Sparsification of the regularized magnetic Laplacian
with multi-type spanning forests

M. Fanuel and R. Bardenet
Université de Lille, CNRS, Centrale Lille
UMR 9189 - CRIStAL, F-59000 Lille, France
{michael.fanuel, remi.bardenet}@univ-lille.fr

September 1, 2022

Abstract

In this paper, we consider a U(1)-connection graph, that is, a graph where each oriented edge is
dowered with a unit modulus complex number which is simply conjugated under orientation flip. A
atural replacement for the combinatorial Laplacian is then the so-called magnetic Laplacian, an Her-
mitian matrix that includes information about the graph’s connection. Connection graphs and magnetic
Laplacians appear, e.g., in the problem of angular synchronization. In the context of large and dense
dgraphs, we study here sparsifiers of the magnetic Laplacian, i.e., spectral approximations based on sub-
dgraphs with few edges. Our approach relies on sampling multi-type spanning forests (MTSFs) using
a custom determinantal point process, a distribution over edges that favours diversity. In a word, an
MTSF is a spanning subgraph whose connected components are either trees or cycle-rooted trees. The
latter partially capture the angular inconsistencies of the connection graph, and thus provide a way to
compress information contained in the connection. Interestingly, when this connection graph has weakly
inconsistent cycles, samples of this distribution can be obtained by using a random walk with cycle pop-
ning. We provide statistical guarantees for a choice of natural estimators of the connection Laplacian,
and investigate the practical application of our sparsifiers in two applications.

Contents

1 Introduction 2
2 Related work 4
3 The magnetic Laplacian and two motivations 5
  3.1 The spectral approach to angular synchronization . . . . . . . . . . . . . . . . . . . . . . . . . 6
    3.1.1 Least eigenvector approximation: sparsify-and-eigensolve . . . . . . . . . . . . . . . . 6
    3.1.2 Preconditioning eigenvalue problem: sparsify-and-precondition . . . . . . . . . . . . . 7
  3.2 Preconditioned magnetic Laplacian systems for semi-supervised learning . . . . . . . . . . . . 7
4 Multi-Type Spanning Forests and determinantal point processes 8
  4.1 A DPP that favors inconsistent cycles . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
  4.2 A few consequences of determinantality . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
5 Statistical guarantees of sparsification 10
  5.1 Sparsification with one MTSF . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
  5.2 Sparsification from a batch of independent MTSFs . . . . . . . . . . . . . . . . . . . . . . . 11
  5.3 Technical results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
6 Cycle-popping sampling for weakly inconsistent graphs 12
1 Introduction

Consider a connected undirected graph of \( n \) nodes and \( m \) edges. A U(1)-connection graph is defined by endowing each oriented edge \( uv \) with a complex phase \( \phi_{uv} = \exp(-i \vartheta(uv)) \) such that \( \vartheta(vu) = -\vartheta(uv) \in [0, 2\pi) \). The corresponding U(1)-connection is the map \( e \mapsto \phi_e \), which associates to an oriented edge a complex phase that is conjugated under orientation flip. The associated magnetic Laplacian matrix is

\[
\Delta = \sum_{euv} w_{uv}(e_u - \phi_{uv}e_v)(e_u - \phi_{uv}e_v)^*,
\]

where \( e_u \) denotes an \( n \times 1 \) vector of the canonical basis associated to node \( u \) and \( w_{uv} \geq 0 \) is the weight of edge \( uv \). The case \( \phi = 1 \) corresponds to the usual combinatorial Laplacian. More generally, the smallest magnetic eigenvalue \( \lambda_{\text{min}}(\Delta) \) is zero iff there exists a real-valued function \( h \) such that \( \vartheta(uv) = h(u) - h(v) \)
for all oriented edges $uv$. In that case, $\Delta$ is unitarily equivalent to the combinatorial Laplacian, and we say that the connection graph is \textit{consistent}, and that the connection is \textit{trivial}. Finally, we note that the magnetic Laplacian is a special case of the vector bundle Laplacian of Kenyon [2011], and a close relative to the \textit{connection Laplacian} [Bandeira et al., 2013], which appears in the context of angular synchronization [Singer, 2011]. Inspired by recent work on ensembles of random spanning trees to sparsify combinatorial Laplacians, see Section 2, and motivated by applications to angular synchronization and graph-based learning, see Section 3, we investigate in this paper approximations of the magnetic Laplacian when the graph is large and dense. More precisely, we propose randomized approximations of $\Delta$ and $\Delta + qI_n$ for any $q > 0$, built by sampling an ensemble of independent spanning subgraphs. The spanning subgraphs we use are \textit{multi-type spanning forests} [MTSF; Kenyon, 2019]. MTSFs are disjoint unions of trees – a connected graph without cycle – and cycle-rooted trees – a connected graph with a unique cycle; see Figure 1d. Section 4 introduces a specific distribution over MTSFs, which builds on recent generalizations of uniform spanning trees. Like the set of edges of a uniform spanning tree, our distribution over MTSFs is a determinantal point process (DPP). DPPs are distributions, here over subsets of edges, that favour diversity and originate in quantum optics [Macchi, 1975; Hough et al., 2006].

Section 5 contains our statistical guarantees for the sparsification of $\Delta$ based on the DPP introduced in Section 4. Our bounds depend on the expected number of edges in the MTSF $d_{\text{eff}} = \text{Tr}(\Delta(\Delta + qI_n)^{-1})$, and on $\kappa = \|\Delta(\Delta + qI_n)^{-1}\|_{\text{op}}$. Informally, we give a sparsifier $\tilde{\Delta}_t$ built with a batch of $t$ independent MTSFs, such that if $d_{\text{eff}}/\kappa$ is large enough and

$$t \gtrsim \frac{\kappa}{\epsilon^2} \log \left( \frac{d_{\text{eff}}}{\kappa \delta} \right),$$

with $\epsilon \in (0, 1)$ then, with probability at least $1 - \delta$,

$$- \epsilon (\Delta + qI) \preceq \Delta - \tilde{\Delta}_t \preceq \epsilon (\Delta + qI);$$

see Theorem 1. As a tool of independent interest, we derive a matrix Chernoff bound with an intrinsic matrix dimension for determinantal point processes, in the spirit of Kaufman, Kyng, and Soldà [2022]. The algorithm we use for sampling MTSFs is an extension of the cycle-popping random walk [Kassel and Kenyon, 2017; Guo and Jerrum, 2021] to the case of a weakly inconsistent $U(1)$-connection graph; see Section 6. In Section 7, we further investigate a self-normalized Monte-Carlo algorithm to deal with larger inconsistencies. In Section 8, we illustrate our results on two case studies: ranking from pairwise comparisons using Sync-Rank [Cucuringu, 2016] and preconditioning regularized Laplacian systems.

The proofs of most of our results about the magnetic Laplacian approximation are deferred to Appendix A. The proof of our Chernoff bound with intrinsic dimension is given in Appendix B. Additional empirical results can be found in Appendix C.

---

1See, e.g., Zhao et al. [2014, Theorem 2.2] or Berkolaiko [2013]; Colin de Verdière et al. [2011]; Fanuel et al. [2018]. In particular, such a function $h$ always exists if the underlying graph is a tree.
Notation. Sets are denoted with a calligraphic font, e.g., $\mathcal{E}$, $\mathcal{V}$, except for $[n] = \{1, \ldots, n\}$. Matrices are denoted by uppercase letters ($\Delta, B, Y$) whereas vectors are typeset as bold lowercase ($x, y, e$). The canonical basis of $\mathbb{R}^n$ is denoted by $(e_n)_{n \in [n]}$, and $\mathbf{1}$ is the all-ones vector. For convenience, we denote the $v$-th entry of $f$ by $f(v)$. The Hermitian conjugate of complex matrix $A$ is written $A^\dagger$, and its transpose is $A^T$. The Moore-Penrose pseudo-inverse of a matrix $A$ is denoted by $A^+$. Also, $|A| = (A^* A)^{1/2}$ denotes the matrix absolute value of the real symmetric matrix $A$. For $A, B$ real symmetric matrices, we write $A \preceq B$ if $x^T A x \leq x^T B x$ for all $x \in \mathbb{C}^n$. We denote by $\text{cond}(A)$ the condition number of a matrix $A$, namely, the ratio of its largest to smallest singular value. The spectrum of a symmetric matrix $A$ is denoted by $\text{Sp}(A)$. The operator norm of $A$ is $\|A\|_{\text{op}} = \max_{\|x\|=1} \|Ax\|_2$. In what follows, $G$ is a connected undirected graph with vertex set $V$ and edge set $E$; for simplicity, denote $n = |V|$ and $m = |E|$. Each edge $e \in E$ is endowed with a weight $w_e > 0$. We denote an oriented edge as $e = vv'$ where $v \in V$ is $e$'s head and $v' \in V$ is $e$'s tail. Note that, by a slight abuse of notation, to denote an edge in $E$ as a pair of nodes, an orientation has to be chosen, and thus $vv' \in E$ and $v'v \in E$ denote the same edge. At this level, the orientation is purely arbitrary, that is, a matter of convention. Let further $1_{S}(\cdot)$ be the indicator function of the set $S$. For conciseness, we finally denote $a \lor b = \max\{a, b\}$ and $a \land b = \min\{a, b\}$.

## 2 Related work

A key question in graph theory concerns approximating the combinatorial Laplacian of a graph by the Laplacian of one of its subgraphs with few edges. The resulting Laplacian is called a sparsifier of the original Laplacian [Batson, Spielman, Srivastava, and Teng, 2013]. Spectral sparsifiers try to preserve properties of the spectrum of the combinatorial Laplacian, whereas cut-sparsifiers intend to preserve the connectivity of the graph, namely a cut, see, e.g., Fung, Hariharan, Harvey, and Panigrahi [2011]. Spectral sparsifiers of the graph Laplacian were defined by Spielman and Teng [2014]. Batson, Spielman, and Srivastava [2014] show the existence of an $(1 \pm \epsilon)$-accurate spectral sparsifier with $O(n/\epsilon^2)$ edges and give a polynomial time deterministic algorithm; see also Zouzias [2012]. These algorithms require $\Omega(n^4)$ running time.

Direct improvements of linear-sized spectral sparsifiers using randomization are given by Allen-Zhu, Liao, and Orecchia [2015]; Lee and Sun [2018] with $O(n^{2+\epsilon})$ and $\Omega(n^2)$ time complexity, respectively. The seminal work of Spielman and Srivastava [2011] showed that, with high probability, a $(1 \pm \epsilon)$-accurate spectral sparsifier of the combinatorial Laplacian is obtained by independent sampling of $O(n \log(n)/\epsilon^2)$ edges. To our knowledge, randomized spectral sparsifiers have shown so far superior performance in practice over their deterministic counterparts.

In the perspective of sampling as few edges as possible, joint edge samples have also been used to yield sparsifiers. An $\epsilon$-accurate cut sparsifier is obtained by a union of $\Theta(\epsilon^{-2} \log^2(n))$ uniform random spanning trees [Fung, Hariharan, Harvey, and Panigrahi, 2011] with high probability. Similarly, Kyng and Song [2018] show that averaging the Laplacians of $O(\epsilon^{-2} \log^2(n))$ uniform random spanning trees yields a $(1 \pm \epsilon)$ spectral sparsifier with high probability. In a recent work, Kaufman, Kyng, and Soldá [2022] improve this result to $O(\epsilon^{-2} \log(n))$ random spanning trees. Also, a short-cycle decomposition approach for the Laplacian sparsification is given by Chu, Gao, Peng, Sachdeva, Sawlani, and Wang [2018].

As we just mentioned, spectral sparsification can be performed by sampling spanning trees. This problem has been studied by using random walks [Aldous, 1990; Broder, 1989; Wilson, 1996]. More recently, Schild [2018] introduced Laplacian solvers within random walks, to improve the shortcutting methods of Madry,

| Abbreviation | Full name                  | Definition                                           |
|--------------|----------------------------|-----------------------------------------------------|
| ST           | spanning tree              | a connected spanning graph without cycle.            |
| SF           | spanning forest            | a spanning graph given by a disjoint union of trees. |
| CRT          | cycle-rooted tree          | a connected subgraph with only one cycle.            |
| CRSF         | cycle-rooted spanning forest| a disjoint union of cycle-rooted trees.              |
| MTSF         | multi-type spanning forest  | a disjoint union of trees and cycle-rooted trees.    |

Table 1: List of abbreviations for subgraphs with their definitions.
Straszak, and Tarnawski [2015]; Kelner and Madry [2009].

The determinantal point processes associated to the edges of a uniform Spanning Tree (ST) are well-known to be associated to the combinatorial Laplacian [Pemantle, 1991]. STs are closely related to random Spanning Forests (SFs) which are in turn associated to the regularized combinatorial Laplacian; see Avena and Gaudillière [2018] for a detailed study of random-walk-based sampling algorithms. Some recent Monte-Carlo approaches for approximating the regularized combinatorial Laplacian thanks to random forests were studied by Pilavci, Amblard, Barthelmé, and Tremblay [2021, 2020]; Barthelmé, Tremblay, Gaudillière, Avena, and Amblard [2020] where variance reduction methods and statistical analyses are also given in the context of graph signal processing. In contrast with our paper, Pilavci, Amblard, Barthelmé, and Tremblay [2021, 2020] consider estimators of the inverse regularized combinatorial Laplacian thanks to the random forests which are in turn associated to the regularized combinatorial Laplacian; see Avena, and Amblard, 2018; Colin de Verdière, Torki-Hamza, and Truc [2011]; Berkolaiko [2013]. It was also used for visualizing directed networks in Fanuel et al. [2018].

Finally, the magnetic Laplacian appears in several works in physics, see, e.g., Colin de Verdière [1998]; Colin de Verdière, Torki-Hamza, and Truc [2011]; Berkolaiko [2013]. It was also used for visualizing directed networks in Fanuel et al. [2018]. In the words of Kenyon [2011], we consider a U(1)-connection such that

\[ \phi_{\pm} \text{ is given by } \phi_{\pm} \equiv \exp(i \varphi_{e} / 2) \]

whereas \( \varphi_{\pm} = \exp(-i \varphi_{e}) \).

3 The magnetic Laplacian and two motivations

Let \( G \) be a connected undirected graph, with vertex set \( V \) and edge set \( E \), of respective cardinality \( n = |V| \) and \( m = |E| \). In what follows, we often abuse these notations and denote the vertex set by \([n]\). It is customary to define the oriented edge-vertex incidence matrix \( B_0 \in \mathbb{R}^{m \times n} \) as

\[
B_0(e,v) = \begin{cases} 
1 & \text{if } v \text{ is } e \text{'s head,} \\
-1 & \text{if } v \text{ is } e \text{'s tail,} \\
0 & \text{otherwise,}
\end{cases}
\]

and the combinatorial Laplacian as

\[
\Lambda = B_0^T W B_0 \in \mathbb{R}^{n \times n},
\]

where \( W = \text{Diag}(w) \in \mathbb{R}^{m \times m} \) is a diagonal matrix containing non-negative edge weights. Obviously, different choices of orientation of the edges yield different \( B_0 \) matrices, but they all produce the same Laplacian matrix.

The matrix \( B_0 \) is a representation of a discrete derivative for a given ordering of the nodes and an ordering of the edges with an arbitrary orientation. More precisely, for an oriented edge \( e = uv \), the discrete derivative of a function \( f \) on \( V \) in the direction \( e \) reads \( d_0 f(e) \equiv f(u) - f(v) = (e_u^T - e_v^T)f = \sum_{w \in [u]} B_0(e,w)f(w) \).

We now twist this construction by following the definition of Kenyon [2011], that we particularize to the group of unit-modulus complex numbers \( U(1) \). Let \( \phi_{ue} = \phi_{eu}^{-1} \in U(1) \) and define \( \phi_{uv'} = \phi_{eu} \phi_{ue} \). Consider the twisted oriented and complex incidence matrix \( B \in \mathbb{C}^{m \times n} \) given by

\[
B(e,v) = \begin{cases} 
\phi_{ue} & \text{if } v \text{ is } e \text{'s head,} \\
-\phi_{eu} & \text{if } v \text{ is } e \text{'s tail,} \\
0 & \text{otherwise.}
\end{cases}
\]

The matrix \( B \) corresponds to a representation of the twisted discrete derivative.\(^2\)

For any oriented edge \( e = uv \), the twisted derivative of \( f \) in the direction \( e \) is given by \( df(e) \equiv \phi_{ue} f(u) - \phi_{ue} f(v) = (\phi_{eu} e_{u} - \phi_{eu} e_{u})^* f = \sum_{w \in [u]} B(e,w)f(w) \). Similarly to the combinatorial Laplacian (2), the magnetic Laplacian is defined as

\[
\Delta = B^* W B \in \mathbb{C}^{n \times n},
\]

\(^2\)In the words of Kenyon [2011], we consider a \( U(1) \)-connection such that \( \phi_{uv'} = \exp(-i \varphi_{uv'}) \), and \( \phi_{uv'} = \exp(-i \varphi_{uv'}) / 2 \) whereas \( \varphi_{uv} = \exp(i \varphi_{uv'}) / 2 \) for \( e = uv' \). Our definition of the (twisted) discrete derivative differs from Kenyon [2011] by a minus sign.
with, again, a diagonal edge weight matrix \( W \in \mathbb{R}^{m \times m} \) with non-negative entries. Thus, by construction, the spectrum of \( \Delta \) is non-negative. As anticipated in Section 1 and in the case of a connected graph, the least eigenvalue of \( \Delta \) vanishes iff the corresponding connection graph is trivial. In contrast, \( \Lambda \) has always an eigenvalue equal to zero.

In this paper, we study spectral sparsifiers of the magnetic Laplacian \( \Delta \) (3), i.e., for \( \epsilon > 0 \), any (typically cheap to compute and inverse) matrix \( \tilde{\Delta} \) such that the \( \epsilon \)-multiplicative guarantee (1) holds. In this case, \( \tilde{\Delta} \) can be used to approximate \( \Delta \) in problems that involve the eigenpairs of \( \Delta \). We now introduce two such applications, which serve as our motivation and empirical benchmark: the spectral approach to angular synchronization and preconditioning Laplacian systems.

### 3.1 The spectral approach to angular synchronization

Denoting by \( \vartheta(uv) = \arg(\phi_{uv}) \), angular synchronization is the problem of finding \( h : \mathcal{V} \to [0, 2\pi) \) such that
\[
\vartheta(uv) \approx h(u) - h(v) \quad \text{for all} \quad uv \in \mathcal{E}.
\] (4)

As explained below, this problem is related to the magnetic Laplacian that we sparsify in this paper. Applications range from synchronizing distributed networks [Cucuringu et al., 2012a] to image reconstruction from pairwise differences [Yu, 2009], or robust ranking from pairwise comparisons [Cucuringu, 2016]; see Section 8.2 for details on the latter. Although we focus on \( U(1) \) in this paper, different groups has also been considered, e.g. synchronization using \( SO(3) \) for identifying the 3D structure of molecules [Cucuringu et al., 2012b].

Following the seminal paper by Singer [2011], the rationale of the spectral approach to angular synchronization goes as follows. Synchronization (4) can be achieved by solving
\[
\max_{h \in [0,2\pi]^n} \sum_{uv \in \mathcal{E}} \cos \left( \vartheta(uv) + h(v) - h(u) \right),
\] (5)

In particular, the cosine loss function is approximately quadratic for small argument values, and promotes robustness as it saturates for large argument values. Several relaxations of the above problem are obtained by noting that \( \cos(t) = \text{Re} \left( e^{it} \right) \). Consider the following spectral relaxation
\[
\max_{f \in \mathbb{C}^n : \|f\|^2 = n} \sum_{uv \in \mathcal{E}} w_{uv} \text{Re} \left( f(u)^* e^{i \vartheta(uv)} f(v) \right),
\] (6)

where we have further introduced weights such that \( w_{uv} \geq 0 \). Once a solution \( f \) to (6) has been found, the angle estimates \( \arg(f(u)) \) are set to the argument of each entry of \( f \), with the convention that \( \arg(0) = 0 \).

We now relate this problem to an eigenvalue problem for the magnetic Laplacian. Indeed, consider the spectral problem
\[
\min_{\|f\|^2 = n} \sum_{uv \in \mathcal{E}} w_{uv} \left| f(u) - e^{i \vartheta(uv)} f(v) \right|^2,
\] (7)

where the objective is proportional to the quadratic form defined by the magnetic Laplacian (3). The objectives of (6) and (7) relate as
\[
\sum_{uv \in \mathcal{E}} w_{uv} \left| f(u) - e^{i \vartheta(uv)} f(v) \right|^2 = -2 \sum_{uv \in \mathcal{E}} w_{uv} \text{Re} \left( f(u)^* e^{i \vartheta(uv)} f(v) \right) + \sum_{uv \in \mathcal{E}} w_{uv} \left( |f(u)|^2 + |f(v)|^2 \right).
\]

Since the last term above is constant when all the components of \( f \) have the same modulus, namely if \( f(u) = \text{cst} \cdot e^{i h(u)} \) for all \( u \in [n] \), (7) is indeed another relaxation of (5).

#### 3.1.1 Least eigenvector approximation: sparsify-and-eigensolve

Solving (7) amounts to finding the least eigenspace of the magnetic Laplacian, which we assume to have dimension one in this paper. We now explain how a spectral sparsifier (1) can be used to approximate this eigenvector with the smallest eigenvalue. This approach may be called sparsify-and-eigensolve, in the spirit of sketch-and-and-solve methods; see Martinsson and Tropp [2020, Section 10.3].
Assume w.l.o.g. that the respective least eigenvectors of $\Delta$ and $\tilde{\Delta}$ satisfy $\|f_1\|_2 = \|\tilde{f}_1\|_2 = \sqrt{n}$ in light of the spectral relaxation (6). Let $\delta_\star = \lambda_2(\Delta) - \lambda_1(\Delta)$ be the spectral gap in the Laplacian spectrum, where $\lambda_1(\Delta) \leq \cdots \leq \lambda_n(\Delta)$ are the eigenvalues of $\Delta$. If $\|\Delta - \tilde{\Delta}\|_{\text{op}} < \delta_\star$, then the Davis-Kahan theorem guarantees that
\[
\text{dist}(f_1, \tilde{f}_1) = \min_{\theta \in [0, 2\pi]} \|f_1 - \tilde{f}_1 e^{i\theta}\|_2 \leq \frac{\sqrt{2}\|\Delta - \tilde{\Delta}\|_2}{\delta_\star - \|\Delta - \tilde{\Delta}\|_{\text{op}}} \leq \frac{\sqrt{2}\|\Delta - \tilde{\Delta}\|_\infty}{\delta_\star - \|\Delta - \tilde{\Delta}\|_{\text{op}}} \sqrt{\frac{2}{n}},
\]
see Zhong and Boumal [2018, Lemma 11]; see also Yu et al. [2015]; Davis and Kahan [1970] for more general statements. Now by definition (1), $\|\Delta - \tilde{\Delta}\|_{\text{op}} \leq \epsilon\|\Delta\|_{\text{op}}$. Thus, if $\epsilon\|\Delta\|_{\text{op}} < \delta_\star$, we obtain
\[
\text{dist}(f_1, \tilde{f}_1) \leq \frac{\epsilon\lambda_n(\Delta)}{\delta_\star - \epsilon\lambda_n(\Delta)} \sqrt{\frac{2}{n}},
\]
so that computing the least eigenvector of $\tilde{\Delta}$ is a controlled replacement for that of $\Delta$. Note that by analogy with the sketch-and-precondition paradigm in randomized linear algebra [Martinsson and Tropp, 2020, Section 10.5]. Sparsify-and-precondition assumes the knowledge of an approximation $\hat{\lambda}_1$ of $\lambda_1$. The least eigenvector is then approximated by the solution of the linear system $\langle \Delta + \hat{\lambda}_1 I, f \rangle = 0$; see Argentati et al. [2017] for more details. Since the convergence rate of iterative linear solvers depends on the condition number of the matrix defining the system, a common technique is to rather solve $T^{-1}(\Delta - \hat{\lambda}_1 I)f = 0$, when $T$ is a well-chosen positive definite matrix, called a preconditioner, such that $\text{cond}(T^{-1}\Delta)$ is smaller than $\text{cond}(\Delta)$ [Knyazev and Neymeyr, 2003].

In this paper, we naturally choose $T = \tilde{\Delta}$ for which derive the multiplicative guarantee (1) with high probability. Incidentally, a preconditioner for another eigenvalue solver, called the Lanczos method, was also discussed earlier by Morgan and Scott [1993]; Morgan [2000].

3.1.2 Preconditioning eigenvalue problem: sparsify-and-precondition

Still in the context of angular synchronization, the spectral sparsifier can also be used to build a preconditioner for computing the least eigenpair $\langle \lambda_1, f_1 \rangle$ of $\Delta$. We dub this approach sparsify-and-precondition, by analogy with the sketch-and-precondition paradigm in randomized linear algebra [Martinsson and Tropp, 2020, Section 10.5]. Sparsify-and-precondition assumes the knowledge of an approximation $\hat{\lambda}_1$ of $\lambda_1$. The least eigenvector is then approximated by the solution of the linear system $\langle \Delta - \hat{\lambda}_1 I, f \rangle = 0$; see Argentati et al. [2017] for more details. Since the convergence rate of iterative linear solvers depends on the condition number of the matrix defining the system, a common technique is to rather solve $T^{-1}(\Delta - \hat{\lambda}_1 I)f = 0$, when $T$ is a well-chosen positive definite matrix, called a preconditioner, such that $\text{cond}(T^{-1}\Delta)$ is smaller than $\text{cond}(\Delta)$ [Knyazev and Neymeyr, 2003].

In this paper, we naturally choose $T = \tilde{\Delta}$ for which derive the multiplicative guarantee (1) with high probability. Incidentally, a preconditioner for another eigenvalue solver, called the Lanczos method, was also discussed earlier by Morgan and Scott [1993]; Morgan [2000].

3.2 Preconditioned magnetic Laplacian systems for semi-supervised learning

Outside angular synchronization, the approximation $\tilde{\Delta}$ can also serve as a preconditioner for more general Laplacian systems; see Vishnoi [2013, Section 17] and references therein. In the context of semi-supervised learning [Zhou et al., 2004], optimization problems of the following form
\[
f_\star \in \arg \min_{f \in \mathbb{R}^n} f^\top \Delta f + r(f, y),
\]
are often considered when only a few outputs are given for training, namely, $y_{\text{train}, \ell}$ are associated to a subset of nodes $u_\ell$ for $1 \leq \ell \leq l$. Here $r(f, y)$ is a risk between $f$ and the vector $y$ which stores label information as follows: $y(u_\ell) = y_{\text{train}, \ell}$ for $1 \leq \ell \leq l$ and $y(u) = 0$ otherwise. Solving one of these problems yields a prediction on the unlabelled nodes, possibly after a rounding step such as $\arg(f_\star)$.

If we choose $r(f, y) = q\|f - y\|_2$ with a parameter $q > 0$, the first order optimality condition is the regularized Laplacian system $\langle \Delta + qI_n, f \rangle = qy$. If $\epsilon \in (0, 1)$ and the following guarantee holds
\[
(1 - \epsilon)(\Delta + qI_n) \preceq \tilde{\Delta} + qI_n \preceq (1 + \epsilon)(\Delta + qI_n),
\]
then $\text{Sp}((\tilde{\Delta} + qI_n)^{-1}(\Delta + qI_n))$ is in the interval $[(1 + \epsilon)^{-1}, (1 - \epsilon)^{-1}]$. Finding a sparsifier $\tilde{\Delta}$ satisfying (10) thus guarantees a well-conditioned linear system.
In the context of angular synchronization, Cucuringu [2016, Section 7] also discusses a semi-supervised approach for ranking from pairwise comparisons with anchors, i.e., with a few nodes for which the ranking is given a priori. Thus, the problem (9) including the magnetic Laplacian can also be considered for ranking with anchors.

Moreover, from the computational perspective, solving the preconditioned linear system might also be fast if $\Delta$ is sparse. In Appendix A.2, we discuss an ordering of the nodes so that the Cholesky factorization $R^t R = \Delta + q I_n$ is very sparse when the sparsifier is associated with one multi-type spanning forest. Since the triangular matrix $R$ is sparse, the system $Rf = b$ can be solved quickly. In particular, in the case $q = 0$ and for a non-trivial connection, a sparsifier $\Delta$ can be obtained by sampling one CRSF. Denote by $V_c$ the set of nodes in the cycles of this CRSF. The Cholesky factorization of $\Delta$ yields a triangular matrix $R$ with $O(n + |V_c|)$ non-zero off-diagonal entries, and is obtained in $O(n + |V_c|)$ operations; see Proposition 3. Thus, the linear system $Rf = b$ can be solved in $O(n + |V_c|)$ operations.

Finally, we note that another possible choice for the risk in (9) is $r(f, y) = -2 \text{Re}(f^t y)$, whose first-order optimality conditions yield an unregularized Laplacian system as first order optimality conditions. Equality constraints can also be imposed by taking the risk as the characteristic function of the constraint set.

4 Multi-Type Spanning Forests and determinantal point processes

We now describe the distribution we sample from to build our sparsifiers satisfying (1) with high probability. To simplify the expressions, we assume hereinafter that all edge weights are equal to 1, i.e., $W = I_m$.

4.1 A DPP that favors inconsistent cycles

For each cycle-rooted tree in an MTSF, with cycle $\eta$, define the monodromy of $\eta$ by

$$m(\eta) = \prod_{v \in F} \phi_{vv'},$$

(11)

We consider the distribution over MTSFs $F$ given by

$$p(F) = \frac{q^{\frac{|\rho(F)|}}}{\det(\Delta + q I_n)} \prod_{\eta \in \text{cycles}(F)} \left(2 - 2 \cos(\arg(m(\eta)))\right),$$

(12)

where $|\rho(F)|$ is the number of trees of the MTSF $F$. Definition (12) calls for comments. First, although the calculation of the monodromy $m(\eta)$ depends on the orientation of the cycle $\eta$, we note that $\cos(\arg(m(\eta)))$ is invariant under orientation flip. In particular, the measure (12) favours MTSFs with inconsistent cycles, namely, cycles $\eta$ such that $1 - \text{Re} m(\eta)$ is large. Second, a small value of $q > 0$ promotes MTSFs with more cycle-rooted trees than trees. Third, for a trivial connection $\phi_{vv'} = 1$, (12) is a measure on spanning forests [SF; Avena and Gaudilli`ere, 2018] (see Figure 1b) and, in the limit $q \to 0$, we recover the uniform measure on spanning trees [ST; Pemantle, 1991; Lyons, 2003]; see Figure 1a. Fourth, for a non-trivial connection, the limit $q \to 0$ gives a measure on cycle-rooted spanning forests [CRSFs; Kenyon, 2011, Section 3]. A CRSF is nothing else than an MTSF where all the connected components are cycle-rooted trees; see Figure 1c. In the most general case of a non-trivial connection with $q > 0$, it is easy to check that the number of edges in an MTSF $F$ is $|F| = n - |\rho(F)|$. In the case $q = 0$, $F$ is almost surely a CRSF and $|F| = n$.

The measure (12) on MTSFs is actually associated with a discrete determinantal point process (DPP) on the edges of the graph. A discrete DPP is a distribution over subsets of a finite ground set $[m] = \{1, \ldots, m\}$ that favours diverse sets, as measured by a kernel matrix. We restrict here to DPPs with symmetric kernels.

**Definition 1** (Discrete DPP). Let $K$ be an $m \times m$ Hermitian matrix with eigenvalues within $[0, 1]$, called a correlation kernel. We say that $F \sim \text{DPP}(K)$ if

$$\Pr(A \subseteq F) = \det(K_{AA}),$$

for all $A \subseteq [m]$. If $K$ has all its eigenvalues being equal to either 0 or 1, DPP($K$) is called projective.
The distribution in Definition 1 is characterized by its so-called \textit{inclusion} probabilities. Its existence is guaranteed by the Macchi-Soshnikov theorem [Macchi, 1975; Soshnikov, 2000]. When the spectrum of $K$ is further restricted to lie within $[0,1)$, the DPP is called an L-ensemble, and the probability mass function actually takes a simple form [Macchi, 1975]. For an exhaustive account on discrete DPPs, we refer to Kulesza and Taskar [2012].

**Proposition 1** (L-ensemble). Let $F \sim \text{DPP}(K)$, with $\text{Sp}(K) \subset [0,1)$. Let $L = (I - K)^{-1}K$ be the so-called \textit{likelihood kernel} of the DPP. Then

$$\Pr(F) = \frac{\det(L_{FF})}{\det(L + I)}.$$  \hfill (13)

We now show that (12) is a DPP. Let $q > 0$. In what follows, we simply set $w_e = 1$ for all $e \in E$ to simplify mathematical expressions. We define an L-ensemble with correlation kernel

$$K = B (\Delta + qI_n)^{-1} B^* = q^{-1}BB^* (q^{-1}BB^* + I_m)^{-1},$$  \hfill (14)

where we read off the likelihood kernel $L = q^{-1}BB^*$. A direct calculation shows that this DPP is precisely the distribution we introduced in (12), i.e., $\Pr(F) = p(F)$. More precisely, using the definition of an L-ensemble and Sylvester’s identity $\det(q^{-1}BB^* + I_m) = \det(q^{-1}B^*B + I_n)$, we can rewrite (13) as

$$\Pr(F) = q^{\rho(F)} \frac{\det(B_{\mathcal{F}}B_{\mathcal{F}}^*)}{\det(\Delta + qI_n)}.$$

The product of factors involving the monodromies in (12) arise from the use of Cauchy-Binet identity applied to $\det(B_{\mathcal{F}}B_{\mathcal{F}}^*)$; see Kenyon [2011] for a complete discussion.

### 4.2 A few consequences of determinantality

From Definition 1 the marginal inclusion probability of any edge is given by

$$\Pr(e \in \mathcal{F}) = K_{ee} \triangleq \mathbb{I}(e),$$  \hfill (15)

which is the so-called leverage score of $e \in \mathcal{E}$. Leverage scores play an important role in the seminal work of Spielman and Srivastava [2011], who build spectral sparsifiers using i.i.d. edge sampling. In this sense, the DPP (12) generalizes the latter construction, where off-diagonal entries of $K$ encode negative dependence between edges.

The expansion of the normalizing constant of (12), see Kenyon [2019, Theorem 2.4] and originally derived for CRSFs by Forman [1993], is given by

$$\det(\Delta + qI_n) = \sum_{\text{MTSFs } \mathcal{F}} q^{\rho(\mathcal{F})} \prod_{\text{cycles } \eta \in \mathcal{F}} \left(2 - m(\eta) - 1/m(\eta)\right),$$  \hfill (16)

where the sum is over all non-oriented MTSFs, and $m(\eta)$ are $1/m(\eta)$ are the monodromies of the two possible orientations of the cycle $\eta$. We refer to Kassel and Lévy [2020] for a generic proof technique.
As another consequence of the determinantal structure [Kulesza and Taskar, 2012] of the distribution (12) is that the expected number of edges in an MTSF and its variance are given in closed form as

\[
E_{F \sim \text{DPP}(K)}[|F|] = \text{Tr} \left( \Delta + qI_n \right)^{-1}, \quad \text{and} \quad \forall F \sim \text{DPP}(K)[|F|] = \text{Tr} \left( \Delta + qI_n \right)^{-2}.
\]

In particular, this confirms that the expected number of edges culminates at \( n \) for \( q \to 0 \) (CRSF) and goes to zero as \( q \to +\infty \). In other words, a large value of \( q \) tends to promote MTSFs with small trees.

**Remark 1** (\( q = 0 \)). In the case of a non-trivial connection, the correlation kernel (14) for \( q = 0 \) is the orthogonal projector \( B\Delta^{-1}B^* \). The corresponding DPP is then a projective DPP whose samples are CRSFs; see Figure 1c for an illustration and Kenyon [2011] for a reference. This DPP samples sets of constant cardinality.

In the absence of any specific structure, sampling a discrete DPP can be done thanks to a generic algorithm [Hough et al., 2006], which relies on the eigendecomposition of the correlation kernel (14), followed by a linear algebraic procedure analogous to a chain rule. The time complexity of this algorithm is \( \mathcal{O}(m^3) \) where \( m \) is the number of edges. For the specific case of (14), in the vein of Wilson [1996] and Kassel and Kenyon [2017], we show in Section 6 how to leverage the graph structure to obtain a cheaper but still exact sampling random walk-based algorithm.

### 5 Statistical guarantees of sparsification

We now give our main statistical results, guaranteeing that the DPP of Section 4 yields a controlled sparse approximation of the magnetic Laplacian. Throughout this section, we work under the following assumption.

**Assumption 1** (Non-singularity of \( \Delta + qI_n \)). Let a connected graph be endowed with a \( U(1) \)-connection and denote by \( \Delta + qI_n \) its (regularized) magnetic Laplacian with \( q \geq 0 \). We assume either that

1. the \( U(1) \)-connection is non-trivial (cf. Table 2), or
2. \( q > 0 \).

Under Assumption 1, \( \Delta + qI_n \) has only strictly positive eigenvalues since either \( \Delta \) is strictly positive definite or \( q > 0 \).

#### 5.1 Sparsification with one MTSF

For an MTSF \( F \), we define the \(|E| \times |E|\) sampling matrix \( S(F) \) by

\[
S_{ee'}(F) = \delta_{ee'}1_F(e')/\sqrt{l(e)},
\]

where \( l(e) \) is the leverage score of the edge \( e \in E \); see (15). This matrix is obtained from the identity matrix by dividing each column indexed by some \( e \in F \) by \( \sqrt{l(e)} \) and filling with zeros all the remaining columns. By construction, under the DPP with correlation kernel (14), this sampling matrix satisfies \( E_F[SS^T] = I_m \), since \( l(e) = \text{Pr}(e \in F) \). Consider now

\[
\tilde{\Delta}(F) = B^*SS^TB. \tag{17}
\]

We emphasize that \( \tilde{\Delta} \) is the Laplacian of a weighted graph with fewer edges than the original graph, thus motivating the word “sparsification”. The weights of the edges in the sparsified graph can be read from the non-zero (diagonal) elements of \( SS^T \).

**Proposition 2** (Sparsification from one MTSF). Let Assumption 1 hold and \( K = B(\Delta + qI_n)^{-1}B^* \). Let \( F \sim \text{DPP}(K) \), and \( \tilde{\Delta}(F) \) be given by (17). Let further \( \kappa = \|K\|_{\text{op}} \) and \( d_{\text{eff}} = \text{Tr}(K) \) be the average number of edges sampled by DPP(\( K \)). Finally, let \( \delta \in (0, 1) \), and \( \epsilon > 0 \) such that

\[
e^2 = 37\kappa \left( 2\log \left( \frac{4d_{\text{eff}}}{\delta \kappa} \right) + \sqrt{3} \right).
\]

With probability at least \( 1 - \delta \),

\[
(1 - \epsilon)(\Delta + qI_n) \preceq \tilde{\Delta}(F) + qI_n \preceq (1 + \epsilon)(\Delta + qI_n).
\]
The proof is given in Appendix A.1 and is based on a new matrix Chernoff bound given in Theorem 2.

It is well-known [Vishnoi, 2013, Theorem 13.3] that we can construct a Cholesky decomposition of a suitably permuted Laplacian matrix of a spanning tree with at most \( O(n) \) non-zero off-diagonal entries. Similarly, we can construct a very sparse Cholesky decomposition of the regularized magnetic Laplacian matrix of an MTSF thanks to a suitable node ordering. The sparsity of this decomposition depends on the set of nodes in the cycles of the MTSF, that we denote by \( \mathcal{V}_c \). In a word, the resulting triangular matrix has \( O(n + |\mathcal{V}_c|) \) non-zero off-diagonal entries, and is obtained in \( O(n + |\mathcal{V}_c|) \) operations. This is stated more formally in Proposition 3.

**Proposition 3** (Sparse Cholesky decomposition for the Laplacian of one MTSF). Consider an MTSF with \( r \) rooted trees and \( c \) cycle-rooted trees, with cycle-lengths \( n_1, \ldots, n_c \). Let \( \tilde{\Delta} \) be the magnetic Laplacian of this MTSF endowed with a unitary connection and let \( q \geq 0 \). There exists an ordering of the nodes of this MTSF such that the Cholesky factor of \( \tilde{\Delta} + qI \) has at most \( n - r + (n_1 - 3) + \cdots + (n_c - 3) \) non-zero off-diagonal entries. This Cholesky factorization then requires \( O(n + n_1 + \cdots + n_c) \) operations.

In particular, Proposition 3 implies that given \( R \) is a triangular matrix with at most \( O(n + |\mathcal{V}_c|) \) non-zero off-diagonal entries, the linear system \( Rf = y \) can be solved in \( O(n + |\mathcal{V}_c|) \) in time. The proof of Proposition 3 given in Appendix A.2 also gives a specific recipe to construct the permutation of the nodes such that \( R \) is sparse.

### 5.2 Sparsification from a batch of independent MTSFs

The value of \( \epsilon \) in the multiplicative bound from Proposition 2 can be large. To take it down, we now consider a batch (or ensemble), \( \mathcal{F}_1, \ldots, \mathcal{F}_t \) of \( t \) MTSFs, drawn independently from the same DPP as in Proposition 2. The corresponding average of sparse Laplacians,

\[
\frac{1}{t} \sum_{\ell=1}^t \tilde{\Delta}(\mathcal{F}_\ell) = B^* S^{(b)} S^{(b)^T} B,
\]

is itself the Laplacian of a sparse graph, with edge set \( \cup_{\ell=1}^t \mathcal{F}_\ell \), with suitably chosen edge weights. Note that each edge in the batch is a multi-edge. The \( |\mathcal{E}| \times |\mathcal{E}| \) sampling matrix of the batch is diagonal with entries given by

\[
S^{(b)}_{ee'} = \delta_{ee'} \sqrt{\frac{n(e)}{t \times 1(e)}} 1_{\cup_{\ell=1}^t \mathcal{F}_\ell}(e'),
\]

where \( n(e) \) is the number of times edge \( e \) appears in the batch of MTSFs.

**Theorem 1** (Sparsification with a batch of MTSFs). Let Assumption 1 hold. Let \( \epsilon \in (0, 1) \) and denote by \( \mathcal{F}_1, \ldots, \mathcal{F}_t \) MTSFs drawn independently from DPP(\( K \)) with \( K = B (\Delta + qI_n)^{-1} B^* \). Let \( \kappa = \|K\|_{op} \) and \( d_{\text{eff}} = \text{Tr}(K) \) the average number of edges sampled by this DPP. Denote by \( \tilde{\Delta}(\mathcal{F}_\ell) \) for \( 1 \leq \ell \leq t \) the corresponding sparsified magnetic Laplacian.

Let \( \delta \in (0, 1) \). Then, if

\[
t \geq \frac{37\kappa}{\epsilon^2} \left( 2 \log \left( \frac{4d_{\text{eff}}}{\delta \kappa} \right) \vee \sqrt{3} \right),
\]

the following event

\[
(1 - \epsilon)(\Delta + qI_n) \preceq \frac{1}{t} \sum_{\ell=1}^t \tilde{\Delta}(\mathcal{F}_\ell) + qI_n \preceq (1 + \epsilon)(\Delta + qI_n),
\]

happens with probability at least \( 1 - \delta \).

The proof of Theorem 1 is given in Appendix A.3. This result is similar to the conclusion of Kaufman et al. [2022] who show that \( \mathcal{O}\left(\epsilon^{-2} \log n\right) \) random spanning trees give a \( (1 \pm \epsilon) \) sparsifier of the combinatorial Laplacian. Let us describe the effect of choosing a large value of \( q \) on the lower bound on \( t \). First, \( \kappa \) decreases as \( q \) increases and the average number of edges \( d_{\text{eff}} \leq \kappa n \) also decreases as \( q \) increases, as already mentioned

\[
11
\]
in Section 4. Hence, intuitively, approximating the $q$-regularized Laplacian is easier if $q$ takes a large value, say, w.r.t. $\lambda_{\max}(\Delta)$. Notice that the expected runtime for sampling an MTSF for a large value of $q$ is also expected to be shorter, as discussed below in Remark 3.

**Remark 2** (Role of the diagonal term). One might wonder why we include the constant diagonal $qI_n$ in the multiplicative bound \(10\). Indeed, if $\epsilon \in (0,1)$ and if we have the multiplicative bound \((1 - \epsilon)\Delta \preceq \Delta(F) \preceq (1 + \epsilon)\Delta\), then we also have the same multiplicative bound for $\Delta(F) + qI_n$. But we argue that there is interest in directly sparsifying $\Delta + qI_n$. In practice, we expect that including this additional diagonal term in the sparsifier also yields faster sampling, as the algorithm in Section 6 is tailored to the structure of the regularized Laplacian. At this stage, we can already mention prior work that states that, for $q > 0$ and in the particular case of the combinatorial Laplacian, sampling a random spanning forest with a natural cycle-popping algorithm has an average running time $O(n/q)$ [Barthelmé et al., 2020; Pilavci et al., 2020; Avena and Gaudilliè, 2018].

5.3 Technical results

Proposition 2 and Theorem 1 rely on a new matrix Chernoff bound, with intrinsic dimension, for a sum of positive semidefinite matrices. We state it here because it might be of independent interest.

**Theorem 2** (Chernoff bound with intrinsic dimension). Let $X \sim$ DPP on $[m]$. Let $Y_1,\ldots,Y_m$ be $d \times d$ Hermitian matrices such that $0 \preceq Y_i \preceq rI$ for all $i \in [n]$ and some $r > 0$. Assume $E_X\left[\sum_{i \in X} Y_i\right] \preceq M$ for some Hermitian matrix $M$. Let $\text{intdim}(M) = \text{Tr}(M)/\|M\|_{\text{op}}$ and $\kappa = \|M\|_{\text{op}}$. Then, for all $\epsilon \in (0,1]$, it holds

$$\Pr \left[ \left\| \sum_{i \in X} Y_i - E_X\left[ \sum_{i \in X} Y_i\right] \right\|_{\text{op}} \geq \epsilon \kappa \right] \leq 2\text{intdim}(M) \left( 1 + c_1 \frac{\epsilon^2}{\kappa^2 \epsilon^4} \right) \exp \left( -c_2 \frac{\kappa^2 r}{r} \right),$$

where $c_1 = 3 \cdot 37^2$ and $c_2 = \frac{1}{2 \cdot 37}$.

We give a proof of Theorem 2 in Appendix B. The proof techniques are borrowed from Kaufman et al. [2022] and adapted to account for a matrix intrinsic dimension à la Tropp [2015]. Let us discuss the interpretation of this result.

An obvious limitation of the bound in Theorem 2 is that it describes the concentration only for $\epsilon$ sufficiently small, whereas the bounds obtained by Tropp [2015] for i.i.d. sampling do not suffer from this limitation. Despite this fact, it is still possible to use this result to prove, e.g., Theorem 1. The upper bound in Theorem 2 can be made small provided that the parameter $r$ is small enough and that $\kappa = \|M\|_{\text{op}}$ remains constant. The latter condition is enforced by making the DPP depend on $r$ so that the expected cardinality of $X$ grows if $r$ goes to zero. Note that the upper bound on the failure probability decreases if the ratio $\kappa^2 r/\epsilon$ increases. Thus, for a fixed $\epsilon \in (0,1]$, the upper bound on the failure probability will take a small value as $r$ goes to zero and as the expected number of sampled points increases; see the proof of Theorem 1 for more details.

6 Cycle-popping sampling for weakly inconsistent graphs

To make our statistical guarantees of Section 5 practical, we need a sampling algorithm for the DPP with kernel \(14\). As mentioned in Section 4, any DPP with a symmetric correlation kernel can be sampled using a mixture argument and a generic linear algebraic chain rule; see Hough et al. [2006, Alg. 18] for the original argument, as well as the discussions around Lavancier et al. [2015, Alg. 1] and Kulesza and Taskar [2012, Alg. 1]). However, this generic approach requires an eigendecomposition of the correlation kernel. In the case of \(14\), we show here how to leverage the structure of the probability distribution \(12\) in order to avoid any eigendecomposition, provided that the connection graph is not too inconsistent. The algorithm described in this section is a simple extension of the work of Wilson [1996] and Kassel and Kenyon [2017]. The main idea is to run a random walk on the nodes of the graph, with an additional absorbing node, popping cycles with some probability as they appear. After post-processing, one obtains an MTSF with the desired distribution. We first discuss a seemingly necessary assumption on the connection, and then the different components of the algorithm, before concluding on its running time.
Oriented CRSF.

Figure 2: Different oriented spanning subgraphs of a grid graph: an oriented Cycle-Rooted Spanning Forest (CRSF), an oriented Multi-Type Spanning Forest (MTSF) and an oriented Spanning Forest (SF). The root of an oriented tree (in blue) is the node with no out-going edge.

Limitation to weakly inconsistent cycles. Define for simplicity \( \theta(\eta) = \arg(m(\eta)) \) the cumulated angle along a cycle \( \eta \), where the monodromy \( m(\eta) \) is defined in (11). Kassel and Kenyon [2017] proposed a variant of the celebrated cycle-popping algorithm of Wilson [1996] for efficiently sampling CRSFs, i.e., joint edge samples w.r.t. the DPP with kernel (14) with \( q = 0 \). In this section, we further adapt this idea to sample MTSFs, i.e., \( q \geq 0 \) in (14). To our knowledge, this cycle-popping algorithm can only be used to sample from (12) if 
\[
1 - \cos \theta(\eta) \in [0, 1], \text{ for all cycles } \eta \subset E,
\]
in which case we refer to the cycles as “weakly inconsistent”. It is yet unclear how to modify this algorithm if there is a cycle \( \eta \) such that \( 1 - \cos \theta(\eta) > 1 \). To overcome this limitation, we shall use self-normalized importance sampling with a modification of (12) in Section 7. For the rest of this section, we assume that all cycles in the graph are weakly inconsistent.

Oriented MTSFs. The algorithm to be described is based on a random walk which naturally generates oriented subgraphs of the original graph. These oriented subgraphs will be denoted with a superscript \( o \). Examples are provided in Figure 2.

An oriented rooted tree is a tree whose edges are oriented towards its root. An oriented cycle-rooted tree is a cycle-rooted tree constituted of an oriented cycle whose edges point in the same direction, whereas all other edges point towards the oriented cycle. Finally, an oriented MTSF contains only oriented rooted trees and oriented cycle-rooted trees. We consider the problem of sampling an oriented MTSF with probability
\[
Pr(F^o) \propto q^{\mid\mid F^o\mid\mid} \prod_{cycles \ \eta^o \in F^o} \alpha(\eta^o),
\]
where \( q > 0 \), and where \( \alpha \) takes values in \([0, 1]\) and is invariant under orientation flip. In particular, taking
\[
\alpha(\eta^o) = \frac{1}{2} \left( 2 - 2 \cos \theta(\eta^o) \right),
\]
sampling an oriented MTSF with probability (20), and forgetting the edge orientation, we obtain an MTSF distributed according to (12). In particular, the factor \( 1/2 \) in (21) compensates for the fact that there are exactly two oriented cycles for each (unoriented) cycle in an MTSF.

An auxiliary oriented MTSF with a root. Following the original approach of Wilson [1996] for spanning trees, we now associate to each oriented MTSF an oriented MTSF with a root. Specifically, we first augment \( G \) with an extra node \( r \) (\( r \) stands for “root”, as shall become clear), to which every node is connected with an edge with weight \( q \), resulting in the graph \( G_r \). This can be seen as adding a Dirichlet boundary condition at node \( r \) [Poncelet, 2018, Section 2.2]. Note that, if \( q = 0 \), node \( r \) will play no role.

An oriented MTSF with root \( r \) is defined as the disjoint union of a single oriented tree rooted at \( r \) and oriented cycle rooted trees. The algorithm to be described below samples an oriented MTSF with root \( r \).

Random successor. Denote by \( d(v) \) the number of neighbors of \( v \) in \( G \). The sampling algorithm relies on a random walk on the nodes of \( G_r \). Drawing from the Markov kernel, a procedure denoted by RANDOMSUCCESSOR(\( v \)), goes as follows: at node \( v \), move to the root \( r \) with probability \( q/(q + d(v)) \) or move to another neighbour of \( v \) with probability \( 1/(q + d(v)) \).
The cycle-popping random walk. Let \( \mathcal{F}_o \) be an oriented subgraph and \( w \notin \mathcal{F}_o \) be a node of \( \mathcal{G}_r \). We first define the procedure \( P(w, \mathcal{F}_o) \), closely following Kassel and Kenyon [2017, Section 2]. Starting from \( w \), perform a random walk thanks to RANDOMSUCCESSOR, until it reaches its first self-intersection \( v \), hits \( \mathcal{F}_o \), or reaches the absorbing node \( r \). This trajectory is an oriented branch. Then,

- if \( v \in \mathcal{F}_o \) or \( v = r \) (the root), add this oriented branch to \( \mathcal{F}_o \).
- if \( v \) is a self-intersection, then the branch contains an oriented cycle \( \eta^o \). Draw a Bernoulli random variable, independent from the rest, with success probability \( \alpha(\eta^o) \in [0, 1] \).
  - If the coin toss is successful, add the branch with the oriented cycle to \( \mathcal{F}_o \) and stop.
  - else pop the cycle (i.e., remove its edges from the branch) and continue RANDOMSUCCESSOR until self-intersection, intersection with \( \mathcal{F}_o \), or until it reaches the absorbing node \( r \) again.

The entire algorithm, denoted by CYCLEPOPPING, goes as follows: initialize \( \mathcal{F}_o \) as empty and take any \( w \in \mathcal{G} \) as starting node to execute \( P(w, \mathcal{F}_o) \). If \( \mathcal{F}_o \) does not span all the nodes of \( \mathcal{G}_r \), repeat \( P(w, \mathcal{F}_o) \) starting from any \( w \notin \mathcal{F}_o \). The procedure stops when \( \mathcal{F}_o \) spans all the nodes.

Post-processing. The cycle-popping procedure yields an oriented MTSF with root \( r \). As a simple post-processing step, we obtain an oriented MTSF of \( \mathcal{G} \) by overriding the extra node \( r \) and the links that point towards it. Each of the nodes formerly connected to \( r \) are then the roots of the remaining oriented MTSF of \( \mathcal{G} \). In particular, the number of links pointing to \( r \) in the underlying oriented MTSF with a root becomes the number of roots of the associated oriented MTSF.

Proposition 4 (Correctness). Under Assumption 1, the algorithm terminates and its output is distributed according to (20).

The proof of Proposition 4 is given in Appendix A.4 and adapted from Kassel and Kenyon [2017]. Note that, if the cycle weights are given by (21), the output of the algorithm is an oriented MTSF without 2-cycles since the latter are popped with probability 1, and therefore, by forgetting the edge orientations we obtain an MTSF with root \( r \). If \( q = 0 \), the output is simply a CRSF.

Remark 3 (Running time for sampling an MTSF vs. CRSF). We give here a qualitative discussion of upper bounds on average running times. First, let \( q = 0 \) and consider a non-trivial connection. If there is only one cycle \( \eta \) with a non-zero weight \( \alpha \in (0, 1] \), the average number of times this cycle is popped before acceptance \( 1/\alpha \) [Kassel and Kenyon, 2017, p. 839]. Indeed, \( 1/\alpha \) is the expectation of a geometric random variable of parameter \( \alpha \). Next, the time to get the trees rooted at this CRT is the running time of Wilson’s algorithm. If there are several cycles with non-zero weights, the average time to get the first cycle is bounded by \( O(1/\alpha_{\text{min}}) \) where \( \alpha_{\text{min}} \) is the smallest non-zero cycle weight.

If \( q > 0 \), the probability that a node \( u \) becomes a root is \( q/(q + d(u)) \) where \( d(u) \) is the degree of \( u \). This probability is lower bounded by \( q/(q + \max_u d(u)) \). Hence, the average time to get the first root is \( O(1 + \max_u d(u)/q) \). Thus, if \( q \) is large enough, the time for sampling an MTSF might be bounded by a smaller number compared with the CRSF counterpart.

7 Beyond weakly inconsistent graphs with importance sampling

If there are strongly inconsistent cycles in the connection graph, i.e., if (19) does not hold, then the cycle-popping algorithm in Section 6 cannot be used to sample from \( p(\mathcal{F}) \propto q^{\ell(\mathcal{F})} \prod_{\eta \in \mathcal{F}} 2(1 - \cos \theta(\eta)) \), since some cycle weights are larger than 1. However, we can cap the cycle weights at 1 and use the cycle-popping algorithm of Section 6 to sample from the auxiliary distribution

\[
p_{\text{IS}}(\mathcal{F}) \propto q^{\ell(\mathcal{F})} \prod_{\eta \in \mathcal{F}} 2 \{1 \wedge (1 - \cos \theta(\eta))\}.
\]

In particular, the cycle-popping algorithm will keep any encountered oriented cycle which is not weakly consistent with probability 1. Note that (22) is not necessarily determinantal. We can sample from (22) with the help of CYCLEPOPPING, indeed the proof of correctness remains valid if some cycles have weight 1.
Now, to correct for this modification, define the importance weight
\[ w(F) = a \frac{P(F)}{P_{\text{IS}}(F)} = a \prod_{\text{cycles } \eta \in F} \left\{ 1 \lor \left( 1 - \cos \theta(\eta) \right) \right\}, \tag{23} \]
where the normalization constant is there for \( w \) to sum to 1. Note that the above construction relies on the simple identity \( \{1 \land x\} \{1 \lor x\} = x \) for all \( x \geq 0 \), and that we shall not need to know the normalization constant for the weights.

The effect of capping the cycle weights to unity will be accounted for by considering the reweighted sparsifier
\[ \tilde{\Delta}_{t}^{\text{IS}} = \frac{1}{\sum_{s=1}^{t} w(F_s^t)} \sum_{\ell=1}^{t} w(F_{\ell}^t) \Delta(F_{\ell}^t), \quad \text{with } F_{\ell}^t \sim p_{\text{IS}} \text{ for } 1 \leq \ell \leq t. \tag{24} \]

**Proposition 5.** Let \( p \in (0, 1) \) and let Assumption 1 hold. Let \( F_1^t, F_2^t, \ldots \) be i.i.d. random MTSFs with the capped distribution (22), and consider the sequence of matrices \( (\tilde{\Delta}_{t}^{\text{IS}})_{t \geq 1} \) defined by (24). Finally, let \( z > 0 \) be such that \( \Pr(\|u\|_2 \leq z) = p \) for \( u \sim \mathcal{N}(0, I_n) \). Then, as \( t \to \infty \),
\[ \Pr \left[ -z(\Delta + q I_n) \preceq \tilde{\Delta}_{t}^{\text{IS}} - \Delta \preceq z(\Delta + q I_n) \right] \to 1 - p. \]

The proof of Proposition 5, given in Appendix A.5, uses standard techniques such as Slutsky theorem. Proposition 5 provides us with an asymptotic guarantee for the reweighted importance sampling procedure (24).

## 8 Empirical results

After describing in Section 8.1 a few settings which are shared in most of our simulations, we provide a few illustrations of the applicability of the approximations given above. In Section 8.2, we consider the unsupervised angular synchronization problem and its application to ranking from pairwise comparisons. Next, Laplacian preconditioning is discussed in Section 8.3. The code to reproduce the experiments is freely available on GitHub.\(^3\)

### 8.1 Settings and baselines

In our numerical simulations, we generate random connection graphs with a controlled amount of noise, so that we can vary the level of consistency of the cycles.

#### 8.1.1 Random graphs

An Erdős-Renyi graph \( \text{ER}(n, p) \) is a random graph with \( n \) nodes, defined by independently adding an edge between each unordered pair of nodes with probability \( p \).

The stochastic block model \( \text{SBM}(n_1, n_2, C) \) is a random graph with two communities of size \( n_1 \) and \( n_2 \), in which the edges are drawn as independent Bernoullis such that \( C_{ij} \geq 0 \) is the mean number of neighbors of a node in block \( i \) belonging to block \( j \), with \( 1 \leq i, j \leq 2 \).

**Noisy comparison models and random connection graphs.** Our \( \text{U}(1) \)-connection graphs are generalizations of Erdős-Renyi graphs which are endowed with a connection. Two statistical models for noisy pairwise comparisons are used: the MUN and ERO models. These models are defined given a fixed ranking score vector \( h \) such that \( h_u \) is the ranking score of \( u \in \mathcal{V} \). Here we choose \( h \) as a (uniform) random permutation of \( [1, \ldots, n] \).

Following Cucuringu [2016], we define the Multiplicative Uniform Noise model, denoted by MUN\((n, p, \eta)\), as the following comparison graph of \( n \) nodes. With probability \( p \), and independently from other edges,
there is an edge \( e = uv \) with \( 1 \leq u < v \leq n \) coming with an angle \( \vartheta(uv) = (h_u - h_v)(1 + \eta \epsilon_{uv})/(\pi(n-1)) \) where \( \epsilon_{uv} \sim \mathcal{U}(0, 1) \) are independent noise variables. Then, \( \vartheta(vu) \triangleq -\vartheta(uv) \).

Another noise model used here and defined by Cucuringu [2016] is the Erdős-Rényi Outliers (ERO) model, denoted by ERO\((n, p, \eta)\) which is given as follows. For all pairs \( uv \) with \( 1 \leq u < v \leq n \), the corresponding edge \( e = uv \) is added with probability \( p \), independently again from other edges. With probability \( 1 - \eta \), this new edge comes with an angle \( \vartheta(uv) = (h_u - h_v)/(\pi(n-1)) \), otherwise the angle is set to \( \vartheta(uv) = \epsilon_{uv}/(\pi(n-1)) \), where \( \epsilon_{uv} \) is drawn from the discrete uniform distribution on \( \{-n+1, \ldots, n-1\} \). Again, \( \vartheta(vu) \triangleq -\vartheta(uv) \).

### 8.1.2 Methods and Baselines

In our simulations, we compute sparsifiers of the form \( \frac{1}{t} \sum_{\ell=1}^{t} \tilde{\Delta}(F_{\ell}) = B^{*}S^{(b)}S^{(b)^{T}}B \) for \( t \geq 1 \) batches of samples; with the exception of CRSFs and MTSFs, for which we use the self-normalized formula in (24).

#### Leverage score heuristics.

The sparsification guarantees presented in Section 5 implicitly assume that the sampling matrix (18) can be calculated exactly. However, evaluating or even approximating leverage scores is far from trivial. This is a known issue even for i.i.d. sampling and in the case of the combinatorial Laplacian; see Spielman and Srivastava [2011] or Durfee et al. [2017] for a recent approach. Hence, we consider two options for the leverage scores in (18).

1. First, we use the exact Leverage Scores (LSs) \( \bar{l}(e) = l(e) = |B(B^{*}B + q_{n}^{\perp})^{-1}B^{*}|_{ee} \) in order to verify the quality of sparsifiers, although this approach is prohibitive in practice. This approach for sparsification is referred to as “DPP(K) LS”.

2. Second, we use the uniform Leverage Scores heuristics, i.e., for each batch \( \{F_{\ell}\}_{1 \leq \ell \leq t} \) we weigh each edge \( e \in F_{\ell} \) in (18) by \( \bar{l}(e) = l_{\text{unif}}(e) \) with \( l_{\text{unif}}(e) = |F_{\ell}|/m \).

This approach for sparsification with (25) is called “DPP(K) unif". The reason for this normalization goes as follows: the exact leverage scores are normalized so that their sum is the expected sample size \( \sum_{e=1}^{m} l(e) = \mathbb{E}_{F \sim \text{DPP(K)}}[|F|] \); see Section 4. Similarly, since \( F_{\ell} \sim \text{DPP}(K) \), in this paper, we choose the normalization so that \( \sum_{e=1}^{m} l_{\text{unif}}(e) = |F_{\ell}| \).

The latter heuristics is expected to be efficient whenever the connection graph is well connected. It is closely related to the heuristics used for preconditioning linear systems in the context of kernel ridge regression by Rudi et al. [2017].

#### Baselines with i.i.d. sampling and with spanning trees.

As a first type of baseline, we use sparsifiers obtained with two i.i.d. sampling methods:

1. with edges sampled from the discrete uniform distribution and the leverage scores are approximated by the uniform LS heuristics – this approach is called “i.i.d. unif.”,

2. with edges sampled w.r.t. the leverage score distribution [Spielman and Srivastava, 2011] and the leverage scores are computed exactly – this approach is referred to as “i.i.d. LS”.

For a fair comparison of these i.i.d. baselines with a batch of MTSFs, we sample \( t \) independent \( F_{\ell} \), \( 1 \leq \ell \leq t \), where each \( F_{\ell} \) is an i.i.d. sample of edges. Next, for a given \( q \geq 0 \), we compute the average sparsifiers \( \frac{1}{t} \sum_{\ell=1}^{t} \Delta_{F_{\ell}} + q_{n}^{\perp} \). In all the simulations below, we fix the cardinality of an i.i.d. batch to \( |F_{\ell}| = n \), i.e., the number of edges in a CRSF. In the case (i), \( \Delta_{F_{\ell}} \) is computed thanks to (18) by using the uniform leverage score heuristics (25). In the case (ii), the exact magnetic leverage scores are used.

As a second type of baseline, we compute two other types of sparsifiers obtained by sampling batches of uniform spanning trees (STs) with Wilson’s algorithm. The corresponding edges are then taken with their corresponding complex phases although the latter are not used for sampling the spanning trees. The sparsifiers are then built by using the formula (18) by using spanning trees. This yields two other baselines:

3. “ST unif.”: with the uniform leverage score heuristics (25) and,

4. “ST LS”: with \( \bar{l}(e) = [B_{0}(B_{0}^{*}B_{0})^{+}B_{0}]_{ee} \) where \( B_{0} \) is the (real) oriented edge-vertex incidence matrix defined in Section 3. These scores are the exact (non-magnetic) leverage scores.
Sampling strategy and self-normalized importance sampling. To sample CRSFs and SFs, we use the algorithm described in Section 6, whereas we use Wilson’s algorithm [Wilson, 1996] to sample STs. As a common feature, the starting node of the cycle popping random walk is chosen uniformly at random. For sampling STs, we first sample an oriented rooted spanning tree with a root node sampled uniformly at random, and then, we forget the root and the orientation of this spanning tree.

For sampling MTSFs or CRSFs in the graphs considered here, since we do not know if the cycles are weakly inconsistent, we use the importance sampling distribution (22) with capped cycle weights, and use the self-normalized formula in (24). By an abuse of notation, these methods are denoted as “DPP(K)”.

Sparsification with batches of edges. In the simulations where we compare the accuracy of sparsifiers for a different batchsize, each sparsifier is drawn independently from all the others. In what follows, we often plot the performance of a sparsification method for a given batchsize and the x-axis then reports the percentage of edges corresponding to the batch. Note that since the number of edges in an MTSF is random, horizontal error bars appear when we average over independent realizations. In our experiments, these error bars are however very small.

8.2 Magnetic Laplacian sparsification and ranking

One of our motivations was angular synchronization; see Section 3. One application of angular synchronization is ranking from pairwise comparisons [Cucuringu, 2016, Sync-Rank], which we use here as a case study to illustrate the interest of sparsifying the magnetic Laplacian. More specifically, we consider Sync-Rank, an algorithm for ranking items from pairwise comparisons, which has favorable robustness properties w.r.t. corruptions in the comparisons [Cucuringu, 2016; Yu, 2012]. The optimization objective of Sync-Rank is directly inspired by the angular synchronization problem as we explain below.

8.2.1 Angular synchronization and spectral relaxation.

The Sync-Rank algorithm of Cucuringu [2016] starts from comparisons between neighbouring nodes, where uv ∈ E can carries either a cardinal comparison κuv ∈ [−n−1, n+1] or an ordinal comparison κuv ∈ {−1, +1}.

A positive κuv is interpreted as u being superior to v.

1. First, we define the following angular embedding of the comparisons ϑ(uv) = πκuv/(n − 1), for all oriented edges uv in the graph.

2. Spectral Sync-Rank simply solves the above spectral problem (7), with the connection graph obtained at step 1. To account for a non-uniform degree distribution, Cucuringu [2016, Eq (12) and (13)] recommends weights wuv = 1/√d(u)d(v), where d(u) denotes the degree of u in the (unweighted) connection graph. This type of normalization also typically improves graph clustering performance, and we use this weighting here in (7). The output of this stage is an angular score ˆhu ∈ [0, 2π) for all u ∈ V, as defined above, by taking ˆhu = arg(f(u)).

3. We now select a ranking from the angular scores. We find a permutation ζ such that ˆhζ(1) ≥ · · · ≥ ˆhζ(n). Let ζ = [ζ(1), . . . , ζ(n)]T. Let r = [r1, . . . , rn]T be the inverse permutation of ζ, so that ru ∈ {1, . . . , n} is an integer giving the induced ranking of the node u ∈ V. The ranking of the n items is then obtained by looking for the circular permutation of r1, . . . , rn minimizing the number of upsets,

σ∗ = arg minσcircular ∑ oriented edge uv |sign(κuv) − sign(rσ(u) − rσ(v))|;

see Cucuringu [2016, Alg. 1]. Explicitly, a circular permutation on integers in [n] depends on an integer s and is defined as σ(ℓ) = 1 + (ℓ + s) mod n, with s ∈ {0, . . . , n − 1}. The output is σ∗.

Note that we have no statistical guarantee on recovering the ranking.
8.2.2 Empirical results for the sparsification of Sync-Rank.

Figure 3 displays the performance of the sparsify-and-eigensolve algorithms described in Section 8.1.2 on the two random graphs described in Section 8.1.1. We plot three performance metrics, as a function of the total number of edges in each batch. In other words, each point on the x-axis corresponds to a sparsifier obtained with a batchsize $t$ ranging from 1 up to 7. The $x$-axis reports the number of edges in $\cup_{t=1}^{\ell} F_t$. The first row of Figure 3 reports the least eigenvector approximation accuracy, i.e., the distance $1 - |\tilde{f}_1^t f_1|$ between the line of $\tilde{f}_1$ and the line of $f_1$. Note that the eigenvectors are normalized and computed exactly. The second row reports Kendall’s tau coefficient between the recovered and exact synthetic ranking, a classical measure of accuracy in ranking recovery. To assess the number of cycles captured by CRSF sampling, the third row displays the average number of sampled cycle rooted trees. In the left column, we take an ERO($n, p, \eta$) random graph and the right column corresponds to a MUN($n, p, \eta$) graph. The parameters are $n = 100$, $p = 0.9$, $\eta = 0.2$. The sampling is repeated 10 times for a fixed connection graph. We display the mean, whereas the error bars are plus/minus one standard deviation.

As expected, all algorithms reach the same performance with enough edges, whatever the metric. Another immediate observation is that, for the random graphs we consider, the degree distribution is rather concentrated. The uniform leverage score heuristic yields good results for the CRSF-based sparsifiers, in terms of the approximation of the least eigenvector of the magnetic Laplacian, as shown in Figure 3.

We also observe that for the MUN and ERO random connection graphs considered, CRSF and ST sampling have a comparable performance, whereas i.i.d. sampling gives a poorer result. For the chosen parameters, the magnetic Laplacian approximation requires fewer edges for MUN (Figure 3c) than for ERO (Figure 3d). For the MUN graph, the number of sampled CRTs is close to minimal, namely there is about one CRT per CRSF; see Figure 3e. Also, the amount of CRTs sampled by our algorithm is larger in the case of the ERO model (Figure 3f), which confirms that the latter corresponds to a more difficult problem. In Appendix C.2, the average inconsistency of the sampled cycles are also displayed to illustrate this observation.

In conclusion, for these well-connected graphs, the sparsifiers obtained with batches of CRSFs or batches of STs both reach approximately the same performance for sparsifying Sync-Rank, in terms of ranking recovery.

8.3 Laplacian preconditioning

As motivated in Section 3.2, we give a few examples of condition number reduction by using sparsifiers. First, we illustrate the preconditioning of the magnetic Laplacian $\Delta$, and second, we analyse the preconditioning of the regularized combinatorial Laplacian $\Lambda + q\hat{\Pi}_n$.

8.3.1 Magnetic Laplacian: sparsify-and-precondition

In our simulations, we consider connection graph models with a planted ranking and a low level of noise since $\text{cond}(\Delta)$ is larger in this case. In Figure 4, on the y-axis, we display $\text{cond}(\Delta^{-1}\Delta)$. The x-axis indicates the total number of edges contained in the union of batches. Note that $\lambda_1(\Delta) \approx 0.0025$ for the MUN graph and $\lambda_1(\Delta) \approx 0.06$ for the ERO graph. The sampling is repeated 10 times for a fixed connection graph and the mean is displayed. The error bars are large for i.i.d. sampling and are not displayed to ease the reading of the figures. We compare the condition number $\text{cond}(\Delta^{-1}\Delta)$ for sparsifiers $\tilde{\Delta}$ obtained from different sampling methods. This problem is motivated by applications to angular synchronization described in Section 3.1 and to semi-supervised learning described in Section 3.2. An advantage of using CRSFs to construct the sparsifier $\tilde{\Delta}$ is that the latter is invertible almost surely even in the case of one MTSF. This is not necessarily true if i.i.d. samples or STs are used in which case we replace the sparsifier by $\tilde{\Delta} + 10^{-12}\mathbf{I}_n$. Both in the case of a MUN($n, p, \eta$) and ERO($n, p, \eta$) connection graph, we observe the sparsifiers obtained with CRSF sampling are more efficient that other sampling techniques.

In the case of the MUN model in Figure 4a, one CRSF is already sufficient to improve the condition number of $\Delta$, whereas at least two batches of STs are necessary to reach the same accuracy. For this type of noise model where the noise is uniform over the edges, batches of spanning trees provide a good approximation.

In the case of the ERO model, the noise is only corrupting certain edges whereas other edges are noiseless. In Figure 4b for a small $\eta$, sparsifiers built with CRSFs yield a better approximation compared with STs.
or other approaches. Intuitively, we can understand this behaviour as follows: this random graph contains only a few inconsistent edges which are likely to be sampled in CRSFs. Recall that an ERO\((n, p, \eta)\) random graph is a connection graph where an edge is present with probability \(p\) and is corrupted by a uniform noise with probability \(\eta\). To confirm our intuition for this ERO model, we computed the average number noisy edges captured by generating random subgraph. We found 1.2(0.42) noisy edges on average in a random CRSF and 0.38(0.6) noisy edges in a random ST; note that the standard deviation over 100 runs is given in parenthesis. Also, notice that “i.i.d. LS” tends to sample edges coming with noisy angles; see Appendix C.3 for a numerical illustration.

**Remark 4 (Bound on the condition number).** Note that the largest eigenvalue of \(\Delta\) is always bounded from above by \(2 \max_v d(v)\), as it can be shown thanks to Gershgorin circle theorem. Thus, the condition number of \(\Delta + qI_n\) is bounded from above by \(2 \max_v d(v)/(q + \lambda_1(\Delta))\). A consequence is that ill-conditioning is more
likely to arise for large U(1)-connection graphs with a low level of inconsistencies.

### 8.3.2 Regularized Combinatorial Laplacian

We consider here the sparsified combinatorial Laplacian defined by $\tilde{\Lambda} = B_0^* S S^T B_0$ where $B_0$ is the (real) oriented incidence matrix – see Section 3 – and the sampling matrix $S_{ee'}(F) = \delta_{ee'} f(e')/\sqrt{l_0(e)}$, where $l_0(e) = [B_0(B_0^* B_0 + B_0^*)_{ee}]$ for all $e \in E$. In what follows, we simply display in the figures the condition number of $(\tilde{\Lambda} + q I_n)^{-1}(\Lambda + q I_n)$ for sparsifiers $\tilde{\Lambda}$ obtained thanks to several sampling strategies.

**Erdős-Renyi graph.** Our first simulation concerns an Erdős-Renyi graph $ER(n, p)$ with $n = 100$ and $p = 0.9$. In Figure 5, on the left-hand side, we display on the $y$-axis $\text{cond}((\tilde{\Lambda} + q I_n)^{-1}(\Lambda + q I_n))$ for $q = 10^{-3}$ for an $\tilde{\Lambda}$ built with batchsizes ranging from 1 to 6, whereas the $x$-axis indicates the total number of edges. Note that the regularizer is $q = 10^{-3}$ and the second least eigenvalue is $\lambda_2(\Lambda) \approx 81$. On the right-hand side, we report the cumulated number of roots sampled for each set of batches. The sampling is repeated 3 times for a fixed Erdős-Rényi graph and the mean is displayed.

We observe that the MTSF sampling strategy outperforms i.i.d. sampling already with a batchsize equal to 1 in the case of this graph. For this kind of random graphs with a rather concentrated degree distribution, we observe that the knowledge of the LSs does not improve much the performance over the uniform leverage score distribution. As it is expected, the SF and i.i.d. approaches tend to reach in Figure 5a the same accuracy as the number of sampled edges increases. In this case, the SF performance is similar to the ST performance since for the chosen parameters each sampled SF contains exactly one root; see Figure 5b.

**Stochastic Block Model.** Contrary to the previous example, we consider here an SBM graph which has a structure with two clear communities, with one community denser than the other, connected by only a few edges. We draw a random SBM($n_1, n_2, C$) graph with $n_1 = 400$, $n_2 = 400$, $C_{11} = 100$, $C_{22} = 20$ and $C_{12} = 10^{-2}$; which is conditioned on being connected. In Figure 6, on the left-hand side, we display on the $y$-axis $\text{cond}((\tilde{\Lambda} + q I_n)^{-1}(\Lambda + q I_n))$ for $q = 10^{-2}$ and for a batchsize ranging from 1 to 6, whereas the $x$-axis indicates the total number of edges in these sets of batches. Note that $\lambda_2(\Lambda) \approx 0.019$. On the right-hand side, we report the cumulated number of roots sampled for each set of batches. The sampling is repeated 3 times for a fixed SBM graph and the mean is displayed.

The condition number is then clearly improved by using the sparsifiers which rely on the exact LSs, although the uniform heuristics also yields good preconditioners; see Figure 6a. In this case, SFs are not trees but are rather made of more than one tree on average as it can be seen in Figure 6b. The performance of STs and DPPs are approximately the same. The only observable improvement comes from the knowledge of the LSs, which can be expected to be important since this SBM has a clear community structure.
Real graph. Now, we perform the same simulations for a real network, the Polblogs data set, which represents the connectivity of blogs about US politics [Adamic and Glance, 2005] and is considered here as an undirected graph. This graph has \( n = 1490 \) nodes and \( m = 16718 \) edges. Only its largest connected component with \( n = 1222 \) nodes and \( m = 16717 \) is considered here. It is known to have two densely connected clusters or communities.

In Figure 7, on the left-hand side, we display on the \( y \)-axis \( \text{cond}((\tilde{\Lambda} + qI_n)^{-1}(\Lambda + qI_n)) \) for \( q = 0.1 \) with up to 6 batches of edges, whereas the \( x \)-axis indicates the total number of edges. Note that \( \lambda_2(\Lambda) \approx 0.17 \). On the right-hand side, we report the cumulated number of roots sampled for each set of batches. We display the average and standard deviations over 3 runs.

We observe a larger difference between the uniform LS heuristics and the exact MTSF sparsifier, due to the more complex LS distribution of this real network.

### 8.4 Empirical sampling time for the cycle popping algorithm

In this section, we compare the sampling time of MTSFs with respect to STs thanks to CyclePopping. This relative comparison is interesting since we use a similar algorithm for sampling MTSFs and STs but the overall time scale might not be reliable since our implementation is not optimized. A difference in sampling times can be visible for well-chosen graphs of rather small sizes.

We consider a graph – which is a variant of barbell graph that we denote by Barbell\((n)\) – made of two cliques of \( n/2 \) nodes (for even \( n \)), namely two fully connected components, which are connected by only one edge. This type of graphs with two cliques linked by a line graph is a worst case example for Wilson’s
cycle popping algorithm in terms of expected runtime\(^4\); see Guo and He [2020] for the partial rejection sampling viewpoint. This graph is considered for estimating the average time for sampling a random SF versus sampling a ST. In Figure 8, we report the average sampling time over 1000 runs and the error bars are the standard errors of the mean. Our empirical observation is that sampling an SF might be faster than sampling an ST if the graph has dense communities linked with bottlenecks. This is particularly visible if \(q\) has a large value. This phenomenon is observed in the extreme case of the barbell graph in Figure 8.

To perform a similar comparison of sampling times for CRSFs and SFs, the same barbell graph is endowed with a connection as follows. Let \(h\) be the planted ranking score such as defined in Section 8.1. For each edge \(uv\) in the barbell graph, with probability \(1 - \eta\), these edge comes with an angle \(\vartheta(uv) = (h_u - h_v)/(\pi(n-1))\) and \(\vartheta(uv) = \epsilon_{uv}/(\pi(n-1))\) with probability \(\eta\) where \(\epsilon_{uv}\) drawn from the discrete uniform distribution on \(-n+1, \ldots, n-1\). Once more, we take \(\vartheta(vu) = -\vartheta(uv)\). The resulting connection graph is denoted by \(\text{Barbell}(n, \eta)\).

In Figure 9, on the left-hand side, we display the average sampling time over 5000 runs for CRSFs and STs as a function of the noise parameter \(\eta\). On the right-hand side, we report the average number of components (CRTs) in the sampled CRSFs as a function of \(\eta\). The error bars are the standard errors of the mean.

Empirically, we observe that the average sampling time for a CRSF indeed decreases as \(\eta\) increases whereas the number of cycle-rooted trees also increases. Informally, for small values of \(\eta\), the average time for sampling a CRSF in this case might be large since \(\text{CyclePopping}\) will pop many cycles before finally accepting one. This behavior is rather similar to the case of SFs in Figure 8.

\[^4\]The mean hitting time for sampling a rooted spanning tree in a barbell graph with two cliques of size \(n/3\) linked by a line graph with \(n/3\) nodes is \(\Omega(n^3)\) when the initial node and the root are sampled according to the stationary distribution of the walk; see Aldous and Fill [2002, Example 5.11] for more details.

---

**Figure 7:** Regularized combinatorial Laplacian preconditioning for the Polblogs graph.

**Figure 8:** Comparison of sampling times for SFs and STs for a Barbell\((n)\) graph with \(n = 500\).
Figure 9: Comparison of sampling times of CRSFs and STs for a Barbell($n, \eta$) with $n = 500$.

9 Discussion

Similarly to how ensembles of uniform spanning trees sparsify the combinatorial Laplacian, we have investigated ensembles of multi-type spanning forests to sparsify the magnetic Laplacian. The distribution we use to draw one such MTSF is a determinantal point process (DPP), like uniform spanning trees. Our theoretical results include a Chernoff bound with intrinsic dimension for DPPs, which might be of independent interest, and a confidence “interval” based on self-normalized importance sampling to treat graphs with large inconsistencies.

Our experiments suggest that sparsifiers based on CRSFs and MTSFs are the solution of choice when only a few inconsistent cycles are present, so that it is important not to miss them. In this circumstance, the least eigenvalue of the magnetic Laplacian is expected to be small yielding a large condition number for the system associated with $\Delta$. In other cases, and in particular when the graph is well-connected, a simpler approach using spanning forests with comparable leverage-score heuristics actually performs on par with our MTSFs. This good performance of STs may appear surprising at first, given that the distribution used for STs does not rely on the graph’s connection. However, in our experiments, the number of cycles in CRSFs remained relatively low, and it would be interesting to further investigate graph structures that favor a large number of cycles in a CRSF. Similarly, the knowledge of the magnetic leverage scores seems to greatly improve i.i.d. sampling as shown in Figure 4b and Figure 13. An interesting future work would be to study the ability of the magnetic leverage scores to detect inconsistencies in a connection graph.

From the perspective of the sampling algorithm, the sampling time for the cycle-popping algorithm of Section 6 is also expected to be large when the least eigenvalue of $\Delta$ is small, i.e., for a low level of inconsistency. Indeed, if the cycles have low inconsistencies, the random walk is likely to spend a long time popping cycles before one cycle is successfully accepted. In this case, the aforementioned “generic” DPP sampling procedure by Hough et al. [2006] might also be costly or prone to numerical errors since it requires the full eigenvalue decomposition of the correlation kernel $K = B\Delta^{-1}B^*$; the latter eigenvalue problem being also possibly ill-conditioned due to the factor $\Delta^{-1}$.

Let us list below a few remarks about technological aspects of MTSF sampling.

- A computational advantage of CYCLEPOPPING over the algebraic algorithm of Hough et al. [2006] is that it is decentralized in the sense that the random walker only needs to query the knowledge of the neighbouring nodes and edges. Hence, the knowledge of the full connection graph is not necessarily needed for running CYCLEPOPPING and the random walker can discover the graph on the fly. This may reduce memory requirements.

- The output of CYCLEPOPPING is also necessarily an MTSF by design. This makes CYCLEPOPPING less sensitive to numerical errors compared with the generic algorithm of Hough et al. [2006] in this specific case, which relies on an iterative linear algebraic procedure. In other words, in case of an ill-conditioned $\Delta$, it is not excluded that an algebraic sampler outputs a graph which is not an MTSF due to numerical errors.
• CyclePopping relies on the condition that all cycles in the connection graph are weakly inconsistent, which seems a rather arbitrary limitation. Currently, we have no strategy for checking that this condition is satisfied. As a possible future work, extending CyclePopping beyond the case of weakly inconsistent connection graphs would be an interesting contribution to avoid using self-normalized importance sampling for which only weaker statistical guarantees for sparsification are given in Proposition 5.

• As illustrated in our simulations, leverage scores are important in order to sparsify Laplacians of graphs with a non-concentrated degree distribution, such as in the case of the Polblogs network of Figure 7. A prospect for future work consists in developing a leverage score approximation scheme for which some statistical guarantees can be derived.

Our implementation of CyclePopping is currently far from optimal. As a perspective, an optimized implementation of CyclePopping would have to be developed in order to compare standard linear algebraic solvers with our simulation results in terms of computational time.

Acknowledgements. We thank Guillaume Gautier for numerous discussions, and his precious support for Julia implementations. The paper further benefited from comments of Simon Barthelmé, and from discussions on partial rejection sampling. We also thank Nicolas Tremblay for discussions about spectral sparsification and suggestions concerning the numerical simulations. Furthermore, we thank Hugo Jaquard for pointing out typos.

We acknowledge support from ERC grant BLACKJACK (ERC-2019-STG-851866) and ANR AI chair BACCARAT (ANR-20-CHIA-0002).

A Deferred proofs about the magnetic Laplacian approximation

A.1 Proof of Proposition 2

Under Assumption 1, $\Delta + qI_n$ has only strictly positive eigenvalues. Hence, we can introduce the $n \times m$ matrix $\Psi = (\Delta + qI_n)^{-1/2}B^*$, so that $K = \Psi^*\Psi$. Now, define

$$Y_e = \frac{1}{l(e)}\psi_e^*\psi_e$$

for all $e \in [m]$, where $\psi_e$ denotes the $e$-th column of $\Psi$ and $l(e) = \psi_e^*\psi_e$. By definition, $\|Y_e\|_{op} \leq 1$ and $E_F \left[ \sum_{e \in F} Y_e \right] = \sum_{e=1}^m \psi_e^*\psi_e = \Psi^\ast$. By construction, $\kappa = \lambda_{\text{max}}(\Psi^\ast) \leq 1$. With the notations of Theorem 2, for all $\varepsilon \in (0,1]$, it holds

$$\Pr \left[ \left\| \sum_{e \in F} Y_e - E_F \left[ \sum_{e \in F} Y_e \right] \right\|_{op} \geq \varepsilon \kappa \right] \leq 2\text{intdim}(M) \left( 1 + \frac{c_1 \varepsilon^2}{\kappa^2 \varepsilon^4} \right) \exp \left( -\frac{c_2 \varepsilon^2 \kappa}{r} \right),$$

where we take $r = 1$ and with $c_1 = 3 \cdot 37^2$ and $c_2 = \frac{1}{37}$.

Now, fix $\varepsilon^4 \geq c_1 / \kappa^2$, so that

$$\Pr \left[ \left\| \sum_{e \in F} Y_e - E_F \left[ \sum_{e \in F} Y_e \right] \right\|_{op} \geq \varepsilon \kappa \right] \leq 4\text{intdim}(M) \exp \left( -c_2 \varepsilon^2 \kappa \right).$$

Let $\delta \in (0,1)$ such that $4\text{intdim}(M) \exp \left( -c_2 \varepsilon^2 \kappa \right) \leq \delta$, provided that $\varepsilon$ is large enough. We identify $M = K$ and $\text{intdim}(M) = d_{\text{eff}}/\kappa$. Now, we choose $\varepsilon^2 \geq \frac{1}{c_2 \kappa} \log \left( \frac{4d_{\text{eff}}}{\kappa \delta} \right)$ and define $\epsilon = \kappa \varepsilon$, whereas we emphasize that $\varepsilon$ and $\epsilon$ are distinct quantities. Therefore, we have

$$-\epsilon I \preceq \sum_{e \in F} Y_e - E_F \left[ \sum_{e \in F} Y_e \right] \preceq \epsilon I,$$
with probability at least $1 - \delta$. Therefore, we have $-\epsilon \leq \Psi S^T \Psi^* - \Psi \Psi^* \leq \epsilon$, by definition of $S = S(\mathcal{F})$ at the beginning of Section 5.1. Note that we take

$$
\epsilon^2 = \left( c_1^{1/2} \right) \vee \left( c_2 \log \left( \frac{4d_{\text{eff}}}{\kappa \delta} \right) \right).
$$

Finally, plugging back $\Psi = (\Delta + qI_n)^{-1/2} B^*$, we find that

$$
-\epsilon (\Delta + qI_n) \preceq \tilde{\Delta} (\mathcal{F}) - \Delta \leq \epsilon (\Delta + qI_n)
$$

with probability at least $1 - \delta$, which is the desired result.

### A.2 Proof of Proposition 3

The proof technique is directly adapted from Vishnoi [2013, Theorem 13.3]. It relies on the observation that the Cholesky decomposition of a matrix $A$ follows from the Schur complement formula

$$
\begin{pmatrix}
d & u^* \\
u & A
\end{pmatrix} =
\begin{pmatrix}
1 & 0^T \\
u/d & 0 & A - uu^*/d
\end{pmatrix} \begin{pmatrix}
1 & u^*/d \\
0 & 1
\end{pmatrix},
$$

(26)

which corresponds to the elimination of the first row and column [Vishnoi, 2013, Proof of Theorem 13.1]. In what follows, we shall associate to a sparse symmetric matrix $A$ a graph which has an edge for each non-zero off-diagonal element of $A$. This matrix is equal to a block of the regularized Laplacian at the first iteration of the decomposition algorithm and is updated at the subsequent iterations. A formula akin to (26) holds for the elimination of the $i$-th row and column. Recursively applying this formula, the Cholesky decomposition appears as a product of triangular matrices.

In our case, we first note that each connected component of the MTSF can be treated separately since, by an appropriate permutation, the regularized magnetic Laplacian of the MTSF $\Delta + qI$ can be reduced to have a diagonal block structure. We thus henceforth assume that $\mathcal{F}$ has a unique cycle, of length $c > 0$. The idea of the proof is to find a permutation matrix $Q$ such that the Cholesky decomposition of $Q(\Delta + qI)Q^T$ has at most $O(n + c)$ non-zero off-diagonal entries. This permutation corresponds to the desired reordering of the nodes.

We now construct the permutation matrix $Q$. Each connected component of a MTSF is either a tree or a cycle-rooted tree. Consider first the case of a rooted tree. We can proceed with the elimination by peeling off the leaves of the tree. Notice that after eliminating a leaf, the resulting structure of $A = uu^*/d$ is a tree with one missing leaf. This procedure specifies an ordering of the nodes, and therefore the permutation matrix $Q$. Since each leaf only has one neighbor, the vector $u$ only has one non-zero entry and therefore, the corresponding column of the triangular matrix

$$
\begin{pmatrix}
1 & 0^T \\
u/d & 0 & A - uu^*/d
\end{pmatrix}
$$

in (26) has only one non-zero off-diagonal entry. For the same reason, the subtraction $A - uu^*/d$ requires $O(1)$ operations. By computing the product of triangular matrices obtained by eliminating one leaf after the other, we see that each elimination corresponds to the update of one column in the final triangular matrix. Thus, the Cholesky decomposition of the Laplacian of a rooted tree of $n_t$ nodes has at most $n_t - 1$ non-zero off-diagonal entries, and requires $O(n_t)$ operations.

Second, consider the case of a cycle-rooted tree, with one cycle of $n_i$ nodes and $n_{\text{crt}}$ nodes in total. We proceed similarly by eliminating one leaf after the other until only the cycle remains. This first stage yields $n_{\text{crt}} - n_i$ non-zero off-diagonal entries. Next, we eliminate the nodes of the cycle. For each such node, the vector $u$ in (26) has two non-zero entries since each node in a cycle has two neighbors. Furthermore, if $u$ has two non-zero entries, the matrix $uu^*$ has 4 non-zero entries and the corresponding update costs $O(1)$ operations. Hence, if $n_i > 3$, eliminating a node in a cycle with $n_i$ nodes yields another cycle with $n_i - 1$ nodes. This continues until only two nodes remain. Eliminating one of them yields one non-zero entry in the triangular matrix. Thus, in total, eliminating a cycle with $n_i \geq 3$ nodes yields $2(n_i - 2) + 1$ non-zero entries. So, for a cycle-rooted tree of $n_{\text{crt}}$ nodes in total with a cycle of $n_i$ nodes, we have at most: $n_{\text{crt}} - n_i$ non-zero entries for the nodes in the trees and $2n_i - 3$ non-zero entries for the nodes in the cycle. Thus, in total, there are at most $n_{\text{crt}} + n_i - 3$ non-zero entries in the triangular matrix and this costs $O(n_{\text{crt}} + n_i)$ operations.

By combining the counts for the rooted trees and cycle-rooted trees, we obtain the desired result. Note that the diagonal of $\Delta + qI$ does not influence the sparsity of the decomposition.
A.3 Proof of Theorem 1

Let \( \mathcal{X} \) be the point process on \([tm]\) made of the concatenation of \( t \geq 1 \) independent copies of \( \text{DPP}(K) \). We first note that \( \mathcal{X} \) is again a DPP, and its correlation kernel is a block diagonal matrix with \( t \) copies of \( K \) along the diagonal. We now apply Theorem 2 to that DPP.

Define the \( n \times m \) matrix \( \Psi = (\Delta + qI_n)^{-1/2}B^* \), so that \( K = \Psi^*\Psi \). We also need \( t \) identical copies of the same set of matrices. Hence, we define \( Y_e^{\ell} = 1_{t \times l(e)}\psi_e\psi_e^* \), for all \( \ell \in \{0, t - 1\} \) and all \( e \in [m] \),

\[
Y_{e+tm} = \frac{1}{t \times l(e)}\psi_e\psi_e^*, \quad \text{for all } \ell \in \{0, t - 1\} \text{ and all } e \in [m], \tag{27}
\]

where \( \psi_e \) denotes the \( e \)-th column of \( \Psi \). By definition, \( \|Y_e\|_{op} \leq 1/t \) and \( E \mathcal{X} \sum_{e \in \mathcal{X}} Y_e = \sum_{r=1}^m \psi_r\psi_r^* = \Psi\Psi^* \). Let now \( M = E \mathcal{X} \sum_{e \in \mathcal{X}} Y_e = \Psi\Psi^* \) and \( \kappa = \kappa(M) = \kappa(K) \leq 1 \). By Theorem 2, for all \( \varepsilon \in (0, 1] \), it holds

\[
\Pr \left[ \left\| \sum_{e \in \mathcal{X}} Y_e - E \mathcal{X} \sum_{e \in \mathcal{X}} Y_e \right\|_{op} \geq \varepsilon \kappa \right] \leq 2 \text{intdim}(M) \left( 1 + c_1 \frac{r^2}{\kappa^2} \right) \exp \left( -c_2 \frac{\varepsilon^2 \kappa}{r} \right),
\]

for strictly positive constants \( c_1 \) and \( c_2 \) and where \( r = 1/t \). Here, \( \text{intdim}(M) = d_{eff}/\kappa \). If we take \( t \geq \sqrt{\frac{c_1}{\kappa^2}} \), then \( 1 + c_1 \frac{1/r^2}{\kappa^2} \leq 2 \). In particular, taking \( t \) large enough so that

\[
1 \geq \delta \geq 4 \frac{d_{eff}}{\kappa} \exp \left( -c_2 \frac{\varepsilon^2 \kappa}{r} \right),
\]

we obtain, with probability at least \( 1 - \delta \),

\[
-\varepsilon \kappa I \leq \sum_{e \in \mathcal{X}} Y_e - E \mathcal{X} \sum_{e \in \mathcal{X}} Y_e \leq \varepsilon \kappa I. \tag{28}
\]

The condition on the batchsize rewrites as follows

\[
t \geq \frac{1}{\kappa \varepsilon^2} \left\{ \left( c_2^{-1} \log \left( \frac{4d_{eff}}{\kappa \delta} \right) \right) \vee \sqrt{c_1} \right\}. \tag{29}
\]

As we did in Appendix A.1, we define \( \epsilon = \varepsilon \kappa \in (0, 1] \). Upon plugging back the definition (27) in (28), we obtain the desired \((1 \pm \epsilon)\)-multiplicative bound with probability larger than \( 1 - \delta \), provided that

\[
t \geq \frac{\kappa}{\epsilon^2} \left\{ \left( c_2^{-1} \log \left( \frac{4d_{eff}}{\kappa \delta} \right) \right) \vee \sqrt{c_1} \right\}. \tag{30}
\]

This completes the proof.

A.4 Proof of Proposition 4

The case \( q = 0 \) is treated in Kassel and Kenyon [2017, Proof of Thm 1]. The case \( q > 0 \) follows exactly the same lines, *mutatis mutandis*. We give here the sketched argument for completeness.

The routine RANDOMSUCCESSOR(v) draws from the Markov kernel \( P(v, \cdot) \) of a random walk, namely

\[
P(v, w) = \frac{1}{q + d(v)} \times \begin{cases} q & \text{if } w = r \\ 1 & \text{if } vw \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}, \tag{29}
\]

where \( d(v) \) is the degree of \( v \in \mathcal{V} \) in the original graph \( \mathcal{G} \); see Section 3. Note that the global factor \( \frac{1}{q + d(v)} \) in (29) is simply the inverse of the degree of \( v \) in the auxiliary graph \( \mathcal{G}_v \).
Sampling with stacks-of-cards. We use the stack-of-cards representation of the random walk. This algorithm is called CyclePopping′, as introduced by Wilson [1996], and we quickly outline it here. To begin, the absorbing node comes with an empty stack. To each other node \( v \neq r \) is attached an independent, infinite stack of cards \( X^{(v)} = (X^{(v)}_1, X^{(v)}_2, \ldots) \), drawn i.i.d. from \( P(v, \cdot) \). In addition, we associate to each oriented cycle \( \eta^o \) in the graph a stack of independent random variables, identically distributed as

\[
B_{\eta^o} \sim \text{Bernoulli}(\alpha(\eta^o));
\]  

(30)

Only the card on top of each stack is visible. For a stack attached to a node \( v \), the top card indicates the current presence of an oriented edge \( uvw \). Present edges form an oriented MTSF with root \( r \), with possibly oriented 2-cycles, i.e., backtracks with two oriented edges of opposite orientations between the same nodes. Next, we proceed as for CyclePopping, that is, cycles are popped or kept in any order, according to the Bernoulli card on top of the cycle’s stack, which is then discarded. When an oriented cycle is popped, the node top cards that form its edges are removed and the next card in each of the corresponding stacks becomes visible. When an oriented cycle is kept, the cards on top of its edges are frozen, in the sense that they will never be discarded. If the weights of cycles are of the form (21), note that 2-cycles are never kept, since their monodromy is 1, and the corresponding stack of Bernoullis thus only contain 0s. The order in which the cycles are popped does not influence the probability distribution of the output of CyclePopping′; see the “colored cycles argument” of Wilson [1996, Theorem 4], see also Lyons and Peres [2017, Lemma 4.2] for a pedagogical presentation.

Note that popping a cycle always yields another oriented MTSF with root \( r \). We emphasize that, so far, we have not shown that CyclePopping′ terminates.

Equivalence with a random walk. Note that this stack-of-cards algorithm CyclePopping′ is equivalent to CyclePopping since the cards in the stacks are distributed in the same way as the random walk (RW) steps (29). A possible way to understand this equivalence is to consider that all the cards are first drawn according to the distribution \( \Pr(X^{(v)}_1 = w) = \Pr(v, w) \) described above and kept fixed. Before the start, the card values are simply unknown. The walker starts from any node and simply reveals sequentially the top value of the card at each step. This value indicates which edge to take for the next step. When the walker’s trajectory cycles, the cycle is popped or kept. In the former case, the top cards of the nodes in the cycle are popped so that the next one in the stacks is visible but still veiled. Thus, the walker’s path simply corresponds to a fixed order for the cycle-popping, which we know to be irrelevant. When the walker reaches an absorbing node, the walk can restart from any non-absorbing node since the RW’s path simply reveals the veiled cards on the stacks.

Termination. Now, we use the equivalence between the stack-of-cards and random walk views. For \( q > 0 \), the algorithm terminates almost surely since the random walker can reach the absorbing node \( r \), even if \( \alpha(\eta^o) = 0 \) for all oriented cycles \( \eta^o \), in which case the output is a random tree with root and the algorithm is Wilson’s.
Sampling probability. Now, we compute the probability that a given oriented MTSF with root $r$ is sampled by following the same strategy as Kassel and Kenyon [2017, Proof of Thm 1]. An oriented MTSF $\mathcal{F}^o$ with root $r$ is the output of the stack-of-cards algorithm provided that there exists a finite set of oriented cycles $\mathcal{S} = \{\eta_1^o, \ldots, \eta_p^o\}$ which are popped and leaves $\mathcal{F}^o$ underneath. Define $\{\text{pop } \mathcal{S}\}$ as the event that the oriented cycles in $\mathcal{S}$ occur and are popped while leaving a valid MTSF underneath. Now, we use that the cards and the Bernoulli variables associated to the cycles are independent; see (29) and (30). Let $\mathcal{F}^{o'}$ denote any other valid MTSF, i.e. a valid assignment of the cards and Bernoulli’s. Then, $\mathcal{F}^{o'}$ can also be an output of CYCLEPopping with the same popping history $\{\text{pop } \mathcal{S}\}$, namely $\Pr(\mathcal{F}^{o'} \text{ occurs and remains}|\text{pop } \mathcal{S}) \neq 0$. Thus, for $\mathcal{F}^o$ valid and $\mathcal{S}$ a valid set of oriented cycles, we have

$$\Pr(\mathcal{F}^o \text{ occurs and remains}|\text{pop } \mathcal{S}) = \prod_{v^o \in \mathcal{F}^o} \Pr(X^{(v)} = v') \prod_{i=1}^k \Pr(B_{\eta_i^o} = 1)$$

$$= \prod_{v^o \in \mathcal{F}^o} \Pr(v, v') \prod_{i=1}^k \alpha(\eta_i^o), \quad (31)$$

where we used that $\alpha(\eta^o)$ is the success probability of Bernoulli($\alpha(\eta^o)$). For a more detailed motivation of (31), we refer to Lyons and Peres [2017, Proof of Theorem 4.1] or Jerrum [2021, Proof of Theorem 1] for a more general argument in the “Partial Rejection Sampling” framework. Next, we use the results in (31) to calculate the probability that CYCLEPopping’ samples $\mathcal{F}^o$ as follows:

$$\Pr(\text{CYCLEPopping’ samples } \mathcal{F}^o) = \sum_{\mathcal{S} = \{\eta_1, \ldots, \eta_p\}} \Pr(\text{pop } \mathcal{S} \text{ and } \mathcal{F}^o \text{ occurs and remains})$$

$$= \sum_{\mathcal{S} = \{\eta_1, \ldots, \eta_p\}} \Pr(\mathcal{F}^o \text{ occurs and remains}|\text{pop } \mathcal{S}) \Pr(\text{pop } \mathcal{S})$$

$$= \sum_{\mathcal{S} = \{\eta_1, \ldots, \eta_p\}} \left( \prod_{v^o \in \mathcal{F}^o} \Pr(v, v') \prod_{i=1}^k \alpha(\eta_i^o) \right) \Pr(\text{pop } \mathcal{S}) \quad (\text{by using (31)})$$

$$= \left( \prod_{v^o \in \mathcal{F}^o} \Pr(v, v') \prod_{i=1}^k \alpha(\eta_i^o) \right) \sum_{\mathcal{S} = \{\eta_1, \ldots, \eta_p\}} \Pr(\text{pop } \mathcal{S}),$$

where $\prod_{v^o \in \mathcal{F}^o} \Pr(v, v') \propto q^{\rho(\mathcal{F}^o)}$ as a consequence of (29). By using the equivalence between CYCLEPopping and CYCLEPopping’, we obtain the desired result.

A.5 Proof of Proposition 5

We consider the event

$$\left\| (\Delta + q I_n)^{-1/2} \left( \sum_{\ell=1}^t \frac{w(F_{\ell})}{\sum_{s=1}^t w(F_s)} \tilde{\Delta}(F_{\ell}) - \Delta \right) \right\|_2 \leq z. \quad (32)$$

To analyse the probability of this event as $t \to \infty$, we first rephrase the problem in a slightly more general way. We introduce the $n \times m$ matrix $\Psi = (\Delta + q I_n)^{-1/2} B^*$. Let $\psi_e$ be the $e$-th column of $\Psi$ and define the following $n \times n$ matrices

$$Y(\mathcal{F}) = \sum_{e \in \mathcal{F}} \frac{1}{l(e)} \psi_e \psi_e^* \quad \text{and} \quad \tilde{Y} = \sum_{e \in \mathcal{F}} \psi_e \psi_e^*,$$

with $l(e) = \psi_e^* \psi_e$ for all $e \in [m]$. Here $\tilde{Y} = \sum_{\mathcal{F} \subseteq [m]} p(\mathcal{F}) Y(\mathcal{F})$. Using the definition of $Y$ and $\Psi$, inequality (32) can be rewritten as

$$\left\| \tilde{Y}_t^{\text{(IS)}} - \tilde{Y} \right\|_2 \leq z \quad \text{with} \quad \tilde{Y}_t^{\text{(IS)}} = \frac{1}{\sum_{s=1}^t w(F_s)} \sum_{\ell=1}^t w(F_{\ell}) Y(F_{\ell}).$$

28
This bound in operator norm is implied by the following bound for the Frobenius norm
\[
\left\| \bar{Y}^{(IS)}_t - \bar{Y}_t \right\|_F \leq z, \tag{33}
\]
the probability of which we shall now investigate.

**Vectorization and writing a CLT.** For a matrix \( A \), denote by \( \text{vec}(A) \) the vector obtained by stacking the columns of \( A \) on top of each other. Let \( y(F) = \text{vec}(Y(F)) \) and \( \bar{y} = \text{vec}(\bar{Y}) \). We aim to approximate the expectation \( \bar{y} = \sum_{\mathcal{F} \subseteq \mathcal{F}} p(\mathcal{F}) y(\mathcal{F}) \), since we cannot sample from \( p(\mathcal{F}) \). The (self-normalized) importance sampling estimator of this expectation writes
\[
\bar{y}_t^{(IS)} = \frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} y(\mathcal{F}_\ell') + \bar{y}_t = \text{vec} \left( \bar{Y}^{(IS)}_t \right).
\]
Since \( \| \bar{y}_t^{(IS)} - \bar{y} \|_F = \| \bar{Y}_t^{(IS)} - \bar{Y} \|_F \), it is enough to control the Frobenius norm of
\[
\frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} \frac{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')}{a} = \frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} \frac{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')}{a},
\]
to be able to establish the bound (33). By the law of large numbers, the limit of the denominator is
\[
\lim_{t \to +\infty} \frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} = \sum_{\mathcal{F}} p_{(IS)}(\mathcal{F}) w(\mathcal{F})/a = 1, \tag{34}
\]
almost surely. Define the properly normalized random vector
\[
x(F') = w(\mathcal{F}') \left( y(F') - \frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} \frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} \frac{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')}{a} \right),
\]
which has expectation zero w.r.t. (22), i.e., \( \sum_{\mathcal{F}} p_{(IS)}(\mathcal{F}) x(\mathcal{F}') = 0 \). Moreover,
\[
\frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} \frac{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')}{a} = \frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} \frac{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')}{a} = \frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} \frac{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')}{a} = \bar{x}_t.
\]
Equivalently, the following equality holds
\[
\frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} \frac{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')}{a} = \frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} \frac{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')}{a},
\]
and implies an equality between squared norms
\[
\left\| \bar{x}_t \right\|_F^2 = \left( \frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} \frac{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')}{a} \right)^2 \left\| \bar{y}_t^{(IS)} - \bar{y} \right\|_F^2 = \left( \frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')} \frac{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell')}{a} \right)^2 \left\| \bar{Y}^{(IS)}_t - \bar{Y} \right\|_F^2. \tag{35}
\]
By using Slutsky’s theorem and a classical CLT, we have the convergence in distribution
\[
\sqrt{t} \left( \bar{y}_t^{(IS)} - \bar{y} \right) \Rightarrow \mathcal{N}(0, \Sigma),
\]
where \( \Sigma = \text{cov}(x) \); see Geweke [1989] for an univariate treatment. The covariance of \( x \) is
\[
\text{cov}(x) = \frac{1}{a^2} \sum_{\mathcal{F}'} p_{(IS)}(\mathcal{F}') w(\mathcal{F}')^2 \left( y(\mathcal{F}') - \bar{y} \right) \left( y(\mathcal{F}') - \bar{y} \right)^* = \sum_{\mathcal{F}'} p(\mathcal{F}') w(\mathcal{F}') \frac{w(\mathcal{F}')}{a} \left( y(\mathcal{F}') - \bar{y} \right) \left( y(\mathcal{F}') - \bar{y} \right)^*. \tag{36}
\]
To obtain an empirical estimate of this covariance, we first remark that \( a = \sum_{\mathcal{F}'} p_{(IS)}(\mathcal{F}') w(\mathcal{F}') \) and thus an estimator of \( a \) is
\[
\hat{a} = \frac{1}{\frac{1}{t} \sum_{\ell=1}^t w(\mathcal{F}_\ell)}. \]
for $\mathcal{F}^\ell_t$ i.i.d. w.r.t. (22). Thus, we build an estimator of the covariance by using the first identity in (36), as follows

$$\hat{\text{cov}}_t(x) = \frac{1}{t} \sum_{\ell=1}^t w(F^\ell_t)^2 \left( y(F^\ell_t) - \bar{y} \right) \left( y(F^\ell_t) - \bar{y} \right)^*. \tag{37}$$

Then, by Slutsky’s theorem again, we obtain the asymptotic CLT

$$\sqrt{t} (\hat{\text{cov}}_t(x))^{-1/2} \bar{x}_t \Rightarrow \mathcal{N}(0, I_n).$$

The rest of the proof consists in bounding the empirical covariance matrix and writing the corresponding confidence ball.

**An upper bound on the empirical covariance matrix.** Note that, since $\|\psi_e \psi^*_e\|_F = \mathbb{I}(e) \leq 1$, we have thanks to a triangle inequality

$$\|y(F^\ell_t) - \bar{y}\|_F = \|Y(F) - \bar{Y}\|_F \leq \sum_{e \in F} \frac{1}{\mathbb{I}(e)} \|\psi_e \psi^*_e\|_F + m \sum_{e=1}^m \|\psi_e \psi^*_e\|_F \leq m + d_{\text{eff}}, \tag{38}$$

where $d_{\text{eff}} = \text{Tr}(\Psi^* \Psi)$. Since, in light of (38), we have

$$\|y(F^\ell_t) - \bar{y}\|_F \leq (m + d_{\text{eff}})^2,$$

an upper bound on this covariance matrix reads

$$\hat{\text{cov}}_t(x) \leq \frac{1}{t} \sum_{\ell=1}^t w(F^\ell_t)^2 \left( m + d_{\text{eff}} \right)^2 I \triangleq \omega_t I, \tag{39}$$

where we used the triangle inequality.

**Confidence ball.** We would like to establish an asymptotic confidence interval. Suppose

$$\|\sqrt{t} (\hat{\text{cov}}_t(x))^{-1/2} \bar{x}_t\|_F^2 \leq z^2,$$

for some $z > 0$. By using the upper bound on the empirical covariance matrix given in (39), we find $\|\bar{x}_t\|_F \leq \frac{z \omega_t}{t}$. By using (35), this bound gives

$$\|\bar{Y}^{(\text{IS})}_t - Y\|_F \leq z \sqrt{\frac{\omega_t}{t}} \left( \frac{1}{t} \sum_{\ell=1}^t w(F^\ell_t)/a \right),$$

where the last factor on the right-hand side tends to 1 asymptotically, see (34). Asymptotically in $t$ and with confidence level $p$, we have the inequality

$$\|\bar{Y}^{(\text{IS})}_t - Y\|_F \leq z \sqrt{\frac{\omega_t}{t}},$$

where $z$ is chosen such that $\Pr(\|u\|_2 \leq z) = p$ for $u \sim \mathcal{N}(0, I_n)$. This proves Proposition 5.

**B Proof of Theorem 2**

We need to prove a matrix Chernoff bound. We strongly rely on the proof techniques in [Kyng and Song, 2018; Kaufman et al., 2022; Tropp, 2015], restated here in the vocabulary of point processes. We specialize arguments to DPPs whenever we can, yielding quicker proofs in some places. In particular, we mix matrix
Throughout this section, we consider a probability measure \( \mu \).

**B.1 Discrete measures, point processes and Corollary, Theorem 2.**

Theorem 3, which we introduce for future reference. We conclude in Appendix B.4 with the proof its cardinality. We remove the latter restriction using a dilation argument due to Lyons [2003]. This provides \( \xi \) Chernoff bound. This bound features Tropp’s intrinsic dimension, applied to the DPP kernel. However, at this stage, the bound only applies to \( k \)-homogeneous DPPs, that is, DPPs with almost surely fixed cardinality. We remove the latter restriction using a dilation argument due to Lyons [2003]. This provides Theorem 3, which we introduce for future reference. We conclude in Appendix B.4 with the proof its corollary, Theorem 2.

### B.1 Discrete measures, point processes and \( \ell_\infty \)-independence

Throughout this section, we consider a probability measure \( \mu \) on \( \{0,1\}^m \). Recall that \( \mu \) is said to be \( k \)-homogeneous if the Boolean vector \( \xi \sim \mu \) almost surely has exactly \( k \) entries equal to 1. Finally, for \( \xi \sim \mu \), we consider

\[
\mathcal{X} = \mathcal{X}(\xi) \triangleq \{ i \in [m] : \xi_i = 1 \} \subset [m].
\]

(40)

\( \mathcal{X} \) is a point process on \([m] \), i.e., a random subset of \([m] \). Similarly, we can talk of the Boolean measure \( \mu \) associated to a point process \( \mathcal{X} \).

#### B.1.1 Intensity and density

For a point process \( \mathcal{X} \), define its **intensity** as \( \rho_{\mathcal{X}} : [m] \to \mathbb{R}_{\geq 0} \) given by

\[
\rho_{\mathcal{X}}(i) = \mathbb{E}[1_{\mathcal{X}(i)}], \quad i \in [m].
\]

**Example 1.** Definition 1 implies that for \( \mathcal{X} \sim DPP(K) \), \( \rho_{\mathcal{X}}(i) = K_{ii} \) for any \( i \in [m] \).

Furthermore, if \( \mathcal{X} \) is \( k \)-homogeneous, its **density** at \( i \in [m] \) is \( \nu_{\mathcal{X}}(i) = \rho_{\mathcal{X}}(i)/k \). Intuitively, the intensity \( \rho_{\mathcal{X}}(i) \) is the probability that \( i \) belongs to a sample of \( \mathcal{X} \), whereas the density \( \nu_{\mathcal{X}} \) is the 1-homogeneous point process whose samples are obtained by first getting a sample of \( k \) points from \( \mathcal{X} \) and then sampling uniformly one of these random \( k \) points.

#### B.1.2 Conditioning

To define \( \ell_\infty \)-independence, we need to express conditioning. First, for any \( A_1 \subset [m] \) with \( \mu(A_1) \neq 0 \), define the reduced probability measure \( \mu_{A_1} \) on \([m] \setminus A_1 \) by

\[
\mu_{A_1}(B) \propto \mu(A_1 \cup B), \quad B \subseteq [m] \setminus A_1.
\]

This reduced measure translates fixing some of the components \( \xi \sim \mu \) to 1. The “dual” operation consists in the exclusion of \( A_0 \subset [m] \), where \( A_0 \) is such that \( \mu(F) < 1 \) for all subsets \( F \subseteq A_0 \). We denote the corresponding reduced measure on \([m] \setminus A_0 \) as

\[
\mu_{A_1}^{A_0}(B) \propto \mu(B), \quad B \subseteq [m] \setminus A_0.
\]

This corresponds to fixing \( \xi_i = 0 \) for all \( i \in A_0 \). For disjoint \( A_0, A_1 \subset [m] \), we can further define the reduced probability measure \( \mu_{A_1}^{A_0} \) corresponding to both excluding \( A_0 \) and including \( A_1 \) by

\[
\mu_{A_1}^{A_0}(B) \propto \mu(A_1 \cup B), \quad B \subseteq [m] \setminus (A_0 \cup A_1).
\]

The notation is consistent, in that, for instance, \( \mu_{A_1}^{A_1} = \mu_{A_1} \). Moreover, the correspondence (40) between point processes and measures allows to define the point process \( \mathcal{X}_{A_1}^{A_0} \) associated to the reduced measure \( \mu_{A_1}^{A_0} \). Finally, for simplicity, for \( e \in [m] \), we respectively write \( \mathcal{X}_e \) and \( \mathcal{X}^e \) instead of \( \mathcal{X}_{\{e\}} \) and \( \mathcal{X}^{\{e\}} \).
Example 2 (Example 1, continued). Let $\mathcal{X} \sim \text{DPP}(K)$ with symmetric kernel $K$, and $e \in [m]$ such that $\rho_X(e) = K_{ee} > 0$, then $\mathcal{X}_e$ is still a DPP, with kernel $f, g \mapsto K_{fg} - K_{fe}K_{ge}/K_{ee}$; see [Shirai and Takahashi, 2003]. More generally, for any disjoint $\mathcal{A}_0, \mathcal{A}_1$ such that $\mathcal{A}_1$ is in the support of $\text{DPP}(K)$, then $\mathcal{X}_{\mathcal{A}_0}^{\mathcal{A}_1}$ exists and is a DPP, see e.g. [Kulesza and Taskar, 2012] for details.

B.1.3 $\ell_\infty$-independence

Kaufman et al. [2022] introduced $\ell_\infty$-independence for Boolean measures, and we restate it here in terms of point processes. Given a point process $\mathcal{X}$ on $[m]$ with intensity $\rho_X$, define

$$\delta_{X,e}(i) = \begin{cases} 1 - \rho_X(e) & \text{if } i = e, \\ \rho_X(i) - \rho_X(e) & \text{otherwise,} \end{cases}$$

(41)

for all $e, i \in \text{supp}(\rho_X)$.

Definition 2 ($\ell_\infty$-independence). We say that $\mu$ is $\ell_\infty$-independent with parameter (w.p.) $d_{\text{inf}}$ if for all disjoint subsets $\mathcal{A}_0$ and $\mathcal{A}_1$ of $[m]$ such that the point process $\mathcal{Y} = \mathcal{X}^{\mathcal{A}_0}_{\mathcal{A}_1}$ exists, for all $e \in \text{supp}(\rho_Y)$,

$$\sum_{i \in \text{supp}(\rho_Y)} |\delta_{\mathcal{Y},e}(i)| \leq d_{\text{inf}}.$$ 

Furthermore, to better interpret $\ell_\infty$-independence, it is useful to see a point process as a random counting measure. Indeed, the quantity that is upper bounded in Definition 2 is the total variation distance

$$d_{\infty}(\delta_e + E_{X_{e}} \left[ \sum_{i \in X_{e}} \delta_{i} \right], E_{X} \left[ \sum_{j \in X} \delta_{j} \right]) = \sum_{i \in \text{supp}(\rho_X)} |\delta_{X,e}(i)|.$$ 

DPPs\(^5\) satisfy $\ell_\infty$-independence with parameter $d_{\text{inf}} = 2$. This is the content of Example 3.

Example 3 (Examples 1 and 2, continued). Let $\mathcal{X} \sim \text{DPP}(K)$ for some symmetric $m \times m$ matrix $K$. Then $\rho_X(e) = K_{ee} \leq 1$ for all $e \in [m]$. Furthermore, for all $e \in [m]$ such that $K_{ee} > 0$, we have $\rho_X(i) = K_{ii} - K_{ie}^2/K_{ee}$ for all $i \in [m]$ such that $i \neq e$. We deduce that, for $e \in [m]$ such that $K_{ee} > 0$,

$$1 - \rho_X(e) = 1 - K_{ee} \geq 0,$$

and that, for $i \neq e$,

$$\rho_X(i) - \rho_X(e) = K_{ei}K_{ie}/K_{ee} = (K_{ie})^2/K_{ee} \geq 0.$$ 

It follows that

$$\sum_{i \neq e} |\rho_X(i) - \rho_X(e)| = \sum_{i \neq e} K_{ei}K_{ie}/K_{ee} = (K_{ee}^2 - (K_{ee})^2)/K_{ee}.$$ 

Now, recall that by the Macchi-Soshnikov theorem, existence of $\mathcal{X}$ implies $K^2 \preceq K$; see Section 4. We thus obtain, using notation (41),

$$\sum_{i \in \text{supp}(\rho_X)} |\delta_{X,e}(i)| = 1 - K_{ee} + (K_{ee}^2 - (K_{ee})^2)/K_{ee} \leq 2 - 2K_{ee},$$

(42)

for all $e \in [m]$ such that $K_{ee} > 0$.

Now, as noted in Example 2, if $\mathcal{X}$ is a DPP, then for any disjoint subsets $\mathcal{A}_0$ and $\mathcal{A}_1$ of $[m]$ such that $\mathcal{X}_{\mathcal{A}_0}^{\mathcal{A}_1}$ exists, $\mathcal{X}_{\mathcal{A}_1}^{\mathcal{A}_0}$ is a DPP. We can thus prove (42) for this new DPP. Consequently, a DPP satisfies $\ell_\infty$-independence with parameter $d_{\text{inf}} = 2$.

Finally, note that $k$-homogeneous DPPs with symmetric kernels are actually those for which $K^2 = K$, so called projection, projective, or elementary DPPs, see, e.g., [Hough et al., 2006]. In particular, we note that the bound in (42) is tight if $\mathcal{X}$ is $k$-homogeneous. In that case, for a fixed $i \in \text{supp}(\rho_X)$, we also have $E_X[|\delta_{X,e}(i)|] = 0$.

\(^5\)And, more generally, measures that satisfy the stochastic covering property [Pemantle and Peres, 2014].
We end this section on DPPs and independence by a lemma relating $\ell_\infty$-independence and marginals. In [Kaufman et al., 2022], this lemma is a consequence of what the authors call average multiplicative independence. For our limited use, we can avoid defining this new notion of independence.

**Lemma 1.** Let $\mathcal{X}$ be a point process on $[m]$, which is $k$-homogeneous and $\ell_\infty$-independent with parameter $d_{\inf}$. Denote by $\rho_{\mathcal{X}}$ and $\nu_{\mathcal{X}}$ its intensity and density; see Appendix B.1.1. For all $e$ and $i \in \text{supp}(\rho_{\mathcal{X}})$, using notation (41), it holds

$$\mathbb{E}_{e \sim \nu_{\mathcal{X}}} |\delta_{\mathcal{X},e}(i)| \leq d_{\inf} \nu_{\mathcal{X}}(i), \text{ for all } i \in \text{supp}(\rho_{\mathcal{X}}).$$

Furthermore, if $\mathcal{X} \sim \text{DPP}(K)$ for some orthogonal projector $K$, then $\mathbb{E}_{e \sim \nu_{\mathcal{X}}}|\delta_{\mathcal{X},e}(i)| = 0$, and

$$\mathbb{E}_{e \sim \nu_{\mathcal{X}}}|\delta_{\mathcal{X},e}(i)| = d_{\mathcal{X}}(i) \nu_{\mathcal{X}}(i), \text{ for all } i \in \text{supp}(\rho_{\mathcal{X}}),$$

where $d_{\mathcal{X}}(i) = 2 - 2K_{ii}$.

For completeness, Lemma 1 is proved in Appendix B.5.1.

**B.2 Manipulating the matrix Laplace transform**

Consider a set of Hermitian $d \times d$ matrices satisfying $0 \leq Y_i \leq rI$ for $1 \leq i \leq m$ and some $r > 0$. Further let $\mathcal{X}$ be a $k$-homogeneous point process on $[m]$. In order to study the concentration of $\sum_{i \in \mathcal{X}} Y_i$ about its expectation for a $k$-homogeneous $\mathcal{X}$, we provide Lemma 2 below. The case $r = 1$ has already been considered by Kaufman et al. [2022]. To make our manuscript explicit and self-contained, we extend this analysis to any $r > 0$.

**Lemma 2 (Extension of Lemma 5.2 in Kaufman et al. [2022]).** Let $\mathcal{X}$ be a $k$-homogeneous point process on $[m]$, which is $\ell_\infty$-independent with parameter $d_{\inf}$. Consider a set of Hermitian $d \times d$ matrices satisfying $0 \leq Y_i \leq rI$ for $1 \leq i \leq m$ and some $r > 0$. Let $c \geq rd_{\inf} \sqrt{\frac{r}{2}}(9d_{\inf}^2 + 1)$. Then, for every $\theta \in [-1/(2c), 1/(2c)]$ and for every Hermitian matrix $H$, the following holds

$$\mathbb{E}_{\mathcal{X}} \text{Tr} \exp \left( H + \theta \sum_{i \in \mathcal{X}} Y_i \right) \leq \text{Tr} \exp \left( H + (\theta + c\theta^2) \mathbb{E}_{\mathcal{X}} \left[ \sum_{i \in \mathcal{X}} Y_i \right] \right).$$

(43)

The proof of Lemma 2 is by induction. Before we proceed, we derive two intermediate results, Fact 1 and Lemma 3.

Fact 1 is an adaptation of Kaufman et al. [2022, Claim 5.4, arxiv v2]. It allows controlling the difference between an expected sum of matrices sampled w.r.t. a $k$-homogeneous point process and the same sum conditioned on a certain term being already present. For any $e$ such that $\rho_{\mathcal{X}}(e) > 0$, define the increment

$$Z_e(\mathcal{X}) = \mathbb{E}_{\mathcal{X}_e} \left[ Y_e + \sum_{i \in \mathcal{X}_e} Y_i \right] - \mathbb{E}_{\mathcal{X}} \left[ \sum_{i \in \mathcal{X}} Y_i \right] = \sum_{i \in \text{supp}(\rho_{\mathcal{X}})} \delta_{\mathcal{X},e}(i) Y_i,$$

(44)

where $\mathcal{X}_e$ is the reduced process and $\delta_{\mathcal{X},e}(i)$ is given in (41). This quantity can be interpreted as the difference sequence of a matrix martingale associated to the sequential revelation of $\mathcal{X} = \{e_1, \ldots, e_k\}$ by the chain rule. Indeed, note that $\mathbb{E}_{e \sim \nu_{\mathcal{X}}}[Z_e(\mathcal{X})] = 0$, and define the martingale

$$M_j = Y_{e_1} + \cdots + Y_{e_j} + \mathbb{E}_{\mathcal{X}_{e_1, \ldots, e_j}} \left[ \sum_{i \in \mathcal{X}_{e_1, \ldots, e_j}} Y_i \right] - \mathbb{E}_{\mathcal{X}} \left[ \sum_{i \in \mathcal{X}} Y_i \right].$$

Then $Z_{e_j}(\mathcal{X}_{e_1, \ldots, e_{j-1}}) = M_j - M_{j-1}$ for $j \in [k]$, with the initialization $M_0 = 0$. Note that $\mathcal{X}_{e_1, \ldots, e_k}$ is simply the empty set. In particular, a telescopic sum argument yields

$$M_k = Z_{e_k}(\mathcal{X}_{e_1, \ldots, e_{k-1}}) + \cdots + Z_{e_2}(\mathcal{X}_{e_1}) + Z_{e_1}(\mathcal{X}) = \sum_{i \in \mathcal{X}} Y_i - \mathbb{E}_{\mathcal{X}} \left[ \sum_{i \in \mathcal{X}} Y_i \right],$$

which will be implicitly used below in the proof of Lemma 2.

---

This martingale appears in [Pemantle and Peres, 2014] in the scalar case.
Fact 1. Let \( k \geq 1 \) and \( \mathcal{X} \) be a \( k \)-homogeneous point process on \([m]\), which is \( \ell_\infty \)-independent with parameter \( d_{\text{inf}} \). Let \( 0 \leq Y_i \leq rI \) for \( 1 \leq i \leq m \). Let \( e \) be such that \( \rho_X(e) > 0 \) and \( Z_e = Z_e(X) \) be defined in (44). Then

(i) \( |Z_e| \leq rd_{\text{inf}}I \),

(ii) \( \mathbb{E}_{e \sim \nu_X} [Z_e^2] \leq rd_{\text{inf}}^2 \mathbb{E}_{e \sim \nu_X}[Y_e] \),

where \( \nu_X \) is the intensity of \( \mathcal{X} \).

Fact 1 straightforwardly relies on Lemma 1, and we give a proof in Appendix B.5.2. Now, we use Fact 1 to derive the following lemma with explicit constants.

Lemma 3 (Extension of Lemma 5.3 in Kaufman et al. [2022]). Under the assumptions of Fact 1, if \( c \geq rd_{\text{inf}} \sqrt{\frac{9d_{\text{inf}}^2 + 1}{2}} \), then for all \( \theta \in [-1/(2c), 1/(2c)] \),

\[
\mathbb{E}_{e \sim \nu_X} \left[ e^{(\theta + c\theta^2)Z_e - c\theta^2Y_e} \right] \leq I.
\]

Proof. The idea of this proof is to use \( \exp(A - B) \leq I + A - B + 2A^2 + 2B^2 \) for \( A \) and \( B \) symmetric such that \( A \leq I \) and \( B \geq 0 \) [Kaufman et al., 2022, Fact 4.2]. Here we choose \( A = (\theta + c\theta^2)Z_e \) and \( B = c\theta^2Y_e \).

First, we have

\[
(\theta + c\theta^2)Z_e \leq \frac{3}{4c}Z_e \quad \text{(if } \theta \leq 1/(2c))
\]

\[
\leq \frac{1}{c} \quad \text{by Fact 1 (i)}
\]

\[
\leq I. \quad \text{(for } c \geq rd_{\text{inf}})
\]

Then, we find

\[
\mathbb{E}_{e \sim \nu_X} \left[ e^{(\theta + c\theta^2)Z_e - c\theta^2Y_e} \right] \leq \mathbb{E}_{e \sim \nu_X} \left[ I + (\theta + c\theta^2)Z_e - c\theta^2Y_e + 2(\theta + c\theta^2)^2Z_e^2 + 2(c\theta^2)^2Y_e^2 \right]
\]

\[
\leq I - c\theta^2\mathbb{E}_{e \sim \nu_X}[Y_e] + 9(\theta^2\mathbb{E}_{e \sim \nu_X}[Z_e^2] + \frac{1}{2}\theta^2r\mathbb{E}_{e \sim \nu_X}[Y_e]) \quad \text{(since } |\theta| \leq 1/(2c) \text{ and } Y_e \leq rI)
\]

\[
\leq I - \theta^2\mathbb{E}_{e \sim \nu_X}[Y_e] \left( c - \frac{9}{2}rd_{\text{inf}}^2 - \frac{1}{2}r \right) \quad \text{(by Fact 1 (ii))}
\]

\[
\leq I,
\]

where we used that \( \mathbb{E}_{e \sim \nu_X}[Z_e] = 0 \) and where the last inequality is due to \( c \geq \frac{9}{2}rd_{\text{inf}}^2 + \frac{1}{2}r \). \( \square \)

We are now ready to prove Lemma 2. This proof simply follows the argument of Kaufman et al. [2022, Lemma 5.2] and is given below in order to make this paper self-contained. The only changes are as follows: the upper bound \( Y_i \leq rI \) and the condition on \( c \) which is derived in Lemma 3.

Proof of Lemma 2. We proceed by induction over the degree of homogeneity of the point process, using the chain rule.

First, we show that the statement is true if \( k = 1 \), i.e., a point process with almost surely one point. In that case, the left-hand side of (43) reduces to

\[
\mathbb{E}_X \text{Tr } \exp \left( H + \theta \sum_{i \in \mathcal{X}} Y_i \right) = \mathbb{E}_{e \sim \nu_X} \text{Tr } \exp (H + \theta Y_e).
\]

We now add and subtract the quantity that appears at the right-hand side of (43) and write

\[
\theta Y_e = (\theta + c\theta^2)\mathbb{E}_{e' \sim \nu_X}Y_{e'} + (\theta + c\theta^2)Z_e - c\theta^2Y_e.
\]

\footnote{It is worth noticing that the proof technique of the Freedman inequality for matrix martingales obtained by Oliveira [2009, Theorem 1.2] uses the Golden-Thomson inequality in a similar way as Kaufman et al. [2022] and in the proof of Lemma 2 below.}
where $Z_e = Y_e - \mathbb{E}_{e' \sim \nu_X} Y_{e'}$. Identity (45) and the Golden-Thomson inequality\footnote{For $A$ and $B$ Hermitian matrices, $\text{Tr}(\exp(A + B)) \leq \text{Tr}(\exp(A) \exp(B))$. To our knowledge, this technique is due to Ahlswede and Winter [2002].} together yield
\[
\mathbb{E}_{e \sim \nu_X} \text{Tr}(H + \theta Y_e) \leq \text{Tr} \left[ \exp(H + (\theta + c\theta^2) \mathbb{E}_{e' \sim \nu_X} Y_{e'}) \mathbb{E}_{e \sim \nu_X} \exp((\theta + c\theta^2) Z_e - c\theta^2 Y_e) \right]
\]
\[
\leq \text{Tr} \left[ \exp(H + (\theta + c\theta^2) \mathbb{E}_{e' \sim \nu_X} Y_{e'}) \right] \mathbb{E}_{e \sim \nu_X} \exp((\theta + c\theta^2) Z_e - c\theta^2 Y_e) \|_{\text{op}} \right],
\]
since $\exp(H + (\theta + c\theta^2) \mathbb{E}_{e' \sim \nu_X} Y_{e'})$ is positive semidefinite. By Lemma 3, if $c \geq r_{d_{\text{inf}}} \lor 1/(2\epsilon)$, for all $\theta \in [-1/(2\epsilon), 1/(2\epsilon)]$, we have the following bound
\[
\|\mathbb{E}_{e \sim \nu_X} \exp((\theta + c\theta^2) Z_e - c\theta^2 Y_e)\|_{\text{op}} \leq 1,
\]
which can be substituted in (46) to give the desired statement,
\[
\mathbb{E}_X \text{Tr} \exp \left( H + \theta \sum_{i \in \mathcal{X}} Y_i \right) = \mathbb{E}_{e \sim \nu_X} \text{Tr} \exp(H + \theta Y_e)
\]
\[
\leq \text{Tr} \left( \exp(H + (\theta + c\theta^2) \mathbb{E}_{e' \sim \nu_X} Y_{e'}) \right)
\]
\[
= \text{Tr} \exp \left( H + (\theta + c\theta^2) \mathbb{E}_X \left[ \sum_{i \in \mathcal{X}} Y_i \right] \right).
\]
Now, let $l \geq 2$, and assume Lemma 2 is true for $k = l - 1$. Then, the chain rule and the induction hypothesis applied to the $(l - 1)$-homogeneous point process $\mathcal{X}_e$ yield
\[
\mathbb{E}_X \text{Tr} \exp \left( H + \theta \sum_{i \in \mathcal{X}} Y_i \right) = \mathbb{E}_{e \sim \nu_X} \mathbb{E}_{X_e} \text{Tr} \exp \left( H + \theta Y_e + \theta \sum_{i \in \mathcal{X}_e} Y_i \right)
\]
\[
\leq \mathbb{E}_{e \sim \nu_X} \text{Tr} \exp \left( H + \theta Y_e + (\theta + c\theta^2) \mathbb{E}_{X_e} \left[ \sum_{i \in \mathcal{X}_e} Y_i \right] \right).
\]
Now we write the argument of the exponential as follows
\[
\theta Y_e + (\theta + c\theta^2) \mathbb{E}_{X_e} \left[ \sum_{i \in \mathcal{X}_e} Y_i \right] = (\theta + c\theta^2) \mathbb{E}_{X_e} \left[ Y_e + \sum_{i \in \mathcal{X}_e} Y_i \right] - c\theta^2 Y_e
\]
\[
= (\theta + c\theta^2) \mathbb{E}_{X_e} \left[ \sum_{i \in \mathcal{X}_e} Y_i \right] + (\theta + c\theta^2) Z_e - c\theta^2 Y_e,
\]
where $Z_e$ is defined in Fact 1. Using the Golden-Thomson inequality again, we deduce
\[
\mathbb{E}_X \text{Tr} \exp \left( H + \theta \sum_{i \in \mathcal{X}} Y_i \right) \leq \mathbb{E}_{e \sim \nu_X} \text{Tr} \exp \left( H + (\theta + c\theta^2) \mathbb{E}_{X_e} \left[ \sum_{i \in \mathcal{X}_e} Y_i \right] + (\theta + c\theta^2) Z_e - c\theta^2 Y_e \right)
\]
\[
\leq \text{Tr} \left( \exp \left( H + (\theta + c\theta^2) \mathbb{E}_{X_e} \left[ \sum_{i \in \mathcal{X}_e} Y_i \right] \right) \right) \mathbb{E}_{e \sim \nu_X} \exp \left( (\theta + c\theta^2) Z_e - c\theta^2 Y_e \right)
\]
\[
\leq \text{Tr} \exp \left( H + (\theta + c\theta^2) \mathbb{E}_{X_e} \left[ \sum_{i \in \mathcal{X}_e} Y_i \right] \right) \|\mathbb{E}_{e \sim \nu_X} \exp \left( (\theta + c\theta^2) Z_e - c\theta^2 Y_e \right)\|_{\text{op}}
\]
\[
\leq \text{Tr} \exp \left( H + (\theta + c\theta^2) \mathbb{E}_{X_e} \left[ \sum_{i \in \mathcal{X}_e} Y_i \right] \right),
\]
where the last bound is due to Lemma 3. This concludes the proof.
B.3 Chernoff bounds

Along the lines of Kaufman et al. [2022], we now use Lemma 2 to obtain a deviation bound on the sum of matrices indexed by a $k$-homogeneous point process.

**Theorem 3 (Intrinsic Matrix Chernoff bound).** Let $\mathcal{X}$ be a point process on $[m]$. Assume that $\mathcal{X}$ is $k$-homogeneous and $\ell_\infty$-independent with parameter $d_{\text{inf}}$. Let $Y_1, \ldots, Y_m$ be a collection of $d \times d$ Hermitian matrices such that $0 \preceq Y_i \preceq rI$ for all $i \in [m]$ and for some $r > 0$. Assume $\mathbb{E}_X \left[ \sum_{i \in \mathcal{X}} Y_i \right] \preceq M$ for some Hermitian matrix $M$. Denote $\operatorname{intdim}(M) = \operatorname{Tr}(M)/\|M\|_{\text{op}}$ and $\kappa = \|M\|_{\text{op}}$. Let $c = rd_{\text{inf}} \vee \frac{r}{2}(9d_{\text{inf}}^2 + 1)$. Then, for all $\varepsilon \in (0, 1]$, it holds

$$\Pr \left[ \left\| \sum_{i \in \mathcal{X}} Y_i - \mathbb{E}_X \left[ \sum_{i \in \mathcal{X}} Y_i \right] \right\|_{\text{op}} \geq \varepsilon \kappa \right] \leq 2^{\operatorname{intdim}(M)} \left( 1 + \frac{12c^2}{\kappa^2 \varepsilon^4} \right) \exp \left( -\frac{\varepsilon^2 \kappa}{4c} \right).$$

Here are the main differences w.r.t. Kaufman et al. [2022]. First, in their assumptions, the authors of the latter paper suppose there exist two non-negative constants $\mu_1$ and $\mu_2$ such that $\mu_1 I \leq \mathbb{E}_X \left[ \sum_{i \in \mathcal{X}} Y_i \right] \leq \mu_2 I$, as well as assume $\|Y_i\|_{\text{op}} \leq 1$ for all $i \in [m]$. We relax both assumptions. Second, we replace the matrix dimension $d$ by the intrinsic dimension of Tropp [2015]. Third, the matrices here are Hermitian rather than symmetric.

**Proof.** The proof consists of four steps. First, we apply the matrix Laplace transform bound of Tropp [2015, Proposition 7.4.1]; see also Ahlswede and Winter [2002]. Let $U$ be a random Hermitian matrix, let $\psi : \mathbb{R} \to \mathbb{R}_{\geq 0}$ be a nonnegative function that is nondecreasing on $[0, \infty)$, then for each $t \geq 0$, we have

$$\Pr [\lambda_{\max}(U) \geq t] \leq \frac{1}{\psi(t)} \mathbb{E}_X [\operatorname{Tr} \psi(U)].$$

(47)

Let $\theta > 0$, and apply the bound to the centered random variable

$$U = \sum_{i \in \mathcal{X}} Y_i - \mathbb{E}_X \left[ \sum_{i \in \mathcal{X}} Y_i \right],$$

(48)

with $^9 \psi(t) = e^{\theta t} - t\theta - 1$. It comes

$$\Pr \left[ \lambda_{\max} \left( \sum_{i \in \mathcal{X}} Y_i - \mathbb{E}_X \left[ \sum_{i \in \mathcal{X}} Y_i \right] \right) \geq t \right] \leq \frac{1}{e^{\theta t} - t\theta - 1} \mathbb{E}_X \operatorname{Tr} \left( \exp \left( \theta \left( \sum_{i \in \mathcal{X}} Y_i - \mathbb{E}_X \left[ \sum_{i \in \mathcal{X}} Y_i \right] \right) \right) - I \right).$$

(49)

The second step of the proof consists in using $\ell_\infty$-independence to bound the right-hand side of (49), following the same strategy as Kaufman et al. [2022, Proof of Thm 5.1]. Recall that Lemma 2 yields

$$\mathbb{E}_X \operatorname{Tr} \exp \left( H + \theta' \sum_{i \in \mathcal{X}} Y_i \right) \leq \operatorname{Tr} \exp \left( H + (\theta' + c\theta'^2) \mathbb{E}_X \left[ \sum_{i \in \mathcal{X}} Y_i \right] \right),$$

(50)

for any Hermitian $H$, and for all $\theta' \in \left[ \frac{1}{2c}, \frac{1}{2c} \right]$, where $c = rd_{\text{inf}} \vee \frac{r}{2}(9d_{\text{inf}}^2 + 1)$. Taking $\theta' = \theta$ and $H = -\theta \mathbb{E}_X \left[ \sum_{i \in \mathcal{X}} Y_i \right]$, (49) yields

$$\Pr \left[ \lambda_{\max} \left( \sum_{i \in \mathcal{X}} Y_i - \mathbb{E}_X \left[ \sum_{i \in \mathcal{X}} Y_i \right] \right) \geq t \right] \leq \frac{1}{e^{\theta t} - t\theta - 1} \operatorname{Tr} \left( \exp \left( -c\theta^2 H \right) - I \right) \leq \frac{1}{e^{\theta t} - t\theta - 1} \operatorname{Tr} \left( c\theta^2 M - I \right),$$

(51)

(52)

for all $\theta \in (0, \frac{1}{2c}]$.

---

^9Here we differ from Kaufman et al. [2022] who take $\psi(t) = e^{\theta t}$. 
The third step is in the introduction of the intrinsic dimension, to further bound the right-hand side of (52). Let \( \varphi \) be a convex function on the interval \([0, \infty)\), and assume that \( \varphi(0) = 0 \). For any positive-semidefinite matrix \( A \), \citep[Lemma 7.5.1]{tropp2015} asserts that \( \text{Tr} \varphi(A) \leq \text{intdim}(A) \varphi(\|A\|) \). Take here \( \varphi(a) = e^a - 1 \), and define \( g(\theta) = e^{\theta^2} \). We obtain

\[
\text{Tr} \left( \exp (g(\theta)M) - I \right) \leq \text{intdim}(M) \varphi \left( g(\theta) \|M\| \right) = \text{intdim}(M) \varphi \left( g(\theta) \kappa \right),
\]

where we used that \( \text{intdim}(\alpha A) = \text{intdim}(A) \) for all \( \alpha \neq 0 \) and \( A \geq 0 \). By a direct substitution, we obtain

\[
\Pr \left[ \lambda_{\max} \left( \sum_{i \in \mathcal{X}} Y_i - \mathbb{E}_{\mathcal{X}} \left[ \sum_{i \in \mathcal{X}} Y_i \right] \right) \geq t \right] \leq \text{intdim}(M) \frac{e^{g(\theta)\kappa}}{e^{\theta t} - t \theta - 1},
\]

for all \( \theta \in (0, \frac{1}{2c}] \).

The fourth step consists in simplifying the obtained bound. Following \cite{tropp2015}, Eq (7.7.4), we have

\[
\frac{e^{g(\theta)\kappa}}{e^{\theta t} - t \theta - 1} = \frac{e^{\theta t}}{e^{\theta t} - t \theta - 1} e^{-\theta t + g(\theta)\kappa} \leq \left( 1 + \frac{3}{\theta^2 t^2} \right) e^{-\theta t + g(\theta)\kappa},
\]

for any positive \( \theta \). Now, we let \( t = \kappa \varepsilon \) with \( \varepsilon \in (0, 1] \). We further note that the argument of the exponential on the right-hand side of (54) is minimized by \( \theta_* = \frac{\varepsilon}{2c} \), so that \( 0 < \theta_* \leq \frac{1}{2c} \). Thus, it holds

\[
\Pr \left[ \lambda_{\max} \left( \sum_{i \in \mathcal{X}} Y_i - \mathbb{E}_{\mathcal{X}} \left[ \sum_{i \in \mathcal{X}} Y_i \right] \right) \geq \varepsilon \kappa \right] \leq \text{intdim}(M) \left( 1 + \frac{12 \varepsilon^2}{\kappa^2 \varepsilon^4} \right) \exp \left( -\frac{\varepsilon^2 \kappa}{4c} \right).
\]

The same deviation bound for \( \lambda_{\max} \left( \mathbb{E}_{\mathcal{X}} \left[ \sum_{i \in \mathcal{X}} Y_i \right] - \sum_{i \in \mathcal{X}} Y_i \right) \) can be obtained along the same four steps, thus proving the statement by a union bound.

\[\square\]

### B.4 A dilation argument to conclude

We now remove the \( k \)-homogeneity assumption in Theorem 3 using a homogenization technique that is proper to DPPs.

**Proof of Theorem 2.** While \( \mathcal{X} \) is any DPP on \([m]\), we follow Lyons [2003] in considering the DPP \( \mathcal{X}' \) on \([2m]\) with correlation kernel

\[
K' = \begin{pmatrix}
K & \sqrt{1-K} \sqrt{K} \\
\sqrt{1-K} \sqrt{K} & 1-K
\end{pmatrix}.
\]

By construction, \( K'^2 = K' \), and \( K' \) has rank \( m \), therefore \( \mathcal{X}' \) is an \( m \)-homogeneous DPP, see, e.g., \citep{kulesza2012}.

Now define the matrices \( Y_i = 0 \) for all \( i \in \{m+1, \ldots, 2m\} \). Elementary properties of DPPs imply that \( \mathcal{X}' \cap [m] \) has distribution DPP(\( K \)); see again, for instance, \citep{kulesza2012}. In particular, the law of

\[
\sum_{i \in \mathcal{X}'} Y_i = \sum_{i \in \mathcal{X}'} 1(i \in [m]) Y_i,
\]

is the same as the law of \( \sum_{i \in \mathcal{X}} Y_i \), and it is enough to prove a deviation bound for \( \sum_{i \in \mathcal{X}} Y_i \). We simply check that Theorem 3 applies. First,

\[
\mathbb{E}_{\mathcal{X}'} \left[ \sum_{i \in \mathcal{X}'} Y_i \right] = \mathbb{E}_{\mathcal{X}} \left[ \sum_{i \in \mathcal{X}} Y_i \right] \leq M.
\]

Additionally, since \( \mathcal{X}' \) is a DPP, it is \( \ell_\infty \)-independent with parameter \( d_{\text{inf}} = 2 \); see Example 3. Theorem 3 thus applies, with the constant \( c = r d_{\text{inf}} \sqrt{2} \left( \theta d_{\text{inf}}^2 + 1 \right) \) where \( d_{\text{inf}} = 2 \). For reference, the numerical values of the absolute constants appearing in Theorem 2 are \( c_1 = 12c^2/r^2 = 3 \cdot 37^2 \) and \( c_2 = \frac{r}{4c} = \frac{1}{2 \cdot 37} \). \[\square\]
B.5 Additional technical details

This section contains a few straightforward derivations, for self-reference.

B.5.1 Proof of Lemma 1

Let $X$ be a $k$-homogeneous point process on $[m]$ for some integer $k \geq 1$. Recall the intensity of $X$ is the probability mass function $\nu_X(e) = \rho_X(e)/k$ for all $e \in [m]$, as given in Appendix B.1.1. First, we notice that for all $i$ and $j$ with $i \neq j$ such that $\rho_X(i) > 0$ and $\rho_X(j) > 0$, we have

$$\nu_X(i)\rho_X(j) = \nu_X(j)\rho_X(i).$$

Now, recall the notation

$$\delta_{X,e}(i) = \begin{cases} 1 - \rho_X(e) & \text{if } i = e, \\ \rho_X(e) - \rho_X(i) & \text{otherwise.} \end{cases}$$

If $\rho_X(i) > 0$, it holds that

$$\mathbb{E}_{e \sim \nu_X} |\delta_{X,e}(i)| = \nu_X(i)|1 - \rho_X(i)| + \sum_{e \neq i, \rho_X(e) > 0} \nu_X(e)|\rho_X(e) - \rho_X(i)|$$

$$= \nu_X(i)|1 - \rho_X(i)| + \sum_{e \neq i, \rho_X(e) > 0} |\nu_X(e)\rho_X(i) - \nu_X(e)\rho_X(i)|$$

$$= \nu_X(i)|1 - \rho_X(i)| + \sum_{e \neq i, \rho_X(e) > 0} |\nu_X(i)\rho_X(e) - \frac{\rho_X(e)\rho_X(i)}{k}| \quad \text{(by using (55))}$$

$$= \nu_X(i)|1 - \rho_X(i)| + \nu_X(i) \sum_{e \neq i, \rho_X(e) > 0} |\rho_X(e) - \rho_X(e)|$$

$$= \nu_X(i)|1 - \rho_X(i)| + \nu_X(i) \sum_{e \neq i} |\rho_X(e) - \rho_X(e)|$$

$$\leq \nu_X(i) d_{\text{inf}}$$

where the last inequality is due to Definition 2. This concludes the first part of the proof.

For the second part of the proof, we simply use that $K^2 = K$ since DPP($K$) is projective and we directly compute the desired expectation by recalling that $|\delta_e(e)| = 1 - K_{ee}$ and $|\delta_e(i)| = K_{ei}K_{ee}/K_{ee}$ for $e \neq i$. The conclusion follows from simple algebra.

B.5.2 Proof of Fact 1

We only introduce minor changes in a proof from Kaufman et al. [2022, arxiv v2]. Let $e \in [m]$ such that $\rho_X(e) > 0$. Recall the following expression

$$Z_e = \mathbb{E}_{X,e} \left[ Y_e + \sum_{i \in X,e} Y_i \right] - \mathbb{E}_X \left[ \sum_{i \in X} Y_i \right].$$

(i) First, we simply rewrite $Z_e$ as follows

$$Z_e = Y_e + \sum_{i \neq e} \rho_X(i)Y_i - \sum_i \rho_X(i)Y_i = (1 - \rho_X(e))Y_e + \sum_{i \neq e} (\rho_X(i) - \rho_X(i))Y_i.$$  

The latter expression is used to upper bound $|Z_e|$, where $|\cdot|$ is the matrix absolute value. We begin by noticing that $\lambda_{\text{max}}(|Z_e|) \leq \|Z_e\|_{\text{op}}$. Next, in light of Definition 2, we can control the following quantity as follows

$$|1 - \rho_X(e)| + \sum_{i \neq e} |\rho_X(i) - \rho_X(i)| \leq d_{\text{inf}}.$$
By using $Y_i \leq rI$ for all $i \in [m]$ and the triangular inequality, we find
\[
|Z_e| \leq \|Z_e\|_{\text{op}} I \leq |1 - \rho_X(e)| \cdot \|Y_e\|_{\text{op}} I + \sum_{i \neq e} |\rho_{\mathcal{X}_i}(i) - \rho_X(i)| \cdot \|Y_i\|_{\text{op}} I \leq d_{\text{inf}} r I.
\]

(ii) We now consider the square of $Z_e$, and we bound its expression by using the identity $AB + BA \leq A^2 + B^2$. To simplify the expressions, we recall the notation
\[
\delta_{X,e}(i) = \begin{cases} 
1 - \rho_X(e) & \text{if } i = e, \\
\rho_{X_i}(i) - \rho_X(i) & \text{otherwise.}
\end{cases}
\]
Define $s_i = \text{sign}(\delta_{X,e}(i))$. By expanding the square, we find
\[
|Z_e|^2 \leq \sum_i |\delta_{X,e}(i)|^2 Y_i^2 + \sum_i \sum_{j<i} |\delta_{X,e}(i)| \cdot |\delta_{X,e}(j)| (s_i Y_i s_j Y_j + s_j Y_j s_i Y_i)
\]
\[
\leq \sum_i |\delta_{X,e}(i)|^2 Y_i^2 + \sum_i \sum_{j<i} |\delta_{X,e}(i)| \cdot |\delta_{X,e}(j)| (Y_i^2 + Y_j^2)
\]
\[
\leq \sum_i |\delta_{X,e}(i)|^2 r Y_i + \sum_i \sum_{j<i} |\delta_{X,e}(i)| \cdot |\delta_{X,e}(j)| r Y_i + \sum_i \sum_{j<i} |\delta_{X,e}(i)| \cdot |\delta_{X,e}(j)| r Y_j
\]
\[
= \sum_i |\delta_{X,e}(i)|^2 r Y_i + \sum_j \sum_i |\delta_{X,e}(i)| \cdot |\delta_{X,e}(j)| r Y_i + \sum_i \sum_{j<i} |\delta_{X,e}(i)| \cdot |\delta_{X,e}(j)| r Y_i
\]
\[
\leq d_{\text{inf}} r \sum_i |\delta_{X,e}(i)| Y_i,
\]
where the indices were renamed in the next-to-last equality, and where we used $Y_i \leq r I$. Now, recall that $\mathbb{E}_{\nu|e} |\delta_{X,e}(i)| \leq d_{\text{inf}} \nu_X(i)$ for all $i$ in the support of $\rho_X$ in light of Lemma 1. Finally, we compute the following expectation of the last bound,
\[
\mathbb{E}_{\nu|e} [|Z_e|^2] \leq r d_{\text{inf}} \sum_i \mathbb{E}_{\nu|e} |\delta_{X,e}(i)| Y_i \leq r d_{\text{inf}}^2 \sum_i \nu_X(i) Y_i = r d_{\text{inf}}^2 \mathbb{E}_{\nu|e} [Y_e].
\]
This is the desired result.

C Additional numerical simulations

C.1 Empirical estimation of leverage scores

In Figure 11, we provide an empirical validation of the MTSF sampling algorithm given in Section 6. We empirically compute the probability that any edge $e$ belongs to a random MTSF $\mathcal{F} \sim \text{DPP}(K)$, i.e., the leverage score $\Pr(e \in \mathcal{F}) = K_{e e} \text{ with } K$ given in (14); see Section 4. For a fixed MUN$(n,p,\eta)$ graph, we compute an empirical counterpart by calculating the frequencies $\frac{1}{t} \sum_{t=1}^T 1_{\mathcal{F}_t}(e)$ where the MTSFs $\{\mathcal{F}_t\}_{1 \leq t \leq T}$ are i.i.d. DPP$(K)$. Figure 11 provides a comparison of the exact and empirical leverage scores for a fixed enumeration of the graph edges both for CRSFs ($q = 0$) and MTSFs ($q > 0$). The empirical leverage scores are computed over $5 \cdot 10^4$ runs for a fixed MUN random graph for $q = 0$ (left-hand side) and $q = 1$ (right-hand side). The exact leverage scores are the diagonal elements of the correlation kernel (14). We observe that the CRSF and MTSF sampling algorithms yield samples with empirical edge marginal probabilities close to the exact edge marginal probabilities.

C.2 Consistency of cycles for the random graphs of Section 8.2.2

In Figure 12, we display the average cycle importance weights $1 \lor (1 - \cos \theta(\eta))$ of the cycles $\eta$ sampled in the CRSFs for the example of Figure 3. These weights are used in the self-normalized importance sampling estimate; see (23). A weakly inconsistent cycle has a unit weight, whereas the largest possible weight is two.
Clearly, the inconsistent cycles in the MUN model have smaller weights compared with the case of the ERO model. Indeed, we see in Figure 12a that the average weight is close to 1, and it is larger than one in Figure 12b.

C.3 Magnetic leverage scores for the ERO graph of Figure 4b

In Figure 13, we display the magnetic leverage score for each edge of the ERO graph of Figure 4b. On the same figure, we plot the error of the edge angle coming from the noise, namely, $\theta(uv) - (h_u - h_v)/(\pi(n - 1))$ for each edge $uv$. We observe that when the magnetic LS is large, the corresponding edge noise is often also large. Hence, “i.i.d. LS” will tend to sample edges associated with angles contaminated with noise. This explains why “i.i.d. LS” in Figure 4b tends to outperform “unif. LS”, since the latter uses no information about edge angles.
Figure 13: Magnetic LS and edge error vs. edge index for the ERO($n,p,\eta$) graph of Figure 4b. Notice that edge indexing is arbitrary.

References

L. A. Adamic and N. Glance. The Political Blogosphere and the 2004 US Election. In Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem, 2005. URL https://dl.acm.org/doi/10.1145/1134271.1134277.

R. Ahlswede and A. Winter. Strong Converse for Identification via Quantum Channels. IEEE Transactions on Information Theory, 48(3):569–579, 2002. URL https://ieeexplore.ieee.org/document/985947.

D. Aldous and J. A. Fill. Reversible Markov Chains and Random Walks on Graphs, 2002. Unfinished monograph, recompiled 2014, available at http://www.stat.berkeley.edu/~aldous/RWG/book.html.

D. J. Aldous. The Random Walk Construction of Uniform Spanning Trees and Uniform Labelled Trees. SIAM Journal on Discrete Mathematics, 3(4):450–465, 1990. URL https://doi.org/10.1137/0403039.

Z. Allen-Zhu, Z. Liao, and L. Orecchia. Spectral Sparsification and Regret Minimization Beyond Matrix Multiplicative Updates. In Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, STOC ’15, pages 237–245, New York, NY, USA, 2015. Association for Computing Machinery. URL https://doi.org/10.1145/2746539.2746610.

M. E. Argentati, A. V. Knyazev, K. Neymeyr, E. E. Ovtchinnikov, and M. Zhou. Convergence Theory for Preconditioned Eigenvalue Solvers in a Nutshell. Foundations of Computational Mathematics, 17(3):713–727, 2017. URL https://doi.org/10.1007/s10208-015-9297-1.

L. Avena and A. Gaudilli`ere. Two Applications of Random Spanning Forests. Journal of Theoretical Probability, 31:1975–2004, 2018. URL https://link.springer.com/article/10.1007/s10959-017-0771-3#citeas.

A. S. Bandeira, A. Singer, and D. A. Spielman. A Cheeger Inequality for the Graph Connection Laplacian. SIAM Journal on Matrix Analysis and Applications, 34(4):1611–1630, 2013. URL https://doi.org/10.1137/120875338.

S. Barthelmé, N. Tremblay, A. Gaudilliere, L. Avena, and P.-O. Amblard. Estimating the Inverse Trace Using Random Forests on Graphs. In XVIIème colloque GRETSI (GRETSI 2019), 2020. URL https://hal.archives-ouvertes.fr/hal-02319194.

J. Batson, D. A. Spielman, N. Srivastava, and S.-H. Teng. Spectral Sparsification of Graphs: Theory and Algorithms. Commun. ACM, 56(8):87–94, 2013. URL https://doi.org/10.1145/2492007.2492029.

J. Batson, D. A. Spielman, and N. Srivastava. Twice-Ramanujan Sparsifiers. SIAM Review, 56(2):315–334, 2014. URL http://www.jstor.org/stable/24248502.
G. Berkolaiko. Nodal Count of Graph Eigenfunctions via Magnetic Perturbation. *Analysis & PDE*, 6: 1213–1233, 2013. URL https://msp.org/apde/2013/6-5/p08.xhtml.

A. Broder. Generating Random Spanning Trees. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, SFCS ’89, pages 442–447, USA, 1989. IEEE Computer Society. URL https://doi.org/10.1109/SFCS.1989.63516.

T. Chu, Y. Gao, R. Peng, S. Sachdeva, S. Sawlani, and J. Wang. Graph Sparsification, Spectral Sketches, and Faster Resistance Computation via Short Cycle Decompositions. *SIAM Journal on Computing*, 0(0): FOCS18–85–FOCS18–157, 2018. URL https://doi.org/10.1137/19M1247632.

Y. Colin de Verdière. *Spectres de Graphes*, volume 4. Paris: Société Mathématique de France, 1998. URL http://www-fourier.univ-grenoble-alpes.fr/~ycolver/All-Articles/98a.pdf.

Y. Colin de Verdière, N. Torki-Hamza, and F. Truc. Essential Self-adjointness for Combinatorial Schrödinger Operators III- Magnetic Fields. *Annales de la Faculté des sciences de Toulouse : Mathématiques*, Ser. 6, 20(3):599–611, 2011. URL https://afst.centre-mersenne.org/articles/10.5802/afst.1319/.

M. Cucuringu. Sync-Rank: Robust Ranking, Constrained Ranking and Rank Aggregation via Eigenvector and SDP Synchronization. *IEEE Transactions on Network Science and Engineering*, 3(1):58–79, 2016. URL https://doi.org/10.1109/TNSE.2016.2523761.

M. Cucuringu, Y. Lipman, and A. Singer. Sensor Network Localization by Eigenvector Synchronization over the Euclidean Group. *ACM Transactions on Sensor Networks*, 8(3), 2012a. URL https://doi.org/10.1145/2240092.2240093.

M. Cucuringu, A. Singer, and D. Cowburn. Eigenvector Synchronization, Graph Rigidity and the Molecule Problem. *Information and Inference: A Journal of the IMA*, 1(1):21–67, 12 2012b. URL https://doi.org/10.1093/imaiai/ias002.

C. Davis and W. M. Kahan. The Rotation of Eigenvectors by a Perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970. URL https://doi.org/10.1137/0707001.

D. Durfee, R. Kyng, J. Peebles, A. B. Rao, and S. Sachdeva. Sampling Random Spanning Trees Faster than Matrix Multiplication. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 730–742, New York, NY, USA, 2017. Association for Computing Machinery. URL https://doi.org/10.1145/3055399.3055499.

M. Fanuel, C. M. Alaiz, A. Fernandez, and J. A. K. Suykens. Magnetic Eigenmaps for the Visualization of Directed Networks. *Applied and Computational Harmonic Analysis*, 44:189–199, 2018. URL https://www.sciencedirect.com/science/article/abs/pii/S1063520317300052.

R. Forman. Determinants of Laplacians on Graphs. *Topology*, 32(1):35–46, 1993. URL https://www.sciencedirect.com/science/article/pii/004093839390035T.

W. S. Fung, R. Hariharan, N. J. Harvey, and D. Panigrahi. A General Framework for Graph Sparsification. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC ’11, pages 71–80, New York, NY, USA, 2011. Association for Computing Machinery. URL https://doi.org/10.1145/1993636.1993647.

J. Geweke. Bayesian Inference in Econometric Models Using Monte Carlo Integration. *Econometrica*, 57(6): 1317–1339, 1989. URL http://www.jstor.org/stable/1913710.

H. Guo and K. He. Tight Bounds for Popping Algorithms. *Random Structures & Algorithms*, 57(2):371–392, 2020. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rsa.20928.

H. Guo and M. Jerrum. Approximately Counting Bases of Bicircular Matroids. *Combinatorics, Probability and Computing*, 30(1):124–135, 2021. URL https://doi.org/10.1017/S0963548320000292.
A. Madry, D. Straszak, and J. Tarnawski. Fast Generation of Random Spanning Trees and the Effective Resistance Metric. In Proceedings of the 2015 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 2019–2036, 2015. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611973730.134.

P.-G. Martinsson and J. A. Tropp. Randomized Numerical Linear Algebra: Foundations and Algorithms. Acta Numerica, 29:403–572, 2020. URL https://doi.org/10.1017/S0962492920000021.

R. Morgan. Preconditioning Eigenvalues and Some Comparison of Solvers. Journal of Computational and Applied Mathematics, 123(1):101–115, 2000. URL https://www.sciencedirect.com/science/article/pii/S0377042700003952.

R. B. Morgan and D. S. Scott. Preconditioning the Lanczos Algorithm for Sparse Symmetric Eigenvalue Problems. SIAM Journal on Scientific Computing, 14(3):585–593, 1993. URL https://doi.org/10.1137/0914037.

R. Oliveira. Concentration of the Adjacency Matrix and of the Laplacian in Random Graphs with Independent Edges. 2009. URL https://arxiv.org/pdf/0911.0600.pdf.

R. Oliveira. Sums of Random Hermitian Matrices and an Inequality by Rudelson. Electronic Communications in Probability, 15:203–212, 2010. URL https://doi.org/10.1214/ECP.v15-1544.

R. Pemantle. Choosing a Spanning Tree for the Integer Lattice Uniformly. The Annals of Probability, 19(4):1559–1574, 1991. URL http://www.jstor.org/stable/2244527.

R. Pemantle and Y. Peres. Concentration of Lipschitz Functionals of Determinantal and Other Strong Rayleigh Measures. Combinatorics, Probability and Computing, 23(1):140–160, 2014. URL https://www2.math.upenn.edu/~pemantle/papers/concentration.pdf.

Y. Y. Pilavci, P.-O. Amblard, S. Barthelmé, and N. Tremblay. Smoothing Graph Signals via Random Spanning Forests. In ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5630–5634, 2020. URL https://ieeexplore.ieee.org/document/9054497.

Y. Y. Pilavci, P.-O. Amblard, S. Barthelmé, and N. Tremblay. Graph tikhonov regularization and interpolation via random spanning forests. IEEE Transactions on Signal and Information Processing over Networks, 7:359–374, 2021. URL https://doi.org/10.1109/TSIPN.2021.3084879.

A. Poncelet. Schramm’s Formula for Multiple Loop-erased Random Walks. Journal of Statistical Mechanics: Theory and Experiment, 2018(10):103106, 2018. URL https://doi.org/10.1088/1742-5468/aae5a6.

A. Rudi, L. Carratino, and L. Rosasco. Falkon: An Optimal Large Scale Kernel Method. In Advances in Neural Information Processing Systems 30, pages 3888–3898, 2017. URL https://proceedings.neurips.cc/paper/2017/file/05546b0e38ab9175cd905eebccc6eb76-Paper.pdf.

A. Schild. An Almost-Linear Time Algorithm for Uniform Random Spanning Tree Generation. In Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, pages 214–227, New York, NY, USA, 2018. Association for Computing Machinery. URL https://doi.org/10.1145/3188745.3188852.

T. Shirai and Y. Takahashi. Random Point Fields Associated with Certain Fredholm Determinants I: Fermion, Poisson and Boson Point Processes. Journal of Functional Analysis, 205(2):414–463, 2003. URL https://www.sciencedirect.com/science/article/pii/S002212360300017X.

A. Singer. Angular Synchronization by Eigenvectors and Semidefinite Programming. Applied and Computational Harmonic Analysis, 30(1):20–36, 2011. URL https://www.sciencedirect.com/science/article/pii/S1063520310000205.

A. Soshnikov. Determinantal random point fields. Russian Mathematical Surveys, 55(5):923–975, 2000. URL https://doi.org/10.1070/rm2000v055n05abeh000321.
D. A. Spielman and N. Srivastava. Graph Sparsification by Effective Resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011. URL https://doi.org/10.1137/080734029.

D. A. Spielman and S.-H. Teng. Nearly Linear Time Algorithms for Preconditioning and Solving Symmetric, Diagonally Dominant Linear Systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014. URL https://doi.org/10.1137/090771430.

J. A. Tropp. An Introduction to Matrix Concentration Inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015. URL http://dx.doi.org/10.1561/2200000048.

N. K. Vishnoi. Lx = b. *Foundations and Trends® in Theoretical Computer Science*, 8(1–2):1–141, 2013. URL http://dx.doi.org/10.1561/0400000054.

D. B. Wilson. Generating Random Spanning Trees More Quickly than the Cover Time. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC ’96, pages 296–303, New York, NY, USA, 1996. Association for Computing Machinery. URL https://doi.org/10.1145/237814.237880.

S. Yu. Angular Embedding: A Robust Quadratic Criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):158–173, 2012. URL https://doi.org/10.1109/TPAMI.2011.107.

S. X. Yu. Angular Embedding: From Jarring Intensity Differences to Perceived Luminance. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2302–2309, 2009. URL https://doi.org/10.1109/CVPR.2009.5206673.

Y. Yu, T. Wang, and R. J. Samworth. A Useful Variant of the Davis—Kahan Theorem for Statisticians. *Biometrika*, 102(2):315–323, 2015. URL http://www.jstor.org/stable/43908537.

W. Zhao, F. Chung, and M. Kempton. Ranking and Sparsifying a Connection Graph. *Internet Mathematics*, 10(1-2):87–115, 2014. URL https://doi.org/10.1080/15427951.2013.800005.

Y. Zhong and N. Boumal. Near-Optimal Bounds for Phase Synchronization. *SIAM Journal on Optimization*, 28(2):989–1016, 2018. URL https://doi.org/10.1137/17M1122025.

D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with Local and Global Consistency. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2004. URL https://proceedings.neurips.cc/paper/2003/file/87682805257e619d49b8e0dfdc14affa-Paper.pdf.

A. Zouzias. A Matrix Hyperbolic Cosine Algorithm and Applications. In A. Czumaj, K. Mehlhorn, A. Pitts, and R. Wattenhofer, editors, *Automata, Languages, and Programming*, pages 846–858, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. URL http://www.cs.toronto.edu/~zouzias/downloads/papers/hypercosine_zouzias_new.pdf.