Directed Reachability from Cut Queries in $\widetilde{O}(n^{3/2})$ Queries

Abstract

We study directed s–t reachability when an unknown digraph $G = (V, E)$ is accessible only through an outgoing-cut oracle: a query $S \subseteq V$ returns $F(S) = |E(S, V \setminus S)|$. We give a randomized adaptive algorithm using $n^{3/2} \log^{O(1)} n$ queries and succeeding with probability $1 - n^{-\Omega(1)}$; computation between queries is unrestricted. The conceptual core is a rank-n fractional-orientation linear program. An inverse-polynomially accurate solution induces an implicit threshold digraph having exactly the same zero out-cuts, and hence the same reachability relation, as G. Although the LP has up to $\Theta(n^2)$ hidden rows, we implement a rank-sensitive Lee–Sidford box-LP method without materializing them. Our main oracle primitive is a spread-independent weighted spectral sparsifier for pair-evaluable conductances, constructed from cut queries by filtered weighted spanners, bundle leverage bounds, and whole-pair sampling. Duplicate hidden copies are kept block-symmetric. Approximate normal solves are handled by the standard stable Lee–Sidford interface, and a public star right inverse keeps every primal update exactly feasible. The result breaks the quadratic query barrier for directed reachability posed in recent work on the cut-query model.

Note: The algorithm and main proof were discovered by an agentic harness developed by George Z. Li and Janardhan Kulkarni that is built on OpenAI's GPT-5.6 Sol and Gemini 3.1 Pro.

1 Introduction

Let $G = (V, E)$ be an unknown directed graph on n labeled vertices. For a set $S \subseteq V$, define its outgoing cut value by

$$F(S) := |E(S, V \setminus S)|.$$

The algorithm may adaptively query F on arbitrary sets, but receives no other information about E. Given $s, t \in V$, the goal is to decide whether G contains a directed path from s to t while minimizing the number of queries. We do not charge computation between queries.

Directed reachability is a particularly stark test of the power of this oracle. The direct reconstruction algorithm uses $O(n^2)$ pair tests, and the outgoing-cut function does not determine the orientation of each arc: reversing a directed cycle leaves every cut value unchanged. Nevertheless, F directly identifies all zero out-cuts, and those cuts characterize reachability. The question of obtaining any $O(n^{2-\gamma})$ bound, for a constant $\gamma > 0$, was highlighted as open in recent work on cut queries [4] and as an explicit open problem by Nanongkai [8].

Our main result gives an exponent 3/2.

Theorem 1.1 (Main theorem). *Let $G = (V, E)$ be a loopless simple directed graph on n vertices (so there is at most one copy of each ordered arc), and let $s, t \in V$. There is a randomized adaptive algorithm which, given oracle access to $F(S) = |E(S, V \setminus S)|$, decides whether t is reachable from s, succeeds with probability $1 - n^{-\Omega(1)}$, and makes*

$$n^{3/2} \log^{O(1)} n$$

oracle queries. More generally, if G is a loopless directed multigraph and the total number of arc copies in both directions on every unordered pair is at most a constant C, the same algorithm uses $O(Cn^{3/2}\log^{O(1)} n)$ queries. (The simple-graph case has $C \le 2$.) Computation between queries is unrestricted, and the displayed query bound is a deterministic cap for every random tape.

Self-loops never affect a cut or reachability between distinct vertices and may be discarded. The case $s = t$ is immediate.

1.1 Overview of the proof

The proof has four layers.

Cut algebra. For each unordered pair $\{u, v\}$, let

$$c_{uv} = \mathbf{1}[(u, v) \in E] + \mathbf{1}[(v, u) \in E] \in \{0, 1, 2\},$$

and let H be the undirected multigraph with c_{uv} copies of $\{u, v\}$. The imbalance

$$b_v = F(\{v\}) - F(V \setminus \{v\})$$

is the outdegree minus the indegree of v. Direct cancellation gives

$$2F(S) = c_H(\delta S) + b(S).$$

Thus the oracle exposes both the cut function of H and the imbalance vector b, even though it does not expose the individual directions.

A fractional orientation certificate. Reference-orient every pair $u < v$ from u to v and associate a variable $z_{uv} \in [-1, 1]$. It represents the fractional orientation

$$p_{u\to v} = \frac{1 + z_{uv}}{2}, \qquad p_{v\to u} = \frac{1 - z_{uv}}{2}.$$

We construct z whose divergence is extremely close to $(1 - \eta)b$, where $\eta = 2^{-\lceil 12\log_2 n\rceil} \le n^{-12}$. If $P_z(S)$ is the total fractional mass leaving S, then

$$2P_z(S) = 2(1 - \eta)F(S) + \eta c_H(\delta S) + \rho(S), \qquad \|\rho\|_1 \le n^{-20}.$$

It follows that $P_z(S) < n^{-4}$ when $F(S) = 0$, whereas $P_z(S) > 1/2$ when $F(S) > 0$. Thresholding individual fractional arcs at n^{-4} therefore preserves every zero out-cut and hence every reachability relation.

A hidden rank-n box LP. We obtain z from a Phase-I box LP with one conceptual row per copy in H and two public star rows per vertex. It may have $\Theta(n^2)$ rows but has rank n. Its optimum is zero: the unknown original orientation itself gives a block-symmetric witness. We apply the rank-sensitive box-LP framework of Lee and Sidford [7], which uses $\sqrt{n}\log^{O(1)} n$ weighted-system-equivalent calls.

Implicit implementation. Each call is implemented in $n \log^{O(1)} n$ cut queries. A new weighted spectral primitive constructs a sparsifier of

$$\sum_{u<v} c_{uv} D_{uv}(e_u - e_v)(e_u - e_v)^\top$$

whenever D_{uv} is computable from the endpoints. It yields both a constant-condition preconditioner and simultaneous leverage estimates for all hidden rows. Bit-class decomposition implements all coordinate aggregates. A KKT formula implements the mixed-norm support oracle. Finally, an explicit star right inverse repairs each approximate direction exactly, so feasibility never drifts. Multiplying the call count by the per-call query cost gives

$$\sqrt{n} \log^{O(1)} n \cdot n \log^{O(1)} n = n^{3/2} \log^{O(1)} n.$$

How to read the proof. The graph-theoretic correctness spine is Sections 2, 3, and 7: cut algebra exposes an undirected support, the fractional certificate preserves zero out-cuts, and the final section turns an approximate LP solution into a reachability answer. On a first reading, Theorem 5.3 may be treated as a black box while following that spine. Section 4 constructs the spectral oracle used to realize the black box; Section 5 implements its path-following primitives; and Section 6 checks the short interface between those cut-query primitives and the stable Lee–Sidford solver. Thus a reader interested first in the reduction need not understand interior-point methods before reading the correctness argument.

Core notation. The symbols used across several layers are collected here for reference.

Symbol	Meaning
$F(S)$	number of input arc copies directed from S to $V \setminus S$
c_{uv}, H	total multiplicity on $\{u, v\}$, and the resulting undirected multigraph
b	input outdegree-minus-indegree vector; $d = (1-\eta)b$ is the Phase-I right-hand side
a_e	reference-oriented incidence vector $e_u - e_v$ for $e = \{u, v\}$, $u < v$
$z_e, p_{u \to v}$	common hidden-pair variable and its induced fractional directed mass
J, τ	threshold digraph and threshold $\tau = n^{-4}$
A, x, M	full Phase-I row matrix, its row-variable vector, and its number of rows

Section 5 gives a second, solver-state table before introducing the IPM notation.

1.2 Our contributions

Beyond the main bound, the proof provides the following reusable ingredients.

(i) A reduction from outgoing-cut reachability to a rank-n fractional-orientation LP whose approximate solutions preserve the complete zero-outcut family.

(ii) A spread-independent spectral sparsifier for unknown weighted multigraphs with pair-evaluable conductances, using $O(\varepsilon^{-2} n \log^8(n/(\varepsilon\delta)))$ cut queries.

(iii) A block-symmetric cut-query implementation of a rank-sensitive box-LP method, including simultaneous leverage estimation without assigning independent random signs to indistinguishable parallel copies.

(iv) A short interface from cut-query primitives to the standard stable Lee–Sidford execution: residual-certified Richardson iteration gives inverse-polynomial solve accuracy, and a public star right inverse preserves affine feasibility exactly.

1.3 Related work

Rubinstein, Schramm, and Weinberg [9] introduced the modern cut-query study of exact graph cuts and obtained $\widetilde{O}(n^{5/3})$ queries for undirected minimum s–t cut. Subsequent work obtained linear-query undirected edge connectivity via star contraction [2], deterministic $\widetilde{O}(n^{5/3})$ undirected flow and connectivity bounds [1], and an $\widetilde{O}(n^{8/5})$ randomized bound for undirected minimum s–t cut [4]. The last work explicitly notes that no subquadratic upper bound was known even for directed reachability. Our oracle is asymmetric: it returns arcs leaving S, rather than undirected edges crossing S.

For linear programming we use the rank-sensitive box-LP framework of Lee and Sidford [7]. Its approximate Lewis-weight interface is part of that framework; the stability analysis for inverse-polynomially accurate linear solves is given in their earlier primal path-following work [6, Section 8]. We use those results as a solver interface rather than reproving path-following stability.

Our spectral routine builds on weighted spanners of Baswana and Sen [3], the bundle-spanner sampling paradigm of Koutis [5], and matrix Bernstein inequalities in the form developed by Tropp [10]. The new points here are the filtered cut-query simulation, frozen phase snapshots, multiplicity-block sampling, and an interface for endpoint-computable weights of arbitrary spread.

2 Preliminaries and cut-query algebra

For an undirected multigraph H, let $c_H(\delta S)$ denote the number of copies crossing $(S, V \setminus S)$, and let $c_H(A, B)$ denote the number of copies with one endpoint in each of two disjoint sets A, B. We use $L \preceq M$ for the Loewner order on symmetric matrices. For a positive semidefinite matrix L, $L^\dagger$ is its Moore–Penrose pseudoinverse. The effective resistance of $e = uv$ in a weighted graph with Laplacian L is $(e_u - e_v)^\top L^\dagger (e_u - e_v)$.

Definition 2.1 (Pair-evaluable scalar)**.** A scalar q_{uv} is *pair-evaluable* if, from the public transcript and the labels u, v, the algorithm can compute q_{uv} without querying whether $c_{uv} > 0$. Offline computation and the size of the representing arithmetic circuit are not charged.

This definition allows dependence on public hashes, current endpoint states, known cluster labels, and all previous answers.

Query-model convention (optional on a first reading). As is standard for query complexity, exact real arithmetic and comparisons between queries are free. A primal coordinate or a path weight may therefore be represented by a public arithmetic circuit rather than by a fixed-precision word. Whenever such a value enters a cut-query reduction, certified evaluation freezes a dyadic approximation of the accuracy requested by the solver. Lemma 2.5 bounds the resulting error, and Section 6 checks that the stable Lee–Sidford interface absorbs it.

Filtered search may evaluate a row formula on an absent candidate pair. We make every such circuit total by clipping virtual slacks away from zero; this choice is irrelevant to an aggregate because an absent pair is multiplied by $c_{uv} = 0$. Before using a batch, one filtered search rejects if an *actual* row is outside the domain required by the original formula. The Lee–Sidford interiority and conditioning bounds imply that this never happens on a successful execution. Fixed caps make an unsuccessful random execution terminate within the advertised query bound.

Lemma 2.2 (Cut algebra)**.** *Let*

$$c_{uv} = \mathbf{1}[(u, v) \in E] + \mathbf{1}[(v, u) \in E] \quad \textit{and} \quad b_v = F(\{v\}) - F(V \setminus \{v\}).$$

Let H be the undirected multigraph with multiplicities c_{uv}. Then for every $S \subseteq V$,

$$2F(S) = c_H(\delta S) + b(S). \tag{1}$$

Moreover, for disjoint $A, B \subseteq V$,

$$c_H(A, B) = F(A) + F(B) - F(A \cup B). \tag{2}$$

Proof. For every arc internal to S or to $V \setminus S$, both sides of (1) receive zero. An arc from S to $V \setminus S$ contributes two to the right-hand side: one through $c_H(\delta S)$ and one through $b(S)$. An arc in the opposite direction contributes one and minus one, hence zero. This proves (1). In $F(A) + F(B) - F(A \cup B)$, all arcs from $A \cup B$ to its complement cancel; the surviving arcs are precisely those in either direction between A and B, proving (2). □

Corollary 2.3 (Filtered-neighbor oracle)**.** *Fix $u \notin T$. Three outgoing-cut queries determine $\sum_{v \in T} c_{uv}$ for every publicly computable candidate set T. Consequently, a qualifying neighbor can be found by binary search in $O(\log n)$ such tests.*

The candidate set may depend on the entire public transcript but not on an unqueried adjacency bit. This distinction is used repeatedly below.

Lemma 2.4 (Bit-class aggregation)**.** *Let q_{uv} be a signed B-bit pair-evaluable scalar. All values*

$$Q_u = \sum_{v \neq u} c_{uv} q_{uv}, \qquad u \in V,$$

can be computed using $O(nB)$ outgoing-cut queries. The same bound applies to signed incidence products, predicate-restricted sums, and weighted Laplacian products whose coordinate data are B-bit pair circuits.

Proof. Write the positive and negative parts of q_{uv} in binary. For a fixed u, sign, and bit position j, form the public set of vertices whose corresponding bit is one and apply Corollary 2.3. Recombining the answers gives Q_u exactly for the stored fixed-point values. Incidence signs and public predicates are incorporated when the candidate sets are formed. □

Example: reconstructing one weighted sum. Fix u and suppose three candidate endpoints v_1, v_2, v_3 have nonnegative three-bit values

$$q_{uv_1} = 5 = (101)_2, \qquad q_{uv_2} = 2 = (010)_2, \qquad q_{uv_3} = 3 = (011)_2.$$

The public endpoint sets for the 4-, 2-, and 1-bits are respectively

$$T_2 = \{v_1\}, \qquad T_1 = \{v_2, v_3\}, \qquad T_0 = \{v_1, v_3\}.$$

Using the rectangle identity to obtain $c_H(\{u\}, T_j) = \sum_{v \in T_j} c_{uv}$, the algorithm reconstructs

$$\sum_{i=1}^{3} c_{uv_i} q_{uv_i} = 4c_H(\{u\}, T_2) + 2c_H(\{u\}, T_1) + c_H(\{u\}, T_0).$$

Signed values are handled by performing the same calculation separately for their positive and negative parts. Notice that the sets T_j depend only on the pair-evaluable values; they do not reveal which candidate pairs actually occur.

Lemma 2.5 (Rounded circuit aggregation). *Let q_{uv} be a pair-evaluable circuit and suppose a public envelope gives $|q_{uv}| \le \Lambda$. If certified evaluation produces a signed dyadic $\widehat{q}_{uv}$ with*

$$|\widehat{q}_{uv} - q_{uv}| \le \beta,$$

then all sums

$$\widehat{Q}_u = \sum_{v \ne u} c_{uv} \widehat{q}_{uv}$$

are computed exactly with $O(n \log(\Lambda/\beta))$ outgoing-cut queries and satisfy

$$|\widehat{Q}_u - Q_u| \le \beta \sum_{v \ne u} c_{uv}.$$

If D and $\widehat{D}$ are diagonal row weights with $|D_i - \widehat{D}_i| \le \beta$, every row of A has squared norm at most two, and A has M rows, then

$$\left\| A^\top (\widehat{D} - D) A \right\|_2 \le 2M\beta. \tag{3}$$

The same conclusions hold for predicate-restricted sums and right-hand sides.

Proof. Apply Lemma 2.4 to the certified dyadic values. The scalar error bound follows by the triangle inequality. For the matrix statement,

$$\left\| \sum_i (\widehat{D}_i - D_i) a_i a_i^\top \right\|_2 \le \sum_i \beta \, \|a_i\|_2^2 \le 2M\beta.$$

□

Lemma 2.6 (Hidden maxima and predicates). *Let q_{uv} be a public pair-evaluable dyadic value and let $P(u,v)$ be a public predicate. The maximum of q_{uv} over actual support pairs satisfying P, an actual maximizing pair when one exists, and emptiness of the qualifying support can be found using $O(n \log n)$ outgoing-cut queries.*

Proof. For each fixed u, order all candidate endpoints v by (q_{uv}, v). Binary search this public order, using Corollary 2.3 on suffixes intersected with P, to find the largest actual qualifying neighbor of u. Take the best of the at most n returned candidates offline. The same search with a constant value implements emptiness and threshold predicates. □

3 A fractional-orientation certificate

Reference-orient each unordered pair $e = \{u, v\}$ from the smaller endpoint u to the larger endpoint v, and write $a_e = e_u - e_v \in \mathbb{R}^n$. Each of the c_e conceptual copies receives a variable $z_{e,j}$. The implementation will keep all copies equal, but it is useful to retain the conceptual copies when defining the LP.

What the Phase-I construction is doing. The reference sign fixes the meaning of every hidden coordinate: a positive z_{uv} contributes positive divergence at the smaller endpoint u and negative divergence at v. There are c_{uv} conceptual coordinates because an unordered pair may support several input arc copies. They are kept equal by the implementation, but retaining the copies makes their total incidence contribution exactly $c_{uv} a_{uv} z_{uv}$. The extra grounded root has no retained coordinate, so a star edge from v to the root has row e_v (and its reverse has row $-e_v$). The difference $y_v^+ - y_v^-$ can therefore absorb any remaining divergence at v. Minimizing $y_v^+ + y_v^-$ drives the ℓ_1 size of this public correction to zero. Finally, the factors $1 - \eta$ and $\eta/2$ keep the optimum away from the hidden box boundary, which supplies the strict interior required by the barrier method.

A three-vertex example. Suppose $\{1,2\}$ supports only the arc $1 \to 2$, while $\{2,3\}$ supports one arc in each direction. Then

$$c_{12} = 1, \quad \Delta_{12} = 1, \qquad c_{23} = 2, \quad \Delta_{23} = 0,$$

and the reference rows are $a_{12} = e_1 - e_2$ and $a_{23} = e_2 - e_3$. The block-symmetric zero-objective witness used below sets

$$z_{12,1} = 1 - \eta, \qquad z_{23,1} = z_{23,2} = 0.$$

Thus the single directed pair supplies its shrunken original orientation, whereas the antiparallel pair has zero net divergence. At the known starting point all hidden variables are instead zero, and the two explicit star variables at each vertex supply the entire target divergence d.

Set

$$\eta = 2^{-\lceil 12 \log_2 n \rceil} \le n^{-12}, \qquad d = (1-\eta)b, \qquad K_0 = 1 + \max_v |d_v|.$$

Add a new root and ground its coordinate. For every original vertex v, add two explicit rows e_v and $-e_v$, with variables y_v^+ and y_v^-. Consider the closed-box LP

$$\begin{aligned}
\min \quad & \sum_{v \in V} (y_v^+ + y_v^-) \\
\text{s.t.} \quad & \sum_e \sum_{j=1}^{c_e} a_e z_{e,j} + \sum_{v \in V} e_v (y_v^+ - y_v^-) = d, \\
& -1 + \eta/2 \le z_{e,j} \le 1 - \eta/2, \\
& 0 \le y_v^+, y_v^- \le 2K_0.
\end{aligned} \tag{4}$$

The barrier algorithm operates in the strict interior of these boxes. If $B \in \mathbb{R}^{m \times n}$ contains one row $a_e^\top$ per conceptual hidden copy, then the full row matrix and primal vector are

$$M = m + 2n, \qquad A = \begin{bmatrix} B \\ I_n \\ -I_n \end{bmatrix} \in \mathbb{R}^{M \times n}, \qquad x = (z, y^+, y^-) \in \mathbb{R}^M,$$

and the equality in (4) is $A^\top x = d$. The conceptual hidden row count is also known without reconstructing the support:

$$m = |E| = \sum_{v \in V} F(\{v\}),$$

because every directed arc is counted once at its tail.

Lemma 3.1 (Phase-I structure)**.** *The constraint matrix of* (4) *has rank n. The point*

$$z_{e,j} = 0, \qquad y_v^+ = K_0 + d_v/2, \qquad y_v^- = K_0 - d_v/2$$

is explicitly known, exactly feasible, and strictly interior. Moreover, the optimum of (4) *is zero, and there is an optimal hidden assignment constant across all copies of an unordered pair.*

Proof. The e_v rows form an identity matrix, so the rank is n. The displayed point has signed star flow d and zero hidden flow. Since $K_0 > |d_v|$, both star variables lie strictly between 0 and $2K_0$.

For $e = \{u, v\}$ with $u < v$, define

$$\Delta_e = \mathbf{1}[(u,v) \in E] - \mathbf{1}[(v,u) \in E].$$

For every copy set

$$z_{e,j} = \frac{(1-\eta)\Delta_e}{c_e}, \qquad y_v^+ = y_v^- = 0.$$

Its divergence is $(1-\eta)b = d$. Because $|\Delta_e| \le c_e$, each hidden coordinate lies in $[-1+\eta, 1-\eta]$, strictly inside the narrower hidden box. If both antiparallel arcs exist, then $c_e = 2$ and $\Delta_e = 0$, so the common-copy assignment remains valid. The objective is nonnegative and the displayed witness has value zero. $\square$

Suppose now that a common pair value z_e satisfies $|z_e| \le 1$. For $e = \{u, v\}$, $u < v$, define

$$p_{u\to v} = \frac{1+z_e}{2}, \qquad p_{v\to u} = \frac{1-z_e}{2}. \tag{5}$$

For an arbitrary ordered pair, this means

$$p_{u\to v} = \begin{cases} (1+z_{uv})/2, & u < v, \\ (1-z_{vu})/2, & v < u. \end{cases}$$

Let $P_z(S)$ be the sum, over all conceptual copies crossing the cut, of the fractional mass directed from S to $V \setminus S$.

Lemma 3.2 (Zero/positive outcut gap)**.** *Assume*

$$|z_e| \le 1 \quad \textit{for every actual pair}, \qquad \left\| B^\top z - (1-\eta)b \right\|_1 \le n^{-20}, \tag{6}$$

where $B^\top z = \sum_e c_e a_e z_e$. Put $\rho = B^\top z - (1-\eta)b$. Then, for every $S \subseteq V$,

$$2P_z(S) = 2(1-\eta)F(S) + \eta c_H(\delta S) + \rho(S). \tag{7}$$

Consequently, for all sufficiently large n,

$$F(S) = 0 \Longrightarrow P_z(S) < n^{-4}, \qquad F(S) > 0 \Longrightarrow P_z(S) > 1/2.$$

Proof. For any fractional orientation, twice the outgoing mass is the undirected crossing multiplicity plus the fractional divergence. Thus

$$2P_z(S) = c_H(\delta S) + (1-\eta)b(S) + \rho(S).$$

Substitute $b(S) = 2F(S) - c_H(\delta S)$ from Lemma 2.2. If $F(S) = 0$, then

$$P_z(S) \le \frac{\eta n^2 + n^{-20}}{2} < n^{-4}.$$

If $F(S) \ge 1$, the nonnegative $\eta c_H(\delta S)$ term and $|\rho(S)| \le n^{-20}$ give

$$P_z(S) \ge 1 - \eta - n^{-20}/2 > 1/2.$$

The finitely many smaller n can be handled by learning all pairs, changing only the absolute constant in the theorem. $\square$

If the total multiplicity of an unordered pair is at most a fixed C, the only change in this argument is $c_H(\delta S) \le Cn^2$. Thus the same choices $\eta \le n^{-12}$ and $\tau = n^{-4}$ give the stated gap for all sufficiently large n; the finitely many remaining sizes are again absorbed into the absolute constant.

Define the implicit threshold digraph

$$J = \{u \to v : c_{uv} > 0 \text{ and } p_{u\to v} \ge \tau\}, \qquad \tau = n^{-4}. \tag{8}$$

The graph J need not be a directed subgraph of G: it may orient an underlying pair opposite to its sole input arc. What is preserved is the zero-outcut family. Moreover, J is not an orientation in the sense of choosing exactly one direction per pair. Since $p_{u\to v} + p_{v\to u} = 1$ and $\tau < 1/2$, both ordered arcs may pass the threshold. The quantity $P_z(S)$ counts fractional mass once for each of the c_{uv} conceptual copies, whereas J records only whether the thresholded ordered pair exists; parallel copies do not create parallel arcs in J.

Lemma 3.3 (Threshold preservation)**.** *For every $S \subseteq V$,*

$$F(S) = 0 \iff J \textit{ has no arc from } S \textit{ to } V \setminus S.$$

Consequently, G and J have identical reachability relations.

Proof. If $F(S) = 0$, Lemma 3.2 gives $P_z(S) < \tau$, so no individual fractional arc leaving S reaches threshold. If $F(S) > 0$, then $P_z(S) > 1/2$. At most Cn^2 conceptual copies cross the cut (with $C \le 2$ for a simple digraph), so if all their leaving masses were below $\tau = n^{-4}$ their sum would be at most $C/n^2 < 1/2$ for all sufficiently large n, a contradiction. As in Lemma 3.2, the finitely many remaining sizes are handled by adjusting the absolute constant.

For the reachability claim, t is not reachable from s exactly when the set of vertices reachable from s is an s-containing, t-excluding set with no outgoing arc. The family of such zero out-cuts is the same in G and J. □

Lemma 3.4 (Implicit search in J)**.** *Given a pair-evaluable circuit for z satisfying* (6)*, reachability from s in J can be computed using $O(n \log n)$ additional outgoing-cut queries.*

Proof. We give the search and its invariant explicitly. Maintain a discovered set R, initially $\{s\}$, and a queue containing the discovered vertices that have not yet been processed. Both are determined by the public query transcript. When u is removed from the queue, repeatedly form

$$T(u, R) = \{v \in V \setminus R : p_{u\to v} \ge \tau\}.$$

This set is public even though the support of H is hidden: membership uses only R and the pair-evaluable circuit for z, and does not test whether $c_{uv} > 0$.

If $T(u, R)$ is nonempty, use the rectangle identity to evaluate

$$c_H(\{u\}, T(u, R)) = F(\{u\}) + F(T(u, R)) - F(T(u, R) \cup \{u\}). \tag{9}$$

The value is positive exactly when some actual support pair $\{u, v\}$ has $v \in T(u, R)$. When it is positive, order $T(u, R)$ by the fixed public vertex order and bisect it. Apply (9) to the first half; if that count is positive, retain the first half, and otherwise retain the second. The retained half always contains an actual neighbor, so after $O(\log n)$ such tests a singleton $\{v\}$ remains. Add v to R and to the queue, recompute $T(u, R)$, and continue processing the same u. When $T(u, R)$ is empty or its count is zero, mark u processed. The algorithm terminates when the queue is empty and returns R.

We prove that the returned set is exactly the set reachable from s in J. Initially this is true for the only discovered vertex s. Whenever v is discovered while processing u, the positive singleton count gives $c_{uv} > 0$, and membership in $T(u, R)$ gives $p_{u\to v} \geq \tau$. Hence $u \to v$ is an arc of J. By induction, every vertex placed in R is reachable from s in J.

Conversely, consider the final failed test for a processed vertex u. At that moment there is no actual pair from u to any then-undiscovered v satisfying $p_{u\to v} \geq \tau$. The discovered set only grows afterwards, so there is also no J-arc from u to the complement of the final set R. Every vertex in the final R is eventually processed; therefore R is closed under outgoing arcs of J. Since $s \in R$, no directed path from s can leave R, proving that no J-reachable vertex was omitted.

There are at most $n - 1$ successful searches, because each discovers a new vertex, and at most one terminal failed search for each of at most n processed vertices. A successful or failed search uses $O(\log n)$ count tests (an empty candidate set needs none), and each count test uses three outgoing-cut queries, which may be issued in parallel. Thus the total is $O(n \log n)$ queries. The straightforward implementation uses $O(n \log n)$ adaptive test rounds; the theorem charges only the number of queries and does not claim a smaller round bound. □

4 Weighted spectral sparsification from cut queries

The normal matrices arising later are weighted graph Laplacians whose support is hidden but whose prospective conductance on a pair is endpoint-computable. This section supplies the required oracle.

Theorem 4.1 (Pair-evaluable weighted sparsification). *Let G be an unknown loopless directed multigraph on $[n]$, accessed through the outgoing-cut oracle $F_G(S) = |E_G(S, [n]\setminus S)|$, where cardinality counts arc copies. Let H be its underlying undirected multigraph, and let $c_e \in \{0, \dots, C\}$ be the total number of arcs of G in both directions on the unordered pair e, where C is a constant. Let $w_e \geq 0$ be symmetric and pair-evaluable, and assume that nonzero weights can be compared and output exactly. For $0 < \varepsilon \leq 1/2$ and $0 < \zeta < 1/2$, there is a randomized algorithm which uses*

$$O\left(C\varepsilon^{-2} n \log^8 \frac{n}{\varepsilon\zeta}\right)$$

queries to F_G and, with probability at least $1 - \zeta$, outputs a weighted graph Q such that

$$(1-\varepsilon)L \preceq L_Q \preceq (1+\varepsilon)L, \qquad L = \sum_e c_e w_e a_e a_e^\top.$$

The output has $O(C\varepsilon^{-2} n \log^6(n/(\varepsilon\zeta)))$ unordered pairs. The bound is independent of the aspect ratio of the nonzero w_e.

Interface with the later normal systems. The exact-weight hypothesis in Theorem 4.1 is an interface condition, not an assertion that the analytic path state is stored exactly. In the LP implementation, certified evaluation first freezes a positive dyadic coefficient $\widehat{D}_e$ for every prospective hidden incidence row. Once the duplicate copies on a pair have a common row state, as proved in Lemma 5.4, their hidden normal-matrix contribution is

$$L_{\text{hid}} = \sum_e c_e \widehat{D}_e a_e a_e^\top.$$

Thus Theorem 4.1 is called with $w_e = \widehat{D}_e$. This value is pair-evaluable because the row state is a public endpoint circuit, and it is exact at this interface because $\widehat{D}_e$ is dyadic. The $O(n)$ explicit star-row contributions are then added to the returned sparsifier offline. In particular, constructing a normal-system preconditioner or simultaneous leverage estimates never requires learning whether a prospective pair is present.

No undirected-cut oracle is assumed. The arc-by-arc proof of Lemma 2.2 is linear in arc multiplicities and therefore applies unchanged to this directed multigraph. Thus, for $v \notin Y$, where Y is determined by the public transcript,

$$\sum_{u \in Y} c_{\{u,v\}} = F_G(\{v\}) + F_G(Y) - F_G(Y \cup \{v\})$$

by Lemma 2.2. Thus one filtered-neighborhood test costs three outgoing-cut queries, and binary search for a qualifying neighbor costs $O(\log n)$ such tests. All bounds below include this translation.

Implicit residual supports. Every graph manipulated below is represented by a public survival predicate, not by an edge list. At recursion depth r, let $P_r(e)$ be a predicate computable from the endpoints and the public transcript, and define

$$\mathcal{E}_r = \{e : c_e > 0 \text{ and } P_r(e)\}, \qquad L_r = \sum_{e \in \mathcal{E}_r} c_e p_{\mathrm{samp}}^{-r} w_e a_e a_e^\top, \qquad p_{\mathrm{samp}} = \tfrac{1}{4}.$$

The symbol p_{samp} distinguishes this pair-sampling probability from the cluster-sampling probability used inside one spanner call. Initially $P_0(e)$ is the public test $w_e > 0$. When a public bundle $B_r \subseteq \mathcal{E}_r$ has been found, draw a fresh public independent Bernoulli hash $h_r(e)$ for every unordered endpoint pair, with $\mathbb{P}[h_r(e) = 1] = p_{\mathrm{samp}}$, and set

$$P_{r+1}(e) = P_r(e) \wedge [e \notin B_r] \wedge [h_r(e) = 1]. \tag{10}$$

The hash is evaluated from the endpoint labels and a public seed, so P_{r+1} remains pair-evaluable. Every filtered-neighbor query simply intersects its candidate set with the current predicate. Thus the algorithm does not query whether an absent pair survived, and it materializes a pair only when a filtered search actually returns it. Conditional on the preceding transcript, the hashes on the current actual pairs are independent fresh quarter-sampling decisions. This is the residual-support invariant used by both the spanner construction and the recursion. Within one spanner call, the phase support E_i is represented in the same way: conjoin P_r with the public block deletions and cluster predicates fixed by the completed phases. No predicate is changed while the queries for one frozen phase are being answered.

4.1 Filtered weighted spanners

Give every support pair e the distinct key

$$\mathrm{key}(e) = (1/w_e, \mathrm{ID}(e)),$$

with zero-weight pairs excluded. For a fixed vertex v, an arbitrary public predicate $P(v,u)$ is enforced by placing in the query set only those candidate endpoints satisfying P. Corollary 2.3 then supplies emptiness and minimum-neighbor queries.

We simulate the simultaneous weighted Baswana–Sen phases [3]. Let E_i and $\mathcal{C}_i$ be the support and clustering at the start of phase i, with $\mathcal{C}_1$ the singleton clustering, and put $p = n^{-1/k}$. In each phase $i < k$, sample every cluster independently with probability p and write $\mathcal{S}_i$ for the sampled family. A vertex in a sampled cluster makes no reports; set $\mathcal{R}_i(v) = \varnothing$ for such a vertex. For a vertex v outside the sampled clusters and $C \in \mathcal{C}_i$, define

$$E_i(v, C) = \{vu \in E_i : u \in C\}, \qquad e_i(v, C) = \arg \min_{e \in E_i(v,C)} \mathrm{key}(e),$$

and let $\kappa_i(v, C)$ be the key of this edge, or $+\infty$ if the set is empty. The distinct pair IDs make every finite minimum unique. Set

$$\sigma_i(v) = \min_{C \in \mathcal{S}_i} \kappa_i(v, C),$$

and, when finite, let $C_i^\star(v)$ be its minimizing cluster. The clusters reported by v are exactly

$$\mathcal{R}_i(v) = \begin{cases} \{C : \kappa_i(v, C) < \sigma_i(v)\} \cup \{C_i^\star(v)\}, & \sigma_i(v) < +\infty, \\ \{C : \kappa_i(v, C) < +\infty\}, & \sigma_i(v) = +\infty. \end{cases}$$

For every $C \in \mathcal{R}_i(v)$, insert $e_i(v, C)$ and record the entire block $E_i(v, C)$ for deletion. After all vertices have been processed, a vertex with finite $\sigma_i(v)$ joins $C_i^\star(v)$; sampled clusters and their members persist, while a vertex with infinite $\sigma_i(v)$ becomes inactive. Apply all recorded deletions simultaneously and then delete each pair whose endpoints have the same new cluster label. These operations define E_{i+1} and $\mathcal{C}_{i+1}$. In phase k, take $\mathcal{S}_k = \varnothing$ and report every neighboring cluster.

One phase, procedurally. The preceding definitions amount to the following frozen-snapshot procedure.

(1) Sample the old clusters, but do not yet change any cluster label or residual-support predicate.
(2) For each vertex v outside the sampled clusters, find its minimum-key edge to any sampled cluster; this determines $\sigma_i(v)$ and $C_i^\star(v)$.
(3) Enumerate the old neighboring clusters whose minimum key is below $\sigma_i(v)$, together with $C_i^\star(v)$ when it exists. For each reported cluster, insert its minimum-key edge and record the whole old vertex–cluster block for later deletion.
(4) After every vertex has been examined against the same snapshot, apply the new cluster assignments, apply all recorded block deletions, and then delete pairs internal to a new cluster.

All minimum, emptiness, and enumeration operations above are filtered searches: their candidate endpoints satisfy the current public survival predicate and the relevant public cluster predicate.

For example, suppose the old clusters adjacent to v have minimum keys

$$\kappa_i(v, C_1) < \kappa_i(v, C_2) < \kappa_i(v, C_3),$$

and among them only C_2 is sampled. Then $\sigma_i(v) = \kappa_i(v, C_2)$, the vertex reports C_1 and C_2, inserts one minimum edge to each, records both corresponding blocks for deletion, and joins C_2. The block to C_3 survives this report unless it is removed by another recorded deletion or becomes internal after the simultaneous reclustering. If v already lies in a sampled cluster, it reports nothing.

Every quantity above refers to the one snapshot $(E_i, \mathcal{C}_i, \mathcal{S}_i)$. The filtered-neighbor oracle first finds $\sigma_i(v)$, then enumerates precisely $\mathcal{R}_i(v)$, removing an entire old cluster from the public candidate set after it is reported. Delayed application of the block records therefore implements the displayed phase exactly: one vertex's queries cannot change a later vertex's minimum, report set, or cluster assignment.

Lemma 4.2 (Query-efficient weighted spanner)**.** *For $k = \lceil \ln n \rceil$, a capped and restarted call returns a $(2k-1)$-spanner containing $O(n \log n)$ pairs using $n \log^{O(1)}(n/\zeta')$ outgoing-cut queries, except with probability ζ'. Its query count is deterministically capped.*

Proof. We verify stretch directly for these simultaneous phases. Give pair e length $\ell_e = 1/w_e$, and let $\mathcal{H}_i$ be the pairs inserted before phase i. Inductively maintain the following invariant: for every $e = uv \in E_i$ and each endpoint $x \in \{u, v\}$, if x belongs to $C \in \mathcal{C}_i$, then $\mathcal{H}_i$ contains an x–center(C) path of at most $i - 1$ edges, every one of length at most ℓ_e. It is trivial for the singleton clustering.

Suppose v joins $C_i^\star(v)$ through $f = e_i(v, C_i^\star(v))$, and let $e = vu$ survive all recorded block deletions, temporarily before the same-new-cluster deletion. The old cluster containing u was not reported by v, so

$$\mathrm{key}(f) < \kappa_i(v, C_u) \leq \mathrm{key}(e), \qquad \text{hence} \qquad \ell_f \leq \ell_e.$$

The old invariant for the endpoint of f in $C_i^\star(v)$, followed by f, gives the required path for v with at most i edges. An endpoint already in a sampled cluster keeps its old path. A vertex that becomes inactive reports every neighboring cluster, so no pair incident to it survives. This proves the claimed center-path property for every pair surviving the block deletions; after pairs with equal new labels are removed, it is precisely the invariant for phase $i + 1$.

Now let $e = vu \in E_i(v, C)$ be deleted through a reported block, and put $f = e_i(v, C)$. Since $\ell_f \leq \ell_e$, the invariant gives paths from u and from the endpoint of f in C to the center of C, each using at most $i - 1$ edges of length at most ℓ_e. Together with f they form a v–u path of length at most $(2i - 1)\ell_e$. If e is instead deleted because its endpoints receive the same new cluster label, the two center paths use at most i edges each, for total length at most $2i\ell_e$; this can happen only when $i < k$. In the final phase every remaining neighboring cluster is reported, so every remaining pair obtains a path of length at most $(2k - 1)\ell_e$. The output is therefore a $(2k - 1)$-spanner.

For size and queries, condition on the frozen snapshot and order the old clusters neighboring a fixed vertex by $\kappa_i(v, C)$. If the vertex does not itself lie in a sampled cluster, then in a sampling phase $|\mathcal{R}_i(v)|$ is the position of the first sampled neighboring cluster, truncated at the number of neighbors; if its own cluster is sampled, it reports zero clusters. Consequently

$$\mathbb{P}[|\mathcal{R}_i(v)| \geq j \mid E_i, \mathcal{C}_i] \leq (1 - p)^{j-1}, \qquad \mathbb{E}[|\mathcal{R}_i(v)| \mid E_i, \mathcal{C}_i] \leq p^{-1}.$$

Moreover, $\mathbb{E}[|\mathcal{C}_{i+1}| \mid \mathcal{C}_i] = p|\mathcal{C}_i|$, so $\mathbb{E}|\mathcal{C}_k| = np^{k-1} = n^{1/k}$. The final phase reports at most $n|\mathcal{C}_k|$ incidences. If R is the total number of reports, then

$$\mathbb{E}R \leq (k - 1)np^{-1} + n\mathbb{E}|\mathcal{C}_k| \leq kn^{1+1/k}.$$

At most one pair is inserted per report. Abort once $R > 2kn^{1+1/k}$; Markov's inequality gives completion probability at least $1/2$. A trial performs one initial search per vertex per phase and $O(1)$ binary searches per report, hence $O((kn + R)\log n)$ filtered-neighborhood tests. After $\lceil \log_2(1/\zeta') \rceil$ independent capped trials, failure is at most ζ', while the query count is deterministic. With $k = \lceil \ln n \rceil$, both the cap and output size are $O(n \log n)$. □

Multiplicity is ignored while building a spanner: all copies of a pair have the same base length, and a selected pair is kept as one block. Its multiplicity is learned by a singleton instance of Corollary 2.3.

4.2 Bundle leverage

Fix one sparsification round. Every current pair has conductance $\omega_e^{(r)} = c_e p_{\mathrm{samp}}^{-r} w_e$, where $p_{\mathrm{samp}} = 1/4$ and r is the number of previous pair-sampling rounds. Construct t spanners successively, deleting every selected unordered pair before the next construction, and let B be their union.

Lemma 4.3 (Grouped bundle leverage)**.** *Let each of the t spanners have stretch α. If e remains outside B, then its leverage in the full current graph G_r satisfies*

$$\omega_e^{(r)} R_e[G_r] \le \frac{C\alpha}{t}.$$

Proof. For every spanner S_i, the surviving pair $e = uv$ has a u–v path P_i satisfying

$$\sum_{f \in P_i} \frac{1}{w_f} \le \frac{\alpha}{w_e}.$$

The paths are pair-edge-disjoint because selected pairs are deleted between successive spanners. Send $1/t$ units of flow on every P_i. The actual resistance of f is $(c_f p_{\mathrm{samp}}^{-r} w_f)^{-1} \le p_{\mathrm{samp}}^r / w_f$, so the flow energy is at most

$$\frac{1}{t^2} \sum_{i=1}^{t} \sum_{f \in P_i} \frac{p_{\mathrm{samp}}^r}{w_f} \le \frac{p_{\mathrm{samp}}^r \alpha}{t w_e}.$$

Thomson's principle bounds $R_e[G_r]$ by this energy. Multiplication by $\omega_e^{(r)} = c_e p_{\mathrm{samp}}^{-r} w_e$ and $c_e \le C$ proves the claim. Internal vertices need not be disjoint. $\square$

4.3 One sampling round

Keep B deterministically. Independently retain every remaining unordered pair with probability $p_{\mathrm{samp}} = 1/4$, retaining all of its copies as one block, and multiply a survivor's conductance by $1/p_{\mathrm{samp}} = 4$. Let L be the Laplacian of the *full* graph at the start of the round, including B, and define on the range of L

$$A_e = L^{\dagger/2} L_e L^{\dagger/2}, \qquad L_e = \omega_e^{(r)} a_e a_e^{\top},$$

where $a_e = e_u - e_v$ is the canonical incidence vector. Lemma 4.3 gives $\|A_e\| \le \lambda$, where $\lambda = C\alpha/t$. Moreover, the sum of A_e over the sampled complement is at most the identity projector.

If ξ_e is the sampling indicator, the normalized zero-mean error is

$$Z_e = (\xi_e / p_{\mathrm{samp}} - 1) A_e.$$

For $p_{\mathrm{samp}} = 1/4$,

$$\|Z_e\| \le 3\lambda, \qquad \mathbb{E}[(\xi_e/p_{\mathrm{samp}} - 1)^2] = 3, \qquad A_e^2 \preceq \lambda A_e.$$

Thus $\left\|\sum_e \mathbb{E} Z_e^2\right\| \le 3\lambda$. Self-adjoint matrix Bernstein [10] yields the following.

Lemma 4.4 (One-round concentration)**.** *For $0 < \delta \le 1$,*

$$\mathbb{P}\left[\left\|L^{\dagger/2}(\widehat{L} - L)L^{\dagger/2}\right\| > \delta\right] \le 2n \exp\left(-\frac{\delta^2 t}{8C\alpha}\right).$$

The use of the full L is important: the sampled complement alone may be singular in directions controlled by the fixed bundle.

4.4 Recursion and proof of Theorem 4.1

Set

$$R = \left\lceil \log_4 \frac{4n^2}{\zeta} \right\rceil, \quad \delta = \frac{\varepsilon}{3R}, \quad k = \lceil \ln n \rceil, \quad \alpha = 2k - 1,$$

and choose

$$t = \Theta\left(C\alpha\delta^{-2} \log \frac{nR}{\zeta}\right).$$

Recursive procedure. Starting with $(r, P_r) = (0, P_0)$, for each $r = 0, \dots, R-1$ perform the following steps.

(1) On the support $\mathcal{E}_r$, construct the t successive spanners and let their explicit union be B_r. When a pair first enters B_r, learn its multiplicity and store it with conductance $c_e p_{\text{samp}}^{-r} w_e$.

(2) Keep B_r in the output, draw the fresh public hash h_r, and define the sampled residual by (10). Recurse on $(r+1, P_{r+1})$, where every surviving pair has conductance $c_e p_{\text{samp}}^{-(r+1)} w_e$.

(3) Return the union of B_r and the graph returned by the recursive call.

At depth R, use one filtered-neighbor emptiness test per vertex to verify that $\mathcal{E}_R$ is empty. If it is not, declare the capped attempt unsuccessful; otherwise the union of the explicit bundles is the returned graph. The invariant above shows that every operation at level r sees exactly the Laplacian L_r, although its support has never been enumerated.

Approximation and termination. Conditioned on the transcript so far, the current hashes are fresh, so Lemma 4.4 applies. If the recursive result approximates the sampled residual within factors a, b, adding the fixed PSD bundle and using the one-round event gives factors $a(1-\delta), b(1+\delta)$ for the preceding graph. Hence the final factors are $(1-\delta)^R, (1+\delta)^R$, contained in $[1-\varepsilon, 1+\varepsilon]$.

The depth-R check uses n filtered-neighbor tests. A fixed original pair remains only if it avoids all bundles and survives R fresh quarter-samplings, so a union bound gives probability at most $n^2 4^{-R} \le \zeta/4$. Capped spanner-call failures and the R Bernstein events consume the remaining failure budget.

Finally,

$$Rt = O\left(C\varepsilon^{-2} \log^5 \frac{n}{\varepsilon\zeta}\right).$$

There are Rt spanner calls, each costing $O(n \log^3(n/(\varepsilon\zeta)))$ queries after amplification. This proves the $O(C\varepsilon^{-2} n \log^8(n/(\varepsilon\zeta)))$ query bound. Multiplying the capped $O(n \log n)$ size per spanner by Rt gives the asserted output size.

5 An implicit rank-sensitive box-LP solver

We next show how to run the required LP algorithm without enumerating hidden rows. Write a box LP in row-variable form as

$$\min\{c_{\text{LP}}^\top x : A^\top x = b_{\text{LP}},\ \ell \le x \le u\}, \qquad A \in \mathbb{R}^{M \times r}. \tag{11}$$

For a supplied exactly feasible strict interior point x^0, satisfying $A^\top x^0 = b_{\text{LP}}$, set

$$U = \max\left\{ \left\|1/(u - x^0)\right\|_\infty, \left\|1/(x^0 - \ell)\right\|_\infty, \|u - \ell\|_\infty, \|c_{\text{LP}}\|_\infty \right\}.$$

5.1 IPM primer and interface dictionary

This section is written so that the Lee–Sidford method can be used as an interface theorem; familiarity with its interior-point analysis is not required. The following paragraph gives the geometric picture needed for the rest of the proof.

For each row-coordinate i, let ϕ_i be the standard finite-box barrier on (ℓ_i, u_i), and put

$$\Phi''(x) = \operatorname{diag}(\phi_i''(x_i)), \qquad W = \operatorname{diag}(w).$$

At path parameter $t > 0$ and positive path weights w, the method follows the weighted barrier objective

$$f_t(x, w) = tc_{\mathrm{LP}}^\top x + \sum_{i=1}^{M} w_i \phi_i(x_i) \quad \text{subject to} \quad A^\top x = b_{\mathrm{LP}}. \tag{12}$$

The exact minimizer of (12) is the *weighted center*. A projected Newton step moves the current point toward that center while remaining in the affine space. The weights are not fixed: Lee–Sidford continually chases a target vector built from Lewis weights of a barrier-scaled copy of A. This choice is what makes the number of path steps depend on the column rank r, rather than on the possibly quadratic number M of row variables.

The mixed norm used to measure both centering error and weight motion is

$$\|v\|_{w+\infty} := \|v\|_\infty + C_{\mathrm{norm}} \|v\|_w, \qquad \|v\|_w^2 = v^\top W v,$$

where $C_{\mathrm{norm}} = \log^{O(1)} M$. In this notation the centrality of a strictly interior feasible point is

$$\delta_t(x, w) = \min_{\nu \in \mathbb{R}^r} \left\| \frac{tc_{\mathrm{LP}} + W\phi'(x) - A\nu}{\sqrt{W\phi''(x)}} \right\|_{w+\infty}. \tag{13}$$

Division and square roots are coordinatewise. Thus $\delta_t(x, w) = 0$ exactly when the first-order stationarity equation holds for some equality multiplier ν; small centrality means that the current point is close enough to the weighted center for the local Newton and path-change estimates to apply. The stability result cited in Proposition 5.1 allows the normal solves and row reductions below to have inverse-polynomial error.

State-variable dictionary. The distinction between row space and constraint space is particularly important here: x, w, X have one coordinate per row, whereas ν, π and normal-system right-hand sides have only r coordinates.

Symbol	Role	Interpretation in this paper
M, r, A, b_{LP}	dimensions, constraint matrix, and LP right-hand side	$A \in \mathbb{R}^{M\times r}$; hidden pair rows and public rows are row variables, while $A^\top x = b_{\mathrm{LP}}$ gives the r equality constraints.
x, ℓ, u	primal point and box	x is kept strictly between its coordinatewise endpoints during the path computation and remains exactly feasible.
t, c_{LP}	path parameter and objective	t controls the objective term in (12); the execution has an artificial-cost phase and a true-cost phase.
w, W, X	path weights	$W = \mathrm{diag}(w)$ and $X = \log w$ is the log-weight used by the chasing update; both are represented by public arithmetic circuits.
ϕ', Φ''	barrier gradient and Hessian	$\Phi'' = \mathrm{diag}(\phi_i''(x_i))$ supplies the local coordinate scaling.
$A_x, g(x)$	Lewis target	$A_x = \Phi''(x)^{-1/2}A$ and $g(x)$ is the regularized Lewis-weight vector that the current w chases.
$D, A^\top DA$	primitive row scaling and normal matrix	D changes from call to call; its hidden part is a weighted graph Laplacian and its public-row part is added explicitly.
π, h	Newton multiplier and primal direction	solving for the r-vector π produces an M-coordinate direction h with $A^\top h = 0$.
$\delta_t(x, w)$	centrality	the mixed-norm stationarity residual in (13).

For the parameter choice used here, the regularized target weight is

$$g_i(x) = c_0 + w_{p_{\mathrm{LS}}}(A_x)_i, \qquad c_0 = 2r/M, \qquad p_{\mathrm{LS}} = 1 - 1/\log(4M). \tag{14}$$

Here $w_{p_{\mathrm{LS}}}(\mathcal{A})$ denotes the vector of $\ell_{p_{\mathrm{LS}}}$ Lewis weights of a matrix $\mathcal{A}$; it is a function of the matrix and should not be confused with the current path-weight vector w. The value $c_0 = 2r/M$ is a constant-factor retuning of the fastest parameter displayed by Lee–Sidford. Their weight-function analysis permits arbitrary nonnegative regularization [7, Theorem 29], with the constants below retuned accordingly. We write $k_{\mathrm{LS}} = O(\log M)$ for the corresponding regularized Lewis parameter in the movement bounds.

Lee–Sidford-to-oracle crosswalk. The source algorithm is invoked only through the following interfaces.

Lee–Sidford operation	Object requested	Realization below
coordinate update	one common formula for every row	a public pair-evaluable circuit, shared by duplicate copies
row reduction	sums, maxima, predicates, and mixed norms	bit-class aggregation and filtered-neighbor search
projected Newton step	solve $A^\top DA$	a spectral preconditioner followed by residual-certified Richardson iteration
Lewis update	simultaneous leverage estimates for a scaled A_x	a sparsifier inverse evaluated on every potential pair
weight chasing	support point of a mixed-norm ball	the one-dimensional KKT search in Subsection 5.4
inexact oracle interface	approximate row reductions and normal solves	dyadic interface values from Lemma 2.5, followed by the exact star repair in Section 6

One representative primitive batch. Starting from the public state (x, w, t), a centering-and-chasing batch can be read as the following data flow. First, common row formulas evaluate the analytic barrier derivatives and requested diagonal row scaling. Second, certified evaluation freezes sufficiently accurate dyadic interface values, and row reductions form the r-dimensional Newton right-hand side from those values. Third, the hidden normal matrix for the frozen, pair-evaluable diagonal is spectrally sparsified, and the resulting preconditioner is refined against the full frozen matrix to obtain the Newton multiplier. Fourth, when the Lewis routine requests leverage scores, a more accurate sparsifier inverse gives all of them simultaneously, after which the Lewis update is coordinatewise. Fifth, the mixed-norm support routine chooses the allowed log-weight motion. The primal direction is repaired on the explicit star, and the batch-end domain check is made. Not every batch invokes every primitive, but Proposition 5.1 asserts that this list is exhaustive. Section 6 verifies that these approximations meet the imported stable-solver interface.

Proposition 5.1 (Lee–Sidford execution interface)**.** *If A has full column rank and no zero row, the box-LP algorithm of Lee and Sidford uses*

$$O\left(\sqrt{r}\,\log^{13} M \,\log \frac{MU}{\varepsilon}\right)$$

primitive batches. The execution in their Algorithms 1–3, including `computeInitialWeight`, the artificial-cost phase, the cost switch, and final centering, can be written so that a batch uses only a polylogarithmic number of the following operations:

(a) apply one common coordinate formula to every row, using that row's bounds, objective coefficient, current coordinate and weight, together with public scalars and r-dimensional vectors;
(b) take sums, inner products, mixed norms, maxima, clipping decisions, and predicate-restricted reductions of row-coordinate data;
(c) solve a positive definite system $A^\top DA$ for a positive diagonal D;
(d) obtain simultaneous multiplicative leverage estimates for a diagonally scaled copy of A and perform the coordinatewise Lewis iteration; and
(e) return a support point of $\{v : \|v\|_\infty + C_{\mathrm{norm}}\|v\|_w \le R\}$.

The proof of the source algorithm uses the leverage routine only through its simultaneous multiplicative output guarantee. In particular, its internal random sketch is not part of the state-transition specification and may be replaced by any routine with that guarantee. With exact versions of (a)–(e), the execution returns an ε-suboptimal point in the box. The same conclusion, with constant-factor retuning of the invariant margins, holds when row reductions, mixed-ball support points, and normal solves meet the inverse-polynomial additive, multiplicative, and energy-norm accuracies requested by the execution.

Proof. Theorem 1 and Algorithm 3 of the arXiv version of Lee–Sidford [7] give the displayed rank-sensitive count and spell out the artificial-cost and true-cost phases. Their Theorem 40 reduces each path step to a centering call, a mixed-ball projection, and approximate Lewis weights. Appendix B implements the initial and subsequent Lewis weights by coordinatewise arithmetic and simultaneous leverage estimates; its Theorem 39 assumes only the resulting multiplicative estimates. Appendix D, culminating in Theorem 62, is precisely the support problem in (e) after the diagonal change of variables used below. The remaining lines of Algorithms 1–3 are the coordinate maps and reductions in (a)–(b). Grouping the polylogarithmically many occurrences in one path step and absorbing that factor into $\log^{13} M$ gives the stated batches. This enumeration includes initialization and the cost switch; no additional row-access operation is used by the source execution.

For the last sentence of the proposition, the journal version explicitly uses approximate Lewis weights and notes that inverse-polynomially accurate linear solves suffice. The detailed stability argument appears in Section 8, especially Theorem 28, of the earlier primal path-following paper [6]. That theorem is an interface statement in energy-relative solve error; decreasing all requested accuracies by a fixed polynomial factor only adds logarithmically many refinement iterations. □

Lemma 5.2 (Conditional Lewis computation)**.** *Fix a matrix $\mathcal{A}$ with full column rank. In the approximate- and initial-weight routines of Lee–Sidford, replace every randomized leverage-score call by arbitrary simultaneously valid multiplicative estimates of the requested accuracy. Conditional on those estimates, all subsequent updates are deterministic coordinatewise formulas, and the conclusions and iteration bounds of Theorems 39 and 58, using Lemma 60, of Lee–Sidford [7] continue to hold. In particular, the initial-weight routine may be started at $p = 2$ with simultaneous leverage estimates for $\mathcal{A}$. At the $\log^{-O(1)} M$ accuracies requested by the path algorithm, every accuracy-dependent factor is absorbed by the number of leverage calls charged in Proposition 5.1.*

Proof. For $p = 2$ the Lewis weights are exactly the leverage scores of $\mathcal{A}$. Algorithm 7 of the cited paper then changes p by its displayed homotopy schedule and invokes `computeApxWeight`. Once the simultaneous leverage inequalities requested by a call are supplied, Algorithms 6–7 use only coordinatewise powers, medians, clipping, and arithmetic. The proofs of Theorems 39 and 58 and Lemma 60 condition only on those simultaneous inequalities; the Johnson–Lindenstrauss signs are one implementation of the interface, not an additional hypothesis about the update. Induction over the homotopy calls therefore proves the statement. The final call in Algorithm 7 accepts the requested accuracy parameter. Its polynomial dependence on reciprocal accuracy is polylogarithmic for the accuracies used here and is already included in the $\log^{13} M$ batch bound. Higher fixed-point precision at an interface is not being conflated with a more accurate Lewis iteration. $\square$

Section 6 checks that our cut-query implementation meets this inexact interface while preserving affine feasibility exactly.

Theorem 5.3 (Implicit incidence box LP)**.** *Fix a constant C. Consider* (11)*, whose r columns are indexed by $[r]$, an accuracy $0 < \varepsilon < 1$, and a supplied point x^0 satisfying*

$$A^\top x^0 = b_{\mathrm{LP}}, \qquad \ell < x^0 < u.$$

The hidden support is the symmetrization of a directed multigraph accessed by its outgoing-cut oracle. Suppose:

(i) for each unordered pair $\{u, v\} \subseteq [r]$, the hidden rows consist of $c_{uv} \le C$ copies of $e_u - e_v$;
(ii) all copies on a pair have identical pair-evaluable interval bounds, objective coefficient, and initial coordinate;
(iii) the remaining $O(r)$ rows are explicit, all rows have Euclidean norm at most $\sqrt{2}$, and the explicit rows include a finite-box $+I_r$ block; and
(iv) every explicit matrix coefficient, right-hand side, box endpoint, objective coefficient, and coordinate of x^0 is dyadic, has $O(\log(MU/\varepsilon))$ total bit length, and has absolute value $(MU/\varepsilon)^{O(1)}$; the hidden incidence coefficients are integral; and
(v) $M, U, \varepsilon^{-1} = r^{O(1)}$.

Then, with probability at least $1 - r^{-\Omega(1)}$, the algorithm returns a pair-evaluable represented point x satisfying

$$\ell \le x \le u, \qquad A^\top x = b_{\mathrm{LP}}, \qquad c_{\mathrm{LP}}^\top x \le \mathrm{OPT} + \varepsilon,$$

using $r^{3/2} \log^{O(1)} r$ outgoing-cut queries. Every represented primal iterate satisfies its equality constraints exactly.

For the Phase-I instance, $b_{\mathrm{LP}} = d$, $M \le n^2 + 2n$, and $r = n$. Moreover $|b_v| \le n - 1$, hence $K_0 \le n$; the artificial start slack is at least $(K_0 + 1)/2$, the hidden start slack is constant, and the above width parameter satisfies $U \le 2n$. Thus the theorem costs $n^{3/2} \log^{O(1)} n$ queries.

Proof boundary and dependency roadmap. The proof of Theorem 5.3 begins here and ends at the close of Section 6. Two facts are imported from Lee–Sidford: Proposition 5.1 gives the exact-arithmetic batch count and the exhaustive primitive interface, while Lemma 5.2 records that their Lewis-weight analysis needs only simultaneous leverage guarantees. The remainder of the proof has four internal components:

(1) Lemma 5.4 collapses identical conceptual copies to one pair circuit without changing their linear contribution to a normal matrix.
(2) Lemma 5.5 inducts over the public transcript and reduces coordinate operations and row aggregates to cut queries.
(3) The next two subsections implement the non-coordinate primitives: normal solves and leverage scores via Theorem 4.1, and mixed-norm support via a scalar KKT search.
(4) Section 6 checks the imported inexact interface and uses the explicit star rows to preserve affine feasibility exactly.

Each batch then costs $r \log^{O(1)} r$ queries. Multiplying by the $\sqrt{r} \log^{O(1)} r$ batch count proves the claimed query bound; the same induction shows that no operation tests an unqueried adjacency bit.

5.2 Duplicate-copy symmetry

A conceptual copy is a separate row coordinate, not a row whose incidence vector has been multiplied by the pair multiplicity. In general, if the rows of A are $a_i^\top$, then $A^\top DA = \sum_i D_i a_i a_i^\top$. Hence c_{uv} identical rows, each with diagonal value D_{uv}, contribute $c_{uv}D_{uv}a_{uv}a_{uv}^\top$. Replacing them by the single scaled row $c_{uv}a_{uv}^\top$ would incorrectly square the multiplicity. The following lemma justifies retaining only one represented formula while still using the former, linear contribution.

Lemma 5.4 (Block symmetry)**.** *If all hidden copies of an unordered pair start identically, every operation of the compiled box-LP algorithm preserves their equality. Consequently, every hidden normal matrix is*

$$L_D = \sum_{u<v} c_{uv}D_{uv}a_{uv}a_{uv}^\top, \tag{15}$$

not a matrix with coefficient $c_{uv}^2 D_{uv}$.

Proof. Equal copies have the same row, box, objective coefficient, coordinate, barrier derivatives, and weight. A Newton coordinate is a common coordinatewise expression plus the endpoint potential difference $a_{uv}^\top \pi$. Its leverage is $D_{uv}a_{uv}^\top (A^\top DA)^{-1}a_{uv}$, also common. Lewis updates use coordinatewise powers, affine combinations, medians, and clipping. The mixed-ball optimizer below assigns equal outputs to equal data. Induction therefore proves equality throughout. Summing the c_{uv} identical rank-one contributions gives (15). □

Independent JL signs cannot be assigned to indistinguishable copies through the cut oracle. We avoid them by obtaining deterministic simultaneous leverage estimates from a sparsifier inverse.

Lemma 5.5 (Transcript simulation invariant)**.** *At the beginning and end of every batch in Proposition 5.1, each hidden row coordinate is represented by one public pair-evaluable circuit, common to all copies of its unordered pair. Every operation (a)–(e) can be simulated with $r \log^{O(1)} r$ outgoing-cut queries, and its represented outputs again satisfy this invariant.*

Proof. Initially the bounds, objective, and primal coordinate are public pair circuits by assumption. There is no circular appeal to an already computed initial weight. At $p = 2$, the Lewis weights of $\mathcal{A}_0 = \Phi''(x^0)^{-1/2}A$ are its leverage scores. One simultaneous-leverage call therefore produces a pair-evaluable approximation to the regularized $p = 2$ target. Lemma 5.2 then inducts through the $p = 2$-to-p_{LS} homotopy, so its final log-weight is also a public pair circuit.

This first leverage call is licensed directly from the supplied start, whose slack is at least $1/U$, so $\Phi''(x^0)$ has polynomial upper and lower bounds, while the explicit $+I_r$ block gives a polynomial lower spectral bound for its normal matrix. To make the induction quantitative, let $w_p(\mathcal{A}_0)$ be the exact p-Lewis weights for $p \in [p_{\mathrm{LS}}, 2]$, let j minimize their coordinates, put $a = 1 - 2/p \le 0$, and write $\alpha_i^\top$ for a row of $\mathcal{A}_0$. The Lewis fixed-point identity and

$$\mathcal{A}_0^\top W_p^a \mathcal{A}_0 \preceq (w_p)_j^a \mathcal{A}_0^\top \mathcal{A}_0$$

give

$$(w_p)_j = (w_p)_j^a \alpha_j^\top (\mathcal{A}_0^\top W_p^a \mathcal{A}_0)^{-1}\alpha_j \ge \alpha_j^\top (\mathcal{A}_0^\top \mathcal{A}_0)^{-1}\alpha_j \ge (MU)^{-O(1)}.$$

The last inequality uses the polynomial spectral upper bound, the absence of zero rows, and the dyadic input-bit hypothesis; a nonzero explicit coefficient and every scaled hidden incidence coefficient have inverse-polynomial magnitude. Every exact Lewis weight is at most one. During the initial-weight homotopy $x = x^0$ is fixed, and Lemma 60 together with the median clipping in Algorithm 7 therefore keeps every intermediate approximate scaling between inverse polynomials and polynomials. Each successive scaled normal matrix is thus polynomially conditioned, so the same direct argument licenses every call counted by Proposition 5.1.

A common coordinate formula preserves the pair-circuit property. Sums, inner products, incidence products, and predicate-restricted sums use Lemmas 2.4 and 2.5; mixed norms add one square root after such sums. Maxima, ℓ_∞ norms, emptiness tests, and clipping predicates use Lemma 2.6. When a certified interval straddles a threshold, either branch changes the corresponding coordinate by at most the interval width. We choose that width below the inverse-polynomial tolerance in Proposition 5.1.

Normal solves and simultaneous leverage estimates are implemented in the next subsection, and the mixed-ball support point in the following one. Their outputs consist of public r-vectors, public scalars, or a common pair-evaluable row formula, so they also preserve the invariant. All random hashes used by the spectral calls are public transcript data.

For completeness, the artificial cost is kept as the exact elementary circuit

$$(c_{\mathrm{art}})_i = -\exp(X_i^0)\phi_i'(x_i^0).$$

Consequently x^0 is the exact center of the analytic artificial objective; only evaluations of this fixed circuit are rounded at aggregate interfaces. The cost switch replaces this circuit by the supplied true-cost circuit and does not inspect hidden support. This covers the initialization, both path phases, and the final centering listed in Proposition 5.1. □

5.3 Normal systems and leverage scores

Lemma 5.6 (Certified dyadic inverse interface)**.** *Let K be an explicit dyadic positive definite $r \times r$ matrix with*

$$K_- I \preceq K \preceq K_+ I,$$

where K_+, K_-^{-1}, and ϑ^{-1} are polynomially bounded in MU/ε. Offline certified arithmetic can output a dyadic symmetric Q with $O(\log(MU/\varepsilon))$ bits per entry such that

$$(1-\vartheta)K^{-1} \preceq Q \preceq (1+\vartheta)K^{-1}.$$

Proof. Compute certified intervals for the entries of K^{-1} until their widths are at most $\delta = \vartheta/(4rK_+)$, and take the symmetric dyadic midpoints. Certified Gaussian elimination or interval Cholesky terminates because $K \succeq K_- I$. For $E = Q - K^{-1}$, the entrywise bound gives $\|E\|_2 \le r\delta \le \vartheta/(2K_+)$ after harmless constant rounding. Since $K^{-1} \succeq K_+^{-1} I$, this implies $-\vartheta K^{-1} \preceq E \preceq \vartheta K^{-1}$. The magnitudes of inverse entries are at most K_-^{-1}, while the requested absolute accuracy is inverse polynomial, proving the bit bound. □

How a normal matrix enters Theorem 4.1. This is the precise bridge between the IPM and the graph primitive. During one normal-system or leverage call, the analytic positive row scaling D is held fixed. Certified interval evaluation chooses a positive dyadic value $\widehat{D}_{uv}$ for every potential hidden pair, using the same value for all conceptual copies of that pair. The resulting hidden contribution is

$$\widehat{L}_{\mathrm{hid}} = \sum_{u<v} c_{uv}\widehat{D}_{uv} a_{uv} a_{uv}^\top.$$

Thus Theorem 4.1 is instantiated with the exact pair-evaluable edge weight

$$w_{\{u,v\}}^{\mathrm{spec}} = \widehat{D}_{uv}.$$

The public dyadic circuit can compare and output these weights exactly, as required by that theorem; the unknown multiplicity c_{uv} remains part of the hidden graph and is learned only through cut queries. In particular, the spectral routine never receives an unevaluated exponential or barrier derivative. All explicit-row contributions are instead formed exactly and added to the returned hidden sparsifier. Consequently the graph theorem is applied to precisely the hidden summand of the frozen dyadic normal matrix, not directly to the analytic matrix and not to a graph in which duplicate rows have been combined by scaling their incidence vectors.

For an analytic system $L = A^\top DA$, certified evaluation first forms a positive dyadic diagonal $\widehat{D}$ satisfying

$$(1 - \beta_N)D \preceq \widehat{D} \preceq (1 + \beta_N)D, \qquad \beta_N = (MU/\varepsilon)^{-C_N}. \tag{16}$$

Corollary 68 and Lemma 69 of Lee–Sidford [7] give a polynomial envelope for every normal system used by their execution. A sufficiently large absolute C_N therefore makes this possible with logarithmic precision. These dyadic values are frozen for the entire spectral call and all subsequent refinement steps belonging to that normal solve. Put $\widehat{L} = A^\top \widehat{D} A$. Apply Theorem 4.1 with constant accuracy to the hidden contribution of $\widehat{L}$ and add every explicit-row contribution exactly. PSD addition gives an explicit preconditioner P with

$$\tfrac{1}{2}\widehat{L} \preceq P \preceq 2\widehat{L}.$$

A full product by $\widehat{L}$ is computed exactly for the dyadic interface values by Lemma 2.4. Preconditioned Richardson therefore converges geometrically for the frozen dyadic system. Here and below an explicit positive definite preconditioner is not inverted as an exact rational matrix. Lemma 5.6 instead produces a dyadic matrix Q_P with

$$(1 - \beta_P)P^{-1} \preceq Q_P \preceq (1 + \beta_P)P^{-1}, \qquad \beta_P \le 1/20. \tag{17}$$

The polynomial spectral envelope supplies the hypotheses of the lemma. Preconditioned Richardson contracts the energy error by a constant factor, so $O(\log(MU/\varepsilon))$ iterations give the inverse-polynomial relative accuracy requested in Proposition 5.1. Each iteration multiplies by the full frozen matrix using Lemma 2.5; hence one such solve uses $r \log^{O(1)} r$ queries. Section 6 compares the resulting direction with the analytic one and applies the exact star repair.

For leverage scores, request a $(1 \pm \theta)$ sparsifier L_S of the full dyadic normal matrix $\widehat{L}$, with $\theta = 1/\operatorname{polylog} r$. Lemma 5.6 produces a dyadic symmetric matrix Q_S satisfying

$$(1 - \theta/4)L_S^{-1} \preceq Q_S \preceq (1 + \theta/4)L_S^{-1}. \tag{18}$$

For a hidden copy on $e = uv$, define

$$\widehat{\sigma}_e = \widehat{D}_e\big((Q_S)_{uu} + (Q_S)_{vv} - 2(Q_S)_{uv}\big). \tag{19}$$

Inverse monotonicity and (18) give

$$(1 - 3\theta)\widehat{D}_e a_e^\top \widehat{L}^{-1} a_e \le \widehat{\sigma}_e \le (1 + 3\theta)\widehat{D}_e a_e^\top \widehat{L}^{-1} a_e$$

for θ below an absolute constant. Together with (16), this is a $1 \pm O(\theta + \beta_N)$ estimate of the analytic leverage score. Requesting half the tolerance from each source gives the tolerance required by Lee–Sidford. The conclusion holds simultaneously for every actual or potential pair. Formula (19) is pair-evaluable and common across duplicate copies, so the approximate Lewis updates remain coordinatewise pair circuits.

5.4 The mixed-norm support primitive

For a symmetric compact convex set $\mathcal{K}$, write

$$h_{\mathcal{K}}(p) := \max_{v\in\mathcal{K}} \langle p, v\rangle\,, \qquad \|p\|_{\mathcal{K}} := h_{\mathcal{K}}(p).$$

The second notation is only shorthand for the support function. The chasing update requires a support point of

$$\mathcal{U}_R = \{u : \|u\|_\infty + C_{\text{norm}}\,\|u\|_w \le R\}\,, \qquad \|u\|_w^2 = u^\top W u.$$

This primitive is the decision step in Lewis-weight chasing. Given the current gradient-like objective p, it chooses the admissible log-weight motion with nearly maximum predicted progress. The ℓ_∞ term prevents any one row weight from moving too far, while the weighted ℓ_2 term controls aggregate motion. Because the coordinates include hidden rows, the optimizer cannot be constructed by scanning the support.

Algorithmic view. The computation has four steps. First, diagonally rescale the ball to (20). Second, introduce a scalar $\tau \in [0,1]$ that splits the unit budget between the Euclidean radius $1-\tau$ and the coordinate caps $\tau\kappa_i$. For a fixed τ, the KKT conditions reduce the optimizer to one scalar multiplier λ, found by bisection from predicate-restricted aggregates. Third, maximize the resulting one-dimensional concave value function $G(\tau)$ by certified interval search. Finally, undo the diagonal scaling and, if necessary, rescale the dyadic output conservatively to preserve feasibility. The formulas below establish the relative support guarantee and the query bound for these four steps. The letter τ here is only this local budget split and is unrelated to the central-path parameter t.

If $R = 0$, the routine returns zero exactly. Assume $R > 0$. For an objective vector p, set

$$x = \frac{C_{\text{norm}}}{R} W^{1/2} u, \qquad \kappa_i = C_{\text{norm}}\sqrt{w_i}, \qquad a = \frac{R}{C_{\text{norm}}} W^{-1/2} p.$$

Then $\langle p, u\rangle = \langle a, x\rangle$ and the normalized problem is

$$\max a^\top x \quad \text{s.t.} \quad \|x\|_2 + \left\|\kappa^{-1}x\right\|_\infty \le 1. \tag{20}$$

For $\tau \in [0,1]$, define

$$G(\tau) = \max\left\{a^\top x : \|x\|_2 \le 1-\tau,\ |x_i| \le \tau\kappa_i \text{ for every } i\right\}. \tag{21}$$

The endpoints have the unique feasible optimizer $x = 0$. If $0 < \tau < 1$ and

$$\sum_{i:a_i\neq 0} \tau^2\kappa_i^2 \le (1-\tau)^2,$$

the optimizer is $x_i = \operatorname{sign}(a_i)\tau\kappa_i$ for $a_i \neq 0$ and $x_i = 0$ otherwise. In the remaining case, the KKT conditions give

$$x_i = \operatorname{sign}(a_i)\min\left\{\tau\kappa_i, \frac{|a_i|}{2\lambda}\right\}, \tag{22}$$

where the unique $\lambda > 0$ satisfies

$$\sum_i \min\left\{\tau^2\kappa_i^2, \frac{a_i^2}{4\lambda^2}\right\} = (1-\tau)^2. \tag{23}$$

Every predicate and sum in (23), and the resulting objective value, is a predicate-restricted pair aggregate.

We record the relative guarantee needed later. The function G is concave: interpolating feasible solutions at σ, τ gives a feasible solution at the same interpolation of the parameters. It is also Lipschitz, with

$$|G(\tau) - G(\sigma)| \le L_a |\tau - \sigma|, \qquad L_a = \max\left\{ \|a\|_2 , \sum_i |a_i| \kappa_i \right\}. \tag{24}$$

Indeed, for $\sigma < \tau$, scale an optimizer at σ by $(1-\tau)/(1-\sigma)$ to obtain a feasible point at τ, losing at most $\|a\|_2 (\tau - \sigma)$; scaling an optimizer at τ by σ/τ gives a feasible point at σ, losing at most $(\tau - \sigma) \sum_i |a_i| \kappa_i$.

If $a = 0$, return zero. Otherwise the one-coordinate point

$$x_i = \operatorname{sign}(a_i) \frac{\kappa_i}{1 + \kappa_i}, \qquad x_j = 0 \quad (j \neq i),$$

is feasible for every i with $a_i \neq 0$. Hence, if $H = \max_\tau G(\tau) = h_{\mathcal{U}_R}(p)$, then

$$H \ge \nu_0 := \max_{i : a_i \neq 0} \frac{|a_i| \kappa_i}{1 + \kappa_i}. \tag{25}$$

At an implemented call the objective is frozen on an inverse-polynomial dyadic mesh. If its support value is below the requested additive tolerance, the zero vector is already an admissible answer. Otherwise a nonzero frozen coordinate has inverse-polynomial magnitude. The source weight bounds (Lee–Sidford [7, Lemma 67]) make $W^{\pm 1/2}$ polynomially bounded, and every positive radius requested by their path schedule is inverse polynomial in MU/ε. Thus $\nu_0 \ge (MU/\varepsilon)^{-O(1)}$ in the only case where a relative search is needed.

Bisection on (23), followed if needed by a conservative common rescaling, evaluates $G(\tau)$ and returns a feasible sign-aligned point to any additive accuracy ζ. A noise-robust one-dimensional concave search then maximizes G: compare certified value intervals at two interior trisection points, discard a side when the intervals separate, and otherwise retain the better queried point. Concavity and (24) show that $O(\log(L_a/\zeta))$ evaluations achieve additive error $O(\zeta)$. Taking $\zeta \le \gamma \nu_0 / 32$ and accounting for both searches yields a feasible, sign-aligned $u \in \mathcal{U}_R$ satisfying

$$\langle p, u \rangle \ge (1 - \gamma) h_{\mathcal{U}_R}(p). \tag{26}$$

For inverse-polynomial γ, all scalar searches use $O(\log(MU/\varepsilon))$ aggregate evaluations, and the query cost is $r \log^{O(1)} r$. All sums involving the analytic weights are evaluated by certified dyadic intervals. The routine rescales using an upper interval for the mixed norm, so the returned point is feasible for the *analytic* ball. Lemma 2.5 and the Lipschitz bound (24) show that width β in each interface changes the support value by at most $\operatorname{poly}(MU/\varepsilon)\beta$. Choosing β below the source tolerance includes this loss in ζ.

The list in Proposition 5.1 is exhaustive. Lemma 5.5 handles its coordinate operations and reductions, while the two preceding subsections handle its remaining primitives. Consequently no step branches on an unqueried adjacency bit. This completes the cut-query simulation part of the proof of Theorem 5.3. It remains only to check the stable-solver accuracy interface and exact affine feasibility; that is the purpose of the next section.

6 The solver interface and exact feasibility

The numerical layer is short once the right boundary is drawn. We use the stable Lee–Sidford execution as an imported solver theorem and prove only that our cut-query primitives meet its accuracy requirements. This is the one place where the explicit $+I_r$ block is used algorithmically.

At a current point let

$$H = W\Phi''(x), \qquad D = H^{-1}, \qquad L = A^\top DA, \qquad q = tc_{\mathrm{LP}} + W\phi'(x).$$

The exact projected Newton direction has the form

$$h = -D(q - A\pi), \qquad L\pi = A^\top Dq, \qquad A^\top h = 0. \tag{27}$$

Let $R_\star \in \mathbb{R}^{M\times r}$ embed a vector on the explicit $+I_r$ rows. Then

$$A^\top R_\star = I_r. \tag{28}$$

Lemma 6.1 (Cut-query solver interface)**.** *Under the hypotheses of Theorem 5.3, there is an absolute constant C_{stab} for which the following implementation is valid. Evaluate every row reduction and mixed-norm support call to additive or multiplicative error at most*

$$\alpha = (MU/\varepsilon)^{-C_{\mathrm{stab}}},$$

and solve every normal system to relative α error in its energy norm. After forming the resulting raw primal direction $\widetilde{h}$, replace it by

$$h_{\mathrm{rep}} = \widetilde{h} - R_\star(A^\top \widetilde{h}). \tag{29}$$

Then the execution satisfies the inexact interface of Proposition 5.1, every represented primal iterate obeys $A^\top x = b_{\mathrm{LP}}$ exactly, and the final point lies in the box and has objective at most $\mathrm{OPT} + \varepsilon$.

Each primitive batch uses $r \log^{O(1)} r$ cut queries. All dyadic values passed to bit-class aggregation need only $\log^{O(1)}(MU/\varepsilon)$ bits.

Proof. We first record what is imported. The inexact path-following analysis of Lee–Sidford [6, Section 8, especially Theorem 28] shows that inverse-polynomial energy-relative errors in the linear solves can be absorbed by constant-factor slack in the centering invariants. The later rank-sensitive formulation uses approximate Lewis weights explicitly [7, Theorems 39–40]. Its Corollary 68 bounds every barrier Hessian encountered by a polynomial in MU/ε, and Lemma 69 gives the corresponding polynomial relative conditioning of the normal systems. Lemma 67 bounds path weights between inverse polynomials and constants; the finite-box barrier formulas and the width parameter U give the complementary lower Hessian bounds. Consequently there is one fixed polynomial accuracy that suffices simultaneously for all the row reductions, support computations, leverage estimates, and normal solves listed in Proposition 5.1. We choose C_{stab} larger than its degree.

The coordinate reductions meet this accuracy directly. Certified evaluation freezes a row circuit to dyadic width β. By Lemma 2.5, every scalar aggregate changes by at most $M\beta$ and every normal matrix changes in operator norm by at most $2M\beta$. The polynomial conditioning just cited lets us choose $\beta = (MU/\varepsilon)^{-C}$, for one sufficiently large constant C, so that the induced error is at most α. Maxima and predicates are handled by Lemma 2.6; an interval that straddles a threshold may take either branch, since the two returned coordinate values then differ by at most the same interface tolerance. All remaining state maps are compositions of arithmetic, barrier derivatives, powers, medians, and clipping. On the source polynomial envelope their denominators are bounded away from zero by an inverse polynomial; the smooth maps are therefore polynomially Lipschitz, while median and clipping are nonexpansive. Hence their accumulated error over one batch is $\mathrm{poly}(MU/\varepsilon)\beta$, which is again at most α after increasing C. The support construction in Subsection 5.4 and the leverage construction in the preceding subsection already give the additive and multiplicative guarantees required by the proposition.

For a normal solve, freeze $\widehat{D}$ and its right-hand side as in (16), and put

$$\widehat{L} = A^\top \widehat{D} A, \qquad \widehat{b} = A^\top \widehat{D} q.$$

The sparsifier gives $\frac{1}{2}\widehat{L} \preceq P \preceq 2\widehat{L}$. Preconditioned Richardson iteration therefore contracts the $\widehat{L}$-energy error by a constant factor. After $O(\log(1/\alpha)) = O(\log(MU/\varepsilon))$ iterations it returns $\widetilde{\pi}$ satisfying

$$\|\widetilde{\pi} - \widehat{L}^{-1}\widehat{b}\|_{\widehat{L}} \le \alpha \|\widehat{L}^{-1}\widehat{b}\|_{\widehat{L}};$$

when $\widehat{b} = 0$ we return zero. Multiplication by the full $\widehat{L}$, rather than by the sparsifier, makes this a solve of the desired frozen system. The perturbation (16) and polynomial conditioning transfer the same bound, up to a polynomial factor already absorbed in C_{stab}, to the analytic system in (27).

It remains to explain why the exact repair does not spoil that bound. Write $e = \widetilde{h} - h$, where h is the analytic direction requested by the stable execution. Since $A^\top h = 0$,

$$h_{\mathrm{rep}} - h = e - R_\star A^\top e.$$

In the local norm this gives

$$\begin{aligned} \|H^{1/2}(h_{\mathrm{rep}} - h)\|_2 &\le \bigl(1 + \|H^{1/2}R_\star\|_2 \|A^\top H^{-1/2}\|_2\bigr)\|H^{1/2}e\|_2 \\ &\le \mathrm{poly}(MU/\varepsilon)\, \|H^{1/2}e\|_2. \end{aligned}$$

The first inequality is the triangle inequality followed by submultiplicativity. The second follows from the source Hessian bounds and from $\|A^\top H^{-1/2}\|_2^2 = \lambda_{\max}(A^\top H^{-1}A)$. Thus requesting the raw direction to a correspondingly smaller fixed inverse-polynomial tolerance makes the repaired direction a valid inexact direction. The raw coordinate circuit is formed from the frozen dyadic row values and the dyadic approximate potential, so Lemma 2.4 computes $A^\top \widetilde{h}$ exactly with $r \log^{O(1)} r$ queries. Therefore (28) gives the exact identity

$$A^\top h_{\mathrm{rep}} = A^\top \widetilde{h} - A^\top R_\star A^\top \widetilde{h} = 0.$$

Starting from the supplied exactly feasible point, induction keeps every iterate exactly feasible. The repair touches only explicit star rows, so it does not disturb duplicate-copy symmetry or pair-evaluability.

All arithmetic between queries is free in the query model. The only finite representations seen by the oracle reductions are the dyadic interfaces, and their accuracy is inverse polynomial, so their bit length is polylogarithmic in MU/ε. Richardson adds only another logarithmic factor. If a randomized spectral call fails, or an actual row fails its domain check, the capped execution rejects; on the simultaneous success event neither occurs. The imported stability conclusion then gives strict interiority and the ε objective guarantee, proving the lemma. $\square$

End of the proof of Theorem 5.3. Proposition 5.1 uses $\sqrt{r} \log^{O(1)} r$ primitive batches. Lemmas 5.5 and 6.1 implement each batch in $r \log^{O(1)} r$ cut queries, while preserving exact affine feasibility. Multiplying the two bounds gives $r^{3/2} \log^{O(1)} r$. The conditional success probabilities of the spectral calls are combined in Section 7. This proves Theorem 5.3.

7 Completing the reachability algorithm

Apply Theorem 5.3 to (4) with output accuracy

$$\varepsilon_{\mathrm{LP}} = n^{-30}.$$

On the successful oracle event, the represented output is in the hidden box, exactly feasible, and satisfies

$$\sum_v (y_v^+ + y_v^-) \le n^{-30}. \tag{30}$$

Exact feasibility gives

$$B^\top z - (1-\eta)b = -(y^+ - y^-).$$

Because the artificial variables are nonnegative,

$$\left\| B^\top z - (1-\eta)b \right\|_1 \le \sum_v |y_v^+ - y_v^-| \le \sum_v (y_v^+ + y_v^-) \le n^{-30}. \tag{31}$$

Acceptance and the final algorithm. Run one capped execution of the implicit Phase-I solver. It returns either a pair-evaluable candidate z or Reject. If it rejects, return No. Otherwise test (6). An out-of-box actual pair is found by filtered search. For the residual test, use the public circuit clipped to $[-1,1]$ (which agrees with z on every actual pair after the box test) and Lemma 2.5 to enclose every coordinate of $B^\top z - (1-\eta)b$ within absolute error n^{-25}, and accept only if the resulting upper bound on its ℓ_1 norm is at most n^{-20}. This uses $n \log^{O(1)} n$ queries. Equation (31) shows that a successful solver output is accepted, since its true residual is at most n^{-30}.

For an accepted candidate, construct the threshold digraph J implicitly and run Lemma 3.4 from s. Return Yes exactly when that search discovers t. Every accepted candidate is deterministically safe: the conservative test certifies (6), so Lemmas 3.2 and 3.3 show that J has the same reachability relation as the input. Returning No is always correct on a no-instance; on a yes-instance it can be wrong only if the implicit solver's high-probability event fails.

Query and failure bound. Let N_{spec} be a deterministic upper bound on the number of spectral calls in the single capped execution; by Proposition 5.1, $N_{\mathrm{spec}} \le \sqrt{n}\log^{O(1)} n$. Give each potential call conditional failure probability at most $n^{-C_{\mathrm{fail}}}/N_{\mathrm{spec}}$, conditional on the entire preceding transcript. The adaptive union bound is then at most $n^{-C_{\mathrm{fail}}}$. Conditional on its complement, every primitive meets the hypotheses of Lemma 6.1, so the candidate is returned and accepted.

The LP execution uses $\sqrt{n}\log^{O(1)} n$ batches of $n\log^{O(1)} n$ queries. The acceptance test costs $n\log^{O(1)} n$, and the final search costs $O(n \log n)$. Fixed caps on every spanner attempt, Richardson refinement, and primitive batch give the same $n^{3/2}\log^{O(1)} n$ upper bound on every random tape. This proves Theorem 1.1.

8 Discussion

The main open question is whether directed reachability admits $n^{1+o(1)}$ outgoing-cut queries. It would also be interesting to obtain meaningful superlinear lower bounds, to make the pair-evaluable spectral routine efficient in computation as well as queries, and to remove the linear dependence on a nonconstant unordered-pair multiplicity cap.

References

[1] A. Anand, T. Saranurak, and Y. Wang. Deterministic edge connectivity and max flow using subquadratic cut queries. In *Proceedings of the 2025 Annual ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pages 124–142, 2025. https://doi.org/10.1137/1.9781611978322.4.

[2] S. Apers, Y. Efron, P. Gawrychowski, T. Lee, S. Mukhopadhyay, and D. Nanongkai. Cut query algorithms with star contraction. arXiv:2201.05674, 2022. https://arxiv.org/abs/2201.05674.

[3] S. Baswana and S. Sen. A simple and linear time randomized algorithm for computing sparse spanners in weighted graphs. *Random Structures & Algorithms*, 30(4):532–563, 2007. https://doi.org/10.1002/rsa.20130.

[4] Y. Jiang, D. Nanongkai, and P. Sawettamalya. Minimum s–t cuts with fewer cut queries. In *Proceedings of the 2026 Annual ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pages 258–296, 2026. https://doi.org/10.1137/1.9781611978971.12.

[5] I. Koutis. Simple parallel and distributed algorithms for spectral graph sparsification. In *Proceedings of the 26th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 61–66, 2014. https://doi.org/10.1145/2612669.2612676.

[6] Y. T. Lee and A. Sidford. Path Finding II: An $\widetilde{O}(m\sqrt{n})$ algorithm for the minimum cost flow problem. arXiv:1312.6713v2, 2015. https://arxiv.org/abs/1312.6713.

[7] Y. T. Lee and A. Sidford. Solving linear programs with $\widetilde{O}(\sqrt{\mathrm{rank}})$ linear system solves. arXiv:1910.08033v2, 2020. https://arxiv.org/abs/1910.08033.

[8] D. Nanongkai. Cut query for reachability: open problem. Open-problem page, first offered at ICALP 2024. https://sites.google.com/site/dannanongkai/open.

[9] A. Rubinstein, T. Schramm, and S. M. Weinberg. Computing exact minimum cuts without knowing the graph. In *9th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 94 of LIPIcs, pages 39:1–39:16, 2018. https://doi.org/10.4230/LIPIcs.ITCS.2018.39.

[10] J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012. https://doi.org/10.1007/s10208-011-9099-z.