Electronic Journal of Statistics
Vol. 14 (2020) 3230–3267
ISSN: 1935-7524
https://doi.org/10.1214/20-EJS1744

Vertex nomination, consistent estimation, and adversarial modification

Joshua Agterberg

Department of Applied Mathematics and Statistics, Johns Hopkins University
e-mail: jagterb1@jhu.edu

Youngser Park

Center for Imaging Sciences, Johns Hopkins University
e-mail: youngser@jhu.edu

Jonathan Larson and Christopher White

Microsoft AI and Research, Microsoft
e-mail: jolarso@microsoft.com; chwh@microsoft.com

Carey E. Priebe

Department of Applied Mathematics and Statistics, Johns Hopkins University
Center for Imaging Sciences, Johns Hopkins University
e-mail: cep@jhu.edu

Vince Lyzinski

Department of Mathematics, University of Maryland College Park
e-mail: vlyzinsk@umd.edu

Abstract: Given a pair of graphs G_1 and G_2 and a vertex set of interest in G_1, the vertex nomination (VN) problem seeks to find the corresponding vertices of interest in G_2 (if they exist) and produce a rank list of the vertices in G_2, with the corresponding vertices of interest in G_2 concentrating, ideally, at the top of the rank list. In this paper, we define and derive the analogue of Bayes optimality for VN with multiple vertices of interest, and we define the notion of maximal consistency classes in vertex nomination. This theory forms the foundation for a novel VN adversarial contamination model, and we demonstrate with real and simulated data that there are VN schemes that perform effectively in the uncontaminated setting, and adversarial network contamination adversely impacts the performance of our VN scheme. We further define a network regularization method for mitigating the impact of the adversarial contamination, and we demonstrate the effectiveness of regularization in both real and synthetic data.

MSC2020 subject classifications: 62H99.
Keywords and phrases: Statistics, random graphs, networks, adversarial machine learning, vertex nomination.

Received July 2019.

1. Introduction and background

Given graphs G_1 and G_2 and vertices of interest $V^* \subset V(G_1)$, the aim of the vertex nomination (VN) problem is to rank the vertices of G_2 into a nomination list with the corresponding vertices of interest concentrating at the top of the nomination list. In recent years, a host of VN procedures have been introduced (see, for example, [14, 30, 26, 17, 37, 48]) that have proven to be effective information retrieval tools in both synthetic and real data applications. Moreover, recent work establishing a fundamental statistical framework for VN has led to a novel understanding of the limitations of VN efficacy in evolving network environments [27]. Herein, we consider a general statistical model for adversarial contamination in the context of vertex nomination—here the adversary model can both randomly add or remove edges and/or vertices in the network —and we examine the effect of both these contaminations on VN performance. In addition, we extend existing theory on consistent vertex nomination to multiple vertices of interest and define and derive Bayes Optimal Classifiers in this setting. We further show that there are infinitely many classes of distribution for which a vertex nomination scheme is not consistent.

The practical additional value of this paper is to

1. extend the results of [27] to the more realistic multiple VOI setting;
2. rigorously frame the concept of an adversary in the random graph framework;
3. develop theory showing how it is possible for an adversary to render vertex nomination schemes inconsistent;
4. demonstrate empirically that although an adversary can have a negative impact, regularization can succeed in recovering consistency.

The reason we do not prove that regularization succeeds is that the regularization scheme depends on the particular graph observation and introduces complex dependence structure into the problem. Such dependence, coupled with the already difficult spectral analysis problem, makes it unclear what exactly is even being estimated when using any spectral nomination scheme with regularization. Furthermore, the regularization scheme we consider is highly model-dependent, and our main theoretical contributions apply to *any* vertex nomination scheme and as such are necessary to begin to understand adversarial vertex nomination.

To motivate our mathematical and statistical results further, we first consider an illustrative real data example in Section 1.1 in which we demonstrate the following: A VN scheme that works effectively with network contamination adversely impacting the performance of our VN scheme. Note that we will provide a more thorough background of the relevant literature after the motivating example in Section 1.2.

1.1. Motivating example

Consider the pair of high school friendship networks in [32]: The first, G_1, has 156 nodes, each representing a student, and has two vertices adjacent if the two

students made contact with each other at school in a given time period; the second, G_2, has 134 vertices, again with each vertex representing a student, and has two vertices adjacent if the two students are friends on Facebook. There are 82 students appearing in both G_1 and G_2, and we pose the VN problem here as follows: given a student-of-interest in G_1, can we nominate the corresponding student (if they exist) in G_2. We note here that the vertex nomination approach outlined below easily adapts to the multiple vertices of interest (v.o.i.) scenario (i.e., given students-of-interest in G_1, can we nominate the corresponding students, if they exist, in G_2)—and we will provide the necessary details for handling both single and multiple v.o.i. below. Recall that the VN problem assumes there is a correspondence between the vertices but that the practitioner does not have access to this correspondence. To this end, we act as though we do not know the corresponding student in each graph.

In one idealized data setting, all students would appear in both graphs as this would potentially maximize the signal present in the correspondence of labels across graphs. This bears itself out in the following illustrative VN experiment. Consider the following simple VN scheme, which we denote VN $\circ$ GMM $\circ$ ASE: Given vertex (or vertices) of interest v^* in G_1 and seeded vertices $S \subset V_1 \cap V_2$ (seeds here represent vertices whose identity across networks is known a priori), we proceed by embedding the graphs into a common Euclidean space $\mathbb{R}^d$ and clustering using Mahalanobis distances between the embeddings of the vertices (see Section 4.1 for full detail).

We can consider running the VN $\circ$ GMM $\circ$ ASE in the idealized data setting where we only consider the induced subgraphs of G_1 and G_2 containing the 82 common vertices across graphs (call these graphs $G_1^{(i)}$ and $G_2^{(i)}$), and we can also consider running the procedure in the setting where the 52 vertices in G_2 without matches across graphs are added to $G_2^{(i)}$ as a form of contamination. These unmatchable vertices can have the effect of obfuscating the correspondence amongst the common vertices across graphs, and thus can diminish VN performance. Indeed, we see this play out in Figure 1.

In Figure 1, we plot the performance of VN $\circ$ GMM $\circ$ ASE averaged over $nMC = 500$ random seed sets of size $s = 10$. In the left figure, the x-axis shows the ranks in the nomination list and the y-axis shows the mean ($\pm$ 2s.e.) number of vertices $v \in G_1^{(i)}$, when viewed as the lone v.o.i., that had their corresponding vertex of interest ranked in the top x by VN $\circ$ GMM $\circ$ ASE. The right figure shows the same results normalized by chance performance, where we plot

$$y = \frac{\text{mean \# of v.o.i. with corresp. v.o.i. ranked in top } x \text{ by VN}\circ\text{GMM}\circ\text{ASE}}{\text{mean \# of v.o.i. with corresp. v.o.i. ranked in top } x \text{ by chance algorithm}}$$

versus x. The blue line represents performance in the idealized networks $G_1^{(i)}$ and $G_2^{(i)}$, and the red line represents performance in the contaminated network pair $(G_1^{(i)}, G_2)$. We see that the contamination detrimentally affects the performance of VN $\circ$ GMM $\circ$ ASE at all levels, as *for all* x, the number of v.o.i. in $G_1^{(i)}$ with their corresponding v.o.i. ranked in the top x in the second graph is larger in $(G_1^{(i)}, G_2^{(i)})$ versus in $(G_1^{(i)}, G_2)$. Note that the chance normalization is computed

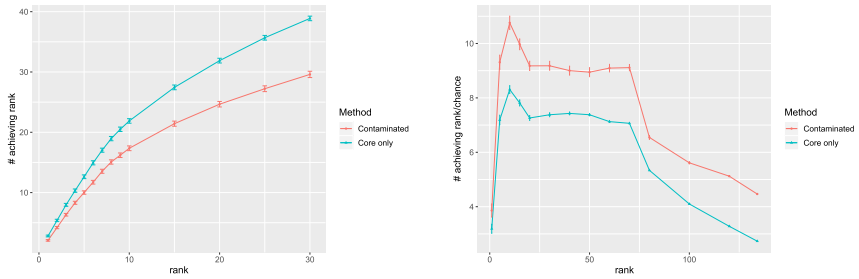


FIG 1. *We plot the performance of VN ∘ GMM ∘ ASE averaged over* $nMC = 500$ *random seed sets of size* $s = 10$. *The left figure shows the number of true vertices achieving the rank, and the right figure shows the same result normalized by chance performance. The blue line represents performance with the truth, and the red line represents the contaminated network. See Section 1.1 for more details.*

separately under the core and noisy models, and the seeming performance gain relative to chance in the contaminated setting is attributable to the fact that G_2 has significantly more vertices than the idealized $G_2^{(i)}$, and chance is therefore significantly worse. We emphasize here the effect of the contamination on VN performance; indeed, the adversarial contamination greatly (negatively) effects the performance of our vertex nomination scheme, suggesting that perhaps the vertex nomination scheme is not consistent for this class of contaminated distributions. In effect, the adversary is knocking the networks out of the consistency class for VN ∘ GMM ∘ ASE; see Section 2.3 for detail. While the results of Section 2.3.2 show that we cannot verify (in an unsupervised manner, without the true labels) the extent to which the contamination negatively impacts the performance of VN, in Section 3.2.1, we empirically explore the impact of regularization strategies for mitigating this contamination.

Remark 1 (The role of seeds). Figure 1 shows performance of VN ∘ GMM ∘ ASE averaged over 500 randomly chosen seed sets of size 10. While performance, on the whole, increases with proper regularization, the story can vary wildly from seed set to seed set. While a full exploration of this is beyond the scope of the present text, this is an active area of our work.

1.2. Background

In modern statistics and machine learning, graphs are a common way to take into account the complex relationships between data objects, and graphs have been used in applications across the biological (see, for example, [42, 7, 1, 31, 21, 33]) and social sciences (see, for example, [35, 41, 20, 22]). In addition to more traditional statistical inference tasks such as clustering [39, 38, 6, 34], classification [46, 11, 1], and estimation [5, 4, 43], there has been significant work in more network-specific inference tasks such as graph matching [12, 18, 47], and vertex nomination [30, 13, 17].

Recall that the vertex nomination problem can be stated loosely as follows: given graphs G_1 and G_2 and vertices of interest $V^* \subset V(G_1)$, rank the vertices of G_2 into a nomination list with the corresponding vertices of interest concentrating at the top of the nomination list (see Definition 10 for full detail). While vertex nomination has found applications in a number of different areas, such as social networks in [37] and data associated with human trafficking in [17], there are relatively few results establishing the statistical properties of vertex nomination. In [17], consistency is developed within the stochastic blockmodel random graph framework, where interesting vertices were defined via community membership. In [27], the authors develop the concepts of consistency and Bayes optimality for a very general class of random graph models and a very general definition of what makes the v.o.i. interesting. In this paper, we further develop the ideas in [27], with the aim of developing a theoretical regime in which to ground the notion of adversarial contamination in VN. In addition, their results are derived in the setting of a single vertex of interest; since many real application problems involve finding similar groups of nodes, we extend their results to multiple vertices of interest.

There has been significant recent attention towards better understanding the impact of adversarial attacks on machine learning methodologies (see, for example, [24, 8, 36, 15, 50]). Herein, we define an adversarial attack on a machine learning algorithm to be a mechanism that changes the data distribution in order to negatively affect algorithmic performance; see Definition 17. From a practical standpoint, adversarial attacks model the very real problem of having data compromised; if an intelligent agent has access to the data and algorithm, the agent may want to modify the data or the algorithm to give the wrong prediction/inferential conclusion. Although there has been much work on adversarial modeling in machine learning, there has been less theory developed for adversarial attacks from a statistical perspective.

The adversarial framework we consider is similar to the model considered in [8], and it is motivated by the example in the previous section in which the addition of the vertices without correspondences to G_2 negatively impacted VN performance. Suppose that we are interested in performing vertex nomination on a graph pair, but an adversary randomly adds and deletes some edges and/or vertices in the second graph. For example, suppose we are trying to find influencers on Instagram by vertex matching to Facebook. An influencer that has knowledge of our procedure may attempt to make our algorithm fail in its nominations, perhaps by friending and de-friending people on Facebook. Even if our vertex nomination scheme was working well prior to encountering the adversary, it may not be after modification by the adversary.

From a statistical standpoint, what can we say about the statistical consistency of our original vertex nomination rule? Our motivating example suggests that there are adversaries that can render our vertex nomination scheme no longer consistent, but theory is needed both to explain why that may be the case and to properly frame the problem. Hence, to answer these questions, we further develop the theory in [27] to situate the notion of adversarial contamination within the idea of maximal consistency classes for a given VN rule (Section 2.3).

In this framework, the goal of an adversary is to move a model out of a rule's consistency class. We demonstrate with real and synthetic data examples how an adversary is able to move a model out of a rule's consistency class. We finish with a brief discussion on how regularization can effectively recover consistency, though we leave this for future work.

Notation See Table 1 for frequently used notation.

TABLE 1
Table of frequently used notation.

Notation	Description
$[k]$	The set of integers $\{1, 2, 3, \ldots, k\}$
$G = (V, E)$	A (random) graph with vertex set V and edge set E
$G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$	Two random graphs with a presumed shared set of vertices
C	A core set of vertices shared between two graphs
J_1, J_2	Junk vertices not shared between graphs
$\mathcal{G}_n$	The set of n-vertex labeled graphs
$F_{c,\theta}^{(n,m)}$	A nominatable distribution on $\mathcal{G}_n \times \mathcal{G}_m$ with c shared vertices and parameter θ
$\mathcal{N}_{n,m}$	The set of nominatable distributions on $\mathcal{G}_n \times \mathcal{G}_m$
g, g_1, g_2	Observed graphs
V^*	A vertex set of interest shared between two graphs
v^*	A single vertex of interest
$\mathfrak{o}$	An obfuscation function changing observed vertex labels
$\mathfrak{O}_W$	The set of obfuscating functions mapping a vertex set to W
$\mathcal{T}_W$	The set of total orderings of the elements of a set W
$\mathcal{I}(u; g)$	The set of vertices in g topologically equivalent to u
$\Phi(g_1, \mathfrak{o}(g_2), V^*)$	A vertex nomination scheme with vertex set of interest V^* and observed graphs g_1 and $\mathfrak{o}(g_2)$
$\mathfrak{r}_\Phi(g_1, g_2, \mathfrak{o}, V^*, S)$	The set of ranks of a set S under $\Phi(g_1, \mathfrak{o}(g_2), V^*)$

2. Vertex nomination and consistency

Before discussing how to define adversarial attacks, we discuss the previous work of [27], the first of its kind to derive the Bayes Optimal vertex nomination scheme for one vertex. This work can be viewed as a follow-on of that work, in which we provide a groundwork for the rigorous framing of an adversary in vertex nomination.

First, we will situate our analysis of the VN problem in the very general framework of nominatable distributions.

Definition 2 (Nominatable Distribution)**.** For a given $n, m \in \mathbb{Z} > 0$, the set of *Nominatable Distributions of order* (n, m), denoted $\mathcal{N}_{n,m}$, is the collection of

all families of distributions $\mathbf{F}_{\Theta}^{(n,m)}$ of the following form

$$\{F_{c,\theta}^{(n,m)} \text{ s.t. } 0 \leq c \leq \min(n,m) \in \mathbb{Z}, \theta \in \Theta \subset \mathbb{R}^{d(n,m)}\}$$

where $F_{c,\theta}^{(n,m)}$ is a distribution on $\mathcal{G}_n \times \mathcal{G}_m$ parameterized by $\theta \in \Theta$ satisfying:

1. The vertex sets $V_1 = \{v_1, v_2, ..., v_n\}$ and $V_2 = \{u_1, u_2, ..., u_m\}$ satisfy $v_i = u_i$ for $0 < i \leq c$. We refer to $C = \{v_1, v_2, ..., v_c\} = \{u_1, u_2, ..., u_c\}$ as the core vertices. These are the vertices that are shared across the two graphs and imbue the model with a natural notion of corresponding vertices.
2. Vertices in $J_1 = V_1 \setminus C$ and $J_2 = V_2 \setminus C$, satisfy $J_1 \cap J_2 = \emptyset$. We refer to J_1 and J_2 as junk vertices. These are the vertices in each graph that have no corresponding vertex in the other graph
3. The induced subgraphs $G_1[J_1]$ and $G_2[J_2]$ are conditionally independent given θ.

The vertices in C are those that have a corresponding paired vertex in each graph; where corresponding can be defined very generally. Corresponding vertices need not correspond to the same person/user/account, rather corresponding vertices are understood as those that share a desired property (for example, a role in the network) across graphs. In particular, we will assume that the vertices of interest in G_1 have corresponding vertices in G_2, and that these corresponding vertices are the vertices of interest in G_2.

Having access to the vertex labels would then render the VN problem trivial. To model the uncertainty often present in data applications, where the vertex labels (or correspondences) are unknown a priori we adopt the notion of *obfuscation functions* from [27].

Definition 3 (Obfuscating Function)**.** Let $(G_1, G_2) \sim F_{c,\theta}^{(n,m)} \in \mathcal{N}_{n,m}$, and let W be a set satisfying $W \cap V_i = \emptyset$ for $i = 1, 2$. An obfuscating function $\mathfrak{o} : V_2 \mapsto W$ is a bijection from V_2 to W. We refer to W as an obfuscating set, and we let $\mathfrak{O}_W$ be the set of all such obfuscation functions.

2.1. VN in the setting of a single vertex of interest

With these two definitions in place, we now present the definition of a vertex nomination scheme for a single vertex of interest as in [27]. In Section 2.2, we will extend the definition of a vertex nomination scheme to encompass multiple vertices of interest. In the remainder of this section, we will let $v^* \in V_1$ be the given vertex of interest in G_1.

Definition 4 (VN Scheme for single VOI)**.** Let $n, m \in \mathbb{Z} > 0$, and for each $g \in \mathcal{G}_m$, $u \in V(g)$, let

$$\mathcal{I}(u; g) = \{w \in V(g) \text{ s.t. } \exists \text{ an automorphism } \sigma \text{ of } g, \text{ s.t. } \sigma(u) = w\}.$$

Let W be an obfuscating set and $\mathfrak{o} \in \mathfrak{O}_W$ be given. For a set A, let $\mathcal{T}_A$ denote the set of all total orderings of the elements of A. A *vertex nomination scheme*

is a function $\Phi : \mathcal{G}_n \times \mathfrak{o}(\mathcal{G}_m) \times V_1 \to \mathcal{T}_W$ satisfying the following consistency property: If for each $u \in V_2$, we define $\text{rank}_{\Phi(g_1,\mathfrak{o}(g_2),v^*)}\big(\mathfrak{o}(u)\big)$ to be the position of $\mathfrak{o}(u)$ in the total ordering provided by $\Phi(g_1, \mathfrak{o}(g_2), v^*)$, and we define $\mathfrak{r}_\Phi : \mathcal{G}_n \times \mathcal{G}_m \times \mathfrak{O}_W \times V_1 \times 2^{V_2} \mapsto 2^{[m]}$ via

$$\mathfrak{r}_\Phi(g_1, g_2, \mathfrak{o}, v^*, S) = \{\text{rank}_{\Phi(g_1,\mathfrak{o}(g_2),v^*)}\big(\mathfrak{o}(u)\big) \text{ s.t. } u \in S\},$$

then we require that for any $g_1 \in \mathcal{G}_n$, $g_2 \in \mathcal{G}_m$, $v^* \subset V_1$, obfuscating functions $\mathfrak{o}_1, \mathfrak{o}_2 \in \mathfrak{O}_W$ and any $u \in V(g_2)$,

$$\begin{aligned} &\mathfrak{r}_\Phi(g_1, g_2, \mathfrak{o}_1, v^*, \mathcal{I}(u; g_2)) = \mathfrak{r}_\Phi(g_1, g_2, \mathfrak{o}_2, v^*, \mathcal{I}(u; g_2)) \qquad (1)\\ &\qquad\Longleftrightarrow\\ &\mathfrak{o}_2 \circ \mathfrak{o}_1^{-1}\big(\mathcal{I}(\Phi(g_1, \mathfrak{o}_1(g_2), V^*)[k]); \mathfrak{o}_1(g_2)\big) = \mathcal{I}\left(\Phi(g_1, \mathfrak{o}_2(g_2), v^*)[k]; \mathfrak{o}_2(g_2)\right)\\ &\qquad\qquad \text{for all } k \in [m], \end{aligned}$$

where $\Phi(g_1, \mathfrak{o}(g_2), v^*)[k]$ denotes the k-th element (i.e., the rank-k vertex) in the ordering $\Phi(g_1, \mathfrak{o}(g_2), v^*)$. We let $\mathcal{V}_{nm}$ denote the set of all such VN schemes.

Remark 5. The consistency criterion, Eq. (1), models the property that a sensibly-defined vertex nomination scheme should view all vertices in a given $\mathcal{I}_g(u)$ as being equally "interesting" in G_2. These vertices are topologically indistinguishable, and thus are only separated by their labels which have been obfuscated via $\mathfrak{o}$. Truly obfuscated vertex labels should be independent of the obfuscation function, and the consistency criterion requires that the set of ranks of each set of equivalent vertices (i.e., each $\mathcal{I}_{g_2}(u)$) does not depend on the particular choice of obfuscation function.

One can already begin to see how one might extend these definitions to multiple vertices of interest; note that Φ is a function of two graphs and a single vertex. It will be natural to require Φ to be a function of two graphs and a vertex *set* instead. We give these definitions in Section 2.2. We first define the error for the vertex nomination scheme defined above.

Definition 6 (VN loss function, level-k error for single VOI)**.** Let Φ be a vertex nomination scheme, and $\mathfrak{o}$ an obfuscating function. For (g_1, g_2) realized from $(G_1, G_2) \sim F_{c,n,m,\theta}$ with vertex of interest $v^* \in C$, and $k \in [m-1]$, we define the level-k nomination loss via

$$\begin{aligned} \ell_k\left(\Phi, g_1, g_2, v^*\right) &= \mathbb{1}\left\{\text{rank}_{\Phi(g_1,\mathfrak{o}(g_2),v^*)}\left(\mathfrak{o}\left(v^*\right)\right) \geq k+1\right\},\\ &= 1 - \mathbb{1}\left\{\text{rank}_{\Phi(g_1,\mathfrak{o}(g_2),v^*)}\left(\mathfrak{o}\left(v^*\right)\right) \leq k\right\}. \end{aligned}$$

The level k error of Φ at v^* is then defined to be

$$\begin{aligned} L_k\left(\Phi, v^*\right) &= \mathbb{E}_{(G_1,G_2)\sim F_{c,n,m,\theta}}\left[\ell_k\left(\Phi, G_1, G_2, v^*\right)\right]\\ &= \mathbb{P}_{(G_1,G_2)\sim F_{c,n,m,\theta}}\left[\text{rank}_{\Phi(G_1,\mathfrak{o}(G_2),v^*)}\left(\mathfrak{o}\left(v^*\right)\right) \geq k+1\right]. \end{aligned}$$

The level k error is simply the probability that the rank of the vertex of interest in g_2 is not in the nomination list; this matches our intuition for what the error should be. To discuss the notion of consistency, we need to assume that the core set C of the nominated are *nested* in the following sense.

Definition 7 (Nested Cores)**.** Let $\mathbf{F} = \left(F^{(n,m_n)}_{c_n,\theta_n}\right)_{n=n_0}^{\infty}$ be a sequence of distributions in $\mathcal{N}$. We say that $\mathbf{F}$ has *nested cores* if there exists an n_1 such that for all $n_1 \leq n < n'$, if $(G_1, G_2) \sim F^{(n,m_n)}_{c_n,\theta_n}$ and $(G_1', G_2') \sim F^{(n',m_{n'})}_{c_{n'},\theta_{n'}}$, we have, letting C and C' be the core vertices associated with $F^{(n,m_n)}_{c_n,\theta_n}$ and $F^{(n',m_{n'})}_{c_{n'},\theta_{n'}}$ respectively, and denoting the junk vertices J_1, J_1', J_2, J_2' analogously,

i. $V(G_1) = C \cup J_1 \subset V(G_1') = C' \cup J_1'$;
ii. $V(G_2) = C \cup J_2 \subset V(G_2') = C' \cup J_2'$;
iii. $C \subset C'$.

In [27], for any given nominatable distribution $F^{n,m}_{c,\theta}$, a Bayes optimal VN scheme is defined that is simultaneously optimal at all levels k. We will denote this optimal scheme via $\Phi^* = \Phi^*_{F^{n,m}_{c,\theta}}$, and its associated level k loss via L^*_k. The notion of consistency in VN is then defined as follows.

Definition 8 (Level k_n Consistent VN Rules in the single v.o.i. setting)**.** Let $\mathbf{F} = (F^{(n,m_n)}_{c_n,\theta_n})_{n=n_0}^{n=\infty}$ be a sequence of nominatable distributions in $\mathcal{N}$ with nested cores satisfying $\lim_{n\to\infty} m_n = \infty$. For a given non-decreasing sequence (k_n), we say that a VN rule $\mathbf{\Phi} = (\Phi_{n,m_n})_{n=n_0}^{n=\infty}$ is level-(k_n) consistent for vertex of interest $v^* \in C_1$ with respect to $\mathbf{F}$ if

$$\lim_{n\to\infty} L_{k_n}(\Phi_{n,m_n}, v^*) - L^*_{k_n}(v^*) = 0.$$

We say that a VN rule $\mathbf{\Phi}$ is *universally level-(k_n) consistent* if it is level-(k_n) consistent for all nested-core nominatable sequences $\mathbf{F}$. Before presenting vertex nomination schemes in the multiple v.o.i. setting, we first present an important consistency result given in [27], which says that there are **no** universally consistent vertex nomination schemes.

Theorem 9 (Corollary 28 of [27])**.** *Let $\varepsilon \in (0,1)$ be arbitrary, and consider a VN rule $\mathbf{\Phi} = (\Phi_{n,m})$. For any nondecreasing sequence $(k_n)_{n=n_0}^{\infty}$ satisfying $k_n = o(m)$, there exists a sequence of distributions $F_{c,n,m,\theta}$ in $\mathcal{N}$ with nested cores such that*

$$\limsup_{n\to\infty} L^*_{k_n}(v^*) = \epsilon < 1 = \lim_{n\to\infty} L_{k_n}(\Phi_{n,m}, v^*).$$

This result is markedly different from the setting of classical classification, in which there exist universally consistent classifiers. In Section 3, we will explore the ramifications of Theorem 9 on our understanding of adversarial attacks on VN rules; effectively such a result might mean that an adversary acts by moving a given distribution outside of the "consistency class" of a given nomination rule (see Section 2.3 for detail).

We next extend definitions to the more practical setting of multiple vertices of interest.

2.2. Extension to multiple vertices of interest

We will now rigorously define the VN problem and consistency within the VN framework for multiple vertices of interest. Combined with the results on consistency classes in Section 2.3, this will allow us to provide a statistical basis for understanding adversarial attacks in VN. Our definitions and notation are based on those in the previous section, though we have a few more general requirements. Recall that [27] defined a vertex nomination scheme as a function from $\Phi : \mathcal{G}_n \times \mathfrak{o}(\mathcal{G}_m) \times V_1 \to \mathcal{T}_W$ satisfying a certain consistency property. The extension to multiple vertices of interest requires that Φ be a function taking in a *set* of vertices. The rigorous definition is given below.

Definition 10 (VN Scheme)**.** Let $n, m \in \mathbb{Z} > 0$, and for each $g \in \mathcal{G}_m$, $u \in V(g)$, and again let

$$\mathcal{I}(u; g) = \{w \in V(g) \text{ s.t. } \exists \text{ an automorphism } \sigma \text{ of } g, \text{ s.t. } \sigma(u) = w\}.$$

Let W be an obfuscating set and $\mathfrak{o} \in \mathfrak{O}_W$ be given. For a set A, let $\mathcal{T}_A$ denote the set of all total orderings of the elements of A. A *vertex nomination scheme* is a function $\Phi : \mathcal{G}_n \times \mathfrak{o}(\mathcal{G}_m) \times 2^{V_1} \to \mathcal{T}_W$ satisfying the following consistency property: If for each $u \in V_2$, we define $\text{rank}_{\Phi(g_1, \mathfrak{o}(g_2), V^*)}\big(\mathfrak{o}(u)\big)$ to be the position of $\mathfrak{o}(u)$ in the total ordering provided by $\Phi(g_1, \mathfrak{o}(g_2), V^*)$, and we define $\mathfrak{r}_\Phi : \mathcal{G}_n \times \mathcal{G}_m \times \mathfrak{O}_W \times 2^{V_1} \times 2^{V_2} \mapsto 2^{[m]}$ via

$$\mathfrak{r}_\Phi(g_1, g_2, \mathfrak{o}, V^*, S) = \{\text{rank}_{\Phi(g_1, \mathfrak{o}(g_2), V^*)}\big(\mathfrak{o}(u)\big) \text{ s.t. } u \in S\},$$

then we require that for any $g_1 \in \mathcal{G}_n$, $g_2 \in \mathcal{G}_m$, $V^* \subset V_1$, obfuscating functions $\mathfrak{o}_1, \mathfrak{o}_2 \in \mathfrak{O}_W$ and any $u \in V(g_2)$,

$$\begin{aligned} \mathfrak{r}_\Phi(g_1, g_2, \mathfrak{o}_1, V^*, \mathcal{I}(u; g_2)) &= \mathfrak{r}_\Phi(g_1, g_2, \mathfrak{o}_2, V^*, \mathcal{I}(u; g_2)) \qquad (2) \\ &\Leftrightarrow \\ \mathfrak{o}_2 \circ \mathfrak{o}_1^{-1}\big(\mathcal{I}(\Phi(g_1, \mathfrak{o}_1(g_2), V^*)[k]); \mathfrak{o}_1(g_2)\big) &= \mathcal{I}\left(\Phi(g_1, \mathfrak{o}_2(g_2), V^*)[k]; \mathfrak{o}_2(g_2)\right) \\ &\qquad \text{for all } k \in [m], \end{aligned}$$

where $\Phi(g_1, \mathfrak{o}(g_2), V^*)[k]$ denotes the k-th element (i.e., the rank-k vertex) in the ordering $\Phi(g_1, \mathfrak{o}(g_2), V^*)$. We let $\mathcal{V}_{nm}$ denote the set of all such VN schemes.

A VN scheme is an information retrieval tool for efficiently querying large network data sets. Rather than naively searching G_2 for interesting vertices, an appropriate VN scheme provides a rank list of the vertices in G_2 that, ideally, allows users to identify v.o.i. in G_2 in a time-efficient manner. As such, to measure the performance of a VN scheme on multiple vertices, we will adopt a recall-at-k/precision-at-k framework. More precisely, we have the following definition.

Definition 11 (Level k Nomination Loss)**.** Let $\Phi \in \mathcal{V}_{n,m}$ be a vertex nomination scheme, W an obfuscating set, and $\mathfrak{o} \in \mathfrak{O}_W$. Let (g_1, g_2) be realized from

$(G_1, G_2) \sim F_{c\theta}^{(n,m)} \in \mathcal{N}_{n,m}$ with a vertex of interest set $V^* \subset C$. For $k \in [m-1]$, we define the level-k nomination losses via

$$\begin{aligned}\ell_k^{(1)}(\Phi, g_1, g_2, V^*) :&= \frac{\sum_{v \in V^*} \mathbb{1}\{\text{rank}_{\Phi(g_1, \mathfrak{o}(g_2), V^*)}(\mathfrak{o}(v)) \geq k+1\}}{|V^*|} \\ &= 1 - \frac{\sum_{v \in V^*} \mathbb{1}\{\text{rank}_{\Phi(g_1, \mathfrak{o}(g_2), V^*)}(\mathfrak{o}(v)) \leq k\}}{|V^*|} \\ \ell_k^{(2)}(\Phi, g_1, g_2, V^*) :&= 1 - \frac{\sum_{v \in V^*} \mathbb{1}\{\text{rank}_{\Phi(g_1, \mathfrak{o}(g_2), V^*)}(\mathfrak{o}(v)) \leq k\}}{|k|},\end{aligned}$$

where the (1) and (2) superscripts refer to recall and precision respectively. The error of a VN scheme is then defined as the expected loss. To wit, we have the following definition.

Definition 12 (Level-k Error)**.** Let $\Phi \in \mathcal{V}_{n,m}$ be a vertex nomination scheme, W an obfuscating set, and $\mathfrak{o} \in \mathfrak{O}_W$. The level-$k$ error of Φ for $V^* \subset C$ and $F_{c,\theta}^{(n,m)} \in \mathcal{N}$ is defined as

$$\begin{aligned}L_k^{(1)}(\Phi, V^*) :&= \mathbb{E}_{(G_1,G_2) \sim F_{c,\theta}^{(n,m)}}[\ell_k^{(1)}(\Phi, G_1, G_2, V^*)] \\ &= \frac{1}{|V^*|} \sum_{v \in V^*} \mathbb{P}_{F_{c,\theta}^{(n,m)}}\left(\text{rank}_{\Phi(G_1, \mathfrak{o}(G_2), V^*)}(\mathfrak{o}(v)) \geq k+1\right) \\ L_k^{(2)}(\Phi, V^*) :&= \mathbb{E}_{(G_1,G_2) \sim F_{c,\theta}^{(n,m)}}[\ell_k^{(2)}(\Phi, G_1, G_2, V^*)] \\ &= 1 - \frac{1}{|k|} \sum_{v \in V^*} \mathbb{P}_{F_{c,\theta}^{(n,m)}}\left(\text{rank}_{\Phi(G_1, \mathfrak{o}(G_2), V^*)}(\mathfrak{o}(v)) \leq k\right)\end{aligned}$$

The *level-k Bayes optimal scheme* is defined as any element

$$\Phi_{k,V^*}^* \in \text{argmin}_{\Phi \in \mathcal{V}_{nm}} L_k^{(1)}(\Phi, V^*) = \text{argmin}_{\Phi \in \mathcal{V}_{nm}} L_k^{(2)}(\Phi, V^*),$$

with corresponding errors $L_k^{*,(1)}$ and $L_k^{*,(2)}$.

In the almost sure absence of symmetries amongst the vertices in V^* (i.e., $\mathcal{I}(v, G_2) = \{v\}$ for all $v \in V^*$), the derivation of the Bayes optimal scheme in the present $|V^*| > 1$ setting mimics that of the $|V^*| = 1$ setting presented in [27].

2.2.1. Bayes optimal VN scheme construction

With notation as above, Let n, m be fixed and let $V^* \subset V_1 \cap V_2$ be fixed. Let W be an obfuscating set and $\mathfrak{o} \in \mathfrak{O}_W$. Further assume that $F = F_{c,\theta}^{(n,m)}$ is such that $\mathcal{I}(v, G_2) \overset{a.s.}{=} \{v\}$ for all $v \in V^*$, so that F is supported on

$$\mathcal{G}_{n,m}^a := \{(g_1, g_2) \in \mathcal{G}_n \times \mathcal{G}_m \text{ s.t. } \mathcal{I}(v; g_2) = \{v\} \text{ for all } v \in V^*\}.$$

For each $(g_1, g_2) \in \mathcal{G}_{n,m}^a$ define

$$(g_1, [\mathfrak{o}(g_2)]) = \left\{(g_1, \tilde{g}_2) \in \mathcal{G}_{n,m}^a : \mathfrak{o}(\tilde{g}_2) \simeq \mathfrak{o}(g_2)\right\}$$

$$= \left\{ (g_1, \tilde{g}_2) \in \mathcal{G}^a_{n,m} : \tilde{g}_2 \simeq g_2 \right\}.$$

where $\simeq$ denotes graph isomorphism. For each $w \in W$ and $u \in V_2$, we also define the following restriction

$$\begin{aligned}(g_1, [\mathfrak{o}(g_2)])_{w=\mathfrak{o}(u)} &= \Big\{ (g_1, \tilde{g}_2) \in \mathcal{G}^a_{n,m} \text{ s.t. } \mathfrak{o}(\tilde{g}_2) = \sigma(\mathfrak{o}(g_2)), \\ &\qquad\qquad \sigma \text{ an isomorphism, } \sigma(w) = \mathfrak{o}(u) \Big\} \\ &= \Big\{ (g_1, \tilde{g}_2) \in \mathcal{G}^a_{n,m} \text{ s.t. } \tilde{g}_2 = \sigma(g_2), \\ &\qquad\qquad \sigma \text{ an isomorphism, } \sigma(\mathfrak{o}^{-1}(w)) = u \Big\},\end{aligned}$$

and for $S \subset V_2$, define

$$(g_1, [\mathfrak{o}(g_2)])_{w \in \mathfrak{o}(S)} = \bigcup_{u \in S} (g_1, [\mathfrak{o}(g_2)])_{w=\mathfrak{o}(u)}.$$

Choose graphs

$$\mathbf{g} = \left\{ \left(g_1^{(i)}, g_2^{(i)} \right) \right\}_{i=1}^{h} \tag{3}$$

so that the sets

$$\left\{ \left(g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})] \right) \right\}_{i=1}^{h}$$

partition $\mathcal{G}^a_{n,m}$. To ease notation, we will denote this partition via $\mathcal{P}^{\mathbf{g}}_{n,m}$. We will next define a Bayes optimal scheme Φ^* (optimal under both loss functions simultaneously for all $k \in [m-1]$ for the above F supported on $\mathcal{G}^a_{n,m}$).

For ease of notation, for each $i \in [h]$ and $u \in W$, define

$$P^i_u := \mathbb{P}_{F^{(n,m)}_{c,\theta}} \left((g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})])_{u \in \mathfrak{o}(V^*)} \;\middle|\; (g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})]) \right)$$

Then, set (where ties are broken in a fixed but arbitrary manner)

$$\begin{aligned}\Phi^*(g_1^{(i)}, \mathfrak{o}(g_2^{(i)}), V^*)[1] &\in \operatorname*{arg\,max}_{u \in W} \; P^i_u \\ \Phi^*(g_1^{(i)}, \mathfrak{o}(g_2^{(i)}), V^*)[2] &\in \operatorname*{arg\,max}_{u \in W \setminus \{\Phi^*[1]\}} P^i_u \\ &\vdots \\ \Phi^*(g_1^{(i)}, \mathfrak{o}(g_2^{(i)}), V^*)[m] &\in \operatorname*{arg\,max}_{u \in W \setminus \{\cup_{j<m}\{\Phi^*[j]\}} P^i_u.\end{aligned}$$

For each element

$$(g_1, g_2) \in (g_1^{(i)}, [\mathfrak{o}(\tilde{g}_2^{(i)})]) \setminus \{(g_1^{(i)}, g_2^{(i)})\},$$

choose an isomorphism σ such that $\mathfrak{o}(g_2) = \sigma(\mathfrak{o}(g_2^{(i)}))$, and define

$$\Phi^*(g_1, \mathfrak{o}(g_2), V^*) = \sigma(\Phi^*(g_1^{(i)}, \mathfrak{o}(g_2^{(i)}), V^*)).$$

See Appendix A for a proof of the optimality of such a scheme.

Bayes optimal schemes when symmetries exist for the v.o.i.—i.e., when there are $v \in V^*$ such that $|\mathcal{I}(v, ; g_2)| > 1$—offer additional complications and, in the case when $|V^*| = 1$ done in [27], little additional insight. Precisely defining the Bayes optimal scheme in the case of symmetries when $|V^*| > 1$ is notationally and technically nontrivial, and is the subject of current research.

2.2.2. Consistency in VN with $|V^*| > 1$

Consistency in the VN framework for multiple vertices is then defined as follows.

Definition 13 (Level k_n Consistent VN Rules)**.** Let $\mathbf{F} = (F_{c_n,\theta_n}^{(n,m_n)})_{n=n_0}^{n=\infty}$ be a sequence of nominatable distributions in $\mathcal{N}$ with nested cores satisfying

$$\lim_{n\to\infty} m_n = \infty.$$

For a given non-decreasing sequence (k_n), we say that a VN rule $\mathbf{\Phi} = (\Phi_{n,m_n})_{n=n_0}^{n=\infty}$ is (where the level k_n-losses here are computed with respect to $F_n = F_{c_n,\theta_n}^{(n,m_n)}$)

i. level-(k_n) recall consistent for nested $V_n^* \in C_n$ with respect to $\mathbf{F}$ if

$$\lim_{n\to\infty} L_{k_n}^{(1)}(\Phi_{n,m_n}, V_n^*) - L_{k_n}^{*,(1)}(V_n^*) = 0,$$

for any sequence of obfuscating functions of V_2 with $|V_2| = m_n$. Note that the level k_n-loss here is computed with respect to $F_n = F_{c_n,\theta_n}^{(n,m_n)}$.

ii. level-(k_n) precision consistent for for nested $V_n^* \in C_n$ with respect to $\mathbf{F}$ if

$$\lim_{n\to\infty} L_{k_n}^{(2)}(\Phi_{n,m_n}, V_n^*) - L_{k_n}^{*,(2)}(V_n^*) = 0,$$

for any sequence of obfuscating functions of V_2 with $|V_2| = m_n$.

We say that a VN rule $\mathbf{\Phi}$ is *universally level-*(k_n) $\binom{precision}{recall}$ *consistent* if it is level-(k_n) $\binom{\text{precision}}{\text{recall}}$ consistent for all nested-core nominatable sequences $\mathbf{F}$. Theorem 9 in the previous section (Corollary 28 from [27]) proves that universally consistent VN schemes do not exist for any nondecreasing integral sequences (k_n) satisfying $k_n = o(m_n)$ and any (V_n^*) satisfying $|V_n^*| = \Theta(1)$. Beyond the ramifications for practically implementing VN in streaming or evolving network environments considered in [27], this lack of universal consistency is also the motivating result for our statistical approach to adversarial contamination in VN. Indeed, a simple consequence of the lack of universal consistency is that for any VN rule there are nominatable sequences for which the rule is not consistent. An adversary could then be understood as a probabilistic mechanism

designed to transform nominatable sequences for which the rule is consistent into nominatable sequences for which the rule is not consistent.

To develop this reasoning further, we next develop the notion of (maximal) consistency classes in the VN framework.

2.3. VN consistency classes

We next explore the concept of consistency classes in VN, with an eye towards the development of a statistical adversarial contamination framework for VN. First, let $\mathfrak{N}_{\mathbf{V}^*}$ be the collection of all nested-core nominatable sequences with nested v.o.i. $\mathbf{V}^* = (V_n^* \subset C_n)$. For a given VN rule $\mathbf{\Phi}$, v.o.i. sequence $\mathbf{V}^*$ satisfying $|V_n^*| = \Theta(1)$, and nondecreasing sequence (k_n) (satisfying the growth condition $k_n = o(n)$ of Theorem 15), the level-(k_n) $\binom{\text{precision}}{\text{recall}}$ consistency class of $\mathbf{\Phi}$ is defined to be

$$\mathfrak{C}_{\mathbf{\Phi}}^{(k_n)} = \left\{ \mathbf{F} \in \mathfrak{N}_{\mathbf{V}^*} \text{ s.t. } \mathbf{\Phi} \text{ is level-}(k_n) \begin{pmatrix} \text{precision} \\ \text{recall} \end{pmatrix} \text{consistent for } \mathbf{F} \right\}.$$

The lack of universal consistency ensures that $\mathfrak{C}_{\mathbf{\Phi}}^{(k_n)} \neq \mathfrak{N}_{\mathbf{V}^*}$ for any rule $\mathbf{\Phi}$.

It is natural to ask if there are a finite number of VN rules $\{\mathbf{\Phi}_i\}$ such that $\cup_i \mathfrak{C}_{\mathbf{\Phi}_i}^{(k_n)} = \mathfrak{N}_{\mathbf{V}^*}$. An affirmative answer would allow for ensemble methods to practically overcome the lack of universally consistent rules, and hence practically overcome any adversarial attack in the VN framework. We will see in Section 2.3.1 that the answer is, as expected, no, and any partition of $\mathfrak{N}_{\mathbf{V}^*}$ into maximal consistency classes necessarily contains infinite parts; see Theorem 15. As a consequence, ensemble methods cannot recover universal consistency in VN. The insights developed in Section 2.3.1 further motivate the development of adversarial contamination regimes for a given rule $\mathbf{\Phi}$. The idea behind adversarial contamination is simple in this framework: the adversary contaminates elements $\mathbf{F} \in \mathfrak{C}_{\mathbf{\Phi}}^{(\mathbf{k_n})}$ transforming them into $\mathbf{F}' \in \mathfrak{N}_{\mathbf{V}^*} \setminus \mathfrak{C}_{\mathbf{\Phi}}^{(\mathbf{k_n})}$.

2.3.1. Counting consistency classes

How can a practitioner mitigate the impact of a lack of universal consistency? One idea would be to consider ensemble methods, as the practical implications of the lack of universal consistency can be mitigated if universally consistent ensemble schemes exist. In this section, we will formalize the notion of maximal VN consistency classes and prove that infinitely many maximal consistency classes exist. We begin with defining the notion of maximal consistency classes in the VN-framework.

Definition 14 (Maximal Consistency Class)**.** As above, let $\mathfrak{N}_{\mathbf{V}^*}$ be the collection of all nested-core nominatable sequences with nested v.o.i. $\mathbf{V}^* = (V_n^* \subset C_n)$. For a nondecreasing integer sequence (k_n), we say that $\mathfrak{C} \in \mathfrak{N}_{\mathbf{V}^*}$ is a maximal level-(k_n) $\binom{\text{precision}}{\text{recall}}$ consistency class for $\mathbf{V}^*$ if the following two conditions hold.

i. There exists a VN rule Φ that is jointly level-(k_n) $\binom{\text{precision}}{\text{recall}}$ consistent for $\mathbf{V}^*$ for each $\mathbf{F} \in \mathfrak{C}$;
ii. If $\mathbf{F}' \notin \mathfrak{C}$, then there does not exist a VN rule Φ that is jointly level-(k_n) $\binom{\text{precision}}{\text{recall}}$ consistent for $\mathbf{V}^*$ for each $\mathbf{F} \in \mathfrak{C} \cup \{\mathbf{F}'\}$.

A natural question to ask is whether it is possible to partition $\mathfrak{N}_{\mathbf{V}^*}$ into a finite number of maximal level-(k_n) consistency classes for a particular sequence $(k_n)_{n=1}^{\infty}$? Our next result—Theorem 15—shows that for any integer sequence (k_n) satisfying a modest growth condition, any partition of $\mathfrak{N}$ into maximal level-(k_n) consistency classes must include at least countably infinite parts, thus erasing the hope that ensemble methods can recover universal consistency and practically mitigate the effect of any VN adversarial attack.

Theorem 15. *Let (k_n) be a sequence of nondecreasing integers satisfying $k_n = o(n)$, and let $\mathbf{V}^*$ be a nested sequence of vertices of interest satisfying $|V_n^*| = \Theta(1)$.*

i. Let $\mathfrak{N}_{\mathbf{V}^*} = \cup_{\alpha \in \mathcal{A}} \mathfrak{C}_\alpha$ be a partition of $\mathfrak{N}_{\mathbf{V}^*}$ into maximal level-(k_n) recall consistency classes, then $|\mathcal{A}| = \infty$.
ii. Let $\mathfrak{N}_{\mathbf{V}^*} = \cup_{\alpha \in \mathcal{A}} \mathfrak{C}_\alpha$ be a partition of $\mathfrak{N}_{\mathbf{V}^*}$ into maximal level-(k_n) precision consistency classes. If $k_n = \Theta(1)$, then $|\mathcal{A}| = \infty$.

The proof of this Theorem can be found in Appendix B.

2.3.2. Verification functions

In the presence of an adversarial attack, is it possible to, without additional supervision, verify if a given VN scheme is working on a given $F_{c,\theta}^{(n,m)} \in \mathcal{N}_{n,m}$? In other words, given a nondecreasing integer sequence (k_n), $(g_1, g_2) \in \mathcal{G}_n \times \mathcal{G}_m$, and v.o.i. V_n^*, can we consistently estimate the *verification function*

$$\begin{aligned} h_{\Phi_n}(g_1, \mathfrak{o}_n(g_2), V_n^*) &= h_{\Phi_n, k_n}(g_1, \mathfrak{o}_n(g_2), V^*) \\ &= \sum_{v \in V_n^*} \mathbb{1}\left\{\text{rank}_{\Phi_n(g_1, \mathfrak{o}_n(g_2), V^*)}(\mathfrak{o}_n(v)) \leq k_n\right\}? \end{aligned}$$

Note that the scaling by $|V_n^*|$ in the recall setting and by k_n in the precision setting do not affect consistent estimation of h if $|V_n^*| = \Theta(1)$ or if in the precision setting $k_n = \Theta(1)$. As such, the scaling is omitted.

The internal consistency criterion, Eq. (2) guarantees that

$$h_{\Phi_n}(g_1, \mathfrak{o}_n(g_2), V_n^*) = h_{\Phi_n}(g_1, \tilde{\mathfrak{o}}_n(g_2), V_n^*) \tag{4}$$

for all obfuscation functions $\mathfrak{o}_n, \tilde{\mathfrak{o}}_n \in \mathfrak{O}_n$. Indeed, the v.o.i.'s in g_2 are identical (though obfuscated differently) in $\mathfrak{o}_n(g_2)$ and $\tilde{\mathfrak{o}}_n(g_2)$. If we consider an alternate $(g_1', g_2') \sim F_n' \subset \mathbf{F}'$, it could be the case that $g_1 = g_1'$ and $g_2 \simeq g_2'$, while

$$h_{\Phi_n}(g_1, \mathfrak{o}_n(g_2), V_n^*) \neq h_{\Phi_n}(g_1, \mathfrak{o}_n(g_2'), V_n^*) \tag{5}$$

for all $\mathfrak{o}_n \in \mathfrak{O}_n$; indeed, consider letting the v.o.i.'s' in g_2' be different from (and not isomorphic to) those in g_2 (i.e., the behavior of the v.o.i. in F_n' is different from the behavior of the v.o.i. in F_n).

Consider the problem of estimating h_{Φ_n} via $\hat{h}_{\Phi_n}$. If the estimator is label-agnostic (i.e., there is no information in the obfuscated labeling of $\mathfrak{o}(g_2)$), then it is sensible to require that for all $g_2 \simeq g_2'$, we have that

$$\hat{h}_{\Phi_n}(g_1, \mathfrak{o}_n(g_2), V_n^*) = \hat{h}_{\Phi_n}(g_1, \mathfrak{o}_n(g_2'), V_n^*). \tag{6}$$

Contrasting this to Eqs. (4) and (5), we see that $(\hat{h}_{\Phi_n})$ cannot universally consistently estimate (h_{Φ_n}), as the sequence of estimators cannot account for the potentially different behaviors of the v.o.i.'s under the umbrella of nominatable distributions. To wit, we have the following lemma.

Lemma 16. *With notation as above, let $(\hat{h}_{\Phi_n})_n$ be any sequence of label-agnostic (i.e., satisfying Eq. (6)) estimators of $(h_{\Phi_n})_n$. There exists sequences of nested-core nominatable distributions $\mathbf{F} = (F_n)$ and $\mathbf{F}' = (F_n')$ such that for n sufficiently large, if $(G_1, G_2) \sim F_n$, and $(G_1', G_2') \sim F_n'$, then*

$$d_{TV}\left(\mathcal{L}(h_{\phi_n}(G_1, \mathfrak{o}(G_2), V_n^*)), \mathcal{L}(h_{\phi_n}(G_1', \mathfrak{o}(G_2'), V_n^*))\right) > 0,$$

while $\mathcal{L}(\hat{h}_{\phi_n}(G_1, \mathfrak{o}(G_2), V_n^*)) = \mathcal{L}(\hat{h}_{\phi_n}(G_1', \mathfrak{o}(G_2'), V_n^*))$ (where d_{TV} is the total variation distance).

As a result of the above discussion and Lemma, we are unable to verify, without additional supervision, if an adversary has moved the distribution out of a given VN rule's consistency class. This points to the primacy of additional supervision, which in the VN framework often comes in the form of a user-in-the-loop. Indeed, we are currently exploring the role/impact a use-in-the-loop in VN—where the user can evaluate the interestingness of the vertices in the top k of the nomination list for a cost c_k. This supervision can also be thought of as a form of regularization, designed to increase the consistency class of a given VN rule.

3. Adversarial vertex nomination

In order to actively model adversarial attacks in the VN-framework, we formalize the notion of an *edge adversary*.

Definition 17 (Adversary)**.** Let F be a distribution on graphs in $\mathcal{G}_m$, and let U be a random variable independent of $G \sim F$. We say $\mathcal{A} = \{f_{\mathcal{A}}, V_{\mathcal{A}}, U, \theta\}$ is an *adversary* parameterized by $\theta \in \Theta$ if

1. $f_{\mathcal{A}} : \mathcal{G}_m \times \mathbb{R} \times \Theta \mapsto \mathcal{G}_m$ is a measurable function such that $V(f_{\mathcal{A}}(G, U, \theta)) = V(G)$, so that $f_{\mathcal{A}}(G, U, \theta)$ is a $\mathcal{G}_m$-valued random variable.
2. $V_{\mathcal{A}} : \mathcal{G}_m \times \mathbb{R} \times \Theta \mapsto 2^{[m]}$ is a measurable function that satisfies $V_{\mathcal{A}}(G, U, \theta) \subset V(G)$, so that $V_{\mathcal{A}}(G, U, \theta)$ is a (potentially) random subset of $V(G)$.

3. If $L = \left\{ v, w \in V(G) \text{ s.t. } (v,w) \in E(f_{\mathcal{A}}(G,U,\theta)) \,\Delta\, E(G) \right\}$, (where Δ represents the symmetric difference) then $L \subset V_{\mathcal{A}}(G,U,\theta)$. Succinctly put, if an edge is added or removed from $E(G)$, then the vertices adjacent to that edge must be in $V_{\mathcal{A}}(G,U,\theta)$.

In the above, U represents an independent source of randomness utilized in the adversarial attack.

Note that $f_{\mathcal{A}}$ is simply a function that adds/deletes edges from a network potentially randomly, and these edges must be incident to the vertices of $V_{\mathcal{A}}$. To that end, we will refer to $V_{\mathcal{A}}$ as the vertices *contaminated* by $\mathcal{A}$.

If we are given a sequence of nominatable distributions $\mathbf{F} = (F_n)_{n=n_0}^{\infty}$, where F_n is a distribution on $\mathcal{G}_n \times \mathcal{G}_m$, then we will let $f_{\mathcal{A}_n}(F_n)$ denote a sequence of graphs realized from F_n, with the second graph G_2 contaminated by $f_{\mathcal{A}_n}$; we call a sequence $(f_{\mathcal{A}_n})_{n=n_0}^{\infty}$ an adversary rule. In the language of VN consistency classes, we posit that an adversary rule aims to contaminate a VN rule $\mathbf{\Phi}$ via

$$\mathbf{F} = (F_n)_{n=n_0}^{\infty} \in \mathfrak{C}_{\mathbf{\Phi}}^{(k_n)} \implies (f_{\mathcal{A}_n}(F_n))_{n=n_0}^{\infty} \in \mathfrak{N}_{\mathbf{V}^*} \setminus \mathfrak{C}_{\mathbf{\Phi}}^{(k_n)}.$$

Remark 18. Let $G_2 = (V_2, E_2)$ and $G_2' = (V_2', E_2')$. Consider an edge adversary $f_{\mathcal{A}}$ acting on G_2'. By considering $V_2 = V(G_2') \setminus V_{\mathcal{A}}$, we can also consider this adversary as a *vertex adversary* that randomly adds vertices to G_2. Vertex addition and deletion can be simultaneously modeled by first considering a mechanism for randomly deleting vertices from $G_2 = (V_2, E_2)$ before using the above approach to add adversarial vertices to the network.

Remark 19. In [50], the authors consider *direct attacks* and *influencer attacks* in which, given a vertex of interest v^*, either $v^* \in V_{\mathcal{A}}$ or $v^* \notin V_{\mathcal{A}}$ respectively. However, note that in [50], the objective is vertex classification, whereas we are not directly classifying vertices. Rather, we are interested in ranking vertices in G_2 by interestingness given limited training data in G_1. We will typically assume that $v^* \notin V_{\mathcal{A}}$ (i.e. the adversary does not control the vertex of interest), so that we are examining *influencer attacks*.

3.1. A simple VN adversarial contamination model

Now that we have developed the requisite setting for framing the idea of adversarial contamination in the VN-setting, we will consider a simple model for adversarial contamination in the stochastic blockmodel (SBM) of [23].

Definition 20 (Stochastic Blockmodel)**.** We say that an n-vertex random graph G is an instantiation of a stochastic blockmodel with parameters (n, K, B, b) (written $A \sim \text{SBM}(n, K, B, \pi)$) if

i. The block membership vector $\pi \in \mathbb{R}^K$ satisfies $\pi_i \geq 0$ for all $i \in [K]$, and $\sum_i \pi(i) = 1$;
ii. The vertex set $V = V(G)$ is the disjoint union of K blocks $V = B_1 \sqcup B_2 \sqcup \cdots \sqcup B_K$, where each vertex $v \in V$ is independently assigned to a block

according to a Multinomial$(1, \pi)$ distribution. If vertex v is assigned to block $i \in [K]$, then the block membership function $b : V \mapsto [K]$ satisfies $b(v) = i$;

iii. The block probability matrix $B \in [0,1]^{K \times K}$ is such that, for each pair of vertices $\{u, v\} \in \binom{V}{2}$, $\mathbb{1}_{u \sim_G v} \sim \text{Bernoulli}(B_{b(u),b(v)})$, and the collection of indicator random variables $\{\mathbb{1}_{u \sim_G v}\}_{\{u,v\} \in \binom{V}{2}}$ is mutually independent given b (here $\{u \sim_G v\} \Leftrightarrow \{\{u, v\} \in E\}$).

In addition, we will say that a pair of graphs (G_1, G_2) is an instantiation of a ρ-correlated SBM(n, K, B, b) (written $(G_1, G_2) \sim \text{SBM}(\rho, n, K, B, \pi)$) if marginally $G_1 \sim \text{SBM}(n, K, B, b)$ and $G_2 \sim \text{SBM}(n, K, B, b)$, and the collection of indicator random variables

$$\left\{ \{\mathbb{1}_{u \sim_{G_1} v}\}_{\{u,v\} \in \binom{V}{2}} \bigcup \{\mathbb{1}_{u \sim_{G_2} v}\}_{\{u,v\} \in \binom{V}{2}} \right\}$$

is mutually independent except that for each $\{u, v\} \in \binom{V}{2}$,

$$\text{Correlation}(\mathbb{1}_{u \sim_{G_1} v}, \mathbb{1}_{u \sim_{G_2} v}) = \rho.$$

Consider G as an n-vertex stochastic blockmodel, with two blocks, B_1 and B_2, and with $\pi = (1/2, 1/2)$. The block-probability matrix B is given by

$$B = \begin{pmatrix} p & r \\ r & q \end{pmatrix}, \tag{7}$$

with $p \geq q \geq r > 0$. Given $G = g$, we define the following VN adversarial contamination procedure $\mathcal{A} = (f_{\mathcal{A}}, V_{\mathcal{A}}, U, \theta)$ acting on g as follows:

1. $\theta = (c_+, c_-, \pi_+, \pi_-, s_+, s_-)$ is a vector of parameters where $c_+, c_- \in \mathbb{Z}$ satisfy $c_+ + c_- \leq n$, π_+, $\pi_- \in (0, 1)$, and s_+, $s_- \in [0, 1]$;
2. U is a uniformly distributed random variable independent of G;
3. $f_{\mathcal{A}}(g, U, \theta) \in \mathcal{G}_n$ is defined as follows:
   i. Initialize $g_c = g$
   ii. Create a set of vertices W_+ by independently selecting each vertex in $V = [n]$ to be in W_+ with probability π_+. Then, create a set of vertices W_- by independently selecting each vertex in $V \setminus W_+ = [n]$ to be in W_- with probability π_-.
   iii. For each vertex pair $\{v, u\}$ such that (v, u) or $(u, v) \in W_+ \times (V \setminus W_-)$,
      i. If $\{v, u\} \in E(g_c)$, nothing happens.
      ii. If $\{v, u\} \notin E(g_c)$, an edge is independently added connecting $\{v, u\}$ in g_c with probability s_+.
   iv. For each vertex pair $\{v, u\}$ such that (v, u) or $(u, v) \in W_- \times (V \setminus W_+)$,
      i. If $\{v, u\} \notin E(g_c)$, nothing happens.
      ii. If $\{v, u\} \in E(g_c)$, the edge is independently deleted from g_c with probability s_-.
   v. Set $f_{\mathcal{A}}(g, U, \theta) = g_c \in \mathcal{G}_n$.

The auxiliary randomness U in $\mathcal{A}$ is utilized to make the random vertex selections in ii., the random edge additions in iii., and the random edge deletions in iv.

Notice that this adversarial model gives rise to a new stochastic blockmodel with the edge-probability matrix $\tilde{B}$ given by

$$\tilde{B} = \begin{array}{c} \\ \tilde{B}_1 \\ \tilde{B}_1^+ \\ \tilde{B}_1^- \\ \tilde{B}_2 \\ \tilde{B}_2^+ \\ \tilde{B}_2^- \end{array} \begin{array}{c} \begin{array}{cccccc} \tilde{B}_1 & \tilde{B}_1^+ & \tilde{B}_1^- & \tilde{B}_2 & \tilde{B}_2^+ & \tilde{B}_2^- \end{array} \\ \begin{pmatrix} \mathbf{p} & x_1 & x_2 & \mathbf{r} & x_3 & x_4 \\ x_1 & x_1 & p & x_3 & x_3 & r \\ x_2 & p & x_2 & x_4 & r & x_4 \\ \mathbf{r} & x_3 & x_4 & \mathbf{q} & x_5 & x_6 \\ x_3 & x_3 & r & x_5 & x_5 & q \\ x_4 & r & x_4 & x_6 & q & x_6 \end{pmatrix} \end{array}$$

where

$$x_1 = p + s_+(1-p), \quad x_2 = p(1-s_-), \quad x_3 = r + s_+(1-r),$$
$$x_4 = (1-s_-)r, \quad x_5 = q + s_+(1-q), \quad x_6 = q(1-s_-),$$

and where $\tilde{B}_1^+$ are the vertices in $W_+ \cap B_1$; $\tilde{B}_1^-$ are the vertices in $B_1 \cap W_-$; and $\tilde{B}_1$ are the vertices in $B_1 \setminus (\tilde{B}_1^+ \cup \tilde{B}_1^-)$; with $\tilde{B}_2$ defined analogously. We note here that this adversarial contamination model is similar to the contamination model considered in [8].

Note also that the original block structure is preserved amongst vertices in $\tilde{B}_1 \cup \tilde{B}_2$, and we can view this contamination model as adding vertices randomly to $G[\tilde{B}_1 \cup \tilde{B}_2]$, i.e., the induced subgraph on $\tilde{B}_1 \cup \tilde{B}_2$. When $(G_1, G_2) \sim \mathrm{SBM}(\rho, n, K, B, \pi)$ and this adversarial procedure is applied to G_2, we will denote

$$G_1^{(i)} = G_1[\tilde{B}_1 \cup \tilde{B}_2] \tag{8}$$
$$G_2^{(i)} = G_2[\tilde{B}_1 \cup \tilde{B}_2] \tag{9}$$

Remark 21. Let $\mathcal{A}_n$ be the simple adversarial rule outlined above. A very simple VN rule $\mathbf{\Phi}$ and nested core nominatable sequence $\mathbf{F}$ for which

$$\mathbf{F} = (F_n)_{n=n_0}^{\infty} \in \mathfrak{C}_{\mathbf{\Phi}}^{(k_n)} \implies (f_{\mathcal{A}_n}(F_n))_{n=n_0}^{\infty} \in \mathfrak{N} \setminus \mathfrak{C}_{\mathbf{\Phi}}^{(k_n)},$$

proceeds as follows. Consider $F_n = \mathrm{SBM}(\rho, n, K, B, \pi)$ supported on $\mathcal{G}_n \times \mathcal{G}_n$ where B is as in Eq. (7) with $\pi = (1/2, 1/2)$, $p > q > r$ fixed, and $\rho > 0$ fixed. Suppose that Φ_n is a VN scheme that runs spectral clustering on the contaminated graph by first selecting the number of communities in a consistent manner (via adjacency spectral clustering for example [28]) and ranking all the vertices in the group with the highest probability of within-group connection (in a fixed but arbitrary order), and then ranks the rest of the vertices in fixed but arbitrary order. Suppose that we consider $k_n = n/2$. It is immediate that $\mathbf{F} = (F_n)_{n=n_0}^{\infty} \in \mathfrak{C}_{\mathbf{\Phi}}^{(k_n)}$ and that the adversary acting on G_2 impacts this consistency. We present the following result as a lemma, but the proof is a simple calculation.

Lemma 22. *In the adversarial contamination model $\mathcal{A}_n$ defined above, if either*

1. $p - q < s_-$, *or*
2. $\frac{p-q}{1-q} < s_+$,

then Φ_n is no longer consistent with respect to the adversarially contaminated model sequence.

3.2. Regularizing the adversary

Given the adversarial model considered above, and the discussion on VN verification in Section 2.3.2, it is natural to seek procedures for mitigating the effect of the contamination in G_2. Network regularization is a natural solution, and we here consider as a regularization strategy the network analogue of the classical trimmed mean estimator. To wit, we consider the regularization procedure in Algorithm 1 inspired by the network trimming procedure in [16]; see also the work in [25] for the impact of trimming regularization on random graph concentration.

Algorithm 1 Regularization via network trimming.

Input: Graph G, $\ell, h \in (0,1)$, seed set S;

1. Initialize $V_t = S$
2. Rank the vertices in $V(G) \setminus S$ by ascending degree (ties are broken via averaging over ranks). For each vertex u in $V(G) \setminus S$, denote the rank via $rk(u)$;
for $u \in V(G) \setminus S$, **do**
  3. If $\ell < \frac{rk(u)}{|V(G)\setminus S|} \leq 1 - h$, add u to V_t;
end for
4. Output: $G^{(\ell,h)} = G[V_t]$, the induced subgraph of G on V_t;

Remark 23. The parameters ℓ and h appearing in Algorithm 1 are unknown a priori, and to data-adaptively choose ℓ and h, we sweep over possible values and choose the values of ℓ and h that leads to the maximum network modularity in $G_2^{(\ell,h)}$ when clustering the vertices of $G_2^{(\ell,h)}$ via $GMM \circ ASE$ clustering; i.e., embed $G_2^{(\ell,h)}$ using ASE and cluster the embedding using a model-based GMM procedure. Given a clustering C, the modularity is defined as usual via

$$Q(C) = \frac{1}{(2|E|)} \sum_{i,j} \left[A_{i,j} - \frac{d_i d_j}{2|E|} \right] \mathbb{1}\{C_i = C_j\},$$

where $|E|$ = the number of edges in $G_2^{(\ell,h)}$; $A_{i,j}$ is the i,j-th element of the adjacency matrix A of $G_2^{(\ell,h)}$; d_i is the degree of vertex i in $G_2^{(\ell,h)}$; and C_i is the cluster containing vertex i in C.

3.2.1. Regularization in our motivating example from Section 1.1

We next explore the impact of regularization on our motivating HS social network example from Section 1.1. In the left panel of Figure 2, we plot the

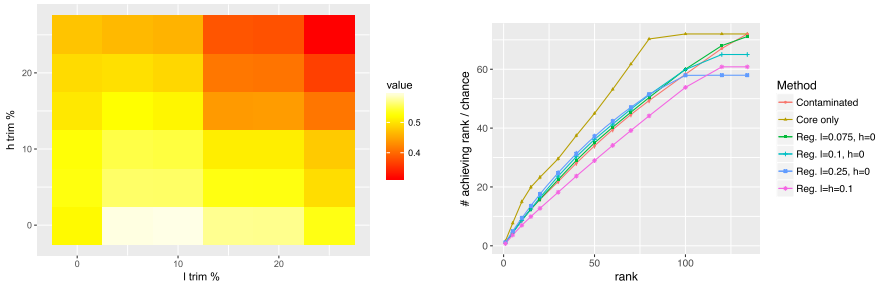


FIG 2. *The left panel shows the modularity of the GMM clustering as a function of the regularization parameters. The color indicates the value of the modularity, with darker red indicating lower values and lighter yellow–to–white indicating larger values. The right panel shows the performance of VN∘GMM∘ASE with higher values indicating a greater percentage of true vertices nominated. In both figures, we average over the same 500 seed sets of size* $s = 10$. *See Section 3.2.1 for details.*

TABLE 2
Mean number of v.o.i. achieving rank $\leq x$ *for the various regularization and contamination settings considered. Results are averaged over 500 MC trials.*

	Mean number of v.o.i. achieving rank $\leq$ x					
	$x = 1$	$x = 5$	$x = 10$	$x = 15$	$x = 20$	$x = 30$
Core only	2.800	12.624	21.882	27.448	31.882	38.884
Contaminated	2.074	10.006	17.346	21.428	24.656	29.598
Reg. $\ell = 0.075, h = 0$	1.920	9.500	16.198	20.536	24.372	31.266
Reg. $\ell = 0.1, h = 0$	1.580	8.296	14.216	19.918	25.004	34.434
Reg. $\ell = 0.25, h = 0$	1.572	8.274	15.146	21.136	26.792	36.630
Reg. $\ell = 0.1, h = 0.1$	1.970	8.756	13.678	17.284	20.470	26.574

modularity of the GMM clustering in the trimmed $G_2^{(\ell,h)}$ as a function of $\ell, h \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$. Note that we average the modularity values over $nMC = 500$ seed sets of size $s = 10$ (the same seed sets as used in Figure 1). The color indicates the value of the modularity, with darker red indicating lower values and lighter yellow–to–white indicating larger values. From the figure, we can see that modularity is maximized when $h = 0$ (i.e., no large degree vertices trimmed) and $\ell \approx 0.05$–0.1. We note that this trimming process can cut core vertices as well as junk vertices, and core vertices cut from G_2 can never be recovered via VN ∘ GMM ∘ ASE. This is demonstrated in the right panel of Figure 2, where the horizontal asymptotes for each trimming value indicates the maximum number of core vertices that are recoverable after regularization. In the figure, the gold line represents performance in the idealized network pair $(G_1^{(i)}, G_2^{(i)})$; the red line for the contaminated $(G_1^{(i)}, G_2)$; the green, teal, blue and pink lines respectively present performance for $(G_1, G_2^{(\ell,h)})$ (i.e., after regularizing) with $(\ell, h) = (0.075, 0)$, $(0.1, 0)$, $(0.25, 0)$, $(0.1, 0.1)$ respectively.

In Figure 3 and Table 2, we see the effect of regularization play out in more detail. Indeed, mean VN ∘ GMM ∘ ASE performance in the regularized setting

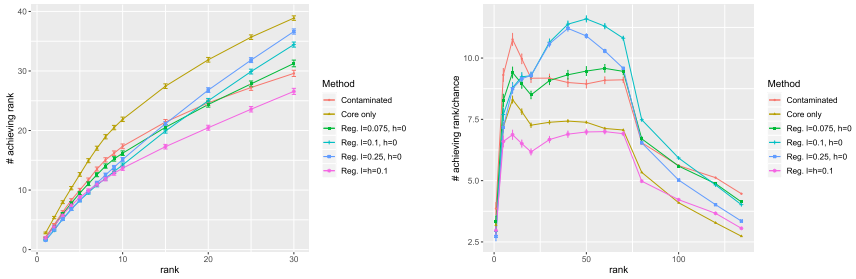


FIG 3. *We plot the performance of VN ∘ GMM ∘ ASE averaged over* $nMC = 500$ *random seed sets of size* $s = 10$. *The left figure shows the number of true vertices achieving the rank, and the right figure shows the same result normalized by chance performance. The gold line represents the original, uncontaminated netowrk, the red represents the contaminated network, and the other colors represent differerent levels of regularization. See Section 3.2.1 for details.*

increases versus in the contaminated setting for

$$(\ell, h) = \{(0.075, 0), (0.1, 0), (0.25, 0)\},$$

whereas mean regularized performance decreases for $(\ell, h) = \{(0.1, 0.1)\}$. While over-regularizing can adversely affect performance, this data-adaptive regularization — while not fully recovering the performance of the idealized setting— nonetheless effectively mitigates the impact of the contamination on our VN ∘ GMM ∘ ASE algorithm in this dataset.

4. Experiments

We next explore the effect of our adversarial noise model in a simulated data experiment, and the effect of adversarial contamination (and a subsequent model for regularization) in a real data example derived from Bing entity transition graphs. First, we explain in detail the steps of the VN scheme we will consider in our experiments.

4.1. Experimental setup

In the contamination model of Section 3.1, we consider the following VN scheme, denoted VN ∘ GMM ∘ ASE. Letting $v^* \in V(G_1)$ (resp., $V^* \subset V(G_1)$) be the vertex (resp., vertices) of interest in G_1, we seek the corresponding vertex (resp., vertices) of interest in $V(G_2)$ as follows:

1. Given two graphs, G_1 and G_2, we use Adjacency Spectral Embedding (ASE) [43] to separately embed G_1 and G_2 into a common Euclidean space $\mathbb{R}^d$. Given the $n \times n$ adjacency matrix A of G_1, the d-dimensional ASE of G_1 is defined as follows.

Definition 24 (Adjacency spectral embedding (ASE))**.** Given $d \in \mathbb{Z} > 0$, the *adjacency spectral embedding* (ASE) of A into $\mathbb{R}^d$ is defined via $\widehat{X} = U_A S_A^{1/2}$ where

$$|A| = [U_A|U_A^{\perp}][S_A \oplus S_A^{\perp}][U_A|U_A^{\perp}]$$

is the spectral decomposition of $|A| = (A^T A)^{1/2}$, $S_A \in \mathbb{R}^{d\times d}$ is the diagonal matrix with the d largest eigenvalues of $|A|$ on its diagonal and $U_A \in \mathbb{R}^{n\times d}$ has columns which are the eigenvectors corresponding to the eigenvalues of S_A.

Simply stated, the ASE of a graph G provides Euclidean features for each vertex in G on which to perform subsequent inference. Combined with recent efforts to prove that the ASE provides consistent estimators of the latent position parameters in random dot product graphs and positive-definite stochastic blockmodels [43, 2], the ASE allows for a host classical inference methodologies to be successfully employed within these random graph frameworks [44, 45, 29]. To choose d above, we use the machinery of [49, 10] to develop the principled heuristic of estimating d as the larger of the two elbows of the associated scree plots of the singular values of G_1 and G_2.

2. Solve the orthogonal Procrustes problem [40] to find an orthogonal transformation aligning the seeded vertices across graphs. Let $\widehat{X}_S$ (resp., $\widehat{Y}_S$) be the matrix composed of the rows of ASE(G_1) (resp., ASE(G_2)) corresponding to the seeded vertices in S. Letting the SVD of $\widehat{Y}_S^T \widehat{X}_S = U\Sigma V^T$, the solution to

$$R = \text{argmin}_{O \text{ s.t. } O^T O = I} \|\widehat{X}_S - \widehat{Y}_S O\|_F,$$

is given by $R = UV^T$. Use this transformation to align the embeddings of G_1 and G_2 in $\mathbb{R}^d$, i.e., rotate $\widehat{Y}$ via $\widehat{Y}O$ to align $\widehat{Y}$ to $\widehat{X}$.

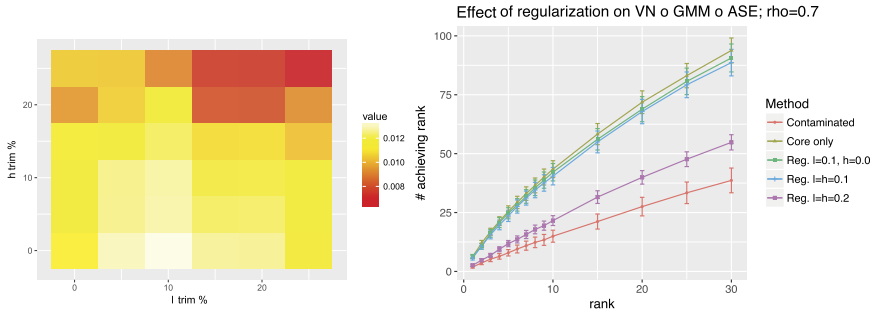
3. Motivated by the central limit theorem of [3] for the residual errors between the rows of the ASE and the latent position parameters in random dot product graphs, we use model-based Gaussian mixture modeling (GMM) to simultaneously cluster the vertices of the embedded graphs. Here, we employ the `R` package `MClust` [19].

4. Rank the candidate matches in G_2 according to the following heuristic. If $u \in V(G_1)$ and $v \in V(G_2)$ are clustered points in the Procrustes-aligned embedding of G_1 and G_2 with respective covariance matrices Σ_u and Σ_v in their components of the GMM, then compute

$$\Delta(u, v) = \max\left(D_u(u, v), D_v(u, v)\right),$$

where

$$D_u(u, v) = \sqrt{(u - v)\Sigma_u^{-1}(u - v)^T}$$



(a) Modularity plot versus (ℓ, h) (b) Regularization in VN ∘ GMM ∘ ASE

FIG 4. *In the left panel, we plot the modularity of the GMM clustering in the trimmed graph as a function of the regularization parameters. The color indicates the value of the modularity, with darker red indicating lower values and lighter yellow–white indicating larger values. In the right panel, we plot the performance of VN ∘ GMM ∘ ASE for a stochastic blockmodel in our contamination model in Section 3.1. Gold represents the uncontaminated network, red represents the contaminated network, and the other colors represent various levels of reuglarization. In both figures, we average over the same 50 seed sets of size* $s = 10$. *See Section 4.2 for details.*

and

$$D_v(u, v) = \sqrt{(u - v)\Sigma_v^{-1}(u - v)^T}$$

are the respective Mahalanobis distances from u to v. In the case of a single v.o.i. v^*, rank the vertices in G_2 then by increasing value of $\Delta(v^*, u)$, i.e., with ties broken in a fixed deterministic fashion, we rank via (where $n_2 = |V(G_2)|$)

$$\begin{aligned}
\Phi_n(g_1, g_2, v^*)[1] &\in \operatorname*{arg\,min}_{u \in V(G_2)} \Delta(v^*, u) \\
\Phi_n(g_1, g_2, v^*)[2] &\in \operatorname*{arg\,min}_{u \in V(G_2) \setminus \{\Phi_n[1]\}} \Delta(v^*, u) \\
&\vdots \\
\Phi_n(g_1, g_2, v^*)[n_2 - 1] &\in \operatorname*{arg\,min}_{u \in V(G_2) \setminus \{\cup_{(j \le n_2 - 2)} \Phi_n[j]\}} \Delta(v^*, u) \\
\Phi_n(g_1, g_2, v^*)[n_2] &\in \operatorname*{arg\,min}_{u \in C_{v^*} \setminus \{\cup_{(j \le n_2 - 1)} \Phi_n[j]\}} \Delta(v^*, u).
\end{aligned}$$

In the case of multiple v.o.i. V^*, rank the vertices in G_2 then by increasing value of $\min_{v \in V^*} \Delta(v, u)$ with ties broken in a fixed deterministic fashion. We choose $\min_{v \in V^*} \Delta(v, u)$ as our ranking metric here as what defines interestingness can vary even among the v.o.i. in G_1; i.e., $\max_{v, v' \in V^*} \Delta(v, v')$ may be relatively large. Being uniformly close to the collection of v.o.i. would be too stringent a condition then, and we merely require highly nominated vertices to have close proximity to a v.o.i., as this would be evidence the highly nominated vertices correspond in G_2 to these proximal v.o.i. in G_1.

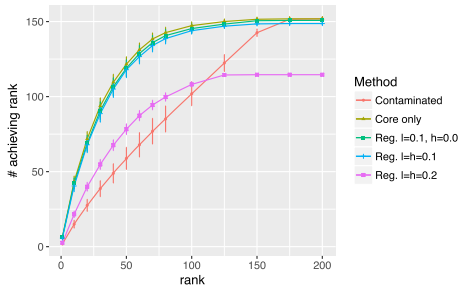


FIG 5. *We plot the performance of VN ∘ GMM ∘ ASE (±2s.e.) in* $(G_1, G_2) \sim SBM(0.3, 200, 2, B, \pi = (1/2, 1/2))$ *with* $\rho = 0.7$ *again averaged over* $nMC = 50$ *random seed sets of size* $s = 10$. *The x-axis shows the ranks in the nomination list and the y-axis shows (on average) how many vertices* $v \in G_1^{(i)}$, *when viewed as the v.o.i., had their corresponding vertex of interest ranked in the top x by VN ∘ GMM ∘ ASE. The gold line represents performance in the idealized network pair* $(G_1^{(i)}, G_2^{(i)})$; *the red line for* $(G_1^{(i)}, G_2^{(c)})$; *the green, blue, and pink lines for* $(G_1^{(i)}, G_2^{(\ell,h)})$ *for varying values of* (ℓ, h).

4.2. Simulation

We consider the model in Section 3.1 with the following parameter choices:

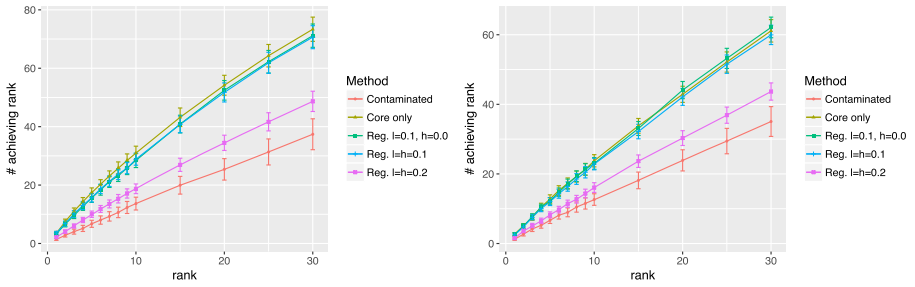
$$\begin{array}{ccc} n = 200; & \pi = (1/2, 1/2); & \pi_- = 0.1,\ \pi_+ = 0.1; \\ p = 0.4; & q = 0.5; & r = 0.3; \\ s_+ = 0.8; & s_- = 0.8; & \rho \in (0.3, 0.5, 0.7). \end{array}$$

Note that these parameter choices yield an illustrative simulation, and we find that the resulting findings hold across multiple parameter choices as well. Note that, in the notation of Section 3.1, if $(G_1, G_2) \sim \text{SBM}(\rho, n, K, B, \pi)$, we will consider

$$\begin{aligned} G_1^{(i)} &= G_1[\tilde{B}_1 \cup \tilde{B}_2] \\ G_2^{(i)} &= G_2[\tilde{B}_1 \cup \tilde{B}_2] \\ G_2^{(c)} &= G_2 \text{ acted upon by the adversary described} \\ &\qquad\qquad \text{in Section 3.1;} \\ G_2^{(\ell,h)} &= G_2^{(c)} \text{ trimmed as in Algorithm 1.} \end{aligned}$$

In this simulation example, we observe that the adversarial contamination model significantly decreases VN performance and that the trimming regularization mitigates this contamination and recovers much of the lost inferential performance.

In Figure 4 we plot the performance of VN ∘ GMM ∘ ASE over a number of (ℓ, h) trimming pairs (we note that for all correlation/regularized/contaminated/trimmed combinations, mean performance is significantly better than chance and chance normalized plots are omitted). In the left panel, we plot

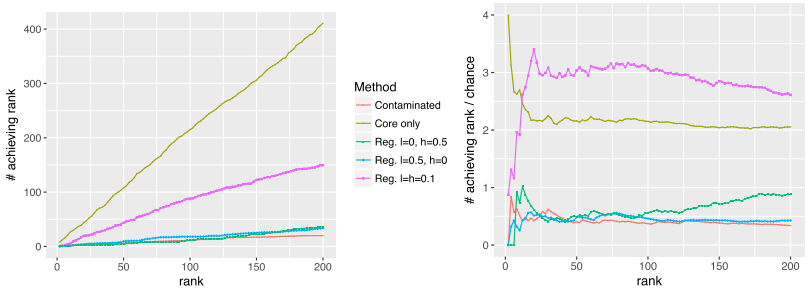


(a) Regularized VN ∘ GMM ∘ ASE; $\rho = 0.5$ (b) Regularized VN ∘ GMM ∘ ASE; $\rho = 0.3$

FIG 6. *In the right panel (resp., left panel), we plot the performance of VN ∘ GMM ∘ ASE in our contamination model in Section 3.1 with correlation $\rho = .5$ and $\rho = .3$ respectively, again averaged over $nMC = 50$ random seed sets of size $s = 10$. The gold line represents the idealized network pair, the red represents the contaminated network pair, and the other colors represent various levels of regularization. See Section 4.2 for further details.*

the modularity of the GMM clustering in the trimmed $G_2^{(\ell,h)}$ as a function of $\ell, h \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$. Note that we average the modularity values over $nMC = 50$ randomly selected seed sets of size $s = 10$. The color indicates the value of the modularity, with darker red indicating lower values and lighter yellow–white indicating larger values. We see that modularity is maximized near $(\ell, h) \approx (0.1, 0)$, and that the model-true trimming values $(\ell, h) = (0.1, 0.1)$ achieves relatively high modularity as well.

In the right panel, we plot the performance of VN ∘ GMM ∘ ASE ($\pm$2s.e.) in $(G_1, G_2) \sim \text{SBM}(0.7, 200, 2, B, \pi = (1/2, 1/2))$ again averaged over $nMC = 50$ random seed sets of size $s = 10$. The x-axis shows the ranks in the nomination list and the y-axis shows (on average) how many vertices $v \in G_1^{(i)}$, when viewed as the v.o.i., had their corresponding vertex of interest ranked in the top x by VN ∘ GMM ∘ ASE. The gold line represents performance in the idealized network pair $(G_1^{(i)}, G_2^{(i)})$; the red line for $(G_1^{(i)}, G_2^{(c)})$; the green line for $(G_1^{(i)}, G_2^{(0.1,0)})$; the blue line for $(G_1^{(i)}, G_2^{(0.1,0.1)})$; and the pink line for $(G_1^{(i)}, G_2^{(0.2,0.2)})$. We see here that, as expected, performance loss due to contamination is mitigated by using the true model-based trimming parameters $\ell = h = 0.1$, and using the modularity maximizing $\ell = 0.1, h = 0$. If we over-trim, here represented by $\ell = h = 0.2$, we see a degradation in performance; as expected from the low modularity value in the left panel for $\ell = h = 0.2$. We again see here the interesting phenomena observed in the motivating high school friendship network example of Section 1.1: modularity and subsequently VN performance tends to emphasize more trimming of the low degree vertices and less trimming of the high degree vertices. This suggests that low-degree contamination is most effective at thwarting the performance on VN ∘ GMM ∘ ASE, perhaps contrary to the intuition that high-degree nodes adversely affect concentration of adjacency matrices [25].



(a) # achieving rank $\leq x$ versus x (b) Chance normalized # achieving rank $\leq x$ versus x

FIG 7. *In panel a), we consider each vertex in* $G_1^{(i)}$ *as the v.o.i., and we plot the number of vertices amongst these v.o.i. (x-axis) that had their corresponding v.o.i. in* G_2 *ranked in the top* x*. The right panel shows the same result normalized by chance. We use 2 Monte Carlo replicates of* $s = 100$ *randomly chosen seeds, with the gold line representing the idealized network pair, the red line representing the contaminated, and the other colors representing various levels of regularization. See Section 4.3 for details.*

As in our motivating example, trimming can have the effect of removing v.o.i. from $G_2^{(c)}$, and we see this play out in Figure 5, in which we plot the performance of VN $\circ$ GMM $\circ$ ASE ($\pm$2s.e.) in $(G_1, G_2) \sim \text{SBM}(0.3, 200, 2, B, \pi = (1/2, 1/2))$ with $\rho = 0.7$ again averaged over $nMC = 50$ random seed sets of size $s = 10$. The x-axis shows the ranks in the nomination list and the y-axis shows (on average) how many vertices $v \in G_1^{(i)}$, when viewed as the v.o.i., had their corresponding vertex of interest ranked in the top x by VN $\circ$ GMM $\circ$ ASE. The gold line represents performance in the idealized network pair $(G_1^{(i)}, G_2^{(i)})$; the red line for $(G_1^{(i)}, G_2^{(c)})$; the green line for $(G_1^{(i)}, G_2^{(0.1,0)})$; the blue line for $(G_1^{(i)}, G_2^{(0.1,0.1)})$; and the pink line for $(G_1^{(i)}, G_2^{(0.2,0.2)})$.

As expected, over-regularizing results in a significant number of v.o.i. being trimmed and significant performance loss as compared to the more moderate choices of regularization. Lastly, exploring the affect of ρ on VN $\circ$ GMM $\circ$ ASE performance, we repeat the above experiment with $\rho = 0.5$, and $\rho = 0.3$. Results are plotted in Figure 6. As expected, the trends observed in Figure 4 hold here as well, with an across the board performance decrease as ρ decreases.

4.3. Microsoft bing entity graph transitions

In the next example, we consider a multigraph derived from one month of aggregate Bing entity graph transitions. The multigraph represents entity transitions, and each weighted edge-type of the multigraph represents aggregated signal that capture a transition rate between two entities while browsing. There are multiple ways that a transition between those entities could be made, so we count each aggregated signal separately using the different edge-types in the multigraph: one edge-type represents transitions that were made via a suggestion interface;

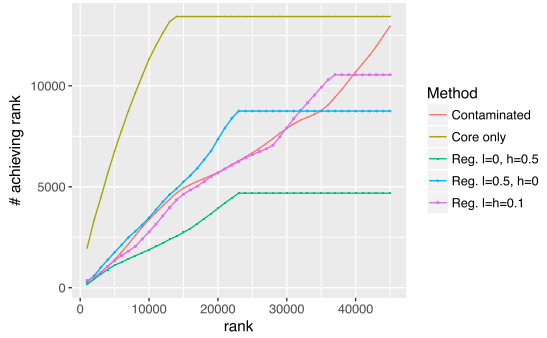


FIG 8. *We consider each vertex in $G_1^{(i)}$ as the v.o.i., and we plot the number of vertices amongst these v.o.i. (x-axis) that had their corresponding v.o.i. in G_2 ranked in the top x. The right panel shows the same result normalized by chance. We use 2 Monte Carlo replicates of $s = 100$ randomly chosen seeds (with the same seed sets as in Figure 7), with the gold line representing the idealized network pair, the red line representing the contaminated, and the other colors representing various levels of regularization. See Section 4 for details.*

the other edge-type represents transitions that we made independent of any suggestion interface. As such, one type will have a constrained set of transition probabilities (it can realistically only connect to a subset of the vertices in the graph), while the other will be more "unlimited" in that it may connect to any other entity in the entire graph.

The resulting graphs are symmetric, weighted and loop-free, with $G_1^{(i)}$ containing 13535 vertices and 519389 edges, $G_2^{(i)}$ containing 13535 vertices and 595047 edges, and the contaminated network $G_2^{(c)}$ containing 45816 vertices and 2848466 edges. Here, there is a 1-to-1 correspondence between the vertex sets of $G_1^{(i)}$ and $G_2^{(i)}$ with the contaminated network adding 32281 vertices to $G_2^{(c)}$ that do not have a corresponding vertex in $G_1^{(i)}$. In Figure 7, we explore the effect of this contamination (and the subsequent regularization) on VN $\circ$ GMM $\circ$ ASE.

Considering two randomly chosen sets of $s = 100$ seeds, we run VN $\circ$ GMM $\circ$ ASE on $(G_1^{(i)}, G_2^{(i)})$ (yellow line in Figure 7), on $(G_1^{(i)}, G_2^{(c)})$ (red line), on $(G_1^{(i)}, G_2^{(0.1,0.1)})$ (pink line); on $(G_1^{(i)}, G_2^{(0,0.5)})$ (green line); and on $(G_1^{(i)}, G_2^{(0.5,0)})$ (blue line). As in the simulations and motivating data example, we see the general trend of contamination adversely affecting performance and regularization ameliorating the effect of the contamination. Here, the regularized graph $G_2^{(0.1,0.1)}$ has 36808 vertices, and as expected, absolute performance (the left panel in Figure 7) in the clean case is better than in the regularized setting. From the right panel, we observe however, that the relative improvement over chance achieved in the regularized setting exceeds that in the clean setting, and we observe that VN $\circ$ GMM $\circ$ ASE performance is worse than chance in the contaminated and over-regularized network settings. While regularization has not recovered the performance in the idealized setting, the improvement induced via regularization is dramatic versus the contaminated setting. We also note

that the modularity levels for automating the choice of (ℓ, h) in this example are relatively stable to the trimming value, with the clustered $G_2^{(c)}$ achieving $Q = 0.52$, the clustered $G_2^{(c)}$ achieving $Q = 0.52$, the clustered $G_2^{(0,0.5)}$ achieving $Q = 0.57$, the clustered $G_2^{(0.5,0)}$ achieving $Q = 0.52$, and the clustered $G_2^{(0.1,0.1)}$ achieving $Q = 0.53$. Indeed, in this data example the graphs do not cluster particularly well under any trimming conditions, and a more modest trimming scheme is more effective for the subsequent VN inference task.

In Figure 8, we again consider the performance of VN $\circ$ GMM $\circ$ ASE with the same $nMC = 2$ randomly chose 100 vertex seed sets and various levels of regularization, here plotting over an extended x-axis. In pink we plot VN $\circ$ GMM $\circ$ ASE run on $(G_1^{(i)}, G_2^{(0.1,0.1)})$; in blue on $(G_1^{(i)}, G_2^{(0.5,0)})$; in green on $(G_1^{(i)}, G_2^{(0.3,0.3)})$; and in red on $(G_1^{(i)}, G_2^{(0,0.5)})$. This figure demonstrates another dramatic side effect of over-regularization: v.o.i. that are trimmed for $G_2^{(c)}$ can never be recovered by VN $\circ$ GMM $\circ$ ASE. This is represented by the horizontal asymptotes in Figure 8.

5. Discussion

Our motivating question is two-fold: What effect does adversarial contamination have on the performance of vertex nomination? Herein, we have demonstrated both theoretically and empirically that an adversary can cause our VN scheme to fail (i.e., nominate the wrong vertices). Empirically, we have also demonstrated that regularization can be effective for mitigating the effect of the contamination model posited herein, though we have not proven this result. Establishing the theoretical effect of regularization on VN is an open problem, and the subject of our present research.

In [27], the authors showed that there can be no universally consistent vertex nomination scheme assuming only one vertex of interest. In this paper, we have seen that with a suitable definition of a maximal consistency class and (possibly) multiple vertices of interest, there are infinitely many such consistency classes, which implies that ensemble methods cannot recover consistency and/or thwart an arbitrary adversary. This allows us to formulate our model of adversarial contamination in terms of consistency classes; indeed, an adversary for a particular VN rule aims to move the distribution out of the rule's consistency class. A natural next question to consider would be what effect regularization has on a VN rule's consistency class. Ideally, regularization enlarges the consistency class of a VN rule thereby making the adversary's job (i.e., moving the model out of the consistency class) more difficult. The interplay between the adversary and regularization in VN is central to this story, although we are only at the infancy of understanding it.

There are several issues compounding the theoretical analysis of regularization, even in the relatively simple setting posited herein. Indeed, the adversarially modified graph G_2 is, under our modeling assumptions of Section 3.1, a stochastic blockmodel, albeit with more blocks than in G_1. Theoretically analyzing the effect of our trimming regularizer in the context of VN $\circ$ GMM $\circ$ ASE

would require novel results in the concentration and spectral properties of regularized random graphs, akin (though different from) those in [25]. Indeed, regularization and its effect on the spectral analysis of random graphs is still not very well understood, as regularization often induces complicated dependency structure into the resulting regularized graph. Existing spectral analysis techniques often require relating differences in eigenvectors/eigenvalues for perturbed matrices with independent (or weakly dependent [9]) entries, which is not directly applicable in the regularized setting. Hence, new techniques must be developed to understand regularization. We believe that our theoretical findings are a necessary first step to begin to understand how an adversary can affect vertex nomination.

Our proposed definition of an adversary is suited to a general random graph setting, and it provides a simple surrogate in which to study the effect of contamination in real data examples. From our simulation study and real data examples we have seen that a particular VN rule (VN∘GMM∘ASE) succeeds before adversarial contamination, fails after contamination, and succeeds after graph regularization. We are currently exploring the effect of contamination on a broader class of VN rules, and considering other models for adversarial contamination and subsequent regularization. Finally, while we have partially answered in the negative our question about whether consistency can be retained in the general adversarial setting, another valid consideration is whether there are adversarial models for which the adversary does *not* affect consistency. While we believe even simple manipulation on the edges of G_2 can affect consistency, it may be possible to derive bounds and phase transitions on the number of edges (or vertices) that an adversary would need to modify to change the result. Mathematically, this is akin to finding limits on the size of $|V_{\mathcal{A}}|$ in our definition of an adversary.

Acknowledgements

This material is based on research sponsored by the Air Force Research Laboratory and DARPA under agreement number FA8750-18-2-0035. This work is also supported in part by the D3M program of the Defense Advanced Research Projects Agency. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and DARPA or the U.S. Government.

Appendix A: Proof of Bayes optimality for the scheme in Section 2.2.1

For each $i \in [h]$, $j \in [m]$, $v \in V^*$, $\Phi \in \mathcal{V}_{nm}$, define

$$U_{i,\mathbf{g}}^{j,v} := \left\{ (g_1, g_2) \in \left(g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})] \right) \text{ s.t. } \operatorname{rank}_{\Phi(g_1, \mathfrak{o}(g_2), V^*)}(\mathfrak{o}(v)) = j \right\}$$

$$
\begin{aligned}
&= \left\{(g_1, g_2) \in \left(g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})]\right) \text{ s.t. } \Phi(g_1, \mathfrak{o}(g_2), V^*)[j] = \mathfrak{o}(v)\right\} \\
&= \left\{(g_1, g_2) \in \left(g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})]\right) \text{ s.t. } \exists \text{ iso. } \sigma \text{ s.t. } \sigma(\mathfrak{o}(g_2^{(i)})) = \mathfrak{o}(g_2)\right. \\
&\qquad\qquad \left. \text{and } \sigma\left(\Phi(g_1^{(i)}, \mathfrak{o}(g_2^{(i)}), V^*)[j]\right) = \mathfrak{o}(v)\right\} \\
&= \left(g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})]\right)_{\Phi(g_1^{(i)}, \mathfrak{o}(g_2^{(i)}), V^*)[j] = \mathfrak{o}(v)}.
\end{aligned}
$$

Lastly, for $(g_1, g_2) \in \mathcal{G}_n^a \times \mathcal{G}_m^a$, define $p_\Phi \in [0,1]^m$ via

$$
\begin{aligned}
p_\Phi^{(i)}[g_1, \mathfrak{o}(g_2), V^*][j] &= p_\Phi^{(i)}[j] \\
&:= \sum_{v \in V^*} \mathbb{P}_{F_{c,\theta}^{(n,m)}}\left[U_{i,\mathbf{g}}^{j,v} \mid (g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})]\right] \\
&= \mathbb{P}_{F_{c,\theta}^{(n,m)}}[E],
\end{aligned}
$$

where E is the (conditional) event

$$
\left\{(g_1, [\mathfrak{o}(g_2)])_{\Phi(g_1, \mathfrak{o}(g_2), V^*)[j] \in \mathfrak{o}(V^*)} \middle| (g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})\right\}
$$

Note that, by definition, p_{Φ^*} majorizes p_Φ.

To show that Φ^* is Bayes optimal for $L_k^{(1)}$ (the proof for $L_k^{(2)}$ being completely analogous), we have that for $k \leq m-1$,

$$
\begin{aligned}
L_k^{(1)}(\Phi, V^*) &= 1 - \frac{1}{|V^*|} \sum_{v \in V^*} \mathbb{P}_{F_{c,\theta}^{(n,m)}}(\operatorname{rank}_{\Phi(G_1, \mathfrak{o}(G_2), V^*)}(\mathfrak{o}(v)) \leq k) \\
&= 1 - \frac{1}{|V^*|} \sum_{v \in V^*} \sum_{j \leq k} \mathbb{P}_{F_{c,\theta}^{(n,m)}}(\operatorname{rank}_{\Phi(G_1, \mathfrak{o}(G_2), V^*)}(\mathfrak{o}(v)) = j) \\
&= 1 - \frac{1}{|V^*|} \sum_{\mathcal{P}_\mathbf{g}} \sum_{j \leq k} \sum_{v \in V^*} \left(\mathbb{P}_{F_{c,\theta}^{(n,m)}}\left[U_{i,\mathbf{g}}^{j,v} \mid \left(g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})]\right)\right]\right. \\
&\qquad\qquad \left. \times \mathbb{P}_{F_{c,\theta}^{(n,m)}}\left[\left(g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})]\right)\right]\right) \\
&= 1 - \frac{1}{|V^*|} \sum_{\mathcal{P}_\mathbf{g}} \sum_{j \leq k} p_\Phi^{(i)}[j]\, \mathbb{P}_{F_{c,\theta}^{(n,m)}}\left[\left(g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})]\right)\right] \\
&\geq 1 - \frac{1}{|V^*|} \sum_{\mathcal{P}_\mathbf{g}} \sum_{j \leq k} p_{\Phi^*}^{(i)}[j]\, \mathbb{P}_{F_{c,\theta}^{(n,m)}}\left[\left(g_1^{(i)}, [\mathfrak{o}(g_2^{(i)})]\right)\right] \\
&= L_k^{(1)}(\Phi^*, V^*),
\end{aligned}
$$

as desired.

Appendix B: Proof of Theorem 15

We first note that the growth condition on $|V_n^*|$ and on k_n in the precision case ensures that the result for precision and recall consistency follow from each

other, and so we will focus our attention on recall consistency. The analogous result for precision follows mutatis mutandis.

Consider the following network construction for a network of size n. Let $\xi_n = \max(k_n, |V_n^*|)$. For a fixed $p \in (0,1)$, let $B_1, \cdots, B_{\lfloor \frac{n/3}{\xi_n} \rfloor}$ be i.i.d. $\mathrm{ER}(\xi_n, p)$ random graphs. Let H_n be a complete graph on $n - \xi_n \left\lfloor \frac{n/3}{\xi_n} \right\rfloor$ vertices. Label the vertices

$$\begin{aligned}
&\text{of } B_i \text{ with } \{1, 2, 3, \cdots, \xi_n\};\\
&\text{of } B_2 \text{ with } \{\xi_n + 1, \xi_n + 2, \xi_n + 3, \cdots, 2\xi_n\};\\
&\qquad\qquad\vdots\\
&\text{of } B_{i-1} \text{ with } \{(i-2)\xi_n + 1, (i-2)\xi_n + 2,\\
&\qquad\qquad (i-2)\xi_n + 3, \cdots, (i-1)\xi_n\};\\
&\text{of } B_1 \text{ with } \{(i-1)\xi_n + 1, (i-1)\xi_n + 2,\\
&\qquad\qquad (i-1)\xi_n + 3, \cdots, i\xi_n\};\\
&\text{of } B_{i+1} \text{ with } \{i\xi_n + 1, i\xi_n + 2, i\xi_n + 3, \cdots, (i+1)\xi_n\};\\
&\qquad\qquad\vdots\\
&\text{of } B_{\lfloor \frac{n/3}{\xi_n} \rfloor} \text{ with } \left\{ \left(\left\lfloor \frac{n/3}{\xi_n} \right\rfloor - 1 \right) \xi_n + 1, \right.\\
&\qquad\qquad \left. \left(\left\lfloor \frac{n/3}{\xi_n} \right\rfloor - 1 \right) \xi_n + 2, \cdots, \left\lfloor \frac{n/3}{\xi_n} \right\rfloor \xi_n \right\};\\
&\text{of } H_n \text{ with } \left\{ \left\lfloor \frac{n/3}{\xi_n} \right\rfloor \xi_n + 1, \left\lfloor \frac{n/3}{\xi_n} \right\rfloor \xi_n + 2, \ldots, n \right\}.
\end{aligned}$$

For each $\ell \in \left[\left\lfloor \frac{n/3}{\xi_n} \right\rfloor \right]$ and each vertex v in $V(B_\ell)$, independent of all other edges in the network, select ℓ vertices uniformly at random from H_n, i.e., from

$$\left\{ \left\lfloor \frac{n/3}{\xi_n} \right\rfloor \xi_n + 1, \left\lfloor \frac{n/3}{\xi_n} \right\rfloor \xi_n + 2, \ldots, n \right\}.$$

Denote this set of ℓ vertices via $V_{v,\ell}$—and place an edge between v and each vertex in $V_{v,\ell}$. Let $\mathcal{H}_{n,i}$ be the collection of all graphs possible under the above construction, and let $F_{n,i}$ be the distribution on $\mathcal{H}_{n,i}$ outlined above.

With $c = n$, the correspondence the identity, and (where $|V_n^*| = \nu_n$) $V_n^* = \{v_i\}_{i=1}^{\nu_n} = \{u_i\}_{i=1}^{\nu_n} = [\nu_n]$, define the collection of nominatable distributions

$$\left\{ \widetilde{F}_{n,i} \right\}_{i=1}^{\lfloor \frac{n/3}{\xi_n} \rfloor}$$

via $\widetilde{F}_{n,i} = F_{n,1} \times F_{n,i}$ (where "$\times$" denotes the usual product measure).

Suppose a VN rule $\mathbf{\Phi} = (\Phi_n)_{n=n_0}^{\infty}$ is level-(k_n) recall consistent for $\mathbf{F}_i = (\widetilde{F}_{n,i})_{n=n_0}^{\infty}$. Then, by definition

$$\lim_{n\to\infty} L_{k_n}^{(1)}(\Phi_n, V^*) - L_{k_n}^{*,(1)}(V^*, \widetilde{F}_{n,i}) = 0.$$

However, note that here

$$L_{k_n}^{*,(1)}(V^*, \widetilde{F}_{n,i}) \leq 1 - \frac{k_n}{\xi_n}.$$

Indeed, for a given $\widetilde{F}_{n,i}$, consider the following VN scheme Ψ_n. First identify the vertices of H_n; this is possible as H_n is a complete subgraph of order $\geq 2n/3$, and each B_i is of order $o(n)$ with vertices of degree at most $\left\lfloor \frac{n/3}{\xi_n} \right\rfloor \leq n/3$. Each B_ℓ can then be recovered and identified by computing the number of edges between H_n and each vertex $v \in V \setminus V(H_n)$; in particular B_i can be identified as the set of vertices in $V \setminus V(H_n)$ with i edges to $V(H_n)$. Let ψ_n then rank the vertices in B_i (in arbitrary order) at the top of its nomination list. It is immediate then that

$$L_{k_n}^{(1)}(\Psi_n, V^*) = 1 - \frac{k_n}{\xi_n}.$$

By the distributional symmetry of the v.o.i., we have that for $v \in V^*$,

$$\mathbb{P}_{\widetilde{F}_{n,i}}(\text{rank}_{\Phi_n(G_1, \mathfrak{o}(G_2), V^*)}(\mathfrak{o}(v)) \geq k_n + 1) = L_{k_n}^{(1)}(\Phi_n, V^*).$$

For any $\epsilon > 0$ and sufficiently large n, consistency ensures that

$$\mathbb{P}_{\widetilde{F}_{n,i}}(\text{rank}_{\Phi_n(G_1, \mathfrak{o}(G_2), V^*)}(\mathfrak{o}(v)) \geq k_n + 1) \leq \epsilon + \left(1 - \frac{k_n}{\xi_n}\right).$$

The internal consistency criterion in the definition of VN schemes (Eq. (2)), then implies that

$$\mathbb{P}_{\widetilde{F}_{n,i}}(\text{rank}_{\Phi_n(G_1, \mathfrak{o}(G_2), V^*)}(\mathfrak{o}(v)) \geq k_n + 1) \leq \epsilon + \left(1 - \frac{k_n}{\xi_n}\right) \tag{10}$$

for each $v \in \{1, 2, \cdots, \xi_n\}$. Now, suppose that Φ is also level-k_n recall consistent for $\mathbf{F}_j$ for $j \neq i$. By similar logic, we must have that

$$\mathbb{P}_{\widetilde{F}_{n,j}}(\text{rank}_{\Phi_n(G_1, \mathfrak{o}(G_2), V^*)}(\mathfrak{o}(v)) \geq k_n + 1) \leq \epsilon + \left(1 - \frac{k_n}{\xi_n}\right) \tag{11}$$

for each $v \in \{1, 2, \cdots, \xi_n\}$ for sufficiently large n.

Let $\sigma_{i \leftrightarrow j}$ be the permutation on $\{1, .., n\}$ defined as

$$\sigma(\ell) := \begin{cases} (i-1)\xi_n + \ell & \ell \in \{1, 2, ..., \xi_n\} \\ \ell - (j-1)\xi_n & \ell \in \{(j-1)\xi_n + 1, ..., j\xi_n\} \\ (j-i)\xi_n + \ell & \ell \in \{(i-1)\xi_n + 1, ..., i\xi_n\} \\ \ell & \text{otherwise.} \end{cases}$$

Now, for each $v \in [\xi_n]$, define the sets

$$E_{n,i}^v := \{(g_1, g_2) \in \mathcal{H}_{n,1} \times \mathcal{H}_{n,i} : \text{rank}_{\Phi_n(g_1, \mathfrak{o}(g_2), V_n^*)}(\mathfrak{o}(v)) \leq k_n\}$$

$$
\begin{aligned}
B^v_{n,i,j} &:= \{(g_1, g_2) \in \mathcal{H}_{n,1} \times \mathcal{H}_{n,i} : \mathrm{rank}_{\Phi_n(g_1, \mathfrak{o}(g_2), V_n^*)}(\mathfrak{o}[(j-1)\xi_n + v]) \leq k_n\} \\
E^v_{n,j} &:= \{(g_1, g_2) \in \mathcal{H}_{n,1} \times \mathcal{H}_{n,j} : \mathrm{rank}_{\Phi_n(g_1, \mathfrak{o}(g_2), V_n^*)}(\mathfrak{o}(v) \leq k_n\} \\
E_{n,j} &:= \{(g_1, \sigma(g_2)) \in \mathcal{G}_n \times \mathcal{G}_n : (g_1, g_2) \in E_{n,i}\} \\
&= \{(g_1, \sigma(g_2)) \in \mathcal{G}_{n,j} : \mathrm{rank}_{\Phi(G_1, o(G_2), v_1)}(o(\sigma(u_1))) \leq k\}.
\end{aligned}
$$

By consistency with respect to $\tilde{F}_{n,i}$ and $\tilde{F}_{n,j}$, i.e., by Eqs. 10–11, we have that for any $\epsilon > 0$, there exists $\tilde{n}$ such that for $n \geq \tilde{n}$, we have

$$
\begin{aligned}
\mathbb{P}_{\tilde{F}_{n,i}}(E^v_{n,i}) &\geq \frac{k_n}{\xi_n} - \epsilon; \\
\mathbb{P}_{\tilde{F}_{n,j}}(E^v_{n,j}) &\geq \frac{k_n}{\xi_n} - \epsilon.
\end{aligned} \tag{12}
$$

As $(G_1, G_2) \sim \widetilde{F}_{n,i} \Leftrightarrow (G_1, \sigma(G_2)) \sim \widetilde{F}_{n,j}$, the

$$
\mathbb{P}_{\tilde{F}_{n,j}}(E^v_{n,j}) = \mathbb{P}_{\tilde{F}_{n,i}}(B^v_{n,i,j}) \geq \frac{k_n}{\xi_n} - \epsilon. \tag{13}
$$

For each $v \in [\xi_n]$ and $h \in [k_n]$ and $i \in [n]$, define the sets

$$
R_{i,v,h} := \left\{(g_1, g_2) \in \mathcal{H}_{n,1} \times \mathcal{H}_{n,i} : \mathrm{rank}_{\Phi_n(g_1, \mathfrak{o}(g_2), V_n^*)}(\mathfrak{o}(v)) = h\right\}
$$

Then, define

$$
\begin{aligned}
\alpha_{v,h} &= \mathbb{P}_{\tilde{F}_{n,i}}\big[\, R_{i,v,h} \,\big]; \\
\beta_{v,h} &= \mathbb{P}_{\tilde{F}_{n,i}}\big[\, S_{i,v,h} \,\big].
\end{aligned}
$$

By Eq. (12), we have that $\sum_{h=1}^{k_n} \alpha_{v,h} \geq \frac{k_n}{\xi_n} - \epsilon$, and by Eq. (13), we have that $\sum_{h=1}^{k_n} \beta_{v,h} \geq \frac{k_n}{\xi_n} - \epsilon$. Noting that for each $h \in [k_n]$

$$
\begin{aligned}
1 &\geq \mathbb{P}_{\tilde{F}_{n,i}}\left[\left(\cup_{v \in \xi_n] } R_{i,v,h}\right) \cup \left(\cup_{v \in [\xi_n]} S_{i,v,h}\right)\right] \\
&= \sum_{v \in [\xi_n]} \mathbb{P}_{\tilde{F}_{n,i}}(R_{i,v,h}) + \mathbb{P}_{\tilde{F}_{n,i}}(S_{i,v,h}) \\
&= \xi_n \alpha_{v,h} + \xi_n \beta_{v,h},
\end{aligned}
$$

and hence

$$
\beta_{v,h} \leq \frac{1}{\xi_n} - \alpha_{v,h}.
$$

Plugging this into Eq. (13) then yields

$$
\begin{aligned}
\frac{k_n}{\xi_n} - \epsilon \leq &= \mathbb{P}_{\tilde{F}_{n,i}}(B^v_{n,i,j}) \\
&= \sum_{h=1}^{k_n} \beta_{v,h}
\end{aligned}
$$

$$\leq \frac{k_n}{\xi_n} - \sum_{h=1}^{k_n} \alpha_{v,h}$$
$$\leq \epsilon.$$

As ϵ was chosen arbitrarily, and $\frac{k_n}{\xi_n}$ is bounded away from 0 by assumption, we reach our desired contradiction, and Φ cannot be consistent with respect to both $\mathbf{F}_i$ and $\mathbf{F}_j$. As $i, j \in \lfloor \frac{n_0/3}{\xi_{n_0}} \rfloor$ were arbitrary, we see that there must be at least countably many consistency classes (since there are at least $\lfloor \frac{n_0/3}{\xi_{n_0}} \rfloor$ and we can let n_0 tend to infinity).

References

[1] J. Arroyo-Relión, D. Kessler, E. Levina, and S. Taylor. Network classification with applications to brain connectomics. *The Annals of Applied Statistics*, 13(3):1648–1677, 09 2019. MR4019153

[2] A. Athreya, D. Fishkind, K. Levin, V. Lyzinski, Y. Park, Y. Qin, D. Sussman, M. Tang, J. Vogelstein, and C. Priebe. Statistical inference on random dot product graphs: A survey. *Journal of Machine Learning Research*, 18, 09 2017. MR3827114

[3] A. Athreya, C. E. Priebe, M. Tang, V. Lyzinski, D.J. Marchette, and D.L. Sussman. A limit theorem for scaled eigenvectors of random dot product graphs. *Sankhya A*, 1–18, 2015. MR3494576

[4] P. J. Bickel and A. Chen. A nonparametric view of network models and Newman-Girvan and other modularities. *Proceedings of the National Academy of Sciences, USA*, 106:21068–21073, 2009.

[5] P. J. Bickel, D. Choi, X. Chang, and H. Zhang. Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *The Annals of Statistics*, 41(4):1922–1943, 2013. MR3127853

[6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008.

[7] E. Bullmore and O. Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186, 2009.

[8] T. T. Cai and X. Li. Robust and computationally feasible community detection in the presence of arbitrary outlier nodes. *The Annals of Statistics*, 43(3):1027–1059, 2015. MR3346696

[9] J. Cape, M. Tang, and C. E. Priebe. The two-to-infinity norm and singular subspace geometry with applications to high-dimensional statistics. *The Annals of Statistics*, 47(5):2405–2439, 2019. MR3988761

[10] S. Chatterjee. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2014. MR3285604

[11] L. Chen, C. Shen, J. T. Vogelstein, and C. E. Priebe. Robust vertex classi-

fication. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):578–590, 2016. MR3338694

[12] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004.

[13] G. Coppersmith. Vertex nomination. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(2):144–153, 2014.

[14] G. A. Coppersmith and C. E. Priebe. Vertex nomination via content and context. *arXiv preprint arXiv:1201.4118*, 2012.

[15] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song. Adversarial attack on graph structured data. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1115–1124, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[16] D. Edge, J. Larson, M. Mobius, and C. White. Trimming the hairball: Edge cutting strategies for making dense graphs usable. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3951–3958. IEEE, 2018.

[17] D. E. Fishkind, V. Lyzinski, H. Pao, L. Chen, and C. E. Priebe. Vertex nomination schemes for membership prediction. *The Annals of Applied Statistics*, 9(3):1510–1532, 2015. MR3418733

[18] P. Foggia, G. Percannella, and M. Vento. Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence*, 28(01):1450001, 2014. MR3190010

[19] C. Fraley and A. E. Raftery. Mclust: Software for model-based cluster analysis. *Journal of Classification*, 16(2):297–306, 1999. MR2019797

[20] K. J. Gile and M. S. Handcock. 7. Respondent-driven sampling: An assessment of current methodology. *Sociological Methodology*, 40(1):285–327, 2010.

[21] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002. MR1908073

[22] D. D. Heckathorn. Respondent-driven sampling: a new approach to the study of hidden populations. *Social Problems*, 44(2):174–199, 1997.

[23] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983. MR0718088

[24] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, pages 43–58. ACM, 2011.

[25] C. M. Le, E. Levina, and R. Vershynin. Concentration and regularization of random graphs. *Random Structures & Algorithms*, 51(3):538–561, 2017. MR3689343

[26] D. S. Lee and C. E. Priebe. Bayesian vertex nomination. *arXiv preprint arXiv:1205.5082*, 2012. MR1182312

[27] V. Lyzinski, K. Levin, and C. E. Priebe. On consistent vertex nomina-

tion schemes. *Journal of Machine Learning Research*, to appear, 2019. MR3960923

[28] V. Lyzinski, D. L. Sussman, M. Tang, A. Athreya, and C. E. Priebe. Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding. *Electronic Journal of Statistics*, 8:2905–2922, 2014. MR3299126

[29] V. Lyzinski, M. Tang, A. Athreya, Y. Park, and C. E. Priebe. Community detection and classification in hierarchical stochastic blockmodels. *IEEE Transactions on Network Science and Engineering*, 4(1):13–26, 2017. MR3625952

[30] D. Marchette, C. E. Priebe, and G. Coppersmith. Vertex nomination via attributed random dot product graphs. In *Proceedings of the 57th ISI World Statistics Congress*, volume 6, page 16, 2011.

[31] S. Maslov and K. Sneppen. Specificity and stability in topology of protein networks. *Science*, 296(5569):910–913, 2002.

[32] R. Mastrandrea, J. Fournet, and A. Barrat. Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PloS One*, 10(9):e0136497, 2015.

[33] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[34] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006. MR2676073

[35] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99(suppl 1):2566–2572, 2002.

[36] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[37] H. G. Patsolic, Y. Park, V. Lyzinski, and C. E. Priebe. Vertex nomination via local neighborhood matching. *arXiv preprint arXiv:1705.00674*, 2017.

[38] T. Qin and K. Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. *Advances in Neural Information Processing Systems*, 2013.

[39] K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39:1878–1915, 2011. MR2893856

[40] P. H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. MR0215870

[41] J. Scott. *Social Network Analysis*. Sage, 2017.

[42] O. Sporns. Graph theory methods: applications in brain networks. *Dialogues in Clinical Neuroscience*, 20(2):111, 2018.

[43] D. L. Sussman, M. Tang, and C. E. Priebe. Consistent latent position estimation and vertex classification for random dot product graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):48–57,

2014.

[44] M. Tang, A. Athreya, D. L. Sussman, V. Lyzinski, Y. Park, and C. E. Priebe. A semiparametric two-sample hypothesis testing problem for random graphs. *Journal of Computational and Graphical Statistics*, 26(2):344–354, 2017. MR3640191

[45] M. Tang, A. Athreya, D. L. Sussman, V. Lyzinski, and C. E. Priebe. A nonparametric two-sample hypothesis testing problem for random dot product graphs. *Bernoulli*, 23(3):1599–1630, 2017. MR3624872

[46] J. T. Vogelstein, W. G. Roncal, R. J. Vogelstein, and C. E. Priebe. Graph classification using signal-subgraphs: Applications in statistical connectomics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1539–1551, 2013. MR3338694

[47] J. Yan, X. Yin, W. Lin, C. Deng, H. Zha, and X. Yang. A short survey of recent advances in graph matching. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 167–174. ACM, 2016.

[48] J. Yoder, L. Chen, H. Pao, E. Bridgeford, K. Levin, D. E. Fishkind, C. E. Priebe, and V. Lyzinski. Vertex nomination: The canonical sampling and the extended spectral nomination schemes. *arXiv preprint arXiv:1802.04960*, 2018. MR4055035

[49] M. Zhu and A. Ghodsi. Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics & Data Analysis*, 51(2):918–930, 2006. MR2297497

[50] D. Zügner, A. Akbarnejad, and S. Günnemann. Adversarial attacks on neural networks for graph data. *arXiv preprint arXiv:1805.07984*, May 2018.