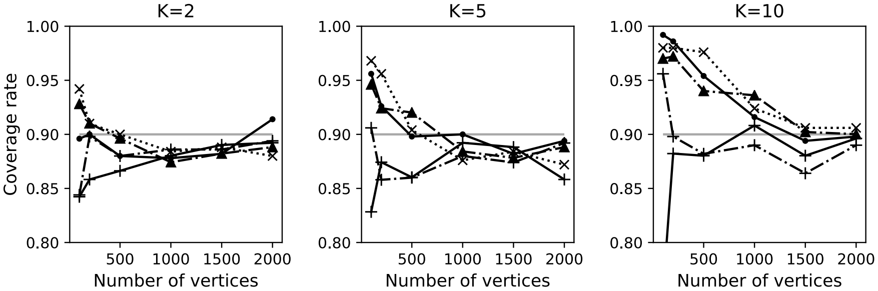

Fig 8. *Coverage rate of a number of bootstrap methods as a function of the number of vertices for stochastic blockmodels with $K = 2$ (left), $K = 5$ (center) and $K = 10$ (right) communities. The lines indicate coverage rates for the U-statistic (circles, solid line), whole-network (crosses, dotted line), empirical graphon (triangles, dotted dash), stochastic blockmodel (plus signs, solid line) and oracle (plus signs, dotted dash) bootstrap methods.*

an edge with probability 0.1. We consider networks with number of vertices ranging from 100 to 2000 and number of communities $K = 2, 5, 10$. In addition to our two novel network bootstrap methods and the empirical graphon, we also apply the stochastic blockmodel resampling scheme discussed in Section 5.1 of the main text and an "oracle" parametric method. This oracle method is given access to the correct community memberships of all of the vertices, estimates the stochastic blockmodel parameters according to those known memberships, and generates resampled networks using those parameter estimates. Figure 8 summarizes the results of this experiment. Each data point is the empirical coverage rate of 500 independent Monte Carlo trials. We see that all methods, including the stochastic blockmodel bootstrap and the oracle method, struggle to achieve the nominal coverate on small networks. Somewhat surprisingly, our methods and the empirical graphon appear to outperform the two stochastic blockmodel methods in the $K = 5$ case, but that these two correctly-specified methods markedly outperform the other methods in the higher-dimensional case $K = 10$.

Just as the stochastic blockmodel is a special case of the random dot product graph, the random dot product graph is a special case of the graphon. Therefore, we now consider the case of a network generated from a graphon. Recall that a graphon is encoded by a symmetric function $g : [0, 1] \times [0, 1] \to [0, 1]$. To generate a network on n vertices, we generate independent uniform random variables $U_1, U_2, \ldots, U_n$, then generate edges (conditionally) independently with probability $\Pr[A_{ij} = 1] = g(U_i, U_j)$. We consider a graphon of the form $g(u, v) = \rho \min\{u, v\}$, where $\rho \in [0, 1]$ is a sparsity parameter. This graphon has infinite spectrum, so while it can be approximated by a random dot product graph, its behavior cannot be recovered exactly by any finite-dimensional random dot product graph. We consider networks generated from this graphon for sparsity values $\rho = 0.1, 0.3, 0.5$ and number of vertices $n = 100, 200, 500, 1000, 1500, 2000$. The results of this experiment are summarized in Figure 9. As in our other ex-

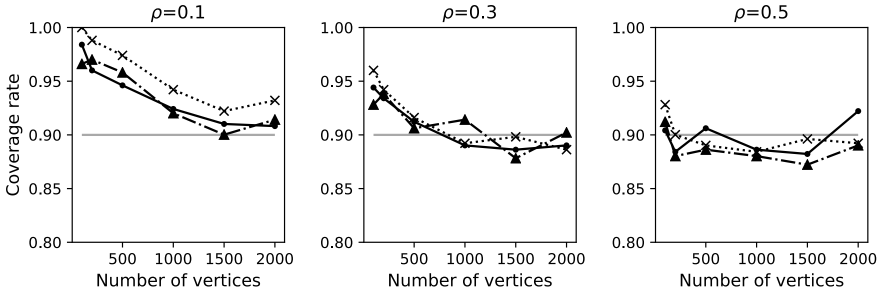

FIG 9. *Coverage rate for the triangle density as a function of the number of vertices for the whole-network (crosses, dotted line), U-statistic (circles, solid line) and empirical graphon (triangles, dotted dash) when the network is generated from a graphon with sparsity parameter* 0.1 *(left),* 0.3 *(center) and* 0.5 *(right).*

periments involving models outside the stochastic blockmodel, the blockmodel-based resampler performed exceptionally poorly in this setting, but the other three bootstrapping methods under study perform broadly similarly. Reassuringly, we see that our two methods are competitive with the empirical graphon. We suspect that this is because while the graphon from which we are sampling is infinite dimensional, its structure can be approximated suitably well (for the purposes of resampling, at least) by a low-rank truncation. See [36] for further discussion of this point.

Appendix B: Technical results

In the sections below, we provide supplemental proofs and technical details supporting the results in the main text. We note that in handling the competing goals of notational precision and conformity with the existing literature, we have opted for the latter, and as a result, a few symbols are overloaded. In particular, in the appendices that follow, P will be used to denote either the subgraph density introduced in Section 3 or the n-by-n expectation of the adjacency matrix A conditional on the latent positions. Which of these two uses is intended will be clear from the context. Similarly, the symbol U is overloaded, denoting a U-statistic in some contexts and denoting an n-by-d matrix with orthonormal columns in others. Again, which of these two is intended will be clear from the context and from the fact that we subscript by n (i.e., $U_n, \hat{U}_n$, etc.) in the case of U-statistics, and leave plain $U, \hat{U}$ to denote the matrices.

We begin by collecting a handful of technical results from the existing literature on random dot product graphs that will be useful in the proofs below. We note that many of the logarithmic terms in the bounds below may be removed via more careful analyses (e.g., those in [37, 6]). We use simpler bounds below for the sake of brevity and simplicity.

Lemma B.1. *Let* $(A, X) \sim \mathrm{RDPG}(F, n)$ *for some* d*-dimensional inner product distribution* F*. Define* $P = XX^T \in \mathbb{R}^{n \times n}$ *so that* $\mathbb{E}[A \mid X] = P$ *and let* USU^T

be the rank-d eigendecomposition of P, so that $S \in \mathbb{R}^{d\times d}$ is a diagonal matrix with entries given by the eigenvalues $\lambda_1(P) \ge \lambda_2(P) \ge \cdots \ge \lambda_d(P) > 0$ and $U \in \mathbb{R}^{n\times d}$ has as its columns the d corresponding unit eigenvectors. Similarly, let $\hat{U}\hat{S}\hat{U}^T = \hat{X}\hat{X}^T \in \mathbb{R}^{n\times n}$ be the rank-d approximation of A given by its top d largest-magnitude eigenvalues and eigenvectors. That is, let $\hat{S} \in \mathbb{R}^{d\times d}$ be the diagonal matrix with entries given by the d largest-magnitude eigenvalues of A and let $\hat{U} \in \mathbb{R}^{n\times d}$ have as its columns the d corresponding unit eigenvectors. There exist constants $C_2 \ge C_1 > 0$ such that with probability at least $1 - Cn^{-2}$,

$$C_1 n \le \lambda_d(P) \le \cdots \le \lambda_1(P) \le C_2 n \tag{13}$$

$$\text{and } \|A - P\| \le C\sqrt{n \log n}. \tag{14}$$

Further, letting $Q \in \mathbb{R}^{d\times d}$ be the orthogonal matrix guaranteed by Lemma 2.3, for all suitably large n it holds with probability at least $1 - Cn^{-2}$ that

$$\|Q - \hat{U}^T U\|_F \le \frac{C \log n}{n}, \tag{15}$$

$$\|Q\hat{S}^{-1/2} - S^{-1/2}Q\|_F \le \frac{C \log^{1/2} n}{n^{3/2}}, \tag{16}$$

$$\|\hat{U}Q - U\|_F \le \frac{C \log^{1/2} n}{\sqrt{n}}. \tag{17}$$

Proof. Equations (13) and (14) are Observations 1 and 2, respectively, in [38]. Equation (15) and (16) follow from, respectively, Proposition 16 and Lemma 17 in [46], with the slight alteration that we use the spectral norm bound of [51] rather than that of [43]. A proof of Equation (17) appears in the course of the proof of Lemma 5 in [38]. We restate it here for the sake of completeness.

By Theorem 2 in [70], there exists orthonormal $Q \in \mathbb{R}^{d\times d}$ such that

$$\|\hat{U}Q - U\|_F \le \frac{C\sqrt{d}\|A - P\|}{\lambda_d(P)}.$$

Applying Equation (14) yields (17). □

Lemma B.2. *With notation as above, letting $Q \in \mathbb{R}^{d\times d}$ denote the orthogonal matrix guaranteed by Lemma 2.3, with probability at least $1 - Cn^{-2}$,*

$$\|A\hat{U}(\hat{S}^{-1/2}Q - QS^{-1/2})\|_F \le Cn^{-1/2}\log^{1/2} n.$$

Proof. Let $E = A - \hat{U}\hat{S}\hat{U}^T$ be the residual after making the best rank-d approximation to A. By definition, the eigenvectors of E are orthogonal to the columns of $\hat{U}$, whence $E\hat{U} = 0$, and thus $A\hat{U} = \hat{U}\hat{S}$.

$$\begin{aligned}\|A\hat{U}(\hat{S}^{-1/2}Q - QS^{-1/2})\|_F &= \|\hat{U}\hat{S}\hat{U}^T\hat{U}(\hat{S}^{-1/2}Q - QS^{-1/2})\|_F \\ &\le \|\hat{S}\|\|\hat{S}^{-1/2}Q - QS^{-1/2}\|_F.\end{aligned}$$

Lemma B.1 bounds the spectral norm as $\|\hat{S}\| = O(n)$ and the Frobenius norm as

$$\|\hat{S}^{-1/2}Q - QS^{-1/2}\|_F = O(n^{-3/2}\log^{1/2} n),$$

which completes the proof. □

The following lemma is a generalization of Lemma 10 of [45] to the case where the second moment matrix $\mathbb{E}_F X_1 X_1^T \in \mathbb{R}^{d\times d}$ may have repeated eigenvalues.

Lemma B.3. *With notation and setup as above, Let* $Q \in \mathbb{R}^{d\times d}$ *be the orthogonal matrix guaranteed by Lemma 2.3. With probability at least* $1 - Cn^{-2}$,

$$\|A(\hat{U}Q - U)S^{-1/2}\|_F \le Cn^{-1/2}\log n.$$

Proof. Let $E = A - \hat{U}\hat{S}\hat{U}^T$ as in the previous proof. Taking $Q \in \mathbb{R}^{d\times d}$ to be as in Lemma 2.3, by the triangle inequality and basic properties of the Frobenius norm,

$$\begin{aligned}\|A(\hat{U}Q - U)S^{-1/2}\|_F &= \|(\hat{U}\hat{S}\hat{U}^T + E)(\hat{U}Q - U)S^{-1/2}\|_F \\ &\le \|\hat{U}\hat{S}\|\|Q - \hat{U}^T U\|_F\|S^{-1/2}\| + \|E\|\|\hat{U}Q - U\|_F\|S^{-1/2}\|.\end{aligned}$$

Applying Lemma B.1, with probability $1 - Cn^{-2}$, Equations (13) and (14), both hold, so that

$$\|S^{-1/2}\| = O(n^{-1/2}) \text{ and } \|\hat{S}\| = O(n),$$

and Equation (15) implies

$$\|\hat{U}\|\|\hat{S}\|\|Q - \hat{U}^T U\|_F\|S^{-1/2}\| \le Cn^{-1/2}\log n.$$

Similarly, since $\|E\| \le \|A - P\| = O(n^{1/2}\log n)$ by Equation (14), Equation (17) implies that

$$\|E\|\|\hat{U}Q - U\|_F\|S^{-1/2}\| \le Cn^{-1/2}\log^{1/2} n.$$

Thus, combining the above two displays,

$$\|A(\hat{U}Q - U)S^{-1/2}\|_F \le Cn^{-1/2}\log n,$$

as we set out to show. □

Appendix C: Proof of Theorems 3.1 and 3.3

Here we provide detailed proofs of the results in Section 3. Both rely on a second-order Taylor expansion of the U-statistic evaluated at the true latent positions. A similar argument appears in [61]. The main technical challenge here comes from the more complicated dependency structure of U-statistics which in turn requires a more involved indexing and counting argument. The following two lemmas will prove useful in bounding the linear and quadratic terms, respectively, in the Taylor expansion. Throughout this appendix we use $\mathcal{C}_m^n$ to denote the set of all m-tuples of strictly increasing integers from $[n]$. That is, $\mathcal{C}_m^n = \{(i_1, i_2, \dots, i_m) : i_1, i_2, \dots, i_m \in [n], 1 \le i_1 < i_2 < \cdots < i_m \le n\}$. We write $\mathbb{R}^{\mathcal{C}_m^n}$ to denote the set of vectors over the reals with entries indexed by the $\binom{n}{m}$ elements of $\mathcal{C}_m^n$, so that if $v \in \mathbb{R}^{\mathcal{C}_m^n}$, then $v_{\mathbf{i}} \in \mathbb{R}$ for each $\mathbf{i} \in \mathcal{C}_m^n$. For $\mathbf{i} \in \mathcal{C}_m^n$ and $X \in \mathbb{R}^{n\times d}$, we use $X_{\mathbf{i}}$ to denote the m-by-d matrix formed by stacking the rows of X whose indices appear in $\mathbf{i} = (i_1, i_2, \dots, i_m)$.

Lemma C.1. *Let F be a d-dimensional inner product distribution with $(A, X) \sim \mathrm{RDPG}(F, n)$ and let $\hat{X} = \mathrm{ASE}(A, d)$. Let $h : (\mathbb{R}^d)^m \to \mathbb{R}$ be a kernel function, symmetric in its arguments, satisfying Assumptions 1 and 2. Then there exists orthogonal matrix $Q \in \mathbb{R}^{d\times d}$ such that for any fixed $v \in \mathbb{R}^{\mathcal{C}_m^n}$, with probability at least $1 - Cn^{-2}$,*

$$\left| \sum_{\mathbf{i}\in\mathcal{C}_m^n} v_{\mathbf{i}} (\hat{X}Q - X)_{\mathbf{i}}^T (\nabla h)(X_{\mathbf{i}}) \right| \le C \max_{\mathbf{i}\in\mathcal{C}_m^n} |v_{\mathbf{i}}| \binom{n-1}{m-1} \log n.$$

Proof. Define the map $\mathcal{T}_m : \mathbb{R}^{n\times d} \to \mathbb{R}^{\mathcal{C}_m^n \times md}$, which transforms the matrix $Y \in \mathbb{R}^{n\times d}$ with rows $Y_i \in \mathbb{R}^d$ for $i = 1, 2, \dots, n$ into the matrix $\tilde{Y} = \mathcal{T}_m(Y) \in \mathbb{R}^{\mathcal{C}_m^n \times md}$ as follows. Indexing the $\binom{n}{m}$ rows of $\tilde{Y} = \mathcal{T}_m(Y)$ by the $\binom{n}{m}$ elements of $\mathcal{C}_m^n$, define the row indexed by $\mathbf{i} = (i_1, i_2, \dots, i_m) \in \mathcal{C}_m^n$ as

$$\tilde{Y}_{\mathbf{i}} = \begin{bmatrix} Y_{i_1} \\ Y_{i_2} \\ \vdots \\ Y_{i_m} \end{bmatrix}^T \in \mathbb{R}^{md}.$$

Define $M = M(\nabla h) \in \mathbb{R}^{\mathcal{C}_m^n \times md}$ with the row indexed by $\mathbf{i} = (i_1, i_2, \dots, i_m) \in \mathcal{C}_m^n$ given by

$$M_{\mathbf{i}} = (\nabla h)(X_{\mathbf{i}}) = (\nabla h)\left(\begin{bmatrix} X_{i_1} \\ X_{i_2} \\ \vdots \\ X_{i_m} \end{bmatrix} \right) \in \mathbb{R}^{md}.$$

That is, the row of M indexed by $\mathbf{i} \in \mathcal{C}_m^n$ is the gradient of $h : \mathbb{R}^{md} \to \mathbb{R}$ evaluated at $X_{\mathbf{i}} = [X_{i_1}^T, X_{i_2}^T, \dots, X_{i_m}^T]^T$. With these three definitions in hand, we have

$$\sum_{\mathbf{i}\in\mathcal{C}_m^n} v_{\mathbf{i}} (\hat{X}Q - X)_{\mathbf{i}}^T (\nabla h)(X_{\mathbf{i}}) = \operatorname{tr} M^T \operatorname{diag}(v) \mathcal{T}_m(\hat{X}Q - X),$$

where $\operatorname{diag}(v) \in \mathbb{R}^{\mathcal{C}_m^n \times \mathcal{C}_m^n}$ denotes the diagonal matrix with entries given by the elements of v. Using the fact that $X = US^{1/2} = PUS^{-1/2}$ and $\hat{X} = \hat{U}\hat{S}^{1/2} = A\hat{U}\hat{S}^{-1/2}$, adding and subtracting appropriate quantities, and using the linearity of the trace and $\mathcal{T}_m$, we have

$$\begin{aligned} &\sum_{\mathbf{i}\in\mathcal{C}_m^n} v_{\mathbf{i}} (\hat{X}Q - X)_{\mathbf{i}}^T (\nabla h)(X_{\mathbf{i}}) \\ &\quad = \operatorname{tr}\left[M^T \operatorname{diag}(v) \mathcal{T}_m \left(A\hat{U}(\hat{S}^{-1/2}Q - QS^{-1/2}) \right) \right] \\ &\qquad + \operatorname{tr}\left[M^T \operatorname{diag}(v) \mathcal{T}_m \left(A(\hat{U}Q - U)S^{-1/2} \right) \right] \\ &\qquad + \operatorname{tr}\left[M^T \operatorname{diag}(v) \mathcal{T}_m \left((A - P)US^{-1/2} \right) . \right] \end{aligned}$$

Applying the triangle inequality, Cauchy-Schwarz and submultiplicativity, we have

$$\begin{aligned}
&\left|\sum_{\mathbf{i}\in\mathcal{C}_m^n} v_{\mathbf{i}}(\hat{X}Q-X)_{\mathbf{i}}^T(\nabla h)(X_{\mathbf{i}})\right| \\
&\quad\le \|M\|_F\|\operatorname{diag}(v)\|\Big[\|\mathcal{T}_m(A\hat{U}(\hat{S}^{-1/2}Q-QS^{-1/2}))\|_F \\
&\qquad\qquad + \|\mathcal{T}_m(A(\hat{U}Q-U)S^{-1/2})\|_F\Big] \\
&\qquad + \left|\operatorname{tr}\left[M^T\operatorname{diag}(v)\mathcal{T}_m((A-P)US^{-1/2})\right]\right|.
\end{aligned} \tag{18}$$

By definition of $\mathcal{T}_m$, each row of $Z\in\mathbb{R}^{n\times d}$ appears in $\binom{n-1}{m-1}$ rows of $\mathcal{T}_m(Z)$, and thus $\|\mathcal{T}_m(Z)\|_F=\binom{n-1}{m-1}^{1/2}\|Z\|_F$. Using this fact and applying Lemma B.2,

$$\begin{aligned}
\left\|\mathcal{T}_m(A\hat{U}(\hat{S}^{-1/2}Q-QS^{-1/2}))\right\|_F &= \binom{n-1}{m-1}^{1/2}\left\|A\hat{U}(\hat{S}^{-1/2}Q-QS^{-1/2})\right\|_F \\
&\le C\binom{n-1}{m-1}^{1/2} n^{-1/2}\log n
\end{aligned} \tag{19}$$

Similarly, this time using Lemma B.3,

$$\begin{aligned}
\left\|\mathcal{T}_m(A(\hat{U}Q-U)S^{-1/2})\right\|_F &\le \binom{n-1}{m-1}^{1/2}\left\|A(\hat{U}Q-U)S^{-1/2}\right\|_F \\
&\le C\binom{n-1}{m-1}^{1/2} n^{-1/2}\log n.
\end{aligned} \tag{20}$$

Combining Equations (19) and (20) and using the fact that

$$\|M\|_F=\left(\sum_{\mathbf{i}\in\mathcal{C}_m^n}\|(\nabla h)(X_{\mathbf{i}})\|^2\right)^{1/2}\le C\sqrt{\binom{n}{m}}$$

by Assumption 2, it follows that with probability at least $1-Cn^{-2}$,

$$\begin{aligned}
&\|M\|_F\|\operatorname{diag}(v)\|\Big[\|\mathcal{T}_m(A\hat{U}(\hat{S}^{-1/2}Q-QS^{-1/2}))\|_F \\
&\qquad + \|\mathcal{T}_m(A(\hat{U}Q-U)S^{-1/2})\|_F\Big] \\
&\quad\le Cn^{-1/2}\|\operatorname{diag}(v)\|\left(\binom{n}{m}\binom{n-1}{m-1}\right)^{1/2}\log n \\
&\quad\le C\|\operatorname{diag}(v)\|\binom{n-1}{m-1}\log n,
\end{aligned} \tag{21}$$

where we have used the fact that m is assumed constant in n.

Returning to Equation (18), it remains to bound

$$\left|\operatorname{tr}\left(M^T \operatorname{diag}(v)\mathcal{T}_m((A-P)US^{-1/2})\right)\right|.$$

By definition, for $\mathbf{i} \in \mathcal{C}_m^n, k \in [m], s \in [d]$,

$$\left(\mathcal{T}_m((A-P)US^{-1/2})\right)_{\mathbf{i},d(k-1)+s} = \left[(A-P)US^{-1/2}\right]_{i_k,s}.$$

For any $\mathbf{i} = (i_1, i_2, \dots, i_m) \in \mathcal{C}_m^n$ and $j \in [n]$, if $j = i_k$ for some $k \in [m]$, define $\tau(j, \mathbf{i}) = k$. With this notation in hand, define the matrix $\tilde{M} \in \mathbb{R}^{n \times d}$ by

$$\tilde{M}_{j,s} = \sum_{\mathbf{i} \in \mathcal{C}_m^n : j \in \mathbf{i}} v_{\mathbf{i}} M_{\mathbf{i},\tau(j,\mathbf{i})+s}, \quad j \in [n], s \in [d],$$

and note that for some constant $C_{F,h} < \infty$ depending on h and F but not depending on n,

$$\left|\tilde{M}_{j,s}\right| \le C_{F,h} \binom{n-1}{m-1} \|\operatorname{diag}(v)\|, \tag{22}$$

where we have again used Assumption 2. With this definition, let $u_s \in \mathbb{R}^n$, $s = 1, 2, \dots, d$ be the eigenvectors of P with non-zero eigenvalues (i.e., the columns of U), so that $u_{s,i}$ denotes the i-th entry of the s-th eigenvector of P. Then

$$\begin{aligned}
\operatorname{tr} M^T \operatorname{diag}(v)\mathcal{T}_m((A-P)US^{-1/2}) &= \sum_{\mathbf{i} \in \mathcal{C}_m^n} v_{\mathbf{i}} \sum_{j=1}^{md} M_{\mathbf{i},j} \mathcal{T}_m\left((A-P)US^{-1/2}\right)_{\mathbf{i},j} \\
&= \sum_{\mathbf{i} \in \mathcal{C}_m^n} v_{\mathbf{i}} \sum_{k=1}^{m} \sum_{s=1}^{d} M_{\mathbf{i},m(k-1)+s} \left[(A-P)US^{-1/2}\right]_{i_k,s} \\
&= \sum_{s=1}^{d} \lambda_s^{-1/2} \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{M}_{i,s}(A-P)_{i,j} u_{s,j} \\
&= \sum_{s=1}^{d} 2\lambda_s^{-1/2} \sum_{1 \le i < j \le n} \tilde{M}_{i,s}(A-P)_{i,j} u_{s,j} - \sum_{s=1}^{d} \lambda_s^{-1/2} \sum_{i=1}^{n} \tilde{M}_{i,s} P_{i,i} u_{s,i}.
\end{aligned} \tag{23}$$

The second term is bounded by

$$\begin{aligned}
\left|\sum_{s=1}^{d} \lambda_s^{-1/2} \sum_{i=1}^{n} \tilde{M}_{i,s} P_{i,i} u_{s,i}\right| &\le \sum_{s=1}^{d} \lambda_s^{-1/2} \left(\sum_{i=1}^{n} \tilde{M}_{i,s}^2 P_{i,i}^2\right)^{1/2} \|u_s\| \\
&\le Cd \|\operatorname{diag}(v)\| \binom{n-1}{m-1},
\end{aligned} \tag{24}$$

where the first inequality follows from the Cauchy-Schwarz inequality, and the second inequality follows from Equation (22), the fact that $\|u_s\| = 1$, the fact that $P_{i,i}^2 \leq 1$, and Equation (13). For fixed $s \in [d]$, the sum over $1 \leq i < j \leq n$ in (23) is a sum of independent mean-0 random variables. Hoeffding's inequality combined with Equation (22) yields

$$\begin{aligned}\Pr\left[\left|\sum_{1 \leq i < j \leq n} \tilde{M}_{i,s}(A-P)_{i,j}u_{s,j}\right| \geq t\right] &\leq 2\exp\left\{\frac{-2t^2}{\|u_s\|^2 \sum_{i=1}^n \tilde{M}_{i,s}^2}\right\} \\ &\leq 2\exp\left\{\frac{-2t^2}{Cn\|\operatorname{diag}(v)\|^2\binom{n-1}{m-1}^2}\right\}.\end{aligned}$$

Taking $t = C\|\operatorname{diag}(v)\|\binom{n-1}{m-1}\sqrt{n\log n}$ for suitably large constant $C > 0$, a union bound over all $s \in [d]$ implies that with probability $1 - Cn^{-2}$, it holds for all $s \in [d]$ that

$$\left|\sum_{1 \leq i < j \leq n} \tilde{M}_{i,s}(A-P)_{i,j}u_{s,j}\right| \leq C\|\operatorname{diag}(v)\|\binom{n-1}{m-1}\sqrt{n\log n}.$$

Applying Equation (13) to bound $\lambda_s^{-1/2}$ and using Assumption 2, it holds with probability at least $1 - Cn^{-2}$ that

$$\left|\sum_{s=1}^d 2\lambda_s^{-1/2} \sum_{1 \leq i < j \leq n} \tilde{M}_{i,s}(A-P)_{i,j}u_{s,j}\right| \leq Cd\|\operatorname{diag}(v)\|\binom{n-1}{m-1}\log^{1/2} n.$$

Combining this with Equation (24), both sums in (23) are bounded by

$$Cd\|\operatorname{diag}(v)\|\binom{n-1}{m-1}\log^{1/2} n,$$

and it holds with probability at least $1 - Cn^{-2}$ that, since d is a constant,

$$\left|\operatorname{tr} M^T \mathcal{T}_m((A-P)US^{-1/2})\right| \leq C\|\operatorname{diag}(v)\|\binom{n-1}{m-1}\log^{1/2} n.$$

Applying this and Equation (21) to Equation (18), we have

$$\left|\sum_{\mathbf{i} \in \mathcal{C}_m^n} v_{\mathbf{i}}(\hat{X}Q - X)_{\mathbf{i}}^T(\nabla h)(X_{\mathbf{i}})\right| \leq C\|\operatorname{diag}(v)\|\binom{n-1}{m-1}\log n,$$

and the result follows by construction of $\operatorname{diag}(v)$. ☐

Lemma C.2. *Let* $(A, X) \sim \operatorname{RDPG}(F, n)$ *for some* d*-dimensional inner product distribution* F*, so that* $X_1, X_2, \dots, X_n \overset{i.i.d.}{\sim} F$ *and let* $\hat{X} = \operatorname{ASE}(A, d)$*, with rows given by* $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_n \in \mathbb{R}^d$*. For each* $\mathbf{i} \in \mathcal{C}_m^n$*, let* $Z_{\mathbf{i}} \in \mathbb{R}^{md}$ *be some point on the line segment connecting* $(\hat{X}Q)_{\mathbf{i}}$ *and* $X_{\mathbf{i}}$*. Suppose that* $h : (\mathbb{R}^d)^m \to \mathbb{R}$ *a* *kernel, symmetric in its arguments, satisfying Assumptions 1 and 2. Let* $v \in \mathbb{R}^{\mathcal{C}_m^n}$ *be a fixed vector and let* $Q \in \mathbb{R}^{d \times d}$ *be the orthogonal matrix guaranteed by Lemma 2.3. For all suitably large* n*, with probability at least* $1 - Cn^{-2}$*,*

$$\left| \sum_{\mathbf{i} \in \mathcal{C}_m^n} v_{\mathbf{i}} (\hat{X}Q - X)_{\mathbf{i}}^T (\nabla^2 h)(Z_{\mathbf{i}}) (\hat{X}Q - X)_{\mathbf{i}} \right| \leq C \max_{\mathbf{i} \in \mathcal{C}_m^n} |v_{\mathbf{i}}| \binom{n-1}{m-1} \log^2 n.$$

Proof. Let $Q \in \mathbb{R}^{d \times d}$ be the orthogonal matrix guaranteed to exist with high probability by Lemma 2.3. By Assumption 2, Lemma 2.3 implies that eventually $X_{\mathbf{i}}, (\hat{X}Q)_{\mathbf{i}} \in \mathcal{S}$ for all $\mathbf{i} \in \mathcal{C}_m^n$, and thus also $Z_{\mathbf{i}} \in \mathcal{S}$ for all $\mathbf{i} \in \mathcal{S}$. Applying the triangle inequality, Cauchy-Schwarz and Assumption 2,

$$\begin{aligned} &\left| \sum_{\mathbf{i} \in \mathcal{C}_m^n} v_{\mathbf{i}} (\hat{X}Q - X)_{\mathbf{i}}^T (\nabla^2 h)(Z_{\mathbf{i}}) (\hat{X}Q - X)_{\mathbf{i}} \right| \\ &\qquad \leq \sum_{\mathbf{i} \in \mathcal{C}_m^n} |v_{\mathbf{i}}| \, \|(\hat{X}Q - X)_{\mathbf{i}}\|_{2,\infty}^2 \|(\nabla^2 h)(Z_{\mathbf{i}})\| \\ &\qquad \leq C_{F,h} \max_{\mathbf{i} \in \mathcal{C}_m^n} |v_{\mathbf{i}}| \binom{n}{m} \|\hat{X}Q - X\|_{2,\infty}^2. \end{aligned}$$

Applying Lemma 2.3 again, we have that with probability at least $1 - Cn^{-2}$,

$$\left| \sum_{\mathbf{i} \in \mathcal{C}_m^n} v_{\mathbf{i}} (\hat{X}Q - X)_{\mathbf{i}}^T (\nabla^2 h)(Z_{\mathbf{i}}) (\hat{X}Q - X)_{\mathbf{i}} \right| \leq C \max_{\mathbf{i} \in \mathcal{C}_m^n} |v_{\mathbf{i}}| \binom{n}{m} \frac{\log^2 n}{n}. \tag{25}$$

Using the fact that $n^{-1}\binom{n}{m} = m^{-1}\binom{n-1}{m-1}$ and m is constant in n completes the proof. □

C.1. Proof of Theorem 3.1

Proof. For $\mathbf{i} \in \mathcal{C}_m^n$ and $X \in \mathbb{R}^{n \times d}$, define

$$X_{\mathbf{i}} = \begin{bmatrix} X_{i_1} \\ X_{i_2} \\ \vdots \\ X_{i_m} \end{bmatrix} \in \mathbb{R}^{md}.$$

Viewing the function $h : (\operatorname{supp} F)^m \to \mathbb{R}$ as as $h : \mathbb{R}^{md} \to \mathbb{R}$ and applying a second-order multivariate Taylor expansion,

$$\begin{aligned}\sqrt{n}\left(\hat{U}_n - U_n\right) &= \sqrt{n}\binom{n}{m}^{-1} \sum_{\mathbf{i} \in \mathcal{C}_m^n} \left(h(\hat{X}_{i_1}, \dots, \hat{X}_{i_m}) - h(X_{i_1}, \dots, X_{i_m})\right) \\ &= \sqrt{n}\binom{n}{m}^{-1} \sum_{\mathbf{i} \in \mathcal{C}_m^n} (\hat{X}Q - X)_{\mathbf{i}}^T (\nabla h)(X_{\mathbf{i}}) \qquad (26) \\ &\quad + \frac{\sqrt{n}}{2\binom{n}{m}} \sum_{\mathbf{i} \in \mathcal{C}_m^n} (\hat{X}Q - X)_{\mathbf{i}}^T (\nabla^2 h)(Z_{\mathbf{i}})(\hat{X}Q - X)_{\mathbf{i}},\end{aligned}$$

where $Q \in \mathbb{O}_d$ is the orthogonal matrix guaranteed by Lemma 2.3 and $Z_{\mathbf{i}} \in \mathbb{R}^{md}$ lies on the line segment connecting $(\hat{X}Q)_{\mathbf{i}}$ and $X_{\mathbf{i}}$. Lemma C.1, with $v_c = \sqrt{n}\binom{n}{m}^{-1}$ for all $c \in \mathcal{C}_m^n$, implies

$$\left| \sqrt{n}\binom{n}{m}^{-1} \sum_{\mathbf{i} \in \mathcal{C}_m^n} (\hat{X}Q - X)_{\mathbf{i}}^T (\nabla h)(X_{\mathbf{i}}) \right| \le \frac{C \log n}{\sqrt{n}}.$$

Lemma C.2 similarly implies that

$$\left| \frac{\sqrt{n}}{2\binom{n}{m}} \sum_{\mathbf{i} \in \mathcal{C}_m^n} (\hat{X}Q - X)_{\mathbf{i}}^T (\nabla^2 h)(Z_{\mathbf{i}})(\hat{X}Q - X)_{\mathbf{i}} \right| \le \frac{C \log^2 n}{\sqrt{n}},$$

both holding with probability $1 - Cn^{-2}$, and thus

$$\left| \sqrt{n}\left(\hat{U}_n - U_n\right) \right| \le \frac{C \log^2 n}{\sqrt{n}}.$$

The Borel-Cantelli lemma implies that $\sqrt{n}(\hat{U}_n - U_n) \to 0$ almost surely, as we wished to show. ☐

C.2. Proof of Theorem 3.3

Proof. We prove the convergence $\sqrt{n}(\hat{U}_n^* - \hat{U}_n) \xrightarrow{\mathcal{L}} \mathcal{N}(0, m^2\zeta_1)$. The proof for the unweighted adjacency spectral embedding bootstrap $\hat{V}_n^*$ follows by a similar argument, and thus details are omitted.

Adding and subtracting appropriate quantities,

$$\sqrt{n}(\hat{U}_n^* - \hat{U}_n) = \sqrt{n}(\hat{U}_n^* - U_n^*) + \sqrt{n}(U_n^* - U_n) + \sqrt{n}(U_n - \hat{U}_n).$$

By Theorem 3.1, the last term converges to zero in probability. By our assumption that U_n^* is distributionally consistent, $\sqrt{n}(U_n^* - U_n) \to \mathcal{N}(0, m^2\zeta_1)$ in law. Thus, by Slutsky's theorem, it will suffice for us to show that

$$\sqrt{n}(\hat{U}_n^* - U_n^*) \to 0 \quad \text{in probability.} \qquad (27)$$

Applying an expansion similar to that in Equation (26) above, we have

$$\begin{aligned}\sqrt{n}(\hat{U}_n^* - U_n^*) &= \sqrt{n} \sum_{\mathbf{i} \in \mathcal{C}_m^n} \mathbb{W}_{\mathbf{i}} \left(h(\hat{X}_{i_1}, \hat{X}_{i_2}, \ldots, \hat{X}_{i_m}) - h(X_{i_1}, X_{i_2}, \ldots, X_{i_m}) \right) \\ &= \sqrt{n} \sum_{\mathbf{i} \in \mathcal{C}_m^n} \mathbb{W}_{\mathbf{i}} (\hat{X}Q - X)_{\mathbf{i}}^T (\nabla h)(X_{\mathbf{i}}) \\ &\quad + \frac{\sqrt{n}}{2} \sum_{\mathbf{i} \in \mathcal{C}_m^n} \mathbb{W}_{\mathbf{i}} (\hat{X}Q - X)_{\mathbf{i}}^T (\nabla^2 h)(Z_{\mathbf{i}}) (\hat{X}Q - X)_{\mathbf{i}}\end{aligned}$$

Condition on the weight vector $\mathbb{W} \in \mathbb{R}^{\mathcal{C}_m^n}$, which is independent of $(A, X) \sim \mathrm{RDPG}(F, n)$. Applying Lemmas C.1 and C.2 with $v_{\mathbf{i}} = \sqrt{n}\mathbb{W}_{\mathbf{i}} / \binom{n}{m}$ implies that with high probability,

$$\sqrt{n}(\hat{U}_n^* - U_n^*) \leq C\sqrt{n} \binom{n-1}{m-1} \frac{\max_{\mathbf{i} \in \mathcal{C}_m^n} |\mathbb{W}_{\mathbf{i}}|}{\binom{n}{m}} \log^2 n \leq \frac{C \max_{\mathbf{i} \in \mathcal{C}_m^n} |\mathbb{W}_{\mathbf{i}}| \log^2 n}{\sqrt{n}},$$

where we have again used the fact that m is constant in n. Unconditioning, Assumption 3 ensures that

$$\sqrt{n}(\hat{U}_n^* - U_n^*) = o(1),$$

which completes the proof. ☐

Appendix D: Proof of Theorems 3.4 and 3.5

Here we give proofs of the sparsity results discussed in Section 3. We first need to ensure that in scaling the latent positions, we do not break the recovery guarantees of the adjacency spectral embedding.

Lemma D.1. *Let $\rho_n \to 0$ be a sparsity parameter, satisfying $\rho_n n = \omega(\log n)$. Let F be a distribution on $\mathbb{R}^d$ with the property that for all suitably large n it holds for all $x, y \in \operatorname{supp} F$ that $\rho_n x^T y \in [0, 1]$. Let $X_1, X_2, \ldots, X_n$ be drawn i.i.d. from F and, conditional on these n points, for all suitably large n such that the Bernoulli success parameter makes sense, generate symmetric adjacency matrix A with independent entries $A_{ij} \sim \mathrm{Bern}(\rho_n X_i^T X_j)$. Letting $\hat{X} = \mathrm{ASE}(A, d)$, there exists a sequence of orthogonal matrices $Q \in \mathbb{O}_d$ such that*

$$\|\hat{X}Q - \sqrt{\rho}X\|_{2,\infty} \leq \frac{\log n}{\sqrt{\rho n}}$$

Proof. Writing $\mathbb{E}[A \mid X] = \rho P = \rho X X^T \in \mathbb{R}^{n \times n}$ and letting $\kappa(M)$ denote the ratio of the largest and smallest non-zero singular values of matrix M (i.e., the condition number ignoring zero eigenvalues), using Lemma 1 in [39], there exists a matrix $Q \in \mathbb{O}_d$ such that with high probability

$$\begin{aligned}\|\hat{X}Q - \sqrt{\rho}X\|_{2,\infty} \leq &\frac{C\|(A - \rho P)U\|_{2,\infty}}{\sqrt{\lambda_d(\rho P)}} + \frac{C\|U^T(A - \rho P)U\|_F}{\sqrt{\lambda_d(\rho P)}} \\ &+ \frac{C\|A - \rho P\|^2 \kappa(\rho P)}{\lambda_d^{3/2}(\rho P)},\end{aligned} \tag{28}$$

provided that

$$\|A - \rho P\| < C_0 \lambda_d(\rho P) \tag{29}$$

for some nonnegative constant $C_0 < 1$. Our assumption that $n\rho = \omega(\log n)$ is enough to ensure that Theorem 3.1 in [51] applies, and it follows that

$$\|A - \rho P\| = O(\sqrt{\rho n \log n})$$

By Equation (13) in Lemma B.1, we have $\lambda_d(P) = \Theta(n)$, whence $\lambda_d(\rho P) = \Theta(\rho n)$. Since $\rho n = \omega(\log n)$ by assumption, it follows that $\|A - \rho P\| = o(\lambda_d(\rho P))$, and we conclude that the bound in Equation (29) holds eventually, and thus so does the bound in Equation (28).

We turn now to bounding the right-hand side of Equation (28). For fixed $k, \ell \in [d]$, consider the quantity

$$R_{k\ell} = \left(U^T(A - \rho P)U\right)_{k,\ell} = \sum_{i,j}(A_{ij} - \rho P_{ij})U_{i,k}U_{j,\ell}.$$

By Bernstein's inequality, for $t > 0$,

$$\Pr\left[|R_{k\ell}| > t\right] \le 2\exp\left\{\frac{-t^2}{2\nu_{k,\ell} + 2t/3}\right\}, \tag{30}$$

where

$$\nu_{k\ell} = 2\sum_{i<j}\operatorname{Var}\left((A_{ij} - \rho P_{ij})U_{i,k}U_{j,\ell}\right) = 2\sum_{i<j}\rho P_{ij}(1 - \rho P_{ij})U_{i,k}^2 U_{j,\ell}^2.$$

Using the fact that $0 \le P_{ij} \le 1$ for all i, j and the fact that the columns of U are orthonormal, we have $\nu_{k\ell} \le \rho \le 1$. Thus, taking $t = C\log n$ in Equation (30) for suitably large constant $C > 0$, we conclude that

$$\Pr\left[|R_{k\ell}| > C\log n\right] \le 2n^{-2}$$

Recalling that the dimension d is constant, a union bound over all $k, \ell \in [d]$ and an application of the Borel-Cantelli Lemma implies that

$$\|U^T(A - \rho P)U\|_F = O(\log n). \tag{31}$$

A similar argument shows that

$$\|(A - \rho P)U\|_{2,\infty} = O(\log n). \tag{32}$$

Applying Equations (29), (31) and (32) to the right-hand side of Equation (28),

$$\|\hat{X} - \rho X\|_{2,\infty} \le \frac{C\log n}{\sqrt{\rho n}} + \frac{C\kappa(\rho P)\rho n \log n}{(\rho n)^{3/2}}.$$

Again using Equation (13) in Lemma B.1, $\kappa(\rho P) = \lambda_1(\rho P)/\lambda_d(\rho P) = O(1)$, and thus

$$\|\hat{X} - \rho X\|_{2,\infty} \le \frac{C\log n}{\sqrt{\rho n}},$$

which completes the proof. □

D.1. Proof of Theorem 3.4

Proof. Using arguments similar to the proof of Lemma D.1 above, one can establish sparse analogues of Lemmas B.1, C.1 and C.2. Details are omitted. Theorem 3.4 then follows by precisely the same line of argument as that used to prove Theorem 3.1. ☐

D.2. Proof of Theorem 3.5

Proof. The proof follows a similar use of Slutsky's theorem to that in Theorem 3.3, using Theorem 3.4 followed by a delta method argument applied to the ratio of $U_n - \rho_n^r \mathbb{E} h(X_1, X_2, \dots, X_m)$ and $\hat{\rho}_n / \rho_n$. The delta method argument is broadly similar to that of Theorem 1 in [12] and is thus omitted. ☐

Appendix E: Proof of Theorem 4.4

Proof. Fix $\epsilon > 0$. Since orthogonal transformation of the latent positions does not change the graphs' distributions, we may assume without loss of generality that $\mathring{d}_1(F_1, F_2) = d_1(F_1, F_2)$, i.e., that $Q = I$ is the minimizer in the right-hand side of

$$\mathring{d}_1(F_1, F_2) = \min_{Q \in \mathbb{O}_d} d_1(F_1, F_2 \circ Q).$$

By definition of the Wasserstein distance d_1, there exists a coupling ν of $X_1 \sim F_1$ and $Z_1 \sim F_2$ such that

$$\int \|X_1 - Z_1\| d\nu \le d_1(F_1, F_2) + \epsilon. \tag{33}$$

We will use this coupling ν to construct a coupling of A and H. Draw pairs

$$(X_1, Z_1), (X_2, Z_2), \dots, (X_n, Z_n) \overset{\text{i.i.d.}}{\sim} \nu.$$

It is a basic fact of Bernoulli random variables that if $\xi_1 \sim \text{Bern}(p_1)$ and $\xi_2 \sim \text{Bern}(p_2)$, then $d_1(\xi_1, \xi_2) \le |p_1 - p_2|$. Using this fact, conditional on (X, Z), we can couple (A_{ij}, H_{ij}) for each $i < j$ so that

$$\Pr[A_{i,j} \ne H_{i,j} \mid X_i, X_j, Z_i, Z_j] \le |X_i^T X_j - Z_i^T Z_j|. \tag{34}$$

By construction, $(A, X) \sim \text{RDPG}(F_1, n)$ and $(H, Z) \sim \text{RDPG}(F_2, n)$ marginally, so this scheme yields a valid coupling of A and H, which we denote $(A, H) \sim \tilde{\nu}$, and thus

$$W_p^p(A, H) \le \int d_{\text{GM}}^p(A, H) d\tilde{\nu}(A, H).$$

By the definition of d_{GM}, Jensen's inequality, and the fact that A and H are binary, we have

$$\begin{aligned} d^p_{\mathrm{GM}}(A,H) &\le \left(\frac{1}{2} \binom{n}{2}^{-1} \|A-H\|_1 \right)^p \\ &\le \binom{n}{2}^{-1} \sum_{i<j} |A-H|^p_{i,j} = \binom{n}{2}^{-1} \sum_{i<j} |A-H|_{i,j}, \end{aligned}$$

whence

$$\int d^p_{\mathrm{GM}}(A,H) d\tilde{\nu}(A,H) \le \binom{n}{2}^{-1} \sum_{i<j} \tilde{\nu}\left(\{A_{ij} \ne H_{ij}\}\right). \tag{35}$$

Since Equation (34) holds under the coupling $\tilde{\nu}$, we have

$$\tilde{\nu}\left(\{A_{ij} \ne H_{ij}\}\right) \le \int\int |X_i^T X_j - Z_i^T Z_j| d\nu(X_i, Z_i) d\nu(X_j, Z_j).$$

We can therefore further bound Equation (35) by

$$\begin{aligned} \int d^p_{\mathrm{GM}}(A,H) d\tilde{\nu}(A,H) &\le \int\int |X_1^T X_2 - Z_1^T Z_2| d\nu(X_1,Z_1) d\nu(X_2,Z_2) \\ &\le \int\int \left(\|X_1\| + \|Z_1\|\right) \|X_2 - Z_2\| d\nu d\nu \\ &\le 2 \int \|X_2 - Z_2\| d\nu \le d_1(F_1, F_2) + \epsilon, \end{aligned}$$

where we have used the fact that both F_1 and F_2, being inner product distributions, have supports contained in the unit ball, and the last inequality follows from Equation (33). Thus, we conclude that

$$W_p^p(A,H) \le 2\left(d_1(F_1,F_2) + \epsilon\right),$$

and the result follows since $\epsilon > 0$ was arbitrary. □

E.1. Proof of Theorem 4.4

Proof. Let us first fix notation. Recall that $(A,X) \sim \mathrm{RDPG}(F,n)$ and that $(H,Z) \sim \mathrm{RDPG}(F,n)$ independently of (A,X). Let $F_n = n^{-1}\sum_{i=1}^n \delta_{X_i}$ denote the empirical distribution of the true latent positions of A, and, conditional on X, let $(A^*, X^*) \sim \mathrm{RDPG}(F_n, n)$. Letting $\hat{F}_n$ denote the empirical distribution of the ASE estimates $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_n$, by definition of $\hat{A}^*$, we have that conditional on A, $(\hat{A}^*, \hat{X}^*) \sim \mathrm{RDPG}(\hat{F}, n)$ analogously. By the triangle inequality,

$$W_p(H, \hat{A}^*) \le W_p(A^*, H) + W_p(A^*, \hat{A}^*). \tag{36}$$

By Lemma 4.3, we have

$$W_p^p(A^*, H) \le 2d_1(F_n, F) = O(n^{-1/d}\log n), \tag{37}$$

where we have used the fact that d-dimensional product distributions have bounded support (hence all moments of $X_1 \sim F$ are finite) to apply Theorem 3.1 and Corollary 5.2 from [35] (with $q = \infty$ and $p = 1$ in the notation of that paper) to bound $d_1(F_n, F) = O(n^{-1/(2\vee d)} \log n)$. To bound $W_p(A^*, \hat{A}^*)$, we will construct a coupling similar to that in the proof of Lemma 4.3.

Letting $\xi = (\xi_1, \xi_2, \dots, \xi_n)$ be a vector of independent draws from the uniform distribution on $[n]$, we can write $\hat{X}_i^* = \hat{X}_{\xi_i}$ for each $i \in [n]$ and $X_i^* = X_{\xi_i}$ analogously. We couple the latent positions of $\hat{A}^*$ and A^* through the Conditional on X, A and ξ, we couple the entries of $\hat{A}^*$ and A^* via the same coupling construction used in the proof of Lemma 4.3 above, so that

$$\Pr[A_{i,j}^* \neq \hat{A}_{i,j}^* \mid A, X, \xi] \leq |X_{\xi_i}^T X_{\xi_j} - \hat{X}_{\xi_i}^T \hat{X}_{\xi_j}|.$$

Letting ν denote the resulting joint measure on $(\xi, X, A, A^*, \hat{A}^*)$,

$$W_p^p(A^*, \hat{A}^*) \leq \int \left(\frac{\|A^* - \hat{A}^*\|_1}{n(n-1)} \right)^p d\nu \leq \int \frac{\|A^* - \hat{A}^*\|_1}{n(n-1)} d\nu, \tag{38}$$

where we have used Jensen's inequality and the fact that A^* and $\hat{A}^*$ are binary, as in the proof of Lemma 4.3. We will proceed to bound the integral on the right-hand side. Let E_n denote the event that the bound in Lemma 2.3 holds. On E_n^c, we can trivially bound $d_{\mathrm{GM}}(A^*, \hat{A}^*)$ by 1. Since E_n depends only on A and X, and the marginal distribution of (A, X) under ν is $(A, X) \sim \mathrm{RDPG}(F, n)$ by construction of ν, Lemma 2.3 implies $\nu(E_n^c) = O(n^{-2})$. Thus,

$$\begin{aligned} \int \frac{\|A^* - \hat{A}^*\|_1}{n(n-1)} d\nu &\leq \int_{E_n} \frac{\|A^* - \hat{A}^*\|_1}{n(n-1)} d\nu + O(n^{-2}) \\ &= \sum_{i<j} \frac{\nu\left(\left\{ A_{i,j}^* \neq \hat{A}_{i,j}^* \right\}, E_n \right)}{n(n-1)} + O(n^{-2}). \end{aligned}$$

By our construction of the coupling ν, we have

$$\nu\left(\left\{ A_{i,j}^* \neq \hat{A}_{i,j}^* \right\} \right) \leq |X_{\xi_i}^T X_{\xi_j} - \hat{X}_{\xi_i}^T \hat{X}_{\xi_j}|.$$

By Lemma 2.3, when E_n holds, this difference of absolute values is bounded by $O(n^{-1/2} \log n)$, and thus we have

$$\begin{aligned} \int \frac{\|A^* - \hat{A}^*\|_1}{n(n-1)} d\nu &\leq \binom{n}{2}^{-1} \sum_{i<j} \nu\left(\left\{ A_{i,j}^* \neq \hat{A}_{i,j}^* \right\}, E_n \right) + O(n^{-2}) \\ &= O(n^{-1/2} \log n) + O(n^{-2}) = O(n^{-1/2} \log n). \end{aligned}$$

Plugging this bound into Equation (38), we conclude that

$$W_p^p(A^*, \hat{A}^*) = O(n^{-1/2} \log n).$$

Applying this and Equation (37) to Equation (36), we conclude that

$$W_p^p(H, \hat{A}^*) = O\left((n^{-1/2} + n^{-1/d}) \log n\right),$$

as we set out to show. ☐

Acknowledgments

The authors would like to thank Joshua Agterberg, Jesús Arroyo, Avanti Athreya, Joshua Cape, Tianxi Li, Robert Lunde, Vince Lyzinski, Carey Priebe, Purna Sarkar and Ji Zhu for helpful comments and discussions that have greatly improved this paper.

Funding

The authors gratefully acknowledge support from NSF-DMS grants DMS-1646108 and DMS-2052918. In addition, the first author was supported by the University of Wisconsin–Madison, Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation.

References

[1] Abbe, E. (2018). Community Detection and Stochastic Block Models. *Foundations and Trends in Communications and Information Theory* **14** 1-162. MR3827065

[2] Airoldi, E. M., Blei, D. M., Fienberg, S. E. and Xing, E. P. (2008). Mixed Membership Stochastic Blockmodels. *Journal of Machine Learning Research* **9** 1981–2014.

[3] Arcones, M. A. and Giné, E. (1992). On the bootstrap of U and V statistics. *The Annals of Statistics* **20** 655–674. MR1165586

[4] Athreya, A., Fishkind, D. E., Levin, K., Lyzinski, V., Park, Y., Qin, Y., Sussman, D. L., Tang, M., Vogelstein, J. T. and Priebe, C. E. (2018). Statistical inference on random dot product graphs: a survey. *Journal of Machine Learning Research* **18** 1–92. MR3827114

[5] Bai, Z. and Silverstein, J. W. (2004). CLT for linear spectral statistics of large-dimensional sample covariance matrices. *The Annals of Probability* **32** 553–605. MR2040792

[6] Bandeira, A. S. and van Handel, R. (2016). Sharp nonasymptotic bounds on the norm of random matrices with independent entries. *The Annals of Probability* **44** 2479–2506. MR3531673

[7] Barbe, P. and Bertail, P. (1995). *The Weighted Bootstrap. Lecture Notes in Statistics* **98**. Springer-Verlag. MR2195545

[8] Bento, J. and Ioannidis, S. (2018). A Family of Tractable Graph Distances. In *Proceedings of SIAM International Conference on Data Mining.*

[9] Bhattacharyya, S. and Bickel, P. J. (2015). Subsampling Bootstrap of Count Features of Networks. *The Annals of Statistics* **43** 2384–2411. MR3405598

[10] Bickel, P., Choi, D., Chang, X. and Zhang, H. (2013). Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *The Annals of Statistics* **41** 1922–1943. MR3127853

[11] Bickel, P. and Sarkar, P. (2015). Hypothesis testing for automated community detection in networks. *Journal of the Royal Statistical Society Series B* **78** 253–273. MR3453655

[12] Bickel, P. J., Chen, A. and Levina, E. (2011). The method of moments and degree distributions for network models. *The Annals of Statistics* **39** 38–59. MR2906868

[13] Bickel, P. J. and Freedman, D. A. (1981). Some asymptotic Theory for the Bootstrap. *The Annals of Statistics* **9** 1196–1217. MR0630103

[14] Blom, G. (1976). Some properties of incomplete U-statistics. *Biometrika* **63** 573–580. MR0474582

[15] Bose, A. and Chatterjee, S. (2018). *U-statistics, M_m-Estimators and Resampling*. Springer. MR3837543

[16] Chang, J., Kolaczyk, E. D. and Yao, Q. (2020). Estimation of Subgraph Densities in Noisy Networks. *Journal of the American Statistical Association* **117** 361–374. MR4399091

[17] Chartrand, G., Kubicki, G. and Schultz, M. (1998). Graph similarity and distance in graphs. *Aequationes mathematicae* **55** 129–145. MR1600596

[18] Chen, F., Roch, S., Rohe, K. and Yu, S. (2021). Estimating graph dimension with cross-validated eigenvalues. *arXiv:2108.03336*.

[19] Chen, X. and Kato, K. (2019). Randomized incomplete U-statistics in high dimensions. *The Annals of Statistics* **47** 3127–3156. MR4025737

[20] Davezies, L., D'Haultfœuille, X. and Guyonvarch, Y. (2021). Empirical process results for exchangeable arrays. *The Annals of Statistics* **49** 845–862. MR4255110

[21] Efron, B. and Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. Chapman and Hall/CRC. MR1270903

[22] Fiedler, M. (1973). Algebraic connectivity of Graphs. *Czechoslovak Mathematical Journal* **23** 298–305. MR0318007

[23] Fishkind, D. E., Sussman, D. L., Tang, M., Vogelstein, J. T. and Priebe, C. E. (2013). Consistent Adjacency-Spectral Partitioning for the Stochastic Block Model When the Model Parameters Are Unknown. *SIAM Journal on Matrix Analysis and Applications* **34** 23–39. MR3032990

[24] Fosdick, B. K. and Hoff, P. D. (2015). Testing and Modeling Dependencies Between a Network and Nodal Attributes. *Journal of the American Statistical Association* **110** 1047-1056. MR3420683

[25] Green, A. and Shalizi, C. R. (2022). Bootstrapping Exchangeable Random Graphs. *Electronic Journal of Statistics* **16** 1058–1095. MR4377133

[26] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B.

and Smola, A. (2012). A Kernel Two-Sample Test. *Journal of Machine Learning* **13** 723–773. MR2913716

[27] Han, X., Yang, Q. and Fan, Y. (2023). Universal rank inference via residual subsampling with application to large networks. *The Annals of Statistics* **51** 1109–1133. MR4630942

[28] Hoeffding, W. (1948). A class of statistics with asymptotically normal distributions. *The Annals of Statistics* **19** 293–325. MR0026294

[29] 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** 1090–1098. MR1951262

[30] Hušková, M. and Janssen, P. (1993). Consistency of the Generalized Bootstrap for Degenerate U-Statistics. *The Annals of Statistics* **21** 1811–1823. MR1245770

[31] Johnstone, I. M. and Onatski, A. (2020). Testing in high-dimensional spiked models. *The Annals of Statistics* **48** 1231–1254. MR4124321

[32] Lahiri, S. N. (2003). *Resampling Methods for Dependent Data*. Springer. MR2001447

[33] Lee, Y., Shen, C., Priebe, C. E. and Vogelstein, J. T. (2019). Network Dependence Testing via Diffusion Maps and Distance-Based Correlations. *Biometrika* **106** 857–873. MR4031202

[34] Lei, J. (2016). A goodness-of-fit test for stochastic block models. *The Annals of Statistics* **44** 401–424. MR3449773

[35] Lei, J. (2020). Convergence and Concentration of Empirical Measures under Wasserstein Distance in Unbounded Functional Spaces. *Bernoulli* **26** 767–798. MR4036051

[36] Lei, J. (2021). Network Representation Using Graph Root Distributions. *The Annals of Statistics* **49** 745–768. MR4255106

[37] Lei, J. and Rinaldo, A. (2015). Consistency of spectral clustering in stochastic block models. *The Annals of Statistics* **43** 215–237. MR3285605

[38] Levin, K., Athreya, A., Tang, M., Lyzinski, V., Park, Y. and Priebe, C. E. (2017). A central limit theorem for an omnibus embedding of random dot product graphs. *arXiv:1705.09355v5*. MR3494576

[39] Levin, K., Lodhia, A. and Levina, E. (2022). Recovering low-rank structure from multiple networks with unknown edge distributions. *Journal of Machine Learning Research* **23** 1–48. MR4420728

[40] Li, T., Levina, E. and Zhu, J. (2020). Network cross-validation by edge sampling. *Biometrika* **107** 257–276. MR4108931

[41] Lin, Q., Lunde, R. and Sarkar, P. (2020). Trading off Accuracy for Speedup: Multiplier Bootstraps for Subgraph Counts. *arXiv:2009.06170*.

[42] Lovász, L. (2012). *Large Networks and Graph Limits*. American Mathematical Society. MR3363148

[43] Lu, L. and Peng, X. (2013). Spectra of edge-independent random graphs. *Electronic Journal of Combinatorics* **20**. MR3158266

[44] Lunde, R. and Sarkar, P. (2023). Subsampling Sparse Graphons under Minimal Assumptions. *Biometrika* **110** 15–32. MR4565441

[45] Lyzinski, V., Sussman, D. L., Tang, M., Athreya, A. and

Priebe, C. E. (2014). Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding. *Electronic Journal of Statistics* **8** 2905–2922. MR3299126

[46] Lyzinski, V., Tang, M., Athreya, A., Park, Y. and Priebe, C. E. (2017). Community detection and classification in hierarchical stochastic blockmodels. *IEEE Transactions on Network Science and Engineering*. MR3625952

[47] Maugis, P.-A. G., Priebe, C. E., Olhede, S. C. and Wolfe, P. J. (2017). Statistical inference for network samples using subgraph counts. *arXiv:1701.00505*.

[48] Menzel, K. (2021). Bootstrap with Cluster-Dependence in Two or More Dimensions. *Econometrica* **89** 2143–2188. MR4325248

[49] Newman, M. E. J. (2010). *Networks*. Oxford University Press. MR2676073

[50] Olhede, S. C. and Wolfe, P. J. (2014). Network histograms and universality of blockmodel approximation. *Proceedings of the National Academy of Sciences* **111** 14722–14727.

[51] Oliveira, R. I. (2009). Concentration of the adjacency matrix and of the Laplacian in random graphs with independent edges. *arXiv:0911.0600*.

[52] Randić, M. (1975). Characterization of molecular branching. *Journal of the American Chemical Society* **97** 6609–6615.

[53] Rossi, R. A. and Ahmed, N. K. (2015). The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference in Artificial Intelligence* **29**.

[54] Rubin-Delanchy, P., Priebe, C. E., Tang, M. and Cape, J. (2022). A statistical interpretation of spectral embedding: the generalised random dot product graph. *Journal of the Royal Statistical Society Series B* **84** 1446–1473. MR4494166

[55] Serfling, R. J. (1980). *Approximation Theorems of Mathematical Statistics*. Wiley. MR0595165

[56] Shalizi, C. R. and Asta, D. (2024). Consistency of Maximum Likelihood for Continuous-Space Network Models I. *Electronic Journal of Statistics* **18** 335–354. MR4700270

[57] Shan, Q. and Levina, E. (2022). Network resampling for estimating uncertainty. *arXiv:2206.13088*.

[58] Sussman, D. L., Tang, M., Fishkind, D. E. and Priebe, C. E. (2012). A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association* **107** 1119–1128. MR3010899

[59] Székely, G. J. and Rizzo, M. L. (2013). Energy statistics: a class of statistics based on distances. *Journal of Statistical Planning and Inference* **143** 1249–‘172. MR3055745

[60] Tang, M., Athreya, A., Sussman, D. L., Lyzinski, V., Park, Y. and Priebe, C. E. (2017). A Semiparametric Two-Sample Hypothesis Testing Problem for Random Graphs. *Journal of Computational and Graphical Statistics* **26** 344–354. MR3640191

[61] Tang, M., Athreya, A., Sussman, D. L., Lyzinski, V. and Priebe, C. E. (2017). A nonparametric two-sample hypothesis testing problem for random graphs. *Bernoulli* **23** 1599–1630. MR3624872

[62] Tang, M., Cape, J. and Priebe, C. E. (2022). Asymptotically efficient estimators for stochastic blockmodels: the naive MLE, the rank-constrained MLE, and the spectral estimator. *Bernoulli* **28** 1049–1073. MR4388929

[63] Torres, L., Suárez-Serrato, P. and Eliassi-Rad, T. (2018). Graph distance from the topological view of non-backtracking cycles. *arXiv:1807.09592*.

[64] Traud, A. L., Mucha, P. J. and Porter, M. A. (2012). Social structure of Facebook networks. *Physica A: Statistical Mechanics and its Applications* **391** 4165–4180.

[65] Wang, D., Yu, Y. and Rinaldo, A. (2021). Optimal Change Point Detection and Localization in Sparse Dynamic Networks. *The Annals of Statistics* **49** 203–232. MR4206675

[66] Wernicke, S. (2006). Efficient detection of network motifs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **3** 347–359.

[67] Wu, D. and Xie, F. (2022). Statistical inference of random graphs with a surrogate likelihood function. *2207.01702*.

[68] Xie, F. and Xu, Y. (2023). Efficient Estimation for Random Dot Product Graphs via a One-Step Procedure. *Journal of the American Statistical Association* **118** 651–664. MR4571148

[69] Young, S. and Scheinerman, E. (2007). Random dot product graph models for social networks. In *Proceedings of the 5th International Conference on Algorithms and Models for the Web-graph* 138–149. MR2504912

[70] Yu, Y., Wang, T. and Samworth, R. J. (2015). A useful variant of the Davis-Kahan Theorem for statisticians. *Biometrika* **102** 315–323. MR3371006

[71] Zhang, Y. and Xia, D. (2022). Edgeworth expansions for network moments. *The Annals of Statistics* **50** 726–753. MR4404918

[72] Zhu, M. and Ghodsi, A. (2006). Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics & Data Analysis* **51** 918–930. MR2297497