

Smoothing complex-valued signals on Graphs with Monte-Carlo

Hugo Jaquard, Michaël Fanuel, Pierre-Olivier Amblard, Rémi Bardenet, Simon Barthelme, Nicolas Tremblay

▶ **To cite this version:**

Hugo Jaquard, Michaël Fanuel, Pierre-Olivier Amblard, Rémi Bardenet, Simon Barthelme, et al.. Smoothing complex-valued signals on Graphs with Monte-Carlo. ICASSP 2023 - IEEE International Conference on Acoustics, Speech and Signal Processing, Jun 2023, Rhodes Island, Greece. 10.1109/ICASSP49357.2023.10096354 . hal-03869873

HAL Id: hal-03869873
https://hal.science/hal-03869873v1

Submitted on 24 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

arXiv:2210.08014v6 [stat.ME] 28 Feb 2023

SMOOTHING COMPLEX-VALUED SIGNALS ON GRAPHS WITH MONTE-CARLO

Hugo Jaquard [†⋆] *Michaël Fanuel* [‡⋆]
Pierre-Olivier Amblard [†] *Rémi Bardenet* [‡] *Simon Barthelmé* [†] *Nicolas Tremblay* [†]

[†] GIPSA-lab, CNRS, Univ. Grenoble Alpes, Grenoble INP
[‡] Univ. Lille, CNRS, Centrale Lille, UMR 9189 – CRIStAL
[⋆] equally contributing authors

ABSTRACT

We introduce new smoothing estimators for complex signals on graphs, based on a recently studied *Determinantal Point Process* (DPP). These estimators are built from subsets of edges and nodes drawn according to this DPP, making up trees and *unicycles*, *i.e.*, connected components containing exactly one cycle. We provide a Julia implementation of these estimators and study their performance when applied to a ranking problem.

***Index Terms*—** magnetic laplacian, spanning forests, determinantal point processes, graph smoothing, angular synchronization, ranking

1. INTRODUCTION

Graph signal processing (GSP, [1]) usually considers real data defined over the nodes of a graph $G = (V, E)$, and classically relies on the graph Laplacian. For instance, a typical task in GSP consists in smoothing (denoising) a signal $g \in \mathbb{R}^V$ by solving the penalized (Tikhonov) problem

$$\underset{f\in\mathbb{R}^V}{\operatorname{argmin}}\ q\|f-g\|^2 + f^\top L f \tag{1}$$

where $L \in \mathbb{R}^{V\times V}$ is the graph Laplacian. We will consider throughout weighted and undirected graphs. In Equation (1), $f^\top L f$ penalises the squared-norm of the discrete derivative on the graph, *i.e.*:

$$f^\top L f = \sum_{e=(v,v')} w_e (f(v') - f(v))^2,$$

where the sum runs over all edges of the graph and $w_e > 0$ is the edge weight. Thus, the quadratic form $f^\top L f$ can be thought of as i/ computing the squared difference between signal values along neighbouring edges, ii/ computing a weighted sum.

For multivariate signals, defining a discrete derivative becomes less obvious, as additional geometry enters the picture. For instance, taking $f(v') - f(v)$ along an edge $e = (v, v')$ assumes that $f(v')$ and $f(v)$ use the same coordinate system, and introducing a local change of basis along the edge (v, v') is a way to precise the relation between $f(v)$ and $f(v')$ [2]. We focus on complex-valued signals $f \in \mathbb{C}^V$, for which a multiplication by $e^{i\theta_e}$, where $\theta_e \in [0, 2\pi]$, can for instance represent a known phase offset between measurements $f(v')$ and $f(v)$. In this setting, we can define a *magnetic Laplacian* $L \in \mathbb{C}^{V\times V}$ [3] (see Appendix A) which acts as follows (f^* denotes the conjugate transpose of f):

$$f^* L f = \sum_{e=(v,v')} w_e |f(v') - e^{i\theta_e} f(v)|^2. \tag{2}$$

Note that this equation supposes an orientation of each edge: each θ_e is thus given with an orientation of e. This choice of orientation is however arbitrary: for a given edge e, one orientation associated to θ_e is equivalent to the other orientation associated to $-\theta_e$ as $|f(v') - e^{i\theta_e} f(v)|^2 = |e^{-i\theta_e} f(v') - f(v)|^2$. The set $\{e^{i\theta_e}\}_e$ describes a *unitary connection* between the nodes of G [3].

Such Laplacians have applications to synchronization [4] and ranking problems [5–7], as we explain in Section 4.

GSP algorithms that use the graph Laplacian often have $\mathcal{O}(|V|^3)$ scaling when implemented exactly, due to the matrix inversions or factorisations that are used (*e.g.*, the solution of Equation (1) reads $f_o = q(L + qI)^{-1} g$). In large graphs approximate methods are necessary, and [8] introduced a Monte-Carlo estimator for the Tikhonov smoothing problem of Equation (1), with favourable asymptotic runtime. In this work we generalise the estimator of [8] to complex signals, using a process recently introduced in [9].

In Section 2, we introduce a slight variation of the random process of [9], defined over graphs with a unitary connection. Our main theoretical results are given in Section 3, where we derive estimators for the solution of the Tikhonov smoothing

This work was supported by the ANR grant GRANOLA (ANR-21-CE48-0009), the ERC grant BLACKJACK (ERC-2019-STG-851866), the ANR AI chair BACCARAT (ANR-20-CHIA-0002), as well as the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01), the Grenoble Data Institute (ANR-15-IDEX- 02), MIAI@Grenoble Alpes (ANR-19-P3IA-0003), the LIA CNRS/Melbourne Univ Geodesic, and the IRS (Initiatives de Recherche Stratégiques) of the IDEX Université Grenoble Alpes.

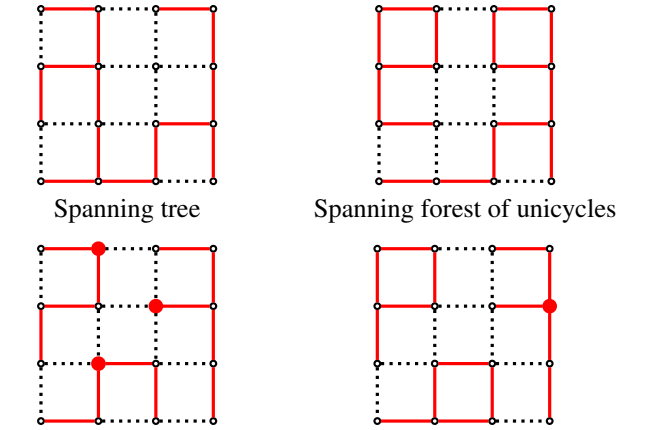


Fig. 1: Different subsets of edges and nodes, in red.

problem. We describe a practical application to ranking in Section 4. Some technical definitions and proofs are deferred to Appendices A, B and C.

2. A PROCESS OVER MULTI-TYPE SPANNING FORESTS

We are interested in a distribution generalizing both the uniform distribution over *spanning trees* (UST) of a graph, and the *random spanning forests* distribution [10]. A spanning tree is a subset of edges $\phi \subseteq E$ such that the graph with nodes V and edges ϕ is both connected and without cycles (cycle-free). A rooted spanning forest (RSF) $\phi \subseteq E \cup V$ is the combination of a spanning forest and a set of distinguished nodes called the roots (one root per tree). A variation of these distributions can be defined over the set of spanning forests of *unicycles* (FU) [3, 11], subsets of edges containing *exactly one* cycle per connected component, and spanning all the nodes in V. The generalization we consider instead draws its samples from the set of rooted *multi-type spanning forests* (MTSF). A rooted MTSF is a spanning subset of edges and nodes $\phi \subseteq E \cup V$ whose connected components are made up of rooted trees and *unicycles*. These different structures are illustrated in Fig. 1.

We will use the following distribution, introduced in [9], over rooted MTSFs $\phi = \phi_\bullet \cup \rho(\phi)$ of G:

$$\mathbb{P}(\phi) \propto \prod_{r \in \rho(\phi)} q_r \prod_{e \in \phi_\bullet} w_e \prod_C (2 - 2\cos(\theta_C)), \qquad (3)$$

with $\rho(\phi)$ the set of roots of ϕ, $\phi_\bullet$ its set of edges and $q_v \in \mathbb{R}_+^*$ positive parameters associated to each node v. The third product is over the cycles of the unicycles in ϕ, where $\theta_C = \sum_{e \in C} \theta_e$.

One may wonder why we do not consider the *uniform* distribution over MTSFs and instead favor sampling unicycles for which $2 - 2\cos(\theta_C)$ is large, meaning that these cycles are *inconsistent*. Part of the answer lies in the fact that Eq. 3 is known to describe a *Determinantal Point Process* (DPP) over $E \cup V$, a useful property for proving the results in Section 3. More details are available in Appendix B.1.

Aforementioned distributions such as USTs and RSFs are conveniently sampled using (a variant of) Wilson's algorithm [12], based on random walks on the graph. These sampling procedures have been generalized to FUs [13] and, recently, to MTSFs [9], under the sampling condition:

$$\cos(\theta_\gamma) \geq 0 \text{ for all cycles } \gamma. \qquad (4)$$

Note that this condition applies to *all cycles* γ in G, and not only in some MTSF ϕ. The sampling algorithm is recalled in Appendix B.2.

Bounds on the running time of the sampling algorithm are discussed in [9, 13]. Here, we only mention that the expected running time is linear in the number of edges, and that it decreases as $\min_v q_v$ increases.

3. ESTIMATORS FOR SMOOTHING

Monte-Carlo estimators for smoothing on graphs have been developped using RSFs [8]. The main idea is to sample a rooted RSF ϕ before propagating the value of the function at the roots to the other nodes in their associated trees. In the following, we show how this can be generalized to graphs with a unitary connections using the distribution in Equation (3). Proofs are given in Appendix C. Specifically, given $g \in \mathbb{C}^V$ we derive estimators of $f_o = (L + Q)^{-1} Q g$. Here Q is the diagonal matrix of the q_v's and $L = D - A_\theta$ is the Hermitian *magnetic Laplacian* matrix, with D the diagonal degree matrix and $(A_\theta)_{v',v} = e^{i\theta_e}$ if $e = (v, v') \in E$ (0 if $e \notin E$). When $q_v = q > 0$ for all $v \in V$, f_o is the optimal solution of the Tikhonov problem:

$$\operatorname*{argmin}_{f \in \mathbb{C}^V} q\|f - g\|^2 + f^* L f.$$

The first estimator $\tilde{f}$ is built by propagating values through the transport maps $z \mapsto e^{i\theta_e} z$ on a rooted multi-type spanning forest ϕ sampled according to Equation (3):

$$\tilde{f}(v) = \begin{cases} \psi_{r_\phi(v) \to v} g(r_\phi(v)) & \text{if the connected component of } v \text{ is a rooted tree,} \\ 0 & \text{otherwise,} \end{cases}$$

where $r_\phi : V \to V$ maps nodes to the root of the tree containing them, $a \to b$ denotes the unique path from a to b in ϕ and $\psi_{a \to b} = \prod_{e \in a \to b} e^{i\theta_e}$.

Proposition 1. *$\tilde{f}$ is an unbiased estimator of f_o:*

$$\mathbb{E}_\phi(\tilde{f}) = f_o.$$

As a consequence of the Central Limit Theorem, Monte-Carlo estimators converge at a $\mathcal{O}(\frac{\sigma}{\sqrt{n}})$ rate, with σ the standard deviation of the estimator and n the number of samples. Instead of increasing the number of samples one may instead focus on decreasing the variance, and in our setting this can be done at little additional cost. A first approach is to use a Rao-Blackwell version of the estimator $\tilde{f}$ (see [14, 15]) by conditioning on the set of *unrooted* connected components $\pi \subseteq E$ of the MTSF. Consider the estimator:

$$\overline{f}(v) = \begin{cases} \psi_{r_\phi(v)\to v} h(r_\phi(v)) & \text{if the connected component} \\ & \text{of } v \text{ is a rooted tree,} \\ 0 & \text{otherwise,} \end{cases}$$

where $h(u) = \frac{\sum_{w\in T_u} q_w \psi_{w\to u} g(w)}{\sum_{w\in T_u} q_w}$, with T_v the tree containing v in ϕ. Variance reduction is achieved solely from computing a mean over the nodes of the rooted trees.

Proposition 2. *We have $\forall v \in V, \overline{f}(v) = \mathbb{E}_\phi(\tilde{f}(v)|\phi_\bullet = \pi)$ and, as a consequence: $\mathbb{E}_\pi(\overline{f}) = \mathbb{E}_\phi(\tilde{f}) = f_o$. Also, by the law of total variance, $\overline{f}(v)$ has a lower variance than $\tilde{f}(v)$:*

$$\mathrm{Var}(\overline{f}(v)) = \mathrm{Var}(\tilde{f}(v)) - \mathbb{E}(\mathrm{Var}(\tilde{f}(v)|\phi_\bullet = \pi)) \leq \mathrm{Var}(\tilde{f}(v))$$

The method of control variates is another classical variance-reduction technique for Monte-Carlo estimators, adding a term with zero mean to obtain a modified estimator with the same expectation but lower variance [16]. The following is proved in [17], when $q_v = q$ for all $v \in V$ (*i.e.* $Q = qI$.)

Proposition 3. *Set $\alpha = \frac{2q}{q+2d_m}$, where d_m is the maximum degree in the graph. Then, the estimator*

$$\hat{f} = \overline{f} - \alpha(q^{-1}(L + qI)\overline{f} - g)$$

is an unbiased estimator of f_o and verifies

$$\forall v \in V \qquad \mathrm{Var}(\hat{f}(v)) \leq \mathrm{Var}(\overline{f}(v))$$

For a graph with heterogeneous degree distribution, with *e.g.* a maximum degree much larger than the mean degree, $\hat{f}$ is only a marginal inprovement over $\overline{f}$. However, on graphs with a nearly-homogeneous degree distribution, which is the case for the graphs considered in Section 4, we obtain substantial improvements over $\overline{f}$.

4. RANKING FROM CORRUPTED MEASUREMENTS

In order to study the actual performance of our estimators, we describe an application to the *ranking* problem. We focus on the ordinal ranking problem, which asks to linearly order a set X of n elements according to an incomplete, possibly incoherent, set of pairwise ordinal comparisons $C_{i,j} \in \{-1, 1\}$, for $j > i$. From this data, build a graph G with n nodes, and a directed edge from i to j if $C_{i,j} = 1$ (resp. from j to i if $C_{i,j} = -1$). Ranking according to [6, 7] then suggests to define the unitary connection $\theta_{i,j} = \frac{\pi\delta C_{i,j}}{n}$ with $\delta \in (0, 1)$, and to perform *angular synchronization* [4, 18] by solving:

$$\underset{(\omega_v)_{v\in V}}{\mathrm{argmin}} \sum_{e=(v,v')} |e^{i\omega_{v'}} - e^{i\theta_e} e^{i\omega_v}|^2. \tag{5}$$

The optimal arguments ω_i then describe an embedding of the n points onto the unit circle, from which we can extract a ranking (see [7] for more details).

In practice, solving such a non-convex optimization problem is difficult, and a spectral relaxation is considered instead:

$$\min_{f\in\mathbb{C}^n, \|f\|^2=n} f^* \tilde{L} f,$$

for $\tilde{L} = D^{-1/2} L D^{-1/2}$ the normalized graph Laplacian, with D the diagonal degree matrix. The solution of this classical problem is the eigenvector associated to the smallest eigenvalue of $\tilde{L}$, which can for instance be computed by iterating the map $x \mapsto Mx/\|Mx\|$ for $M = q(\tilde{L} + qI)^{-1}$, this is the power method [19]. Computing $x \mapsto Mx$ can either be performed directly by solving a linear system, or using the estimators $\tilde{f}$, $\overline{f}$ and $\hat{f}$.

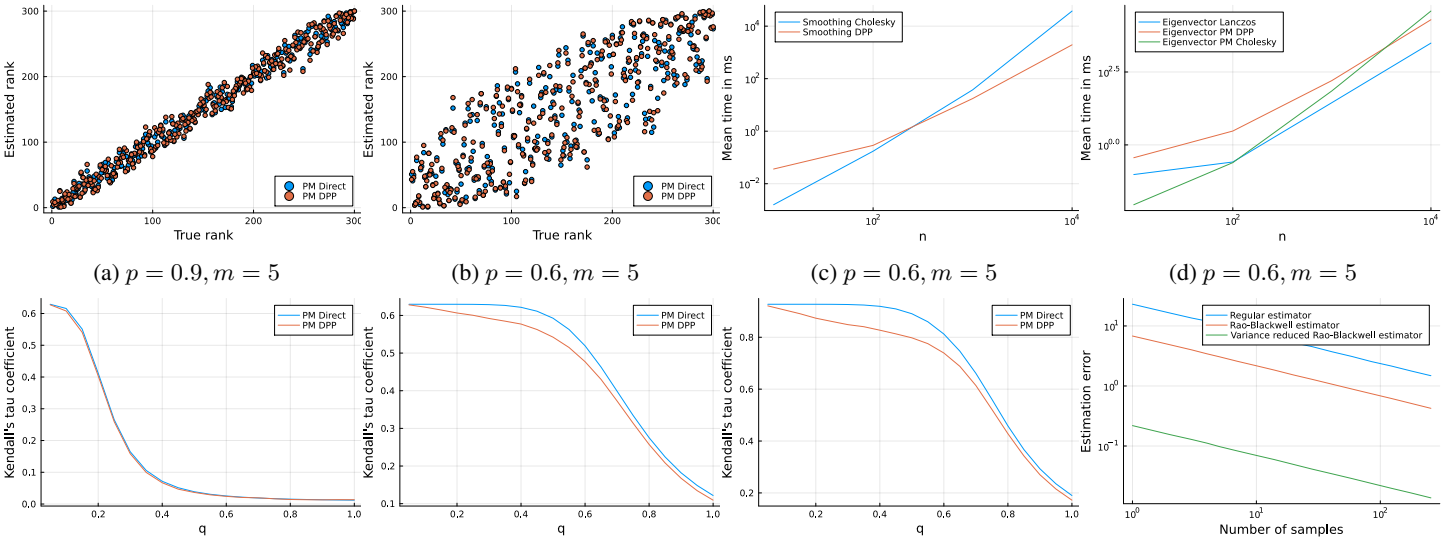
If we set $\delta = \frac{1}{4}$, the sampling condition in Equation (4) is satisfied and fast sampling can be achieved using the variation of Wilson's algorithm recalled in Appendix B.2.

4.1. Experimental results

We illustrate in Fig. 2 some results regarding the performance of this approach obtained with our Julia implementation[1]. We work with comparisons $C_{i,j}$ randomly obtained from a ground-truth ranking r_{GT} according to the Erdös-Rényi Outliers model [7]: a comparison is observed with probability s; if a comparison is observed, it follows the ranking r_{GT} with probability p, or is chosen uniformly in $\{-1, 1\}$ otherwise. We focus on the computationally challenging dense case and set $s = 0.8$. Unless otherwise specified, we use $q = 0.1$. . The performance of the estimator $\hat{f}$ in recovering the underlying ranking, using $m = 5$ MTSFs *for each* of the $k = 10$ iterations of the power method, is illustrated in Figs. 2a and 2b for two graphs of size $n = 300$ and $p \in \{0.6, 0.9\}$. The eigenvector of L computed from the power method *without* using the estimator is also plotted. Here, the initialization vector y_0 used for the power method is a random embedding of the n points in the unit circle, spaced out with angle $\frac{\pi}{2n}$.

Runtime benchmarks are available in Figs. 2c and 2d for the smoothing problem (computation of My_0) solved either using $\hat{f}$ or using a Cholesky decomposition, and for the computation of the eigenvector of M for the power method iteration using $\hat{f}$ or a Cholesky decomposition, and for the Lanczos method. We display the mean running time over 100 measurements on fixed graphs of size $n \in \{10, 10^2, 10^3, 10^4\}$.

[1] https://gricad-gitlab.univ-grenoble-alpes.fr/tremblan/mtsf_for_graph_smoothing



(a) $p = 0.9, m = 5$ (b) $p = 0.6, m = 5$ (c) $p = 0.6, m = 5$ (d) $p = 0.6, m = 5$

(e) $k = 5, p = 0.6, m = 5$ (f) $k = 10, p = 0.6, m = 5$ (g) $k = 10, p = 0.9, m = 5$ (h) $p = 0.6, m = 5$

Fig. 2: Experimental results.

We plot Kendall's τ coefficients [20] (the larger the better) for the rankings recovered from the power method in Figs. 2e, 2f and 2g for $n = 3000$ and varying q, averaging over 20 realisations.

Reconstruction errors $\|My - \mathrm{e}(f_o)\|$ across varying m, averaged over 20 runs for each of the estimators $\mathrm{e}(f_o)$ of f_o, are in Fig. 2h with $n = 3000$.

4.2. Discussion

The results in Figs. 2a and 2b compare the performance of the reconstructions obtained by the power method with and without using $\hat{f}$, on only one realization of the graph. For these noise regimes, the ground truth ranking τ_{GT} can no longer be recovered (which would result in a straight diagonal line). In both cases, the recovered ranking is slightly more spread out when using the estimator instead of the exact power method.

We obtain faster runtime in the smoothing problem than a standard Cholesky decomposition, in graphs with more than $n \sim 10^3$ nodes (Fig. 2c). For the eigenvector computation in Fig. 2d, cross-over occurs at $n \sim 10^4$ against the power method, while we do not outperform a direct Lanczos iteration. Note that performance for the eigenvector computation may be increased by sampling only one set of $m = 5$ MTSFs, used for all iterations of the power method. Moreover, our current implementation is far from optimal.

As q decreases, the τ coefficient increases until it reaches a fixed value in Figs. 2e, 2f and 2g. Comparing of Figs. 2e and 2f shows that this behavior actually reflects the convergence of the power method, as the spectral gap of M is lower for larger q, requiring more iterations to converge. This suggests that there is a trade-off for the best choice of q: it should be as large as possible in order to reduce sampling time, and not too large so that it allows fast convergence of the power method. Note also that Kendall's τ seems to reach the value of p when convergence occurs.

Regarding the convergence of the estimators, all three versions of the estimators in Fig. 2h have linear decay in log-log-scale, which is characteristic of the $\mathcal{O}(1/\sqrt{n})$ convergence rate of Monte-Carlo estimators. In our simulations, the estimator $\hat{f}$ improves on the regular estimator by a factor of 10.

Throughout these experiments, we sampled $m = 5$ MTSFs to compute the estimators. Here, this choice was arbitrary, but we notice nonetheless that this is of the order of the $\mathcal{O}(\log |V|)$ uniform spanning trees necessary to obtain spectral sparsifiers for connection-free graphs [21], recently adapted to MTSFs in [9].

5. CONCLUSION

We define new estimators built by propagating values along edges sampled according to a recently introduced DPP, for the smoothing problem on graphs endowed with a unitary connection, thus generalizing previous existing approaches on graphs. The evaluation of these estimators on the ranking problem, using our Julia implementation, show that they can be advantageous for smoothing starting from moderately sized graphs, as compared to a Cholesky decomposition, while they do not improve on a Lanczos iteration for eigenvector computation. Nonetheless, the proposed estimators exhibit a potentially useful computational property that is uncommon among deterministic algorithms: their computation can be carried out without any prior global knowledge of the graph, from local neighbor queries only, which in some sce-

narii may be the only practical interaction. The choice of the parameters q and m is application-dependent, and requires further investigation. Possible extensions of this work include designing new applications of the proposed estimators, as well as generalizing this approach to higher-dimensional signals, where current arguments do not carry over.

6. REFERENCES

[1] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[2] Amit Singer and H-T Wu, "Vector diffusion maps and the connection laplacian," *Communications on pure and applied mathematics*, vol. 65, no. 8, pp. 1067–1144, 2012.

[3] Richard Kenyon, "Spanning forests and the vector bundle Laplacian," *The Annals of Probability*, vol. 39, no. 5, pp. 1983–2017, 2011.

[4] Amit Singer, "Angular synchronization by eigenvectors and semidefinite programming," *Applied and computational harmonic analysis*, vol. 30, no. 1, pp. 20–36, 2011.

[5] X Yu Stella, "Angular embedding: from jarring intensity differences to perceived luminance," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2302–2309.

[6] Stella Yu, "Angular embedding: A robust quadratic criterion," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 1, pp. 158–173, 2011.

[7] Mihai Cucuringu, "Sync-rank: Robust ranking, constrained ranking and rank aggregation via eigenvector and sdp synchronization," *IEEE Transactions on Network Science and Engineering*, vol. 3, no. 1, pp. 58–79, 2016.

[8] Yusuf Yiğit Pilavcı, Pierre-Olivier Amblard, Simon Barthelmé, and Nicolas Tremblay, "Graph Tikhonov regularization and interpolation via random spanning forests," *IEEE transactions on Signal and Information Processing over Networks*, vol. 7, pp. 359–374, 2021.

[9] Michaël Fanuel and Rémi Bardenet, "Sparsification of the regularized magnetic Laplacian with multi-type spanning forests," *arXiv preprint arXiv:2208.14797*, 2022.

[10] Luca Avena and Alexandre Gaudillière, "On some random forests with determinantal roots," *Preprint: Weierstraß-Institut für Angewandte Analysis und Stochastik*, vol. 1881, 2013.

[11] Robin Forman, "Determinants of Laplacians on graphs," *Topology*, vol. 32, no. 1, pp. 35–46, 1993.

[12] David Bruce Wilson, "Generating random spanning trees more quickly than the cover time," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 296–303.

[13] Adrien Kassel and Richard Kenyon, "Random curves on surfaces induced from the Laplacian determinant," *The Annals of Probability*, vol. 45, no. 2, pp. 932–964, 2017.

[14] David Blackwell, "Conditional expectation and unbiased sequential estimation," *The Annals of Mathematical Statistics*, pp. 105–110, 1947.

[15] C Radhakrishna Rao, "Information and the accuracy attainable in the estimation of statistical parameters," in *Breakthroughs in statistics*, pp. 235–247. Springer, 1992.

[16] Zdravko Botev and Ad Ridder, "Variance reduction," *Wiley statsRef: Statistics reference online*, pp. 1–6, 2017.

[17] Yusuf Pilavcı, Pierre-Olivier Amblard, Simon Barthelmé, and Nicolas Tremblay, "Variance reduction in stochastic methods for large-scale regularised least-squares problems," *arXiv preprint arXiv:2110.07894*, 2021.

[18] Afonso S Bandeira, "Ten lectures and forty-two open problems in the mathematics of data science," *Lecture Notes*, 2015.

[19] Yousef Saad, *Numerical methods for large eigenvalue problems: revised edition*, SIAM, 2011.

[20] Maurice G Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[21] Tali Kaufman, Rasmus Kyng, and Federico Soldá, "Scalar and matrix chernoff bounds from l_∞-independence," in *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2022, pp. 3732–3753.

[22] Fan RK Chung and Fan Chung Graham, *Spectral graph theory*, vol. 92, American Mathematical Soc., 1997.

[23] Odile Macchi, "The coincidence approach to stochastic point processes," *Advances in Applied Probability*, pp. 83–122, 1975.

[24] Alex Kulesza and Ben Taskar, "Determinantal point processes for machine learning," *Foundations and Trends® in Machine Learning*, vol. 5, no. 2–3, pp. 123–286, 2012.

[25] Michał Derezinski and Michael W Mahoney, "Determinantal point processes in randomized numerical linear algebra," *Notices of the American Mathematical Society*, vol. 68, no. 1, pp. 34–45, 2021.

A. TECHNICAL DEFINITIONS

We provide some technical definitions used in Appendices B and C.

Recall the definition of the graph Laplacian $L = \nabla^*\nabla$ [22], with $\nabla \in \mathbb{R}^{E\times V}$ the (weighted) $|E| \times |V|$ edge-vertex incidence matrix. The graph Laplacian reveals many properties of functions $f \in \mathbb{R}^V$ defined on G. For instance, if the graph is connected, the one-dimensional kernel of L is generated by the constant function.

When considering complex signals $f \in \mathbb{C}^V$ on G, one first defines a *unitary connection* given by a family $\psi_{v,e} = e^{i\theta_{v,e}}$ of unitary complex numbers, extended by $\psi_{v,e} = \psi^*_{e,v}$ and $\psi_{v,v'} = \psi_{e,v'}\psi_{v,e} = e^{i\theta v,v'}$. This connection gives rise to a *magnetic Laplacian* $L = \nabla^*\nabla$ [3], with ∇ a twisted discrete differential:

$$\nabla_{e,v} = \begin{cases} -\sqrt{w_e}\psi_{v,e} & \text{if } v = a, \\ \sqrt{w_e}\psi_{v,e} & \text{if } v = b, \\ 0 & \text{otherwise,} \end{cases}$$

for $e = (a, b)$. Unlike the usual graph Laplacian, the magnetic Laplacian has a trivial kernel ($\ker L = 0$), unless the connection is *trivial* [9], which is a key difference in the analysis of the DPP described in Appendix B.1.

B. DPP AND SMOOTHING ESTIMATORS

B.1. Background on DPP and the MTSF process

A (discrete) Determinantal Point Process (DPP) is a distribution over subsets of a finite set $\mathcal{X}$ parameterized by a symmetric matrix $K \in M_{|\mathcal{X}|}(\mathbb{C})$ whose eigenvalues all lie in $[0, 1]$, this matrix is the *marginal kernel* of the process. We say that $X \subseteq \mathcal{X}$ is distributed according to this DPP if $\mathbb{P}(A \subseteq X) = \det K_{A,A}$ for every $A \subseteq \mathcal{X}$, where $K_{S,T}$ denotes the submatrix matrix of K whose rows (resp. columns) have been restricted to S (resp. T). If K is a projection matrix (i.e. its eigenvalues are either 0 or 1), additional structural properties are ensured. For instance, a DPP based of K will only give non-zero probability to samples of size $\mathrm{rk}(K)$, the rank of K.

DPPs were introduced as *repulsive* processes [23], and have been sought after in machine learning [24] and randomized linear algebra [25].

Proposition 4. *The probabilities in Equation* (3) *define a (projective) DPP over $E \cup V$, with marginal kernel*

$$K_m = \begin{bmatrix} \nabla \\ \sqrt{Q} \end{bmatrix} (L + Q)^{-1} \begin{bmatrix} \nabla^* & \sqrt{Q} \end{bmatrix},$$

with Q the $|V| \times |V|$ diagonal matrix of the q_v's.

Proof. See [9]. This can also be proved by a straightforward extension of the computations in [3]. □

This property further relates our study to that of USTs, RSFs and FUs, which have also been showed to be DPPs [3, 10].

When working with DPPs, the Cauchy-Binet formula is an important technical tool, and states that for an $m \times m$ matrix $M = AB$, with A $m \times n$, B $n \times m$ and where $n > m$ we have:

$$\det M = \sum_T \det A_{:,T} \det B_{T,:},$$

where the sum is over $T \subseteq \{1, ..., n\}$ of size $|T| = m$.

B.2. Sampling

We briefly recall the sampling algorithm from [9]. Under the condition in Equation (4), a MTSF ϕ distributed according to Equation (3) is built by iterating the following until all nodes belong to ϕ.

a) Start a random walk from any node v not in ϕ, while keeping track of the path p built from this random walk. The transitions are as follows: with probability proportional to q, set the current node as a root. Otherwise, choose a successor u in the neighborhood of the current node w with probability proportional to $w_{(w,u)}$.

b) If a node r is designated as a root, add it to the forest ϕ together with the path p and stop the random walk.

c) If the path p self-intersects at $w \in V$ before reaching a node in ϕ, and forms a loop C: keep this loop in p with probability $1 - \cos(\theta_C)$, add p to the forest ϕ and stop the random walk. Otherwise, erase this loop and continue the random walk from w.

d) If p reaches a node in ϕ, stop the random walk and add the path p to the forest ϕ.

Note that the θ_C *a priori* depends on the orientation of the loop, but $\cos(\theta_C)$ is orientation-agnostic. Such random walk based sampling algorithms are a noteworthy instance of efficient DPP sampling, which usually require a diagonalization of the kernel K and would result here in $\mathcal{O}((|E|+|V|)^3)$ sampling time.

C. PROOFS FOR SECTION 3

We provide straightforward proofs of the Propositions 1 and 2 in Section 3. Unless otherwise specified, we work with rooted MTSFs sampled according to Equation (3).

C.1. Proof of Proposition 1

The main part of the proof consists in rewriting $(L+Q)^{-1}$ as an expectation (see Equation (7)). We first use Cramer's rule to express the coefficients of $(L + Q)^{-1}$:

$$(L + Q)^{-1}_{i,j} = (-1)^{i+j} \frac{\det(L + Q)_{-j,-i}}{\det(L + Q)},$$

where $-k = V \setminus \{k\}$.

For the denominator, we obtain the following sum over MTSFs (see e.g. [9])

$$\det(L+Q) = \sum_{\phi} \prod_{r \in \rho(\phi)} q_r \prod_{e \in \phi_\bullet} w_e \prod_{C} (2 - 2\cos(\theta_C)),$$

which is the normalizing constant in Equation (3). The Cauchy-Binet formula further allows to express $\det(L+Q)_{-j,-i}$ as:

$$\sum_{\phi \subseteq E \cup V, |\phi| = |V|-1} \det\left[\nabla^* \sqrt{Q}\right]_{-j,\phi} \det \begin{bmatrix} \nabla \\ \sqrt{Q} \end{bmatrix}_{\phi,-i},$$

which can be re-arranged to obtain:

$$\sum_{\phi \subseteq E \cup V, |\phi| = |V|-1} (-1)^{i+j} \det\left[\left[\nabla^* \sqrt{Q}\right]_{:,\phi} \delta_j\right] \det \begin{bmatrix} \begin{bmatrix} \nabla \\ \sqrt{Q} \end{bmatrix}_{\phi,:} \\ \delta_i^* \end{bmatrix} \tag{6}$$

with the canonical basis vector $\delta_i \in \mathbb{R}^V$.

The product of determinants in Equation (6) can be split along the connected components $\phi^k \subseteq E \cup V$ of ϕ. Using similar computations as [3], we find this product is non-zero if one of these connected components ϕ^k is a tree containing i and j with no root for which the product of determinants is $\psi_{j \to i} \prod_{e \in \phi^k_\bullet} w_e$ and, if this component does not span the entire graph, the remaining subgraph contains disjoint connected components which have to be taken among the following subgraphs:

- a unicycle with cycle C for which the product is $(2 - 2\cos(\theta_C)) \prod_{e \in \phi^k_\bullet} w_e$,
- a rooted tree with root r for which the product is $q_r \prod_{e \in \phi^k_\bullet} w_e$.

In the end, when computing the sum, this translates to:

$$\begin{aligned} (L+Q)^{-1}_{i,j} &= \frac{1}{q_j} \sum_{\phi} \mathbf{1}_{i \sim j} \psi_{j \to i} \mathbb{P}(\phi) \\ &= \frac{1}{q_j} \mathbb{E}(\mathbf{1}_{i \sim j} \psi_{j \to i}), \end{aligned} \tag{7}$$

where $\mathbf{1}_{i \sim j}$ is the indicator that i belongs to a tree whose root is j.

Writing out the formula for the optimal solution f_o finally yields:

$$\begin{aligned} f_o(i) &= \delta_i^* (L+Q)^{-1} Q g \\ &= \sum_j q_j (L+Q)^{-1}_{i,j} g(j) \\ &= \sum_j \mathbb{E}(\mathbf{1}_{i \sim j} \psi_{j \to i} g(j)) \\ &= \mathbb{E}(\tilde{f}(i)), \end{aligned} \tag{8}$$

where the second to last equation is obtained from Equation (7), and the last line follows from the law of total expectation.

C.2. Proof of Proposition 2

Suppose that $\phi_\bullet = \pi$, denote by r_k the root of the connected component $\phi^k_\bullet$, and by V_k the set of vertices in ϕ^k. Then, if v belongs to π^k a tree:

$$\begin{aligned} \mathbb{E}_{\phi^k}(\tilde{f}(v) \mid \phi^k_\bullet = \pi^k) &= \sum_{r \in V_k} \mathbb{P}(r_k = r) \tilde{f}(v) \\ &= \sum_{r \in V_k} \frac{q_r}{\sum_{w \in V_k} q_w} \psi_{r \to v} g(r) \\ &= \overline{f}(v). \end{aligned}$$

If π^k is a unicycle, then $\mathbb{E}(\tilde{f}(v)) = 0$.

The two consequences follow from the laws of total expectation and variance.