Consistent Recovery Threshold of Hidden Nearest Neighbor Graphs

Jian Ding, Yihong Wu, *Member, IEEE*, Jiaming Xu, and Dana Yang

***Abstract*— Motivated by applications such as discovering strong ties in social networks and assembling genome subsequences in biology, we study the problem of recovering a hidden $2k$-nearest neighbor (NN) graph in an n-vertex complete graph, whose edge weights are independent and distributed according to P_n for edges in the hidden $2k$-NN graph and Q_n otherwise. The special case of Bernoulli distributions corresponds to a variant of the Watts-Strogatz small-world graph. We focus on two types of asymptotic recovery guarantees as $n \to \infty$: (1) exact recovery: all edges are classified correctly with probability tending to one; (2) almost exact recovery: the expected number of misclassified edges is $o(nk)$. We show that the maximum likelihood estimator achieves (1) exact recovery for $2 \leq k \leq n^{o(1)}$ if $\liminf \frac{2\alpha_n}{\log n} > 1$; (2) almost exact recovery for $1 \leq k \leq o\left(\frac{\log n}{\log \log n}\right)$ if $\liminf \frac{kD(P_n||Q_n)}{\log n} > 1$, where $\alpha_n \triangleq -2\log\int\sqrt{dP_n dQ_n}$ is the Rényi divergence of order $\frac{1}{2}$ and $D(P_n||Q_n)$ is the Kullback-Leibler divergence. Under mild distributional assumptions, these conditions are shown to be information-theoretically necessary for any algorithm to succeed. A key challenge in the analysis is the enumeration of $2k$-NN graphs that differ from the hidden one by a given number of edges. We also analyze several computationally efficient algorithms and provide sufficient conditions under which they achieve exact/almost exact recovery. In particular, we develop a polynomial-time algorithm that attains the threshold for exact recovery under the small-world model.**

***Index Terms*— Nearest neighbor (NN) graphs, small-world graphs, information-theoretic lower bounds.**

I. Introduction

THE strong and weak ties are essential for information diffusion, social cohesion, and community organization in social networks [1]. The strong ties between close friends are responsible for forming tightly-knit groups, while the weak ties between acquaintances are crucial for binding groups of strong ties together [2]. The celebrated Watts-Strogatz small-world graph [3] is a simple network model that exhibits both strong and weak ties. It posits that n nodes are located on a ring and starts with a $2k$-nearest neighbor (NN) graph of strong ties, where each node is connected to its $2k$ nearest neighbors (k on the left and k on the right) on the ring. Then to generate weak ties, for every node, each of its strong ties is rewired with probability ϵ to a node chosen uniformly at random. As ϵ varies from 0 to 1, the graph interpolates between a ring lattice and an Erdős-Rényi random graph; for intermediate values of ϵ, the graph is a small-world network: highly clustered with many triangles, yet with a small diameter.

The Watts-Strogatz small-world graph and its variants, albeit simple, have been extensively studied and widely used in various disciplines to model real networks beyond social networks, such as academic collaboration network [4], metabolic networks [5], brain networks [6], and word co-occurrence networks in language modeling [7], [8]. Most of the previous work focuses on studying the structures of small-world graphs [9] and their algorithmic consequences [10]–[12]. However, in many practical applications, it is also of interest to distinguish strong ties from weak ones [13]–[16]. For example, in Facebook [17] or Twitter network [18], identifying the close ties among a user's potentially hundreds of friends provides valuable information for marketing and ad placements. Even when additional link attribute information (such as the communication time in who-talks-to-whom networks [19]) are available to be used to measure the strength of the tie, the task of discovering strong ties could still be challenging, as the link attributes are potentially noisy or only partially observed, obscuring the inherent tie strength. Therefore, it is of fundamental importance, in both theory and practice, to understand when and how we can infer strong ties from the noisy and partially observed network data. In this paper, we address this question in the following statistical model:

Definition 1 (Hidden $2k$-NN Graph Recovery):

Given: $n \geq 1$, and two distributions P_n and Q_n, parametrized by n.

Observation: A randomly weighted, undirected complete graph w with a hidden $2k$-NN graph x^* on n vertices, such that the edge weights are independent, and for each edge e, the edge weight w_e is distributed as P_n if e is an edge in x^* and as Q_n otherwise.

Inference Problem: Recover the hidden $2k$-NN graph x^* from the observed random graph.

Manuscript received November 18, 2019; revised May 5, 2021; accepted May 14, 2021. Date of publication June 3, 2021; date of current version July 14, 2021. The work of Jian Ding was supported by NSF under Grant DMS-1757479. The work of Yihong Wu was supported in part by NSF under Grant CCF-1900507, in part by the NSF CAREER Award under Grant CCF-1651588, and in part by the Alfred Sloan Fellowship. The work of Jiaming Xu was supported by NSF under Grant IIS-1838124, Grant CCF-1850743, and Grant CCF-1856424. The work of Dana Yang was supported by NSF under Grant CCF-1850743. This article was presented in part at the 2020 Annual Conference on Learning Theory (COLT). *(Corresponding author: Dana Yang.)*

Jian Ding is with the Department of Statistics, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: dingjian@wharton.upenn.edu).

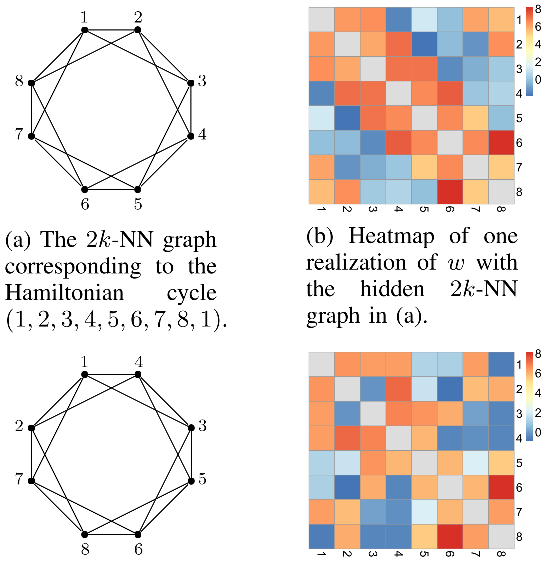
Yihong Wu is with the Department of Statistics and Data Science, Yale University, New Haven, CT 06520 USA (e-mail: yihong.wu@yale.edu).

Jiaming Xu and Dana Yang are with The Fuqua School of Business, Duke University, Durham, NC 27708 USA (e-mail: jx77@duke.edu; xiaoqian.yang@duke.edu).

Communicated by A. Gohari, Associate Editor At Large.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2021.3085773.

Digital Object Identifier 10.1109/TIT.2021.3085773



(a) The $2k$-NN graph corresponding to the Hamiltonian cycle $(1, 2, 3, 4, 5, 6, 7, 8, 1)$.

(b) Heatmap of one realization of w with the hidden $2k$-NN graph in (a).

(c) The $2k$-NN graph corresponding to the Hamiltonian cycle $(1, 4, 3, 5, 6, 8, 7, 2, 1)$.

(d) Heatmap of one realization of w with the hidden $2k$-NN graph in (c).

Fig. 1. Examples of $2k$-NN graphs for $n = 8$ and $k = 2$. The edge weight w_e is distributed $P_n = \mathcal{N}(6, 1)$ if e is an edge in the $2k$-NN graph. Otherwise $w_e \sim Q_n = \mathcal{N}(0, 1)$.

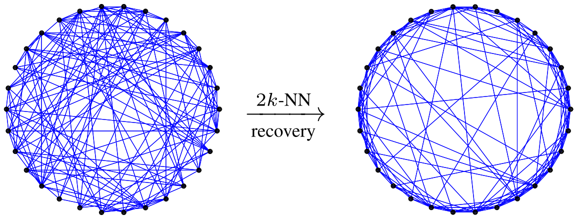


Fig. 2. Left: An observed graph generated by the hidden $2k$-NN graph model with $n = 30$ vertices, $k = 4$, $P_n = \text{Bern}(0.8)$, and $Q_n = \text{Bern}(0.09)$; Right: the observed graph with vertices rearranged according to the latent $2k$-NN graph.

See Figs. 1 and 2 for graphical illustrations of the model and the reconstruction problem. Note that every $2k$-NN graph x can be described by a permutation σ on $[n]$ as follows: first, construct a Hamiltonian cycle $(\sigma(1), \sigma(2), \ldots, \sigma(n), \sigma(1))$, then connect pairs of vertices that are at distance at most k on the cycle (cf. Figs. 1a and 1c).

The $2k$-NN model encompasses partially observed networks as a special case. This can be accomplished by considering $P_n = \epsilon\delta_* + (1-\epsilon)P'_n$ and $Q_n = \epsilon\delta_* + (1-\epsilon)Q_{n'}$ where $*$ is a special symbol outside of the support of P'_n and Q'_n indicating those edge weights that are unobserved. When P_n and Q_n are Bernoulli distributions with corresponding success probabilities $p_n > q_n$, we arrive at a variant of the Watts-Strogatz small-world graph.

The problem of recovering a hidden NN graph is also motivated by *de novo genome assembly*, the reconstruction of an organism's long sequence of A, G, C, T nucleotides from fragmented sequencing data, which is one of the most pressing challenges in genomics [20], [21]. A key obstacle of the current high-throughput genome assembly technology is *genome scaffolding*, that is, extending genome subsequences (so-called contigs) to the whole genome by ordering them according to their positions on the genome. Thanks to recent advances in sequencing technology [22], [23], this process is aided by long-range linking information between contigs in the form of randomly sampled Hi-C reads, where a much larger concentration of Hi-C reads exist between nearby contigs on the genome than those that are far apart. By representing each contig as a node, the underlying true ordering of contigs on the genome as the hidden Hamiltonian cycle, and the counts of the Hi-C reads linking the contigs as edge weights, the previous work [24] casts genome scaffolding as a hidden Hamiltonian cycle recovery problem with $P_n = \text{Pois}(\lambda_n)$ and $Q_n = \text{Pois}(\mu_n)$ with $\lambda_n \geq \mu_n$, where λ_n and μ_n are the average number of Hi-C reads between adjacent and non-adjacent contigs respectively; this is a special case of our model for $k = 1$. However, this hidden Hamiltonian cycle model only takes into account of the signal – an elevated mean number of Hi-C reads – in the immediately adjacent contigs on the genome; in reality, nearby contigs (e.g. two-hop neighbors) also demonstrate stronger signal than those that are far apart. By considering $k > 1$, our general $2k$-NN graph model is a closer approximation to the real data, capturing the large Hi-C counts observed between near contigs which can be used to better assemble the genome. Indeed, as our theory later suggests, the information provided by multi-hop neighbors strictly improves the recovery threshold.

Note that in the aforementioned applications we often have $k \ll n$; thus in this paper we focus on the regime of $k = n^{o(1)}$ and study the following two types of recovery guarantees. Let $x^* \in \{0,1\}^{\binom{n}{2}}$ denote the adjacency vector of the hidden $2k$-NN graph, where $x^*_e = 1$ for every edge e in the hidden $2k$-NN graph and $x^*_e = 0$ otherwise. Let $\mathcal{X}$ denote the collection of adjacency vectors of all $2k$-NN graphs with vertex set $[n]$.

Definition 2 (Exact Recovery): An estimator $\widehat{x} = \widehat{x}(w) \in \{0,1\}^{\binom{n}{2}}$ achieves exact recovery if, as $n \to \infty$,

$$\sup_{x^* \in \mathcal{X}} \mathbb{P}\{\widehat{x} \neq x^*\} = o(1),$$

where w is distributed according to the hidden $2k$-NN graph model in Definition 1 with hidden $2k$-NN graph x^*.

Depending on the applications, we may not be able to reconstruct the hidden $2k$-NN graph x^* perfectly; instead, we may consider correctly estimating all but a small number of edges, which is required to be $o(nk)$, since a $2k$-NN graph contains kn edges. In particular, let $d(x^*, \widehat{x})$ be the Hamming distance $d(x^*, \widehat{x}) = \sum_e \mathbb{1}_{\{x^*_e \neq \widehat{x}_e\}}$.

Definition 3 (Almost Exact Recovery): An estimator $\widehat{x} = \widehat{x}(w) \in \{0,1\}^{\binom{n}{2}}$ achieves almost exact recovery if, as $n \to \infty$,

$$\sup_{x^* \in \mathcal{X}} \mathbb{E}\left[d(x^*, \widehat{x})\right] = o(nk).$$

Instead of using a permutation-based metric such as the Kendall tau distance, we choose the edge Hamming distance d for defining almost exact recovery because for the practical application of discovering strong ties in social networks, there is more value in recovering the edges rather than the permutation. Moreover, the edge sets arise naturally in the analysis of $\widehat{x}_{\rm ML}$: the distribution of the log-likelihood ratio between a $2k$-NN graph x and the truth x^* only depends on the number of edges in x that differ from x^*. It is also worth noting that many computationally efficient algorithms, some of which to be discussed in Section V-A, output an edge set instead of a permutation. One can further project the edge set to a $2k$-NN graph to recover the permutation, but it is not clear whether this can be done in polynomial time.

Intuitively, for a fixed network size n and a fixed number k of nearest neighbors, as the distributions P_n and Q_n get closer, the recovery problem becomes harder. This leads to an immediate question: *From an information-theoretic perspective, computational considerations aside, what are the fundamental limits of recovering the hidden* $2k$*-NN graph?* To answer this question, we derive necessary and sufficient conditions in terms of the model parameters (n, k, P_n, Q_n) under which the hidden $2k$-NN graph can be exactly or almost exactly recovered. These results serve as benchmarks for evaluating practical algorithms and aid us in understanding the performance limits of polynomial-time algorithms.

Specifically, we discover that the following two information measures characterize the sharp thresholds for exact and almost exact recovery, respectively. Define the Rényi divergence of order $1/2$[1]:

$$\alpha_n = -2\log \int \sqrt{\mathrm{d}P_n \mathrm{d}Q_n}; \tag{1}$$

and the Kullback-Leibler divergence:

$$D(P_n \| Q_n) = \int \mathrm{d}P_n \log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}.$$

Under some mild assumptions on P_n and Q_n, we show that the necessary and sufficient conditions are as follows:

- Exact recovery ($2 \le k \le n^{o(1)}$):
$$\liminf_{n\to\infty} \frac{2\alpha_n}{\log n} > 1; \tag{2}$$
- Almost exact recovery $\left(1 \le k \le o\left(\frac{\log n}{\log\log n}\right)\right)$:
$$\liminf_{n\to\infty} \frac{kD(P_n\|Q_n)}{\log n} > 1. \tag{3}$$

The conditions for exact recovery and almost exact recovery are characterized by two different distance measures, namely α_n and $D(P_n\|Q_n)$. This arises from the large deviation analysis for the loss function $d(x^*, \widehat{x})$ in different regimes. In particular, for almost exact recovery, it turns out to be beneficial to obtain a tighter upper bound on the fluctuations of edge weights for those edges not in x^* than those in x^*, because there are many more edges of the former type, thus many more possible ways to arrange them into a different $2k$-NN graph x with $d(x^*, x) = \Omega(kn)$. See Remark 3 for a detailed explanation. For the special case of $k = 1$ (Hamiltonian cycle), the exact recovery condition was shown to be $\liminf_{n\to\infty} \frac{\alpha_n}{\log n} > 1$ [24]. Comparing this with (2) for $k \ge 2$, we find that, somewhat surprisingly, the exact recovery threshold is halved when k increases from 1 to 2, and then stays unchanged as long as k remains $n^{o(1)}$. In contrast, the almost exact recovery threshold decreases inversely proportional to k over the range of $[1, o(\log n/\log\log n)]$. The sharp thresholds of exact recovery for $k \ge n^{\Omega(1)}$ and almost exact recovery for $k = \Omega(\log n/\log\log n)$ remain open.

For the Bernoulli distribution (in other words, unweighted graphs) with $P_n = \mathrm{Bern}(p_n)$ and $Q_n = \mathrm{Bern}(q_n)$, we have the explicit expressions of

$$\alpha_n = -2\log\left(\sqrt{p_n q_n} + \sqrt{(1-p_n)(1-q_n)}\right), \quad \text{and}$$
$$D(P_n\|Q_n) = p_n \log\frac{p_n}{q_n} + (1-p_n)\log\frac{1-p_n}{1-q_n}.$$

As an interesting special case, consider the parametrization

$$p_n = 1 - \epsilon_n + \frac{2\epsilon_n k}{n-1} \quad \text{and} \quad q_n = \frac{2\epsilon_n k}{n-1}, \tag{4}$$

so that the mean number of edges in the observed graph stays at nk for all $\epsilon_n \in [0, 1]$. This can be viewed as an approximate version of the Watts-Strogatz small-world graph, in which we start with a $2k$-NN graph, then rewire each edge with probability ϵ_n independently at random. In this case, our main results (combined with earlier results in [24] for the case $k = 1$) specialize to:

- The sharp threshold for exact recovery is at
$$\liminf_{n\to\infty} \frac{\log\frac{1}{\epsilon_n}}{n} = 1 \quad \text{for } k = 1;$$
$$\liminf_{n\to\infty} \frac{2\log\frac{1}{\epsilon_n}}{n} = 1 \quad \text{for } 2 \le k \le n^{o(1)}. \tag{5}$$
- The sharp threshold for almost exact recovery is at
$$\liminf_{n\to\infty} k(1-\epsilon_n) = 1 \tag{6}$$
for $1 \le k \le o(\log n/\log\log n)$.

In the related work [25], a similar case of Bernoulli distributions has been studied.[2] It is shown in [25] that exact recovery is impossible if $1 - \epsilon_n = o\left(\sqrt{\frac{\log n}{n}} \vee \frac{\log n}{k}\frac{1}{\log\frac{n\log n}{k^2}}\right)$. In particular, this impossibility result requires $\epsilon_n \to 1$, which is highly suboptimal compared to the sharp exact recovery condition (5). It is also shown in [25] that almost exact recovery can be achieved efficiently via thresholding on the number of common neighbors when $1-\epsilon_n = \omega\left((\frac{\log n}{n})^{1/4} \vee (\frac{\log n}{k})^{1/2}\right)$ and via spectral ordering when $1 - \epsilon_n = \omega\left(\frac{n^{3.5}}{k^4}\right)$; these sufficient conditions, however, are very far from being optimal.

The sufficient conditions for both exact and almost exact recovery are established by analyzing the maximum likelihood estimator (MLE). While it is a priori clear that the MLE is optimal for exact recovery,[3] that it also achieves the sharp

[1] It is also related to the so-called Battacharyya distance $B(P_n, Q_n)$ via $\alpha_n = 2\ B(P_n, Q_n)$.

[2] To be precise, the previous work [25] considers Bernoulli distributions under a slightly different parameterization: $p_n = 1 - \epsilon_n + \frac{2\epsilon_n^2\ k}{n-1}$ and $q_n = \frac{2\epsilon_n k}{n-1}$. In addition to exact recovery and approximate recovery, a hypothesis testing problem between the small-world graph and Erdős-Rényi random graph is studied.

[3] Indeed, the MLE minimizes the probability of error under the uniform prior, which is least favorable due to the permutation invariance of the model.

threshold for almost exact recovery is not. The proof of correctness for the MLE relies on tight combinatorial arguments that count the number of $2k$-NN graphs at a given distance to the ground truth. Conversely, we also show that below the sharp thresholds, no estimator can achieve recovery. For exact recovery, this is done by constructing a set of modified $2k$-NN graphs that are hard to distinguish from the truth; for almost exact recovery, we use the mutual information method that compares the information between the observed graph and the ground truth with its rate distortion function.

Finally, we remark that although the MLE achieves the sharp thresholds for both exact and almost exact recovery, it is computationally intractable in the worst case. For the special case of $k = 1$ (Hamiltonian cycle), a linear programming (LP) relaxation of the MLE (namely, the fractional 2-factor LP) is shown to achieve the sharp exact recovery condition [24]. For $k \geq 2$, however, it remains open whether the exact recovery threshold or the almost exact recovery threshold can be achieved efficiently in polynomial time. In Section V we analyze several efficient algorithms and obtain sufficient conditions for them to achieve exact/almost exact recovery; these conditions do not meet with the sharp threshold in general. In the special case of the small-world model, we give a polynomial-time greedy algorithm that attains the threshold for exact recovery.

The paper is organized as follows. In Section II we state and discuss our main results on the sharp recovery thresholds. In Section III and Section IV we prove the results for exact recovery and almost exact recovery. In Section V we analyze the computationally efficient recovery algorithms.

II. Main Results

This section contains our main results on the sharp thresholds for exact and almost exact recovery.

A. Exact Recovery

Recall that an estimator $\widehat{x}$ is said to achieve exact recovery if $\sup_{x^* \in \mathcal{X}} \mathbb{P}\{\widehat{x} \neq x^*\} = o(1)$, where we recall that $\mathcal{X}$ denotes the collection of adjacency vectors of all $2k$-NN graphs on $[n]$. Under a uniform prior over $\mathcal{X}$, the maximum likelihood estimator $\widehat{x}_{\mathrm{ML}}$ minimizes the Bayes risk $\mathbb{P}\{\widehat{x} \neq x^*\}$. Since the uniform prior is the least favorable in permutation invariant models, we have that $\widehat{x}_{\mathrm{ML}}$ is the optimal estimator for exact recovery. Maximizing the likelihood for the hidden $2k$-NN graph problem is equivalent to finding the *max-weighted $2k$-NN subgraph* with weights given by the log likelihood ratios. Specifically, assuming that dP_n/dQ_n is well-defined, for each edge e, let $L_e = \log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}(w_e)$. Then the MLE is the solution to the following combinatorial optimization problem:

$$\widehat{x}_{\mathrm{ML}} = \arg\max_{x \in \mathcal{X}} \langle L, x \rangle . \tag{7}$$

When $k = 1$, (7) reduces to the *max-weighted Hamiltonian cycle* problem. Note that in the Poisson, Gaussian or Bernoulli model where the log likelihood ratio is an affine function of the edge weight, we can simply replace L in (7) by the edge weights w.

Recall that $\alpha_n = -2\log \int \sqrt{dP_n dQ_n}$. We show that if $2 \leq k \leq n^{o(1)}$, then the condition $\liminf_{n\to\infty}(2\alpha_n/\log n) > 1$ is sufficient for $\widehat{x}_{\mathrm{ML}}$ to achieve exact recovery. This condition is also necessary, with the following additional assumption:

Assumption 1 ([24], Assumption 1): Let $X = \log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}(\omega_x)$ for some $\omega_x \sim P_n$ and $Y = \log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}(\omega_y)$ for some $\omega_y \sim Q_n$. Assume that

$$\begin{aligned}&\sup_{\tau\in\mathbb{R}} \left(\log \mathbb{P}\left\{Y \geq \tau\right\} + \log \mathbb{P}\left\{X \leq \tau\right\}\right)\\ &\geq -\left(1+o(1)\right)\alpha_n + o(\log n).\end{aligned}$$

Remark 1 (Generality of Assumption 1): Via Chernoff's inequality, it can be shown that (see [24, page 67] for a derivation)

$$\sup_{\tau\in\mathbb{R}} \left(\log \mathbb{P}\left\{Y \geq \tau\right\} + \log \mathbb{P}\left\{X \leq \tau\right\}\right) \leq -\alpha_n.$$

We rely on this inequality in the large deviation analysis to establish the sufficient condition for $\widehat{x}_{\mathrm{ML}}$ to achieve exact recovery. Assumption 1 essentially ensures that the Chernoff's inequality is asymptotically tight, so we can invert the large deviation analysis to show that the sufficient condition is also almost necessary. It was shown in [24, Lemma 6] that Assumption 1 is fulfilled by a wide class of weight distributions including Poisson, Gaussian and Bernoulli distributions.

The following is our main result regarding exact recovery.

Theorem 1 (Exact Recovery): Let $k \geq 2$.

- Suppose

$$\alpha_n - \frac{1}{2}(\log n + 17\log k) \to +\infty. \tag{8}$$

  Then the MLE (7) achieves exact recovery: $\mathbb{P}\{\widehat{x}_{\mathrm{ML}} \neq x^*\} \to 0$. In particular, this holds if $k = n^{o(1)}$ and

$$\liminf_{n\to\infty} \frac{2\alpha_n}{\log n} > 1.$$

- Conversely, assume that $k < n/12$ and Assumption 1 holds. If exact recovery is possible, then

$$\liminf_{n\to\infty} \frac{2\alpha_n}{\log n} \geq 1.$$

When $k = 1$, as shown in [24] the sharp threshold for exact recovery is $\liminf_{n\to\infty} \frac{\alpha_n}{\log n} > 1$, which is stronger than the condition in Theorem 1 by a factor of 2. In other words, from $k = 1$ to $k \geq 2$ there is a strict decrease in the required level of signal. A simple explanation is that the hidden $2k$-NN graph x^* contains more edges when $k \geq 2$, and the elevated weights on these edges provide extra signal for determining the latent permutation σ^*. However, this extra information ceases to help as k increases from 2 to $n^{o(1)}$, which can be attributed to the following fact: when we swap any pair of adjacent vertices on σ^*, we always get a $2k$-NN graph x which differ from x^* by 4 edges, regardless of how large k is. In fact for all $2 \leq k \leq n^{o(1)}$, the bottleneck for exact recovery is formed by such swaps, resulting in the k-independent necessary condition $\liminf_{n\to\infty} \frac{2\alpha_n}{\log n} \geq 1$ (see Section III-C for details).

Theorem 1 can be applied to a wide range of continuous and discrete edge weight distributions. See the corollary below for the implication of Theorem 1 when the edge weights are distributed as Gaussian, Poisson, or Bernoulli (unweighted).

Corollary 1: For $2 \le k \le n^{o(1)}$, we have

- If $P_n = \mathcal{N}(\mu_n, 1)$ and $Q_n = \mathcal{N}(\nu_n, 1)$, then exact recovery is possible (resp. impossible) if
$$\liminf_{n\to\infty} \frac{(\mu_n - \nu_n)^2}{2\log n} > 1 \text{ (resp. } < 1).$$
- If $P_n = \text{Pois}(\mu_n)$ and $Q_n = \text{Pois}(\nu_n)$, then exact recovery is possible (resp. impossible) if
$$\liminf_{n\to\infty} \frac{2\left(\sqrt{\mu_n} - \sqrt{\nu_n}\right)^2}{\log n} > 1 \text{ (resp. } < 1).$$
- If $P_n = \text{Bern}(p_n)$ and $Q_n = \text{Bern}(q_n)$, then exact recovery is possible (resp. impossible) if
$$\liminf_{n\to\infty} \frac{-4\log\left(\sqrt{p_n q_n} + \sqrt{(1-p_n)(1-q_n)}\right)}{\log n}$$
$$>1 \text{ (resp. } < 1).$$

In particular, under the small-world model, for p_n, q_n parametrized as in (4), exact recovery is possible (resp. impossible) if[4]
$$\liminf_{n\to\infty} \frac{2\log\frac{1}{\epsilon_n}}{\log n} > 1 \text{ (resp. } < 1).$$

B. Almost Exact Recovery

Before presenting our main results for almost exact recovery, we need to introduce some notations. Let $\psi_P(\lambda)$ and $\psi_Q(\lambda)$ be the log moment generating functions of the log likelihood ratio $\log\frac{dP_n}{dQ_n}$ under P_n and Q_n respectively. That is,
$$\begin{aligned}\psi_Q(\lambda) &\triangleq \log \mathbb{E}_{Q_n}\left[\exp\left(\lambda \log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}\right)\right] \\ &= \log \int \mathrm{d}P_n^{\lambda}\mathrm{d}Q_n^{1-\lambda}, \qquad (9)\\ \psi_P(\lambda) &\triangleq \log \mathbb{E}_{P_n}\left[\exp\left(\lambda \log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}\right)\right] \\ &= \log \int \mathrm{d}P_n^{1+\lambda}\mathrm{d}Q_n^{-\lambda} = \psi_Q(\lambda+1). \qquad (10)\end{aligned}$$

Denote the Legendre transforms of ψ_P and ψ_Q as
$$\begin{aligned}E_Q(\tau) &\triangleq \sup_{\lambda\in\mathbb{R}} \lambda\tau - \psi_Q(\lambda),\\ E_P(\tau) &\triangleq \sup_{\lambda\in\mathbb{R}} \lambda\tau - \psi_P(\lambda)\\ &= \sup_{\lambda\in\mathbb{R}} \lambda\tau - \psi_Q(1+\lambda) = E_Q(\tau) - \tau. \qquad (11)\end{aligned}$$

Then E_P and E_Q are convex and monotone functions, such that as τ increases from $-D(Q_n\|P_n)$ to $D(P_n\|Q_n)$, $E_Q(\tau)$ increases from 0 to $D(P_n\|Q_n)$ and $E_P(\tau)$ decreases from $D(Q_n\|P_n)$ to 0. The following assumption postulates a quadratic lower bound of E_P at the boundary:

Assumption 2: There exists an absolute constant $c > 0$, such that for all $\eta \in [0,1]$,
$$E_P((1-\eta)D(P_n\|Q_n)) \ge c\eta^2 D(P_n\|Q_n). \qquad (12)$$

Remark 2 (Generality of Assumption 2): Note that $E_P(\tau)$ is convex with minimum 0 and curvature (second-order derivative) $1/\text{Var}_P(\log(dP_n/dQ_n))$ at $\tau = D(P_n\|Q_n)$. In view of Taylor expansion of $E_P(\tau)$ at $\tau = D(P_n\|Q_n)$, Assumption 2 essentially ensures that $E_P(\tau)$ satisfies a quadratic lower bound with curvature at least $\Omega(1/D(P_n\|Q_n))$, giving us the desired stability in the large-deviation behavior of the log-likelihood ratios when τ is near $D(P_n\|Q_n)$. When the weight distributions are Gaussian, $E_P(\tau)$ is exactly a quadratic function with curvature $1/(2D(P_n\|Q_n))$ at $\tau = D(P_n\|Q_n)$ and thus Assumption 2 holds. It can also be shown that Assumption 2 is satisfied whenever the distribution of $\log(dP_n/dQ_n)$ under P_n is sub-Gaussian with proxy variance $O(D(P_n\|Q_n))$ (see [26], Section 3).

Theorem 2 (Almost Exact Recovery): Suppose Assumption 2 holds. If $k\log k = o(\log n)$ and
$$\liminf_{n\to\infty} \frac{kD(P_n\|Q_n)}{\log n} > 1, \qquad (13)$$
then the MLE (7) achieves almost exact recovery. Conversely, assume that $k = O(\log n)$. If almost exact recovery is possible, then
$$\liminf_{n\to\infty} \frac{kD(P_n\|Q_n)}{\log n} \ge 1. \qquad (14)$$

Theorem 2 should be compared with the exact recovery threshold $\liminf(2\alpha_n/\log n) > 1$ for $2 \le k \le n^{o(1)}$; the latter is always stronger, since
$$\begin{aligned}\alpha_n &= -2\log\int\sqrt{dP_n dQ_n}\\ &= -2\log \mathbb{E}_{P_n}\sqrt{\frac{dQ_n}{dP_n}}\\ &\le -2\mathbb{E}_{P_n}\log\sqrt{\frac{dQ_n}{dP_n}} = D(P_n\|Q_n),\end{aligned}$$
by Jensen's inequality. Unlike exact recovery, the almost exact recovery threshold is inversely proportional to k. Intuitively, this is because almost exact recovery only requires one to distinguish the latent $2k$-NN graph x^* from those $2k$-NN graphs that differ from x^* by $\Omega(kn)$ edges; in contrast, as we will show in Section III-C, the condition for exact recovery arises from eliminating those solutions differing from x^* by only four edges.

Similar to Theorem 1, Theorem 2 is applicable to a wide class of weight distributions. Some examples are discussed in the following corollary.

Corollary 2: For $1 \le k \le o(\log n/\log\log n)$, we have

[4] If k stays bounded as n grows, it can be shown that exact recovery is possible if and only if $\epsilon_n = o(1/\sqrt{n})$. The sufficient direction follows from condition (8). The necessary direction is proven by slightly modifying the proof of Theorem 1: consider an alternative solution that reverses the roles of two adjacent vertices i and $i+1$. The likelihood at the alternative beats (or equals) the likelihood at the truth if both edges $(i-k, i)$ and $(i+1, i+k+1)$ are absent from the observed graph, which occurs with probability $\Theta(\epsilon_n^2)$. Unless $\epsilon_n = o(1/\sqrt{n})$, the alternative beats (or is on par with) the truth for some $i \in [n]$ with non-vanishing probability, deeming exact recovery impossible.

- if $P_n = \mathcal{N}(\mu_n, 1)$ and $Q_n = \mathcal{N}(\nu_n, 1)$, then almost exact recovery is possible (resp. impossible) if
$$\liminf_{n\to\infty} \frac{k(\mu_n - \nu_n)^2}{2\log n} > 1 \text{ (resp. } < 1).$$
- if $P_n = \mathrm{Pois}(\mu_n)$ and $Q_n = \mathrm{Pois}(\nu_n)$, then almost exact recovery is possible (resp. impossible) if
$$\liminf_{n\to\infty} \frac{k\left(\mu_n \log(\mu_n/\nu_n) + \nu_n - \mu_n\right)}{\log n} > 1 \text{ (resp. } < 1).$$
- if $P_n = \mathrm{Bern}(p_n)$ and $Q_n = \mathrm{Bern}(q_n)$, then almost exact recovery is possible (resp. impossible) if
$$\liminf_{n\to\infty} \frac{k\left(p_n \log \frac{p_n}{q_n} + (1-p_n)\log\frac{1-p_n}{1-q_n}\right)}{\log n} > 1 \text{ (resp. } < 1).$$

In particular, under the small-world model, for p_n, q_n parametrized as in (4), almost exact recovery is possible (resp. impossible) if
$$\liminf_{n\to\infty} k(1-\epsilon_n) > 1 \text{ (resp. } < 1).$$

III. Analysis for Exact Recovery

In this section we prove Theorem 1. The proof of the upper bound (sufficient condition for exact recovery) is contained in Section III-A. The upper bound proof involves analyzing the maximum likelihood estimator. The most crucial step of the proof is to obtain a tight upper bound on the number of $2k$-NN graphs that differ from the true one by a given number of edges. The combinatorial arguments leading to this upper bound is the main contribution of this paper, and they are given in Section III-B. In Section III-C we prove the information-theoretic lower bound. The lower bound is proved by constructing a set of alternative $2k$-NN graphs that are difficult to distinguish from the truth x^* unless P_n and Q_n are far enough apart. The analysis closely follows that in [24, Section 6.1] where they consider the case $k = 1$, except that we construct a difference set of alternative solutions when $k \geq 2$.

A. Proof of Correctness of MLE for Exact Recovery

To analyze the MLE, we first introduce the notion of *difference graph*,[5] which encodes the difference between a proposed $2k$-NN graph and the ground truth. Given $x, x^* \in \{0,1\}^{\binom{n}{2}}$, let $G = G(x)$ be a bi-colored simple graph on $[n]$ whose adjacency vector is $x - x^* \in \{0, \pm 1\}^{\binom{n}{2}}$, in the sense that each pair (i,j) is connected by a blue (resp. red) edge if $x_{ij} - x^*_{ij} = 1$ (resp. -1). See Fig. 3 for an example. By definition, red edges in $G(x)$ are true edges in x^* that are missed by the proposed solution x, and blue edges correspond to spurious edges that are absent in the ground truth.

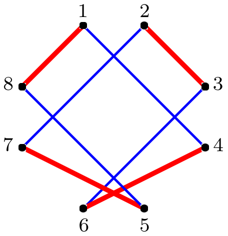


Fig. 3. An example for a difference graph G. Here G is obtained by letting x^* (resp. x) be the $2k$-NN graph in Fig. 1a (resp. 1c), and then taking the difference $x - x^*$. The red (thick) edges stand for edges that in x^* but not x, while the blue (thin) edges are in x but not x^*.

A key property of difference graphs is the following: Since $2k$-NN graphs are $2k$-regular, the difference graph G is *balanced*, in the sense that for each vertex, its red degree (the number of incident red edges) coincides with its blue degree. Consequently, G has equal number of red edges and blue edges, and the number of red (or blue) edges measures the closeness of x to the truth x^*. Denote
$$\begin{aligned}\mathcal{X}_\Delta &= \{x \in \mathcal{X} : d(x, x^*) = 2\Delta\} \\ &= \{x \in \mathcal{X} : G(x) \text{ contains exactly } \Delta \text{ red edges}\}. \qquad (15)\end{aligned}$$

In particular, $\{\mathcal{X}_\Delta : \Delta \geq 0\}$ partitions the feasible set $\mathcal{X}$. The analysis of the MLE relies crucially on bounding the size of $\mathcal{X}_\Delta$. Once we have a tight bound on $|\mathcal{X}_\Delta|$, the proof of the correctness of $\widehat{x}_{\mathrm{ML}}$ follows from the Chernoff bound and the union bound.

Proof of Sufficiency Part of Theorem 1: First partition $\mathcal{X}$ according to the value of Δ:
$$\begin{aligned}&\mathbb{P}\{\exists x \in \mathcal{X} : \langle L, x - x^*\rangle \geq 0\} \\ &\leq \sum_{\Delta \geq 1} \mathbb{P}\{\exists x \in \mathcal{X}_\Delta : \langle L, x - x^*\rangle \geq 0\}. \qquad (16)\end{aligned}$$

Recall that $L_e = \log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}(w_e)$. Hence for each $x \in \mathcal{X}_\Delta$, the law of $\langle L, x - x^*\rangle$ only depends on Δ, which can be represented as follows:
$$\langle L, x - x^*\rangle \overset{d}{=} \sum_{i \leq \Delta} Y_i - \sum_{i \leq \Delta} X_i,$$
where $X_1, \ldots, X_\Delta$ are i.i.d. copies of $\log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}$ under P_n, $Y_1, \ldots, Y_\Delta$ are i.i.d. copies of $\log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}$ under Q_n, and $\overset{d}{=}$ denotes equality in distribution. Applying the Chernoff bound yields
$$\begin{aligned}&\mathbb{P}\left\{\sum_{i \leq \Delta} Y_i - \sum_{i \leq \Delta} X_i \geq 0\right\} \\ &\leq \inf_{\lambda > 0}\{\exp(\Delta(\psi_Q(\lambda) + \psi_P(-\lambda)))\}, \qquad (17)\end{aligned}$$
where ψ_P and ψ_Q are the log MGFs of $\log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}$ under P_n and Q_n, as defined in (9), (10). In particular, the Rényi divergence in (3) is given by
$$\alpha_n = -2\psi_Q\left(\frac{1}{2}\right) = -2\psi_P\left(-\frac{1}{2}\right). \qquad (18)$$

[5] Here the notion of difference graph originates from simply subtracting one adjacency matrix from another and using colors (blue or red) to encode the plus or minus. This is not to be confused with the definition in [27] for a different context.

Choosing $\lambda = 1/2$ in (17) yields

$$\begin{aligned}&\mathbb{P}\left\{\langle L, x - x^*\rangle \geq 0\right\}\\ =&\mathbb{P}\left\{\sum_{i\leq\Delta} Y_i - \sum_{i\leq\Delta} X_i \geq 0\right\}\\ \leq&\exp\left(2\Delta\psi_Q\left(\frac{1}{2}\right)\right) = \exp\left(-\alpha_n\Delta\right). \end{aligned} \tag{19}$$

A union bound yields

$$\begin{aligned}&\mathbb{P}\left\{\exists x \in \mathcal{X}_\Delta : \langle L, x - x^*\rangle \geq 0\right\}\\ \leq& \sum_{x\in\mathcal{X}_\Delta} \mathbb{P}\left\{\langle L, x - x^*\rangle \geq 0\right\}\\ \leq& |\mathcal{X}_\Delta| \exp\left(-\alpha_n\Delta\right). \end{aligned} \tag{20}$$

The most critical component of our analysis is to derive a tight upper bound on $|\mathcal{X}_\Delta|$. We will use the bound (21) stated in Lemma 1 below and proved in Section III-B. Assuming Lemma 1, we arrive at

$$\mathbb{P}\left\{\exists x \in \mathcal{X}_\Delta : \langle L, x - x^*\rangle \geq 0\right\} \leq 2\exp\left(-\Delta\kappa_n\right),$$

where $\kappa_n \triangleq \alpha_n - \frac{\log(Ck^{17}n)}{2} \to \infty$ by assumption. Finally, from (16),

$$\begin{aligned}&\mathbb{P}\left\{\exists x \in \mathcal{X} : \langle L, x - x^*\rangle \geq 0\right\}\\ \leq& \sum_{\Delta\geq 1} 2\exp\left(-\Delta\kappa_n\right)\\ =&\frac{2\exp(-2\kappa_n)}{1-\exp(-\kappa_n)} \xrightarrow{n\to\infty} 0.\end{aligned}$$

In other words, $\mathbb{P}\{\widehat{x}_{\mathrm{ML}} = x^*\} \to 1$ as $n \to \infty$. □

Lemma 1: There exists an absolute constant C such that for any $\Delta \geq 0$ and any $2 \leq k \leq n$

$$|\mathcal{X}_\Delta| \leq 2\left(Ck^{17}n\right)^{\Delta/2}. \tag{21}$$

The proof of Lemma 1 is contained in Section III-B. It is the most involved component of our analysis, and the main contribution of this paper. To provide some intuition on the bound (21), let us first prove a simple bound

$$|\mathcal{X}_\Delta| \leq (4kn)^\Delta. \tag{22}$$

This simple bound was proved in [24, Sec. 4.2] for $k = 1$ (Hamiltonian cycles), but it holds for general $k \geq 1$ via similar arguments. Substituting (22) into (26) immediately yields that $\mathbb{P}\left\{\widehat{x}_{\mathrm{ML}} \neq x^*\right\} \to 0$, provided that $\alpha_n - \log(nk) \to +\infty$, which falls short of the desired sufficient condition (8) by roughly a factor of 2 when $k \geq 2$.

The simple bound (22) is proved as follows. For each $x \in \mathcal{X}_\Delta$, suppose its difference graph G consists of m connected components $G_1, \ldots, G_m$. Then each connected component is also a balanced bi-colored graph. Let $\Delta_i \geq 1$ denote the number of red edges in G_i. There are at most 2^Δ configurations for the sequence $(\Delta_1, \ldots, \Delta_m)$ since $\sum_{i\leq m}\Delta_i = \Delta$. From [24, Lemma 1], every connected balanced bi-colored graph has an *alternating Eulerian circuit*, *i.e.* a circuit with colors alternating between red and blue that passes through every edge exactly once. To bound the total number of configurations for a connected component G_i with Δ_i red edges, it suffices to count the number of such alternating Eulerian circuits, which is upper bounded by the number of length-$2\Delta_i$ path $(v_0, v_1, \ldots, v_{2\Delta_i - 1})$, such that (v_i, v_{i+1}) is a red edge if i is even, and a blue edge if i is odd. To complete the circuit, $(v_{2\Delta_i-1}, v_0)$ must be a blue edge.

We now sequentially enumerate v_0 to $v_{2\Delta_i-1}$: given v_0, which takes n values, there are only $2k$ possibilities for v_1, because (v_0, v_1) is a red edge so that v_1 must belong to the neighborhood of v_0 in the true $2k$-regular graph x^*. Overall, we conclude that the path $(v_0, \ldots, v_{2\Delta_i-1})$ can take at most $(2kn)^{\Delta_i}$ possible values. Summing over the connected components, we have

$$\begin{aligned}|\mathcal{X}_\Delta| \leq& \sum_{(\Delta_1,\ldots,\Delta_m):\sum\Delta_i=\Delta}\left(\prod_{i\leq m}(2kn)^{\Delta_i}\right)\\ \leq& 2^\Delta(2kn)^\Delta = (4kn)^\Delta.\end{aligned}$$

It turns out this bound is only tight for $k = 1$. For $k \geq 2$, Lemma 1 gives a much better bound on the cardinality of $\mathcal{X}_\Delta$. In comparison with the simple bound (22), Lemma 1 improves the dependency on n from n^Δ to $n^{\Delta/2}$. We have already seen that the red edges play an important role in the proof of (22), as the number of red edges is much lower than that of blue edges. The tight bound (21) is obtained by further exploiting the structural properties of red edges in the difference graph G. In particular, we find that for each red edge in G, there is at least another red edge "close" to it, which allows us to count red edges in groups and further reduce the number of ways they can appear in a difference graph. The precise notion of closeness will be given in Section III-B, but let us illustrate with a simple example.

Example 1: Recall that each $2k$-NN graph x can be identified with a permutation $(\sigma(1), \sigma(2), \ldots, \sigma(n))$. By connecting adjacent nodes on σ and connecting $\sigma(n)$ with $\sigma(1)$, σ determines a Hamiltonian cycle, from which one can connect pairs of vertices whose distance is at most k to construct a $2k$-NN graph. Suppose the true $2k$-NN graph x^* is identified with the identity permutation $\sigma^* = (1, 2, \ldots, n)$. Consider the alternative graph x identified with a permutation that traverses part of the vertices in the opposite direction, *i.e.* $\sigma = (1, 2, \ldots, i, j, j-1, \ldots, i+1, j+1, j+2, \ldots, n)$ for some i, j that are far apart (see Fig. 4a). The corresponding difference graphs in the $k = 1$ and $k = 2$ cases are illustrated in Fig. 4b, 4c respectively.

The crucial observation is that when $k \geq 2$, there is more structure in the set of red edges in the sense that red edges do not appear in isolation. For example in Fig. 4c, the indices of the three red edges $(i, i+1)$, $(i-1, i+1)$ and $(i, i+2)$ are all close to each other; in particular, this triple can only take n values in total. We find that when $k \geq 2$, this observation holds in greater generality. As a result, each red edge can help determine at least one other red edge, allowing us to enumerate the red edges in bundles. This is the main reason why when upper bounding $|\mathcal{X}_\Delta|$, we can reduce the exponent on n from Δ to $\Delta/2$. This structural property, however, is specific to $k \geq 2$:

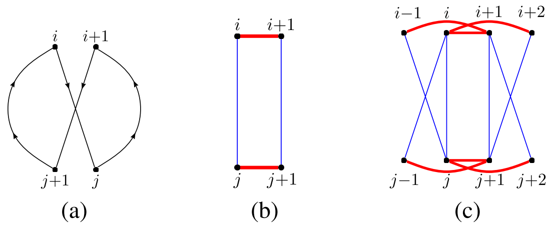


Fig. 4. From left to right, (a): Hamiltonian cycle associated with the permutation $\sigma = (\ldots, i, j, j-1, \ldots, i+1, j+1, j+2, \ldots)$; (b): the difference graph $G(x)$ when $k = 1$, where x is the $2k$-NN graph identified by σ; (c): the difference graph $G(x)$ when $k = 2$.

as shown by Fig. 4b ($\Delta = 2$), the simple bound (22) is tight for $k = 1$.

Let us also point out that the exponent $\Delta/2$ in (21) is tight for $k \ge 2$. It is easy to see that when the nodes $i, i+1$ in the permutation σ^* are swapped, a difference graph $G(x)$ is formed with $\Delta = 2$ red edges (see Section III-C and Fig. 9 for a more extensive discussion of this example). Taking $i = 1, \ldots, n$ yields n distinct difference graphs that are all members of $\mathcal{X}_2$, meaning the size of $\mathcal{X}_\Delta$ is no smaller than $n^{\Delta/2}$, at least when $\Delta = 2$.

B. Proof of Lemma 1: Counting Difference Graphs

To prove Lemma 1, we begin with some notations. For a $2k$-NN graph x, let $E_{\text{red}}(x)$ (resp. $E_{\text{blue}}(x)$) be the set of red (resp. blue) edges in $G(x)$. The proof strategy is as follows: First, in Lemma 3 we count

$$\mathcal{E}_{\text{red}}(\Delta) = \{E_{\text{red}}(x) : x \in \mathcal{X}_\Delta\}.$$

Then for each $E_{\text{red}} \in \mathcal{E}_{\text{red}}(\Delta)$, Lemma 4 enumerates

$$\mathcal{X}(E_{\text{red}}) = \{x \in \mathcal{X}_\Delta : E_{\text{red}}(x) = E_{\text{red}}\}.$$

which contains all sets of blue edges that are compatible with E_{red}. This completely specifies the difference graph $G(x)$, and hence the $2k$-NN graph x.

For a given $2k$-NN graph x associated with the permutation σ, let $\mathcal{N}_x(i)$ denote the set of neighbors of i in x. Let $d_x(i,j) = \min\{|\sigma^{-1}(i) - \sigma^{-1}(j)|, n - |\sigma^{-1}(i) - \sigma^{-1}(j)|\}$, which is the distance between i and j on the Hamiltonian cycle defined by σ. It is easy to check that d_x is a well-defined metric on $[n]$. For the hidden $2k$-NN graph x^*, define $\mathcal{N}_{x^*}(\cdot)$ and $d_{x^*}(\cdot,\cdot)$ accordingly.

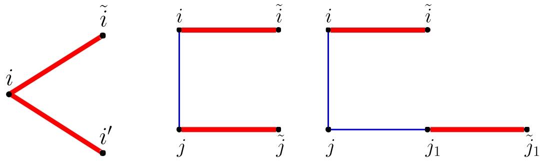
Definition 4: In the $2k$-NN graph x^*, define the distance between two edges $e = (i, \widetilde{i})$ and $f = (j, \widetilde{j})$ as

$$d(e,f) = \min\{d_{x^*}(i,j), d_{x^*}(i,\widetilde{j}), d_{x^*}(\widetilde{i},j), d_{x^*}(\widetilde{i},\widetilde{j})\}.$$

We say e and f are *nearby* if $d(e,f) \le 2k$.

Since a $2k$-NN graph has a total of kn edges, the cardinality of $\mathcal{E}_{\text{red}}(\Delta)$ is at most $\binom{kn}{\Delta}$. The following lemma provides additional structural information for elements of $\mathcal{E}_{\text{red}}(\Delta)$ that allows us to improve this trivial bound.

Lemma 2: Suppose $k \ge 2$. For each red edge e in the difference graph G, there exists a nearby red edge f in G that is distinct from e.



(a) Case 1. (b) Case 2(a). (c) Case 2(b).

Fig. 5. Three cases considered in the proof of Lemma 2.

Lemma 2 allows us to enumerate the red edges in groups, leading to the following Lemma 3 which gives an upper bound for the size of $\mathcal{E}_{\text{red}}(\Delta)$.

Lemma 3: Suppose $k \ge 2$. Then

$$|\mathcal{E}_{\text{red}}(\Delta)| \le (96k^2)^\Delta \binom{kn}{\lfloor \Delta/2 \rfloor}.$$

With the enumeration of the red edge sets complete, the following lemma controls the number of $2k$-NN graphs that are compatible with a fixed set of red edges. A key observation is that the bound does not depend on n.

Lemma 4: Suppose $k \ge 2$. For each $E_{\text{red}} \in \mathcal{E}_{\text{red}}(\Delta)$,

$$|\mathcal{X}(E_{\text{red}})| \le 2(32k^3)^{2\Delta} \Delta^{\Delta/k}. \tag{23}$$

The desired bound in Lemma 1 immediately follows from combining Lemma 3 and Lemma 4:

$$\begin{aligned} |\mathcal{X}_\Delta| &= \left| \bigcup_{E_{\text{red}} \in \mathcal{E}_{\text{red}}(\Delta)} \mathcal{X}(E_{\text{red}}) \right| \\ &\le (96k^2)^\Delta \binom{kn}{\lfloor \Delta/2 \rfloor} \cdot 2(32k^3)^{2\Delta} \Delta^{\Delta/k} \\ &\le 2\left(Ck^{17}n\right)^{\Delta/2} \end{aligned} \tag{24}$$

for a universal constant $C > 0$, where the last inequality follows from $\binom{a}{b} \le (ea/b)^b$ and $k \ge 2$.

Next we prove Lemmas 2, 3, 4.

Proof of Lemma 2: We divide the proof into two cases according to the degree of one of the endpoints of $e = (i, \widetilde{i})$, say i, in the difference graph.

1) The degree of i is strictly larger than 2. Then by balancedness the number of red edges attached to i is at least 2. Other than $(i, \widetilde{i})$, there must exist at least one other red edge (i, i'). By definition

$$d((i,\widetilde{i}), (i,i')) \le d_{x^*}(i,i) = 0 < 2k.$$

That is, (i, i') and $(i, \widetilde{i})$ are nearby. See Fig. 5a.

2) The degree of i is equal to 2. Then i is only attached to one red edge and one blue edge in G. Denote the blue edge as (i, j). Since the only red edge attached to i is $(i, \widetilde{i})$, we have that in the proposed solution x, the vertex i is connected to all its old neighbors in x^* except $\widetilde{i}$. Thus we get that $\mathcal{N}_x(i) = \mathcal{N}_{x^*}(i) \cup \{j\} \backslash \{\widetilde{i}\}$. As a result, when $k \ge 2$, out of the two vertices j_1, j_2 that are right next to j in the x cycle ($d_x(j, j_1) = d_x(j, j_2) = 1$), at least one of them is an old neighbor of i. WLOG say $j_1 \in \mathcal{N}_{x^*}(i)$. Consider these cases:

a) $d_{x^*}(j, j_1) \le k$. By triangle inequality $d_{x^*}(j,i) \le d_{x^*}(j, j_1) + d_{x^*}(i, j_1) \le 2k$. Because G is a balanced graph, there is at least one red edge $(j, \widetilde{j})$ attached to j, and

$$d((i,\widetilde{i}),(j,\widetilde{j})) \le d_{x^*}(j,i) \le 2k.$$

In other words, $(j, \widetilde{j})$ and $(i, \widetilde{i})$ are nearby. See Fig. 5b.

b) $d_{x^*}(j, j_1) > k$. In this case (j, j_1) appears in the difference graph as a blue edge. Therefore j_1 is one of the vertices in G and attached to at least one red edge $(j_1, \widetilde{j}_1)$. Recall that $j_1 \in \mathcal{N}_{x^*}(i)$. Therefore

$$d((i,\widetilde{i}),(j_1,\widetilde{j}_1)) \le d_{x^*}(i,j_1) \le k.$$

In other words, $(j_1, \widetilde{j}_1)$ and $(i, \widetilde{i})$ are nearby. See Fig. 5c.

□

Proof of Lemma 3: To each member E_{red} of $\mathcal{E}_{\mathrm{red}}(\Delta)$, we associate an undirected graph $\widetilde{G}(E_{\mathrm{red}})$ that takes the red edge set E_{red} as its vertex set. Two vertices in $\widetilde{G}(E_{\mathrm{red}})$, or equivalently, two members e, f of E_{red} are connected in $\widetilde{G}(E_{\mathrm{red}})$ if and only if e and f are nearby per Definition 4. It suffices to enumerate all E_{red} for which $\widetilde{G}(E_{\mathrm{red}})$ is compliant with the structural property enforced by Lemma 2. Our enumeration scheme is as follows:

1) Fix $m \in [\Delta]$ to be the number of connected components of $\widetilde{G}(E_{\mathrm{red}})$. Select $\{e_1, \dots, e_m\}$ from the edge set of x^*. Since x^* is a $2k$-NN graph with kn edges, there are $\binom{kn}{m}$ ways to select this set.
2) Let $\Delta_1, \dots, \Delta_m$ be the sizes of the connected components $\mathcal{C}_1, \dots, \mathcal{C}_m$ of $\widetilde{G}(E_{\mathrm{red}})$. Since $\Delta_i \ge 1$ and $\sum \Delta_i = \Delta$, the total number of such (Δ_i) sequences is $\binom{\Delta-1}{m-1}$, as each sequence can be viewed as the result of replacing $m-1$ of the "$+$" symbols with "," in the expression $\Delta = 1 + 1 + \ldots + 1 + 1$.
3) For each $\mathcal{C}_i$, there is at least one spanning tree T_i. Since $\mathcal{C}_i$ and T_i share the same vertex set, it suffices to enumerate T_i. First enumerate the isomorphism class of T_i, that is, count the total number of unlabeled rooted trees with Δ_i vertices. From [28], there are at most 3^{Δ_i} such unlabeled trees.
4) For $i = 1, \dots, m$, let e_i be the root of T_i. Enumerate the ways to label the rest of tree T_i. To start, label the vertices on the first layer of T_i, that is, the children of e_i. A red edge f being a child of e_i on T_i means f and e_i are nearby, limiting the number of labels to at most $16k^2$. To see why, note that at least one endpoint of f is at most $2k$ away from one of the endpoints of e_i, measured in terms of the distance d_{x^*}. No more than $8k$ vertices fit this description. The other endpoint of f can then only choose from $2k$ vertices because f is in the edge set of x^*.

   The remaining layers of T_i can be labeled similarly, with at most $16k^2$ possibilities to label each vertex. In total there are at most $(16k^2)^{\Delta_i - 1}$ to label T_i.

This enumeration scheme accounts for all members of $\mathcal{E}_{\mathrm{red}}(\Delta)$. By Lemma 2, $\widetilde{G}$ does not contain singletons, *i.e.* $\Delta_i \ge 2$ for all i. Thus $m \le \lfloor \Delta/2 \rfloor$, and

$$\begin{aligned}
&|\mathcal{E}_{\mathrm{red}}(\Delta)| \\
&\le \sum_{m \le \lfloor \Delta/2 \rfloor} \binom{kn}{m} \binom{\Delta-1}{m-1} \prod_{i \le m} 3^{\Delta_i} (16k^2)^{\Delta_i - 1} \\
&\le \binom{kn}{\lfloor \Delta/2 \rfloor} 2^{\Delta-1} 3^{\Delta} (16k^2)^{\Delta} \le (96k^2)^{\Delta} \binom{kn}{\lfloor \Delta/2 \rfloor}.
\end{aligned}$$

□

Before proving Lemma 4, notice the factor $\Delta^{\Delta/k}$ in (23). This factor turns out to be crucial. To appreciate this subtlety, let us first derive a simple bound $|\mathcal{X}(E_{\mathrm{red}})| \le 4^{\Delta}\Delta!$. Note that there is a one-to-one correspondence between $2k$-NN graph x and the difference graph $G(x)$. Hence, it is equivalent to enumerating all possible difference graphs with the given set of red edges. Following the similar alternating Eulerian-circuit based argument for proving (22), we can get that

$$|\mathcal{X}(E_{\mathrm{red}})| \le \sum_{(\Delta_1, \dots, \Delta_m): \sum \Delta_i = \Delta} \left(2^{\Delta}\Delta!\right) \le 4^{\Delta}\Delta!,$$

where $2^{\Delta}\Delta!$ counts all the possible orderings of oriented red edges.[6] However, this simple bound falls short of proving the desired (24), as $\binom{kn}{\lfloor \Delta/2 \rfloor}\Delta! \ge (ckn)^{\Delta/2}\Delta^{\Delta/2}$ for a universal constant $c > 0$.

Lemma 4 improves over this simple bound by further exploiting the structure in the difference graph. In particular, Lemma 4 counts $|\mathcal{X}_{\Delta}|$ by enumerating all Hamiltonian cycles $(\sigma(1), \sigma(2), \dots, \sigma(n), \sigma(1))$ such that $E_{\mathrm{red}}(x(\sigma)) = E_{\mathrm{red}}$. A key idea is to sequentially determine each neighborhood $\mathcal{N}_x(\sigma(i))$ starting from $i = 1$. Suppose $\mathcal{N}_x(\sigma(j))$ has been determined for all $1 \le j \le i$ and we are about to specify $\mathcal{N}_x(\sigma(i+1))$, which reduces to enumerating $\sigma(i+k+1)$. Roughly, there are three cases to consider:

1) $\sigma(i+1)$ is not in the difference graph $G(x)$. In this case, $\mathcal{N}_x(\sigma(i+1)) = \mathcal{N}_{x^*}(\sigma(i+1))$ and thus $\sigma(i+k+1)$ has already been fixed.
2) $\sigma(i+1)$ is in the difference graph $G(x)$ and $\sigma(i+k+1)$ has fewer than k blue edges connecting to $\{\sigma(j) : i+1 \le j \le i+k\}$. In this case, at least one of $\{\sigma(j) : i+1 \le j \le i+k\}$ must be a true neighbor of $\sigma(i+k+1)$, which implies that $\sigma(i+k+1)$ has at most $2k^2$ possibilities.
3) $\sigma(i+1)$ is in the difference graph $G(x)$ and $\sigma(i+k+1)$ has k blue edges connecting to $\{\sigma(j) : i+1 \le j \le i+k\}$. In this case, $\sigma(i+k+1)$ has at most 2Δ possibilities, because the difference graph has at most 2Δ different vertices.

Note that whenever the last case occurs, it gives rise to k new blue edges. Since the total number of blue edges is Δ, the last

[6] To be more precise, to count the difference graphs with the given set of Δ red edges, it suffices to enumerate all possible edge-disjoint unions of alternating Eulerian circuits with the given set of Δ red edges. To this end, for a fixed m and sequence $(\Delta_1, \dots, \Delta_m)$ such that $\sum \Delta_i = \Delta$, we enumerate all possible edge-disjoint unions of m alternating Eulerian circuits consisting of $(\Delta_1, \dots, \Delta_m)$ red edges, respectively. First, determine an ordering of oriented red edges, which has $2^{\Delta}\Delta!$ possibilities. Then we connect the first Δ_1 oriented red edges by blue edges to form the first alternating Eulerian circuit, the next Δ_2 oriented red edges by blue edges to form the second alternating Eulerian circuit, and proceed similarly to form the rest of alternating Eulerian circuits.

case can occur at most Δ/k times, which immediately yields the desired factor $\Delta^{\Delta/k}$ in (23).

Next, building upon this intuition, we present the rigorous proof of Lemma 4.

Proof of Lemma 4: For a given permutation σ, let $x(\sigma)$ denote the corresponding $2k$-NN graph. Hereafter the dependence on σ is suppressed whenever it is clear from the context. These are some useful facts about the difference graph G:

1) Let V denote the collection of the endpoints of edges in $E_{\rm red}$. Then the difference graph $G = (V(G), E(G))$ is given by $V(G) = V$. Since $|E_{\rm red}| = \Delta$, $|V(G)| \le 2\Delta$.
2) For each $2k$-NN graph, σ is determined up to cyclic shifts and a reversals.
3) For two vertices $j \neq j'$, $|\mathcal{N}_x(j) \cap \mathcal{N}_x(j')|$ is completely determined by $d_x(j, j')$ via the formula below.

$$|\mathcal{N}_x(j) \cap \mathcal{N}_x(j')| = \begin{cases} 2k - 1 - d_x(j, j') & \text{if } d_x(j, j') \le k; \\ 2k + 1 - d_x(j, j') & \text{if } k < d_x(j, j') \le 2k; \\ 0 & \text{if } d_x(j, j') > 2k. \end{cases}$$

By fact 2, it suffices to enumerate all σ such that $E_{\rm red}(x(\sigma)) = E_{\rm red}$ and $\sigma(1) = 1$. WLOG assume that the ground truth x^*, $\sigma^*(i) = i$. The following is the outline of our enumeration scheme:

1) Enumerate all possibilities for the set $\mathcal{N}_x(1) = \{\sigma(n - k + 1), \ldots, \sigma(n), \sigma(2), \ldots, \sigma(k + 1)\}$.
2) With $\mathcal{N}_x(1)$ determined, enumerate all possibilities for the (ordered) sequence $(\sigma(n - k + 1), \ldots, \sigma(n), \sigma(2), \ldots, \sigma(k + 1))$.
3) For i from 1 to $n - 2k - 1$, enumerate $\sigma(i + k + 1)$ sequentially, assuming at step i that σ were determined from $\sigma(n - k + 1)$ up to $\sigma(i + k)$.

Now we give the details on how cardinality bounds are obtained for each step of the enumeration scheme.

Step 1: Decompose $\mathcal{N}_x(1)$ according to the set of true neighbors and false neighbors. The set of true neighbors $\mathcal{N}_x(1) \cap \mathcal{N}_{x^*}(1)$ is determined by the set of red edges in G. Indeed, this set consists of all members $i \in \mathcal{N}_{x^*}(1)$ for which $(1, i) \notin E_{\rm red}$.

The set $\mathcal{N}_x(1) \backslash \mathcal{N}_{x^*}(1)$ cannot be read directly from the set of red edges. However we know all members of this set must be connected to 1 via a blue edge. Hence $\mathcal{N}_x(1) \backslash \mathcal{N}_{x^*}(1)$ is a subset of $V(G)$, the vertex set of G. Since $V(G)$ is determined by $E_{\rm red}$ and $|V(G)| \le 2\Delta$, the number of possibilities for $\mathcal{N}_x(1) \backslash \mathcal{N}_{x^*}(1)$ does not exceed the number of subsets of $V(G)$, which is at most $2^{2\Delta}$.

Step 2: With the set $\mathcal{N}_x(1)$ determined, we next enumerate all ways to place the elements in $\mathcal{N}_x(1)$ on the Hamiltonian cycle specified by σ. That is, we specify the sequence $(\sigma(n - k + 1), \ldots, \sigma(n), \sigma(2), \ldots, \sigma(k + 1))$, or equivalently, specify $\sigma^{-1}(j)$ for all $j \in \mathcal{N}_x(1)$.

We start with $\mathcal{N}_x(1) \cap V(G)^c$. A vertex in $V(G)^c$ is one whose neighborhood is preserved, *i.e.*, $V(G)^c = \{j \in [n] : \mathcal{N}_x(j) = \mathcal{N}_{x^*}(j)\}$. For each $j \in \mathcal{N}_x(1) \cap V(G)^c$, we have by fact 3,

$$d_x(1, j) = 2k - 1 - |\mathcal{N}_x(1) \cap \mathcal{N}_x(j)|.$$

Since $d_x(1, j)$ is completely determined by $\mathcal{N}_x(1)$, there are only two possibilities for $\sigma^{-1}(j)$.

Furthermore, for every pair $j, j' \in \mathcal{N}_x(1) \cap V(G)^c$, again by fact 3,

$$d_x(j, j') = \begin{cases} 2k - 1 - |\mathcal{N}_x(j) \cap \mathcal{N}_x(j')| & \text{if } j' \in \mathcal{N}_x(j); \\ 2k + 1 - |\mathcal{N}_x(j) \cap \mathcal{N}_x(j')| & \text{otherwise.} \end{cases}$$

So $d_x(j, j')$ is also determined by $\mathcal{N}_x(j)$ and $\mathcal{N}_x(j')$. Therefore the entire sequence $(\sigma^{-1}(j) : j \in \mathcal{N}_x(1) \cap V(G)^c)$ is determined up to a global reflection around 1.

Next we handle all $j \in \mathcal{N}_x(1) \cap V(G)$. Note that $\sigma^{-1}(j) \in \{n - k + 1, \ldots, n, 2, \ldots, k + 1\}$ because $j \in \mathcal{N}_x(1)$. Among those $2k$ possible values, some are already taken by $\{\sigma^{-1}(j) : j \in \mathcal{N}_x(1) \cap V(G)^c\}$, leaving $|\mathcal{N}_x(1) \cap V(G)|$ values to which all $j \in \mathcal{N}_x(1) \cap V(G)$ are to be assigned. The number of possible assignments is bounded by $|\mathcal{N}_x(1) \cap V(G)|!$. Since $|\mathcal{N}_x(1) \cap V(G)| \le \min\{2k, 2\Delta\}$, $\big|\mathcal{N}_x(1) \cap V(G)\big|! \le (2k)^{2\Delta}$.

Overall, the number of possible choices of the ordered tuple $(\sigma(n - k + 1), \ldots, \sigma(n), \sigma(2), \ldots, \sigma(k + 1))$ is at most

$$2 \cdot 2^{2\Delta} \cdot (2k)^{2\Delta} = 2(4k)^{2\Delta}.$$

Step 3: In the previous two steps the values of $(\sigma(n - k + 1), \ldots, \sigma(k + 1))$ have been determined, and so are the blue edges between members of $\{\sigma(n - k + 1), \ldots, \sigma(k + 1)\}$. That is because $(\sigma(j), \sigma(j'))$ is a blue edge if and only if $d_{x^*}(j, j') \le k$ and $d_{x^*}(\sigma(j), \sigma(j')) > k$. Denote this set of blue edges as $E^{(1)}_{\rm blue}$, which can be empty. Recall that, by balancedness, the total number of blue edges in G is Δ. If $|E^{(1)}_{\rm blue}|$ is already Δ, then the enumeration scheme is complete because x is completely specified by the difference graph. Otherwise we determine the value of $\sigma(i + k + 1)$ sequentially, starting from $i = 1$. At the i'th iteration, we first assign the value of $\sigma(i + k + 1)$, the only remaining undetermined neighbor of $\sigma(i + 1)$ in x. Then we update the set of blue edges based on the value of $\sigma(i + k + 1)$: let $E^{(i+1)}_{\rm blue} = E^{(i)}_{\rm blue} \cup E^{(i)}_{\rm update}$, where

$$E^{(i)}_{\rm update} \triangleq \{(\sigma(j), \sigma(i + k + 1)) : d_{x^*}(\sigma(j), \sigma(i + k + 1)) > k, j = i + 1, \ldots, i + k\}.$$

In other words, $E^{(i)}_{\rm blue}$ stands for the set of blue edges that have been determined after the $i - 1$'th iteration. We repeat this process until all Δ blue edges are determined, *i.e.*, $|E^{(i)}_{\rm blue}| = \Delta$.

At the start of the i'th iteration, all of $\sigma(n - k + 1), \ldots, \sigma(i + k)$ have been determined. Unless $|E^{(i)}_{\rm blue}| = \Delta$, specify $\sigma(i + k + 1)$ as follows.

Consider three cases according to the red degree of $\sigma(i + 1)$, *i.e.*, the number of red edges incident to $\sigma(i + 1)$ in $E_{\rm red}$. Note that after the value of $\sigma(i + k + 1)$ is assigned, $\mathcal{N}_x(\sigma(i + 1))$ would be completely specified and all blue edges in G that are incident to $\sigma(i + 1)$ would be determined. Therefore exactly one of the following three cases must occur (for otherwise there would be more red edges than blue edges incident to $\sigma(i + 1)$ in G, contradicting the balancedness of G):

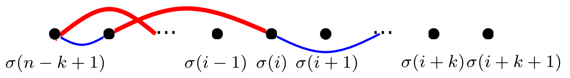


Fig. 6. Vertices arranged by their order on the Hamiltonian cycle corresponding to σ. At the ith iteration, the values of $\sigma(n-k+1)$ to $\sigma(i+k)$ are determined. The figure shows an example of case 1: the vertex $\sigma(i+1)$ is not attached to any red edges.

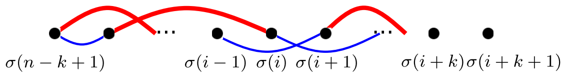


Fig. 7. Case 2: $\sigma(i+1)$ is attached to some red edge(s) and is already balanced at step i. In the figure the red degree and blue degree of $\sigma(i+1)$ are both 1, thus $(\sigma(i+1), \sigma(i+k+1))$ cannot be a blue edge in G.

1) (Fig. 6) The red degree of $\sigma(i+1)$ is zero, meaning that $\mathcal{N}_x(\sigma(i+1)) = \mathcal{N}_{x^*}(\sigma(i+1))$. We claim that the value of $\sigma(i+k+1)$ has already been uniquely determined. Indeed, at the ith iteration, all but one members of $\mathcal{N}_x(\sigma(i+1))$ are determined, and $\sigma(i+k+1)$ has to be the true neighbor of $\sigma(i+1)$ that is not in $\{\sigma(i-k+1), \ldots, \sigma(i), \sigma(i+2), \ldots, \sigma(i+k)\}$.
2) (Fig. 7) The red degree of $\sigma(i+1)$ is nonzero and equals the number of blue edges in $E^{(i)}_{\text{blue}}$ incident to $\sigma(i+1)$. We claim that the number of possible values of $\sigma(i+k+1)$ is at most $2k$. In this case by balancedness all blue edges incident to $\sigma(i+1)$ are contained in $E^{(i)}_{\text{blue}}$ and therefore the edge $(\sigma(i+1), \sigma(i+k+1))$ does not appear in the difference graph G. That implies $\sigma(i+k+1)$ is connected to $\sigma(i+1)$ in x^*, limiting the number of choices for $\sigma(i+k+1)$ to at most $2k$.
3) (Fig. 8) The red degree of $\sigma(i+1)$ is nonzero and equals one plus the number of blue edges in $E^{(i)}_{\text{blue}}$ incident to $\sigma(i+1)$. By balancedness, $(\sigma(i+1), \sigma(i+k+1))$ is a blue edge in G. In this case, either $1 \le |E^{(i)}_{\text{update}}| < k$ or $|E^{(i)}_{\text{update}}| = k$. Suppose this is the t'th time case 3 happens. Let ξ_t encode which of the two possibilities occurs and specify the value $\sigma(i+k+1)$ as follows:
    a) Let $\xi_t = 0$ and specify $\sigma(i+k+1)$ such that $1 \le |E^{(i)}_{\text{update}}| < k$. That is, at least one of $\{(\sigma(j), \sigma(i+k+1)) : i+2 \le j \le i+k\}$ is not a blue edge in G. In this case $\sigma(i+k+1)$ is a true neighbor of at least one of $\{\sigma(i+2), \ldots, \sigma(i+k)\}$; in other words, $\sigma(i+k+1) \in \cup_{i+2 \le j \le i+k} \mathcal{N}_{x^*}(j)$. Thus, the number of possibilities of $\sigma(i+k+1)$ is at most $2k(k-1)$.
    b) Let $\xi_t = 1$ and specify $\sigma(i+k+1)$ such that $|E^{(i)}_{\text{update}}| = k$. That is, each one of $\{(\sigma(j), \sigma(i+k+1)) : i+2 \le j \le i+k\}$ is a blue edge in G. Here $\sigma(i+k+1)$ can choose from at most $|V(G)| \le 2\Delta$ vertices.

The above process terminates when $|E^{(i)}_{\text{blue}}| = \Delta$, at which point the sequence $(\sigma(k+2), \ldots, \sigma(n-k))$ are determined. Note that each iteration, which one of the cases 1, 2 or 3 occurs is automatically determined. Therefore it suffices to enumerate (i) the value of $\sigma(i+k+1)$ at the ith iteration; (ii) the binary sequence ξ which determines case 3a or case 3b whenever case 3 occurs. Note that

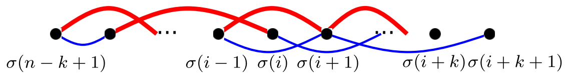


Fig. 8. Case 3: $\sigma(i+1)$ is attached to some red edge(s) and is not already balanced at step i. In the figure $\sigma(i+1)$ has red degree 2 and blue degree 1. Therefore $(\sigma(i+1), \sigma(i+k+1))$ must appear G as a blue edge.

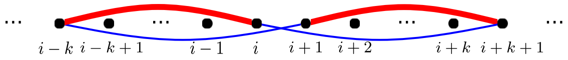


Fig. 9. The difference graph $G(x^{(i)})$.

- In total, case 3b) can occur at most $\lfloor \Delta/k \rfloor$ times because $|E^{(i)}_{\text{blue}}|$ increases by k each time.
- Also, case 2) and case 3) combined can occur at most 2Δ times, because they only occur when $\sigma(i+1) \in V(G)$.
- From the previous fact, the length of the ξ sequence is at most 2Δ.

Overall, the total number of possibilities is at most

$$\sum_{\xi \in \{0,1\}^{2\Delta}} (2k(k-1))^{2\Delta} (2\Delta)^{\Delta/k} \le \left(8k^2\right)^{2\Delta} \Delta^{\Delta/k}.$$

Combined with the cardinality bounds from step 1 and step 2, we have

$$|\mathcal{X}(E_{\text{red}})| \le 2(4k)^{2\Delta} \cdot (8k^2)^{2\Delta} \Delta^{\Delta/k} = 2(32k^3)^{2\Delta} \Delta^{\Delta/k}.$$

□

C. Information-Theoretic Lower Bound for Exact Recovery

Consider the Bayesian setting where x^* is drawn uniformly at random from the set $\mathcal{X}$ of all $2k$-NN graphs. Then MLE maximizes the probability of success, which, by definition, can be written as follows:

$$\mathbb{P}\left\{\widehat{x}_{\text{ML}} = x^*\right\} = \mathbb{P}\left\{\langle L, x - x^* \rangle < 0, \quad \forall x \neq x^*\right\}. \tag{25}$$

Due to the symmetry of $\mathcal{X}$, the probabilities in (25) are equal to the corresponding conditional probabilities, conditional on each $x^* \in \mathcal{X}$. WLOG, assume that x^* is the $2k$-NN graph associated with the identity permutation $\sigma^*(i) = i$. It is difficult to work with the intersection of dependent events in (25). The proof strategy is to select a subset of feasible solutions for which the events $\langle L, x - x^* \rangle < 0$ are mutually independent.

To this end, define $x^{(i)}$ to be the $2k$-NN graph corresponding to the permutation σ that swaps i and $i+1$, i.e., $\sigma(i) = i+1$, $\sigma(i+1) = i$, and $\sigma = \sigma^*$ everywhere else. It is easy to see that the difference graph $G(x^{(i)})$ contains four edges: (see Fig. 9)

$$\begin{aligned} \text{red edges:} \quad & (i-k, i), \quad (i+1, i+k+1); \\ \text{blue edges:} \quad & (i-k, i+1), \quad (i, i+k+1). \end{aligned}$$

Furthermore, for two such graphs $x^{(i)}$ and $x^{(j)}$ with $k+1 \le i < j \le n-k$, the edges sets $E\left(G(x^{(i)})\right)$ and $E\left(G(x^{(j)})\right)$ intersect if and only if $j - i \in \{k, k+1\}$. To avoid such pairs, we divide the x^* cycle into blocks of $3k$, each further divided into three sections of length k, and only consider those i which lies in the middle section of a block. Formally, define

$$\begin{aligned} D = \{&k+1, k+2, \dots, 2k, 4k+1, \dots, 5k, \dots, \\ &3k(\lfloor n/3k \rfloor - 1) + k + 1, \dots, 3k(\lfloor n/3k \rfloor - 1) + 2k\}. \end{aligned}$$

Then for distinct i and j in D, the difference graph of $x^{(i)}$ and $x^{(j)}$ have disjoint edge sets. This means all elements of $\left\{\langle L, x^{(i)} - x^* \rangle : i \in D\right\}$ are mutually independent.

For each $i \in D$, we have

$$\begin{aligned} &\mathbb{P}\{\langle L, x^{(i)} - x^* \rangle < 0\} \\ =&\mathbb{P}\{L(i-k, i+1) + L(i, i+k+1) \\ &\quad - L(i-k, i) - L(i+1, i+k+1) < 0\} \\ =&\mathbb{P}\left\{Y_1 + Y_2 - X_1 - X_2 < 0\right\}, \end{aligned}$$

where X_1, X_2 are independent copies of $\log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}$ under P_n, and Y_1, Y_2 are independent copies of $\log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}$ under Q_n. Therefore

$$\begin{aligned} &\mathbb{P}\left\{\langle L, x - x^* \rangle < 0, \quad \forall x \neq x^*\right\} \\ \leq&\mathbb{P}\left\{\langle L, x^{(i)} - x^* \rangle < 0, \quad \forall i \in D\right\} \\ =&\left(\mathbb{P}\left\{Y_1 + Y_2 - X_1 - X_2 < 0\right\}\right)^{|D|} \\ \leq&\exp\left(-|D|\mathbb{P}\left\{Y_1 + Y_2 - X_1 - X_2 \geq 0\right\}\right). \end{aligned} \tag{26}$$

From the mutual independence of X_1, X_2, Y_1, Y_2, for any $\tau \in \mathbb{R}$, we have

$$\begin{aligned} &\mathbb{P}\left\{Y_1 \geq \tau\right\}\mathbb{P}\left\{Y_2 \geq \tau\right\}\mathbb{P}\left\{X_1 \leq \tau\right\}\mathbb{P}\left\{X_2 \leq \tau\right\} \\ \leq&\mathbb{P}\left\{Y_1 + Y_2 - X_1 - X_2 \geq 0\right\} \end{aligned}$$

and hence

$$\begin{aligned} &\log \mathbb{P}\left\{Y_1 + Y_2 - X_1 - X_2 \geq 0\right\} \\ \geq&2 \sup_{\tau \in \mathbb{R}} \left(\log \mathbb{P}\left\{Y_1 \geq \tau\right\} + \log \mathbb{P}\left\{X_1 \leq \tau\right\}\right). \end{aligned}$$

Since (26) is an upper bound for $\mathbb{P}\{\widehat{x}_{\mathrm{ML}} = x^*\}$, the success of the MLE must require that

$$\begin{aligned} &\log |D| + 2 \sup_{\tau \in \mathbb{R}} \left(\log \mathbb{P}\left\{Y_1 \geq \tau\right\} + \log \mathbb{P}\left\{X_1 \leq \tau\right\}\right) \\ &\to -\infty. \end{aligned}$$

On one hand, by Assumption 1,

$$\begin{aligned} &\sup_{\tau \in \mathbb{R}} \left(\log \mathbb{P}\left\{Y_1 \geq \tau\right\} + \log \mathbb{P}\left\{X_1 \leq \tau\right\}\right) \\ \geq&-(1 + o(1))\alpha_n + o(\log n). \end{aligned}$$

On the other hand, by construction we have $|D| \geq n/3 - k \geq n/4$ under the assumption $k < n/12$, from which we conclude the necessity of $2\alpha_n \geq (1 + o(1)) \log n$ for $\mathbb{P}\{\widehat{x}_{\mathrm{ML}} = x^*\} \to 1$.

IV. Almost Exact Recovery

In this section we give the proof of Theorem 2. The proof follows the same strategy as that in [26], which studies recovering a hidden community (densely-connected subgraph) in a large weighted graph; specifically, the sufficient condition for almost exact recovery is established by analyzing the (suboptimal) MLE[7] and the necessary condition follows from a mutual information and rate-distortion argument. Nevertheless, as our model differs significantly from the hidden community model, the proof here requires much more sophisticated techniques, involving a delicate union bound to separate the contributions of the red edges from blue edges and crucially relying on the counting lemmas for $2k$-NN graphs shown in Section III-B.

A. Proof of Correctness of MLE for Almost Exact Recovery

We abbreviate the MLE $\widehat{x}_{\mathrm{ML}}$ as $\widehat{x}$ in the proof below. For any $2k$-NN graph $x \in \mathcal{X}$, recall from Section III-A the difference graph $G(x)$ defined by $x - x^*$. Let $E_{\mathrm{red}}(x)$ and $E_{\mathrm{blue}}(x)$ denote the set of red and blue edges in $G(x)$, respectively. Let $\Delta = |E_{\mathrm{red}}(\widehat{x}))| = d(\widehat{x}, x^*)/2$. Then $0 \leq \Delta \leq nk$. To prove the sufficiency, it suffices to show that $\mathbb{P}\left\{\Delta \geq \epsilon_n nk\right\} = o(1)$ for some $\epsilon_n = o(1)$ to be chosen.

Recall that $\mathcal{X}_\ell$ is the set of all $x \in \mathcal{X}$ such that $G(x)$ contains exactly ℓ red edges, i.e., $d(x, x^*) = 2\ell$. For any $1 \leq \Delta \leq nk$ and any $\tau \in \mathbb{R}$, we have that

$$\begin{aligned} \{\Delta = \ell\} &\subset \{\exists x \in \mathcal{X}_\ell : \langle L, x - x^* \rangle > 0\} \\ &\subset \left\{\exists x \in \mathcal{X}_\ell : \sum_{e \in E_{\mathrm{red}}(x)} L_e < \sum_{e \in E_{\mathrm{blue}}(x)} L_e\right\} \\ &\subset \left\{\exists x \in \mathcal{X}_\ell : \sum_{e \in E_{\mathrm{red}}(x)} L_e \leq \ell\tau\right\} \cup \\ &\quad\ \left\{\exists x \in \mathcal{X}_\ell : \sum_{e \in E_{\mathrm{blue}}(x)} L_e \geq \ell\tau\right\}. \end{aligned}$$

For each $x \in \mathcal{X}_\ell$, we have that

$$\sum_{e \in E_{\mathrm{red}}(x)} L_e \overset{d}{=} \sum_{i=1}^{\ell} X_i, \qquad \sum_{e \in E_{\mathrm{blue}}(x)} L_e \overset{d}{=} \sum_{i=1}^{\ell} Y_i,$$

where X_i's and Y_i's denote i.i.d. copies of the log-likelihood ratio $\log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}$ under distribution P_n and Q_n respectively. Chernoff bound gives that for all $\tau \in [-D(Q_n \| P_n), D(P_n \| Q_n)]$ and $\ell \geq 1$,

$$\begin{aligned} \mathbb{P}\left\{\sum_{i=1}^{\ell} X_i \leq \ell\tau\right\} &\leq e^{-\ell E_P(\tau)}, \\ \mathbb{P}\left\{\sum_{i=1}^{\ell} Y_i \geq \ell\tau\right\} &\leq e^{-\ell E_Q(\tau)}. \end{aligned} \tag{27}$$

[7] For almost exact recovery, the optimal estimator that minimizes the objective $\mathbb{E}\left[d(\widehat{x}, x^*)\right]$ is the bit-wise maximum a posterior (MAP) estimator: $\widehat{x}_e(w) = 1$ if $\mathbb{P}\left\{x_e^* = 1|w\right\} \geq \mathbb{P}\left\{x_e^* = 0|w\right\}$; and $\widehat{x}_e(w) = 0$ otherwise.

Recall that $\mathcal{E}_{\rm red}(\ell) = \{E_{\rm red}(x) : x \in X_\ell\}$ stands for the set of all possible $E_{\rm red}(x)$ where x ranges over all possible $2k$-NN graphs in $\mathcal{X}$ with $d(x, x^*) = 2\ell$. Note that

$$\left\{ \exists x \in \mathcal{X}_\ell : \sum_{e \in E_{\rm red}(x)} L_e \le \ell\tau \right\} = \left\{ \exists E \in \mathcal{E}_{\rm red}(\ell) : \sum_{e\in E} L_e \le \ell\tau \right\}.$$

By the union bound and Chernoff's bound (27), we get that

$$\begin{aligned} &\mathbb{P}\left\{\Delta = \ell\right\} \\ \le& |\mathcal{E}_{\rm red}(\ell)|\,\mathbb{P}\left\{\sum_{i=1}^{\ell} X_i \le \ell\tau\right\} + |\mathcal{X}_\ell|\,\mathbb{P}\left\{\sum_{i=1}^{\ell} Y_i \ge \ell\tau\right\} \\ \le& |\mathcal{E}_{\rm red}(\ell)|\, e^{-\ell E_P(\tau)} + |\mathcal{X}_\ell|\, e^{-\ell E_Q(\tau)}. \end{aligned} \tag{28}$$

We first focus on the case $k \ge 2$. Note that for $\ell \ge \epsilon_n nk$, it follows from Lemma 3 that

$$\begin{aligned} |\mathcal{E}_{\rm red}(\ell)| \le (96k^2)^\ell \binom{kn}{\lfloor \ell/2 \rfloor} &\le (96k^2)^\ell \left(\frac{2enk}{\ell}\right)^{\ell/2} \\ &\le (96k^2)^\ell \left(\frac{2e}{\epsilon_n}\right)^{\ell/2}, \end{aligned}$$

where we used $\binom{n}{m} \le (en/m)^m$ and $\ell \le \epsilon_n nk$. Similarly, combining Lemma 3 and Lemma 4, we have for $\epsilon_n nk \le \ell \le nk$,

$$\begin{aligned} |\mathcal{X}_\ell| &\le (96k^2)^\ell \binom{kn}{\lfloor \ell/2 \rfloor} \cdot 2(32k^3)^{2\ell} \ell^{\ell/k} \\ &\le (96k^2)^\ell \left(\frac{2ekn}{\ell}\right)^{\ell/2} 2(32k^3)^{2\ell}\ell^{\ell/k} \\ &\le (96k^2)^\ell \left(\frac{2e}{\epsilon_n}\right)^{\ell/2} 2(32k^3)^{2\ell}(nk)^{\ell/k}, \end{aligned}$$

where the last inequality comes from the range of ℓ. Thus for any $\epsilon_n nk \le \ell \le nk$,

$$\mathbb{P}\left\{\Delta = \ell\right\} \le e^{-\ell E_1} + e^{-\ell E_2}$$

with

$$\begin{aligned} E_1 &\triangleq E_P(\tau) - \frac{1}{2}\log\frac{1}{\epsilon_n} - 2\log k - O(1), \\ E_2 &\triangleq E_Q(\tau) - \frac{1}{k}\log n - \frac{1}{2}\log\frac{1}{\epsilon_n} - 8\log k - O(1). \end{aligned}$$

By (13), we have $kD(P_n\|Q_n)(1-\eta) \ge \log n$ for some $\eta \in (0,1)$. Choose $\tau = (1-\eta)D(P_n\|Q_n)$. By the assumption (12), we have

$$E_1 \ge c\eta^2\, D(P_n\|Q_n) - \frac{1}{2}\log\frac{1}{\epsilon_n} - 2\log k - O(1).$$

Using the fact that $E_P(\tau) = E_Q(\tau) - \tau$, we have

$$\begin{aligned} E_2 \ge\ & c\eta^2 D(P_n\|Q_n) - \frac{1}{2}\log\frac{1}{\epsilon_n} + \\ & D(P_n\|Q_n)(1-\eta) - \frac{1}{k}\log n - 8\log k - O(1) \\ \ge\ & c\eta^2 D(P_n\|Q_n) - \frac{1}{2}\log\frac{1}{\epsilon_n} - 8\log k - O(1). \end{aligned}$$

Since $k\log k = o(\log n)$ and $kD(P_n\|Q_n) \ge \log n$, it follows that $D(P_n\|Q_n) = \omega(\log k)$. Therefore, setting $\epsilon_n = 1/\left(kD(P_n\|Q_n)\right)$, it follows that $E \triangleq \min\{E_1, E_2\} = \Omega(D(P_n\|Q_n))$. Hence,

$$\begin{aligned} \mathbb{P}\left\{\Delta \ge \epsilon_n nk\right\} &= \sum_{\ell \ge \epsilon_n nk}^{nk} \mathbb{P}\left\{\Delta = \ell\right\} \\ &\le \sum_{\ell = \epsilon_n nk}^{\infty} \left(\mathrm{e}^{-\ell E_1} + \mathrm{e}^{-\ell E_2}\right) \\ &\le \frac{2\exp(-\epsilon_n nkE)}{1-\exp(-E)} = \exp\left(-\Omega(n)\right). \end{aligned}$$

In other words, the MLE achieves almost exact recovery. The conclusion for $k = 1$ is shown similarly using the combinatorial upper bounds $|\mathcal{E}_{\rm red}(\ell)| \le \binom{nk}{\ell}$ and $|\mathcal{X}_\ell| \le (4n)^\ell$ from (22).

Remark 3: This is a good place to explain why the conditions for exact and almost exact recovery involve two different distance measures between P_n and Q_n. The two types of recovery demand control of $\mathbb{P}\{\Delta = \ell\}$ for different ranges of ℓ. For almost exact recovery, we need to control $\mathbb{P}\{\Delta = \ell\}$ for $\ell \ge \epsilon nk$. In this range, there is a large difference between $|\mathcal{E}_{\rm red}(\ell)|$ and $|\mathcal{X}_\ell|$. Indeed from Lemma 4, there may be up to $(ck)^{2\ell}\ell^{\ell/k}$ members of $\mathcal{X}_\ell$ with the same set of red edges. Hence for large ℓ, it pays to separately account for the contributions of the red edges and blue edges, as done in (28). To balance out the two terms in (28), the exponential tilting parameter τ is chosen so that $E_Q(\tau)$ is large. Given that $E_Q(\tau)$ is an increasing function on $[-D(Q_n\|P_n), D(P_n\|Q_n)]$, we choose τ close to $D(P_n\|Q_n)$ with $E_Q(\tau) \approx D(P_n\|Q_n)$. As a result, the condition for almost exact recovery emerges from the tension between $D(P_n\|Q_n)$ and $|\mathcal{X}_\ell|$.

Exact recovery, on the other hand, requires upper bounding $\mathbb{P}\{\Delta = \ell\}$ for all $\ell \ge 2$. In fact as seen from the lower bound proof of Theorem 1, the bottleneck for exact recovery happens at $\ell = 2$, where $|\mathcal{E}_{\rm red}(\ell)|$ and $|\mathcal{X}_\ell|$ are around the same order. In this regime, there is no more gains in separating the red and blue edge weights, and it is more favorable to directly applying the Chernoff bound to their differences:

$$\begin{aligned} \mathbb{P}\{\Delta = \ell\} &\le |\mathcal{X}_\ell|\,\mathbb{P}\left\{\sum_{i\le\ell} X_i \le \sum_{i\le\ell} Y_i\right\} \\ &\le |\mathcal{X}_\ell|\,\mathbb{P}\left\{-\ell\inf_{\tau}(E_P(\tau) + E_Q(\tau))\right\} \\ &= |\mathcal{X}_\ell|\, e^{-\ell\alpha_n}. \end{aligned}$$

See [24, page 67] for a derivation of the equality $\inf_\tau(E_P(\tau) + E_Q(\tau)) = \alpha_n$. As a result, the condition for exact recovery is governed by the distance α_n.

B. Information-Theoretic Lower Bound for Almost Exact Recovery

Suppose that almost exact recovery of x^* is achieved by some estimator $\widehat{x}$, such that $\mathbb{E}\left[d(\widehat{x}, x^*)\right] = 2nk\epsilon_n$, for some $\epsilon_n \to 0$. We show that (14) must hold. First, we can assume, WLOG, $\widehat{x}$ takes value in $\mathcal{X}$, the set of all $2k$-NN graphs. Indeed, if we set

$$\widehat{x}' = \mathrm{argmin}_{x \in \mathcal{X}} d(x, \widehat{x}), \tag{29}$$

then $d(\widehat{x}', x^*) \le d(\widehat{x}', \widehat{x}) + d(\widehat{x}, x^*) \le 2\ d(\widehat{x}, x^*)$ and hence $\mathbb{E}\left[d(\widehat{x}', x^*)\right] \le 4nk\epsilon_n$; in other words, $\widehat{x}'$ also achieves almost exact recovery.

Since $x^* \to w \to \widehat{x}$ form a Markov chain, by the data processing inequality of mutual information, we have

$$\begin{aligned}
&I(w; x^*) \\
\ge& I(\widehat{x}, x^*) \\
\ge& \inf\{I(\widetilde{x}, x^*) : \widetilde{x} \in \mathcal{X}, \mathbb{E}\left[d(\widetilde{x}, x^*)\right] \le 2nk\epsilon_n\} \qquad (30) \\
=& H(x^*) - \sup\{H(x^*|\widetilde{x}) : \widetilde{x} \in \mathcal{X}, \\
&\qquad \mathbb{E}\left[d(\widetilde{x}, x^*)\right] \le 2nk\epsilon_n\} \qquad (31)
\end{aligned}$$

where the infimum in (30), known as the rate-distortion function, is taken over all conditional distributions $P_{\widetilde{x}|x^*}$ satisfying the constraints. Note that $H(x^*) = \log(n!) = (1+o(1))n \log n$. Moreover from Lemma 3 and Lemma 4, for any fixed $\widetilde{x} \in \mathcal{X}$, the number of possible $x^* \in \mathcal{X}$ such that $d(\widetilde{x}, x^*) = 2\ell$ is at most

$$(96k^2)^\ell \binom{kn}{\lfloor \ell/2 \rfloor} \cdot 2(32k^3)^{2\ell} \ell^{\ell/k} \le 2(c_k n/\ell)^{\ell/2} (nk)^{\ell/k},$$

where $c_k = 2e96^2 2^{20} k^{17}$ and we used the fact that $\ell \le nk$. Therefore,

$$\begin{aligned}
&H(x^*|\widetilde{x}, d(\widetilde{x}, x^*) = 2\ell) \\
\le& \frac{\ell}{2}\log(c_k n) - \frac{\ell}{2}\log \ell + \frac{\ell}{k}\log(nk) + \log 2.
\end{aligned}$$

By the convexity of $x \mapsto x \log x$ and Jensen's inequality, it follows that

$$\begin{aligned}
&H(x^*|\widetilde{x}, d(\widetilde{x}, x^*)) \\
\le& \frac{\mathbb{E}\left[d(\widetilde{x}, x^*)\right]}{4}\Bigg[\log(c_k n) - \log\frac{\mathbb{E}\left[d(\widetilde{x}, x^*)\right]}{2} + \\
&\qquad \frac{2}{k}\log(nk)\Bigg] + \log 2.
\end{aligned}$$

Furthermore, $d(x, x^*)$ takes values in $\{0, \ldots, 2nk\}$. Thus from the chain rule,

$$\begin{aligned}
H(x^*|\widetilde{x}) =& H(d(\widetilde{x}, x^*)|\widetilde{x}) + H(x^*|\widetilde{x}, d(\widetilde{x}, x^*)) \\
\le& \log(1+2nk) + H(x^*|\widetilde{x}, d(\widetilde{x}, x^*)),
\end{aligned}$$

and hence

$$\begin{aligned}
&\sup\{H(x^*|\widetilde{x}) : \widetilde{x} \in \mathcal{X}, \mathbb{E}\left[d(\widetilde{x}, x^*)\right] \le 2nk\epsilon_n\} \\
\le& \frac{1}{2}\epsilon_n nk\left[\log(c_k n) - \log\left(\epsilon_n nk\right)\right] + \\
&\epsilon_n n \log(nk) + \log 2 + \log(1+2nk) \\
=& o(n \log n),
\end{aligned}$$

where the last equality holds due to the assumption that $k \log k = o(\log n)$. Therefore, we get from (31) that $I(w; x^*) \ge (1+o(1))n \log n$. On the other hand,

$$\begin{aligned}
I(w; x^*) =& \min_{Q_w} \mathbb{E}_{x^*}\left[D(P_{w|x^*} \| Q_w)\right] \\
\le& \mathbb{E}_{x^*}\left[D(P_{w|x^*} \| Q_n^{\otimes\binom{n}{2}})\right] \\
=& nkD(P_n \| Q_n),
\end{aligned}$$

where the minimum is taken over all distribution Q_w, achieved at $Q_w = P_w$. This yields the desired $kD(P_n\|Q_n) \ge (1+o(1))\log n$.

V. Discussion on Efficient Recovery Algorithms

As shown in Section III and Section IV, the sharp thresholds for exact and almost exact recovery can both be attained by the MLE (7), which, however, entails solving the computationally intractable max-weight $2k$-NN subgraph problem. So far no polynomial-time algorithm is known to achieve the sharp thresholds for exact or almost exact recovery except when $k = 1$ [24]. In Section V-A, we consider several computationally efficient algorithms to recover the hidden $2k$-NN graph and analyze their statistical properties. In Section V-C, we focus on the special case of small-world graphs where the edge weights are distributed Bernoulli and give a polynomial time algorithm that achieves the sharp threshold for exact recovery.

A. Efficient Recovery Algorithms Under the General Hidden 2k-NN Graph Model

Recall that the MLE is the solution to

$$\widehat{x}_{\mathrm{ML}} = \arg\max_{x \in \mathcal{X}} \ \langle L, x \rangle,$$

where $L_e = \log \frac{\mathrm{d}P_n}{\mathrm{d}Q_n}(w_e)$ is the log likelihood ratio. In the special case of $k = 1$, this reduces to the max-weighted Hamiltonian cycle problem. The previous work [24] analyzes its 2-factor integer linear program (ILP) relaxation and fractional 2-factor linear program (LP) relaxation, and show that they achieve the sharp exact recovery threshold $\liminf_{n\to\infty}(\alpha_n/\log n) > 1$. This motivates us to consider the ILP and LP relaxation for general k.

1) 2k-Factor ILP Relaxation: By relaxing the $2k$-NN graph constraint in the MLE to a degree constraint, we arrive at the following $2k$-factor ILP:

$$\begin{aligned}
\widehat{x}_{2k\mathrm{F}} = \mathrm{argmax}_x \ & \langle L, x \rangle \\
\text{s.t.} \ & \sum_{v \sim u} x_{(u,v)} = 2k, \ \ \forall u, \\
& x_e \in \{0, 1\}, \ \ \forall e
\end{aligned} \tag{32}$$

where the first constraint enforces that every vertex has degree $2k$. It is known that for constant k, the ILP (32) can be solvable in $O(n^4)$ time [29].

To analyze $\widehat{x}_{2k\mathrm{F}}$, note that each feasible solution x to the ILP is a $2k$-regular graph. Therefore, the difference graph $x - x^*$ is still balanced and the simple bound (22) continues to hold:

$$|\mathcal{X}_\Delta| \le (4kn)^\Delta,$$

where $\mathcal{X}_\Delta$ is the collection of $2k$-regular graphs x such that the difference graph $x - x^*$ contains exactly Δ red edges.

Then following the same argument as in the upper bound proof of Theorem 1 in Section III-A, we have

$$\begin{aligned}\mathbb{P}\{\widehat{x}_{2k\mathrm{F}} \neq x^*\} &\leq \sum_{\Delta\geq 1}(4kn)^{\Delta}\exp\left(-\alpha_n\Delta\right)\\ &= \sum_{\Delta\geq 1}\exp\left[-\Delta\left(\alpha_n-\log(4kn)\right)\right],\end{aligned}$$

which is order $o(1)$ whenever $\liminf_{n\to\infty}(\alpha_n/\log n) > 1$ and $k = n^{o(1)}$. This is suboptimal by a multiplicative factor of 2 compared to the sharp threshold $\liminf_{n\to\infty}\frac{\alpha_n}{2\log n} > 1$.

In fact, $\widehat{x}_{2k\mathrm{F}}$ provably fails to attain the sharp exact recovery threshold when $2 \leq k \leq n^{o(1)}$. To see this, assume that x^* is associated with the identity permutation and consider its modifications of the following form: fix two vertices i, j for which $d_{x^*}(i,j) > k$, remove the edges $(i, i+1)$ and $(j, j+1)$ in x^* and add the edges (i,j) and $(i+1, j+1)$, resulting in a $2k$-regular graph $x^{(i,j)}$ feasible to (32). There are $O(n^2)$ such modified solutions and the difference in weights $\langle w, x^{(i,j)} - x^*\rangle$ are close to being mutually independent. Each $x^{(i,j)}$ corresponds to a difference graph with $\Delta = 2$ red edges. By following the similar lower bound proof for exact recovery in Section III-C, we can conclude that under Assumption 1, if $\liminf_{n\to\infty}(\alpha_n/\log n) < 1$, then with high probability there is at least one feasible solution $x^{(i,j)}$ such that $\langle L, x^{(i,j)} - x^*\rangle > 0$, yielding $\widehat{x}_{2k\mathrm{F}} \neq x^*$.

2) LP Relaxation: By further relaxing the integer constraint in $\widehat{x}_{2k\mathrm{F}}$, we arrive at the following LP:

$$\begin{aligned}\widehat{x}_{\mathrm{LP}} = \mathrm{argmax}_x\ &\langle L, x\rangle\\ \text{s.t.}\ &\sum_{v\sim u} x_{(u,v)} = 2k,\ \ \forall u,\\ &x_e \in [0,1],\ \ \forall e.\end{aligned}$$

Since $\widehat{x}_{\mathrm{LP}}$ is a relaxation of $\widehat{x}_{2k\mathrm{F}}$, it follows from the above negative result of ILP that under Assumption 1, $\widehat{x}_{\mathrm{LP}} \neq x^*$ when $\liminf_{n\to\infty}(\alpha_n/\log n) < 1$. In the positive direction, one can show that $\widehat{x}_{\mathrm{LP}}$ also achieves exact recovery for $1 \leq k \leq n^{o(1)}$ when $\liminf_{n\to\infty}(\alpha_n/\log n) > 1$. That is because firstly, the feasible set of the LP is a fractional $2k$-factor polytope, the entries of whose extreme points are all half-integrals by the determinant analysis in [30, p 280]. That is, $(\widehat{x}_{\mathrm{LP}})_e \in \{0, 1/2, 1\}$ for all e. Moreover, the difference graph $x - x^*$ can be represented by a balanced multigraph with edge multiplicity at most 2 (we refer the reader to [24] for details). The rest of the proof follows exactly from the proof of [24, Theorem 1].

To sum up, when it comes to exact recovery, the statistical performance for $\widehat{x}_{2k\mathrm{F}}$ and $\widehat{x}_{\mathrm{LP}}$ match in the asymptotics. They both require $\liminf_{n\to\infty}(\alpha_n/\log n) > 1$, which is suboptimal by a factor of two. Whether they can achieve almost exact recovery under weaker conditions remains open.

3) Simple Thresholding: To partially address the problem of almost exact recovery, we consider a naïve thresholding estimator $\widehat{x}_{\mathrm{TH}}$ given by

$$\widehat{x}_{\mathrm{TH}}(e) = \mathbb{1}\left\{L_e > \tau_n\right\},$$

where the sequence $\tau_n = (1-\eta)D(P_n\|Q_n)$ for some small fixed constant η that will be later specified. By definition of $\widehat{x}_{\mathrm{TH}}$,

$$\begin{aligned}&\mathbb{E}\left[d(\widehat{x}_{\mathrm{TH}}, x^*)\right]\\ &= \sum_{e:x_e^*=1}\mathbb{P}\left\{L_e \leq \tau_n\right\} + \sum_{e:x_e^*=0}\mathbb{P}\left\{L_e > \tau_n\right\}.\end{aligned}$$

From the Chernoff bound (27), we have for $\tau_n \in [-D(Q_n\|P_n), D(P_n\|Q_n)]$,

$$\begin{aligned}\mathbb{P}\{L_e \leq \tau_n\} &\leq e^{-E_P(\tau_n)}\ \text{ if }\ x_e^* = 1;\\ \mathbb{P}\{L_e > \tau_n\} &\leq e^{-E_Q(\tau_n)}\ \text{ if }\ x_e^* = 0.\end{aligned}$$

Assume that $\liminf\frac{D(P_n\|Q_n)}{\log(n/k)} > 1$. Then there exists some fixed constant $\eta \in (0,1)$ such that $(1-\eta)D(P_n\|Q_n) \geq \log(n/k)$ for all sufficiently large n. Under Assumption 2, we get that for some constant $c > 0$,

$$\begin{aligned}&\sum_{e:x_e^*=1}\mathbb{P}\left\{L_e \leq \tau_n\right\}\\ &\leq kne^{-E_P(\tau_n)}\\ &\leq kn\exp(-c\eta^2\ D(P_n\|Q_n))\\ &\leq kn\exp(-c\eta^2\log(n/k)) = o(kn).\end{aligned}$$

From $E_Q(\tau_n) = E_P(\tau_n) + \tau_n$, we also have

$$\begin{aligned}&\sum_{e:x_e^*=0}\mathbb{P}\left\{L_e > \tau_n\right\}\\ &\leq n^2\ e^{-E_Q(\tau_n)}\\ &= n^2\exp\left(-(1-\eta)D(P_n\|Q_n) - E_P(\tau_n)\right)\\ &\leq knE_P(\tau_n) = o(kn).\end{aligned}$$

Thus we have shown that under Assumption 2, the thresholding estimator $\widehat{x}_{\mathrm{TH}}$ achieves almost exact recovery provided that $\liminf\frac{D(P_n\|Q_n)}{\log(n/k)} > 1$, which is optimal for $k = 1$, in view of the necessary condition $\liminf\frac{kD(P_n\|Q_n)}{\log n} \geq 1$.

It is worth pointing out that $\widehat{x}_{\mathrm{TH}}$ may not be a valid $2k$-NN graph. One can of course consider the modified estimator (29) by projecting $\widehat{x}$ to the set of $2k$-NN graphs; however, it is unclear whether this can be done in polynomial time. It is an interesting open problem whether a computationally efficient $2k$-NN graph estimator can be obtained from $\widehat{x}_{\mathrm{TH}}$ and still inherits the almost exact recovery guarantee $\liminf\frac{D(P_n\|Q_n)}{\log(n/k)} > 1$.

4) Spectral Methods: For a variety of problems such as clustering and community detection, spectral methods have been successfully used to recover the hidden structures based on the principal eigenvectors of the observed graph [31], [32]. In our model, with slight abuse of notation, let L denote the $n\times n$ matrix of log likelihood ratios, where $L_{ij} = \log\frac{\mathrm{d}P_n}{\mathrm{d}Q_n}(w_e)$ for edge $(i,j) = e$ and $L_{ii} = 0$. Note that the principal eigenvectors of $\mathbb{E}(L)$ contain perfect information about the hidden $2k$-NN graph. Indeed, to see this, rewrite L as

$$\begin{aligned}L =& (D(P_n\|Q_n) + D(Q_n\|P_n))\Pi B\Pi^\top -\\ &D(Q_n\|P_n)\left(\mathbf{J}-\mathbf{I}\right) + (L - \mathbb{E}(L)),\end{aligned}$$

where Π is the permutation matrix associated with the hidden $2k$-NN graph; B is the adjacency matrix of the basic $2k$-NN graph where $B_{ij} = 1$ if $\min\{|i-j|, n-|i-j|\} \leq k$

and $B_{ij} = 0$ otherwise; $\mathbf{J}$ is the all-ones matrix and $\mathbf{I}$ is the identity matrix. Since B is a circulant matrix, its eigenvalues and eigenvectors can be determined by the discrete Fourier transform of the window function:

$$\lambda_j = \sum_{\ell=-k}^{k} \exp\left(\frac{\mathbf{i}2\pi j \ell}{n}\right) - 1 = \frac{\sin\frac{(2k+1)j\pi}{n}}{\sin\frac{j\pi}{n}} - 1$$

for $j = 0, \ldots, n-1$, where $\mathbf{i} = \sqrt{-1}$ is the imaginary unit, $\lambda_0 = 2k$ (degree) and $\lambda_{n-j} = \lambda_j$ which decays similarly to the sinc function. Furthermore, the eigenvector v_1 of $\Pi B \Pi^\top$ corresponding to λ_1 encodes the permutation matrix Π perfectly as $v_1 = \Pi(\omega^0, \ldots, \omega^{n-1})^\top$, where $\omega = \exp\left(\mathbf{i}\frac{2\pi}{n}\right)$ is the n^{th} root of unity. Thus in the noiseless case one can exactly recover the underlying permutation Π and hence the hidden $2k$-NN graph from v_1. That prompts us to obtain an estimator $\widehat{\Pi}$ by sorting the entry-wise angles of the second eigenvector of L, and define $\widehat{x}_{\text{spectral}} = \widehat{\Pi} B \widehat{\Pi}^\top$.

Unfortunately, it turns out that v_1 can be very sensitive to the noise perturbation $(L - \mathbb{E}L)$ due to the small eigengap. For example in the special case $P_n = \mathcal{N}(\mu_n, 1)$ and $Q_n = \mathcal{N}(0,1)$, we have

$$L = \mu_n^2 \Pi B \Pi^\top - \frac{\mu_n^2}{2}(\mathbf{J} - \mathbf{I}) + \mu Z,$$

where Z is a symmetric Gaussian matrix with zero diagonal and $Z_{ij} = Z_{ji}$ independently drawn from $\mathcal{N}(0,1)$ for $i < j$. The eigengap of $\lambda_1 - \lambda_2 \sim k^3/n^2$, while the spectral norm of the noise perturbation $\|Z\|_2$ is on the order of $\sqrt{n}$. Thus, by the Davis-Kahan theorem, the second eigenvector of L is close to v_1 if $\mu_n k^3/n^2 \gtrsim \sqrt{n}$, i.e., $\mu_n \gtrsim n^{5/2}/k^3$.[8] When $k = n^{o(1)}$, this is highly suboptimal compared to the sharp thresholds for exact and almost exact recovery given by (33), (34) in the succeeding subsection.

B. Numerical Experiments

In this subsection we carry out a numerical experiment to evaluate the performance of the algorithms discussed in Section V-A above. We focus on the Gaussian model with weight distribution $P_n = \mathcal{N}(\mu_n, 1)$ and $Q_n = \mathcal{N}(0,1)$ for $\mu_n > 0$. From Corollaries 1 and 2, under the Gaussian model, the sharp thresholds for exact recovery (for $2 \le k \le n^{o(1)}$) and almost exact recovery (for $1 \le k \le o(\log n/\log\log n)$) are

$$\liminf_{n\to\infty} \frac{\mu_n^2}{2\log n} > 1 \tag{33}$$

and

$$\liminf_{n\to\infty} \frac{k\mu_n^2}{2\log n} > 1, \tag{34}$$

respectively.

The setup of the simulation is as follows. Let $n = 500$, $k = 5$, and choose a sequence of μ_n so that $\mu_n^2/\log n$ takes value on a fine grid between 0 and 10 with step size 0.1. We run the following experiment for 200 independent trials. For each grid value of μ_n, we generate a random weighted graph w from the hidden $2k$-NN model with Gaussian edge weights. Given w, the four estimators $\widehat{x}_{2k\text{F}}$, $\widehat{x}_{\text{LP}}$, $\widehat{x}_{\text{TH}}$ and $\widehat{x}_{\text{spectral}}$ discussed in Section V-A are computed. The ILP $\widehat{x}_{2k\text{F}}$, and the LP $\widehat{x}_{\text{LP}}$ are solved with the branch-and-cut algorithm [33] and the simplex method [34], respectively; the thresholding estimator $\widehat{x}_{\text{TH}}$ selects those edges with weights at least $\sqrt{(4 + 1/\sqrt{\log n})\log n}$, which is slightly above the maximum weight of edges not in x^* with high probability. The outcome of the experiment is reported in Figures 10 and 11. In Fig. 10, the x-axis represents the signal strength $\mu_n^2/\log n$; the y-axis is the empirical proportion of an estimator exactly recovering the true NN graph x^* out of the 200 independent trials. For estimators that return fractional edge adjacency vectors $\widehat{x}$, it is viewed as achieving exact recovery if $\sum_e |\widehat{x}_e - x_e^*| < 10^{-5}$.

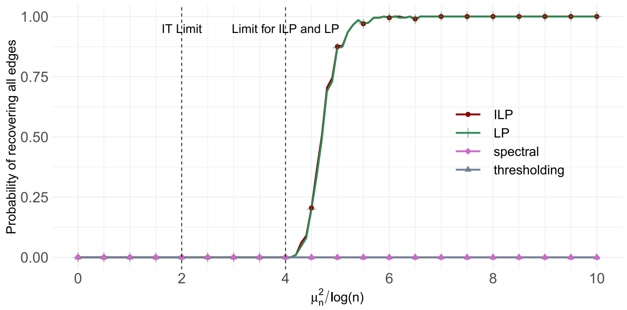


Fig. 10. Empirical proportion of exact recovery of the true $2k$-NN graph x^* for the four polynomial-time algorithms discussed in Section V-A applied to 200 independent instances drawn from the Gaussian weight distributions: $P_n = \mathcal{N}(\mu_n, 1)$ and $Q_n = \mathcal{N}(0,1)$, with $k = 5$ and $n = 500$.

One interesting observation from the simulation outcomes is that the performance of the ILP $\widehat{x}_{2k\text{F}}$ and the LP $\widehat{x}_{\text{LP}}$ are almost identical. In fact, we found that even though the LP can potentially return non-integral entries ($\widehat{x}_{\text{LP}}(e) = 1/2$ since the fractional solutions must be half-integral), those entries are very rare, and the solutions to the ILP and the LP often coincide.

In Fig. 10, the vertical line at $\mu_n^2/\log n = 2$ represents the asymptotic sharp threshold for exact recovery given by (33), while the vertical line at $\mu_n^2/\log n = 4$ is threshold for $\widehat{x}_{2k\text{F}}$ and $\widehat{x}_{\text{LP}}$ to achieve exact recovery as shown in Section V-A. This asymptotic result agrees well with the empirical result in Fig. 10 which is run for $n = 500$. Indeed, for all $\mu_n^2/\log n < 4$, both estimators failed to achieve exact recovery for all 200 trials. As $\mu_n^2/\log n$ exceeds 4, the fraction of exact recovery quickly converges to 1.

In comparison, the thresholding estimator $\widehat{x}_{\text{TH}}$ and the spectral estimator $\widehat{x}_{\text{spectral}}$ perform much worse. Perfect recovery never occurs across the entire parameter regime. To better compare these four estimators, we also quantify the edge discrepancy between the estimators and the truth. The results are shown in Fig. 11.

In Fig. 11, the y-axis is the normalized risk $\sum_e |\widehat{x}_e - x_e^*|/(kn)$ averaged across all trials. Note that an estimator can make errors by missing the edges in the true $2k$-NN graph x^* and adding those outside x^*. As a result, when the signal is very weak (small μ_n), the normalized risk can be close to

[8] In fact, following the analysis of spectral ordering in [25], one can show that the almost exact recovery can be efficiently achieved by $\widehat{x}_{\text{spectral}}$ under a slightly higher SNR: $\mu_n \gg n^{7/2}/k^4$.

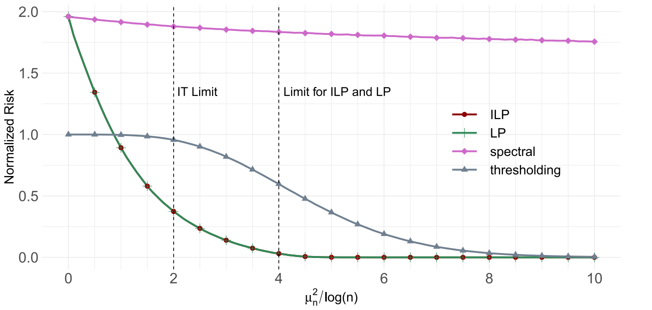


Fig. 11. Comparison on the normalized risk (fraction of misclassified edges) of the four polynomial-times algorithms under the same setup as in Fig. 10.

its maximal value 2, which is the case for all of $\widehat{x}_{2k\text{F}}$, $\widehat{x}_{\text{LP}}$ and $\widehat{x}_{\text{spectral}}$; in comparison, for weak signal, the thresholding estimator has a normalized risk close to 1 simply because very few edge weights exceed the threshold and $\widehat{x}_{\text{TH}}$ essentially output an almost empty graph. As μ_n increases, the risk of both $\widehat{x}_{2k\text{F}}$ and $\widehat{x}_{\text{LP}}$ decays rapidly and become almost zero near the threshold $\mu_n^2/\log(n) = 4$. In comparison, the normalized risk for the thresholding estimator $\widehat{x}_{\text{TH}}$ converges to zero at a slower rate. Furthermore, we argued in Section V-A that the spectral methods are highly sensitive to the noise perturbation. This is confirmed by Figure 11, in which $\widehat{x}_{\text{spectral}}$ performs considerably worse than the other three estimators.

C. Achieving the Exact Recovery Threshold Under the Small-World Model

Although designing efficient algorithms that achieve the sharp thresholds appears challenging under the general hidden $2k$-NN graph model, the task turns out to be more manageable for the special case of the Watts-Strogatz small-world graph model. Recall the special case (4) considered in the introduction with $P_n = \text{Bern}(p_n)$ and $Q_n = \text{Bern}(q_n)$, where

$$p_n = 1 - \epsilon_n + \frac{2\epsilon_n k}{n-1} \quad \text{and} \quad q_n = \frac{2\epsilon_n k}{n-1}. \tag{35}$$

For succinctness the subscript n will be suppressed for the remainder of this subsection. The observed graph $w \in \{0,1\}^{\binom{n}{2}}$ can be viewed as a noisy version of the true $2k$-NN graph x^*. By Corollaries 1, for $2 \le k \le n^{o(1)}$, the sharp threshold for exact recovery is $\liminf_{n\to\infty}(-2\log\epsilon/\log n) > 1$, i.e., $\epsilon \le n^{-\frac{1}{2}-\Omega(1)}$. We show that the greedy algorithm below succeeds under this condition. Note that the results in this section only apply to the parametrization under the small-world model and not the general case where P_n and Q_n are Bernoulli distributions.

As mentioned in Section I, to exactly recover x^*, it suffices to recover the corresponding Hamiltonian cycle identified by a permutation σ^*. Similar to the enumeration scheme that lies at the heart of the proof of Lemma 4, the algorithm first determines the neighborhood of one vertex and their ordering on the Hamiltonian cycle, and then sequentially finds the remaining vertices to complete the cycle.

Since it suffices to recover σ^* up to cyclic shifts and reversals, we can assume WLOG that $i_0 = \sigma^*(1)$. In Step 1 of Algorithm 1, one needs to order the members of $\mathcal{N}$ from the subgraph of x^* induced by $\mathcal{N}$. We will show, with high probability, $\mathcal{N} = \mathcal{N}_w(i_0)$ coincides with the true neighborhood $\mathcal{N}_{x^*}(\widehat{\sigma}(1))$ and the subgraph of w induced by $\mathcal{N}$ is identical to that of x^*. Therefore, we can infer the ordering of members of $\mathcal{N}$ using the nearest-neighbor structure of x^*. In particular, $\sigma^*(n-k+1)$ and $\sigma^*(k+1)$ are the only two vertices in $\mathcal{N}$ that are incident to exactly $k-1$ vertices in $\mathcal{N}$; having determined $\sigma^*(n-k+1)$, $\sigma^*(n-k+2)$ is the only vertex in $\mathcal{N}$ that is incident to $\sigma^*(n-k+1)$ and exactly $k-1$ other vertices in N; similarly $\sigma^*(n-k+3)$ can be uniquely determined given $\sigma^*(n-k+1)$ and $\sigma^*(n-k+2)$, so on and so forth.

Algorithm 1 Greedy Algorithm for Exact Recovery Under the Small-World Model

```
Start from an arbitrary vertex $i_0$ and let $\widehat{\sigma}(1) = i_0$;
**Step 1 (label the neighbors of $\widehat{\sigma}(1)$):**
Let $\mathcal{N} \triangleq \mathcal{N}_w(\widehat{\sigma}(1))$ be the set of vertices incident to $\widehat{\sigma}(1)$ in $w$;
**if** *The subgraph of $w$ induced by $\mathcal{N}$ is isomorphic to that of $x^*$ induced by $\mathcal{N}_{x^*}(1)$* **then**
    Use the subgraph of $w$ induced by $\mathcal{N}$ to determine (up to a reversal) the ordering $(\widehat{\sigma}(n-k+1), \ldots, \widehat{\sigma}(n), \widehat{\sigma}(2), \ldots, \widehat{\sigma}(k+1))$
**else** Report error and terminate;
**Step 2 (label the remaining vertices sequentially):**
$\mathcal{V}_{labeled} \triangleq \{\widehat{\sigma}(n-k+1), \ldots, \widehat{\sigma}(n), \widehat{\sigma}(1), \ldots, \widehat{\sigma}(k+1)\}$;
**for** *i=1 to n-2k-1* **do**
    $\mathcal{U} \triangleq \mathcal{N}_w(\widehat{\sigma}(i+1)) \backslash \mathcal{V}_{labeled}$;
    **switch** $|\mathcal{U}|$ **do**
        **case** $|\mathcal{U}| \ge 2$
            **if** *exactly one member $u$ of $\mathcal{U}$ is incident to $\widehat{\sigma}(i+2)$* **then**
                Set $\widehat{\sigma}(i+k+1) = u$.
            **else** Report error and terminate;
        **case** $|\mathcal{U}| = 1$
            Set $\widehat{\sigma}(i+k+1)$ be the vertex in $\mathcal{U}$;
        **case** $|\mathcal{U}| = 0$
            **if** *exactly one member $v$ of $\mathcal{N}_w(\widehat{\sigma}(i+2)) \backslash \mathcal{V}_{labeled}$ is incident to exactly $k$ members of $\mathcal{V}^c_{labeled}$* **then**
                Set $\widehat{\sigma}(i+k+1) = v$
            **else** Report error and terminate;
    $\mathcal{V}_{labeled} \triangleq \mathcal{V}_{labeled} \cup \{\widehat{\sigma}(i+k+1)\}$;
Output $x(\widehat{\sigma})$, the $2k$-NN graph corresponding to $\widehat{\sigma}$.
```

Step 1 of Algorithm 1 relies on the fact that with high probability, w and x^* completely agree in the neighborhood near the fixed vertex i_0. This does not hold uniformly for all vertices. However we will show that uniformly for all $i \in [n]$, w and x^* differ by at most one edge in the neighborhood near i. This fact is crucial for the success of the second step. Now we present the exact recovery guarantee of the algorithm.

Theorem 3: Consider the Watts-Strogatz small-world graph model under the parameterization (35). Assume that $k = n^{o(1)}$ and $\liminf_{n\to\infty}(-2\log\epsilon/\log n) > 1$, then with probability $1-o(1)$, Algorithm 1 runs successfully and returns $x(\widehat{\sigma}) = x^*$. In other words, Algorithm 1 achieves exact recovery.

Proof: Let us start with some notations. Recall that the set of true neighbors of a vertex j is denoted as $\mathcal{N}_{x^*}(j)$. Let $E_{x^*}(j)$ (resp. $E_w(j)$) denote the set of edges in x^* (resp. w) whose endpoints contain at least one member of $\mathcal{N}_{x^*}(\sigma^*(j)) \cup \{\sigma^*(j)\}$. We claim that under the assumption $\liminf_{n\to\infty}(-2\log\epsilon/\log n) > 1$, the following events occur simultaneously with high probability:

- $\mathcal{A}_n = \{E_w(1) = E_{x^*}(1)\}$;
- $\mathcal{B}_n = \{|E_w(j)\Delta E_{x^*}(j)| \le 1, \forall j\}$;
- $\mathcal{C}_n = \{w(\sigma^*(j+1), \sigma^*(j+k+2)) = 0, \forall j = 1, \ldots, n-2k-1\}$.

First we argue that on $\mathcal{A}_n \cap \mathcal{B}_n \cap \mathcal{C}_n$, Algorithm 1 correctly recovers x^*. Under $\mathcal{A}_n$, the subgraphs of w induced by $\mathcal{N}_{x^*}(\sigma^*(1))$ coincide with that of x^*. Hence Algorithm 1 successfully recovers $(\sigma^*(n-k+1), \ldots, \sigma^*(n), \sigma^*(2), \ldots, \sigma^*(k+1))$ up to a reversal. WLOG say $\widehat{\sigma}(i) = \sigma^*(i)$ for all $\sigma^*(i) \in \mathcal{N}_{x^*}(\sigma^*(1))$.

Next we show inductively that the algorithm correctly labels all the remaining vertices. Start from the inductive hypothesis that $\widehat{\sigma}(j) = \sigma^*(j)$ for all $j \le i+k$. Recall that $\mathcal{U} = \mathcal{N}_w(\widehat{\sigma}(i+1))\backslash\mathcal{V}_{labeled}$ and the algorithm considers the following three cases according to the size of $\mathcal{U}$:

1) $|\mathcal{U}| \ge 2$. Under $\mathcal{B}_n$, we must have $|\mathcal{U}| = 2$ because otherwise $E_w(i+1)\backslash E_{x^*}(i+1)$ contains at least two edges, contradicting $\mathcal{B}_n$. Write $\mathcal{U} = \{u, v\}$. One of u, v must be $\sigma^*(i+k+1)$, because otherwise $E_w(i+1)\backslash E_{x^*}(i+1)$ contains at least the two edges $(\sigma^*(i+1), u)$ and $(\sigma^*(i+1), v)$, contradicting $\mathcal{B}_n$. Say $u = \sigma^*(i+k+1)$. Then $(\sigma^*(i+1), u)$ is the only member of $E_w(i+1)\Delta E_{x^*}(i+1)$. Thus u and $\sigma^*(i+2)$ must be neighbors in w. Under $\mathcal{C}_n$, v cannot be $\sigma^*(i+k+2)$. Hence v and $\sigma^*(i+2)$ are not neighbors in x^*. Consequently, they cannot be neighbors in w, because otherwise both $(\sigma^*(i+2), v)$ and $(\sigma^*(i+1), v)$ belong to $E_w(v)\Delta E_{x^*}(v)$, contradicting $\mathcal{B}_n$. Under the induction hypothesis, $\widehat{\sigma}(i+2) = \sigma^*(i+2)$. Hence, u is the only member of $\mathcal{U}$ that is incident to $\widehat{\sigma}(i+2)$ in w and thus the algorithm successfully identifies u as $\sigma^*(i+k+1)$.
2) $|\mathcal{U}| = 1$. In this case the element u in $\mathcal{U}$ must be $\sigma^*(i+k+1)$ because otherwise both $(\sigma^*(i+1), \sigma^*(i+k+1))$ and $(\sigma^*(i+1), u)$ are contained in $E_w(i+1)\Delta E_{x^*}(i+1)$, contradicting $\mathcal{B}_n$.
3) $|\mathcal{U}| = 0$. Under $\mathcal{B}_n$, $(\sigma^*(i+1), \sigma^*(i+k+1))$ is the only member of $E_w(i+1)\Delta E_{x^*}(i+1)$. As a result, $\sigma^*(i+2)$ must be incident to all of its $2k$ true neighbors. These $2k$ neighbors contain $\sigma^*(i+k+1)$, and under $\mathcal{B}_n$, $\sigma^*(i+k+1)$ is the only one that is incident to exactly k members of $\mathcal{V}^c_{labeled}$. Thus the algorithm can always identify $\sigma^*(i+k+1)$.

It remains to show that all three events $\mathcal{A}_n, \mathcal{B}_n, \mathcal{C}_n$ occur with high probability. Under the assumption $\liminf_{n\to\infty}(-2\log\epsilon/\log n) > 1$, there exists some positive constant η such that $\epsilon < n^{-1/2-\eta}$ for large enough n. By (35), $1-p < n^{-1/2-\eta}$ and $q < n^{-3/2-\eta+o(1)}$ for $k \le n^{o(1)}$.

- The event $\mathcal{A}_n$: To show $\mathbb{P}\{\mathcal{A}^c_n\} = o(1)$, note that there are $O(k^2)$ edges in $E_{x^*}(1)$. The probability that one of them does not appear in w is upper bounded by $O(k^2)\cdot(1-p) \le n^{-1/2-\eta+o(1)} = o(1)$. Similarly the probability that an false edge shows up in $E_w(1)$ is at most $O(nk)\cdot q \le n^{-1/2-\eta+o(1)} = o(1)$. Thus $E_{x^*}(1) = E_w(1)$ with high probability.
- The event $\mathcal{B}_n$: Similar as above, $|E_{x^*}(j)\Delta E_w(j)|$ equals in distribution to $X+Y$ with $X \sim \text{Binom}(n_1, 1-p)$, $Y \sim \text{Binom}(n_2, q)$ with $n_1 = O(k^2)$, $n_2 = O(nk)$ and X, Y independent. Thus
$$\begin{aligned}&\mathbb{P}\{|E_{x^*}(j)\Delta E_w(j)| > 1\}\\ =&\mathbb{P}\{X+Y > 1\}\\ \le&\mathbb{P}\{X > 1\} + \mathbb{P}\{Y > 1\} + \mathbb{P}\{X = Y = 1\}.\end{aligned}$$
Using the Binomial distributions of X, Y, the above can be further bounded by
$$\binom{n_1}{2}(1-p)^2 + \binom{n_2}{2}q^2 + n_1 n_2 (1-p) q = o(1/n).$$
By the union bound,
$$\mathbb{P}\{\mathcal{B}^c_n\} \le \sum_{j \le n} \mathbb{P}\{|E_{x^*}(j)\Delta E_w(j)| > 1\} = o(1).$$
- The event $\mathcal{C}_n$: The edge $(\sigma^*(j+1), \sigma^*(j+k+2))$ is not in x^*. Therefore $\mathbb{P}\{w(\sigma^*(j+1), \sigma^*(j+k+2)) = 1\} = q = o(n^{-3/2})$. Thus $\mathbb{P}\{\mathcal{C}^c_n\} = o(1)$ follows from the union bound.

□

Acknowledgment

The authors thank the anonymous reviewers for the valuable feedback, and David Pollard for his comments and suggestions.

References

[1] M. S. Granovetter, "The strength of weak ties," in *Social Networks*. Amsterdam, The Netherlands: Elsevier, 1977, pp. 347–367.

[2] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[3] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[4] M. E. J. Newman, "The structure of scientific collaboration networks," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 2, pp. 404–409, Jan. 2001.

[5] A. Wagner and D. A. Fell, "The small world inside large metabolic networks," *Proc. Roy. Soc. London. B, Biol. Sci.*, vol. 268, no. 1478, pp. 1803–1810, Sep. 2001.

[6] D. S. Bassett and E. Bullmore, "Small-world brain networks," *Neuroscientist*, vol. 12, no. 6, pp. 512–523, Dec. 2006.

[7] R. F. I. Cancho and R. V. Solé, "The small world of human language," *Proc. Roy. Soc. London. B, Biol. Sci.*, vol. 268, no. 1482, pp. 2261–2265, Nov. 2001.

[8] A. E. Motter, A. P. De Moura, Y.-C. Lai, and P. Dasgupta, "Topology of the conceptual network of language," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 65, no. 6, Jun. 2002, Art. no. 065102.

[9] M. E. Newman and D. J. Watts, "Scaling and percolation in the small-world network model," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 60, no. 6, p. 7332, 1999.

[10] J. Kleinberg, "The small-world phenomenon: An algorithm perspective," in *Proc. 32nd Annu. ACM Symp. Theory Comput. (STOC)*, New York, NY, USA, May 2000, pp. 163–170, doi: 10.1145/335305.335325.

[11] C. Moore and M. E. J. Newman, "Epidemics and percolation in small-world networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 61, no. 5, p. 5678, 2000.

[12] J. Saramäki and K. Kaski, "Modelling development of epidemics with dynamic small-world networks," *J. Theor. Biol.*, vol. 234, no. 3, pp. 413–421, Jun. 2005.

[13] P. V. Marsden and K. E. Campbell, "Measuring tie strength," *Social Forces*, vol. 63, no. 2, pp. 482–501, Dec. 1984.

[14] E. Gilbert and K. Karahalios, "Predicting tie strength with social media," in *Proc. 27th Int. Conf. Hum. Factors Comput. Syst. (CHI)*, 2009, pp. 211–220.

[15] E. Gilbert, "Predicting tie strength in a new medium," in *Proc. ACM Conf. Comput. Supported Cooperat. Work (CSCW)*, 2012, pp. 1047–1056.

[16] R. Rotabi, K. Kamath, J. Kleinberg, and A. Sharma, "Detecting strong ties using network motifs," in *Proc. 26th Int. Conf. World Wide Web Companion (WWW Companion)*, 2017, pp. 983–992.

[17] C. Marlow, L. Byron, T. Lento, and I. Rosenn. (Mar. 2009). *Maintained Relationships on Facebook*. Accessed: Sep. 22, 2019. [Online]. Available: https://overstated.net/2009/03/09/maintained-relationships-on-facebook

[18] B. A. Huberman, D. M. Romero, and F. Wu, "Social networks that matter: Twitter under the microscope," *1st Monday*, vol. 14, no. 1, pp. 1–9, 2008. [Online]. Available: https://firstmonday.org/ojs/index.php/fm/article/view/2317

[19] J.-P. Onnela *et al.*, "Structure and tie strengths in mobile communication networks," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 18, pp. 7332–7336, May 2007.

[20] S. Gnerre *et al.*, "High-quality draft assemblies of mammalian genomes from massively parallel sequence data," *Proc. Nat. Acad. Sci. USA*, vol. 108, no. 4, pp. 1513–1518, Jan. 2011.

[21] J. A. Chapman, I. Ho, S. Sunkara, S. Luo, G. P. Schroth, and D. S. Rokhsar, "Meraculous: De novo genome assembly with short paired-end reads," *PLoS ONE*, vol. 6, no. 8, Aug. 2011, Art. no. e23501.

[22] E. Lieberman-Aiden *et al.*, "Comprehensive mapping of long-range interactions reveals folding principles of the human genome," *Science*, vol. 326, no. 5950, pp. 289–293, Oct. 2009.

[23] N. H. Putnam *et al.*, "Chromosome-scale shotgun assembly using an in vitro method for long-range linkage," *Genome Res.*, vol. 26, no. 3, pp. 342–350, Mar. 2016.

[24] V. Bagaria, J. Ding, D. Tse, Y. Wu, and J. Xu, "Hidden Hamiltonian cycle recovery via linear programming," *Oper. Res.*, vol. 68, no. 1, pp. 53–70, Jan. 2020.

[25] T. Cai, T. Liang, and A. Rakhlin, "On detection and structural reconstruction of small-world random networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 4, no. 3, pp. 165–176, Jul. 2017.

[26] B. Hajek, Y. Wu, and J. Xu, "Information limits for recovering a hidden community," *IEEE Trans. Inf. Theory*, vol. 63, no. 8, pp. 4729–4745, Aug. 2017.

[27] P. L. Hammer, U. N. Peled, and X. Sun, "Difference graphs," *Discrete Appl. Math.*, vol. 28, no. 1, pp. 35–44, 1990.

[28] R. Otter, "The number of trees," *Ann. Math.*, vol. 49, no. 3, pp. 583–599, 1948.

[29] A. N. Letchford, G. Reinelt, and D. O. Theis, "Odd minimum cut sets and b-matchings revisited," *SIAM J. Discrete Math.*, vol. 22, no. 4, pp. 1480–1487, Jan. 2008.

[30] M. L. Balinski, "Integer programming: Methods, uses, computations," *Manage. Sci.*, vol. 12, no. 3, pp. 253–313, Nov. 1965.

[31] N. Alon, M. Krivelevich, and B. Sudakov, "Finding a large hidden clique in a random graph," *Random Struct. Algorithms*, vol. 13, nos. 3–4, pp. 457–466, Oct. 1998.

[32] F. McSherry, "Spectral partitioning of random graphs," in *Proc. 42nd IEEE Symp. Found. Comput. Sci.*, Oct. 2001, pp. 529–537.

[33] M. Padberg and G. Rinaldi, "A branch- and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems," *SIAM Rev.*, vol. 33, no. 1, pp. 60–100, Mar. 1991.

[34] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, Jan. 1965.

Jian Ding is currently an Associate Professor with the Department of Statistics, University of Pennsylvania. His research area is probability theory with a focus on interactions with statistical physics and theory of computer science. In particular, his recent research topics include random constraint satisfaction problems, random planar geometry, Anderson localization, and disordered spin models.

Yihong Wu (Member, IEEE) received the B.E. degree from Tsinghua University in 2006 and the Ph.D. degree from Princeton University in 2011. From 2011 to 2012, he was a Post-Doctoral Fellow with the Department of Statistics, The Wharton School, University of Pennsylvania, and an Assistant Professor with the Department of Electronics and Communication Engineering (ECE), University of Illinois at Urbana-Champaign, from 2013 to 2015. He is currently an Associate Professor with the Department of Statistics and Data Science, Yale University. His research interests include the theoretical and algorithmic aspects of high-dimensional statistics, information theory, and optimization. He was a recipient of the NSF CAREER Award in 2017 and the Sloan Research Fellowship in mathematics in 2018.

Jiaming Xu is currently an Assistant Professor with The Fuqua School of Business, Duke University. His research interests include data science, high-dimensional statistical inference, information theory, convex and non-convex optimization, queueing theory, and game theory. He received the Simons-Berkeley Fellowship in 2016.

Dana Yang received the B.S. degree from Tsinghua University in 2013 and the M.A. and Ph.D. degrees from Yale University in 2014 and 2019, respectively. She is currently a Post-Doctoral Associate with The Fuqua School of Business, Duke University. Her research interests include the theoretical aspects of high-dimensional statistics and machine learning. She received the Simons-Berkeley Fellowship in 2021.