CONSTANT-FACTOR LOCAL APPROXIMATION FOR BIPARTITE DENSEST SUBGRAPH

Anonymous authors
Paper under double-blind review

ABSTRACT

Given a vertex v in a large bipartite graph G, can one quickly find a small densely connected region near v? This is the algorithmic core of many fraud detection, community detection, and recommender systems. One would like a *local* algorithm, i.e., an algorithm that probes G in only a small number of locations to find such a region. Our main contribution is the first constant-factor local approximation, independent of the maximum degree of the graph, for the classical notion of bipartite density: if the sides of a bipartite subgraph are S and T, its density is $d(S,T) \coloneqq e(S,T)/\sqrt{|S||T|}$, where $e(S,T)$ is the number of edges between S and T. We consider *cohesive* dense subgraphs, since every dense subgraph contains a cohesive core of at least the same density. For such a cohesive dense subgraph $S \dot\cup T$ with at most k vertices per side, our algorithm, starting from any vertex in $S \dot\cup T$, uses $O_\epsilon(k^{4+\epsilon})$ neighbor queries to find a subgraph $X \dot\cup Y$ of density $\Omega_\epsilon(d(S,T))$ on $O_\epsilon(k^{2+\epsilon})$ vertices, for every fixed $\epsilon > 0$. Our result improves upon the result of Andersen (2010), whose approximation guarantee depends logarithmically on the maximum degree. We complement our positive result with information-theoretic and computational limitations showing that the quadratic dependence on k in the output size is essentially unavoidable for constant-factor approximation. Our main technical innovation is an adaptation of the Lovász–Simonovits curve to the bipartite density problem, based on weak-ℓ_p norm bounds.

1 INTRODUCTION

Identifying dense bipartite subgraphs is a fundamental primitive across machine learning, network science, and web data mining. Bipartite graphs model interactions between two kinds of entities: accounts and products in e-commerce, users and multimedia content on social platforms, authors and venues in bibliometrics, or queries and destination URLs in web search (Dhillon, 2001). In fraud detection and cybersecurity, researchers search for dense bipartite subgraphs to detect botnets, astroturfing campaigns, and lockstep review manipulation, where malicious actors coordinate attacks against targeted subsets of resources (Beutel et al., 2013; Jiang et al., 2016; Hooi et al., 2017; Shin et al., 2018). In web mining, dense subgraphs correspond to link-spam farms (Gibson et al., 2005) and authoritative topic hubs (Kleinberg, 1999). In computational biology, dense subgraphs are used to identify regulatory motifs in genomic DNA (Fratkin et al., 2006), analyze drug–target interaction graphs, and uncover multi-functional complexes in gene annotation networks (Saha et al., 2010).

To quantify how tightly two sets of vertices are connected, we use the bipartite density objective (Kannan & Vinay, 1999; Andersen, 2010). Throughout, $G = (V, E)$ is a finite simple undirected bipartite graph with a fixed bipartition $V = L \dot\cup R$ and maximum degree Δ. For nonempty vertex sets $S \subseteq L$ and $T \subseteq R$ with $e(S,T)$ edges between them, their density is

$$d(S,T) \coloneqq \frac{e(S,T)}{\sqrt{|S||T|}} = \sqrt{\frac{e(S,T)}{|S|} \cdot \frac{e(S,T)}{|T|}},$$

the geometric mean of the average degree from S into T and the average degree from T into S.

The maximum of $d(S,T)$ over all nonempty sets S, T on opposite sides can be computed in polynomial time by linear programming (Charikar, 2000). However, on modern web-scale networks—frequently spanning hundreds of millions of vertices and billions of edges (Qin et al., 2015)—such

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

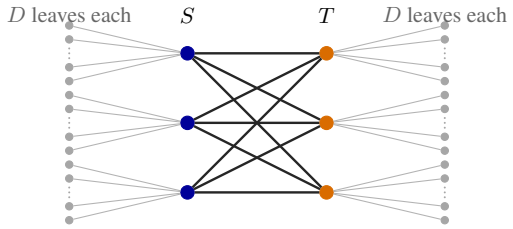


Figure 1: A complete bipartite subgraph (S, T) with k vertices in S (blue) and k in T (orange), each with $D \geq 1$ leaves (gray) outside $S \dot{\cup} T$. Here $d(S, T) = k$ for every D. Each vertex of $S \dot{\cup} T$ has D of its $k + D$ edges leaving $S \dot{\cup} T$, so the conductance of $S \dot{\cup} T$ is at least $D/(k + D)$, which tends to one as $D/k \to \infty$. Every star in this graph has density at most $\sqrt{k + D}$, which is $o(k)$ for $D = o(k^2)$.

global algorithms are computationally prohibitive, as their running time and memory footprint scale with the size of the entire graph. We therefore study the local formulation (Andersen, 2010): given a starting vertex v and a target size parameter k, find a dense bipartite subgraph in the vicinity of v using a number of queries and running time that depend solely on k, completely independent of the graph size $|V|$ and maximum degree Δ.

Classical local clustering algorithms for low-conductance sets (Spielman & Teng, 2013; Andersen et al., 2006; Allen-Zhu et al., 2013) seek sets with few boundary edges relative to their total degree. In contrast, $d(S, T)$ is independent of edges leaving $S \dot{\cup} T$. Dense bipartite subgraphs in interaction graphs often have many edges to the rest of the graph—for example, when fraudulent accounts add large numbers of legitimate interactions as "camouflage" (Hooi et al., 2017). With these camouflage edges, the cut conductance of $S \dot{\cup} T$ can approach 1. Standard random-walk and diffusion methods can then quickly leave the target, so we need local algorithms designed specifically for the bipartite density objective (Figure 1).

More formally, the local algorithm receives a starting vertex v, an integer $k \geq 1$, and adjacency-list access to G, as in Andersen (2010). We count neighbor queries: we read a specified entry of an adjacency list, or receive $\perp$ past its end. The guarantee is measured against unknown cohesive sets $S \subseteq L$, $T \subseteq R$ (Definition 1.1) with at most k vertices on each side. We require the algorithm to succeed from every starting vertex in $S \dot{\cup} T$, using a number of queries independent of $|V|$. An approximation ratio $\alpha \geq 1$ means returning nonempty sets $X \subseteq L$, $Y \subseteq R$ with $d(X, Y) \geq d(S, T)/\alpha$. For a single vertex, we write $e(u, T) \coloneqq e(\{u\}, T)$ and $e(S, w) \coloneqq e(S, \{w\})$.

We ask for approximation, not recovery: the union of the returned sets need not contain the starting vertex or intersect $S \dot{\cup} T$ (after adding v to the returned set on its side, the density is at least $d(X, Y)/\sqrt{2}$; see Lemma B.3). A star with at least $d(S, T)^2$ leaves has density at least $d(S, T)$ and is therefore an acceptable output.

Andersen's algorithm can be implemented using $O(\Delta k^2)$ neighbor queries and returns sets X, Y with $O(\Delta k^2)$ vertices each (this follows from Andersen's analysis (Andersen, 2010, proof of Theorem 2); see also Section C.2). Its approximation ratio is bounded by $O(1 + \log_2(\Delta/d(S, T)))$ (Andersen, 2010, Theorem 1). This bound is constant only when $d(S, T) = \Omega(\Delta)$. For the complete bipartite subgraph of Figure 1 with D camouflage edges per vertex, the lower bound on the returned density is $\Omega(k/(1+\log_2((k+D)/k)))$, which decreases as D grows even though $d(S, T) = k$. The density is invariant under the addition of edges outside the subgraph induced by $S \dot{\cup} T$, and the guarantee is not. Whether this dependence on the maximum degree is necessary for local approximation was open.

1.1 Our results

Our main result is a constant-factor local approximation algorithm for the bipartite densest subgraph problem. We cannot guarantee local success from every vertex of $S \dot{\cup} T$ for arbitrary target sets S, T, since a vertex in $S \dot{\cup} T$ may have few or no edges into the opposite target set. We therefore require every target vertex to have at least half the average degree of its side within the target subgraph.

Definition 1.1 (Cohesive sets). *Nonempty sets $S \subseteq L$, $T \subseteq R$ with $e(S,T) > 0$ are* cohesive *if $e(u,T) \geq \frac{e(S,T)}{2|S|}$ for all $u \in S$, and $e(S,w) \geq \frac{e(S,T)}{2|T|}$ for all $w \in T$.*

We use the term *cohesiveness* to formalize the minimum-degree structure already implicit in Andersen's proof of his main theorem (Andersen, 2010, Theorem 1). For any sets $S \subseteq L$, $T \subseteq R$ with $e(S,T) > 0$, there are cohesive subsets $S^* \subseteq S$, $T^* \subseteq T$ of at least the same density, $d(S^*,T^*) \geq d(S,T)$ (Lemma A.1). For arbitrary S,T, our setting is exactly that of Andersen, whose algorithm succeeds from every vertex of subsets that are selected by a minimum-degree argument and contain at least half of the edges (see the discussion in Corollary A.2).

Theorem 1.2 (Main result). *For every $\epsilon > 0$, all cohesive sets $S \subseteq L$, $T \subseteq R$ with $|S|, |T| \leq k$, and every starting vertex $v \in S \dot\cup T$, the deterministic algorithm $\mathsf{LocalDense}_{2+\epsilon}(G, v, k)$ (Algorithm 2) returns nonempty sets $X \subseteq L$, $Y \subseteq R$ with the following guarantees:*

- *(i)* Locality and complexity: *The output (X,Y) has $O_\epsilon(k^{2+\epsilon})$ vertices, and the algorithm makes $O_\epsilon(k^{4+\epsilon})$ neighbor queries and runs in time $\widetilde{O}_\epsilon(k^{4+\epsilon})$.*
- *(ii)* Constant-factor approximation: $d(X,Y) \geq \frac{\sqrt{3}}{16\sqrt{e}} \frac{\epsilon}{1+\epsilon}\, d(S,T)$, *independent of edges outside the subgraph induced by $S \dot\cup T$.*

We complement our positive results by proving limitations on the output size:

Theorem 1.3 (Limitations on the output size, informal). *No deterministic algorithm with neighbor-query access that, for all cohesive sets $S \subseteq L$, $T \subseteq R$ with k vertices on each side, returns nonempty sets $X \subseteq L$, $Y \subseteq R$ with $d(X,Y) = \Omega(d(S,T))$ from at least one vertex in $S \dot\cup T$ can guarantee an output bound of $o(k^2)$ vertices on each side if either:*

- *(i)* Unconditional: *Its number of queries depends only on k; or*
- *(ii)* Computational: *Assuming the Strongish Planted Clique Hypothesis (SPCH), its running time is at most $f(k)\,|V|^c$ for a constant c and a computable function f of k alone.*

Up to the factor k^ϵ, our output size is therefore the best possible in both settings. We discuss the main result in Section 2 and the limitations in Section 3.

1.2 Our technique

From power iteration to prefix sweeps. Our starting point is Andersen's pruned power iteration (Andersen, 2010). We maintain a nonnegative vector x indexed by vertices and multiply it by the adjacency matrix, setting $y(w) \coloneqq \sum_u x(u)$, where the sum is over the neighbors of w. For now, we set aside rescaling and pruning. The mass of x on a set Q is $\sum_{u \in Q} x(u)$. Suppose every vertex of S has at least a neighbors in T and every vertex of T at least b in S. After one multiplication, the mass of y on T is at least a times the mass of x on S, and after two, the mass on S is at least ab times its original value. These bounds hold even with edges leaving the target, since all coordinates are nonnegative. By cohesiveness, there exists powers of two $a, b \in [1,k]$, with $\sqrt{ab} > d(S,T)/4$ (see Section 2.2; we remove this choice later). We cannot locate a dense pair from the growth of target mass alone: the target is unknown, and mass outside it may grow faster. Andersen extracts a dense pair from growth of the Euclidean norm. Instead we sort consecutive vectors and consider their initial segments, or *prefixes*. We seek a Lovász–Simonovits-type *win–win* (Spielman & Teng, 2013; Andersen et al., 2006): either a sweep finds a dense prefix pair, or we obtain upper bounds on prefix sums that eventually become incompatible with the mass that remains on the target. Here set size means number of vertices, rather than volume.

The choice of norm. Which bound on prefix sums makes this alternative useful? Write $P_i(x)$ for the vertices with the i largest positive coordinates, breaking ties by identifier, and $I_x(m) \coloneqq \max_{|Q| \leq m} \sum_{u \in Q} x(u)$. A natural choice is the Euclidean norm, since $I_x(m) \leq \|x\|_2 \sqrt{m}$. It turns out, however, that with only this inequality we incur a logarithmic loss when summing comparable bounds over dyadic ranges of prefix sizes. Changing the sweep alone is therefore not enough. We instead fix $p > 2$ and use the *weak-ℓ_p norm* (Bennett & Sharpley, 1988, Chapter 4)

$$\|x\|_{p,\infty} \coloneqq \sup_{m \in \{1,2,\dots\}} \frac{I_x(m)}{m^{1-1/p}}\,.$$

The bounds over dyadic ranges now form a geometric series, so we incur only a p-dependent factor when summing them (a detailed explanation of this comparison can be found in Section C.3). More precisely, let $y = xA$, where $x \geq 0$ is supported on L and A is an $L \times R$ adjacency matrix with column sums at most B_∞. Let $\sigma > 0$ be the density we aim to guarantee in the returned pair. For a prefix $P_m(y)$, if $d(P_i(x), P_m(y)) \leq \sigma$ for every prefix $P_i(x)$, then

$$I_y(m) \leq C_p \, \sigma^{2/p} B_\infty^{1-2/p} \|x\|_{p,\infty} m^{1-1/p}, \qquad C_p \coloneqq \frac{p-1}{p-2}\,.$$

If the inequality fails for $P_m(y)$, we find a dense pair by sweeping the prefixes of x against $P_m(y)$. If no prefix pair exceeds density σ, we take the supremum over m to bound $\|y\|_{p,\infty}$ (Sections A.3 and 2.1).

Removing the degree dependence. The logarithm is gone, but the column sums remain. For the full adjacency matrix, we only know the bound $B_\infty = \Delta$ on the column sums, where Δ is the maximum degree. Since $\sqrt{ab} > d(S,T)/4$, we obtain a constant-factor approximation if we can guarantee density σ equal to a fixed fraction of $\sqrt{ab}$, depending only on p. The upper bound on the norm after two steps has factor $C_p^2 \sigma^{4/p} \Delta^{2-4/p}$, compared with the factor ab for the mass on S. To make the norm factor smaller than ab, we need $\sigma < C_p^{-p/2} \sqrt{ab} (\sqrt{ab}/\Delta)^{(p-2)/2}$, at the cost of an additional approximation factor $(\Delta/\sqrt{ab})^{(p-2)/2}$.

After rescaling, we want the target mass to grow while the weak norm decreases over pairs of steps. We use *peeling* to bound receiving degrees while keeping enough target edges. In a step from L to R, we repeatedly remove a receiving vertex with at most $b/2$ remaining edges and select those edges, or a vertex in the current support with at most $a/4$ remaining edges and discard those edges. If vertices remain but neither removal is possible, their density exceeds $\sqrt{ab/8}$ (Section 2.2). If every vertex is removed, each receiving vertex has at most $b/2$ selected edges, while at least $3a/4$ target edges were selected at each current target vertex. Let A be the adjacency matrix of the selected edges and set $y = (2/b)xA$, so that the weighted column sums are at most one. Choosing σ appropriately, we either find a prefix pair of density greater than σ, or

$$\sum_{w \in T} y(w) \geq \frac{3a}{2b} \sum_{u \in S} x(u), \qquad \|y\|_{p,\infty} \leq \left(\frac{3a}{4b}\right)^{1/p} \|x\|_{p,\infty}\,.$$

For the reverse step, we exchange a and b. If no dense pair is returned in either step, then after two steps without pruning the mass on S is at least $9/4$ times its earlier value, while the weak norm is at most $(3/4)^{2/p}$ times its earlier value. Crucially, neither factor now depends on Δ (Section 2.3).

The local search. We are now ready to combine the ideas above and present the algorithm. So far, we have assumed that suitable degree bounds a, b are known. In reality, since they are unknown, we run a search for every pair of powers of two $a, b \in [1, k]$ and return the densest candidate. Assume $v \in L$, exchanging the sides if necessary. Each search starts with $x_0 = \mathbb{1}_v$, the vector equal to one at v and zero elsewhere, and alternates between the sides. At step t, the algorithm peels the edges incident to the support of x_t and, unless a dense pair is returned, performs the rescaled multiplication and sets coordinates below δ to zero. We choose δ to keep the support small without losing too much target mass.

We abandon the guess if the pruned vector z is zero and otherwise accept $x_{t+1} = z$ only if its weak norm satisfies the bound from the inequalities above, which decreases over pairs of steps. If this bound is exceeded, we sweep x_t against a prefix of z whose sum exceeds the corresponding bound. Its vertices and sum are unchanged by pruning, so we find a dense pair.

For the degree bounds chosen using cohesiveness and $v \in S$, we measure target mass with a factor b/a on T and no factor on S. Before pruning, this mass is at least $3/2$ times its previous value, and we choose δ so that the loss during pruning is at most $1/2$. The weighted mass therefore stays at least one, its initial value, and the vector never becomes zero. At every even step reached by the search,

$$1 \leq \sum_{u \in S} x_t(u) \leq \|x_t\|_{p,\infty} k^{1-1/p} \leq (3/4)^{t/p} k^{1-1/p}\,.$$

After $O_p(\log_2(2k))$ steps, the right-hand side would be less than one, so the search must return a pair of density at least σ (Section 2.3).

1.3 Related Work

Spielman & Teng (2013) used the Lovász–Simonovits curve (Lovász & Simonovits, 1990; 1993) to analyze their local partitioning algorithm Nibble, and Andersen et al. (2006) adapted this analysis to personalized PageRank. Other local graph-partitioning algorithms use heat-kernel PageRank (Chung, 2009) or evolving sets (Andersen et al., 2016), and local clustering by p-norm flow diffusions and p-norm cuts (Fountoulakis et al., 2020; Liu & Gleich, 2020) has been extended to vertex attributes and noisy labels (Yang & Fountoulakis, 2023; Back de Luca et al., 2024). Local algorithms for dense bipartite-like subgraphs (Peng, 2012; Li & Peng, 2013; Macgregor & Sun, 2021) minimize the bipartiteness ratio, a cut objective analogous to conductance.

Local and seeded variants of densest subgraph have also been studied from a practical perspective. When densest subgraphs are extracted sequentially, fragmented sub-regions of a single dominant global core may be returned while distinct local communities are missed. Qin et al. (2015) therefore introduced *locally densest subgraphs* (LDS), based on ρ-compact subgraphs, with structural core bounds and pruning techniques for exact verification on billion-edge graphs, but their algorithm traverses the whole graph. Dai et al. (2022) introduced the *anchored densest subgraph* (ADS) problem, with explicit edge or vertex penalties for expanding beyond a specified reference set, and solved it via localized parametric flow networks. Huang et al. (2024) generalized this formulation to hypergraphs using negative supermodular optimization, with strongly localized routines when the penalization parameter is sufficiently large. For dense-block discovery in fraud detection, Hooi et al. (2017) gave a camouflage-resistant algorithm for the whole graph. The local problem of Andersen (2010) was revisited by Liu (2026, Theorems 1.2.2 and 1.2.3), who obtains an output size of $O(k^2)$ with Andersen's approximation ratio, and $O(k^{1+\epsilon})$ with a ratio that also depends on $|V|$. The techniques involved are different than ours and are based on spectral analysis, degree sorting, and majorization. Our algorithm, with $\epsilon = 1/\lceil \log_2(2k) \rceil$, also returns $O(k^2)$ vertices, using $O(k^4)$ neighbor queries with an $O(\log_2(2k))$ approximation ratio (Corollary B.4).

2 The local algorithm and its analysis

Our algorithm is based on an alternating power iteration from the starting vertex: with $x_0 \coloneqq \mathbb{1}_v$, it repeatedly multiplies the current vector by the adjacency matrix of a subgraph of G and then prunes, alternating between the two sides. For suitable degree guesses, the mass inside a cohesive target grows by a factor of at least $9/4$ over two weighted steps, before accounting for pruning. The target is unknown to the algorithm, and the mass outside it may grow as well, so we cannot identify dense sets from growth alone. For the analysis, we compare prefixes: either prefixes P, Q of consecutive vectors have density $d(P, Q) > \sigma$ and the algorithm returns them, or the next prefix sums are bounded in terms of the current ones, with geometrically decreasing bounds over pairs of steps. This cannot last forever: the target has at most k vertices per side, but its mass on the starting side stays at least one at every even step.

We use neighbor queries with fixed adjacency-list orders and distinct vertex identifiers. Vectors are row vectors indexed by vertices, with indicators $\mathbb{1}_S$, $\mathbb{1}_v \coloneqq \mathbb{1}_{\{v\}}$ and support $\operatorname{supp}(x) \coloneqq \{u : x(u) \neq 0\}$. Vectors are restricted or extended by zero to match matrix indices. For any subgraph H of G, numbers of edges and densities in H are denoted by $e_H(S, T)$ and $d_H(S, T)$, and $N_G(U)$ denotes the set of neighbors of $U \subseteq V$ in G.

2.1 Prefix sweeps

In a step of the iteration from L to R, the current vector $x \geq 0$ is supported on L and the next vector is $y \coloneqq xA$, where A is the adjacency matrix of a subgraph of G, so that each coordinate $y(w)$ is the sum of $x(u)$ over the neighbors u of w in this subgraph. To compare the largest coordinates of y with those of x, we adapt the Lovász–Simonovits curve (Lovász & Simonovits, 1990; 1993) to density, as outlined in Section 1.2: in place of the curve we use $I_x(m)$, the largest mass of the current vector on m vertices, and in place of a sweep set of low conductance we look for prefixes P, Q with $d(P, Q) > \sigma$, where $\sigma > 0$ is a density threshold, fixed in (4).

Definition 2.1 (Prefix sums and the weak norm [Spielman & Teng, 2013, equation (3); Bennett & Sharpley, 1988, Chapter 4]). *For a vector $x \geq 0$ indexed by a finite set and $\ell \coloneqq |\operatorname{supp}(x)|$, write $x_{(1)}, \ldots, x_{(\ell)}$ for its positive coordinates in nonincreasing order, with ties broken by identifier, and*

set $x_{(i)} \coloneqq 0$ for $i > \ell$. For $1 \le i \le \ell$, the prefix $P_i(x)$ consists of the first i vertices in this order. For $p \ge 2$, the prefix sums and weak-ℓ_p norm are defined by

$$I_x(m) \coloneqq \max_{|P| \le m} \sum_{u \in P} x(u) = \sum_{i=1}^{m} x_{(i)}, \qquad \|x\|_{p,\infty} \coloneqq \sup_{s>0} \frac{I_x(s)}{s^{1-1/p}}, \tag{1}$$

where m is a nonnegative integer and I_x is extended to real arguments $s \ge 0$ by linear interpolation between consecutive integers, with $I_x(0) = 0$. In particular, $I_x(s) \le \|x\|_{p,\infty} s^{1-1/p}$ for every $s > 0$.

Definition 2.2 (Column-sum bound)**.** *A nonnegative matrix A has* column-sum bound B_∞ *if $\sum_u A_{uw} \le B_\infty$ for every column w.*

After sorting y, we take a prefix $P_m(y)$, whose sum is $I_y(m)$, and in a *sweep* we look for a prefix $P_i(x)$ with $d(P_i(x), P_m(y)) > \sigma$. If there is none, then there are at most $\sigma\sqrt{im}$ edges between $P_i(x)$ and $P_m(y)$ for every i. For $p > 2$ and a column-sum bound B_∞ for A (Definition 2.2),

$$I_y(m) \le C_p\, \sigma^{2/p} B_\infty^{1-2/p} \|x\|_{p,\infty} m^{1-1/p}, \qquad \text{where } C_p \coloneqq \frac{p-1}{p-2}. \tag{2}$$

This bound is a special case of Proposition A.4, proved in Section A.3 by summation by parts. If $I_y(m)$ exceeds the right side of (2), then $d(P_i(x), P_m(y)) > \sigma$ for some prefix $P_i(x)$, which the algorithm finds in the sweep. If $d(P_i(x), P_m(y)) \le \sigma$ for all prefixes $P_i(x), P_m(y)$, then $\|y\|_{p,\infty} \le C_p \sigma^{2/p} B_\infty^{1-2/p} \|x\|_{p,\infty}$. The bound holds only for $p > 2$: for $p = 2$, we obtain a logarithmic factor in place of C_p, as in the guarantee of Andersen (2010), to which we compare ours in Section C. The bound also contains the factor $B_\infty^{1-2/p}$. For the adjacency matrix of G itself, whose column sums are the degrees, we can only take $B_\infty \coloneqq \Delta$, so that the bound contains a factor $\Delta^{1-2/p}$, even though $d(S,T)$ is independent of edges outside the subgraph induced by $S \dot\cup T$. To keep σ a fixed fraction of $d(S,T)$, we therefore need to bound B_∞ independently of Δ.

2.2 Vertex peeling

We select the edges of each multiplication by peeling: we repeatedly remove a vertex of low degree, selecting its remaining edges if it is on the side of the next vector, so that its column sum is small, and discarding them if it is on the side of the current vector, so that only a few of its edges are discarded. For each target vertex in the support, most edges into the opposite target set are selected for multiplication. The thresholds for low degree are fractions of the degrees a, b guaranteed by cohesiveness.

Lemma 2.3 (Degree guesses for cohesive sets)**.** *For cohesive sets $S \subseteq L$, $T \subseteq R$, there are powers of two $a, b \ge 1$, depending only on S, T, such that $e(u,T) \ge a$ for every $u \in S$, $e(S,w) \ge b$ for every $w \in T$, and $\sqrt{ab} > d(S,T)/4$. For every k with $|S|, |T| \le k$, $a, b \le k$, $a|S| \le 2bk$ and $b|T| \le 2ak$.*

The proof is in Section A.1. Since a, b depend on unknown S, T, the algorithm tries all pairs of powers of two in $[1, k]$.

For a step from L to R with $U \coloneqq \mathrm{supp}(x)$, the *neighborhood graph* of U is $H \coloneqq (U \dot\cup W, F)$, where $W \coloneqq N_G(U)$ is the set of neighbors of U and F is the set of edges incident to U. (For a step from R to L, we argue in the same way after exchanging a and b.) We seek a set $F' \subseteq F$ of selected edges such that every $w \in W$ is incident to at most $b/2$ selected edges and every $u \in U$ to at most $a/4$ discarded edges. With each selected edge assigned weight $2/b$, the weighted matrix has column-sum bound $B_\infty = 1$. Each $u \in S \cap U$ has at least a edges into T, of which at most $a/4$ are discarded, so at least $3a/4$ of its edges into T belong to F', regardless of its degree in G. Hence

$$\sum_{w \in T} y(w) \ge \frac{3a}{2b} \sum_{u \in S} x(u). \tag{3}$$

Here $\sum_{u \in S \cap U} x(u) = \sum_{u \in S} x(u)$ since $U = \mathrm{supp}(x)$. Peel (Algorithm 1) selects edges by removing low-degree vertices, or returns dense subsets of the vertices it cannot remove.

If Peel removes every vertex, then an edge belongs to F' exactly when its endpoint in W is removed before its endpoint in U (line 5). Thus every $w \in W$ has at most $b/2$ selected edges and every

Algorithm 1 $\mathsf{Peel}(H, x, a, b)$: select edges or return subsets

Input: A bipartite graph $H = (U \dot\cup W, F)$, a nonnegative vector x indexed by U, and $a, b > 0$.
Output: A nonnegative vector y indexed by W, or sets U_*, Y with $\varnothing \neq U_* \subseteq U$ and $\varnothing \neq Y \subseteq W$.

1: $U_* \leftarrow U, \quad W_* \leftarrow W, \quad F' \leftarrow \varnothing$
2: **while** there exists $u \in U_*$ with $e_H(u, W_*) \le a/4$ or $w \in W_*$ with $e_H(U_*, w) \le b/2$ **do**
3: Let q be the vertex with smallest identifier satisfying either condition.
4: **if** $q \in W_*$ **then**
5: $F' \leftarrow F' \cup \{uq \in F : u \in U_*\}$ and $W_* \leftarrow W_* \setminus \{q\}$
6: **else**
7: $U_* \leftarrow U_* \setminus \{q\}$
8: **if** $U_* \dot\cup W_* = \varnothing$ **then return** y indexed by W with $y(w) \coloneqq \frac{2}{b} \sum_{u:\, uw \in F'} x(u)$
9: Let Y consist of the first $\min\left\{|W_*|, \left\lceil \frac{a|U_*|}{2b} \right\rceil\right\}$ vertices of W_* in identifier order.
10: **return** (U_*, Y)

Algorithm 2 Local approximation, for $p > 2$

$\mathsf{LocalDense}_p(G, v, k)$
Input: Neighbor-query access to $G = (L \dot\cup R, E)$, a vertex $v \in L \dot\cup R$, and an integer $k \ge 1$.
Output: Nonempty $X \subseteq L$, $Y \subseteq R$, or $\perp$ if v is isolated.

1: Query the first neighbor u of v.
2: **if** $u = \perp$ **then return** $\perp$
3: **if** $v \in R$ **then** exchange the roles of L and R below, and swap the returned sets
4: $\mathcal{C} \leftarrow \{(\{v\}, \{u\})\}$
5: **for** each $a, b \in \{2^0, 2^1, \ldots, 2^{\lfloor \log_2 k \rfloor}\}$ **do**
6: $r \leftarrow \mathsf{Search}_p(G, v, k, a, b)$
7: **if** $r \neq \perp$ **then** add r (ordered with its current L-subset first) to $\mathcal{C}$
8: **return** $(X, Y) \in \mathcal{C}$ maximizing $d(X, Y)$

$\mathsf{Search}_p(G, v, k, a, b)$
Input: The inputs above, with $v \in L$, and a guess $a, b \in [1, k]$.
Output: Two nonempty vertex sets on opposite sides of G, or $\perp$.

9: Take $\sigma, \delta, (K_t)_{t \ge 0}$ from (4), and set $x_0 \leftarrow \mathbb{1}_v$.
10: **for** $t = 0, 1, 2, \ldots$ **do**
11: $(a_t, b_t) \leftarrow (a, b)$ if t is even, and (b, a) if t is odd
12: **for** each $u \in \operatorname{supp}(x_t)$ **do**
13: Read the adjacency list of u, stopping after $\lceil \sigma^2 \rceil$ entries, and let N_u be the set of neighbors read.
14: **if** $|N_u| = \lceil \sigma^2 \rceil$ **then return** $(\{u\}, N_u)$
15: Let $H \coloneqq (\operatorname{supp}(x_t) \dot\cup W, F)$ be the neighborhood graph of $\operatorname{supp}(x_t)$, with $W \coloneqq N_G(\operatorname{supp}(x_t))$.
16: $r \leftarrow \mathsf{Peel}(H, x_t, a_t, b_t)$
17: **if** r consists of sets U_*, Y **then return** (U_*, Y)
18: $y_{t+1} \leftarrow r, \quad z \leftarrow \operatorname{prune}(\delta, y_{t+1})$
19: **if** $z = 0$ **then return** $\perp$
20: **if** $\|z\|_{p,\infty} \le K_{t+1}^{1/p}$ **then**
21: $x_{t+1} \leftarrow z$
22: **else**
23: Let $m \ge 1$ be the smallest index such that $I_z(m) > K_{t+1}^{1/p} m^{1-1/p}$.
24: **return** $(P_i(x_t), P_m(z))$ for the smallest i with $d_H(P_i(x_t), P_m(z)) > \sigma$, or $\perp$ if none exists

$u \in U$ at most $a/4$ discarded edges, since these are its remaining edges when it is removed (line 2). If vertices remain, then every vertex of U_* has more than $a/4$ neighbors in W_* and every vertex of W_* more than $b/2$ neighbors in U_* (line 2), so that $d_H(U_*, W_*) > \sqrt{ab/8}$. In the search, $|U_*| \le |\operatorname{supp}(x)|$, and the support size is bounded by pruning as in (6), but $|W_*|$ is not. To bound the output size, we replace W_* by a subset Y (line 9) with $|Y| \le a|U_*|/(2b) + 1$. Every vertex of Y still has more than $b/2$ neighbors in U_*, and the density remains above $\sqrt{ab/8}$ (see Lemma A.6(b)).

2.3 The algorithm and its guarantees

$\mathsf{LocalDense}_p$ (Algorithm 2) keeps the edge from v to its first neighbor as the first candidate, so that the output is nonempty with density at least one (lines 1 to 4). By guessing a, b among the powers of two in $[1, k]$ (line 5), we lose only a constant factor in the approximation. To keep the computation

local, Search_p reads at most $\lceil \sigma^2 \rceil$ entries of each adjacency list and returns a star if it finds that many neighbors, since a star with that many leaves has density at least σ (line 14). It accepts the pruned vector as the next one only if it passes the *weak-norm test* of line 20, analyzed below.

For a starting vertex in L and a guess (a, b), the search uses the density threshold σ, the pruning threshold δ, and the weak-norm thresholds $K_t^{1/p}$, where

$$\sigma \coloneqq \frac{\sqrt{3ab}}{4C_p^{p/2}}, \qquad \delta \coloneqq \frac{\max\{1, a/b\}}{4k}, \qquad K_t \coloneqq \begin{cases} (3/4)^t, & t \text{ even}, \\ (a/b)(3/4)^t, & t \text{ odd}. \end{cases} \tag{4}$$

The density threshold σ is a constant fraction of $\sqrt{ab}$, and $\sigma < \sqrt{ab/8}$ since $C_p^p > 2$.

For a step from L to R, write $x \coloneqq x_t$ and $y \coloneqq y_{t+1} = (2/b)xA$, where A is the adjacency matrix of the selected edges and has column-sum bound $B_\infty = b/2$ (line 16). If $d(P_i(x), P_m(y)) \le \sigma$ for every prefix $P_i(x)$, then, using (2), rescaling by $2/b$, and substituting σ from (4), we get

$$I_y(m) \le \Big(\frac{3a}{4b}\Big)^{1/p} \|x\|_{p,\infty} m^{1-1/p}. \tag{5}$$

For $\delta > 0$ and a nonnegative vector y, $\mathrm{prune}(\delta, y)$ is obtained from y by setting every coordinate smaller than δ to zero, and the pruned vector is $z = \mathrm{prune}(\delta, y_{t+1})$ (line 18). For every prefix $P_m(z)$, $P_m(z) = P_m(y_{t+1})$ and $I_z(m) = I_{y_{t+1}}(m)$, so (5) holds with z in place of y as well. Every positive coordinate of z is at least δ, and the sum over any set Q decreases by at most $\delta|Q|$ (Section A.2).

For cohesive sets $S \subseteq L$, $T \subseteq R$, a starting vertex $v \in S$, and the search with the guess (a, b) from Lemma 2.3, we consider the mass on the target side, multiplied by b/a when that side is T. Whenever Peel returns a vector, its weighted target mass before pruning is at least $3/2$ times the previous value, by (3). By Lemma 2.3, the bounds $|S|, |T| \le k$, and the choice of δ in (4),

$$|S| \le \min\{k, 2bk/a\} \le \frac{1}{2\delta}, \qquad (b/a)|T| \le \min\{(b/a)k, 2k\} \le \frac{1}{2\delta}.$$

Thus at most $\max\{\delta|S|, (b/a)\delta|T|\} \le 1/2$ is lost during pruning. The initial weighted target mass is one, so by induction it remains at least one after every pruning step reached for this guess. In particular, $z \neq 0$ at each such step.

The search accepts z as the next vector only if $\|z\|_{p,\infty} \le K_{t+1}^{1/p}$, so every x_t satisfies $\|x_t\|_{p,\infty} \le K_t^{1/p}$ (lines 20 and 21). The thresholds are chosen to match (5): starting from $K_0 = 1$, $K_t^{1/p}$ is multiplied by $(3a/(4b))^{1/p}$ in each step from L to R and by $(3b/(4a))^{1/p}$ in each step from R to L, as specified for K_t in (4). Hence, if $d(P_i(x_t), P_m(z)) \le \sigma$ for all prefixes $P_i(x_t), P_m(z)$, then, by (5) and $\|x_t\|_{p,\infty} \le K_t^{1/p}$, $I_z(m) \le K_{t+1}^{1/p} m^{1-1/p}$ for every integer $1 \le m \le |\operatorname{supp}(z)|$. By Lemma A.3(a), the supremum defining $\|z\|_{p,\infty}$ is attained at such an integer, so z passes the test. Over two steps, the threshold decreases by a factor of $(3/4)^{2/p}$. If z fails the test, Search_p finds the shortest prefix $P_m(z)$ whose sum exceeds $K_{t+1}^{1/p} m^{1-1/p}$ (line 23). Since the bound (5) fails for this prefix, some prefix $P_i(x_t)$ has $d(P_i(x_t), P_m(z)) > \sigma$, and the search returns these two prefixes (line 24).

Output size and locality. Every positive coordinate of x_t is at least δ and $\|x_t\|_{p,\infty} \le K_t^{1/p}$, so, by the choice of parameters in (4),

$$|\operatorname{supp}(x_t)| \le K_t \delta^{-p} \le (4k)^p (3/4)^t, \qquad \sum_t |\operatorname{supp}(x_t)| \le 4(4k)^p, \tag{6}$$

and each set returned by a sweep or by peeling has at most $(4k)^p$ vertices (Lemma B.1(d)), so the output has $O_p(k^p)$ vertices. Every vertex of $\operatorname{supp}(x_{t+1})$ has a neighbor in $\operatorname{supp}(x_t)$, so the sets returned at step t lie within distance $t+1$ of v, where $t \le p \log_{4/3}(4k)$ by (6). In particular, for $p = 2 + \epsilon$, every returned vertex lies within distance $O_\epsilon(\log_2(2k))$ of v (Lemma B.3).

The algorithm makes $O_p\big(k^p \min\{(\Delta+1)\log_2^2(2k),\ k^2\}\big)$ neighbor queries (Section B.2).

Constant approximation. We prove that, for every $p > 2$ and S, T, k, v as in Theorem 1.2, the output (X, Y) of $\mathsf{LocalDense}_p(G, v, k)$ satisfies

$$d(X, Y) > \frac{\sqrt{3}}{16}\, C_p^{-p/2}\, d(S, T). \tag{7}$$

We choose (a, b) as in Lemma 2.3 and suppose first that $v \in S$. By the induction above, $z \neq 0$ at every pruning step reached for this guess. Thus Search_p terminates with sets of density at least σ by Lemma B.1(c,d). Since $\sqrt{ab} > d(S, T)/4$, their density satisfies (7). For $v \in T$, we exchange L, R and a, b. All other searches also terminate by (6), and $\mathsf{LocalDense}_p$ returns the densest candidate. For $\epsilon > 0$ and $p \coloneqq 2 + \epsilon$, we obtain the guarantees of Theorem 1.2, with approximation ratio $O(1 + 1/\epsilon)$, by direct calculation from (6), (7), and (15); see Section B.3 for details.

3 Limitations on improving the output size

For a target with at most k vertices per side, our algorithm returns $O_\epsilon(k^{2+\epsilon})$ vertices with a constant approximation ratio. Can the output be reduced to $O(k)$ or $o(k^2)$ while keeping the approximation ratio constant? The threshold k^2 appears because we ask for approximation rather than recovery: a star with k^2 leaves has density k and is as dense as any target with at most k vertices per side, whereas a star with $o(k^2)$ leaves has density $o(k)$. We rule out output size $\phi(k) = o(k^2)$ per side in two settings:

- *Restricted queries.* If the algorithm is deterministic and its number of queries is bounded by a function of k alone, or more generally is $o(\Delta)$ for every fixed k, while computation between queries is unrestricted, then there is an instance where it fails to return a constant approximation. We run the algorithm from the centers of many disjoint stars with Δ leaves each, choose k centers per side so that each output from a chosen center contains at most one chosen center per side, and hide a complete k-by-k bipartite subgraph on the chosen centers among the unread adjacency-list entries. Since the algorithm is deterministic and never reads the changed entries, it behaves the same regardless of that change. Each output from a chosen center has at most one edge in the target, and its other edges lie in disjoint stars, so it has fewer than $2\phi(k)$ of them and density at most $\sqrt{2\phi(k)}+1 = o(k)$, whereas the target has density k. The formal statement is Theorem D.2, proved in Section D.1.
- *Restricted running time.* If the algorithm may read the entire graph but must run in time $f(k)|V|^{O(1)}$ for a computable function f of k, the limitation is that we need to find not only a dense subgraph, but also a *small* one. For a graph J promised to contain a k-clique, the two copies S, T of the clique in its bipartite double cover have $d(S, T) = k - 1$. From returned sets X, Y with $d(X, Y) = \Omega(k)$ and $|X|, |Y| \leq \phi(k)$, we extract a set U of k vertices of J spanning $\Omega\big(k^3/(k + \phi(k))\big)$ edges. Relative to the clique optimum $\binom{k}{2}$, the approximation ratio for Densest k-Subgraph would be $O(1 + \phi(k)/k) = o(k)$, which under the Strongish Planted Clique Hypothesis (SPCH) of Manurangsi et al. (2021) is impossible in this running time, even with randomization. The formal statement is Theorem D.8, proved in Section D.2.2, where we also treat randomized algorithms. Under ETH, we obtain a weaker output-size bound by the same reduction (Theorem D.12, proved in Section D.2.3).

4 Conclusion

We give the first constant-factor local approximation algorithm for the bipartite densest subgraph problem, with output size, queries and running time depending on k alone (Theorem 1.2). Like the density itself, the guarantee does not depend on edges outside the subgraph induced by $S \dot\cup T$.

The main technical question left unanswered is the following: while *sub*-quadratic output is impossible in the settings of Section 3, quadratic output is not ruled out, and we leave open whether it is possible to achieve a constant-factor approximation with output of size $O(k^2)$.

Beyond improving the output size and approximation ratio, is it possible, under suitable assumptions, to ensure that the returned sets overlap with the target sets? Such results are known in the case of conductance, i.e., the classical application of the Lovász–Simonovits curve—see, for example, Yang & Fountoulakis (2023); Back de Luca et al. (2024).

AI USE STATEMENT

In this work, we used generative AI tools for brainstorming, writing and editing. We have reviewed all AI-assisted work, and constructed the main arguments in the paper ourselves. We checked the LLM-generated ideas for potential plagiarism through a manual literature survey. We take responsibility for the final content of this work, including its presentation, text, claims and proofs.

REFERENCES

Zeyuan Allen-Zhu, Silvio Lattanzi, and Vahab Mirrokni. A local algorithm for finding well-connected clusters. In *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*, volume 28 of *Proceedings of Machine Learning Research*, pp. 396–404, 2013. URL https://proceedings.mlr.press/v28/allenzhu13.html.

Reid Andersen. A local algorithm for finding dense subgraphs. *ACM Transactions on Algorithms*, 6 (4):60:1–60:12, 2010. doi: 10.1145/1824777.1824780. URL https://doi.org/10.1145/1824777.1824780.

Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using PageRank vectors. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, pp. 475–486, 2006. doi: 10.1109/FOCS.2006.44. URL https://doi.org/10.1109/FOCS.2006.44.

Reid Andersen, Shayan Oveis Gharan, Yuval Peres, and Luca Trevisan. Almost optimal local graph clustering using evolving sets. *Journal of the ACM*, 63(2):15:1–15:31, 2016. doi: 10.1145/2856030. URL https://doi.org/10.1145/2856030.

Artur Back de Luca, Kimon Fountoulakis, and Shenghao Yang. Local graph clustering with noisy labels. In *12th International Conference on Learning Representations (ICLR 2024)*, 2024. URL https://openreview.net/forum?id=89A5c6enfc.

Colin Bennett and Robert C. Sharpley. *Interpolation of Operators*, volume 129 of *Pure and Applied Mathematics*. Academic Press, 1988. ISBN 978-0-12-088730-9.

Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. CopyCatch: Stopping group attacks by spotting lockstep behavior in social networks. In *Proceedings of the 22nd International Conference on World Wide Web (WWW 2013)*, pp. 119–130, 2013. doi: 10.1145/2488388.2488400. URL https://doi.org/10.1145/2488388.2488400.

Yonatan Bilu and Nathan Linial. Lifts, discrepancy and nearly optimal spectral gap. *Combinatorica*, 26(5):495–519, 2006. doi: 10.1007/s00493-006-0029-7. URL https://doi.org/10.1007/s00493-006-0029-7.

Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *APPROX 2000*, volume 1913 of *Lecture Notes in Computer Science*, pp. 84–95. Springer, 2000. doi: 10.1007/3-540-44436-X_10. URL https://doi.org/10.1007/3-540-44436-X_10.

Fan Chung. A local graph partitioning algorithm using heat kernel PageRank. *Internet Mathematics*, 6(3):315–330, 2009. doi: 10.1080/15427951.2009.10390643. URL https://doi.org/10.1080/15427951.2009.10390643.

Yizhou Dai, Miao Qiao, and Lijun Chang. Anchored densest subgraph. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD 2022)*, pp. 1200–1213, 2022. doi: 10.1145/3514221.3517890. URL https://doi.org/10.1145/3514221.3517890.

Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001)*, pp. 269–274, 2001. doi: 10.1145/502512.502550. URL https://doi.org/10.1145/502512.502550.

Kimon Fountoulakis, Di Wang, and Shenghao Yang. p-norm flow diffusion for local graph clustering. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3222–3232, 2020. URL https://proceedings.mlr.press/v119/fountoulakis20a.html.

Eugene Fratkin, Brian T Naughton, Douglas L Brutlag, and Serafim Batzoglou. MotifCut: regulatory motifs finding with maximum density subgraphs. *Bioinformatics*, 22(14):e150–e157, 2006. doi: 10.1093/bioinformatics/btl243. URL https://doi.org/10.1093/bioinformatics/btl243.

David Gibson, Ravi Kumar, and Andrew Tomkins. Discovering large dense subgraphs in massive graphs. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB 2005)*, pp. 721–732, 2005. URL https://www.vldb.org/conf/2005/papers/p721-gibson.pdf.

Bryan Hooi, Kijung Shin, Hyun Ah Song, Alex Beutel, Neil Shah, and Christos Faloutsos. Graph-based fraud detection in the face of camouflage. *ACM Transactions on Knowledge Discovery from Data*, 11(4):44:1–44:26, 2017. doi: 10.1145/3056563. URL https://doi.org/10.1145/3056563.

Yufan Huang, David F. Gleich, and Nate Veldt. Densest subhypergraph: Negative supermodular functions and strongly localized methods. In *Proceedings of the ACM Web Conference 2024 (WWW 2024)*, pp. 881–892, 2024. doi: 10.1145/3589334.3645624. URL https://doi.org/10.1145/3589334.3645624.

Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi: 10.1006/jcss.2000.1727. URL https://doi.org/10.1006/jcss.2000.1727.

Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi: 10.1006/jcss.2001.1774. URL https://doi.org/10.1006/jcss.2001.1774.

Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, and Shiqiang Yang. Catching synchronized behaviors in large networks: A graph mining approach. *ACM Transactions on Knowledge Discovery from Data*, 10(4):35:1–35:27, 2016. doi: 10.1145/2746403. URL https://doi.org/10.1145/2746403.

Ravi Kannan and V. Vinay. Analyzing the structure of large graphs. Manuscript, July 13, 1999. URL https://www.cs.yale.edu/homes/kannan/Papers/webgraph.pdf.

Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46 (5):604–632, 1999. doi: 10.1145/324133.324140. URL https://doi.org/10.1145/324133.324140.

Angsheng Li and Pan Peng. Detecting and characterizing small dense bipartite-like subgraphs by the bipartiteness ratio measure. In *Algorithms and Computation (ISAAC 2013)*, volume 8283 of *Lecture Notes in Computer Science*, pp. 655–665. Springer, 2013. doi: 10.1007/978-3-642-45030-3_61. URL https://doi.org/10.1007/978-3-642-45030-3_61.

Meng Liu and David F. Gleich. Strongly local p-norm-cut algorithms for semi-supervised learning and local graph clustering. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5023–5035, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3501672ebc68a5524629080e3ef60aef-Abstract.html.

Raymond Liu. Bipartite density: from mixing time to local algorithms for dense subgraphs. Master of mathematics thesis, University of Waterloo, 2026. URL https://hdl.handle.net/10012/24018.

László Lovász and Miklós Simonovits. The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume. In *31st Annual Symposium on Foundations of Computer Science (FOCS 1990)*, pp. 346–354, 1990. doi: 10.1109/FSCS.1990.89553. URL https://doi.org/10.1109/FSCS.1990.89553.

László Lovász and Miklós Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Structures & Algorithms*, 4(4):359–412, 1993. doi: 10.1002/rsa.3240040402. URL https://doi.org/10.1002/rsa.3240040402.

Peter Macgregor and He Sun. Local algorithms for finding densely connected clusters. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7268–7278, 2021. URL https://proceedings.mlr.press/v139/macgregor21a.html.

Pasin Manurangsi. Almost-polynomial ratio ETH-hardness of approximating densest k-subgraph. In *STOC 2017*, pp. 954–961. ACM, 2017. doi: 10.1145/3055399.3055412. URL https://doi.org/10.1145/3055399.3055412. Theorem numbers refer to arXiv:1611.05991v2.

Pasin Manurangsi, Aviad Rubinstein, and Tselil Schramm. The strongish planted clique hypothesis and its consequences. In *ITCS 2021*, volume 185 of *Leibniz International Proceedings in Informatics*, pp. 10:1–10:21. Schloss Dagstuhl, 2021. doi: 10.4230/LIPIcs.ITCS.2021.10. URL https://doi.org/10.4230/LIPIcs.ITCS.2021.10.

Pan Peng. A local algorithm for finding dense bipartite-like subgraphs. In *Computing and Combinatorics (COCOON 2012)*, volume 7434 of *Lecture Notes in Computer Science*, pp. 145–156. Springer, 2012. doi: 10.1007/978-3-642-32241-9_13. URL https://doi.org/10.1007/978-3-642-32241-9_13.

Lu Qin, Rong-Hua Li, Lijun Chang, and Chengqi Zhang. Locally densest subgraph discovery. In *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2015)*, pp. 965–974, 2015. doi: 10.1145/2783258.2783299. URL https://doi.org/10.1145/2783258.2783299.

Barna Saha, Allison Hoch, Samir Khuller, Louiqa Raschid, and Xiao-Ning Zhang. Dense subgraphs with restrictions and applications to gene annotation graphs. In *Annual International Conference on Research in Computational Molecular Biology*, volume 6044 of *Lecture Notes in Computer Science*, pp. 456–472. Springer, 2010. doi: 10.1007/978-3-642-12683-3_30. URL https://doi.org/10.1007/978-3-642-12683-3_30.

Kijung Shin, Bryan Hooi, and Christos Faloutsos. Fast, accurate, and flexible algorithms for dense subtensor mining. *ACM Transactions on Knowledge Discovery from Data*, 12(3):28:1–28:30, 2018. doi: 10.1145/3154414. URL https://doi.org/10.1145/3154414.

Daniel A. Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM Journal on Computing*, 42(1):1–26, 2013. doi: 10.1137/080744888. URL https://doi.org/10.1137/080744888.

Shenghao Yang and Kimon Fountoulakis. Weighted flow diffusion for local graph clustering with node attributes: An algorithm and statistical guarantees. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 39252–39276, 2023. URL https://proceedings.mlr.press/v202/yang23d.html.

A TECHNICAL LEMMAS FOR THE LOCAL ALGORITHM

We prove the statements deferred from Sections 1 and 2.

A.1 COHESIVE SUBSETS AND DEGREE GUESSES

Lemma A.1 (Existence of cohesive subsets)**.** *For all sets $S \subseteq L$ and $T \subseteq R$ with $e(S,T) > 0$, there are cohesive subsets $S^* \subseteq S$ and $T^* \subseteq T$ satisfying $d(S^*,T^*) \geq d(S,T)$.*

Proof. Choose nonempty sets $S^* \subseteq S$ and $T^* \subseteq T$ maximizing $d(S^*,T^*)$. Then $d(S^*,T^*) \geq d(S,T) > 0$. For $u \in S^*$, the bound $e(u,T^*) \geq e(S^*,T^*)/(2|S^*|)$ is trivial if $S^* = \{u\}$. Otherwise, by maximality, $d(S^* \setminus \{u\}, T^*) \leq d(S^*,T^*)$, that is,

$$\frac{e(S^*,T^*) - e(u,T^*)}{\sqrt{(|S^*|-1)|T^*|}} \leq \frac{e(S^*,T^*)}{\sqrt{|S^*||T^*|}}\,.$$

Rearranging and using $\sqrt{1-t} \leq 1 - t/2$, we get

$$1 - \frac{e(u,T^*)}{e(S^*,T^*)} \leq \sqrt{1 - \frac{1}{|S^*|}} \leq 1 - \frac{1}{2|S^*|}\,,$$

so $e(u,T^*) \geq e(S^*,T^*)/(2|S^*|)$. By symmetry, $e(S^*,w) \geq e(S^*,T^*)/(2|T^*|)$ for every $w \in T^*$, so S^*, T^* are cohesive. □

Proof of Lemma 2.3. Let $a_0 \coloneqq e(S,T)/(2|S|)$ and $b_0 \coloneqq e(S,T)/(2|T|)$. Since S,T are cohesive, $e(u,T) \geq a_0$ for every $u \in S$ and $e(S,w) \geq b_0$ for every $w \in T$. The guesses a and b are the largest powers of two not exceeding $\max(a_0,1)$ and $\max(b_0,1)$, respectively, so that

$$a_0 < 2a, \qquad b_0 < 2b\,. \tag{8}$$

Degrees are integers and $a_0, b_0 > 0$, so $e(u,T) \geq \max(a_0,1) \geq a$ for every $u \in S$, and similarly $e(S,w) \geq b$ for every $w \in T$. By (8),

$$\sqrt{ab} > \frac{\sqrt{a_0 b_0}}{2} = \frac{d(S,T)}{4}\,.$$

For any k with $|S|, |T| \leq k$, $a_0, b_0 \leq k/2$ and hence $a, b \leq k$. It remains to show $a|S| \leq 2bk$, as $b|T| \leq 2ak$ follows by symmetry. If $a = 1$, then $a|S| = |S| \leq k \leq 2bk$. If $a > 1$, then $a \leq a_0$ and

$$a|S| \leq a_0|S| = b_0|T| < 2b|T| \leq 2bk\,.$$

□

Corollary A.2 (Arbitrary target sets)**.** *For every $\epsilon > 0$ and all sets $S \subseteq L$, $T \subseteq R$ with $e(S,T) > 0$ and $|S|, |T| \leq k$, there are $S' \subseteq S$ and $T' \subseteq T$ with $e(S',T') \geq e(S,T)/2$ such that, from every starting vertex $v \in S' \dot\cup T'$, the algorithm* $\mathsf{LocalDense}_{2+\epsilon}(G,v,k)$ *satisfies (i) of Theorem 1.2 and its output (X,Y) satisfies $d(X,Y) \geq \frac{\sqrt{3}}{32\sqrt{e}} \frac{\epsilon}{1+\epsilon}\, d(S,T)$.*

Proof. By Lemma C.2, there are $S' \subseteq S$ and $T' \subseteq T$ with $e(S',T') \geq e(S,T)/2$ such that $e(u,T') > a_0$ for every $u \in S'$ and $e(S',w) > b_0$ for every $w \in T'$, where now $a_0 \coloneqq e(S,T)/(4|S|)$ and $b_0 \coloneqq e(S,T)/(4|T|)$. Since $e(S,T) \leq |S||T|$, both a_0 and b_0 are at most $k/4$. Rounding as in the proof of Lemma 2.3, we choose powers of two $a, b \leq k$ with $e(u,T') \geq a$ for every $u \in S'$, $e(S',w) \geq b$ for every $w \in T'$, and $\sqrt{ab} > \sqrt{a_0 b_0}/2 = d(S,T)/8$.

If $a > 1$, then $a|S'| \leq a_0|S| = b_0|T| < 2b|T| \leq 2bk$, and if $a = 1$, then $a|S'| \leq k \leq 2bk$. By symmetry, $b|T'| \leq 2ak$.

In the proof of (7) in Section 2.3, we use only these properties of the target sets and of the guess (a,b), so from every $v \in S' \dot\cup T'$ the output has density above $\frac{\sqrt{3}}{32} C_p^{-p/2} d(S,T)$. The stated bound follows from (17). The bounds in (i) hold for every starting vertex by Theorem B.2. □

A.2 WEAK-ℓ_p PRELIMINARIES AND PRUNING

Lemma A.3 (Prefix sums, support size, and pruning)**.** *For a nonzero vector $x \geq 0$ with $\ell := |\operatorname{supp}(x)|$, $p \geq 2$ and $\delta > 0$:*

(a) The supremum in the definition of $\|x\|_{p,\infty}$ is attained at an integer $m \leq \ell$, that is,

$$\|x\|_{p,\infty} = \max_{1 \leq m \leq \ell} \frac{I_x(m)}{m^{1-1/p}}\,. \tag{9}$$

(b) If every positive coordinate of x is at least δ, then $\ell \leq \|x\|_{p,\infty}^p/\delta^p$.

(c) For $y \geq 0$ and $z := \operatorname{prune}(\delta, y)$, $z \leq y$, every positive coordinate of z is at least δ, and

$$0 \leq \sum_{w \in Q} \big(y(w) - z(w)\big) \leq \delta|Q| \qquad (Q \subseteq V)\,. \tag{10}$$

Proof of Lemma A.3. (a) On $(0, 1]$ the ratio $I_x(s)/s^{1-1/p} = x_{(1)}s^{1/p}$ increases, and for $s > \ell$ the numerator is constant while the denominator increases. On each interval $[i-1, i]$ with $2 \leq i \leq \ell$, I_x is linear (Definition 2.1) and $s^{1-1/p}$ is concave. With c the maximum of the ratio over this interval, $I_x(s) - cs^{1-1/p}$ is convex with maximum zero, and a convex function attains its maximum at an endpoint of the interval, so the ratio equals c at an endpoint. Hence the supremum is attained at an integer $m \leq \ell$.

(b) Every positive coordinate is at least δ, so $\ell\delta \leq I_x(\ell) \leq \|x\|_{p,\infty}\ell^{1-1/p}$ by Definition 2.1, and hence $\ell \leq \|x\|_{p,\infty}^p/\delta^p$.

(c) Each coordinate of $y - z$ lies in $[0, \delta)$. □

Remark. The norm $\|x\|_{p,\infty}$ is the norm $\|x\|_{(p,\infty)}$ of Bennett & Sharpley (1988, Chapter 4) on the weak-ℓ_p sequence space, the Lorentz space $\ell_{p,\infty}$. It lies between the quasinorm $\sup_i i^{1/p}x_{(i)}$ and $p/(p-1)$ times that quasinorm.

A.3 THE INTERPOLATION INEQUALITY

Proposition A.4. *For $p > 2$, a vector $x \geq 0$ with nonempty support U and $\ell := |U|$, and a nonnegative $U \times W$ matrix B with column-sum bound $B_\infty > 0$ (Definition 2.2), if $\gamma > 0$ and a set $Q \subseteq W$ of size $m > 0$ satisfy*

$$\sum_{u \in P_i(x)} \sum_{w \in Q} B_{uw} \leq \gamma\sqrt{im} \qquad (1 \leq i \leq \ell)\,, \tag{11}$$

then

$$\sum_{w \in Q} (xB)(w) \leq C_p\, \gamma^{2/p} B_\infty^{1-2/p} \|x\|_{p,\infty} m^{1-1/p}\,. \tag{12}$$

Proof of Proposition A.4. We write $U = \{u_1, \ldots, u_\ell\}$ in the order of Definition 2.1 and set $c_i := \sum_{w \in Q} B_{u_i,w}$, so that $\sum_{w \in Q}(xB)(w) = \sum_{i=1}^{\ell} x_{(i)}c_i$. By (11) and the column-sum bound,

$$\sum_{j \leq i} c_j \leq g(i) \qquad (1 \leq i \leq \ell), \qquad \text{where} \qquad g(s) := \min\{\gamma\sqrt{ms},\, B_\infty m\} \quad (s \geq 0)\,.$$

Then $g(0) = 0$, and the two terms defining g are equal at $r := B_\infty^2 m/\gamma^2$, so $g'(s) = \gamma\sqrt{m}/(2\sqrt{s})$ for $0 < s < r$ and $g'(s) = 0$ for $s > r$. By summation by parts, with $x_{(\ell+1)} = 0$,

$$\sum_{i=1}^{\ell} x_{(i)}c_i = \sum_{i=1}^{\ell} (x_{(i)} - x_{(i+1)}) \sum_{j \leq i} c_j \leq \sum_{i=1}^{\ell} (x_{(i)} - x_{(i+1)})g(i) = \sum_{i=1}^{\ell} x_{(i)}\big(g(i) - g(i-1)\big)\,.$$

Since I_x is linear with slope $x_{(i)}$ on $[i-1, i]$ and constant beyond ℓ (Definition 2.1), we can write the last sum as an integral. Integrating by parts, with a zero boundary term at 0 because $I_x(s) = x_{(1)}s$

for $0 \le s \le 1$, we obtain

$$\begin{aligned}\sum_{w\in Q}(xB)(w) &\le \int_0^\infty I'_x(s)g'(s)\,ds \\ &= \frac{\gamma\sqrt{m}}{2}\int_0^r \frac{I'_x(s)}{\sqrt{s}}\,ds \\ &= \frac{\gamma\sqrt{m}}{2}\left[\frac{I_x(r)}{\sqrt{r}} + \frac{1}{2}\int_0^r \frac{I_x(s)}{s^{3/2}}\,ds\right].\end{aligned}$$

By Definition 2.1, $I_x(s) \le \|x\|_{p,\infty} s^{1-1/p}$ for every real $s > 0$. With $\tau \coloneqq 1/2 - 1/p > 0$, the bracketed expression is at most

$$\|x\|_{p,\infty}\left(r^\tau + \frac{1}{2}\int_0^r s^{\tau-1}\,ds\right) = \|x\|_{p,\infty} r^\tau\Big(1 + \frac{1}{2\tau}\Big) = 2C_p\,\|x\|_{p,\infty} r^\tau\,.$$

Finally

$$\gamma\sqrt{m}\,r^\tau = \gamma\sqrt{m}\Big(\frac{B_\infty^2 m}{\gamma^2}\Big)^{1/2-1/p} = \gamma^{2/p} B_\infty^{1-2/p} m^{1-1/p}\,,$$

and (12) holds. □

The restriction $p > 2$ is needed only for $\tau > 0$, so that the integral of $s^{\tau-1}$ converges at zero. For $p = 2$, the integrand is $1/s$, whose integral over $[1, r]$ is $\ln r$ for $r \ge 1$, and we must bound the small prefixes separately, as in Section C.3.

The prefix-sum bound. The bound (2) in Section 2.1 is the case of Proposition A.4 in which B is the adjacency matrix A of a subgraph of G with column-sum bound B_∞, $x \ge 0$ is supported on L, $y = xA$, and $Q = P_m(y)$ is a prefix of y. If $d(P_i(x), P_m(y)) \le \sigma$ for every prefix $P_i(x)$, then

$$\sum_{u\in P_i(x)}\sum_{w\in P_m(y)} A_{uw} \le e(P_i(x), P_m(y)) \le \sigma\sqrt{im} \qquad (1 \le i \le \ell)\,,$$

which is (11) with $\gamma = \sigma$, and (12) is (2).

A.4 The one-step search bound

Lemma A.5 (Prefix sums after peeling)**.** *For $p > 2$, $a, b, \sigma > 0$, a vector $x \ge 0$ with nonempty support $U \subseteq L$, the neighborhood graph $H = (U \dot\cup W, F)$ of U with $W \coloneqq N_G(U)$, and a vector y returned by* Peel(H, x, a, b)*, every nonempty $Q \subseteq W$ with $d(P_i(x), Q) \le \sigma$ for all $1 \le i \le |U|$ satisfies*

$$\sum_{w\in Q} y(w) \le \left(\frac{4C_p^p\sigma^2}{b^2}\right)^{1/p} \|x\|_{p,\infty}|Q|^{1-1/p}\,. \tag{13}$$

The same bound holds for $z \coloneqq \mathrm{prune}(\delta, y)$, for every $\delta > 0$. For $U \subseteq R$, the same holds with a and b exchanged, so that a^2 replaces b^2 in (13).

Proof of Lemma A.5. We write F' for the selected edges and B for the $U \times W$ matrix with $B_{uw} \coloneqq 2/b$ for $uw \in F'$ and $B_{uw} \coloneqq 0$ otherwise, so that $y = xB$. Since Peel selects at most $b/2$ edges incident to each $w \in W$,

$$\sum_{u\in U} B_{uw} \le 1 \qquad (w \in W)\,.$$

For every prefix $P_i(x)$,

$$\sum_{u\in P_i(x)}\sum_{w\in Q} B_{uw} \le \frac{2}{b}\,e(P_i(x), Q) \le \frac{2\sigma}{b}\sqrt{i|Q|}\,.$$

We apply Proposition A.4 with $B_\infty = 1$ and $\gamma = 2\sigma/b$ to obtain (13). Since $0 \le z \le y$, the same bound holds for z. □

With σ from (4) and $Q = P_m(y)$, (13) is (5).

Lemma A.6 (Scanning and returned sets)**.** *The following holds for one step of* Search_p *with peeling parameters* $a, b > 0$ *and density threshold* $\sigma > 0$ *from a nonnegative vector* x *with nonempty support* $U \subseteq L$*, where* $W \coloneqq N_G(U)$*.*

(a) If no star is returned, then the adjacency lists read in line 13 contain every edge of the neighborhood graph H *of* U *(line 15), and* $\deg(u) < \sigma^2$ *for every* $u \in U$*.*
(b) If Peel *returns sets* $U_* \subseteq U$ *and* $Y \subseteq W$*, both are nonempty and*

$$|Y| \le \frac{a|U_*|}{2b} + 1, \qquad d(U_*, Y) > \sqrt{ab/8}\,.$$

For $U \subseteq R$*, the same conclusions hold with* a *and* b *exchanged.*

Proof of Lemma A.6. (a) If no star is returned, every adjacency list of a vertex in U is read to its end (line 14). Thus these lists contain every edge incident to U, and $\deg(u) \le \lceil \sigma^2 \rceil - 1 < \sigma^2$ for every $u \in U$.

(b) Search_p calls Peel only if no star is returned, so by (a), $d_H(P, Q) = d(P, Q)$ for all $P \subseteq U$ and $Q \subseteq W$. Both U_* and W_* are nonempty, since an isolated vertex would have been removed. Y is W_* or consists of $\lceil a|U_*|/(2b) \rceil$ vertices of W_* (line 9), so Y is nonempty and

$$|Y| \le \left\lceil \frac{a|U_*|}{2b} \right\rceil \le \frac{a|U_*|}{2b} + 1\,.$$

If $Y = W_*$, then $e_H(U_*, Y) > (a/4)|U_*|$ and $e_H(U_*, Y) > (b/2)|Y|$ by the loop condition (line 2), so

$$d_H(U_*, Y)^2 = \frac{e_H(U_*, Y)}{|U_*|} \cdot \frac{e_H(U_*, Y)}{|Y|} > \frac{a}{4} \cdot \frac{b}{2} = \frac{ab}{8}\,.$$

If $Y \neq W_*$, then $|Y| = \lceil a|U_*|/(2b) \rceil$, and every vertex of Y has more than $b/2$ neighbors in U_*, so

$$d_H(U_*, Y) > \frac{b}{2}\sqrt{\frac{|Y|}{|U_*|}} \ge \frac{b}{2}\sqrt{\frac{a}{2b}} = \sqrt{ab/8}\,.$$

□

B Completing the analysis of the local algorithm

We complete the proof of Theorem 1.2 and prove Corollary B.4.

B.1 Search invariants and termination

Lemma B.1. *For* $p > 2$*,* $k \ge 1$*,* $a, b \in [1, k]$ *and a vertex* $v \in L$ *of positive degree,* $\mathsf{Search}_p(G, v, k, a, b)$ *with the parameters in (4) satisfies the following.*

(a) Every vector x_t *satisfies* $\|x_t\|_{p,\infty} \le K_t^{1/p}$*.*
(b) Every positive coordinate of x_t *is at least* δ*, and* $|\operatorname{supp}(x_t)| \le K_t/\delta^p \le (4k)^p(3/4)^t$*.*
(c) The search terminates and never returns $\perp$ *from the sweep.*
(d) In each step, the search reads at most $\lceil \sigma^2 \rceil \le ab$ *entries of each adjacency list. Whenever the search returns sets* X, Y*, they satisfy* $d(X, Y) \ge \sigma$ *and* $|X|, |Y| \le (4k)^p$*.*

For $v \in R$*, the same holds for the search with* L *and* R *exchanged.*

Proof of Lemma B.1. (a) $\|x_0\|_{p,\infty} = 1 = K_0^{1/p}$, and every later x_t satisfies the weak-norm test $\|x_t\|_{p,\infty} \le K_t^{1/p}$ (line 20).

(b) Positive coordinates are at least δ at $t = 0$, since $x_0 = \mathbb{1}_v$ and $\delta = \max\{1, a/b\}/(4k) \le 1/4$, and at later steps by pruning (Lemma A.3(c)). By (a) and Lemma A.3(b), $|\operatorname{supp}(x_t)| \le \|x_t\|_{p,\infty}^p/\delta^p \le K_t/\delta^p$. Since $\delta \ge 1/(4k)$ and $K_t \le 4k\delta(3/4)^t$ by (4),

$$\frac{K_t}{\delta^p} \le 4k\delta^{1-p}\Big(\frac{3}{4}\Big)^t \le (4k)^p\Big(\frac{3}{4}\Big)^t\,. \tag{14}$$

(c) Every vector x_t is nonzero (line 19), so, by (b), $1 \le |\operatorname{supp}(x_t)| \le (4k)^p(3/4)^t$. The upper bound tends to zero, so the search computes only finitely many vectors x_t and terminates.

For the second claim of (c), we show that in the sweep at step t we find i with $d(P_i(x_t), P_m(z)) > \sigma$, where $z = \operatorname{prune}(\delta, y_{t+1})$ and $P_m(z)$ is the prefix of line 23. Otherwise, for even t, by Lemma A.5 with $Q = P_m(z)$ and by (a),

$$I_z(m) \le \left(\frac{4C_p^p\sigma^2}{b^2}\right)^{1/p} K_t^{1/p} m^{1-1/p}\,.$$

By (4), $4C_p^p\sigma^2 = 3ab/4$, so the factor is $(3a/(4b))^{1/p} = (K_{t+1}/K_t)^{1/p}$. For odd t, (13) holds with a^2 in place of b^2, and the factor is $(3b/(4a))^{1/p} = (K_{t+1}/K_t)^{1/p}$. In both cases, $I_z(m) \le K_{t+1}^{1/p} m^{1-1/p}$, contradicting the choice of m. The graph H is the neighborhood graph of $\operatorname{supp}(x_t)$ (line 15), so $d_H(P_i(x_t), P_m(z)) = d(P_i(x_t), P_m(z)) > \sigma$, and the sets $P_i(x_t), P_m(z)$ are returned by the sweep.

(d) In each step, the search reads at most $\lceil\sigma^2\rceil$ entries of each adjacency list (line 13). By Bernoulli's inequality, $C_p^p = (1 + \frac{1}{p-2})^p \ge 1 + \frac{p}{p-2} > 2$, so $\sigma^2 = 3ab/(16C_p^p) < ab/8$, and since $ab \ge 1$, it follows that $\lceil\sigma^2\rceil \le \lceil ab/8\rceil \le ab \le k^2 \le (4k)^p$. A returned star therefore has at most $(4k)^p$ leaves, and its density is $\sqrt{\lceil\sigma^2\rceil} \ge \sigma$. The sets returned by the sweep have density above σ (line 24), and the sets returned by peeling have density above $\sqrt{ab/8} > \sigma$ (Lemma A.6(b)).

For sets $P_i(x_t), P_m(z)$ returned by the sweep at step t, the prefix $P_i(x_t)$ lies in $\operatorname{supp}(x_t)$, so $i \le (4k)^p$ by (b). By Lemma A.3(a), $m \le |\operatorname{supp}(z)|$, so $z_{(m)} \ge \delta$ by pruning, and since m is the smallest index with $I_z(m) > K_{t+1}^{1/p} m^{1-1/p}$,

$$(m-1)\delta \le I_z(m-1) \le K_{t+1}^{1/p}(m-1)^{1-1/p}\,,$$

so, by (14) and $(4k)^p \ge 16$,

$$m \le \frac{K_{t+1}}{\delta^p} + 1 \le \frac{3}{4}(4k)^p + 1 \le (4k)^p\,.$$

For sets U_*, Y returned by peeling at step t, the set U_* lies in $\operatorname{supp}(x_t)$, so $|U_*| \le K_t/\delta^p$ by (b). By Lemma A.6(b), $|Y| \le a|U_*|/(2b) + 1$ for even t, and $|Y| \le b|U_*|/(2a) + 1$ for odd t. In both cases, the coefficient of $|U_*|$ is $2K_{t+1}/(3K_t)$ by (4). Hence, by (14),

$$|Y| \le \frac{2K_{t+1}}{3K_t}|U_*| + 1 \le \frac{2K_{t+1}}{3\delta^p} + 1 \le \frac{(4k)^p}{2} + 1 \le (4k)^p\,.$$

□

B.2 Query complexity, running time, and locality

Graph access. The algorithm accesses G through neighbor queries. Each adjacency list has a fixed order. When the algorithm makes a query (u, i) for an integer $i \ge 1$, it receives the ith neighbor of u, or $\perp$ if $i > \deg(u)$, where $\deg(u)$ is the degree of u. To read the list to its end, the algorithm makes $\deg(u) + 1$ queries. We assume that the side of a vertex can be read from its identifier. We count neighbor queries and elementary operations at unit cost, including reads, copies, and comparisons of vertex identifiers. For $\Theta(\log|V|)$-bit identifiers, reading an identifier takes $\Theta(\log|V|)$ bit operations.

Theorem B.2 (Local computation)**.** *For $p > 2$, $k \ge 1$ and every starting vertex v, the algorithm* $\mathsf{LocalDense}_p(G, v, k)$ *terminates. If v is isolated, it returns $\perp$ after one query. Otherwise, the output has at most $2(4k)^p$ vertices and density at least one, the number of neighbor queries is at most*

$$1 + 4(4k)^p \min\left\{(\Delta+1)(1+\lfloor\log_2 k\rfloor)^2,\ 4k^2\right\}\,, \tag{15}$$

and the running time is

$$O_p\big(\min\left\{\Delta k^p \log_2^3(2k),\ k^{p+2}\log_2(2k)\right\}\big)\,. \tag{16}$$

Proof of Theorem B.2. If v is isolated, the first query is answered with $\perp$, and $\mathsf{LocalDense}_p$ returns $\perp$ as well (line 2). Otherwise, every search terminates by Lemma B.1(c). By Lemma B.1(d), the sets returned by the searches have at most $(4k)^p$ vertices per side, so the output has at most $2(4k)^p$ vertices. Its density is at least one because the initial edge is a candidate.

By Lemma B.1(b), for each guess (a, b), the algorithm scans at most $\sum_t |\operatorname{supp}(x_t)| \le 4(4k)^p$ adjacency lists in total, with at most $\min\{\Delta+1, \lceil\sigma^2\rceil\} \le \min\{\Delta+1, ab\}$ queries per list by Lemma B.1(d). Including the initial query, the total is at most

$$1 + 4(4k)^p \sum_{a,b} \min\{\Delta+1, ab\}\,,$$

where there are $(1+\lfloor\log_2 k\rfloor)^2$ guesses and

$$\sum_{a,b} ab = \left(\sum_{j=0}^{\lfloor\log_2 k\rfloor} 2^j\right)^2 < 4k^2\,.$$

Thus the number of queries is at most (15).

For the running time, we consider a step in which no star is returned, with the neighborhood graph $H = (U \dot\cup W, F)$ of $U := \operatorname{supp}(x_t)$. Every vertex of U has a neighbor, since v has positive degree and every vertex of $\operatorname{supp}(x_{t+1})$ is joined to $\operatorname{supp}(x_t)$ by a selected edge (line 8), so $|U| + |W| \le 2|F|$. By Lemma B.1(b), $|U| \le (4k)^p$, and by Lemma A.6(a) and (4), $\deg(u) < \sigma^2 < ab \le k^2$ for every $u \in U$, so

$$|F| = \sum_{u\in U} \deg(u) < k^2(4k)^p\,.$$

The time spent in this step sorting the vectors, running Peel and performing any sweep is $O(|F|\log_2|F|) = O_p(|F|\log_2(2k))$, and the algorithm makes at least $|F|$ queries to build H. When a star is returned, the work in that step is linear in the number of queries. Summing over all steps and guesses, we obtain a time bound of $O_p(\log_2(2k))$ per query and hence (16). □

Lemma B.3 (Distance from the starting vertex)**.** *For sets $X \subseteq L$ and $Y \subseteq R$ returned by $\mathsf{LocalDense}_p(G, v, k)$, every vertex of $X \dot\cup Y$ lies within distance $1 + p\log_2(4k)/\log_2(4/3)$ of v in G. If $v \in L$ then $d(X \cup \{v\}, Y) \ge d(X, Y)/\sqrt{2}$, and if $v \in R$ then $d(X, Y \cup \{v\}) \ge d(X, Y)/\sqrt{2}$.*

Proof of Lemma B.3. In every search, $\operatorname{supp}(x_0) = \{v\}$, and every vertex of $\operatorname{supp}(x_{t+1})$ has a neighbor in $\operatorname{supp}(x_t)$, since y_{t+1} is indexed by $N_G(\operatorname{supp}(x_t))$ and $x_{t+1} \le y_{t+1}$. By induction, $\operatorname{supp}(x_t)$ lies within distance t of v. At step t, the vertices of a returned star, of the sets returned by peeling and of the prefixes returned by the sweep lie in $\operatorname{supp}(x_t) \dot\cup N_G(\operatorname{supp}(x_t))$ and hence within distance $t+1$ of v. The initial edge is within distance one of v. Every vector x_t is nonzero, so $1 \le |\operatorname{supp}(x_t)| \le (4k)^p(3/4)^t$ by Lemma B.1(b), and thus $t \le p\log_2(4k)/\log_2(4/3)$.

For the last claim, if $v \in L$, then $e(X \cup \{v\}, Y) \ge e(X, Y)$ and $|X \cup \{v\}| \le 2|X|$, so $d(X \cup \{v\}, Y) \ge d(X, Y)/\sqrt{2}$. The case $v \in R$ is symmetric. □

B.3 Choosing the exponent

For $\epsilon > 0$ and $p := 2 + \epsilon$, the constant C_p equals $1 + 1/\epsilon$. By $\ln(1+1/\epsilon) \le 1/\epsilon$,

$$C_p^{-p/2} = \left(\frac{\epsilon}{1+\epsilon}\right)^{1+\epsilon/2} = \frac{\epsilon}{1+\epsilon}\left(1+\frac{1}{\epsilon}\right)^{-\epsilon/2} \ge \frac{\epsilon}{\sqrt{\mathrm{e}}(1+\epsilon)}\,. \tag{17}$$

By (7) and (17), $d(X, Y) > \frac{\sqrt{3}}{16\sqrt{\mathrm{e}}}\frac{\epsilon}{1+\epsilon}\,d(S, T)$, which is (ii) of Theorem 1.2. By Theorem B.2, the output has $O_\epsilon(k^{2+\epsilon})$ vertices, and the algorithm makes $O_\epsilon(k^{4+\epsilon})$ neighbor queries and runs in time $\widetilde{O}_\epsilon(k^{4+\epsilon})$, as claimed in Theorem 1.2.

Corollary B.4 (Quadratic output size)**.** *For cohesive sets $S \subseteq L$ and $T \subseteq R$, $k \ge 1$ with $|S|, |T| \le k$, a vertex $v \in S \dot\cup T$, $\epsilon := 1/\lceil\log_2(2k)\rceil$ and $p := 2 + \epsilon$, the algorithm $\mathsf{LocalDense}_p(G, v, k)$ returns nonempty sets $X \subseteq L$ and $Y \subseteq R$ with $|X| + |Y| \le 128k^2$ and*

$$d(X, Y) > \frac{\sqrt{6}}{128}\,\frac{d(S, T)}{\log_2(2k)}\,.$$

For every starting vertex, the number of neighbor queries is at most

$$1 + 256k^2 \min\left\{(\Delta+1)(1+\lfloor \log_2 k \rfloor)^2,\ 4k^2\right\}.$$

Proof of Corollary B.4. The choice $\epsilon = 1/\lceil \log_2(2k) \rceil$ satisfies $1/(2\log_2(2k)) \le \epsilon \le 1$ and $2^{1/\epsilon} \ge 1 + 1/\epsilon$. By the identity in (17),

$$C_p^{-p/2} \ge \frac{\epsilon}{2\sqrt{2}} \ge \frac{1}{4\sqrt{2}\log_2(2k)}.$$

By this inequality and (7), $d(X,Y) > \frac{\sqrt{6}}{128}\, d(S,T)/\log_2(2k)$. Since $\log_2(2k) \ge 1$, $\log_2(4k) = \log_2(2k) + 1 \le 2\lceil \log_2(2k) \rceil$, and hence

$$(4k)^{\epsilon} = 2^{\log_2(4k)/\lceil \log_2(2k) \rceil} \le 2^2 = 4.$$

Thus $(4k)^p \le 64k^2$, and by Theorem B.2 with this bound, there are at most $2(4k)^p \le 128k^2$ output vertices, and (15) gives the stated query bound. □

C THE $p = 2$ BARRIER AND COMPARISON WITH ANDERSEN

Andersen's guarantee has two parts: growth of the Euclidean norm inside a dense bipartite subgraph, and extraction of a dense bipartite subgraph from that growth (Section C.1). After recalling his iteration and the two lemmas, we describe the neighbor-query implementation of his algorithm in Section C.2. With the weak-ℓ_2 bound $I_x(i) \le \sqrt{i}$ in place of the bound on $\|x\|_{p,\infty}$, we incur a logarithmic factor in the prefix-sum argument of Section 2.1 (Section C.3), and there is a weighted matrix for which the logarithmic factor in the extraction is necessary (Section C.4).

C.1 ANDERSEN'S GROWTH AND EXTRACTION LEMMAS

Andersen's iteration. Andersen's algorithm runs the pruned growth process (Andersen, 2010, Definition 5) for $t_0 := \lceil \log_2(2\sqrt{k}) \rceil$ steps. Starting from $x_0 := \mathbb{1}_v$, with the pruning thresholds $\delta_{t+1} > 0$ of that definition and the adjacency matrix A of G, it computes

$$y_{t+1} := \frac{x_t A}{\|x_t A\|_2}, \qquad x_{t+1} := \mathrm{prune}(\delta_{t+1}, y_{t+1}).$$

Each y_{t+1} is a unit vector, so $\|x_t\|_2 \le 1$.

Definition C.1 (Dyadic level sets [Andersen, 2010, Definition 6])**.** *For a nonnegative vector x, its* dyadic level sets *are $D_j(x) := \{u : 2^{-j-1} < x(u) \le 2^{-j}\}$ for $j \in \mathbb{Z}$.*

At each step $0 \le t < t_0$, Andersen's algorithm tests the nonempty sets

$$X := D_i\left(\frac{x_t}{\|x_t\|_2}\right), \qquad Y := D_j\left(\frac{x_t A}{\|x_t A\|_2}\right) \quad \text{with } |i-j| \le 4 + \log_2 \Delta.$$

It returns the tested sets X, Y of maximum density.

Lemma C.2 (Growth inside a dense bipartite subgraph [Andersen, 2010, Lemma 2 and proof of Theorem 1])**.** *For $S \subseteq L$ and $T \subseteq R$ with $e(S,T) > 0$ and k with $|S|, |T| \le k$, there are $S' \subseteq S$ and $T' \subseteq T$ with $e(S',T') \ge e(S,T)/2$ such that every vertex of S' has more than $e(S,T)/(4|S|)$ neighbors in T' and every vertex of T' has more than $e(S,T)/(4|T|)$ neighbors in S'.*

From every starting vertex $v \in S' \dot\cup T'$, the vectors of Andersen's algorithm with parameter k satisfy $\|x_t A\|_2 \ge (d(S,T)/16)\|x_t\|_2$ for some $t < t_0$. If $d(S,T) < 16$, then also $\|x_0 A\|_2 \ge \|x_0\|_2$.

For $d(S,T) \ge 16$, Andersen's proof of Theorem 1 applies with $\lambda = d(S,T)/16$. For the last claim, every $v \in S' \dot\cup T'$ has a neighbor, so $\|x_0 A\|_2 \ge 1 = \|x_0\|_2$.

Lemma C.3 (From growth to dense level sets [Andersen, 2010, Lemma 1, weakened form])**.** *For a nonzero vector $x \ge 0$ supported on one side and $\lambda \ge 1$ with $\|xA\|_2 \ge \lambda \|x\|_2$, there are nonempty dyadic level sets X of $x/\|x\|_2$ and Y of $xA/\|xA\|_2$ with $d(X,Y)$ at least*

$$\frac{\lambda}{72\,(1+\log_2(\Delta/\lambda))}. \tag{18}$$

In Andersen's Lemma 1, the density is at least $\lambda/(72+16\log_2(\Delta/\lambda))$ for some level-set indices i, j satisfying $|i-j| \le 4+\log_2\Delta$. Since $\lambda \le \Delta$, (18) is weaker, and we state it without the restriction on the level-set indices.

C.2 Neighbor-query implementation of Andersen's algorithm

As in Andersen's proof of Theorem 2 (Andersen, 2010), the iteration in Section C.1 has $|\operatorname{supp}(x_t)| = O(k^2 4^{-t})$. At step t, we read the adjacency list of each vertex $u \in \operatorname{supp}(x_t)$ to its end, using $\deg(u)+1 \le \Delta+1$ neighbor queries. Using the cached lists, we compute x_tA and $e(X,Y)$ for all level sets X of $x_t/\|x_t\|_2$ and Y of $x_tA/\|x_tA\|_2$ without further queries. The total number of neighbor queries is therefore at most

$$(\Delta+1)\sum_{t=0}^{t_0-1}|\operatorname{supp}(x_t)| = O(\Delta k^2)\,,$$

since $\Delta \ge 1$ for a target of positive density. If Δ is unknown, we omit the restriction $|i-j| \le 4+\log_2\Delta$ and consider all level sets X, Y with $e(X,Y) > 0$. No additional queries are needed, and the returned density is no smaller.

C.3 Logarithmic accumulation for weak-ℓ_2

Andersen's vectors are normalized in the Euclidean norm. For $x \ge 0$ supported on L with $\|x\|_2 \le 1$, by Cauchy–Schwarz, $I_x(i) \le \sqrt{i}$ for every i, that is, $\|x\|_{2,\infty} \le 1$ in the notation of Definition 2.1. We bound the prefix sums of $y \coloneqq xA$ as in Section 2.1, for a prefix $P_m(y)$ with $d(P_i(x), P_m(y)) \le \sigma$ for every prefix $P_i(x)$ of x.

We also assume that every vertex in $\operatorname{supp}(x)$ has degree below σ^2, as in Search_p after the adjacency lists of $\operatorname{supp}(x)$ have been read and no star has been returned (Lemma A.6(a)). By the density assumption, and since the vertices of $P_i(x)$ have degree below σ^2 and those of $P_m(y)$ degree at most Δ, $e(P_i(x), P_m(y)) \le g(i)$ for $g(i) \coloneqq \min\{\sigma^2 i,\, \sigma\sqrt{im},\, \Delta m\}$. We write $r_0 \coloneqq m/\sigma^2$ and $r_1 \coloneqq \Delta^2 m/\sigma^2$. Since $y \ne 0$, $\Delta \ge 1$ and $r_1/r_0 = \Delta^2 \ge 1$. The degree bound $\sigma^2 i$ is the smallest of the three for $i \le r_0$, the density bound $\sigma\sqrt{im}$ for $r_0 \le i \le r_1$, and the maximum-degree bound Δm for $i \ge r_1$. By summation by parts, as in the proof of Proposition A.4,

$$I_y(m) \le \sum_{i=1}^{\ell} x_{(i)}\big(g(i)-g(i-1)\big)\,.$$

We split this sum at r_0 and r_1 and bound the three parts separately. The terms with $i \le r_0$ have $g(i)-g(i-1) = \sigma^2$, so their total contribution is at most $\sigma^2 I_x(r_0) \le \sigma^2\sqrt{r_0} = \sigma\sqrt{m}$. The terms with $i \ge r_1+1$ are zero, since $g(i) = g(i-1) = \Delta m$. If $i-1 < r_0 < i$ or $i-1 < r_1 < i$, the corresponding term is at most $x_{(i)}g(i) \le \sigma\sqrt{m}$. There are at most two such indices. For the remaining indices, $r_0 \le i-1 < i \le r_1$, so $g(i)-g(i-1) = \sigma\sqrt{m}(\sqrt{i}-\sqrt{i-1}) \le \sigma\sqrt{m}/\sqrt{i}$. Since $x_{(i)} \le I_x(i)/i \le 1/\sqrt{i}$, the terms in a dyadic range $2^j \le i < 2^{j+1}$ therefore add up to

$$\sum_{\substack{2^j \le i < 2^{j+1}\\ r_0 \le i-1 < i \le r_1}} x_{(i)}\big(g(i)-g(i-1)\big) = O(\sigma\sqrt{m}) \sum_{2^j \le i < 2^{j+1}} \frac{1}{i} = O(\sigma\sqrt{m})\,.$$

Since $r_1/r_0 = \Delta^2$, at most $2\log_2\Delta + O(1)$ dyadic ranges contain these remaining indices. Altogether,

$$I_y(m) = O\big(\sigma\sqrt{m}\,(1+\log_2\Delta)\big)\,. \tag{19}$$

Using only $I_x(i) \le \sqrt{i}$, we cannot improve the logarithmic dependence in the preceding upper bound on $\sum_i x_{(i)}(g(i)-g(i-1))$. The vector x with $x_{(i)} \coloneqq \sqrt{i}-\sqrt{i-1}$ for $i \le \ell$, where $\ell > r_1$ and $m \ge \sigma^2$, satisfies $I_x(i) = \sqrt{i}$, and the contribution to this sum from each dyadic range between r_0 and r_1 is $\Theta(\sigma\sqrt{m})$. This vector has $\|x\|_2 > 1$. For $\|x\|_2 \le 1$, however, the sum is $O(\sigma\sqrt{m}\sqrt{1+\log_2\Delta})$: by Cauchy–Schwarz, it is at most $\|x\|_2\big(\sum_i (g(i)-g(i-1))^2\big)^{1/2}$, and

$(g(i)-g(i-1))^2 = \left(\int_{i-1}^{i} g'(s)\,ds\right)^2 \le \int_{i-1}^{i} g'(s)^2\,ds$, so that

$$\sum_i \big(g(i)-g(i-1)\big)^2 \le \int_0^\infty g'(s)^2\,ds = \sigma^4 r_0 + \frac{\sigma^2 m}{4}\ln\frac{r_1}{r_0} = \sigma^2 m\Big(1+\frac{1}{2}\ln\Delta\Big)\,.$$

The logarithm in Andersen's Lemma 1 also comes from a sum over logarithmically many scales: in its proof, he sums over all level-set indices i, j with $|i-j| \le 4 + \log_2(\Delta/\lambda)$.

For $p > 2$, in contrast, $x_{(i)} \le \|x\|_{p,\infty} i^{-1/p}$. The sum over the dyadic range $2^j \le i < 2^{j+1}$ is then $O(\sigma\sqrt{m}\,\|x\|_{p,\infty} 2^{j(1/2-1/p)})$. The sum over all ranges is therefore $O_p(1)$ times the largest of these bounds, and there is no logarithmic factor in Proposition A.4.

C.4 A TIGHT EXAMPLE FOR NORM-GROWTH EXTRACTION

For nonnegative matrices, the logarithmic factor in (18) is necessary (see Bilu & Linial, 2006). For a nonnegative matrix B and nonempty sets S, T of row and column indices, we write

$$d_B(S,T) := \frac{\sum_{u\in S}\sum_{w\in T} B_{uw}}{\sqrt{|S||T|}}\,.$$

In Construction C.4, we define a symmetric matrix $\bar{A}$ and a vector ξ_L with $\|\xi_L\bar{A}\|_2 = \lambda\|\xi_L\|_2$ such that $d_{\bar{A}}(S,T) = O(\lambda/\log_2\Delta)$ for all nonempty $S \subseteq L$ and $T \subseteq R$, where Δ is the largest row sum.

Construction C.4. For an integer $\nu \ge 1$ and $\zeta := 2^\nu$, the sets L and R are partitioned into blocks L_i, R_i of size 4^i for $i = 0, \dots, \nu-1$. The symmetric matrix $\bar{A}$, indexed by $L \dot\cup R$, has entry $\zeta 2^{-i-j}$ between L_i and R_j and zero entries on $L\times L$ and $R\times R$. The vector ξ_L equals 2^{-i} on L_i and zero on R, the vector ξ_R is defined symmetrically, and $\lambda := \|\xi_L\bar{A}\|_2/\|\xi_L\|_2$.

Proposition C.5. *The matrix of Construction C.4 has spectral radius $\nu\zeta$ and largest row sum $\Delta = \zeta(\zeta-1)$. Moreover,*

$$\lambda = \nu\zeta, \qquad \zeta \le \max_{\substack{\varnothing\ne S\subseteq L\\ \varnothing\ne T\subseteq R}} d_{\bar{A}}(S,T) \le 9\zeta\,.$$

The nonempty dyadic level sets of $\xi_L/\|\xi_L\|_2$ and $\xi_L\bar{A}/\|\xi_L\bar{A}\|_2$ are $L_0,\dots,L_{\nu-1}$ and $R_0,\dots,R_{\nu-1}$, respectively. Every block density $d_{\bar{A}}(L_i,R_j)$ equals ζ. The ratio of λ to the maximum of $d_{\bar{A}}(S,T)$ over nonempty $S\subseteq L$ and $T\subseteq R$ is $\Theta(\nu) = \Theta(\log_2\Delta)$. The right-hand side of (18), evaluated at $\lambda = \nu\zeta$, is $\Theta(\zeta)$.

Proof. For $u \in R_j$ with $0 \le j < \nu$, the contribution from each left block L_i to $(\xi_L\bar{A})(u)$ is $4^i\cdot 2^{-i}\cdot\zeta 2^{-i-j} = \zeta 2^{-j}$, so

$$(\xi_L\bar{A})(u) = \nu\zeta 2^{-j} = \nu\zeta\,\xi_R(u)\,.$$

By symmetry, $\xi_R\bar{A} = \nu\zeta\,\xi_L$, so $\xi_L+\xi_R$ is a positive eigenvector with eigenvalue $\nu\zeta$, which is therefore the spectral radius. Since $\|\xi_L\|_2 = \|\xi_R\|_2 = \sqrt{\nu}$, $\lambda = \nu\zeta$. By direct summation, we obtain the stated largest row sum and block densities. Consecutive block values differ by a factor of two, so the nonempty dyadic level sets are exactly the blocks.

For nonempty $S \subseteq L$ and $T \subseteq R$, we write $s_i := |S\cap L_i|$ and take the integer $q \ge 0$ with $\sqrt{|S|} \le 2^q < 2\sqrt{|S|}$. Since $s_i \le 4^i$,

$$\sum_{i=0}^{\nu-1} s_i 2^{-i} \le \sum_{i=0}^{q-1} 2^i + 2^{-q}|S| \le 2^q + |S|2^{-q} \le 3\sqrt{|S|}\,.$$

We apply the same bound to $t_j := |T\cap R_j|$ and obtain

$$\sum_{u\in S}\sum_{w\in T}\bar{A}_{uw} = \zeta\Big(\sum_{i=0}^{\nu-1} s_i 2^{-i}\Big)\Big(\sum_{j=0}^{\nu-1} t_j 2^{-j}\Big) \le 9\zeta\sqrt{|S||T|}\,.$$

Finally, since $\Delta = \zeta(\zeta-1)$ and $\lambda = \nu\zeta$, both $\log_2\Delta$ and $1+\log_2(\Delta/\lambda)$ are $\Theta(\nu)$, so the right-hand side of (18) is $\Theta(\zeta)$. □

D LOWER BOUNDS

Since sets X, Y with $d(X, Y) = \Omega(k)$ have $\Omega(k^2)$ edges between them, an output of $o(k^2)$ vertices per side must have many more edges than vertices, as the target does, and the algorithm has to find these edges. The query lower bound is proved by hiding a complete bipartite graph among unread adjacency-list entries (Section D.1), and the computational lower bounds by reducing Densest k-Subgraph to the local problem (Section D.2).

Definition D.1. *A deterministic algorithm* A *receives a starting vertex v, an integer $k \geq 1$, and neighbor-query access to a simple bipartite graph G, as specified in the graph-access model of Section B.2. The algorithm has* output bound ϕ *if, for every input, it returns either $\perp$ or nonempty sets $X \subseteq L$, $Y \subseteq R$ with $|X|, |Y| \leq \phi(k)$. Unless a running-time bound is stated, computation between queries is unrestricted.*

D.1 THE QUERY LOWER BOUND

Theorem D.2 (Queries sublinear in Δ)**.** *For a constant $0 < \rho \leq 1$ and $\phi(k) = o(k^2)$, if an algorithm* A *as in Definition D.1 has output bound ϕ and makes at most $q(k, \Delta)$ queries on every input, where $q(k, \Delta) = o(\Delta)$ as $\Delta \to \infty$ for every fixed k, then, for every sufficiently large k, there is a simple bipartite graph G containing a complete bipartite subgraph with sides $S \subseteq L$ and $T \subseteq R$ of k vertices each, such that from every $v \in S \,\dot\cup\, T$ the output of* A *is $\perp$ or has density below ρk.*

Proof of Theorem D.2. We run A from the centers of disjoint stars, choose target centers from which A returns at most one target vertex per side, and connect the target centers through unread adjacency-list entries without changing any query answer.

Since $\phi(k) = o(k^2)$, the inequality $\sqrt{2\phi(k)} + 1 < \rho k$ holds for all sufficiently large k, and we fix such a k. We set $n := k(\lfloor 8\phi(k) \rfloor + 1)$ and choose Δ with $\Delta - 2nq(k, \Delta) \geq k$, which is possible because $q(k, \Delta) = o(\Delta)$ as $\Delta \to \infty$. The graph G_0 consists of $2n$ disjoint stars with Δ leaves each, with left centers $u_1, \ldots, u_n$ and right centers $w_1, \ldots, w_n$, and we write $\Pi_i := \{u_i, w_i\}$. We run $\mathsf{A}(G_0, v, k)$ from every center v and write C_v for the set of centers in its output ($C_v := \varnothing$ if it returns $\perp$). Then $|C_v| \leq 2\phi(k)$, and A makes at most $2nq(k, \Delta)$ queries across the $2n$ runs.

We select k indices i such that no output from a center of Π_i contains a center of Π_j for another selected index j. The graph $\mathcal{H}$ has vertex set $\{1, \ldots, n\}$ and joins distinct i, j if

$$(C_{u_i} \cup C_{w_i}) \cap \Pi_j \neq \varnothing \qquad \text{or} \qquad (C_{u_j} \cup C_{w_j}) \cap \Pi_i \neq \varnothing \,.$$

For each i, the two outputs contain at most $4\phi(k)$ centers in total, hence centers of at most $4\phi(k)$ other sets Π_j. Thus for every set I of indices,

$$|E(\mathcal{H}[I])| \leq 4\phi(k)|I| \,.$$

Every nonempty induced subgraph of $\mathcal{H}$ therefore has a vertex of degree at most $\lfloor 8\phi(k) \rfloor$. We repeatedly select such a vertex and delete it and its neighbors from the remaining graph. Since we delete at most $\lfloor 8\phi(k) \rfloor + 1$ vertices per selection, we obtain at least k indices, no two of them adjacent in $\mathcal{H}$, of which we keep exactly k. The sets S and T consist of the corresponding left and right centers. For every selected index i and each $v \in \Pi_i$,

$$C_v \cap (S \,\dot\cup\, T) \subseteq \Pi_i \,.$$

A star edge is *untouched* if its position in the center's list was never queried and its leaf was never queried. With each query, at most one edge is touched, so every center has at least $\Delta - 2nq(k, \Delta) \geq k$ untouched edges. For each $u \in S$ and $w \in T$, we choose untouched edges uw' and $u'w$ that were not chosen before. We replace w' by w in the list of u, replace u' by u in the list of w, and isolate the two leaves. The resulting graph G is simple and bipartite, all centers still have degree Δ, and S, T span a complete bipartite subgraph and are hence cohesive, with $d(S, T) = k$.

From every center, A receives the same answers on G and G_0, since only unqueried positions in the centers' lists and the lists of unqueried leaves have changed. By induction on the number of queries, it therefore follows the same computation and returns the same output.

For sets $X \subseteq L$ and $Y \subseteq R$ returned from a selected center $v \in S \dot\cup T$, we write e_0 for the number of edges of G_0 between X and Y. Since G_0 is a forest, $e_0 \le |X| + |Y| \le 2\phi(k)$ and $e_0 \le |X||Y|$. Hence

$$\frac{e_0}{\sqrt{|X||Y|}} \le \sqrt{e_0} \le \sqrt{2\phi(k)}\,.$$

If $v \in \Pi_i$, then the output from v contains no target vertices other than u_i and w_i, so at most one edge between X and Y is not in G_0, and

$$d_G(X,Y) \le \sqrt{2\phi(k)} + \frac{1}{\sqrt{|X||Y|}} \le \sqrt{2\phi(k)} + 1 < \rho k = \rho\, d_G(S,T)\,.$$

□

Corollary D.3 (Approximation and output size with restricted queries)**.** *For $\alpha(k) \ge 1$ with $\alpha(k) = o(k)$ and $\phi(k) = o(k^2/\alpha(k)^2)$, if an algorithm as in Definition D.1 has output bound ϕ and makes at most $q(k,\Delta)$ queries, where $q(k,\Delta) = o(\Delta)$ as $\Delta \to \infty$ for every fixed k, then, for every sufficiently large k, there is a simple bipartite graph containing a complete bipartite subgraph with sides S, T of k vertices each, such that, from every $v \in S \dot\cup T$, the algorithm returns $\perp$ or sets of density below $d(S,T)/\alpha(k)$.*

Proof. We apply the construction in the proof of Theorem D.2 with $\rho = 1/\alpha(k)$, since its condition $\sqrt{2\phi(k)} + 1 < \rho k$ holds for all sufficiently large k:

$$\frac{\alpha(k)}{k}\big(\sqrt{2\phi(k)} + 1\big) = \sqrt{\frac{2\alpha(k)^2\phi(k)}{k^2}} + \frac{\alpha(k)}{k} < 1\,,$$

as both terms tend to zero. After fixing such a k and setting n as there, we choose Δ with $\Delta - 2nq(k,\Delta) \ge k$, and from every target vertex the algorithm returns $\perp$ or sets X, Y with $d(X,Y) \le \sqrt{2\phi(k)} + 1 < k/\alpha(k) = d(S,T)/\alpha(k)$. □

D.2 Computational lower bounds

For both lower bounds under SPCH and ETH, we use the same reduction. In Section D.2.1, we construct a dense subgraph on k vertices of a graph J from the outputs of a local algorithm on its bipartite double cover, and in Sections D.2.2 and D.2.3, we prove the two lower bounds. Polynomial time means time polynomial in the number of vertices of the input graph, uniformly over all k up to that number.

D.2.1 Reduction from Densest k-Subgraph

For a simple graph J on N vertices, $e_J(U)$ is the number of edges induced by $U \subseteq V(J)$, and the *bipartite double cover* G of J has two copies L, R of $V(J)$ and edges $u_L w_R, w_L u_R$ for every $uw \in E(J)$. The graph G has $2N$ vertices, and each vertex of G has the same degree as the corresponding vertex of J, which is below N. For $k \ge 2$, the two copies $S \subseteq L$ and $T \subseteq R$ of a k-clique in J are cohesive with $d(S,T) = k - 1$, since every vertex has $k-1$ neighbors in the opposite set.

Lemma D.4 (Extracting a dense k-vertex subgraph)**.** *For a simple graph J on at least $k \ge 2$ vertices and its bipartite double cover G, $0 < \rho \le 1$, $B \ge 1$, and nonempty sets $X \subseteq L$ and $Y \subseteq R$ with $|X|, |Y| \le Bk$ and $d(X,Y) \ge \rho(k-1)$, a set U of k vertices of J can be found in polynomial time with*

$$e_J(U) \ge \frac{\rho^2}{16B}\binom{k}{2}\,.$$

Proof. For $h \coloneqq \lfloor k/2 \rfloor$, we take X' to be the $\min\{|X|, h\}$ vertices of X with the most neighbors in Y, and Y' the $\min\{|Y|, h\}$ vertices of Y with the most neighbors in X'.

Since $|X'| + |Y'| \le 2h \le k$, we form U from the corresponding vertices of J, adding arbitrary vertices until $|U| = k$. Each edge of J has at most two copies in the double cover, so $e_J(U) \ge e(X', Y')/2$.

The vertices of X' have average degree into Y at least $e(X,Y)/|X|$, so $e(X',Y) \geq (|X'|/|X|)e(X,Y)$, and similarly $e(X',Y') \geq (|Y'|/|Y|)e(X',Y)$. Therefore,

$$e(X',Y') \geq \frac{|X'|}{|X|}\frac{|Y'|}{|Y|}e(X,Y) = \frac{h^2 e(X,Y)}{\max\{|X|,h\}\max\{|Y|,h\}}\,.$$

By the density assumption, $e(X,Y) \geq \rho(k-1)\sqrt{|X||Y|}$ and, since $d(X,Y) \leq \sqrt{e(X,Y)}$, also $e(X,Y) \geq \rho^2(k-1)^2$. If one side has at most h vertices, one factor in the denominator of the averaging bound is h and the other is at most Bk. We therefore keep at least an $h/(Bk)$ fraction of the edges, so $e(X',Y') \geq h\rho^2(k-1)^2/(Bk)$. Otherwise, $|X|,|Y| > h$, so the denominator in the averaging bound is $|X||Y|$. Since $\sqrt{|X||Y|} \leq Bk$,

$$e(X',Y') \geq \frac{h^2\rho(k-1)}{\sqrt{|X||Y|}} \geq \frac{h^2\rho(k-1)}{Bk} \geq \frac{h\rho^2(k-1)^2}{2Bk}\,,$$

using $h \geq (k-1)/2$ and $\rho \leq 1$. In either case,

$$e_J(U) \geq \frac{h\rho^2(k-1)^2}{4Bk} \geq \frac{\rho^2}{16B}\binom{k}{2}\,,$$

using $h \geq (k-1)/2$ and $k-1 \geq k/2$. □

Lemma D.5. *For a simple graph J on $N \geq k \geq 2$ vertices and its bipartite double cover G, $0 < \rho \leq 1$, $B \geq 1$, and an algorithm* A *that returns $\perp$ or nonempty sets $X \subseteq L$ and $Y \subseteq R$ with $|X|,|Y| \leq Bk$ from every $v \in V(G)$, we can find, with $2N$ calls to* A *and polynomial additional time, a set U of k vertices of J such that, if $\mathsf{A}(G,v,k)$ returns sets X,Y with $d(X,Y) \geq \rho(k-1)$ for some v, then*

$$e_J(U) \geq \frac{\rho^2}{16B}\binom{k}{2}\,.$$

Proof. We run A from every vertex of G, apply the construction in Lemma D.4 to each output X,Y, and keep a set spanning the most edges. If every call returns $\perp$, we take any k vertices. If A returns sets X,Y with $d(X,Y) \geq \rho(k-1)$, then by Lemma D.4 the corresponding set of k vertices satisfies the claimed bound, as does the set we keep. We construct the double cover and select the sets of k vertices in polynomial time. □

D.2.2 The limitation under SPCH

Definition D.6 (Strongish Planted Clique Hypothesis [Manurangsi et al., 2021, Hypothesis 1])**.** *The* Strongish Planted Clique Hypothesis *(SPCH) is the assumption that, for some $0 < \theta < 1/2$, no randomized algorithm running in time $n^{o(\log_2 n)}$ distinguishes the random graph $\mathcal{G}(n,\frac{1}{2})$ on n vertices, in which each edge is present independently with probability $1/2$, from the same graph $\mathcal{G}(n,\frac{1}{2},\lceil n^\theta\rceil)$ with a clique added on $\lceil n^\theta\rceil$ uniformly chosen vertices, with success probability at least $2/3$ under each distribution. The probability is over the input graph and the algorithm's randomness.*

We use the following hardness theorem for Densest k-Subgraph, with the promise that the input contains a k-clique.

Theorem D.7 ([Manurangsi et al., 2021, Theorem 2])**.** *Assuming SPCH, for every β with $\beta(k) = o(k)$ and every computable f, no algorithm, given an N-vertex graph J and k such that J contains a k-clique, returns in time $f(k)N^{O(1)}$ a set U of k vertices with*

$$e_J(U) \geq \frac{1}{\beta(k)}\binom{k}{2}\,.$$

Theorem D.8 (Limitations on the output size under SPCH)**.** *Assuming SPCH, for a constant $0 < \rho \leq 1$ and $\phi(k) = o(k^2)$, if an algorithm* A *as in Definition D.1 has output bound ϕ and runs in time $f(k)|V|^{O(1)}$ on every input (G,v,k), for a computable function f of k alone, then, for some $k \geq 2$, there is a simple bipartite graph G with cohesive sets $S \subseteq L$ and $T \subseteq R$ of k vertices each, such that from every $v \in S \dot\cup T$ the output is $\perp$ or has density below $\rho\, d(S,T)$.*

Proof of Theorem D.8. Suppose, for a contradiction, that for all cohesive target sets $S \subseteq L$ and $T \subseteq R$, the algorithm A returns sets X, Y with $d(X,Y) \geq \rho\, d(S,T)$ from some $v \in S \mathbin{\dot\cup} T$. For J with a k-clique, the two copies S, T of the clique in the double cover are cohesive with $d(S,T) = k-1$, so A returns sets X, Y with $d(X,Y) \geq \rho(k-1)$ from some vertex, and, by Lemma D.5 with $B = 1 + \phi(k)/k$, we can find k vertices spanning at least $\binom{k}{2}/\beta(k)$ edges, where

$$\beta(k) \coloneqq \frac{16(k+\phi(k))}{\rho^2 k} = o(k)\,.$$

The running time is $f'(k)N^{O(1)}$ for a computable f', contradicting Theorem D.7. □

Remark. We do not require β in Theorem D.7 to be nondecreasing. In Manurangsi et al. (2021, Section 4.1), the ratio is a function $g(k) \leq k$ growing with k, so we replace β with the nondecreasing $\tilde\beta(k) \coloneqq \max_{1 \leq j \leq k} \beta(j)$. For every $\delta > 0$, with $C_\delta \geq 0$ bounding $\beta(j)$ for the finitely many j with $\beta(j) > \delta j$, $\tilde\beta(k) \leq \max\{C_\delta, \delta k\}$, so $\tilde\beta = o(k)$. In particular, $\tilde\beta(k) \leq k$ for all sufficiently large k, and exhaustive search handles the remaining values of k.

Corollary D.9 (Approximation and output size under SPCH)**.** *Assuming SPCH, for $\alpha(k) \geq 1$ with $\alpha(k) = o(\sqrt{k})$ and $\phi(k) = o(k^2/\alpha(k)^2)$, no algorithm as in Definition D.1 with output bound ϕ and running time $f(k)|V|^{O(1)}$ for a computable f guarantees, for every $k \geq 2$ and every cohesive target S, T of k vertices per side, an output of density at least $d(S,T)/\alpha(k)$ from some starting vertex in $S \mathbin{\dot\cup} T$.*

Proof. For such an algorithm, by Lemma D.5 with $\rho = 1/\alpha(k)$ and $B = 1 + \phi(k)/k$, we obtain an approximation ratio $\beta(k) \coloneqq 16\alpha(k)^2(k+\phi(k))/k$ for Densest k-Subgraph, with

$$\frac{\beta(k)}{k} = 16\left(\frac{\alpha(k)^2}{k} + \frac{\alpha(k)^2\phi(k)}{k^2}\right) \longrightarrow 0\,.$$

The running time is $f'(k)N^{O(1)}$ for a computable f', contradicting Theorem D.7. □

Randomized algorithms. We also extend the proof of Theorem D.8 to a randomized A with the same output bound and running time on every run. We assume that for every cohesive target S, T there is a vertex $v \in S \mathbin{\dot\cup} T$ from which A returns sets X, Y with $d(X,Y) \geq \rho\, d(S,T)$ with probability at least $2/3$. If J contains a k-clique, then with probability at least $2/3$ the set U of Lemma D.5 spans at least $\binom{k}{2}/\beta(k)$ edges, with β as in the proof of Theorem D.8. We can compute $e_J(U)$ for each returned set, so we repeat the procedure of Lemma D.5 independently five times and keep a set spanning the most edges, obtaining such a set with probability at least 0.99. Any fixed positive success probability can be amplified to 0.99 with a constant number of independent repetitions.

The reduction from planted clique in Manurangsi et al. (2021, Section 4.1) is randomized, with completeness and soundness each holding with probability at least 0.9, so, using the amplified procedure inside that reduction, we distinguish the two distributions of Definition D.6 with success probability at least $0.9 \cdot 0.99 > 2/3$, contradicting SPCH. For Corollary D.9, we use the same proof with $\rho = 1/\alpha(k)$ and the corresponding $\beta(k) = o(k)$.

Quadratic output on the clique-promise instances. On the double covers used in this reduction, we can achieve a quadratic output bound in time $k^{O(k)}N^{O(1)}$. If a vertex of J has at least k^2 neighbors, one of its copies and k^2 of that copy's neighbors form a star of density k. Otherwise, every neighborhood in J has fewer than k^2 vertices, and we find the promised k-clique by testing all $(k-1)$-subsets of each neighborhood together with its center. The two copies of the clique have density $k-1$. Thus, on these instances, subquadratic output is essential to the lower bound even with full graph access. This does not give a local algorithm with quadratic output and running time polynomial in k for arbitrary cohesive targets.

D.2.3 The limitation under ETH

Definition D.10 (Exponential Time Hypothesis [Impagliazzo & Paturi, 2001])**.** *The* Exponential Time Hypothesis *(ETH) is the assumption that 3-SAT for formulas with m clauses cannot be decided*

in time $2^{o(m)}$. By the sparsification lemma of Impagliazzo et al. (2001), this is equivalent to the formulation in terms of variables.

We use the following hardness theorem, again with a clique promise.

Theorem D.11 ([Manurangsi, 2017, Theorem 1])**.** *Assuming ETH, there is a constant $c_0 > 0$ such that no polynomial-time algorithm, given a graph J on $N \geq 4$ vertices and an integer $2 \leq k \leq N$, distinguishes the case that J contains a k-clique from the case that every k-vertex set spans at most $\binom{k}{2}/\chi(N)$ edges, where*

$$\chi(N) \coloneqq N^{1/(\log_2 \log_2 N)^{c_0}} .$$

Theorem D.12 (Limitations on the output size under ETH)**.** *Assuming ETH, for constants $0 < \rho \leq 1$ and $c \geq 1$, if a deterministic polynomial-time algorithm A returns $\perp$ or nonempty sets $X \subseteq L$ and $Y \subseteq R$ with $|X|, |Y| \leq c\,k \log_2^c(2 + k + \Delta)$ on every input (G, v, k), then, for some $k \geq 2$, there is a simple bipartite graph G with cohesive sets $S \subseteq L$ and $T \subseteq R$ of k vertices each, such that from every $v \in S \,\dot\cup\, T$ the output is $\perp$ or has density below $\rho\, d(S, T)$. The same conclusion holds for running time polynomial in k, Δ and $\log_2 |V|$.*

Proof. Suppose, for a contradiction, that for all cohesive target sets $S \subseteq L$ and $T \subseteq R$, the algorithm A returns sets X, Y with $d(X, Y) \geq \rho\, d(S, T)$ from some $v \in S \,\dot\cup\, T$. For an N-vertex graph J and $k \leq N$, the double cover has $\Delta < N$, so the output bound is at most Bk for $B \coloneqq c\lceil \log_2(4N) \rceil^c$, and we apply the procedure of Lemma D.5. If J contains a k-clique, its two copies S, T are cohesive with $d(S, T) = k - 1$, so A returns sets X, Y with $d(X, Y) \geq \rho(k - 1)$ from some vertex, and the set U of Lemma D.5 satisfies

$$e_J(U) \geq \frac{\rho^2}{16B}\binom{k}{2} > \frac{1}{\chi(N)}\binom{k}{2}$$

for all sufficiently large N, since $\chi(N)$ grows faster than every power of $\log_2 N$. In the other case of the promise, every set of k vertices, including U, spans at most $\binom{k}{2}/\chi(N)$ edges. By comparing $e_J(U)$ with $\binom{k}{2}/\chi(N)$, we can therefore distinguish the two cases of Theorem D.11 in polynomial time, with exhaustive search for bounded N. Finally, a running time polynomial in k, Δ and $\log_2 |V|$ is polynomial in $|V|$ when $k \leq |V|$ and $\Delta < |V|$. □

Why the limitation under ETH is weaker. We cannot obtain the quadratic bound of Theorem D.8 from the hard instances used under ETH, because they have $N = k^{1+o(1)}$ vertices. Manurangsi (2017, Section 3.2) constructs graphs whose vertices are partial assignments to ℓ of the n variables of a 3-SAT formula. Each graph has $N = 2^\ell\binom{n}{\ell}$ vertices. If the formula is satisfiable, the restrictions of a satisfying assignment to all ℓ-element sets of variables form a clique of size $k = \binom{n}{\ell}$. With the parameter choices in that proof, $\ell = o(n)$. Hence

$$\frac{\log_2 N}{\log_2 k} = 1 + \frac{\ell}{\log_2 k} \leq 1 + \frac{1}{\log_2(n/\ell)} = 1 + o(1) .$$

For fixed $0 < \delta < 1$ and large k, the entire double cover has fewer than $k^{2-\delta}$ vertices per side, and nonempty subsets $X \subseteq L$ and $Y \subseteq R$ maximizing $d(X, Y)$ can then be found in polynomial time (Charikar, 2000, Section 4, Theorem 3), meeting the output bound $k^{2-\delta}$ with approximation ratio one.