Planted Models for the Densest k-Subgraph Problem

Yash Khanna
Indian Institute of Science, Bangalore, India
yashkhanna@iisc.ac.in

Anand Louis
Indian Institute of Science, Bangalore, India
anandl@iisc.ac.in

Abstract

Given an undirected graph G, the Densest k-subgraph problem (DkS) asks to compute a set $S \subset V$ of cardinality $|S| \leq k$ such that the weight of edges inside S is maximized. This is a fundamental NP-hard problem whose approximability, inspite of many decades of research, is yet to be settled. The current best known approximation algorithm due to Bhaskara et al. (2010) computes a $\mathcal{O}\left(n^{1/4+\epsilon}\right)$ approximation in time $n^{\mathcal{O}(1/\epsilon)}$, for any $\epsilon > 0$.

We ask what are some "easier" instances of this problem? We propose some natural semi-random models of instances with a planted dense subgraph, and study approximation algorithms for computing the densest subgraph in them. These models are inspired by the semi-random models of instances studied for various other graph problems such as the independent set problem, graph partitioning problems etc. For a large range of parameters of these models, we get significantly better approximation factors for the Densest k-subgraph problem. Moreover, our algorithm recovers a large part of the planted solution.

2012 ACM Subject Classification Theory of computation → Semidefinite programming; Theory of computation → Discrete optimization; Theory of computation → Graph algorithms analysis

Keywords and phrases Densest k-Subgraph, Semi-Random models, Planted Models, Semidefinite Programming, Approximation Algorithms, Beyond Worst Case Analysis

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2020.27

Related Version A full version of the paper is available at `https://arxiv.org/abs/2004.13978`.

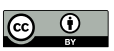
Funding *Anand Louis*: AL was supported in part by SERB Award ECR/2017/003296 and a Pratiksha Trust Young Investigator Award.

Acknowledgements We thank Rakesh Venkat for helpful discussions. We also thank the anonymous reviewers for their suggestions and comments on earlier versions of this paper.

1 Introduction

Given a weighted undirected graph $G = (V, E, w)$ with non-negative edge weights given by $w : E \to \mathbb{R}^+$, and an integer $k \in \mathbb{Z}^+$, the Densest k-subgraph problem (DkS) asks to compute a set $S \subset V$ of cardinality $|S| \leq k$ such that the weight of edges inside S (i.e., $\sum_{i,j \in S} w(\{i,j\})$) is maximized (if $\{i,j\} \notin E$, we assume w.l.o.g. that $w(\{i,j\}) = 0$). Computing the DkS of a graph is a fundamental NP-hard problem. There has been a lot of work on studying approximation algorithms for DkS, we give a brief survey in Section 1.3.

The current best known approximation algorithm [6] computes an $\mathcal{O}\left(n^{1/4+\epsilon}\right)$ approximation in time $n^{\mathcal{O}(1/\epsilon)}$ for any $\epsilon > 0$. On the hardness side, Manurangsi [31] showed that assuming the exponential time hypothesis (ETH), there is no polynomial time algorithm that approximates this to within $n^{1/(\log\log n)^c}$ factor where $c > 0$ is some fixed constant. There are hardness of approximation results known for this problem assuming various other



hardness assumptions, see Section 1.3 for a brief survey. But there is still a huge gap between the upper and lower bounds on the approximability of this problem.

Given this status of the approximability of the DENSEST k-SUBGRAPH problem, we ask what are some "easier" instances of this problem? We propose some natural semi-random models of instances with a planted dense subgraph, and study approximation algorithms for computing the densest subgraph in them. Studying semi-random models of instances has been a very fruitful direction of study towards understanding the complexity for various NP-hard problems such as graph partitioning problems [28, 29, 26, 27], independent sets [13, 32], graph coloring [1, 11, 12], etc. By studying algorithms for instances where some parts are chosen to be arbitrary and some parts are chosen to be random, one can understand which aspects of the problem make it computationally intractable. Besides being of natural theoretical interest, studying approximation algorithms for semi-random models of instances can also be practically useful since some natural semi-random models of instances can be better models of instances arising in practice than the worst-case instances. Therefore, designing algorithms specifically for such models can help to bridge the gap between theory and practice in the study of algorithms. Some random and semi-random models of instances of the DENSEST k-SUBGRAPH problem (and its many variants) have been studied in [2, 6, 8, 19, 20, 21, 33, 34], we discuss them in Section 1.3. Our models are primarily inspired by the densest subgraph models mentioned above as well as the semi-random models of instances for other problems [13, 32] studied in the literature. For a large range of parameters of these models, we get significantly better approximation factors for the DENSEST k-SUBGRAPH problem, and also show that we can recover a large part of the planted solution.

We note that semidefinite programming (SDP) based methods have been popularly used in many randomized models for different problems, including the DENSEST k-SUBGRAPH problem [19, 20, 21]. And thus, another motivation for our work is to understand the power of SDPs in approximating the DENSEST k-SUBGRAPH problem. Since even strong SDP relaxations of the problem have a large integrality gap [7] for worst case instances (see Section 1.3), we ask what families of instances can SDPs approximate well? In addition to being of theoretical interest, algorithms using the basic SDP also have a smaller running time. In comparison, the algorithm of [6] produces an $\mathcal{O}\left(n^{1/4+\epsilon}\right)$ approximation for worst-case instances in time $n^{\mathcal{O}(1/\epsilon)}$; their algorithm is based on rounding an LP hierarchy, but they also show that their algorithm can be executed without solving an LP and obtain the same guarantees.

1.1 Our models and results

The main inspiration for our models are the semi-random models of instances for the independent set problem [13, 32]. Their instances are constructed as follows. Starting with a set of vertices V, a subset of k vertices is chosen to form the independent set S, and edges are added between each pair in $S \times (V \setminus S)$ independently with probability p. Finally, an arbitrary graph is added on $V \setminus S$. They study the values of k and p for which they can recover a large independent set. Our models can be viewed as analogs of this model to the DENSEST k-SUBGRAPH problem: edges are added between each pair in $S \times (V \setminus S)$ independently with probability p, and then edges are added in S to form a dense subset. Since we also guarantee that we can recover a large part of the planted dense subgraph S, we also need to assume that the graph induced on $V \setminus S$ is "far" from containing a dense subgraph. We now define our models.

▶ **Definition 1.1** (DkSExp$(n, k, d, \delta, d', \lambda)$)**.** *An instance of DkSExp$(n, k, d, \delta, d', \lambda)$ is generated as follows,*

1. *We partition V into two sets, S and $V \setminus S$ with $|S| = k$. We add edges (of weight 1) between pairs in $S \times (V \setminus S)$ independently with probability $p \stackrel{\text{def}}{=} \delta d/k$.*
2. *We add edges of arbitrary non-negative weights between arbitrary pairs of vertices in S such that the graph induced on S has average weighted degree d.*
3. *We add edges of arbitrary non-negative weights between arbitrary pairs of vertices in $V \setminus S$ such that the graph induced on $V \setminus S$ is a (d', λ)-expander (see Definition 1.10 for definition).*
4. *(*Monotone adversary*) Arbitrarily delete any of the edges added in step 1 and step 3.*
5. *Output the resulting graph.*

We note that the step 2, step 3, and step 4 in the construction of the instance above are adversarial steps.

DkSExp$(n, k, d, \delta, d', \lambda)$ are a class of instances that have a prominent dense subset of size k. Note that, since the graph induced on $V \setminus S$ is a subset of an expander graph, it would not have any dense subsets. We also note that the monotone adversary can make significant changes to graph structure. For example, the graph induced on $V \setminus S$ can be neither d'-regular nor an expander after the action of the monotone adversary.

We require $\delta < 1$ in step 1 for the following reason. For any fixed set $S' \subset V \setminus S$ such that $|S'| = \mathcal{O}(k)$, the expected weight of edges in the bipartite graph induced on $S \cup S'$ is $\mathcal{O}(\delta k d)$. Since we want the graph induced on S to be the densest k-subgraph (the total of edges in the graph induced on S is $kd/2$), we restrict δ to be at most 1.

We present our main results below, note that our algorithm outputs a dense subgraph of size k and its performance is measured with respect to the density of the planted subgraph $G[S]$, i.e. $kd/2$.

▶ **Definition 1.2.** *We define $\rho(V') \stackrel{\text{def}}{=} \left(\sum_{i,j \in V'} w(\{i,j\})\right)/2$ for any $V' \subseteq V$.*

▶ **Theorem 1.3** (Informal version of Theorem 2.1)**.** *Given an instance of DkSExp$(n, k, d, \delta, d', \lambda)$ where*

$$\delta = \Theta\left(\frac{kd'}{nd}\right), \qquad \frac{\delta d}{k} = \Omega\left(\frac{\log n}{n}\right), \qquad \text{and} \qquad \nu = \Theta\left(\sqrt{\delta + \frac{\lambda + \sqrt{d'}}{d}}\right),$$

there exists a deterministic polynomial time algorithm that outputs with high probability (over the instance) a vertex set $\mathcal{Q}$ of size k such that $\rho(\mathcal{Q}) \geq (1 - \nu)\frac{kd}{2}$. The above algorithm also computes a vertex set T such that

(a) $|T| \leq (1 + \mathcal{O}(\nu))\, k$.

(b) $\rho(T \cap S) \geq (1 - \mathcal{O}(\nu))\frac{kd}{2}$.

▶ Remark 1.4. In Theorem 1.3, we restrict the range of δ for the following reason. An interesting setting of parameters is when the average degree of vertices in S and $V \setminus S$ are within constant factors of each other. Then the expected average degree of a vertex in S is $d + p(n-k)$. And for a vertex in $V \setminus S$, the expected average degree is $d' + kp$. Thus setting,

$$d + p(n-k) = \Theta(d' + kp) \implies \delta = \Theta\left(\frac{kd'}{nd}\right) \qquad \left(\text{Recall, } p = \frac{\delta d}{k}\right).$$

We also study another interesting model with a different assumption on the subgraph $G[V \setminus S]$.

▶ **Definition 1.5.** *$DkS(n,k,d,\delta,\gamma)$ is generated similarly to $DkSExp(n,k,d,\delta,d',\lambda)$ except in step 3, where we add edges between arbitrary pairs of vertices in $V \setminus S$ such that the graph induced on $V \setminus S$ has the following property : $\rho(V') \leq \gamma d\,|V'| \quad \forall V' \subseteq V \setminus S$.*

By construction, the graph induced on $V \setminus S$ does not have very dense subsets.

▶ **Theorem 1.6.** *Given an instance of $DkS(n,k,d,\delta,\gamma)$ where*

$$\delta = \Theta\left(\frac{k}{n}\right), \qquad \frac{\delta d}{k} = \Omega\left(\frac{\log n}{n}\right), \qquad \text{and} \qquad \tau = \Theta\left(\sqrt{\delta + \gamma + \frac{1}{\sqrt{d}}}\right),$$

there is a deterministic polynomial time algorithm that outputs with high probability (over the instance) a vertex set $\mathcal{Q}$ of size k such that $\rho(\mathcal{Q}) \geq (1-\tau)\frac{kd}{2}$. The above algorithm also computes a vertex set T such that

(a) $|T| \leq (1+\mathcal{O}(\tau))\,k$.

(b) $\rho(T \cap S) \geq (1-\mathcal{O}(\tau))\frac{kd}{2}$.

Other results

We also study two variants of $\mathrm{D}k\mathsf{SExp}(n,k,d,\delta,d',\lambda)$ and $\mathrm{D}k\mathrm{S}(n,k,d,\delta,\gamma)$ where the subgraph $G[S]$ is d-regular.

1. $\mathrm{D}k\mathsf{SExpReg}(n,k,d,\delta,d',\lambda)$ is same as $\mathrm{D}k\mathsf{SExp}(n,k,d,\delta,d',\lambda)$ except in step 2, which requires the subgraph $G[S]$ to be an arbitrary $d-$regular graph.
   ▶ **Theorem 1.7.** *Given an instance of $DkSExpReg(n,k,d,\delta,d',\lambda)$where*

$$\delta = \Theta\left(\frac{kd'}{nd}\right), \qquad \frac{\delta d}{k} = \Omega\left(\frac{\log n}{n}\right), \qquad \text{and} \qquad \nu' = \Theta\left(\frac{\sqrt{d'}}{d\left(1-\delta-\frac{\lambda}{d}\right)}\right),$$

   there is a deterministic polynomial time algorithm that outputs with high probability (over the instance) a vertex set $\mathcal{Q}$ of size k such that

   a. $\rho(\mathcal{Q}) \geq (1-\nu')\frac{kd}{2}$.

   b. $|\mathcal{Q} \cap S| \geq (1-\mathcal{O}(\nu'))\,k$.

2. $\mathrm{D}k\mathsf{SReg}(n,k,d,\delta,\gamma)$ is same as $\mathrm{D}k\mathrm{S}(n,k,d,\delta,\gamma)$ except in step 2, which requires the subgraph $G[S]$ to be an arbitrary $d-$regular graph.
   ▶ **Theorem 1.8.** *Given an instance of $DkSReg(n,k,d,\delta,\gamma)$where*

$$\delta = \Theta\left(\frac{k}{n}\right), \qquad \frac{\delta d}{k} = \Omega\left(\frac{\log n}{n}\right), \qquad \text{and} \qquad \tau' = \Theta\left(\frac{1}{\sqrt{d}\,(1-\gamma-\delta)}\right),$$

   there is a deterministic polynomial time algorithm that outputs with high probability (over the instance) a vertex set $\mathcal{Q}$ of size k such that

   a. $\rho(\mathcal{Q}) \geq (1-\tau')\frac{kd}{2}$.

   b. $|\mathcal{Q} \cap S| \geq (1-\mathcal{O}(\tau'))\,k$.

We will show that for most natural regime of parameters, we get a better approximation factors in the case when $G[S]$ is a d-regular graph.

▶ Remark 1.9. It has been pointed out to us by anonymous reviewers that for a large range of parameters of the DkS(n,k,d,δ,γ) and DkSReg(n,k,d,δ,γ) models, $\arg\max_{W\subseteq V}\rho(W)/|W|$ will be a subset of S; for any graph $G=(V,E)$, the algorithm due to Charikar [10] can be used to compute $\arg\max_{W\subseteq V}\rho(W)/|W|$ in polynomial time. It is plausible that using this algorithm iteratively, one can recover a "large" part of S. However the algorithm described in Theorem 1.6 and Theorem 1.8 gives a more direct approach to recover a large part of S.

1.2 Notation

We use $n \stackrel{\text{def}}{=} |V|$, and use V and $[n] \stackrel{\text{def}}{=} \{1,2,\ldots,n\}$ interchangeably. We assume w.l.o.g. that G is a complete graph: if $\{i,j\}\notin E$, we add $\{i,j\}$ to E and set $w(\{i,j\})=0$. We use A to denote the weighted adjacency matrix of G, i.e. $A_{ij}=w(\{i,j\})\ \forall i,j\in V$. The degree of vertex i is defined as $d_i \stackrel{\text{def}}{=} \sum_{j\in V} w(\{i,j\})$.

For $V'\subseteq V$, we use $G[V']$ to denote the subgraph induced on V' and $\overline{V'}$ to denote $V\setminus V'$. For a vector v, we use $\|v\|$ to denote the $\|v\|_2$. For a matrix A, we use $\|A\|$ to denote the spectral norm $\|A\| \stackrel{\text{def}}{=} \max_{x\neq 0}\frac{\|Ax\|}{\|x\|}$.

We define probability distributions μ over finite sets Ω. For a random variable (r.v.) $X:\Omega\to\mathbb{R}$, its expectation is denoted by $\mathbb{E}_{\omega\sim\mu}[X]$. In particular, we define the two distributions which we use below.

1. For a vertex set $V'\subseteq V$, we define a probability (uniform) distribution $(f_{V'})$ on the vertex set V' as follows. For a vertex $i\in V'$, $f_{V'}(i)=\frac{1}{|V'|}$. We use $i\sim V'$ to denote $i\sim f_{V'}$ for clarity.
2. For a vertex set $V'\subseteq V$, we define a probability distribution $(f_{E(G[V'])})$ on the edges of $G[V']$ as follows. For an edge $e\in E(G[V'])$, $f_{E(G[V'])}(e)=\frac{w(e)}{\rho(V')}$. Again, we use $e\sim E(G[V'])$ to denote $e\sim f_{E(G[V'])}$ for convenience.

▶ **Definition 1.10** ((d,λ)-expanders). *A graph $H=(V,E,w)$ is said to be a (d,λ)-expander if H is d-regular and $|\lambda_i|\leq\lambda$, $\forall i\in[n]\setminus\{1\}$, where $\lambda_1\geq\lambda_2\ldots\geq\lambda_n$ are the eigenvalues of the weighted adjacency matrix of H.*

1.3 Related Work

Densest k-subgraph. There has been a lot of work on the Densest k-subgraph problem and its variants. The current best known approximation algorithm, due to Bhaskara et al. [6], gives an approximation ratio of $\mathcal{O}(n^{1/4+\epsilon})$ in time $n^{\mathcal{O}(1/\epsilon)}$, for all values of $\epsilon>0$ (for $\epsilon=1/\log n$, we get a ratio of $\mathcal{O}\left(n^{1/4}\right)$). They also extend their approach to give a $\mathcal{O}(n^{1/4-\epsilon})$ approximation algorithm which runs in time $2^{n^{\mathcal{O}(\epsilon)}}$. They improved the prior results of Feige et al. [14] which gave a $n^{1/3-\epsilon}$ approximation for some small $\epsilon>0$. [14] also give a greedy algorithm which has an approximation factor of $\mathcal{O}(n/k)$.

When $k=\Theta(n)$, Asahiro et al. [3] gave a constant factor approximation algorithm. Many other works have looked at this problem using linear and semidefinite programming techniques. Srivastav et al. [37] gave a randomized rounding algorithm using a SDP relaxation in the case when $k=n/c$ for $c>1$, they improved the constants for certain values of k over the results of [3]. Feige and Langberg [15] use a different SDP to get an approximation of slightly above k/n for the case when k is roughly $n/2$. Feige and Seltser [16] construct examples for which their SDP has an integrality gap of $\Omega(n^{1/3})$.

There has been work done on a related problem called the maximum density subgraph, where the objective is to find a subgraph which maximizes the ratio of number of edges to the number of vertices. Goldberg [18] and Gallo et al. [17] had given an algorithm to solve this problem exactly using maximum flow techniques. Later, Charikar [10] gave an algorithm based on a linear programming method. This paper also solves the problem for directed graphs using a notion of density given by Kannan and Vinay [22]. Khuller and Saha [24] gave a max-flow based algorithm in the directed setting.

On the hardness side, Khot [23] showed that it does not have a PTAS unless NP has subexponential algorithms. There has been some works based on some other hardness assumptions. Assuming the small-set expansion hypothesis, Raghavendra and Steurer [35] show that it is NP-hard to approximate DkS to any constant factor. Under the deterministic ETH assumption, Braverman et al. [9] show that it requires $n^{\Omega(\log n)}$ time to approximate DkS with perfect completeness to within $1+\varepsilon$ factor (for a universal constant $\varepsilon > 0$). More recently Manurangsi [31] showed assuming the exponential time hypothesis (ETH), that there is no polynomial time algorithm that approximates this to within $n^{1/(\log\log n)^c}$ factor where $c > 0$ is some fixed constant independent of n.

Bhaskara et al. [7] study strong SDP relaxations of the problem and show that the integrality gap of DkS remains $n^{\Omega_\varepsilon(1)}$ even after $n^{1-\varepsilon}$ rounds of the Lasserre hierarchy. Also for $n^{\Omega(\varepsilon)}$ rounds, the gap is as large as $n^{2/53-\varepsilon}$. Moreover for the Sherali-Adams relaxation, they show a lower bound of $\Omega\left(n^{1/4}/\log^3 n\right)$ on the integrality gap for $\Omega\left(\log n/\log\log n\right)$ rounds.

Ames [2] studies the planted DkS problem using a non-SDP convex relaxation for instances of the following kind. Let S be the planted dense subgraph (of size k), they claim that if $G[S]$ contains at least $\binom{k}{2} - c_1 k^2$ edges and the subgraph $G[V \setminus S]$ contains at most $c_2 k^2$ edges where c_1, c_2 are constants depending on other parameters of the graph like the density of the subgraph $G[S]$ etc, then under some mild technical conditions, they show that the unique optimal solution to their convex program is integral and corresponds to the set S. They also study analogous models for bipartite graphs.

Random models for DkS. Bhaskara et al. [6] study a few random models of instances for the Densest k-subgraph problem, we describe them here. Let $\mathcal{D}_1$ denote the distribution of Erdős-Rényi random graphs $G(n,p)$ and let $\mathcal{D}_2$ denote the distribution of graphs constructed as follows. Starting with a "host graph" of average degree D ($D \stackrel{\text{def}}{=} np$), a set S of k vertices is chosen arbitrarily and the subgraph on S is replaced with a dense subgraph of average degree d. Given $G_1 \sim \mathcal{D}_1$ and $G_2 \sim \mathcal{D}_2$, the problem is to distinguish between the two distributions. They consider this problem in three different models with varying assumptions on $\mathcal{D}_2$, (i) *Random Planted Model* : the host graph and the planted dense subgraph are random, (ii) *Dense in Random Model* : an arbitrary dense graph is planted inside a random graph, and (iii) *Dense vs Random Model* : an arbitrary dense graph is planted inside an arbitrary graph.

The *planted dense subgraph* recovery problem is similar in spirit to the *Random Planted Model* where the goal is to recover a hidden community of size k within a larger graph which is constructed as follows : two vertices are connected by an edge with probability p if they belong to the same community and with probability q otherwise. The typical setting of parameters is, $p > q$. The works by [33, 20, 34, 19, 21, 8, 2] studies this problem using SDP based, spectral, statistical, message passing algorithms etc.

We give a brief overview of their distinguishing algorithms in the three models. Given a graph on n vertices with average degree d_{avg}, its log-density is defined as $\dfrac{\log d_{\text{avg}}}{\log n}$. Let

Θ_1 and Θ_2 denote the log-density of G_1 and the log-density of the planted subgraph $G_2[S]$ respectively. Their algorithm is based on the counts of a specially constructed small-sized tree (the size of which is parameterized by relatively prime integers r, s such that $s > r > 0$) as a subgraph in G_1 and G_2. They show that if $\Theta_1 \leq r/s$, then G_1 will have at most poly-logarithmic $(\mathcal{O}(\log n)^{s-r})$ number of such subtrees. On the other hand, when $\Theta_2 \geq r/s + \varepsilon$ where $\epsilon > 0$ is a small constant, they show that there at least k^{ϵ} such subtrees (even in the *Dense vs Random Model*). Now if $k > (\log n)^{\omega(1)}$, they use this difference in the log-densities to show the gap between counts of such trees in G_1 and G_2, and hence are able to distinguish between the two distributions. They show that the running time of this algorithm is $n^{\mathcal{O}(r)}$. Also for constant Θ_1 and Θ_2, the running time is $n^{\mathcal{O}(1/(\Theta_2 - \Theta_1))}$ ([6, 5]). We call this algorithm the "subgraph counting" algorithm.

The distinguishing problem can be restated as the following : For a given n, k, p, we are interested in finding the smallest value of d for which the problem can be solved. For a certain range of parameters, spectral, SDP based methods, etc. can be used to work for small values of d. For example, in the *Dense vs Random Model*, when $k > \sqrt{n}$ a natural SDP relaxation of DkS can be used to distinguish between G_1 and G_2 for $d > \sqrt{D} + kD/n$ (which is smaller than $D^{\log_n k}$, the threshold of the subgraph counting algorithm). They upper bound the cost of the optimal SDP solution for a random graph G_1, by constructing a feasible dual solution which certifies (w.h.p.) that it cannot contain a k-subgraph with density more than that of $\sqrt{D} + kD/n$. We use their results in bounding the cost of the SDP contribution from $G[V \setminus S]$ in the DkSExp$(n, k, d, \delta, d', \lambda)$ and DkSExpReg$(n, k, d, \delta, d', \lambda)$ models.

The distribution $\mathcal{D}_2$ of graphs considered in the *Dense in Random Model* (arbitrary dense graph planted in a random graph) is similar to a subset of DkSExp$(n, k, d, \delta, d', \lambda)$ instances since $G[S]$ is an arbitrary dense subgraph in both models and $G[S, V \setminus S]$ is a random graph in both the models. The difference is in the subgraph $G[V \setminus S]$, where this is a random graph in the *Dense in Random* model whereas our models require it to be a regular expander. While our proofs require the expander to be regular, they can also be made to work for random graphs since we use the bound on the SDP value from [6] (analysis in Section 2.2). We note that while random graphs are good expanders w.h.p., the converse of this fact is not true in general, since there are known deterministic constructions of expander graphs.

We look at the range of parameters where the following two algorithms can be used to solve the *Dense in Random* problem. One is the SDP based algorithm proposed in our work (closely related to DkSExp$(n, k, d, \delta, d', \lambda)$ model) and second is the subgraph counting algorithm which uses the difference in the log-densities of the planted subgraph and the host graph to distinguish the two distributions from [6, 5]. For the purposes of comparison, we consider the case when $k, d = poly(n)$ and $p = 1/poly(n)$. Also we ignore the low-order terms in these expressions. In this regime, our algorithms' threshold is

$$d = \Omega\left(\max\left\{pk, \sqrt{np}\right\}\right) \tag{1}$$

since we can use the objective value of the SDP 1.11 to distinguish between the cases in this range of d. For G_1, this value is at most $k\left(pk + \sqrt{np}\right)/2$ (Lemma 2.12) while for G_2 it is at least $kd/2$. Moreover, Algorithm 1 can be used to recover a part of the planted solution as the value of ν is small (when d satisfies Equation (1), ν is bounded away and smaller than 1) in this regime (see Section 2 and Theorem 2.1).

The counting algorithms' threshold (or the log-density threshold) is

$$\frac{\log d}{\log k} - \frac{\log np}{\log n} > 0 \iff \log d > \frac{\log k \log np}{\log n} \iff d = \Omega\left((np)^{\log_n k}\right)$$

and its running time is $n^{\mathcal{O}\left(\frac{1}{\log_k d - \log_n np}\right)}$. We look at different ranges of k and compare the values of d for which the two algorithms can solve the distinguishing problem.

1. $k = \Theta(\sqrt{n})$.
   In this case, $\max\{pk, \sqrt{np}\} = \sqrt{np}$. This matches with the log-density threshold. Note that for $p = \Theta(1/\sqrt{n})$, we get $d = \Omega\left(n^{1/4}\right)$. To the best of our knowledge, there is no poly-time algorithm which beats this lower bound.
2. $k = \omega(\sqrt{n})$.
   In this setting, $(np)^{\log_n k} = \omega\left(\sqrt{np}\right)$. Also, $(np)^{\log_n k} = k(p)^{\log_n k} = \omega(pk)$. Thus our algorithm has a better threshold in this regime. There is a spectral algorithm, see Section 6.2 of [6], which uses the second eigenvalue of the adjacency matrix which can distinguish with the same threshold as our algorithm in this regime.
3. $k = o(\sqrt{n})$.
   In this case, $(np)^{\log_n k} = o\left(\sqrt{np}\right)$. Here the log-density threshold is smaller than our threshold. Therefore the algorithm by Bhaskara et al. [6] works for a larger range of parameters than our algorithms.

Other semi-random models. Semi-random instances of many other fundamental problems have been studied in the literature. This includes the unique games problem [25], graph coloring [1, 11, 12], graph partitioning problems such as balanced-cut, multi-cut, small set expansion [28, 29, 26, 27], etc. [30] studies the problem of learning communities in the Stochastic Block Model in the presence of adversarial errors.

McKenzie, Mehta and Trevisan [32] study the complexity of the independent set problem in the Feige-Killian model [13]. Instead of using a SDP relaxation for the problem, they use a "crude" SDP (introduced in [25]) which exploits the geometry of vectors (orthogonality etc.) to reveal the planted set. They bound the SDP contribution by the vertex pairs, $S \times V \setminus S$ using the Grothendieck inequality and thereby showing that the vectors in S are "clustered" together. Their algorithm outputs w.h.p. a large independent set when $k = \Omega\left(n^{2/3}/p^{1/3}\right)$. Also, for the parameter range $k = \Omega\left(n^{2/3}/p\right)$, it outputs a list of at most n independent sets of size k, one of which is the planted one.

Semi-random models for graph partitioning problems. The problem of DkS is very closely related to the Small Set Expansion problem (SSE, henceforth). This problem has been very well studied in the literature. At the first glance, the problem of DkS can be thought of as finding a small set S of size k which is non-expanding. The densest set is typically a non-expanding set because most of the edges incident on S would remain inside it than leaving it. But the converse is not true, since all sets of cardinality k which have small expansion are not dense. In particular, in our model, by the action of the monotone adversary on $V \setminus S$, there can exist many small sets (of size $\mathcal{O}(k)$) which not only have a very small fraction of edges going outside but can have very few edges left inside as well. This makes the problem of DkS very different from the SSE problem. Nevertheless, we survey some related works of semi-random models of SSE. The works [36, 4] study the worst-case approximation factors for the SSE problem and give bi-criteria approximation algorithms for the same. Their algorithms are also based on rounding a SDP relaxation.

Makarychev, Markarychev and Vijayaraghavan [28] study the complexity of many graph partitioning problems including balanced cut, SSE, and multi-cut etc. They consider the following model : Partition V into $(S, V \setminus S)$ such that $G[S]$ and $G[V \setminus S]$ are arbitrary while

$G[S, V \setminus S]$ is a random graph with some probability ε. They allow an adversary to add edges within S and $V \setminus S$, and delete any edges across these sets. They get constant factor bi-criteria approximation algorithms (under some mild technical conditions) in this model. In the case of balanced cut and SSE problems, when the partitions themselves have enough expansion within them, they can recover the planted cut upto a small error.

Louis and Venkat [26] study the problem of balanced vertex expansion in a natural semi-random model and get a bi-criteria approximation algorithm for the same. They even get an exact recovery for a restricted set of parameters in their model. Their proof consisted of constructing an optimal solution to the dual of the SDP relaxation and using it to show the integrality of the optimal primal solution. In [27], they study the problem for a general, balanced $k-$way vertex (and edge) expansion and give efficient algorithms for the same. Their construction consists of k (almost) regular expander graphs (over vertices $\{S_i\}_{i=1}^{k}$, each of size n/k) and then adding edges across them ensuring that the expansion of each of the $G[S_i]'s$ is small. Their algorithm is based on rounding a SDP relaxation and then showing that the vertices of each S_i are "clustered" together around the mean vector μ_i and for different sets S_i and S_j, μ_i and μ_j are sufficiently apart. This gives a way to recover a good solution. Our approach also shows that the SDP vectors for the vertices in S are "clustered" together. However arriving at such a conclusion requires different ideas because of the new challenges posed by the nature of the problem and assumptions on our models.

1.4 SDP formulation

We use the following Semidefinite/Vector Programming relaxation for our problem, over the vectors X_i ($i \in [n]$) and I.

▶ **SDP 1.11.**

$$\textbf{\textit{maximize}} \qquad \frac{1}{2}\sum_{i,j=1}^{n} A_{ij}\langle X_i, X_j\rangle \tag{2}$$

$$\textbf{\textit{subject to}} \qquad \sum_{i=1}^{n}\langle X_i, X_i\rangle = k \tag{3}$$

$$\sum_{j=1}^{n}\langle X_i, X_j\rangle \le k\langle X_i, X_i\rangle \qquad \forall i \in [n] \tag{4}$$

$$0 \le \langle X_i, X_j\rangle \le \langle X_i, X_i\rangle \qquad \forall i, j \in [n],\ (i \ne j) \tag{5}$$

$$\langle X_i, X_i\rangle \le 1 \qquad \forall i \in [n] \tag{6}$$

$$\langle X_i, I\rangle = \langle X_i, X_i\rangle \qquad \forall i \in [n] \tag{7}$$

$$\langle I, I\rangle = 1 \tag{8}$$

We note that these programs can be solved efficiently using standard algorithms, like ellipsoid and interior point methods. To see, why the above SDP 1.11 is a relaxation, let S be the optimal set and v be any unit vector. It is easy to verify the solution set,

$$X_i = \begin{cases} v & i \in S \\ 0 & i \in V \setminus S \end{cases} \qquad \text{and} \qquad I = v\,.$$

is feasible for SDP 1.11 and gives the objective value equal to its optimal density.

1.5 Proof Overview

Our algorithms are based on rounding an SDP relaxation (SDP 1.11) for the Densest k-subgraph problem. At a high level, we show that most of the SDP mass is concentrated on the vertices in S (Proposition 2.16). To show this, we begin by observing that the SDP objective value is at least $kd/2$ since the integer optimal solution to the SDP has value at least $kd/2$. Therefore, by proving an appropriate upper bound on the SDP value from edges in $S \times (V \setminus S)$ (Proposition 2.2) and the edges in $V \setminus S$ (Proposition 2.11), we can get a lower bound on the SDP value from the edges inside S.

The edges in $S \times (V \setminus S)$ form a random bipartite graph. We can bound the contribution towards the SDP mass from this part by bounding the contribution from the "expected graph" (Lemma 2.5) and the contribution from the random graph minus the expected graph (Corollary 2.10). The contribution from the latter part can be bounded using bounds on the spectra of random matrices (Corollary 2.8). Since the expected graph is a complete weighted graph with edge weights equal to the edge probability, the contribution from this part can be bounded using the SDP constraints (Lemma 2.5).

For DkSExp$(n, k, d, \delta, d', \lambda)$ and DkSExpReg$(n, k, d, \delta, d', \lambda)$, we use a result by [6]. They construct a feasible solution to the dual of the SDP for random graphs, thereby bounding the cost of the optimal solution of the primal. Their proof only uses a bound on the spectral gap of the graph, and therefore, holds also for expander graphs. Therefore, this result gives us the desired bound on the SDP value on the edges inside $V \setminus S$ in these models (Proposition 2.11). We also give an alternate proof of the same result using the spectral properties of the adjacency matrix of $V \setminus S$ in the full version of the paper; this approach is similar in spirit to the proof of the classical *expander mixing lemma*.

For DkS$(n, k, d, \delta, \gamma)$ and DkSReg$(n, k, d, \delta, \gamma)$, we bound the SDP value on the edges inside $V \setminus S$ using a result of Charikar [10]. This work showed that for a graph $H = (V', E')$, a natural LP relaxation can be used to compute $\max_{W \subseteq V'} \rho(W)/|W|$. We show that we can use our SDP solution to construct a feasible solution for this LP. Since $\rho(W)/|W| \leq \gamma d$, $\forall W \subset V \setminus S$ in this model, Charikar's result [10] implies that the cost of any feasible LP solution can be bounded by γd. This gives us the desired bound on the SDP value on the edges inside $V \setminus S$ in these models.

These bounds establish that most of the SDP mass is on the edges inside S. Using the SDP constraints, we show that the set of vertices corresponding to all the "long" vectors will contain a large weight of edges inside S (Corollary 2.19). Moreover, since the sum of squared lengths of the vectors is k (from the SDP constraints), we can only have $\mathcal{O}(k)$ long vectors (Lemma 2.20). Using standard techniques from the literature, we can prune this set to obtain a set of size at most k and having large density [37]. In the case when the graph induced on S is d-regular, we show that if a set contains a large fraction of the edges inside S, then it must also have a large intersection with S. We present our complete procedure in Algorithm 1.

We note that while this framework for showing that the SDP mass is concentrated on the planted solution has been used for designing algorithms for semi-random instances of other problems as well, proving quantitative bounds is problem-specific and model-specific: different problems and different models require different approaches.

Organization of the paper

Due to space constraints, we present the complete version (with all the details and proofs) of Section 2 in the full version of the paper, however we do state the key technical results

here with the proof of Theorem 2.1. We state and prove the formal versions of Theorem 1.6, Theorem 1.7, and Theorem 1.8 in the full version of the paper.

2 Analysis of DkSExp$(n, k, d, \delta, d', \lambda)$

In this section, we will analyse the DkSExp$(n, k, d, \delta, d', \lambda)$ model. Our main result is the following.

▶ **Theorem 2.1** (Formal version of Theorem 1.3)**.** *There exist universal constants $\kappa, \xi \in \mathbb{R}^+$ and a deterministic polynomial time algorithm, which takes an instance of Dk*SExp*$(n, k, d, \delta, d', \lambda)$ where*

$$\nu = 2\sqrt{3\left(6\delta + \xi\sqrt{\frac{\delta n}{dk}} + \frac{\lambda}{d} + \frac{d'k}{(n-k)\,d}\right)},$$

satisfying $\nu \in (0,1)$, and $\delta d/k \in [\kappa \log n/n, 1)$, and outputs with high probability (over the instance) a vertex set $\mathcal{Q}$ of size k such that

$$\rho(\mathcal{Q}) \geq (1-\nu)\,\frac{kd}{2}\,.$$

The above algorithm also computes a vertex set T such that

(a) $|T| \leq k\left(1+\frac{\nu}{5}\right)$. **(b)** $\rho(T \cap S) \geq \left(1-\frac{\nu}{2}\right)\frac{kd}{2}$.

In the analysis below, without loss of generality we can ignore the adversarial action (step 4 of the model construction) to have taken place. Let us assume the montone adversary removes edges arbitrarily from the subgraphs $G[V \setminus S]$ & $G[S, V \setminus S]$ and the new resulting adjacency matrix is A'. Then for any feasible solution $\{\{Y_i\}_{i=1}^n, I\}$ of the SDP, we have $\sum_{i\in P, j\in Q} A'_{ij}\langle Y_i, Y_j\rangle \leq \sum_{i\in P, j\in Q} A_{ij}\langle Y_i, Y_j\rangle$ for $\forall P, Q \subseteq V$. This holds because of the non-negativity constraint Equation (5). Thus the upper bounds on SDP contribution by vectors in $G[S, V\setminus S]$ and $G[V\setminus S]$ as claimed by Proposition 2.2 and Proposition 2.11 respectively are intact and the rest of the proof follows exactly. Hence, without loss of generality, we can ignore this step in the analysis of our algorithm.

2.1 Edges between S and $V \setminus S$

In this section, we show an upper bound on $\sum_{i\in S, j\in V\setminus S} A_{ij}\langle X_i, X_j\rangle$.

▶ **Proposition 2.2.** *W.h.p. (over the choice of the graph), we have*

$$\sum_{i\in S, j\in V\setminus S} A_{ij}\langle X_i, X_j\rangle \leq 3pk^2\left(1 - \mathop{\mathbb{E}}_{i\sim S}\|X_i\|^2\right) + \xi k\sqrt{np}\sqrt{\left(\mathop{\mathbb{E}}_{i\sim S}\|X_i\|^2\right)\left(1 - \mathop{\mathbb{E}}_{i\sim S}\|X_i\|^2\right)}\,.$$

Note that

$$\sum_{i\in S, j\in V\setminus S} A_{ij}\langle X_i, X_j\rangle = p\sum_{i\in S, j\in V\setminus S}\langle X_i, X_j\rangle + \sum_{i\in S, j\in V\setminus S}(A_{ij}-p)\langle X_i, X_j\rangle\,. \tag{9}$$

We will bound the two terms in the R.H.S. of Equation (9) separately. The first term relies only on the *expected graph* and can be bounded using the SDP constraints. We use bounds on the eigenvalues of random bipartite graphs to bound the second term.

Bound the contribution from the *expected graph*

We first prove some properties of the SDP solutions that we will use to bound this term. The following lemma shows that if the expected value of the squared norm of the vectors corresponding to the set S is "large", then their expected pairwise inner product is "large" as well.

▶ **Lemma 2.3.** *Let $\{\{Y_i\}_{i=1}^n, I\}$ be any feasible solution of SDP 1.11 and $T \subseteq V$ such that, $\mathop{\mathbb{E}}_{i\sim T} \|Y_i\|^2 \geq 1-\epsilon$ where $0 \leq \epsilon \leq 1$, then $\mathop{\mathbb{E}}_{i,j\sim T} \langle Y_i, Y_j\rangle \geq 1-4\epsilon$.*

▶ **Corollary 2.4.**

$$\mathop{\mathbb{E}}_{i,j\sim S} \langle X_i, X_j\rangle \geq 4 \mathop{\mathbb{E}}_{i\sim S} \|X_i\|^2 - 3\,.$$

We are now ready to bound the first term in Equation (9).

▶ **Lemma 2.5.**

$$\sum_{i\in S, j\in V\setminus S} \langle X_i, X_j\rangle \leq 3k^2 \left(1 - \mathop{\mathbb{E}}_{i\sim S} \|X_i\|^2\right)\,.$$

Bounding the deviation from the *expected graph*

We now prove the following lemmas which we will use to bound the second term in Equation (9). Let B be the $n\times n$ matrix defined as follows.

$$B_{ij} \stackrel{\text{def}}{=} \begin{cases} A_{ij} - p & i\in S, j\in V\setminus S \text{ or } i\in V\setminus S, j\in S \\ 0 & \text{otherwise} \end{cases}\,.$$

▶ **Lemma 2.6.**

$$\sum_{i,j\in V} B_{ij}\langle X_i, X_j\rangle \leq 2k\|B\| \sqrt{\left(\mathop{\mathbb{E}}_{i\sim S}\|X_i\|^2\right)\left(1-\mathop{\mathbb{E}}_{i\sim S}\|X_i\|^2\right)}\,.$$

Now, we use the following folklore result to bound $\|B\|$.

▶ **Theorem 2.7** ([21], Lemma 30)**.** *Let M be a symmetric matrix of size $n\times n$ with zero diagonals and independent entries such that $M_{ij} = M_{ji} \sim \mathsf{Bern}(p_{ij})$ for all $i<j$ with $p_{ij}\in[0,1]$. Assume $p_{ij}(1-p_{ij}) \leq r$ for all $i<j$ and $nr = \Omega(\log n)$. Then, with high probability (over the randomness of matrix M),*

$$\|M - \mathbb{E}[M]\| \leq \mathcal{O}(1)\sqrt{nr}\,.$$

▶ **Corollary 2.8.** *There exists universal constants $\kappa, \xi \in \mathbb{R}^+$ such that if $p \in \left[\frac{\kappa\log n}{n}, 1\right)$, then*

$$\|B\| \leq \xi\sqrt{np}$$

with high probability (over the choice of the graph).

▶ Remark 2.9. Note that, Corollary 2.8 holds with high probability when $p = \Omega(\log n/n)$. In the rest of the paper, we work in the range of parameters where this lower bound on p is satisfied. However, we do restate it when explicitly using this bound.

▶ **Corollary 2.10.** *W.h.p. (over the choice of the graph),*

$$\sum_{i,j\in V} B_{ij}\langle X_i, X_j\rangle \leq 2\xi k\sqrt{np}\sqrt{\left(\mathop{\mathbb{E}}_{i\sim S}\|X_i\|^2\right)\left(1-\mathop{\mathbb{E}}_{i\sim S}\|X_i\|^2\right)}\,.$$

2.2 Edges in $V \setminus S$

We recall, the subgraph $G[V \setminus S]$ is a $(d', \lambda)-$expander in the DkSE$\mathsf{xp}(n, k, d, \delta, d', \lambda)$ model. We show the following upper bound on the SDP mass contribution by the vectors in $V \setminus S$.

▶ **Proposition 2.11.**

$$\sum_{i,j \in V \setminus S} A_{ij} \langle X_i, X_j \rangle \leq \left(\lambda k + \frac{d'k^2}{n-k}\right)\left(1 - \mathop{\mathbb{E}}_{i \sim S} \|X_i\|^2\right) .$$

To prove the above proposition, we use the following results from the Bhaskara et al. [6] paper.

▶ **Lemma 2.12** ([6], Theorem 6.1)**.** *For a $G(n,p)$ (Erdős-Rényi model) graph, the value of the SDP (SDP 1.11) is at most $k^2 p + \mathcal{O}\left(k\sqrt{np}\right)$ with high probability when $p = \Omega(\log n/n)$.*

▶ **Lemma 2.13** ([6], Theorem 6.1)**.** *For a (d', λ)-expander graph on n vertices, the value of the SDP (SDP 1.11) is at most $\frac{k^2 d'}{n} + k\lambda$.*

We note that, though the statement proved in [6] is about random graphs (Lemma 2.12), their proof follows as is for an expander graph. Since, we are only applying Lemma 2.13 to the subgraph $G[V \setminus S]$, we use a scaling factor of $\left(1 - \mathbb{E}_{i \sim S} \|X_i\|^2\right)$. The proof of Proposition 2.11 follows directly from the above lemma. We also provide an alternate proof of this in the full version of the paper.

▶ Remark 2.14. If the subgraph, $G[V \setminus S]$ is a random graph ($G(n-k,p)$) as considered in our discussion in Section 1.3, we can analogously use Lemma 2.12 to get upper bounds on $\sum_{i,j \in V \setminus S} A_{ij} \langle X_i, X_j \rangle$.

2.3 Putting things together

We have shown upper bounds on the SDP mass from the edges in $S \times (V \setminus S)$ (Proposition 2.2) and from the edges in $V \setminus S$ (Proposition 2.11). We combine these results to show that the average value of $\langle X_u, X_v \rangle$ where $\{u, v\} \in E(G[S])$ is "large" (Proposition 2.16). The SDP constraint Equation (5) implies the corresponding vertices, u and v have large squared norms as well. This immediately guides us towards a selection criteria/recovery algorithm. However we need to output a vertex set of size at most k, we prune this set using a greedy strategy (Algorithm 1).

▶ **Lemma 2.15.**

$$\sum_{i,j \in S} A_{ij} \langle X_i, X_j \rangle = (kd) \mathop{\mathbb{E}}_{\{i,j\} \sim E(G[S])} \langle X_i, X_j \rangle .$$

▶ **Proposition 2.16.** *W.h.p. (over the choice of the graph), we have $\mathop{\mathbb{E}}_{\{i,j\} \sim E(G[S])} \langle X_i, X_j \rangle \geq 1 - \eta$, where*

$$\eta = 6\delta + \xi\sqrt{\frac{\delta n}{dk}} + \frac{\lambda}{d} + \frac{d'k}{(n-k)d} .$$

Now, we present the complete algorithm below.

Algorithm 1 Recovering a dense set $\mathcal{Q}$.

Input: An Instance of DkSExp$(n,k,d,\delta,d',\lambda)$ / DkSExpReg$(n,k,d,\delta,d',\lambda)$ / DkS(n,k,d,δ,γ) / DkSReg(n,k,d,δ,γ) and a parameter $0<\eta<1$.
Output: A vertex set $\mathcal{Q}$ of size k.

Solve SDP 1.11 to get the vectors $\left\{\{X_i\}_{i=1}^n, I\right\}$.
$\alpha = \begin{cases} 1/\sqrt{3\eta} & \text{For instances of type, D}k\text{SExp}(n,k,d,\delta,d',\lambda) \text{ or D}k\text{S}(n,k,d,\delta,\gamma)\ . \\ 2/\sqrt{\eta} & \text{For instances of type, D}k\text{SExpReg}(n,k,d,\delta,d',\lambda) \text{ or D}k\text{SReg}(n,k,d,\delta,\gamma). \end{cases}$
Let $T = \left\{i \in V : \|X_i\|^2 \geq 1-\alpha\eta\right\}$.
Initialize $\mathcal{Q} = T$.
if $|\mathcal{Q}| < k$ **then**
    Arbitrarily add remaining vertices to set $\mathcal{Q}$ to make its size k.
else
    while $|\mathcal{Q}| \neq k$ **do**
        Remove the minimum weighted vertex from the set $\mathcal{Q}$.
    end while
end if
Return $\mathcal{Q}$.

Note that if $\eta = 0$, the SDP returns an integral solution and we can recover the set S exactly. Therefore, w.l.o.g. we assume $\eta \neq 0, 1$.

To analyse the cost of the solution returned by Algorithm 1, we define two sets as follows.

$$T' \stackrel{\text{def}}{=} \{\{i,j\} \in E : \langle X_i, X_j\rangle \geq 1-\alpha\eta\} \qquad \text{and} \qquad T \stackrel{\text{def}}{=} \left\{i \in V : \|X_i\|^2 \geq 1-\alpha\eta\right\},$$

where $1 < \alpha < 1/\eta$ is a parameter to be fixed later.

We show that a *large* weight of the edges inside S also lies in the set T'.

▶ **Lemma 2.17.** *W.h.p. (over the choice of the graph),*

$$\sum_{e \in T' \cap E(G[S])} w(e) \geq \frac{kd}{2}\left(1-\frac{1}{\alpha}\right).$$

The following lemma shows that the subgraph induced on $T \cap S$ contains all the edges in $T' \cap E\left(G[S]\right)$.

▶ **Lemma 2.18.** *W.h.p. (over the choice of the graph),*

$$T' \cap E\left(G[S]\right) \subseteq E(G[T \cap S])\,.$$

▶ **Corollary 2.19.** *W.h.p. (over the choice of the graph),*

$$\rho\left(T\right) \geq \rho\left(T \cap S\right) \geq \frac{kd}{2}\left(1-\frac{1}{\alpha}\right).$$

We have shown that the subgraph induced on T has a large weight ($\approx kd/2$). In the next lemma, we show that the size of set T is not too large compared to k.

▶ **Lemma 2.20.** *W.h.p. (over the choice of the graph),*

$$|T| \leq \frac{k}{1-\alpha\eta}\,.$$

To prune the set T and obtain a set of size k, we use a lemma from the work by Srivastav et al. [37].

▶ **Lemma 2.21** ([37], Lemma 1). *Let $V', V'' \subseteq V$ be non-emply subsets such that $|V''| \geq |V'|$, then the greedy procedure which picks the lowest weighted vertex from V'' and removes it iteratively till we have $|V'|$ vertices left ensures, $\rho(V') \geq \frac{|V'|(|V'|-1)}{|V''|(|V''|-1)} \rho(V'')$.*

We are now ready to prove the main result which gives the approximation guarantee of our algorithm. We also set the value of parameter α which maximizes the density of the output graph.

Proof of Theorem 2.1. We run Algorithm 1 on DkSExp$(n, k, d, \delta, d', \lambda)$ with η as given in Proposition 2.16. From Lemma 2.21, we have a handle on the density of the new set $(\mathcal{Q})$ *after pruning* T to a set of size k. The algorithm performs this exactly in the steps 5 to 11. Let ALG denote the density of this new set (output of Algorithm 1). We have,

$$\begin{aligned}
\mathsf{ALG} &\geq \left(\frac{k(k-1)}{|T|(|T|-1)}\right)\left(1-\frac{1}{\alpha}\right)\frac{kd}{2} && \text{(by Corollary 2.19 and Lemma 2.21)}\\
&\geq \left(\frac{(1-\alpha\eta)^2}{1+\alpha\eta/(k-1)}\right)\left(1-\frac{1}{\alpha}\right)\frac{kd}{2} && \text{(by Lemma 2.20 and dividing by } k-1)\\
&\geq \left(\frac{(1-\alpha\eta)^2}{1+\alpha\eta}\right)\left(1-\frac{1}{\alpha}\right)\frac{kd}{2} && \text{(w.l.o.g., } k \geq 2)\\
&\geq (1-2\alpha\eta)(1-\alpha\eta)\left(1-\frac{1}{\alpha}\right)\frac{kd}{2} && \left((1-x)^2 \geq 1-2x \;\&\; \frac{1}{1+x} \geq 1-x,\ \forall x \in \mathbb{R}_{\geq 0}\right)\\
&\geq \left(1-3\alpha\eta-\frac{1}{\alpha}\right)\frac{kd}{2} && \text{(rearranging and bounding the positive terms by 0)}\\
&= \left(1-2\sqrt{3\eta}\right)\frac{kd}{2} && \left(\text{we fix } \alpha = 1/\sqrt{3\eta}\right).
\end{aligned}$$

Letting $\nu \stackrel{\text{def}}{=} 2\sqrt{3\eta}$, we get that $\mathsf{ALG} \geq (1-\tau)\,kd/2$ where

$$\nu = 2\sqrt{3\left(6\delta + \xi\sqrt{\frac{\delta n}{dk}} + \frac{\lambda}{d} + \frac{d'k}{(n-k)d}\right)} \qquad \text{(using the value of } \eta \text{ from Proposition 2.16)}.$$

From Lemma 2.20, $|T| \leq \frac{k}{1-\alpha\eta} = \frac{k}{1-(\nu/6)} \leq k\left(1+\frac{\nu}{5}\right)$. And from Corollary 2.19, $\rho(T \cap S) \geq \frac{kd}{2}\left(1-\frac{1}{\alpha}\right) = \frac{kd}{2}\left(1-\frac{\nu}{2}\right)$. ◀

Note that for the parameter range $0 < 2\sqrt{3\eta} < 1 \iff 0 < \nu < 1$, the value of α $(= 1/\sqrt{3\eta})$ fixed by the algorithm lies in the interval $(1, 1/\eta)$ as required.

▶ Remark 2.22 (on Theorem 1.3). In the restricted parameter case, we simplify the arguments in our informal theorem statements, i.e. the case when the average degree of vertices in S and $V \setminus S$ is close, we have $\delta = \Theta\left(\frac{kd'}{nd}\right)$. Assuming $\nu = 2\sqrt{3\eta}$, we rewrite $\frac{\delta n}{dk}$ as $\frac{d'}{d^2}$ from the above value of δ and the term $\frac{(d'-\lambda)\,k}{(n-k)\,d}$ is at most a constant for "large" n. So, the new value of τ is $\Theta\left(\sqrt{\delta + \frac{\lambda+\sqrt{d'}}{d}}\right)$. A similar argument gives the new value of ν' in Theorem 1.7.

References

1 Noga Alon and Nabil Kahale. A spectral technique for coloring random 3-colorable graphs. *SIAM J. Comput.*, 26(6):1733–1748, December 1997. doi:10.1137/S0097539794270248.

2 Brendan P. Ames. Guaranteed recovery of planted cliques and dense subgraphs by convex relaxation. *J. Optim. Theory Appl.*, 167(2):653–675, November 2015. doi:10.1007/s10957-015-0777-x.

3 Yuichi Asahiro, Kazuo Iwama, Hisao Tamaki, and Takeshi Tokuyama. Greedily finding a dense subgraph. In *Algorithm Theory — SWAT'96*, pages 136–148, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

4 Nikhil Bansal, Uriel Feige, Robert Krauthgamer, Konstantin Makarychev, Viswanath Nagarajan, Joseph Naor, and Roy Schwartz. Min-max graph partitioning and small set expansion. *SIAM J. Comput.*, 43(2):872–904, 2014. doi:10.1137/120873996.

5 Aditya Bhaskara. *Finding dense structures in graphs and matrices.* PhD thesis, Princeton University, 2012. URL: https://www.cs.utah.edu/~bhaskara/files/thesis.pdf.

6 Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 201–210, 2010. doi:10.1145/1806689.1806719.

7 Aditya Bhaskara, Moses Charikar, Aravindan Vijayaraghavan, Venkatesan Guruswami, and Yuan Zhou. Polynomial integrality gaps for strong sdp relaxations of densest k-subgraph. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, page 388–405, USA, 2012. Society for Industrial and Applied Mathematics.

8 Polina Bombina and Brendan Ames. Convex optimization for the densest subgraph and densest submatrix problems, 2019. arXiv:1904.03272.

9 Mark Braverman, Young Kun Ko, Aviad Rubinstein, and Omri Weinstein. Eth hardness for densest-k-subgraph with perfect completeness. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 1326–1341, Philadelphia, PA, USA, 2017. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=3039686.3039772.

10 Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *Proceedings of the Third International Workshop on Approximation Algorithms for Combinatorial Optimization*, APPROX '00, pages 84–95, Berlin, Heidelberg, 2000. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=646688.702972.

11 Amin Coja-Oghlan. Colouring semirandom graphs. *Comb. Probab. Comput.*, 16(4):515–552, July 2007. doi:10.1017/S0963548306007917.

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. ACM. doi:10.1145/2897518.2897561.

13 Uriel Feige and Joe Kilian. Heuristics for semirandom graph problems. *J. Comput. Syst. Sci.*, 63(4):639–671, December 2001. doi:10.1006/jcss.2001.1773.

14 Uriel Feige, Guy Kortsarz, and David Peleg. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001. doi:10.1007/s004530010050.

15 Uriel Feige and Michael Langberg. Approximation algorithms for maximization problems arising in graph partitioning. *Journal of Algorithms*, 41(2):174–211, 2001. doi:10.1006/jagm.2001.1183.

16 Uriel Feige and Michael Seltser. On the densest k-subgraph problem. *Algorithmica*, 29:2001, 1997.

17 G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.*, 18(1):30–55, February 1989. doi:10.1137/0218003.

18 A. V. Goldberg. Finding a maximum density subgraph. Technical report, University of California at Berkeley, Berkeley, CA, USA, 1984.

19 B. Hajek, Y. Wu, and J. Xu. Achieving exact cluster recovery threshold via semidefinite programming: Extensions. *IEEE Transactions on Information Theory*, 62(10):5918–5937, 2016.

20 Bruce Hajek, Yihong Wu, and Jiaming Xu. Computational Lower Bounds for Community Detection on Random Graphs. *arXiv e-prints*, page arXiv:1406.6625, June 2014. `arXiv:1406.6625`.

21 Bruce Hajek, Yihong Wu, and Jiaming Xu. Semidefinite programs for exact recovery of a hidden community. *Journal of Machine Learning Research*, 49(June):1051–1095, June 2016. 29th Conference on Learning Theory, COLT 2016 ; Conference date: 23-06-2016 Through 26-06-2016.

22 Ravi Kannan and V Vinay. *Analyzing the structure of large graphs*. Rheinische Friedrich-Wilhelms-Universität Bonn Bonn, 1999.

23 Subhash Khot. Ruling out ptas for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM J. Comput.*, 36(4):1025–1071, December 2006. `doi:10.1137/S0097539705447037`.

24 Samir Khuller and Barna Saha. On finding dense subgraphs. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming: Part I*, ICALP '09, pages 597–608, Berlin, Heidelberg, 2009. Springer-Verlag. `doi:10.1007/978-3-642-02927-1_50`.

25 Alexandra Kolla, Konstantin Makarychev, and Yury Makarychev. How to play unique games against a semi-random adversary: Study of semi-random models of unique games. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 443–452, 2011. `doi:10.1109/FOCS.2011.78`.

26 Anand Louis and Rakesh Venkat. Semi-random graphs with planted sparse vertex cuts: Algorithms for exact and approximate recovery. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pages 101:1–101:15, 2018. `doi:10.4230/LIPIcs.ICALP.2018.101`.

27 Anand Louis and Rakesh Venkat. Planted models for k-way edge and vertex expansion. In *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019, December 11-13, 2019, Bombay, India*, pages 23:1–23:15, 2019. `doi:10.4230/LIPIcs.FSTTCS.2019.23`.

28 Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 367–384, New York, NY, USA, 2012. ACM. `doi:10.1145/2213977.2214013`.

29 Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Constant factor approximation for balanced cut in the pie model. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 41–49, New York, NY, USA, 2014. ACM. `doi:10.1145/2591796.2591841`.

30 Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Learning communities in the presence of errors. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1258–1291, Columbia University, New York, New York, USA, 2016. PMLR. URL: `http://proceedings.mlr.press/v49/makarychev16.html`.

31 Pasin Manurangsi. Almost-polynomial ratio eth-hardness of approximating densest k-subgraph. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 954–961, 2017. `doi:10.1145/3055399.3055412`.

32 Theo McKenzie, Hermish Mehta, and Luca Trevisan. A new algorithm for the robust semi-random independent set problem. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 738–746, 2020. `doi:10.1137/1.9781611975994.45`.

33 F. McSherry. Spectral partitioning of random graphs. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 529–537, 2001.

34 Andrea Montanari. Finding one community in a sparse graph. *Journal of Statistical Physics*, 161, February 2015. doi:10.1007/s10955-015-1338-2.

35 Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC '10, pages 755–764, New York, NY, USA, 2010. ACM. doi:10.1145/1806689.1806792.

36 Prasad Raghavendra, David Steurer, and Prasad Tetali. Approximations for the isoperimetric and spectral profile of graphs and related parameters. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, page 631–640, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1806689.1806776.

37 Anand Srivastav and Katja Wolf. Finding dense subgraphs with semidefinite programming. In *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization*, APPROX '98, pages 181–191, London, UK, UK, 1998. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=646687.702946.