Recovering Planted Colorings in Sublinear Time

Weronika Wrzos-Kaminska
EPFL, Lausanne, Switzerland

Abstract

We study the problem of recovering a planted k-coloring in sublinear time. Given an expander G with a planted coloring, the goal is to efficiently construct a small-space data structure that allows consistent color queries: given a vertex v, the algorithm quickly returns the color of v according to the planted solution.

We work in the adversarial planted coloring model of David and Feige [STOC 2016], where an adversary chooses a d-regular spectral λ-expander G on n vertices and plants a balanced k-coloring by partitioning the vertices into k equal parts and deleting all edges within each part. This model generalizes the earlier random graph models studied by Blum and Spencer [J. Algorithms 1995] and Alon and Kahale [STOC 1994].

We give the first sublinear-time algorithm for recovering planted colorings in this model. In the adjacency list model, our algorithm has preprocessing time and space $\widetilde{O}\left(n^{1/2+O(1/\log(d/\lambda))}\right)$, and produces a data structure that answers color queries in time $\widetilde{O}\left(n^{1/2+O(1/\log(d/\lambda))}\right)$. With a high constant probability, the resulting labeling agrees with the planted coloring on all but an $O(\sqrt{\lambda/d})$ fraction of vertices, up to a permutation of the k colors.

Our algorithm gives sublinear time inner product access to the bottom eigenspace of the normalized adjacency matrix by using random walks, which allows us to adapt the classical spectral approach of Alon and Kahale in sublinear time.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases sublinear algorithms, spectral algorithms, graph coloring

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2026.45

Category RANDOM

Related Version *Full Version*: `https://arxiv.org/abs/2608.02338`

1 Introduction

Graph coloring is a fundamental problem in theoretical computer science. In the k-coloring problem, the goal is to assign each vertex a color from $[k]$ so that adjacent vertices receive different colors. Finding k-colorings in graphs for $k \geq 3$ is among the first computational problems shown to be NP-hard [24, 18]. This has motivated extensive work on approximation algorithms and on identifying assumptions under which the problem can be solved efficiently [33, 5, 7, 6, 8, 1, 23, 27, 28, 2, 13, 3, 12, 25, 29, 26, 9, 20].

One influential line of work studies planted coloring models, where a hidden coloring is embedded into a graph, and the task is to recover (approximately) the planted solution. Blum and Spencer [6] and Alon and Kahale [1] considered random graph models with a planted k-coloring, and showed that polynomial-time spectral algorithms can recover the planted k-coloring with high probability. David and Feige [12] extended this to a more challenging adversarial setting, in which an adversary chooses an expander and plants a balanced k-coloring by deleting all edges within each color class. They showed that, even in this adversarial setting, the spectral approach of Alon and Kahale recovers the planted coloring on a 0.99 fraction of the vertices, while exact recovery is NP-hard.

While planted coloring models have been extensively studied in the polynomial-time regime, the question of *sublinear-time* recovery remains largely unexplored. This motivates the central question of this work:

Is it possible to recover a planted k-coloring in sublinear time?

Of course, any algorithm that explicitly outputs a coloring requires $\Omega(n)$ time simply to write down the labels. We therefore focus on constructing a *coloring oracle*: a small-space randomized data structure that quickly answers queries of the form "what color does vertex v have?". The oracle must be *consistent*, meaning that its answers agree with a single underlying assignment of colors (not necessarily a proper k-coloring), and *approximately correct*, meaning that the oracle disagrees with the planted coloring on only a small fraction of vertices, up to a permutation of colors (see Definition 8 for a formal definition of coloring oracle).

We consider a variant of the adversarial planted model of [12]. In this model, an adversary chooses a d-regular spectral λ-expander (i.e., eigenvalues of its adjacency matrix, other than the top eigenvalue d, have absolute value at most λ). The adversary also chooses a balanced partitioning of the vertices into k color classes, for a constant k. The planted instance is obtained by deleting all the monochromatic edges, i.e., all edges that have both endpoints in the same color class. Additionally, we require that the partitioning is chosen so that the degrees in the planted instance are balanced, in the sense that all vertices have degree $\Omega(d)$ after deleting the monochromatic edges (see Definition 6 for a formal definition of the input model).

A d-regular graph has $\lambda_1 = d$ and that the second largest eigenvalue in absolute value satisfies $\lambda = \Omega(\sqrt{d})$ [30]. Random d-regular graphs are essentially best possible spectral expanders, satisfying $\lambda = O(\sqrt{d})$ with high probability [14, 17, 16]. The adversarial planted coloring model therefore generalizes random planted models, in which the host graph is drawn at random and/or the planted partitioning is chosen uniformly at random.

Our contribution

We give the first sublinear-time algorithm for this planted coloring problem. Our result works in the adversarial planted model, and therefore also applies to the corresponding random planted models. Specifically, in the adversarial planted k-coloring model, we construct a coloring oracle with $\widetilde{O}(n^{1/2+O(1/\log(d/\lambda)}$ preprocessing time and space, where n is the number of vertices. The oracle answers each color query in $\widetilde{O}(n^{1/2+O(1/\log(d/\lambda))})$ time and, with a high constant probability, agrees with the planted coloring on at least a 0.99 fraction of the vertices.

▶ **Theorem 1.** *Let $k \geq 3$ be a constant, let c_k be a sufficiently small constant (depending on k), and let C_k be a sufficiently large constant (depending on k). For every $C_k < d < n$ and every $\lambda \leq c_k d$, there exists a randomized algorithm for the adversarial planted k-coloring model (Definition 6) that constructs a coloring oracle using $\widetilde{O}\big(n^{1/2+O(1/\log(d/\lambda))}\big)$ preprocessing time and space, where n denotes the number of vertices in the graph. For every query vertex v, the oracle outputs a color in $[k]$ in time $\widetilde{O}\big(n^{1/2+O(1/\log(d/\lambda))}\big)$, and its answers are consistent across queries. With probability at least 0.9, the oracle agrees with the planted coloring (up to a permutation of the colors) on all but at most $O\left(\sqrt{\frac{\lambda}{d}}n\right)$ vertices.*

In particular, the running time is $\approx n^{1/2+o(1)}$ when $\lambda/d = o(1)$. For dense graphs with $d = n^\delta$ and $\lambda = O(\sqrt{d})$ for a constant δ, the running time becomes $\widetilde{O}(n^{1/2})$. As in prior works on planted coloring, we treat the number of colors k as a constant.

Our algorithm may be viewed as a sublinear-time implementation of the classical spectral algorithm of [1]. The key technical ingredient is providing sublinear-time dot-product access to the bottom eigenspace of the adjacency matrix, which we achieve by adapting recent techniques from spectral clustering [22].

Related work on planted coloring

Blum and Spencer [6] and Alon and Kahale [1] studied the problem of planted k-coloring in random graphs, and showed that exact recovery is possible in polynomial time. [12] studied the role of randomness in the planted model, and showed that exact recovery is possible if the coloring is planted at random in an adversarial spectral expander, and that approximate recovery is possible even when the coloring is adversarial. [29] proved that approximate recovery is possible in low threshold rank graphs for pseudorandom colorings. Recently, [4, 9, 20] further considered the setting in which the coloring is planted in a one-sided expander or a low threshold rank graph.

Related work on sublinear spectral algorithms

Our techniques are closely related to work on sublinear spectral clustering [11, 10, 19, 32, 15, 22]. Most relevant to our work is [22], which uses a low-degree polynomial of the random walk matrix to provide access to the top eigenspace of the adjacency matrix under a clusterability assumption. Although this setting differs from ours (our graphs are not clusterable, and the signal lies in a different part of the spectrum), we adopt their general approach. [31] and [21] applied spectral techniques to approximate max-cut (and more generally max-2Lin) in sublinear time on expanders and clusterable graphs. The work of [31] also showed that testing whether an expander is 3-colorable versus far from 3-colorable requires $\Omega(n)$ time. We note that the hard instances in [31] are constant degree expanders with expansion $\phi = \Omega(1/d)$, whereas our results require stronger spectral expansion.

2 Technical Overview

Our algorithm can be viewed as a sublinear implementation of the spectral approaches of Alon and Kahale [1] and David and Feige [12]. These works show that planted colorings can be approximately recovered from the bottom eigenspace of the adjacency matrix. The main difficulty is that computing this exactly requires reading the whole graph and computing the eigenvectors.

Recent progress on sublinear spectral algorithms has focused on approximating inner products of spectral embeddings associated with the *top* eigenspace of a graph, typically under strong clusterability assumptions [19, 22]. Our main technical contribution is to extend this line of work to the setting of graph coloring by providing sublinear-time access to the *bottom* eigenspace of the adjacency matrix. While bottom eigenvectors have long played a central role in polynomial-time spectral algorithms, sublinear-time methods for accessing this part of the spectrum have remained largely undeveloped.

The spectral approach of [1] and [12]

We start by reviewing the classical approach of [1] and [12]. These works showed that an approximate coloring can be recovered from the bottom eigenspace of the adjacency matrix of the planted input graph. Specifically, the bottom $k-1$ eigenvalues of the normalized

adjacency matrix have value $-\frac{1}{k-1} \pm O(\sqrt{\lambda/d})$, while all remaining eigenvalues (except for the trivial top eigenvalue 1) are bounded in absolute value by $O(\sqrt{\lambda/d})$ [1]. The coloring can be approximately recovered from the spectral embedding in this bottom eigenspace.

▶ **Definition 2** (Spectral embedding). *Let $\bar{A} = D^{-1/2}AD^{-1/2}$ denote the normalized adjacency matrix of the input graph G', and let $U_{[k-1]} \in \mathbb{R}^{n\times(k-1)}$ denote the matrix whose columns are the bottom $k-1$ eigenvectors of $\bar{A}$. For every vertex $v \in V$, the spectral embedding of v, denoted $f_v \in \mathbb{R}^{k-1}$, is the v-th row of $U_{[k-1]}$, i.e.,*

$$f_v \coloneqq U_{[k-1]}^\top \mathbb{1}_v.$$

▶ Remark 3. The spectral embeddings f_v are not uniquely defined. However, the dot products $\langle f_u, f_v\rangle$, and hence also the Euclidean distances $\|f_u - f_v\|_2$, are uniquely defined (see Remark 17).

As shown in [12], spectral embeddings of vertices belonging to the same color class are close to each other, while spectral embeddings of vertices belonging to different color classes are far from each other. Specifically, for all but an $O(\lambda/d)$ fraction of vertices, the following holds[2]:

$$\|f_u - f_v\|_2^2 = \begin{cases} \leq \dfrac{k}{70n}, & \text{if } \chi(u) = \chi(v), \\ \geq \dfrac{k}{2n}, & \text{if } \chi(u) \neq \chi(v). \end{cases} \tag{1}$$

By Equation (1), approximate recovery of the planted coloring can be achieved using the following two steps:

1. Identify a set of *color representatives* $c_1 \in C_1, \ldots, c_k \in C_k$, i.e., one vertex from each color class.
2. Given a query vertex u, output the color of the closest color representative

$$\textsc{Color}(u) = \arg\min_{i\in[k]} \|f_u - f_{c_i}\|_2^2.$$

The main challenge is to perform this in sublinear time, without explicitly computing the full spectral embedding.

Spectral embeddings via polynomials

In order to approximate inner products $\langle f_u, f_v\rangle$ between spectral embeddings (and hence also the Euclidean distances), we will approximate the relevant projection matrix by a low-degree polynomial in the random walk matrix. The inner products can then be approximated by collision counting. This idea was used in [22] in the context of sublinear-time spectral clustering, and we adapt their general approach. Let $\bar{A} = U\Sigma U^\top$ denote the eigendecomposition of the normalized adjacency matrix $\bar{A}$. We are interested in approximating

$$\langle f_u, f_v\rangle = \mathbb{1}_u^\top U_{[k-1]} U_{[k-1]}^\top \mathbb{1}_v,$$

[1] [12] analyze the unnormalized adjacency matrix. Since our algorithm is based on random walks, it is more convenient for us to work with the normalized adjacency matrix. An analogous statement for the normalized adjacency matrix $\bar{A}$ follows by a similar argument (see Lemma 16).

[2] Equation (1) can be extracted from the analysis of [12] for embeddings of the *unnormalized* adjacency matrix, although it is not stated there explicitly in this form. We prove an analogous guarantee for the normalized adjacency matrix (see Lemma 20).

where $U_{[k-1]}U_{[k-1]}^\top$ is the projection onto the eigenspace corresponding to eigenvalues with value $\approx -1/(k-1)$. To this end, we seek a polynomial $p(x) = \sum_{t\geq 0} c_t x^t$ of the form $p(x) = x^t q(x)$, where $t = \Omega\left(\frac{\log n}{\log(d/\lambda)}\right)$, such that the coefficients c_t are small and the following properties hold:

- $p(x) \approx 1$ when $\left|x + \frac{1}{k-1}\right| \leq O\left(\sqrt{\frac{\lambda}{d}}\right)$, and
- $p(x) \approx 0$ when $|x| \leq O\left(\sqrt{\frac{\lambda}{d}}\right)$ or $x = 1$.

With such a polynomial, the matrix $p(\bar{A}) = Up(\Sigma)U^\top$ acts as an approximate projector onto the bottom $k-1$ eigenvectors. As a result,

$$\begin{aligned}\langle f_u, f_v\rangle &= \mathbb{1}_u^\top U_{[k-1]}U_{[k-1]}^\top \mathbb{1}_v \\ &\approx \mathbb{1}_u^\top U p(\Sigma)^2 U^\top \mathbb{1}_v = \langle p(\bar{A})\mathbb{1}_u, p(\bar{A})\mathbb{1}_v\rangle = \left\langle \sum_t c_t \bar{A}^t \mathbb{1}_u, \sum_t c_t \bar{A}^t \mathbb{1}_v \right\rangle \\ &= \sum_{t,t'} c_t c_{t'} \langle \bar{A}^t \mathbb{1}_u, \bar{A}^{t'} \mathbb{1}_v\rangle.\end{aligned}$$

Each inner product $\langle \bar{A}^t \mathbb{1}_u, \bar{A}^{t'} \mathbb{1}_v\rangle$ can be estimated efficiently using random walks. Indeed, let $M = AD^{-1}$ denote the random walk transition matrix. For $t = \Omega\left(\frac{\log n}{\log(d/\lambda)}\right)$, inner products of the form $\langle M^t \mathbb{1}_u, M^{t'} \mathbb{1}_v\rangle$ can be approximated using $\approx \sqrt{n}$ random walks via standard collision-counting arguments. Using the relation $\bar{A} = D^{-1/2}MD^{1/2}$, inner products involving powers of $\bar{A}$ can be approximated similarly.

It is crucial that the coefficients c_t of the polynomial p are small: each term $\langle \bar{A}^t \mathbb{1}_u, \bar{A}^{t'} \mathbb{1}_v\rangle$ must be estimated to precision inverse polynomial in $|c_t c_{t'}|$, and hence the overall running time depends polynomially on the size of coefficients.

[22] used a similar idea to cluster graphs. In their setting, the input graph is assumed to be clusterable, meaning that the top k eigenvalues lie in the interval $[1-\epsilon, 1]$ for some error parameter ϵ, while all remaining eigenvalues are bounded away from 1. The goal is to approximate embeddings in this *top* eigenspace of the graph, which they achieve by constructing a polynomial that approximates the indicator on $[1-\epsilon, 1]$.

A key challenge with adapting their construction to our setting is that the coefficients of their polynomial grows superpolynomially in their error parameter ϵ, and can be as large as $\approx (1/\epsilon)^{O(\log(1/\epsilon))} n^{O(\epsilon)}$. Since the running time depends polynomially on the coefficient size, the algorithm is sublinear only for relatively large values of ϵ.

In our setting, the relevant analogue of the error parameter is $\sqrt{\lambda/d}$, and a superpolynomial dependence on $\sqrt{d/\lambda}$ would lead to too large running times. For example, for random d-regular graphs with $d = n^{\Omega(1)}$, this would lead to running times that are superpolynomial in n.

To overcome this issue, we modify the polynomial construction of [22] and give a more careful analysis of the resulting polynomial. This gives a polynomial that approximates an indicator function around $-1/(k-1)$, while having coefficients with absolute at value most $n^{O(1/\log(d/\lambda))}$. In particular, the coefficient size, and hence the running time, decreases as λ/d decreases.

▶ **Theorem 4.** *Let $\delta, C > 0$ be constants. For every $\frac{10C\log n}{\log(d/\lambda)} \leq t \leq O\left(\frac{\log n}{\log(d/\lambda)}\right)$, there exists a polynomial p of the form $p(x) = x^t q(x)$, where q is a polynomial of degree $O(\sqrt{\lambda/d}\log n)$, such that the coefficients of p are bounded in absolute value by $n^{O\left(\frac{1}{\log(d/\lambda)}\right)}$, and*

1. *$|p(x) - 1| \leq \delta$ for $x \in \left[-\frac{1}{k-1} - C\sqrt{\frac{\lambda}{d}}, -\frac{1}{k-1} + C\sqrt{\frac{\lambda}{d}}\right]$*
2. *$|p(x)| \leq n^{-2}$ for $|x| \leq C\sqrt{\frac{\lambda}{d}}$.*

In fact, our proof establishes a more general result: for any constant $\alpha \in [-1, 1]$ with $\alpha \neq 0$ and any $0 < \epsilon \ll |\alpha|$, we construct a polynomial approximating an indicator on $[\alpha - \epsilon, \alpha + \epsilon]$ with similar coefficient bounds (see Theorem 28). The proof of Theorem 4 is presented in Section 5.4.

While this polynomial successfully maps the eigenvalues near $-1/(k-1)$ to values close to 1 and eigenvalues of size at most $\sqrt{\lambda/d}$ to values close to 0, it does not control the value of $p(1)$, which corresponds to the top eigenvalue. We handle this issue by explicitly removing the contribution of the top eigenvector. Since the top eigenvector of $\bar{A}$ is $e_1 = D^{1/2}\mathbb{1}/\sqrt{2m}$, its contribution can be computed exactly, and we can therefore apply the polynomial to the matrix $\bar{A} - e_1 e_1^\top$, in which the top eigenvalue has been shifted to zero.

The algorithms for approximating $\langle f_u, f_v \rangle$ (Algorithm 2) and $\|f_u - f_v\|_2^2$ (Algorithm 3) are presented in Section 4. Their guarantees (Lemma 10 and Lemma 11) are proved in Section 5.2.

Finding color representatives

Given access to pairwise distances between spectral embeddings, it remains to pick one representative vertex from each planted color class. The polynomial-time approach of [12] finds such representatives by repeatedly sampling k random vertices until exactly one comes from each class. However, their method for verifying whether a given sample is valid requires $\Omega(n)$ time, which is too costly in our setting. We instead sample $\Theta(k \log k)$ vertices and build a small "similarity graph" on the sample, connecting two sampled vertices when their embeddings are similar. This is similar to approaches used for clustering [11, 32].

The similarity graph decomposes into exactly k connected components (one per color class) with a high constant probability, so choosing one vertex from each component yields the required representatives. We present the algorithm for obtaining color representatives in Section 4.1.

3 Preliminaries and Notation

We start by formally defining the graph access model and the input model.

Graph access model. We work in the adjacency-list query model. In this model, an algorithm can perform neighbor queries, i.e., for the i-th neighbor of any vertex, and degree queries, i.e., for the degree of any vertex. We additionally assume that the number of edges m is given as part of the input to the algorithm.

▶ **Definition 5** (λ-expander). *Let $\lambda_1(A_G) \geq \lambda_2(A_G) \geq \cdots \geq \lambda_n(A_G)$ denote the eigenvalues of the unnormalized adjacency matrix A_G of a graph G. Say that G is a λ-expander if $\max\{|\lambda_2(A_G)|, |\lambda_n(A_G)|\} \leq \lambda$.*

▶ **Definition 6** (Adversarial planted k-coloring). *Let $k \geq 3$ be a constant. An adversary chooses an n-vertex d-regular λ-spectral expander $G = (V, E)$ together with a balanced partition $\mathcal{C} = (C_1, \ldots, C_k)$ of V, such that $|C_i| = n/k$ for all $i \in [k]$ and such that every vertex has at least $\Omega(d)$ neighbors outside its color class. The partition $\mathcal{C}$ defines a* planted k-coloring $\chi : V \to [k]$ *by setting $\chi(v) = i$ if $v \in C_i$.*

The planted instance is obtained by deleting all monochromatic edges from G. That is, let

$$E' \coloneqq \{(u, v) \in E : \chi(u) \neq \chi(v)\},$$

and let $G' = (V, E')$ denote the resulting k-colorable graph. We write $G' = P_{\mathcal{C}}(G)$ when G' is obtained this way from the graph G and the partition $\mathcal{C}$.

▶ Remark 7 (Model assumptions). The model above is almost identical to the planted adversarial model of [12], except that we impose the additional condition that every vertex has at least a small constant fraction of its neighbors outside of its own color class. This ensures that after deleting monochromatic edges, every vertex in G' has degree $\Omega(d)$. Since the typical degree in G' is $(1-\frac{1}{k})d$, this is a mild restriction that only rules out vertices that are almost isolated by the adversary. In particular, if the color classes are selected *at random* and $d = \Omega(\log n)$, then this holds with high probability. The purpose of this assumption is to ensure that the normalized adjacency matrix of G' has the required spectral properties. Our sublinear-time algorithm is based on random walks and therefore relies on spectral properties of the normalized adjacency matrix. In contrast, prior polynomial-time spectral algorithms for planted k-coloring work directly with the *unnormalized* adjacency matrix, and therefore do not need this additional balanced degree condition.

For simplicity of presentation, Definition 6 assumes that the planted coloring is exactly balanced, i.e., $|C_i| = n/k$ for all $i \in [k]$. Our results extend to approximately balanced planted colorings satisfying $|C_i|/|C_j| = O(1)$ for all $i, j \in [k]$. The exact balance assumption is consistent with the standard formulation in prior work (e.g., [6, 1, 12]).

Similar to prior work, we assume that $k = O(1)$, and suppress the dependency on k in the O-notation.

The goal is to construct a coloring oracle, as defined below.

▶ **Definition 8** (Coloring oracle). *A coloring oracle is a randomized data structure $\mathcal{O}$ that, given query access to a graph G' from the adversarial planted k-coloring model, provides consistent query access to a function $\textsc{Color} : V \to [k]$.*

We say that $\mathcal{O}$ defines an ε-approximate coloring if, with probability at least 0.9 over the random bits of $\mathcal{O}$, there exists a permutation $\pi : [k] \to [k]$ such that

$$|\{v \in V : \textsc{Color}(v) \neq \pi(\chi(v))\}| \leq \varepsilon n,$$

where χ is the planted coloring.

Notation

Given the input graph $G' = (V, E')$, we let m denote the number of edges $|E'|$. We write A for the unnormalized adjacency matrix of G' and D for the diagonal degree matrix. Write $\bar{A} = D^{-1/2}AD^{-1/2}$ for its normalized adjacency matrix, and write $M = AD^{-1}$ for the random walk matrix. Observe that $M = D^{1/2}\bar{A}D^{-1/2}$.

Let $\lambda_1(\bar{A}) \geq \lambda_2(\bar{A}) \geq \cdots \geq \lambda_n(\bar{A})$ denote the eigenvalues of $\bar{A}$, and let $e_1(\bar{A}), \ldots, e_n(\bar{A})$ denote a corresponding orthonormal basis of eigenvectors. When it is clear that we are talking about the matrix $\bar{A}$, we omit the dependence on $\bar{A}$ from the notation.

For a vertex $v \in V$, we write $\mathbb{1}_v \in \mathbb{R}^n$ for the indicator of v, i.e., the vector with entry 1 at index v and 0 everywhere else. Similarly, for a set $S \subseteq V$, we write $\mathbb{1}_S \in \mathbb{R}^n$ for the indicator vector of S. We use the notation $\mathbb{1} \in \mathbb{R}^n$ for the vector with all entries equal to 1.

▶ **Observation 9.** *$\bar{A}D^{1/2}\mathbb{1} = D^{1/2}\mathbb{1}$. In particular,*

$$e_1(\bar{A}) = \frac{D^{1/2}\mathbb{1}}{\|D^{1/2}\mathbb{1}\|_2} = \frac{D^{1/2}\mathbb{1}}{\sqrt{2m}} \qquad \text{and} \qquad \lambda_1(\bar{A}) = 1.$$

For a polynomial q, write $\deg(q)$ to denote the degree of q and $\max|\text{coeff}(q)|$ to denote the absolute value of the largest (in absolute value) coefficient of q.

We use $\widetilde{O}$ notation to suppress $\operatorname{polylog} n$ factors.

4 The Algorithm

In this section, we present the algorithm. The algorithm SpectralDotProduct (Algorithm 2) approximates inner products $\langle f_u, f_v \rangle$ of spectral embeddings. It runs $\approx \sqrt{n}$ random walks of various lengths starting from u and v, and then combines the resulting collision counts with the coefficients of the polynomial–p from Theorem 4.

To change from the random walk matrix $M = AD^{-1}$ to the normalized adjacency matrix $\bar{A} = D^{-1/2}MD^{1/2}$, we appropriately reweigh collisions according to the matrix $D^{-1/2}$. Finally, we subtract the term $\frac{1}{2m}$ in order to cancel out the contribution of the top eigenvalue.

Algorithm 1 RandomWalks(r, t, u).

Run r random walks of length t starting from u
Let $\hat{p}_u(v)$ be the fraction of random walks that end at v ▷ vector $\hat{p}_u$ has support at most r
return $\hat{p}_u$

Algorithm 2 SpectralDotProduct(u, v).

$t_{\min} \leftarrow O\left(\frac{\log n}{\log(d/\lambda)}\right)$ ▷ shortest random walk length
$t_\Delta \leftarrow O(\sqrt{\lambda/d}\log n)$ ▷ number of different walk lengths
$r \leftarrow \widetilde{O}\left(n^{1/2+O\left(\frac{1}{\log(d/\lambda)}\right)}\right)$ ▷ number of random walks
for $t = t_{\min}, \dots t_{\min} + t_\Delta$ **do**
    $\hat{p}_u^t \leftarrow$ RandomWalks(r, t, u) ▷ vector $\hat{p}_u^t$ has support at most r
    $\hat{p}_v^t \leftarrow$ RandomWalks(r, t, v) ▷ vector $\hat{p}_v^t$ has support at most r
end for
return $\sqrt{\deg(u)}\sqrt{\deg(v)}\sum_{t,t'} c_t c_{t'}\left(\left\langle D^{-1/2}\hat{p}_u^t, D^{-1/2}\hat{p}_v^{t'}\right\rangle - \frac{1}{2m}\right)$
▷ c_t is the coefficient of x^t in the polynomial $p(x)$ from Theorem 4

SpectralDotProduct has the following guarantee: outside a small deterministic set B of "bad" vertices, SpectralDotProduct(u, v) approximates $\langle f_u, f_v \rangle$ up to an additive error of $O(1/n)$. The set B is defined formally in Definition 19 and consists of vertices whose degree or spectral embedding is atypical relative to their planted color class. We bound the size of B in Lemma 20.

▶ **Lemma 10** (Correctness of SpectralDotProduct). *The procedure SpectralDotProduct (Algorithm 2) runs in time* $\widetilde{O}\left(n^{1/2+O\left(\frac{1}{\log(d/\lambda)}\right)}\right)$. *For every* $u, v \in V \setminus B$,

$$\Pr\left[\left|\textsc{SpectralDotProduct}(u, v) - \langle f_u, f_v \rangle\right| \le \frac{1}{80}\frac{k}{n}\right] \ge 0.9,$$

where the probability is over the internal randomness of SpectralDotProduct.

We prove the correctness guarantee of SpectralDotProduct (Lemma 10) in Section 5.2.

To compute distances $\|f_u - f_v\|_2^2$, we use the algorithm SpectralDistance (Algorithm 3). The algorithm boosts the success probability of SpectralDotProduct and approximates $\|f_u - f_v\|_2^2 = \|f_u\|_2^2 + \|f_v\|_2^2 - 2\langle f_u, f_v \rangle$ by approximating each of the terms.

Algorithm 3 SpectralDistance(u, v).

for $i = 1, \dots O(\log n)$ **do**
  $\|f_u\|^2_{\text{apx},i} \leftarrow$ SpectralDotProduct(u, u) ▷ Approximate $\|f_u\|_2^2$
  $\|f_v\|^2_{\text{apx},i} \leftarrow$ SpectralDotProduct(v, v) ▷ Approximate $\|f_v\|_2^2$
  $\langle f_u, f_v\rangle_{\text{apx},i} \leftarrow$ SpectralDotProduct(u, v) ▷ Approximate $\langle f_u, f_v\rangle$
  $\|f_u - f_v\|^2_{\text{apx},i} \leftarrow \|f_u\|^2_{\text{apx},i} + \|f_v\|^2_{\text{apx},i} - 2\langle f_u, f_v\rangle_{\text{apx},i}$
end for
return $\text{median}_i(\|f_u - f_v\|^2_{\text{apx},i})$

▶ **Lemma 11** (Correctness of SpectralDistance)**.** *The procedure SpectralDistance (Algorithm 3) runs in time* $\widetilde{O}\left(n^{1/2+O\left(\frac{1}{\log(d/\lambda)}\right)}\right)$*. With probability at least* $1 - n^{-100}$*,*

$$\left|\textsc{SpectralDistance}(u, v) - \|f_u - f_v\|_2^2\right| \le \frac{k}{20n} \qquad \text{for all } u, v \in V \setminus B,$$

where the probability is over the internal randomness of SpectralDistance.

As a corollary, SpectralDistance(u, v) can be used to determine if u and v belong to the same color class, for all $u, v \in V \setminus B$.

▶ **Corollary 12.** *With probability at least* $1 - n^{-100}$*, the following holds for all* $u, v \in V \setminus B$*:*

- *If* $\chi(u) = \chi(v)$*, then* SpectralDistance$(u, v) \le \frac{k}{4n}$*.*
- *If* $\chi(u) \neq \chi(v)$*, then* SpectralDistance$(u, v) \ge \frac{k}{2n}$*.*

We prove Lemma 11 and Corollary 12 in Section 5.3. In particular, Algorithm 3 can be used as a coloring oracle: given a set of color representatives $c_1, \dots, c_k$, where each c_i belongs to a different color class and none of them lies in the set of bad vertices B, one can answer each query u by assigning the color of its nearest neighbor among the c_i. Algorithm 4 does this.

Algorithm 4 Color$\left(u, \{c_i\}_{i\in[k]}\right)$.
Input: vertex u and set of color representatives $\{c_i\}_{i\in[k]}$ (see Definition 13)
Output: Index $i \in [k]$ such that $\chi(u) = \chi(c_i)$

return $\arg\min_{i\in[k]}$ SpectralDistance(u, c_i)

4.1 Obtaining color representatives

It remains to find a valid set of color representatives. The procedure FindRepresentatives (Algorithm 5) below achieves this. The algorithm samples $\Theta(k \log k)$ vertices uniformly at random, which ensures that, with high probability, we sample at least one vertex from each color class. The algorithm then constructs a "similarity graph" on the sampled vertices, connecting two sampled vertices if their distance is sufficiently small. By the clustering properties of the spectral embeddings (see Lemma 20), with a good probability, vertices from the same color class form a clique in this similarity graph, while vertices from different color classes are disconnected. Finally, the algorithm selects one arbitrary vertex from each connected component of the similarity graph and discards the rest. We show that, with probability at least 0.99, this procedure returns exactly one representative from each color class, yielding a valid set of representatives for use in the coloring oracle. Algorithm 5 is very similar to the preprocessing routines of [32] and [15].

Algorithm 5 FindRepresentatives.

$S \leftarrow$ Multiset of $10k \log k$ vertices sampled uniformly at random from V
$H \leftarrow (S, \emptyset)$ ▷ H will be the same-color graph on S (see definition in full version)
for all $u, v \in S$ **do**
  if SpectralDistance$(u, v) \leq \frac{k}{3n}$ **then**
    Add edge $\{u, v\}$ to H
  end if
end for
$i \leftarrow 1$
while $V(H) \neq \emptyset$ **do**
  Select arbitrary vertex u from $V(H)$ ▷ Select a representative of a color class
  $c_i \leftarrow u$
  $T_i \leftarrow \{u\} \cup N_H(u)$ ▷ The sampled vertices from the same color class as u
  $H \leftarrow H[V(H) \setminus T_i]$ ▷ Remove all vertices from the color class of u
  $i \leftarrow i + 1$
end while
return $c_1, c_2, \ldots$

▶ **Definition 13** (Color Representatives). *A set of vertices $\{c_i\}_{i \in [k]}$ is called a* set of color representatives *if the following conditions hold:*

1. *Every c_i is a good vertex (see Definition 19).*
2. *The vertices have distinct colors, i.e., $\chi(c_i) \neq \chi(c_j)$ for all $i \neq j$.*

▶ **Lemma 14** (Correctness of FindRepresentatives). *The procedure FindRepresentatives (Algorithm 5) runs in time $\widetilde{O}(n^{1/2+O(1/\log(d/\lambda))})$. With probability at least 0.99, it returns a set of k color representatives (as per Definition 13).*

The proof of Lemma 14 follows the analysis of the preprocessing procedure in [15], and is presented in the full version.

5 Analysis

In this section, we analyze the algorithms presented in Section 4 and prove the main result, Theorem 1. The section is organized as follows. In Section 5.1, we define the set B of bad vertices and bound its size. In Section 5.2, we prove Lemma 10, establishing the correctness of SpectralDotProduct. In Section 5.3, we prove Lemma 11 and Corollary 12, establishing the correctness of SpectralDistance. In Section 5.4 we construct the polynomial and prove Theorem 4. Finally, in Section 5.5 we combine everything to prove Theorem 1.

5.1 Properties of the input instance

We start by stating some of the key properties of the input instance G'. The proofs of these properties are included in the full version.

▶ **Observation 15.** *By the model assumption (Definition 6),* $\deg(v) = \Theta(d)$ *for all $v \in G'$ and $\Omega(nd) \leq m \leq nd$.*

▶ **Lemma 16** (Normalized version of Lemma C.4 in [12]). *Let G be a d-regular λ-expander, where $\lambda \le c_k d$ (for some constant c_k), and let $G' = P_{\mathcal{C}}(G)$ (as per Definition 6). Then*

1. *The eigenvalues of the normalized adjacency matrix $\bar{A}$ of G' have the following spectrum:*
   a. $\lambda_1(\bar{A}) = 1$
   b. $\left|\lambda_i(\bar{A}) + \frac{1}{k-1}\right| \le O\left(\sqrt{\frac{\lambda}{d}}\right)$ *for* $n-k+2 \le i \le n$.
   c. $|\lambda_i(\bar{A})| \le O\left(\sqrt{\frac{\lambda}{d}}\right)$ *for* $2 \le i \le n-k+1$
2. *For every $x \in \mathrm{span}(\{\mathbb{1}_{C_i}\}_{i\in[k]})$ with $\langle x, \mathbb{1}\rangle = 0$, there exists $x^* \in \mathrm{span}(\{e_i(G')\}_{i\in[n-k+2,n]})$ such that $\|D^{1/2}x - x^*\|_2^2 = O\left(\sqrt{\frac{\lambda}{d}}\right)\|D^{1/2}x\|_2^2$. Here $C_1, \dots C_k$ denotes the planted partition into color classes.*

[12] proves analogous guarantees to Lemma 16 for the eigenvalues and eigenvectors of the *unnormalized* adjacency matrix (see Lemma C.4 in [12]). Our proof of Lemma 16 is similar and is included in the full version.

▶ Remark 17. It follows from guarantee 1 in Lemma 16 that the space spanned by the bottom $k-1$ eigenvectors of $\bar{A}$ is uniquely defined, i.e., the choice of $U_{[k-1]}$ is unique up to multiplication by an orthogonal matrix $R \in \mathbb{R}^{(k-1)\times(k-1)}$ on the right.

While the choice of f_v for $v \in V$ is not unique, the dot product between the spectral embedding of $u \in V$ and $v \in V$ is well defined, since for every orthogonal $R \in \mathbb{R}^{(k-1)\times(k-1)}$ one has

$$\langle R^\top f_u, R^\top f_v\rangle = (R^\top f_u)^\top (R^\top f_v) = f_u^\top R R^\top f_v = \langle f_u, f_v\rangle.$$

A useful property of the spectral embeddings is that they concentrate around their color class center.

▶ **Definition 18** (Color class center). *For each color class $C_i \in \mathcal{C}$, let μ_i denote the* color class center

$$\mu_i \coloneqq \frac{1}{|C_i|}\sum_{v\in C_i} f_v = \frac{1}{|C_i|} U_{[k-1]}^\top \mathbb{1}_{C_i}.$$

We define a set of *good* vertices consisting of those vertices that are close to their own color class center, far from other color class centers, and whose degree is close to the "typical" value $\frac{d(k-1)}{k}$.

▶ **Definition 19** (Good vertex). *Say that a vertex $v \in C_i$ is* good *if*

1. $\frac{d(k-1)}{k}\left(1 - \sqrt{\frac{\lambda}{d}}\right) \le \deg(v) \le \frac{d(k-1)}{k}\left(1 + \sqrt{\frac{\lambda}{d}}\right)$, *and*
2. $\|f_v - \mu_i\|_2^2 \le \frac{0.05k}{n}$, *and*
3. *for all $j \ne i$, $\|f_v - \mu_j\|_2^2 \ge \frac{k}{n}$.*

Say that the vertex is bad otherwise. Let $B \subseteq V$ denote the set of bad vertices.

We will show that the oracle is correct on all good vertices. Observe that the set B of bad vertices is deterministic once the input graph and planted coloring are fixed. It consists of vertices whose degree or spectral embedding is atypical relative to their planted color class.

▶ **Lemma 20.** *Let B be the set of bad vertices (as per Definition 19). Then*

$$|B| \le O\left(\sqrt{\frac{\lambda}{d}} \cdot n\right).$$

A similar bound for vertices with bad spectral properties was proved in [12]. While their bound is similar, the definition of "bad" vertices in [12] differs slightly from ours. The proof of Lemma 20 is included in the full version.

Additionally, we will use the fact that the ℓ_2 norm of good vertices is not too large.

▶ **Lemma 21.** *If v is good (as per Definition 19), then*

$$\|f_v\|_2^2 = O\left(\frac{1}{n}\right).$$

The proof of Lemma 21 is presented in the full version.

5.2 Analysis of SpectralDotProduct (Proof of Lemma 10)

In this section, we prove the correctness guarantee of SpectralDotProduct, namely Lemma 10. Recall that we wish to apply the polynomial from Theorem 4 to the normalized adjacency matrix $\bar{A}$ in order to approximate the projection $U_{[k-1]}U_{[k-1]}^\top$. However, since Theorem 4 does not provide any guarantees for $p(1)$, we need to shift the top eigenvalue. Therefore, it will be convenient for the analysis to consider the matrix $\bar{A} - e_1 e_1^\top$.

▶ **Definition 22.** *Let*

$$\bar{A}_\star = \bar{A} - e_1 e_1^\top.$$

We start by showing that $\mathbb{1}_u^\top p(\bar{A}_\star)^2 \mathbb{1}_v$ approximates the inner product $\langle f_u, f_v\rangle$. Later, in Claim 27, we will show the output of Algorithm 2 approximates $\mathbb{1}_u^\top p(\bar{A}_\star)^2 \mathbb{1}_v$.

▶ **Lemma 23.** *Let $\delta > 0$ be a sufficiently small constant, let p be the polynomial from Theorem 4 (applied with the parameter δ), and let B be the set of bad vertices from Definition 19. For every $u, v \in V \setminus B$,*

$$\left|\mathbb{1}_u^\top p(\bar{A}_\star)^2 \mathbb{1}_v - \langle f_u, f_v\rangle\right| \le \frac{1}{160}\frac{k}{n}.$$

Proof. The high-level idea is to decompose the matrix $\bar{A}_\star$ according to its spectrum. We separate it into two parts: one corresponding to the bottom $k-1$ eigenvalues, and one corresponding to the remaining $n-k+1$ eigenvalues. By Lemma 16, the bottom $k-1$ eigenvalues are all close to $-1/(k-1)$. Applying the polynomial p to this part therefore "boosts" these eigenvalues up to (approximately) 1, so this part closely approximates the spectral projection $U_{[k-1]}U_{[k-1]}^\top$. On the other hand, the remaining $n-k+1$ eigenvalues have absolute value at most $O(\sqrt{\lambda/d})$ (again by Lemma 16). The polynomial p maps these values close to zero, so the error introduced by this part is small.

We now do this more formally. Let $\bar{A}_\star = U\Sigma^\star U^\top$ denote the eigendecomposition of $\bar{A}_\star$, where $\Sigma^\star = \mathrm{diag}(0, \lambda_2, \lambda_3, \ldots, \lambda_n)$. Observe that $\bar{A}$ and $\bar{A}_\star$ share the same eigenbasis, so the matrix $U_{[k-1]}$ from Definition 2 is the matrix consisting of the last $k-1$ columns of U.

We write $\Sigma^\star_{[k-1]}$ for the $(k-1)\times(k-1)$ diagonal matrix consisting of the bottom $k-1$ diagonal entries of $\Sigma^\star$, and $\Sigma^\star_{\mathrm{top}}$ for the $(n-k+1)\times(n-k+1)$ diagonal matrix consisting of the remaining (top) entries. We let U_{top} denote the matrix formed by the first $n-k+1$ columns of U.

With this notation, we can decompose

$$p(\bar{A}_\star)^2 = Up(\Sigma^\star)^2U^\top = U_{[k-1]}p(\Sigma^\star_{[k-1]})^2U_{[k-1]}^\top \;+\; U_{\mathrm{top}}p(\Sigma^\star_{\mathrm{top}})^2U_{\mathrm{top}}^\top. \tag{2}$$

Recalling from Definition 2 that $\langle f_u, f_v\rangle = \mathbb{1}_u^\top U_{[k-1]} U_{[k-1]}^\top \mathbb{1}_v$, we can bound the error by decomposing $p(\bar{A}_\star)^2$ into the two terms in Equation (2):

$$\begin{aligned}
&\left|\mathbb{1}_u^\top p(\bar{A}_\star)^2 \mathbb{1}_v - \langle f_u, f_v\rangle\right| \\
&= \left|\mathbb{1}_u^\top U p(\Sigma^\star)^2 U^\top \mathbb{1}_v - \mathbb{1}_u^\top U_{[k-1]} U_{[k-1]}^\top \mathbb{1}_v\right| \\
&= \left|\mathbb{1}_u^\top U_{[k-1]} p(\Sigma^\star_{[k-1]})^2 U_{[k-1]}^\top \mathbb{1}_v + \mathbb{1}_u^\top U_{\text{top}} p(\Sigma^\star_{\text{top}})^2 U_{\text{top}}^\top \mathbb{1}_v - \mathbb{1}_u^\top U_{[k-1]} U_{[k-1]}^\top \mathbb{1}_v\right| \\
&= \left|\mathbb{1}_u^\top U_{[k-1]} \left(p(\Sigma^\star_{[k-1]})^2 - I_{k-1}\right) U_{[k-1]}^\top \mathbb{1}_v + \mathbb{1}_u^\top U_{\text{top}} p(\Sigma^\star_{\text{top}})^2 U_{\text{top}}^\top \mathbb{1}_v\right| \\
&\le \left|f_u^\top \left(p(\Sigma^\star_{[k-1]})^2 - I_{k-1}\right)\ f_v\right| + \left|\mathbb{1}_u^\top U_{\text{top}} p\left(\Sigma^\star_{\text{top}}\right)^2 U_{\text{top}}^\top \mathbb{1}_v\right|.
\end{aligned} \tag{3}$$

Here the second equality follows by Equation (2), and the last inequality follows by the triangle inequality and the identity $f_u = U_{[k-1]}^\top \mathbb{1}_u$.

We now bound the first summand in Equation (3). By Lemma 16 and the definition of $\Sigma^\star_{[k-1]}$, the matrix $\Sigma^\star_{[k-1]}$ is a diagonal matrix with entries in $\left[-\frac{1}{k-1} - O\left(\sqrt{\frac{\lambda}{d}}\right), -\frac{1}{k-1} + O\left(\sqrt{\frac{\lambda}{d}}\right)\right]$. Therefore, by Theorem 4, $p(\Sigma^\star_{[k-1]})^2$ is a diagonal matrix with entries in $[1-3\delta, 1+3\delta]$. Hence, $\left\|p(\Sigma^\star_{[k-1]})^2 - I_{k-1}\right\|_2 \le 3\delta$, so

$$\left|f_u^\top (p(\Sigma^\star_{[k-1]})^2 - I_{k-1})\ f_v\right| \le \|f_u\|_2 \left\|\left(p(\Sigma^\star_{[k-1]})^2 - I_{k-1}\right) f_v\right\|_2 \le 3\delta \|f_u\|_2 \|f_v\|_2 \le \frac{1}{320n}, \tag{4}$$

where the last inequality follows by Lemma 21 and the assumption that δ is a sufficiently small constant.

We now bound the second summand in Equation (3). The matrix $U_{\text{top}} p(\Sigma^\star_{\text{top}})^2 U_{\text{top}}^\top$ is positive semidefinite, and therefore

$$\begin{aligned}
\left|\mathbb{1}_u^\top U_{\text{top}} p(\Sigma^\star_{\text{top}})^2 U_{\text{top}}^\top\ \mathbb{1}_v\right| &\le \mathbb{1}_u^\top U_{\text{top}} p(\Sigma^\star_{\text{top}})^2 U_{\text{top}}^\top \mathbb{1}_u + \mathbb{1}_v^\top U_{\text{top}} p(\Sigma^\star_{\text{top}})^2 U_{\text{top}}^\top \mathbb{1}_v \\
&\le \left\|p\left(\Sigma^\star_{\text{top}}\right)^2\right\|_2 \left\|U_{\text{top}}^\top \mathbb{1}_u\right\|_2^2 + \left\|p\left(\Sigma^\star_{\text{top}}\right)^2\right\|_2 \left\|U_{\text{top}}^\top \mathbb{1}_v\right\|_2^2 \\
&\le 2\left\|p\left(\Sigma^\star_{\text{top}}\right)^2\right\|_2,
\end{aligned}$$

where the last inequality uses that $\left\|U_{\text{top}}^\top \mathbb{1}_w\right\|_2^2 \le \|U^\top \mathbb{1}_w\|_2^2 = \|\mathbb{1}_w\|_2^2 = 1$ for all $w \in V$.

By Lemma 16 and the definition of $\Sigma^\star_{\text{top}}$, the matrix $\Sigma^\star_{\text{top}}$ is a diagonal matrix with entries bounded in absolute value by $O(\sqrt{\lambda/d})$ Therefore, by Theorem 4, $\left\|p\left(\Sigma^\star_{\text{top}}\right)^2\right\|_2 \le n^{-2}$. So the second summand in Equation (3) can be bounded as

$$\left|\mathbb{1}_u^\top U_{\text{top}} p(\Sigma^\star_{\text{top}})^2 U_{\text{top}}^\top\ \mathbb{1}_v\right| \le 2\left\|p\left(\Sigma^\star_{\text{top}}\right)^2\right\|_2 \le 2n^{-2} \le \frac{1}{320n}. \tag{5}$$

Combining Equation (3), Equation (4) and Equation (5) gives the claim. ◀

The following lemma shows that the quantity computed by Algorithm 2 has expected value $\mathbb{1}_u^\top p(\bar{A}_\star)^2 \mathbb{1}_v = \sum_{t,t'} c_t c_{t'} \langle \bar{A}_\star^t \mathbb{1}_u, \bar{A}_\star^{t'} \mathbb{1}_v\rangle$.

▶ **Lemma 24.** *For every $u, v \in V$,*

$$\sqrt{\deg(u)}\sqrt{\deg(v)}\left(\langle D^{-1/2} M^t \mathbb{1}_u, D^{-1/2} M^{t'} \mathbb{1}_v\rangle - \frac{1}{2m}\right) = \langle \bar{A}_\star^t \mathbb{1}_u, \bar{A}_\star^{t'} \mathbb{1}_v\rangle.$$

Proof. We start by relating the left-hand side to $\langle \bar{A}^t \mathbb{1}_u, \bar{A}^{t'} \mathbb{1}_v \rangle$. From the definition of $M = AD^{-1}$ and $\bar{A} = D^{-1/2}AD^{-1/2}$, for every t, it holds that $M^t = D^{1/2}\bar{A}^t D^{-1/2}$. Therefore,

$$\begin{aligned}
&\langle D^{-1/2}M^t \mathbb{1}_u, D^{-1/2}M^{t'} \mathbb{1}_v \rangle = \mathbb{1}_u^\top \left(M^t\right)^\top D^{-1} M^{t'} \mathbb{1}_v \\
&= \mathbb{1}_u^\top D^{-1/2} (\bar{A}^t)^\top D^{1/2} D^{-1} D^{1/2} A^{t'} D^{-1/2} \mathbb{1}_v = \left(D^{-1/2}\mathbb{1}_u\right)^\top (\bar{A}^t)^\top A^{t'} D^{-1/2} \mathbb{1}_v \\
&= \frac{\langle \bar{A}^t \mathbb{1}_u, \bar{A}^{t'} \mathbb{1}_v \rangle}{\sqrt{\deg(u)}\sqrt{\deg(v)}}.
\end{aligned} \tag{6}$$

Next, we relate this to $\langle \bar{A}_\star^t \mathbb{1}_u, \bar{A}_\star^{t'} \mathbb{1}_v \rangle$. Recall from Observation 9 that $e_1 = \frac{D^{1/2}\mathbb{1}}{\sqrt{2m}}$. Since $\bar{A}e_1 = e_1$, we have $\bar{A}_\star^t = (\bar{A} - e_1 e_1^\top)^t = \bar{A}^t - e_1 e_1^\top$ for every $t \geq 1$. Therefore,

$$\begin{aligned}
\left\langle \bar{A}_\star^t \mathbb{1}_u, \bar{A}_\star^{t'} \mathbb{1}_v \right\rangle &= \left\langle \bar{A}^t \mathbb{1}_u, \bar{A}^{t'} \mathbb{1}_v \right\rangle - \langle e_1, \mathbb{1}_u \rangle \langle e_1, \mathbb{1}_v \rangle \\
&= \left\langle \bar{A}^t \mathbb{1}_u, \bar{A}^{t'} \mathbb{1}_v \right\rangle - \frac{\langle \mathbb{1}_u, D^{1/2}\mathbb{1} \rangle \langle \mathbb{1}_v, D^{1/2}\mathbb{1} \rangle}{2m} \\
&= \left\langle \bar{A}^t \mathbb{1}_u, \bar{A}^{t'} \mathbb{1}_v \right\rangle - \frac{\sqrt{\deg(u)}\sqrt{\deg(v)}}{2m}.
\end{aligned} \tag{7}$$

Combining Equation (6) and Equation (7) gives the lemma. ◀

To prove Lemma 10, we need to argue that we can estimate the quantity $\sqrt{\deg(u)}\sqrt{\deg(v)} \sum_{t,t'} c_t c_{t'} \left(\left\langle D^{-1/2}M^t \mathbb{1}_u, D^{-1/2}M^{t'} \mathbb{1}_v \right\rangle - \frac{1}{2m} \right)$ to a sufficiently high precision. We will argue this on a term-by-term basis, using the collision counting result stated in Lemma 25.

▶ **Lemma 25** (Weighted Collision Counting)**.** *Let p, q be two distributions $p, q \in \mathbb{R}^n$, let $\xi \in (0,1)$ be the desired precision and let $\zeta \in (0,1)$ be the desired failure probability. Let $w_1, \dots w_n \in (0, w_{\max}]$ and let $W = diag(w_1, \dots w_n)$. One can estimate $p^\top W q$ up to precision $\xi \cdot w_{\max}(\|p\|_2^2 + \|q\|_2^2)$ with success probability $1 - \zeta$ using $r = 5\xi^{-2}\zeta^{-1}(\|p\|_2^2 + \|q\|_2^2)^{-1/2}$ samples from each of p and q. That is, it is possible to compute $\hat{z} \in \mathbb{R}$ such that*

$$\Pr\left[|p^\top W q - \hat{z}| > \xi \cdot w_{\max}\left(\|p\|_2^2 + \|q\|_2^2\right)\right] \leq \zeta.$$

The proof of Lemma 25 is a standard collision counting argument, and is included in the full version.

We will apply Lemma 25 to the distributions $p = M^t \mathbb{1}_u$ and $q = M^{t'} \mathbb{1}_v$, and the weight matrix $W = D^{-1}$. Since both the error and the sample complexity in Lemma 25 depend on $\|p\|_2 = \|M^t \mathbb{1}_u\|_2$, we need to bound the norm $\|M^t \mathbb{1}_u\|_2$ for $u \in V \setminus B$. This is achieved by Lemma 26 below.

▶ **Lemma 26.** *Let C be a sufficiently large constant, and let $t \geq \frac{C \log n}{\log(d/\lambda)}$. Then for every $u \in V \setminus B$,*

$$\|\bar{A}^t \mathbb{1}_u\|_2^2 = \Theta\left(\frac{1}{n}\right) \qquad \textit{and} \qquad \|M^t \mathbb{1}_u\|_2^2 = \Theta\left(\frac{1}{n}\right).$$

Proof. Recall that $1 = \lambda_1 \geq \dots \geq \lambda_n$ denotes the eigenvalues of $\bar{A}$, and $e_1, \dots e_n$ denotes the corresponding eigenvectors. We have

$$\|\bar{A}^t \mathbb{1}_u\|_2^2 = \sum_{i=1}^n \lambda_i^{2t} \langle \mathbb{1}_u, e_i \rangle^2 = \frac{\langle \mathbb{1}_u, D^{1/2}\mathbb{1} \rangle^2}{2m} + \sum_{i=2}^n \lambda_i^{2t} \langle \mathbb{1}_u, e_i \rangle^2 = \frac{\deg(u)}{2m} + \sum_{i=2}^n \lambda_i^{2t} \langle \mathbb{1}_u, e_i \rangle^2 \tag{8}$$

where the second transition uses $\lambda_1 = 1$ and $e_1 = \frac{D^{1/2}\mathbb{1}}{\sqrt{2m}}$, and the last transition uses $\langle \mathbb{1}_u, D^{1/2}\mathbb{1}\rangle^2 = \deg(u)$. By Observation 15, $m = O(nd)$ and by the model assumption (Definition 6), $\deg(u) = \Omega(d)$. Since every term in Equation (8) is non-negative, this gives $\|\bar{A}^t\mathbb{1}_u\|_2^2 \geq \frac{\deg(u)}{2m} = \Omega\left(\frac{1}{n}\right)$.

We now prove that $\|\bar{A}^t\mathbb{1}_u\|_2^2 = O(1/n)$. By Lemma 16, for every $i \in [2, n-k+1]$, $|\lambda_i| \leq C'\sqrt{\lambda/d}$, for some constant C'. By the assumption that C is sufficiently large, we can assume that $t \geq \frac{C\log n}{\log(d/\lambda)} \geq \frac{\log n}{\log(\sqrt{d/\lambda}/C')}$. This gives $\lambda_i^{2t} \leq \left(C'\sqrt{\lambda/d}\right)^{2t} \leq n^{-2}$. Substituting this into Equation (8) gives

$$\begin{aligned}
\|\bar{A}^t\mathbb{1}_u\|_2^2 &= \frac{\deg(u)}{2m} + \sum_{i=2}^{n-k+1}\lambda_i^{2t}\langle \mathbb{1}_u, e_i\rangle^2 + \sum_{i=n-k+2}^{n}\lambda_i^{2t}\langle \mathbb{1}_u, e_i\rangle^2 \\
&\leq \frac{\deg(u)}{2m} + \frac{1}{n^2}\sum_{i=2}^{n-2}\langle \mathbb{1}_u, e_i\rangle^2 + \sum_{i=n-k+2}^{n}\lambda_i^{2t}\langle \mathbb{1}_u, e_i\rangle^2 \\
&\leq \frac{\deg(u)}{2m} + \frac{1}{n^2}\|\mathbb{1}_u\|_2^2 + \|f_u\|_2^2 \\
&= O\left(\frac{1}{n}\right).
\end{aligned}$$

Here, the last transition uses that $\|f_u\|_2^2 = O(1/n)$ by Lemma 21, since $u \in V \setminus B$, and that $\deg(u) \leq d$ and $m = \Omega(nd)$, by Observation 15.

The guarantee on $\|M^t\mathbb{1}_u\|_2^2$ follows from $M^t\mathbb{1}_u = D^{1/2}\bar{A}^tD^{-1/2}\mathbb{1}_u$, together with the assumption that $\deg(v) = \Theta(d)$ for all $v \in V$ (see Observation 15). ◀

We are now ready to prove the correctness of SpectralDotProduct(Lemma 10).

Proof of Lemma 10. Let C be a sufficiently large constant, let $\delta > 0$ be a sufficiently small constant. Let $t_{\min} \coloneqq \frac{10C\log n}{\log(d/\lambda)}$, and let $p(x) = x^{t_{\min}}q(x)$ be the polynomial from Theorem 4 applied with parameters δ, C and $t = t_{\min}$. Let $t_\Delta = \deg q = O\left(\sqrt{\lambda/d}\log n\right)$. For a sufficiently large constant κ, let

$$r = \kappa\sqrt{n}\cdot(\max|\mathrm{coeff}(p)|)^4(t_\Delta+1)^6. \tag{9}$$

By Theorem 4, $\max|\mathrm{coeff}(p)| = n^{O\left(\frac{1}{\log(d/\lambda)}\right)}$, so $r = \widetilde{O}\left(\sqrt{n}\cdot n^{O\left(\frac{1}{\log(d/\lambda)}\right)}\right)$.

We now show that with probability at least 0.9, the quantity computed by the algorithm approximates $\sum_{t,t'} c_t, c_{t'}\langle \bar{A}_\star\mathbb{1}_u, \bar{A}_\star\mathbb{1}_v\rangle = \langle p(\bar{A}_\star)\mathbb{1}_u, p(\bar{A}_\star)\mathbb{1}_v\rangle$ to a high precision.

▷ Claim 27. With probability at least 0.9, it holds that

$$\left|\sqrt{\deg(u)}\sqrt{\deg(v)}\sum_{t,t'}c_tc_{t'}\left(\left\langle D^{-1/2}\hat{p}_u^t, D^{-1/2}\hat{p}_v^{t'}\right\rangle - \frac{1}{2m}\right) - \langle p(\bar{A}_\star)\mathbb{1}_u, p(\bar{A}_\star)\mathbb{1}_v\rangle\right| \leq \frac{1}{160}\frac{k}{n}.$$

Proof. Fix $t, t' \in [t_{\min}, t_{\min}+t_\Delta]$. Set $p = M^t\mathbb{1}_u$, $q = M^{t'}\mathbb{1}_v$ and $W = D^{-1}$ in Lemma 25. Then $p^\top Wq = \langle D^{-1/2}M^t\mathbb{1}_u, D^{-1/2}M^{t'}\mathbb{1}_v\rangle$ and $\|p\|_2^2 + \|q\|_2^2 = \|M^t\mathbb{1}_u\|_2^2 + \|M^{t'}\mathbb{1}_v\|_2^2 = \Theta(1/n)$, where the last equality follows by Lemma 26. Therefore, applying Lemma 25 with $w_{\max} = O(1/d)$, $\zeta = \frac{0.1}{(t_\Delta+1)^2}$, $\xi = \frac{\gamma}{c_tc_{t'}(t_\Delta+1)^2}$ for a sufficiently small constant γ, and r as in Equation (9), we get that with probability at least $1 - \frac{0.1}{(t_\Delta+1)^2}$, it holds that

$$\left|\left\langle D^{-1/2}\hat{p}_u^t, D^{-1/2}\hat{p}_v^{t'}\right\rangle - \langle D^{-1/2}M^t\mathbb{1}_u, D^{-1/2}M^{t'}\mathbb{1}_v\rangle\right| \leq \frac{1}{160|c_t||c_{t'}|(t_\Delta+1)^2}\cdot\frac{k}{dn}$$

By taking a union bound over the $(t_\Delta+1)^2$ possible pairs of $t,t' \in [t_{\min}, t_{\min}+t_\Delta]$, we get

$$\Pr\left[\left|\sum_{t,t'} c_t c_{t'}\left(\left\langle D^{-1/2}\hat{p}_u^t, D^{-1/2}\hat{p}_v^{t'}\right\rangle - \langle D^{-1/2}M^t\mathbb{1}_u, D^{-1/2}M^{t'}\mathbb{1}_v\rangle\right)\right| \le \frac{k}{160dn}\right] \ge 0.9. \quad (10)$$

From Lemma 24, we have

$$\begin{aligned}
&\sum_{t,t'} c_t c_{t'}\left(\left\langle D^{-1/2}\hat{p}_u^t, D^{-1/2}\hat{p}_v^{t'}\right\rangle - \langle D^{-1/2}M^t\mathbb{1}_u, D^{-1/2}M^{t'}\mathbb{1}_v\rangle\right)\\
=&\sum_{t,t'} c_t c_{t'}\left(\left\langle D^{-1/2}\hat{p}_u^t, D^{-1/2}\hat{p}_v^{t'}\right\rangle - \frac{1}{2m}\right) - \sum_{t,t'} c_t c_{t'}\left(\langle D^{-1/2}M^t\mathbb{1}_u, D^{-1/2}M^{t'}\mathbb{1}_v\rangle - \frac{1}{2m}\right)\\
=&\sum_{t,t'} c_t c_{t'}\left(\left\langle D^{-1/2}\hat{p}_u^t, D^{-1/2}\hat{p}_v^{t'}\right\rangle - \frac{1}{2m}\right) - \frac{1}{\sqrt{\deg(u)}\sqrt{\deg(v)}}\sum_{t,t'} c_t c_{t'}\langle \bar{A}_\star^t\mathbb{1}_u, \bar{A}_\star^{t'}\mathbb{1}_v\rangle\\
=&\sum_{t,t'} c_t c_{t'}\left(\left\langle D^{-1/2}\hat{p}_u^t, D^{-1/2}\hat{p}_v^{t'}\right\rangle - \frac{1}{2m}\right) - \frac{1}{\sqrt{\deg(u)}\sqrt{\deg(v)}}\langle p(\bar{A}_\star)\mathbb{1}_u, p(\bar{A}_\star)\mathbb{1}_v\rangle.
\end{aligned}$$

Combining with Equation (10) gives the claim. ◁

Finally, by Lemma 23, since $u,v \in V\setminus B$, we have

$$\left|\langle p(\bar{A}_\star)\mathbb{1}_u, p(\bar{A}_\star)\mathbb{1}_v\rangle - \langle f_u, f_v\rangle\right| \le \frac{1}{160}\frac{k}{n}. \quad (11)$$

Combining Claim 27 and Equation (11) gives

$$\Pr\left[\left|\textsc{SpectralDotProduct}(u,v) - \langle f_u, f_v\rangle\right| \le \frac{1}{80}\frac{k}{n}\right] \ge 0.9.$$

Running time. The running time of SpectralDotProduct is dominated by the $2(t_\Delta+1) = \widetilde{O}(1)$ calls to RandomWalks (Algorithm 1) in lines 5-6, and the $(t_\Delta+1)^2 = \widetilde{O}(1)$ inner product computations in line 8. Each call to RandomWalks$(r,t,\cdot)$ takes time $O(t\cdot r) = \widetilde{O}(r)$ and returns a vector $\hat{p}_u^t$ of support at most r. Then, each of the inner product computations $\langle D^{-1/2}\hat{p}_u^t, D^{-1/2}\hat{p}_v^{t'}\rangle$ in line 8 can be performed in time

$$O\left(\min\left\{|\mathrm{supp}(D^{-1/2}\hat{p}_u^t)|, |\mathrm{supp}(D^{-1/2}\hat{p}_v^{t'})|\right\}\right) = O(r)$$

by computing $\sum_{w\in\mathrm{supp}(\hat{p}_u^t)\cap\mathrm{supp}(\hat{p}_v^{t'})}\hat{p}_u^t(w)\hat{p}_v^{t'}(w)/\deg(w)$. So the overall running time is $\widetilde{O}(r) = \widetilde{O}\left(n^{1/2+O\left(\frac{1}{\log(d/\lambda)}\right)}\right)$. ◀

5.3 Analysis of SpectralDistance (Proof of Lemma 11 and Corollary 12)

In this section, we prove Lemma 11 and Corollary 12. We start by proving Lemma 11.

Proof of Lemma 11. Let $i_{\max} = 5\cdot 10^6 \log n$ denote the number of iterations in SpectralDistance (Algorithm 3). Let $u,v \in V\setminus B$. For every $i \in [1, i_{\max}]$, each of the calls to SpectralDotProduct in lines 2, 3, 4 succeeds with probability at least 0.9. Therefore, by Lemma 10 and the union bound, with probability at least 0.7, it holds that

$$\begin{aligned}
&\left|\|f_u - f_v\|^2_{\mathrm{apx},i} - \|f_u - f_v\|_2^2\right| \\
&\quad = \left|\left(\|f_u\|^2_{\mathrm{apx},i} + \|f_v\|^2_{\mathrm{apx},i} - 2\langle f_u, f_v\rangle_{\mathrm{apx},i}\right) - \left(\|f_u\|_2^2 + \|f_v\|_2^2 - 2\langle f_u, f_v\rangle\right)\right| \\
&\quad \le \left|\|f_u\|^2_{\mathrm{apx},i} - \|f_u\|_2^2\right| + \left|\|f_v\|^2_{\mathrm{apx},i} - \|f_v\|_2^2\right| + 2\left|\langle f_u, f_v\rangle_{\mathrm{apx},i} - \langle f_u, f_v\rangle\right| \\
&\quad \le \frac{k}{80n} + \frac{k}{80n} + \frac{2k}{80n} = \frac{k}{20n}.
\end{aligned}$$

Therefore, by a Chernoff bound, for each pair $u, v \in V \setminus B$,

$$\Pr\left[\left|\underset{i\in[1,i_{\max}]}{\mathrm{median}}\left(\|f_u - f_v\|^2_{\mathrm{apx},i}\right) - \|f_u - f_v\|_2^2\right| \ge \frac{k}{20n}\right] \le \exp(-0.00025 i_{\max}) \le n^{-102},$$

where the last inequality follows from the setting $i_{\max} = 5 \cdot 10^6 \log n$. Taking a union bound over all possible pairs of $u, v \in V \setminus B$ (of which there are at most n^2) gives the required guarantee.

Running time. By Lemma 10, each call to SpectralDotProduct takes time $\widetilde{O}\left(n^{1/2+O\left(\frac{1}{\log(d/\lambda)}\right)}\right)$. Since the total number of calls to SpectralDotProduct is $O(\log n) = \widetilde{O}(1)$, we get the claimed running time. ◀

Next, we prove Corollary 12.

Proof of Corollary 12. Condition on the event that for all $u, v \in V \setminus B$,

$$\left|\textsc{SpectralDistance}(u, v) - \|f_u - f_v\|_2^2\right| \le \frac{k}{20n}. \tag{12}$$

By Lemma 11, this happens with probability at least $1 - n^{-100}$. Suppose that $\chi(u) = \chi(v) = i$. Then by the triangle inequality and Definition 19,

$$\|f_u - f_v\|_2 \le \|f_u - \mu_i\|_2 + \|f_v - \mu_i\|_2 \le 2 \cdot \sqrt{\frac{0.05k}{n}} = \sqrt{\frac{0.2k}{n}}. \tag{13}$$

By Equations (12) and (13), we get

$$\begin{aligned}
&\textsc{SpectralDistance}(u, v) \le \left|\textsc{SpectralDistance}(u, v) - \|f_u - f_v\|_2^2\right| + \|f_u - f_v\|_2^2 \\
&\le \frac{0.05k}{n} + \frac{0.2k}{n} = \frac{k}{4n}.
\end{aligned}$$

Suppose instead that $\chi(u) = i \neq \chi(v)$. Then by the triangle inequality and Definition 19,

$$\|f_u - f_v\|_2 \ge \|f_v - \mu_i\|_2 - \|f_u - \mu_i\|_2 \ge \sqrt{\frac{k}{n}} - \sqrt{\frac{0.05k}{n}} \ge \sqrt{\frac{0.56k}{n}}. \tag{14}$$

By Equation (12) and (14) we get

$$\begin{aligned}
&\textsc{SpectralDistance}(u, v) \ge \|f_u - f_v\|_2^2 - \left|\textsc{SpectralDistance}(u, v) - \|f_u - f_v\|_2^2\right| \\
&\ge \frac{0.56k}{n} - \frac{0.05k}{n} > \frac{k}{2n}.
\end{aligned}$$

◀

5.4 Construction of the polynomial (Proof of Theorem 4)

In this section, we prove Theorem 4. More generally, we prove:

▶ **Theorem 28.** *Let $\alpha \in [-1,1]$ be a non-zero constant (independent of n), and let $\delta > 0$. For every $\epsilon > 0$ smaller than a sufficiently small constant (depending on $|\alpha|$ and δ), and every $\frac{10\log n}{\log(1/\epsilon)} \le t \le O\left(\frac{\log n}{\log(1/\epsilon)}\right)$, there exists a polynomial p of the form*

$$p(x) = x^t q(x)$$

where q is a polynomial of degree $O(\epsilon \log n)$ such that the coefficients of p are bounded in absolute value by $n^{O_{\alpha,\delta}\left(\frac{1}{\log(1/\epsilon)}\right)}$ and

1. *$|p(x) - 1| \le \delta$ for $x \in [\alpha - \epsilon, \alpha + \epsilon]$*
2. *$|p(x)| \le n^{-2}$ for $|x| \le \epsilon$.*

As a corollary of Theorem 28, we obtain Theorem 4.

Proof of Theorem 4. Immediate from Theorem 28 by setting $\alpha = \frac{-1}{(k-1)}$ and $\epsilon = C\sqrt{\lambda/d}$. ◀

For the rest of the section, we prove Theorem 28. Our polynomial builds on the polynomial construction from [22], and we use the following two results.

▶ **Theorem 29** (Theorem A.2 in [22])**.** *Let $t \ge 2$, $\epsilon \in (0, 0.1]$, $d \ge 10\epsilon t$, and let*

$$P(x) = \sum_{v=0}^{d} T_v(x) c_{v,\epsilon,t}$$

be the polynomial obtained by taking the first $d+1$ terms of the Chebyshev expansion of $(1-\epsilon x)^{-t}$ on $[-1,1]$, where T_v denotes the polynomial given by $T_v(\cos\theta) = \cos(v\theta)$. Then

- *$\max_{v \le d} |c_{v,\epsilon,t}| \le 2^{O(\epsilon t)}$, and*
- *$\sup_{x\in[-1,1]} |P(x) - (1-\epsilon x)^{-t}| = O(de^{-d})$.*

▷ Claim 30 (Claim A.3 in [22]). There exists an absolute constant C such that for every $\epsilon, t > 0$ and every $d \ge 10\epsilon t$, the maximum coefficient of the polynomial $P\left(\frac{1-x}{\epsilon}\right)$ is bounded in absolute value by $(1/\epsilon)^{C\cdot d}$, where $P(y)$ denotes the degree d polynomial from Theorem 29.

Proof of Theorem 28. The main idea is to split the proof into the two cases $\epsilon \lesssim |\alpha|/\log n$, in which case the polynomial $p(x) = (x/\alpha)^t$ is already enough, and the case $\epsilon \gtrsim |\alpha|/\log n$, in which case we will use the polynomial from Theorem 29.

To keep the coefficients small enough in the first case, we choose $t \approx \frac{\log n}{\log(1/\epsilon)}$ rather than $t \approx \log n$, as in [22]. We then analyze the second case more carefully for this choice of t, instead of relying directly on the analysis in [22].

First, we need to set up some parameters. Let C_{29} denote the hidden universal constant in Theorem 29 such that $\sup_{x\in[-1,1]} |P(x) - (1-\epsilon x)^{-t}| \le C_{29} de^{-d}$ for all ϵ. Suppose that ϵ satisfies

$$\log(1/\epsilon) \ge \frac{\log(C_{29}/\delta)}{\delta} \tag{15}$$

(this holds by the assumption that ϵ is smaller than a sufficiently small constant). Let $\tau \ge 10$ be the constant such that

$$t = \frac{\tau \log n}{\log(1/\epsilon)}.$$

Additionally, define the constant

$$\kappa := \frac{\log(C_{29}/\delta)}{4\tau} \leq \frac{\log(C_{29}/\delta)}{40}. \tag{16}$$

We now split the proof into the two cases $\epsilon \leq \frac{\kappa|\alpha|}{\log n}$ or $\epsilon > \frac{\kappa|\alpha|}{\log n}$.

Case 1. Suppose $\epsilon \leq \frac{\kappa|\alpha|}{\log n}$. Let $p(x) = \frac{x^t}{\alpha^t}$. Then clearly p is of the right form, and the only non-zero coefficient of p is bounded in absolute value by

$$\left(\frac{1}{|\alpha|}\right)^t = \left(\frac{1}{|\alpha|}\right)^{\frac{\tau \log n}{\log(1/\epsilon)}} = n^{\frac{\tau \log(1/\alpha)}{\log(1/\epsilon)}} = n^{O_\alpha\left(\frac{1}{\log(1/\epsilon)}\right)},$$

by the assumption that $t = \frac{\tau \log n}{\log(1/\epsilon)}$ for some constant τ.

We now verify Condition 1. By the assumption $\epsilon \leq \frac{\kappa|\alpha|}{\log n}$, we have

$$\begin{aligned}
\frac{4\epsilon t}{|\alpha|} &= \frac{4\tau\epsilon \log n}{|\alpha| \log(1/\epsilon)} && \text{by the definition of } \tau \\
&\leq \frac{4\tau\kappa}{\log(1/\epsilon)} && \text{by the assumption } \epsilon \leq \frac{\kappa|\alpha|}{\log n} \\
&\leq \delta && \text{by Equation (16) and Equation (15).}
\end{aligned}$$

Therefore,

$$\max_{x \in [\alpha-\epsilon, \alpha+\epsilon]} |p(x) - 1| = \max_{x \in [\alpha-\epsilon, \alpha+\epsilon]} \left| \left(\frac{x}{\alpha}\right)^t - 1 \right| \leq \exp\left(\frac{2t\epsilon}{|\alpha|}\right) - 1 \leq \frac{4t\epsilon}{|\alpha|} \leq \delta.$$

Next, we verify Condition 2. We have

$$\max_{x \in [-\epsilon, \epsilon]} |p(x)| = \max_{x \in [-\epsilon, \epsilon]} \left| \left(\frac{x}{\alpha}\right)^t \right| = \left(\frac{\epsilon}{|\alpha|}\right)^t \leq \exp\left(-t \log(|\alpha|/\epsilon\right) \leq n^{-2},$$

where the third transition uses the assumption $t \geq \frac{10 \log n}{\log(1/\epsilon)}$, and the last transition holds for n sufficiently large, using the assumption α is a constant independent of n.

Case 2. Suppose $\epsilon \geq \frac{\kappa|\alpha|}{\log n}$. Let $P(x)$ be the polynomial from Theorem 29 applied with error parameter $\epsilon/|\alpha|$ and $d = \frac{10 \log(1/\epsilon)\epsilon t}{|\alpha|} = \frac{10\tau\epsilon \log n}{|\alpha|} = O_\alpha(\epsilon \log n)$, and let $p(x) := \left(\frac{x}{\alpha}\right)^t P\left(\frac{1 - \frac{x}{\alpha}}{\frac{\epsilon}{|\alpha|}}\right)$. We start by bounding the coefficients of $p(x)$. By Claim 30 and the setting of d, the coefficients of the polynomial $q(x) := x^t P\left(\frac{1-x}{\frac{\epsilon}{|\alpha|}}\right)$ are bounded in absolute value by $(|\alpha|/\epsilon)^{C_{30} \cdot d} = n^{O_\alpha(\epsilon \log(1/\epsilon))} = n^{O_\alpha(1/\log(1/\epsilon))}$, where C_{30} denotes the universal constant from Claim 30. Since $p(x) = q(x/\alpha)$, the coefficients of p are bounded in absolute value by

$$\begin{aligned}
\max |\text{coeff}(p)| &\leq \left(\frac{1}{|\alpha|}\right)^{\deg p} \cdot \max |\text{coeff}(q)| = \left(\frac{1}{|\alpha|}\right)^{t+d} n^{O_\alpha\left(\frac{1}{\log(1/\epsilon)}\right)} \\
&\leq \left(\frac{1}{|\alpha|}\right)^{\frac{\tau \log n}{\log(1/\epsilon)} + \frac{10\tau\epsilon \log n}{|\alpha|}} n^{O_\alpha\left(\frac{1}{\log(1/\epsilon)}\right)} = n^{O_\alpha\left(\frac{1}{\log(1/\epsilon)}\right)},
\end{aligned}$$

as required. We now prove that Condition 1 holds. We have

$$\begin{aligned}
&\max_{x\in[\alpha-\epsilon,\alpha+\epsilon]} |p(x)-1| \\
&= \max_{x\in[\alpha-\epsilon,\alpha+\epsilon]} \left|\left(\frac{x}{\alpha}\right)^t P\left(\frac{1-x/\alpha}{\epsilon/|\alpha|}\right) - 1\right| \\
&\le \left(1+\frac{\epsilon}{|\alpha|}\right)^t \max_{x\in[\alpha-\epsilon,\alpha+\epsilon]} \left|P\left(\frac{1-x/\alpha}{\epsilon/|\alpha|}\right) - \left(\frac{x}{\alpha}\right)^{-t}\right| \\
&\le C_{29}\cdot\left(1+\frac{\epsilon}{|\alpha|}\right)^t de^{-d} && \text{by Theorem 29} \\
&\le C_{29}\cdot\exp\left(\frac{\epsilon t}{|\alpha|}\right)\exp\left(-\frac{d}{2}\right) && \text{since } x\le e^{x/2} \text{ and } 1+x\le e^x \\
&= C_{29}\cdot\exp\left(\frac{\epsilon t}{|\alpha|} - 5\log(1/\epsilon)\frac{\epsilon}{|\alpha|}t\right) && \text{by the setting } d = 10\log(1/\epsilon)\epsilon t/|\alpha| \\
&\le C_{29}\cdot\exp\left(-4\log(1/\epsilon)\frac{\epsilon}{|\alpha|}t\right) \\
&\le C_{29}\exp\left(-4\kappa\tau\right) && \text{by } t=\frac{\tau\log n}{\log(1/\epsilon)} \text{ and } \epsilon\ge\frac{\kappa|\alpha|}{\log n} \\
&= \delta && \text{by setting } \kappa \text{ (16).}
\end{aligned}$$

We now prove Condition 2. Since p has at most $(d+1)$ non-zero coefficients, we have

$$\begin{aligned}
&\max_{x\in[-\epsilon,\epsilon]} |p(x)| \le \epsilon^t\cdot(d+1)\cdot\max|\mathrm{coeff}(p)| \\
&\le \epsilon^{10\log n/\log(1/\epsilon)}\cdot O\left(\epsilon\log n/|\alpha|\right)\cdot n^{O_\alpha(1/\log(1/\epsilon))} \le n^{-10}\cdot n\cdot n^{O_\alpha(1/\log(1/\epsilon))} \le n^{-2},
\end{aligned}$$

where the last inequality uses that ϵ is smaller than a sufficiently small constant depending on α, so we can assume that the value of the exponent $O_\alpha(1/\log(1/\epsilon))$ is at most 7. ◀

5.5 Putting everything together (Proof of Theorem 1)

We now combine Theorem 4, Lemma 11, Corollary 12 and Lemma 20 to obtain Theorem 1.

Proof of Theorem 1. The oracle runs the preprocessing algorithm FindRepresentatives (Algorithm 5) to obtain a set of k color representatives $\{c_i\}_{i\in[k]}$. Then, for every query u, the oracle returns $\textsc{Color}(u,\{c_i\}_{i\in[k]})$.

Let B be the set of bad vertices from Definition 19. We now show that with probability at least 0.9, the oracle is correct on all vertices in $V\setminus B$.

By Lemma 14, with probability at least 0.99, the output of FindRepresentatives is a valid set of color representatives (as per Definition 13). Claim 31 below shows that conditioned on the success of FindRepresentatives, the oracle $\textsc{Color}(v,\{c_i\}_{i\in[k]})$ returns the correct color (up to a permutation of colors) for every $v\in V\setminus B$:

▷ Claim 31. Suppose that $\{c_i\}_{i\in[k]}$ is a set of color representatives (as per Definition 13). Let π be the permutation such that $\pi(\chi(c_i))=i$ for every $i\in[k]$. Then with probability at least $1-n^{-100}$, for all $v\in V\setminus B$,

$$\textsc{Color}(v,\{c_i\}_{i\in[k]}) = \pi(\chi(v)).$$

Proof. Let $v \in V \setminus B$, and let c_i be the color representative such that $\chi(c_i) = \chi(v)$. Since $\{c_j\}_{j\in[k]}$ is a set of color representatives, by Definition 13, every c_j belongs to a different color class, and $c_j \in V \setminus B$ for all $j \in [k]$. Therefore, by Corollary 12,

$$\textsc{SpectralDistance}(v, c_i) \leq \frac{k}{4n}$$

and for all $j \neq i$,

$$\textsc{SpectralDistance}(v, c_j) \geq \frac{k}{2n}.$$

In particular, $\arg\min_j \textsc{SpectralDistance}(v, c_j) = i$. By definition of the permutation π, we have $i = \pi(\chi(c_i)) = \pi(\chi(v))$, so Color returns $\pi(\chi(v))$ as claimed. ◁

Taking a union bound over the success of FindRepresentatives and Claim 31, we get that with probability at least $0.99 - n^{-100} \geq 0.9$, every vertex in $V \setminus B$ is colored correctly (up to a permutation of the colors). In that case, by Lemma 20,

$$|\{v : \textsc{Color}(v, \{c_i\}_{i\in[k]}) \neq \pi(\chi(v))\}| \leq |B| \leq O\left(\sqrt{\frac{\lambda}{d}} \cdot n\right).$$

Running time. The preprocessing stage runs FindRepresentatives (Algorithm 5). By Lemma 14, this has running time $\widetilde{O}\big(n^{1/2+O(1/\log(d/\lambda))}\big)$.

Each query is answered by Color (Algorithm 4), which calls SpectralDistance (Algorithm 3) $k = O(1)$ times. By Lemma 11, this takes time $\widetilde{O}\big(n^{1/2+O(1/\log(d/\lambda))}\big)$. ◀

References

1. Noga Alon and Nabil Kahalé. A spectral technique for coloring random 3-colorable graphs. *SIAM J. Comput.*, 26(6):1733–1748, 1997. doi:10.1137/S0097539794270248.
2. Sanjeev Arora and Eden Chlamtac. New approximation guarantee for chromatic number. In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 215–224. ACM, 2006. doi:10.1145/1132516.1132548.
3. Sanjeev Arora and Rong Ge. New tools for graph coloring. In Leslie Ann Goldberg, Klaus Jansen, R. Ravi, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 14th International Workshop, APPROX 2011, and 15th International Workshop, RANDOM 2011, Princeton, NJ, USA, August 17-19, 2011. Proceedings*, volume 6845 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2011. doi:10.1007/978-3-642-22935-0_1.
4. Mitali Bafna, Jun-Ting Hsieh, and Pravesh K. Kothari. Rounding large independent sets on expanders. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 631–642, New York, NY, USA, 2025. Association for Computing Machinery. doi:10.1145/3717823.3718137.
5. Bonnie Berger and John Rompel. A better performance guarantee for approximate graph coloring. *Algorithmica*, 5(3):459–466, 1990. doi:10.1007/BF01840398.
6. A. Blum and J. Spencer. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms*, 19(2):204–234, 1995. doi:10.1006/jagm.1995.1034.
7. Avrim Blum. New approximation algorithms for graph coloring. *J. ACM*, 41(3):470–516, 1994. doi:10.1145/176584.176586.
8. Avrim Blum and David R. Karger. An õ(n^{3/14})-coloring algorithm for 3-colorable graphs. *Inf. Process. Lett.*, 61(1):49–53, 1997. doi:10.1016/S0020-0190(96)00190-1.

9 Rares-Darius Buhai, Yiding Hua, David Steurer, and Andor Vári-Kakas. Finding colorings in one-sided expanders. *CoRR*, abs/2508.02825, 2025. doi:10.48550/arXiv.2508.02825.

10 Ashish Chiplunkar, Michael Kapralov, Sanjeev Khanna, Aida Mousavifar, and Yuval Peres. Testing graph clusterability: Algorithms and lower bounds. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 497–508. IEEE, 2018. doi:10.1109/FOCS.2018.00054.

11 Artur Czumaj, Pan Peng, and Christian Sohler. Testing cluster structure of graphs. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 723–732. ACM, 2015. doi:10.1145/2746539.2746618.

12 Roee David and Uriel Feige. On the effect of randomness on planted 3-coloring models. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 77–90, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2897518.2897561.

13 Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 344–353, New York, NY, USA, 2006. Association for Computing Machinery. doi:10.1145/1132516.1132567.

14 Uriel Feige and Eran Ofek. Spectral techniques applied to sparse random graphs. *Random Struct. Algorithms*, 27(2):251–275, September 2005. doi:10.1002/RSA.20089.

15 Hendrik Fichtenberger, Michael Kapralov, Ekaterina Kochetkova, Silvio Lattanzi, Davide Mazzali, and Weronika Wrzos-Kaminska. Spectral clustering with side information. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4327–4341, 2026. doi:10.1137/1.9781611978971.159.

16 J. Friedman, J. Kahn, and E. Szemerédi. On the second eigenvalue of random regular graphs. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, pages 587–598, New York, NY, USA, 1989. Association for Computing Machinery. doi:10.1145/73007.73063.

17 Z. Füredi and J. Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981. doi:10.1007/BF02579329.

18 M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified np-complete graph problems. *Theor. Comput. Sci.*, 1(3):237–267, 1976. doi:10.1016/0304-3975(76)90059-1.

19 Grzegorz Gluch, Michael Kapralov, Silvio Lattanzi, Aida Mousavifar, and Christian Sohler. Spectral clustering oracles in sublinear time. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1598–1617. SIAM, 2021. doi:10.1137/1.9781611976465.97.

20 Jun-Ting Hsieh. Coloring 3-colorable graphs with low threshold rank. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 718–726, 2026. doi:10.1137/1.9781611978971.27.

21 Agastya Vibhuti Jha and Akash Kumar. A sublinear time tester for max-cut on clusterable graphs. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, Tallinn, Estonia, July 8-12, 2024*, volume 297 of *LIPIcs*, pages 91:1–91:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ICALP.2024.91.

22 Michael Kapralov, Ekaterina Kochetkova, and Weronika Wrzos-Kaminska. Spectral clustering in birthday paradox time. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3057–3071, 2026. doi:10.1137/1.9781611978971.113.

23 David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, 1998. doi:10.1145/274787.274791.

24 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.

25 Ken-ichi Kawarabayashi and Mikkel Thorup. Coloring 3-colorable graphs with less than $n^{1/5}$ colors. *J. ACM*, 64(1):4:1–4:23, 2017. doi:10.1145/3001582.

26 Ken-ichi Kawarabayashi, Mikkel Thorup, and Hirotaka Yoneda. Better coloring of 3-colorable graphs. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 331–339. ACM, 2024. doi:10.1145/3618260.3649768.

27 Sanjeev Khanna, Nathan Linial, and Shmuel Safra. On the hardness of approximating the chromatic number. *Comb.*, 20(3):393–415, 2000. doi:10.1007/S004930070013.

28 S. Khot. Improved inapproximability results for maxclique, chromatic number and approximate graph coloring. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, FOCS '01, page 600, USA, 2001. IEEE Computer Society.

29 Akash Kumar, Anand Louis, and Madhur Tulsiani. Finding pseudorandom colorings of pseudorandom graphs. In Satya V. Lokam and R. Ramanujam, editors, *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2017, Kanpur, India, December 11-15, 2017*, volume 93 of *LIPIcs*, pages 37:1–37:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.FSTTCS.2017.37.

30 A. Nilli. On the second eigenvalue of a graph. *Discrete Math.*, 91(2):207–210, August 1991. doi:10.1016/0012-365X(91)90112-F.

31 Pan Peng and Yuichi Yoshida. Sublinear-time algorithms for max cut, max e2lin (q), and unique label cover on expanders. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4936–4965. SIAM, 2023. doi:10.1137/1.9781611977554.CH180.

32 Ranran Shen and Pan Peng. A sublinear-time spectral clustering oracle with improved preprocessing time. *Advances in Neural Information Processing Systems*, 36:41809–41828, 2023.

33 Avi Wigderson. Improving the performance guarantee for approximate graph coloring. *J. ACM*, 30(4):729–735, 1983. doi:10.1145/2157.2158.