On Solving Asymmetric Diagonally Dominant Linear Systems in Sublinear Time

Tsz Chiu Kwok
Shanghai University of Finance and Economics, China

Zhewei Wei
Renmin University of China, Beijing, China

Mingji Yang
Renmin University of China, Beijing, China

Abstract

We initiate a study of solving a row/column diagonally dominant (RDD/CDD) linear system $\mathbf{M}\boldsymbol{x} = \boldsymbol{b}$ in sublinear time, with the goal of estimating $\boldsymbol{t}^\top\boldsymbol{x}^*$ for a given vector $\boldsymbol{t} \in \mathbb{R}^n$ and a specific solution $\boldsymbol{x}^*$. This setting naturally generalizes the study of sublinear-time solvers for symmetric diagonally dominant (SDD) systems [Andoni-Krauthgamer-Pogrow, ITCS 2019] to the asymmetric case, which has remained underexplored despite extensive work on nearly-linear-time solvers for RDD/CDD systems.

Our first contributions are characterizations of the problem's mathematical structure. We express a solution $\boldsymbol{x}^*$ via a Neumann series, prove its convergence, and upper bound the truncation error on this series through a novel quantity of $\mathbf{M}$, termed the maximum p-norm gap. This quantity generalizes the spectral gap of symmetric matrices and captures how the structure of $\mathbf{M}$ governs the problem's computational difficulty.

For systems with bounded maximum p-norm gap, we develop a collection of algorithmic results for locally approximating $\boldsymbol{t}^\top\boldsymbol{x}^*$ under various scenarios and error measures. We derive these results by adapting the techniques of random-walk sampling, local push, and their bidirectional combination, which have proved powerful for special cases of solving RDD/CDD systems, particularly estimating PageRank and effective resistance on graphs. Our general framework yields deeper insights, extended results, and improved complexity bounds for these problems. Notably, our perspective provides a unified understanding of Forward Push and Backward Push, two fundamental approaches for estimating random-walk probabilities on graphs.

Our framework also inherits the hardness results for sublinear-time SDD solvers and local PageRank computation, establishing lower bounds on the maximum p-norm gap or the accuracy parameter. We hope that our work opens the door for further study into sublinear solvers, local graph algorithms, and directed spectral graph theory.

2012 ACM Subject Classification Mathematics of computing → Spectra of graphs; Theory of computation → Graph algorithms analysis; Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Spectral Graph Theory, Linear Systems, Sublinear Algorithms

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.89

Related Version *Full Version*: https://arxiv.org/abs/2509.13891 [28]

Funding This research was supported by National Natural Science Foundation of China (No. 92470128, No. U2241212).

Acknowledgements We thank the anonymous reviewers for their valuable comments. Mingji Yang wishes to thank Prof. Shang-Hua Teng for inspiring discussions and encouragement, and Guanyu Cui for helpful discussions on effective resistance.

1 Introduction

Solving systems of linear equations is one of the most fundamental problems in numerical linear algebra and theoretical computer science. In the classic version of this problem, we are given a matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ and a vector $\boldsymbol{b} \in \mathbb{R}^n$ in range($\mathbf{M}$), and the goal is to compute a solution vector $\boldsymbol{x} \in \mathbb{R}^n$ satisfying $\mathbf{M}\boldsymbol{x} = \boldsymbol{b}$. Beyond general-purpose solvers for arbitrary linear systems (e.g., using fast matrix multiplication or the conjugate gradient method), extensive research has focused on developing efficient solvers for special classes of systems.

In particular, for the important classes of Laplacian and symmetric diagonally dominant (SDD) systems, the breakthrough work of Spielman and Teng [39, 40] established the first nearly-linear-time (in nnz($\mathbf{M}$)) solvers. This gave rise to the influential Laplacian Paradigm, which revolutionized algorithmic graph theory and numerical linear algebra, with widespread applications ranging from network science to machine learning (see, e.g., [42, 44]). Among subsequent efforts to generalize the SDD solvers, a line of work [13, 12, 11] developed nearly-linear-time solvers for asymmetric row/column diagonally dominant (RDD/CDD) systems, which significantly expanded the scope of the Laplacian Paradigm.

On the other hand, partly motivated by the advances in quantum algorithms for solving linear systems in sublinear time [24], Andoni, Krauthgamer, and Pogrow [4] pioneered the study of classical algorithms for approximately solving a single entry of SDD systems in sublinear time. Under specific access models and error measures, they established:

- a polylog(n)-time solver for well-conditioned SDD systems; [1]
- a $\widetilde{\Omega}\left(\kappa^2\right)$ lower bound for general SDD systems, where κ is the condition number of $\mathbf{M}$.

The second result demonstrates the necessity of a quadratic dependence on the condition number (equivalently, the reciprocal of the spectral gap) for sublinear-time SDD solvers.

In light of the previous research for nearly-linear-time RDD/CDD solvers and sublinear-time SDD solvers, it is natural to ask whether the sublinear-time SDD solvers can be extended to the more general RDD/CDD cases. In this paper, we initiate a study in this direction and give partial positive answers to this question. We show that the sublinear-time solvers for well-conditioned SDD systems can be extended to "well-structured" RDD/CDD systems, provided that we first define an appropriate generalization of the key structural quantity, the spectral gap for symmetric matrices, to asymmetric matrices. We achieve this by re-characterizing the problem's mathematical structure and introducing a new concept called the *maximum p-norm gap*.

Algorithmically, the sublinear SDD solver in [4] works by solely generating *random-walk samplings* based on $\mathbf{M}$ to approximate a truncated Neumann series of the solution. In contrast, we conduct a deeper investigation of the complexity upper bounds by applying two techniques in addition to random-walk sampling: the *local push method*, which performs local exploration in $\mathbf{M}$; and the *bidirectional method*, which integrates random-walk sampling with local push. Together, we derive a suite of upper bounds for solving RDD/CDD systems under diverse access models and error measures. For instance, we extend the algorithmic result in [4] to RDD systems and derive new results for RCDD systems with smaller dependence on some parameters.

Our algorithmic toolkit and investigation of the diverse upper bounds are inspired by recent advances in local algorithms for estimating PageRank [9] and effective resistance [17] on graphs [50, 49, 47, 45, 6, 43, 14, 52], which are important special cases of solving RDD/CDD systems. The techniques of random-walk sampling [39, 19], local push [2, 1], and the

[1] In fact, [4] considers the (effective) condition number of a normalized version of the involved SDD matrix. See Section 1.3 for details.

bidirectional method [32] have been extensively studied for these problems, and recent works have further uncovered their new properties and optimality in certain settings. Nonetheless, previous works typically analyze their applications to PageRank and effective resistance computation separately. Our perspective of formulating these problems as solving linear systems, however, provides a more general and unified framework for understanding these techniques and problems, revealing their deeper connections. As we shall see, this bigger picture yields novel insights, extended results, and improved complexity bounds.

Notably, our perspective reveals a connection between two fundamental local push algorithms on graphs, namely `ForwardPush` [2] and `BackwardPush` [1][2]. These algorithms iteratively perform local push operations to explore the graph in opposite directions. Although both algorithms share similar approaches, they have been treated as distinct methods for different problems, with each analyzed separately. In contrast, by abstracting both methods as a single algebraic primitive, we demonstrate that `ForwardPush` and `BackwardPush` are equivalent to applying this primitive to different linear systems. This characterization helps to explain their distinct properties and enables unified analysis of both approaches.

On the lower-bound side, our framework inherits the hardness result for sublinear-time SDD solvers, establishing the necessity of our assumption on the maximum p-norm gap; also, known lower bounds for local PageRank computation imply lower bounds on the accuracy parameter for our setting. As our work bridges the study of sublinear-time solvers and local graph algorithms, we believe that further investigation could uncover more connections and results for these topics.

In the remainder of this section, we formally define the problem, present our main contributions, and provide a technical overview.

1.1 Basic Notations

For $n \in \mathbb{Z}^+$, we define $[n] := \{1, 2, \ldots, n\}$. We call a matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ *RDD* (*row diagonally dominant*) if it satisfies $\mathbf{M}(j,j) \geq \sum_{k\neq j} \left|\mathbf{M}(j,k)\right|$ for all $j \in [n]$ and its diagonal entries are positive. We call a matrix *CDD* (*column diagonally dominant*) if its transpose is RDD, and call a matrix *SDD* (*symmetric diagonally dominant*) if it is symmetric and RDD. It is well-known that any SDD matrix is *PSD (positive semidefinite)*. We call a square matrix *Z-matrix* if its off-diagonal entries are nonpositive. We use $\boldsymbol{e}_k$ to denote the k-th canonical unit vector and $\mathbf{1}$ to denote the all-one vector. For any matrix or vector, we use $|\cdot|$ to denote taking the entrywise absolute value.

We call two real numbers $p, q > 1$ *Hölder conjugates* (or q is *conjugate to* p) if they satisfy $1/p + 1/q = 1$. By convention, we also formally let $1/\infty := 0$ and view ∞ and 1 as Hölder conjugates. For any $p \in [1, \infty]$, we use $\|\boldsymbol{x}\|_p$ to denote the *p-norm* of a vector $\boldsymbol{x}$, and $\|\mathbf{M}\|_p$ to denote the *matrix norm induced by vector p-norm* of a matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$.

Restriction and Pseudoinverse. For a subspace $U \subseteq \mathbb{R}^n$, we use $\mathbf{M}|_U$ to denote the restriction of the linear map $\mathbf{M}$ to U, with induced norm $\left\|\mathbf{M}|_U\right\|_p := \max_{\boldsymbol{x}\in U, \|\boldsymbol{x}\|_p=1} \|\mathbf{M}\boldsymbol{x}\|_p$ for any $p \in [1, \infty]$. We write the *pseudoinverse* (a.k.a. *Moore–Penrose inverse*) of $\mathbf{M}$ as $\mathbf{M}^+$.

Spectral Gap. For an SDD matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$, we define its *spectral gap* $\gamma(\mathbf{S})$ as half the smallest nonzero eigenvalue of $\widetilde{\mathbf{S}} := \mathbf{D}_{\mathbf{S}}^{-1/2}\mathbf{S}\mathbf{D}_{\mathbf{S}}^{-1/2}$, where $\mathbf{D}_{\mathbf{S}}$ is the diagonal matrix that satisfies $\mathbf{D}(k,k) = \mathbf{S}(k,k)$ for each $k \in [n]$. The *(effective) condition number* of $\mathbf{S}$, denoted by $\kappa(\mathbf{S})$, is defined as the ratio between the largest and smallest nonzero eigenvalues of $\mathbf{S}$. It holds that $\kappa(\widetilde{\mathbf{S}}) = \Theta\big(1/\gamma(\mathbf{S})\big)$.

[2] Their original names are `ApproximatePageRank` and `ApproxContributions`, respectively.

Graphs. We consider directed graphs $G = (V, E)$, with $n := |V|$ and $m := |E|$. We assume that $V = [n]$. If $(u, v) \in E$, we write $u \to v$. We denote the (possibly weighted) adjacency matrix as $\mathbf{A}_G \in \mathbb{R}^{n \times n}$. For each $v \in V$, we define its indegree $d_G^-(v) := \sum_{u \to v} \mathbf{A}_G(u, v)$ and outdegree $d_G^+(v) := \sum_{v \to u} \mathbf{A}_G(v, u)$. The *outdegree matrix* $\mathbf{D}_G \in \mathbb{R}^{n \times n}$ is the diagonal matrix with $\mathbf{D}_G(v, v) = d_G^+(v)$ for each $v \in V$. We denote G's minimum and maximum outdegree as $\delta_G^+ := \min_{v \in V} \left\{d_G^+(v)\right\}$ and $\Delta_G^+ := \max_{v \in V} \left\{d_G^+(v)\right\}$, respectively.

Eulerian Graphs and Laplacian. We call a graph *Eulerian* if $d_G^-(v) = d_G^+(v)$ for all $v \in V$. On Eulerian graphs, we simply write $d_G(v) := d_G^+(v)$, $\delta_G := \delta_G^+$, and $\Delta_G := \Delta_G^+$. Undirected graphs constitute a special case of Eulerian graphs where each edge corresponds to two directed edges in opposite directions. The *directed Laplacian matrix* is defined as $\mathbf{L}_G := \mathbf{D}_G - \mathbf{A}_G^\top$, which satisfies $\mathbf{1}^\top \mathbf{L}_G = \mathbf{0}^\top$ and is CDDZ. $\mathbf{L}_G$ is RCDDZ for Eulerian graphs and is SDDZ for undirected graphs.

1.2 Problem Formulation

We consider a linear system $\mathbf{M}\boldsymbol{x} = \boldsymbol{b}$, where $\mathbf{M} \in \mathbb{R}^{n \times n}$ is an RDD/CDD matrix and $\boldsymbol{b} \in \operatorname{range}(\mathbf{M})$, and a coefficient vector $\boldsymbol{t} \in \mathbb{R}^n$. We assume that $\boldsymbol{b}, \boldsymbol{t} \neq \mathbf{0}$ and all nonzero entries in $\mathbf{M}$, $\boldsymbol{b}$, and $\boldsymbol{t}$ have absolute values in $\left[1/\operatorname{poly}(n), \operatorname{poly}(n)\right]$. We assume that the algorithms are given the dimension n and have oracle access to $\mathbf{M}$, $\boldsymbol{b}$, and $\boldsymbol{t}$ via the following basic queries:

- Diagonal queries for $\mathbf{M}$: return $\mathbf{M}(k, k)$ in $O(1)$ time for a given index $k \in [n]$;
- Row/column queries for $\mathbf{M}$: return the indices and corresponding values of nonzero entries for a specified row/column of $\mathbf{M}$, in time linear in the number of returned indices;
- Entrywise queries for $\boldsymbol{b}$ and $\boldsymbol{t}$: return $\boldsymbol{b}(k)$ or $\boldsymbol{t}(k)$ in $O(1)$ time for a given index $k \in [n]$.

Our results will assume additional access operations, which will be specified in the statements.

Following the concept of local computation algorithms [36] and the previous work [4], we consider a fixed solution $\boldsymbol{x}^*$ that is determined by $\mathbf{M}$ and $\boldsymbol{b}$, and require invoking the algorithm with different $\boldsymbol{t}$ and accuracy parameters returns estimates of $\boldsymbol{t}^\top \boldsymbol{x}^*$ that are all consistent with the "global" solution $\boldsymbol{x}^*$. Our choice of $\boldsymbol{x}^*$ will be given in Theorem 1.

We also consider the problems of computing (Personalized) PageRank [9] and effective resistance [17] on graphs, viewing them as special cases of our formulation of solving RDD/CDD systems. For these graph problems, we assume the standard adjacency-list model [22], where each degree query takes $O(1)$ time and each neighbor query returns a neighbor index along with the edge weight in $O(1)$ time.

For *Personalized PageRank (PPR)*, we consider a directed graph G, a *decay factor* $\alpha \in (0, 1)$, and a source distribution $\boldsymbol{s} \in \left\{\boldsymbol{y} \in \mathbb{R}^n_{\geq 0} : \|\boldsymbol{y}\|_1 = 1\right\}$. To ensure that PPR is well-defined, we assume that $\delta_G^+ > 0$. The PPR vector $\boldsymbol{\pi}_{G,\alpha,\boldsymbol{s}}$ is defined as the unique solution to the following two equivalent forms of the *PPR equation*:

$$\left(\mathbf{I} - (1-\alpha)\mathbf{A}_G^\top \mathbf{D}_G^{-1}\right) \boldsymbol{\pi}_{G,\alpha,\boldsymbol{s}} = \alpha \boldsymbol{s}, \tag{1}$$

$$\left(\mathbf{D}_G - (1-\alpha)\mathbf{A}_G^\top\right) \left(\mathbf{D}_G^{-1} \boldsymbol{\pi}_{G,\alpha,\boldsymbol{s}}\right) = \alpha \boldsymbol{s}. \tag{2}$$

Both equations can be viewed as linear systems of the form $\mathbf{M}\boldsymbol{x} = \boldsymbol{b}$, where the coefficient matrices $\mathbf{I} - (1-\alpha)\mathbf{A}_G^\top \mathbf{D}_G^{-1}$ and $\mathbf{D}_G - (1-\alpha)\mathbf{A}_G^\top$ are both CDDZ and invertible. Note that for the second form, the solution to the corresponding system is $\mathbf{D}_G^{-1}\boldsymbol{\pi}_{G,\alpha,\boldsymbol{s}}$, an outdegree-scaled version of the PPR vector. We define the PPR value from s to t as $\boldsymbol{\pi}_{G,\alpha}(s, t) :=$

$\boldsymbol{\pi}_{G,\alpha,\boldsymbol{e}_s}(t)$, and the *PageRank* vector $\boldsymbol{\pi}_{G,\alpha}$ as $\boldsymbol{\pi}_{G,\alpha} := \boldsymbol{\pi}_{G,\alpha,1/n\cdot\mathbf{1}}$. It holds that $\boldsymbol{\pi}_{G,\alpha}(t) = \frac{1}{n}\sum_{s\in V}\boldsymbol{\pi}_{G,\alpha}(s,t)$ for all $t \in V$. We also consider the following two equivalent forms of the *PageRank contribution equation*:

$$\big(\mathbf{I} - (1-\alpha)\mathbf{D}_G^{-1}\mathbf{A}_G\big)\boldsymbol{\pi}_{G,\alpha,t}^{-1} = \alpha\boldsymbol{e}_t, \tag{3}$$

$$\big(\mathbf{D}_G - (1-\alpha)\mathbf{A}_G\big)\boldsymbol{\pi}_{G,\alpha,t}^{-1} = \alpha\mathbf{D}_G\boldsymbol{e}_t, \tag{4}$$

where $t \in V$ is a specified target node and $\boldsymbol{\pi}_{G,\alpha,t}^{-1}$ is called the *PageRank contribution vector* to t. It holds that $\boldsymbol{\pi}_{G,\alpha,t}^{-1}(s) = \boldsymbol{\pi}_{G,\alpha}(s,t)$ for all $s \in V$. Similarly, both equations can be viewed as linear systems of the form $\mathbf{M}\boldsymbol{x} = \boldsymbol{b}$, where the coefficient matrices $\mathbf{I} - (1-\alpha)\mathbf{D}_G^{-1}\mathbf{A}_G$ and $\mathbf{D}_G - (1-\alpha)\mathbf{A}_G$ are both RDDZ and invertible. Note that for the second form, the corresponding vector $\boldsymbol{b}$ is $\alpha\mathbf{D}_G\boldsymbol{e}_t$.

For effective resistance, we consider a connected undirected graph G and two distinct nodes $s, t \in V$. The *effective resistance* (a.k.a. *resistance distance*) between s and t, denoted by $R_G(s,t)$, is defined as the equivalent resistance between s, t if the graph is thought of as an electrical network with each edge $(u,v) \in E$ having resistance $1/\mathbf{A}_G(u,v)$. Algebraically, $R_G(s,t) = (\boldsymbol{e}_s - \boldsymbol{e}_t)^\top \mathbf{L}_G^+(\boldsymbol{e}_s - \boldsymbol{e}_t)$. As we will establish in Lemma 27, setting $\mathbf{M} = \mathbf{L}_G$ and $\boldsymbol{b} = \boldsymbol{t} = \boldsymbol{e}_s - \boldsymbol{e}_t$ in our formulation yields $\boldsymbol{t}^\top\boldsymbol{x}^* = R_G(s,t)$. Here, $\mathbf{M} = \mathbf{L}_G$ is SDDZ and $\boldsymbol{b} = \boldsymbol{e}_s - \boldsymbol{e}_t \in \mathrm{range}(\mathbf{M})$.

1.3 Previous Work for SDD Systems

[4] studies the case when the linear system is $\mathbf{S}\boldsymbol{x} = \boldsymbol{b}$ for some SDD matrix $\mathbf{S}$. Define $\mathbf{D}_\mathbf{S}$ as the diagonal matrix that satisfies $\mathbf{D}(k,k) = \mathbf{S}(k,k)$ for each $k \in [n]$ and $\widetilde{\mathbf{S}} := \mathbf{D}_\mathbf{S}^{-1/2}\mathbf{S}\mathbf{D}_\mathbf{S}^{-1/2}$. They formulate the fixed solution as $\boldsymbol{x}^* := \mathbf{D}_\mathbf{S}^{-1/2}\widetilde{\mathbf{S}}^+\mathbf{D}_\mathbf{S}^{-1/2}\boldsymbol{b}$ and give a Neumann series expansion of $\boldsymbol{x}^*$. For the algorithmic results, they assume that the algorithm is given κ, an upper bound on $\kappa(\widetilde{\mathbf{S}})$ (alternatively, γ as a lower bound on $\gamma(\mathbf{S})$) and set a truncation parameter for the Neumann series as $L := \Theta\left(\kappa \log\left(\kappa \cdot \kappa(\mathbf{D}_\mathbf{S})\|\boldsymbol{b}\|_0/\varepsilon\right)\right) = \widetilde{\Theta}(\kappa)$.

Based on the truncated Neumann series, they present a randomized algorithm that, given a coordinate $t \in [n]$, computes an estimate $\hat{x}_t$ satisfying $\left|\hat{x}_t - \boldsymbol{x}^*(t)\right| \le \varepsilon\left\|\mathbf{D}_\mathbf{S}^{-1}\boldsymbol{b}\right\|_\infty$ with probability at least $3/4$. The algorithm runs in time $O\left(f(\mathbf{S})L^3\log L/\varepsilon^2\right) = \widetilde{O}\left(f(\mathbf{S})\kappa^3\varepsilon^{-2}\right) = \widetilde{O}\left(f(\mathbf{S})\gamma^{-3}\varepsilon^{-2}\right)$, where $f(\mathbf{S})$ is the maximum time cost to simulate one step in the random walk defined by $\mathbf{S}$. This result is implicit in the proof of [3, Theorem 5.1].

On the negative side, [4] proves an $\Omega\left(\kappa(\mathbf{S})^2/\log^3 n\right) = \widetilde{\Omega}\left(\kappa(\mathbf{S})^2\right)$ query lower bound (in terms of probing $\boldsymbol{b}$) for achieving a weaker absolute error bound of $\varepsilon\left\|\boldsymbol{x}^*\right\|_\infty$, for $\kappa(\mathbf{S}) = O(\sqrt{n}/\log n)$ and $\varepsilon = \Theta(1/\log n)$. The matrix $\mathbf{S}$ in their hard instance is a Laplacian matrix of a fixed unweighted undirected graph with maximum degree 4 and thus satisfies $\kappa(\widetilde{\mathbf{S}}) = \Theta\big(\kappa(\mathbf{S})\big)$. Therefore, this lower bound can also be written as $\widetilde{\Omega}\left(1/\gamma(\mathbf{S})^2\right)$. To our knowledge, no other work has explicitly studied sublinear-time SDD/RDD/CDD solvers.

1.4 Formulation of x^* and the p-Norm Gaps

Our first contribution is a Neumann-series-based characterization of a solution $\boldsymbol{x}^*$ to $\mathbf{M}\boldsymbol{x} = \boldsymbol{b}$, which is consistent with the solution considered by [4] for SDD systems.

We decompose $\mathbf{M}$ uniquely as $\mathbf{M} = \mathbf{D}_\mathbf{M} - \mathbf{A}_\mathbf{M}^\top$, where $\mathbf{D}_\mathbf{M}$ is a diagonal matrix and all diagonal entries of $\mathbf{A}_\mathbf{M}^\top$ are 0. Define $\widetilde{\mathbf{M}} := \mathbf{D}_\mathbf{M}^{-1/2}\mathbf{M}\mathbf{D}_\mathbf{M}^{-1/2}$ and $\widetilde{\mathbf{A}}_\mathbf{M}^\top := \mathbf{D}_\mathbf{M}^{-1/2}\mathbf{A}_\mathbf{M}^\top\mathbf{D}_\mathbf{M}^{-1/2}$.

The next theorem gives the definition of $\boldsymbol{x}^*$ and its properties.

▶ **Theorem 1.** *For any RDD/CDD* $\mathbf{M}$ *and* $\boldsymbol{b} \in \operatorname{range}(\mathbf{M})$*, define* $\boldsymbol{x}^*$ *to be*

$$\boldsymbol{x}^* := \frac{1}{2}\sum_{\ell=0}^{\infty}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^{\top}\right)\right)^{\ell}\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}.$$

Then $\boldsymbol{x}^*$ *is well-defined and satisfies* $\mathbf{M}\boldsymbol{x}^* = \boldsymbol{b}$*. If* $\mathbf{M}$ *is SDD, then* $\boldsymbol{x}^* = \mathbf{D}_{\mathbf{M}}^{-1/2}\widetilde{\mathbf{M}}^{+}\mathbf{D}_{\mathbf{M}}^{-1/2}\boldsymbol{b}$*.*

Next, we study a truncated version of $\boldsymbol{x}^*$ and upper bound the truncation error. As previous analysis based on eigendecomposition is not directly applicable to asymmetric matrices, we introduce a novel concept called the p-norm gap of $\mathbf{M}$: for any $p \in [1,\infty]$, we define the *p-norm gap* of $\mathbf{M}$ as

$$\gamma_p(\mathbf{M}) := 1 - \left\| \frac{1}{2}\left(\mathbf{I}+\mathbf{D}_{\mathbf{M}}^{-1/q}\mathbf{A}_{\mathbf{M}}^{\top}\mathbf{D}_{\mathbf{M}}^{-1/p}\right)\Big|_{\operatorname{range}\left(\mathbf{I}-\mathbf{D}_{\mathbf{M}}^{-1/q}\mathbf{A}_{\mathbf{M}}^{\top}\mathbf{D}_{\mathbf{M}}^{-1/p}\right)} \right\|_p,$$

where q is conjugate to p. We further define the *maximum p-norm gap* of $\mathbf{M}$ as $\gamma_{\max}(\mathbf{M}) := \max_{p\in[1,\infty]}\gamma_p(\mathbf{M})$.[3] To our knowledge, no prior work has explicitly studied these quantities.

The following theorem guarantees that for any RDD/CDD matrix $\mathbf{M}$, the maximum p-norm gap $\gamma_{\max}(\mathbf{M})$ lies in $(0,1]$, and when $\mathbf{M}$ is SDD, $\gamma_{\max}(\mathbf{M})$ coincides with the spectral gap $\gamma(\mathbf{M})$. Thus, it is a natural generalization of the spectral gap to asymmetric matrices.

▶ **Theorem 2.** *If* $\mathbf{M}$ *is RDD/CDD, then* $0 < \gamma_{\max}(\mathbf{M}) \le 1$*; if* $\mathbf{M}$ *is SDD, then* $\gamma_{\max}(\mathbf{M}) = \gamma_2(\mathbf{M}) = \gamma(\mathbf{M})$*.*

We will devise algorithms to estimate the truncated version of $\boldsymbol{x}^*$, defined as

$$\boldsymbol{x}_L^* := \frac{1}{2}\sum_{\ell=0}^{L-1}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^{\top}\right)\right)^{\ell}\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b} \tag{5}$$

for an integer truncation parameter L. The next theorem upper bounds the truncation error in terms of a given lower bound on the maximum p-norm gap $\gamma_{\max}(\mathbf{M})$.

▶ **Theorem 3.** *Suppose* $0 < \gamma \le \gamma_{\max}(\mathbf{M})$*. To ensure that* $\left|\boldsymbol{t}^{\top}\boldsymbol{x}_L^* - \boldsymbol{t}^{\top}\boldsymbol{x}^*\right| \le \frac{1}{2}\varepsilon$*, it suffices to set*

$$L := \Theta\left(\frac{1}{\gamma}\log\left(\frac{1}{\gamma\varepsilon}\cdot d_{\max}(\mathbf{M})\|\boldsymbol{t}\|_0\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{t}\right\|_{\infty}\|\boldsymbol{b}\|_0\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_{\infty}\right)\right) = \widetilde{\Theta}\left(\frac{1}{\gamma}\right), \tag{6}$$

where $d_{\max}(\mathbf{M})$ *denotes the largest diagonal entry in* $\mathbf{M}$*.*[4]

Relationship with the Formulation in [4]. Our formulations of $\boldsymbol{x}^*$ and the maximum p-norm gap generalize the ones in [4], in the sense that when $\mathbf{M}$ is SDD, $\boldsymbol{x}^* = \mathbf{D}_{\mathbf{M}}^{-1/2}\widetilde{\mathbf{M}}^{+}\mathbf{D}_{\mathbf{M}}^{-1/2}\boldsymbol{b}$ matches their solution and $\gamma_{\max}(\mathbf{M})$ equals $\gamma(\mathbf{M})$, the spectral gap of $\mathbf{M}$. Therefore, the quadratic lower bound on $1/\gamma(\mathbf{M})$ for SDD systems in that paper translates into a quadratic lower bound on $1/\gamma_{\max}(\mathbf{M})$ for general RDD/CDD systems. Also, our setting of the truncation parameter L in Equation (6) matches theirs when $\mathbf{M}$ is SDD and $\boldsymbol{t}$ is a canonical unit vector.

[3] One could alternatively define $\gamma_{\max}(\mathbf{M}) := \max_{p\in\{1,2,\infty\}}\gamma_p(\mathbf{M})$, which yields a possibly smaller quantity but our main results continue to hold with this definition.

[4] Here and after, we use $\widetilde{\Theta}$ and $\widetilde{O}$ to hide polylogarithmic factors in n, $1/\gamma$, $1/\varepsilon$, and (reciprocals of) quantities in $\mathbf{M}$, $\boldsymbol{b}$, and $\boldsymbol{t}$.

1.5 Main Algorithmic Results

For our algorithmic results, we assume that the algorithm is given a quantity $\gamma > 0$ as a lower bound on $\gamma_{\max}(\mathbf{M})$ and an accuracy parameter $\varepsilon > 0$. We use γ, ε, and the specific terms in the accuracy guarantee to set the truncation parameter L according to Equation (6), where we assume that suitable upper bounds on the quantities in the logarithmic factor are known.

The statements of our results will use the following definition. For RDD $\mathbf{M}$, we use $f_{\text{row}}(\mathbf{M})$ to denote the maximum time cost to simulate a single step in the random walk defined by the row substochastic matrix $\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\left|\mathbf{A}_{\mathbf{M}}^{\top}\right|\right)$. This quantity depends on the structure and representation of $\mathbf{M}$. For instance, if each row of $\mathbf{M}$ has at most d nonzero entries, then $f_{\text{row}}(\mathbf{M}) = O(d)$; if the nonzero entries in $\mathbf{A}_{\mathbf{M}}$ have equal absolute values and we are allowed to sample a uniformly random index of the nonzero entries in each column of $\mathbf{A}_{\mathbf{M}}$ in $O(1)$ time, then $f_{\text{row}}(\mathbf{M}) = O(1)$. For CDD $\mathbf{M}$, we define $f_{\text{col}}(\mathbf{M}) := f_{\text{row}}\left(\mathbf{M}^{\top}\right)$.

Each of our algorithmic results excels under different system types, access models, and parameter regimes. We present a selection of our results here, with additional results given in the full version of this paper [28]. Furthermore, a remark at the end of this subsection establishes that most of our results have "symmetric" counterparts (e.g., for CDD systems) that can be obtained by exchanging the roles of certain quantities.

We first present our results of using random-walk sampling for RDD systems.

▶ **Theorem 4.** *Suppose that $\mathbf{M}$ is RDD, we can sample from the distribution $|\boldsymbol{t}|/\|\boldsymbol{t}\|_1$ in $O(1)$ time, and $\|\boldsymbol{t}\|_1$ is known. Then there exists a randomized algorithm that computes an estimate $\hat{x}$ such that $\Pr\left\{\left|\hat{x} - \boldsymbol{t}^{\top}\boldsymbol{x}^*\right| \le \varepsilon\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_{\infty}\right\} \ge \frac{3}{4}$ in time $O\left(f_{\text{row}}(\mathbf{M})\|\boldsymbol{t}\|_1^2 L^3\varepsilon^{-2}\right) = \widetilde{O}\left(f_{\text{row}}(\mathbf{M})\|\boldsymbol{t}\|_1^2\gamma^{-3}\varepsilon^{-2}\right)$.*

This theorem subsumes the main algorithmic result of [4] for SDD systems and extends it to the RDD case while achieving a mild ($\log L$)-factor improvement. Their original result is recovered as a special case when $\mathbf{M}$ is SDD and $\boldsymbol{t}$ is a canonical unit vector. Our algorithm is similar to theirs, but we achieve the improved complexity by adopting a different random-walk sampling scheme.

We also show that for RDDZ $\mathbf{M}$ along with nonnegative vectors $\boldsymbol{b}$ and $\boldsymbol{t}$, if we allow the relaxed error bound $\varepsilon\|\boldsymbol{x}^*\|_{\infty}$, the complexity can be improved to depend quadratically on L via a variance analysis of the random-walk sampling process. We adopt this error measure since it provides a natural accuracy guarantee and has been previously studied in [4]. It is shown in [4] that $\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_{\infty} \le 2\|\boldsymbol{x}^*\|_{\infty}$ for RDD systems.

▶ **Theorem 5.** *Suppose that $\mathbf{M}$ is RDDZ, $\boldsymbol{b}, \boldsymbol{t} \ge \mathbf{0}$, we can sample from the distribution $\boldsymbol{t}/\|\boldsymbol{t}\|_1$ in $O(1)$ time, and $\|\boldsymbol{t}\|_1$ is known. Then there exists a randomized algorithm that computes a $\hat{x}$ such that $\Pr\left\{\left|\hat{x} - \boldsymbol{t}^{\top}\boldsymbol{x}^*\right| \le \varepsilon\|\boldsymbol{x}^*\|_{\infty}\right\} \ge \frac{3}{4}$ in time $O\left(f_{\text{row}}(\mathbf{M})\|\boldsymbol{t}\|_1^2 L^2\varepsilon^{-2}\right) = \widetilde{O}\left(f_{\text{row}}(\mathbf{M})\|\boldsymbol{t}\|_1^2\gamma^{-2}\varepsilon^{-2}\right)$.*

The following theorem further considers the relative error guarantee of $\varepsilon \cdot \boldsymbol{t}^{\top}\boldsymbol{x}^*$. The complexity depends quadratically on L and linearly on $\|\boldsymbol{t}\|_1\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_{\infty}/\boldsymbol{t}^{\top}\boldsymbol{x}^*$ and is also achieved using random-walk sampling.

▶ **Theorem 6.** *Suppose that $\mathbf{M}$ is RDDZ, $\boldsymbol{b}, \boldsymbol{t} \ge \mathbf{0}$, we can sample from the distribution $\boldsymbol{t}/\|\boldsymbol{t}\|_1$ in $O(1)$ time, $\|\boldsymbol{t}\|_1$ and $\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_{\infty}$ are known, and $\boldsymbol{t}^{\top}\boldsymbol{x}^* > 0$. Then there exists a randomized algorithm that computes an estimate $\hat{x}$ such that $\Pr\left\{\left|\hat{x} - \boldsymbol{t}^{\top}\boldsymbol{x}^*\right| \le \varepsilon \cdot \boldsymbol{t}^{\top}\boldsymbol{x}^*\right\} \ge \frac{3}{4}$ in expected time*

$$O\left(\frac{f_{\text{row}}(\mathbf{M})\|\boldsymbol{t}\|_1\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_{\infty} L^2}{\varepsilon^2 \cdot \boldsymbol{t}^{\top}\boldsymbol{x}^*}\right) = \widetilde{O}\left(\frac{f_{\text{row}}(\mathbf{M})\|\boldsymbol{t}\|_1\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_{\infty}}{\gamma^2\varepsilon^2 \cdot \boldsymbol{t}^{\top}\boldsymbol{x}^*}\right).$$

Our next result leverages the local push method to derive a deterministic algorithm for special RCDD systems, whose complexity depends linearly on $1/\varepsilon$.

▶ **Theorem 7.** *Suppose that* $\mathbf{M}$ *is RCDD, its nonzero entries have absolute values of* $\Omega(1)$*, and we can scan the nonzero entries of* $\boldsymbol{b}$ *in* $O(\|\boldsymbol{b}\|_0)$ *time. Then there exists a deterministic algorithm that computes a* $\hat{x}$ *such that* $|\hat{x} - \boldsymbol{t}^\top \boldsymbol{x}^*| \le \varepsilon\|\boldsymbol{t}\|_1$ *in time* $O(\|\boldsymbol{b}\|_0)$ *plus* $O\left(\|\boldsymbol{b}\|_1 L^3 \varepsilon^{-1}\right) = \widetilde{O}\left(\|\boldsymbol{b}\|_1 \gamma^{-3}\varepsilon^{-1}\right)$.

Lastly, applying the bidirectional method to special RCDD systems yields complexity bounds with improved dependence on L and ε, achieving either $L^{7/3}\varepsilon^{-2/3}$ or $L^{5/2}\varepsilon^{-1}$ using different parameter settings.

▶ **Theorem 8.** *Suppose that* $\mathbf{M}$ *is RCDD, its nonzero entries have absolute values of* $\Omega(1)$*, we can sample from the distribution* $|\boldsymbol{t}|/\|\boldsymbol{t}\|_1$ *in* $O(1)$ *time,* $\|\boldsymbol{t}\|_1$ *and* $f_{\text{row}}(\mathbf{M})$ *are known, and we can scan through the nonzero entries of* $\boldsymbol{b}$ *in* $O(\|\boldsymbol{b}\|_0)$ *time. Then there exists a randomized algorithm that computes an estimate* $\hat{x}$ *such that* $\Pr\left\{|\hat{x} - \boldsymbol{t}^\top \boldsymbol{x}^*| \le \varepsilon\right\} \ge \frac{3}{4}$ *in time* $O(\|\boldsymbol{b}\|_0)$ *plus*

$$O\left(\min\left(f_{\text{row}}(\mathbf{M})^{\frac{1}{3}}\|\boldsymbol{t}\|_1^{\frac{2}{3}}\|\boldsymbol{b}\|_1^{\frac{2}{3}} L^{\frac{7}{3}}\varepsilon^{-\frac{2}{3}}, f_{\text{row}}(\mathbf{M})^{\frac{1}{2}}\|\boldsymbol{t}\|_1\|\boldsymbol{b}\|_1^{\frac{1}{2}}\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_\infty^{\frac{1}{2}} L^{\frac{5}{2}}\varepsilon^{-1}\right)\right)$$
$$= \widetilde{O}\left(\min\left(\frac{f_{\text{row}}(\mathbf{M})^{1/3}\|\boldsymbol{t}\|_1^{2/3}\|\boldsymbol{b}\|_1^{2/3}}{\gamma^{7/3}\varepsilon^{2/3}}, \frac{f_{\text{row}}(\mathbf{M})^{1/2}\|\boldsymbol{t}\|_1\|\boldsymbol{b}\|_1^{1/2}\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_\infty^{1/2}}{\gamma^{5/2}\varepsilon}\right)\right).$$

▶ Remark. As we shall see, all theorems in this subsection except Theorem 5 still hold if we replace RDD/RDDZ by CDD/CDDZ, swap $\boldsymbol{b}$ and $\boldsymbol{t}$ (except in $\boldsymbol{t}^\top\boldsymbol{x}^*$), and replace $f_{\text{row}}(\mathbf{M})$ by $f_{\text{col}}(\mathbf{M})$ in the statements. Theorem 5 is an exception because its proof relies on the property that $\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_\infty \le 2\|\boldsymbol{x}^*\|_\infty$ for RDD $\mathbf{M}$, which does not have a straightforward analog for the CDD case.

1.6 Connections to PageRank and Effective Resistance Computation

Our results can be directly applied to the PPR or PageRank contribution equations and the single-pair effective resistance problem to yield complexity bounds for different graph types under various access models and accuracy guarantees. Notably, as we shall see, for PageRank computation, half the decay factor serves as a lower bound on the maximum p-norm gap of the corresponding systems; for effective resistance computation, the spectral gap of the graph Laplacian equals the maximum p-norm gap of the system. Rather than exhaustively presenting all applications of our results, here we highlight selected results that generalize and improve upon the previously best complexity bounds.

The following theorem is derived by applying Theorem 6 to the PageRank equation (2) combined with tighter lower bounds on $\boldsymbol{\pi}_{G,\alpha}(t)$ that we establish for Eulerian graphs.

▶ **Theorem 9.** *For any unweighted Eulerian graph* G*, given the decay factor* α*, a target node* $t \in V$*, and* δ_G*, there exists a randomized algorithm that estimates the PageRank value* $\boldsymbol{\pi}_{G,\alpha}(t)$ *within relative error* ε *with probability at least* $3/4$ *in time*

$$O\left(\frac{1}{\alpha\varepsilon^2}\cdot\frac{d_G(t)}{\delta_G}\cdot\frac{1}{n\boldsymbol{\pi}_{G,\alpha}(t)}\right) = O\left(\frac{1}{\varepsilon^2\delta_G}\cdot\min\left(\frac{d_G(t)}{\alpha^2}, \frac{m/d_G(t)}{\alpha^2}, \frac{\Delta_G}{\alpha}, \frac{\sqrt{m}}{\alpha}\right)\right).$$

This improves over the previously best upper bounds of $O\left(\frac{1}{\varepsilon^2\delta_G}\cdot\min\left(\frac{d_G(t)}{\alpha^2},\frac{\sqrt{m}}{\alpha^2}\right)\right)$, which was given by [45] and stated for unweighted undirected graphs. Our algorithm is essentially the same as that in [45], which generates random walks from the target node t, and the improvement comes from our tighter lower bounds on $\boldsymbol{\pi}_{G,\alpha}(t)$ for Eulerian graphs.

For effective resistance computation, recall that we consider connected undirected graphs G. Lemma 27 establishes that the value $\boldsymbol{t}^\top\boldsymbol{x}^*$ that our algorithms approximate equals $R_G(s,t)$ when setting $\mathbf{M}=\mathbf{L}_G$ and $\boldsymbol{b}=\boldsymbol{t}=\boldsymbol{e}_s-\boldsymbol{e}_t$. Thus, Theorems 4 and 8 directly imply the following complexity bounds for estimating effective resistance.

▶ **Corollary 10.** *For any connected unweighted undirected graph G, given nodes $s,t\in V$ and $\gamma>0$ as a lower bound on $\gamma(\mathbf{L}_G)$, there exists a randomized algorithm that estimates the effective resistance $R_G(s,t)$ within absolute error ε with probability at least $3/4$ in time*

$$O\left(\min\left(\frac{L^3}{\varepsilon^2\cdot\min\left(d_G(s),d_G(t)\right)^2},\frac{L^{7/3}}{\varepsilon^{2/3}},\frac{L^{5/2}}{\varepsilon\cdot\min\left(d_G(s),d_G(t)\right)^{1/2}}\right)\right),$$

where $L:=\Theta\left(\frac{1}{\gamma}\log\left(\frac{1}{\gamma\varepsilon}\left(\frac{1}{d_G(s)}+\frac{1}{d_G(t)}\right)\right)\right)$.

This subsumes the previous bounds of $O\left(\min\left(\frac{L^3\log L}{\varepsilon^2\cdot\min(d_G(s),d_G(t))^2},\frac{L^{7/3}\log L}{\varepsilon^{2/3}}\right)\right)$ given by [14] with essentially the same setting of L. Moreover, since $R_G(s,t)$ can be lower bounded by $1/2/\min\left(d_G(s),d_G(t)\right)$ [34, Corollary 3.3], by setting the absolute error parameter ε to be $\varepsilon_r\cdot 1/2/\min\left(d_G(s),d_G(t)\right)$, the last bound in our result implies that an estimate of $R_G(s,t)$ within relative error ε_r can be computed in time $O\left(\min\left(d_G(s),d_G(t)\right)^{1/2}L^{5/2}\varepsilon_r^{-1}\right)$. This improves over the previous bound of $O\left(\min\left(d_G(s),d_G(t)\right)^{1/2}L^3\log L\cdot\varepsilon_r^{-1}\right)$ given by [52]. Our algorithms are essentially the same as those in [14, 52], i.e., using random-walk sampling and the bidirectional method, and the improvements stem from our simpler sampling scheme that avoids using extra data structures and a refined analysis.

On the other hand, known hardness results for local PageRank and effective resistance computation can potentially yield lower bounds for sublinear-time solvers. We highlight the following result, derived by establishing a reduction from estimating single-node PageRank on undirected graphs to solving SDD systems and applying the lower bound for PageRank computation from [45].

▶ **Theorem 11.** *For any large enough n and $\varepsilon=\Omega(1/n)$, there exist $\boldsymbol{b}\in\mathbb{R}^n$ and $t\in[n]$ that satisfy the following. Every randomized algorithm that, given access to an invertible SDD matrix $\mathbf{S}\in\mathbb{R}^{n\times n}$ whose spectral gap is $\Omega(1)$, succeeds with probability at least $3/4$ to approximate $\boldsymbol{x}^*(t)$ within absolute error $\varepsilon\|\boldsymbol{x}^*\|_\infty$, must probe $\Omega(1/\varepsilon)$ coordinates of $\mathbf{S}$ in the worst case. Here, $\boldsymbol{x}^*=\mathbf{S}^{-1}\boldsymbol{b}$.*

This result gives an $\Omega(1/\varepsilon)$ lower bound for local SDD solvers with accuracy guarantee $\varepsilon\|\boldsymbol{x}^*\|_\infty$, demonstrating the necessity of a linear dependence on $1/\varepsilon$ in our Theorem 4. This lower bound on the accuracy parameter complements the lower bound on the spectral gap given by [4].

1.7 Understanding `ForwardPush` and `BackwardPush` on Graphs

Inspired by the local push algorithms `ForwardPush` and `BackwardPush`, we formulate our `Push` algorithm as a unified primitive for estimating a summation of matrix powers applied to a vector, which is closely related to our approach to solving RDD/CDD systems. This abstraction reveals that `ForwardPush` and `BackwardPush`, despite appearing as distinct algorithms for two problems with different propagation strategies, are equivalent to applying `Push` to different linear systems, modulo variable scaling. The apparent differences arise from the scaling and the distinct behaviors that `Push` exhibits on different types of linear systems.

Specifically, for PageRank computation, `ForwardPush` from node s corresponds to applying `Push` to Equation (2) with $\boldsymbol{s} = \boldsymbol{e}_s$ and outdegree scaling, while `BackwardPush` from node t applies `Push` to the contribution equations (3) or (4). Our analysis demonstrates that `Push` provides closed-form complexity bounds on certain CDD systems and accuracy guarantees on RDD systems, which explains the known computational properties of `ForwardPush` and `BackwardPush` on directed graphs.

For RCDD systems, `Push` inherits both complexity and accuracy advantages, clarifying why `ForwardPush` and `BackwardPush` perform well on undirected graphs. This perspective further reveals that on Eulerian graphs, `ForwardPush` from any node is equivalent to `BackwardPush` from that node on the transpose graph, establishing a basic connection previously unrecognized.

1.8 Technical Overview

Our characterizations of the solution $\boldsymbol{x}^*$ and the p-norm gaps are based on fundamental properties of Neumann series and restricted linear maps. Since asymmetric matrices may be non-diagonalizable, the eigendecomposition techniques used for symmetric matrices become inapplicable. We address this challenge by analyzing the operator norms of restricted linear maps to establish series convergence and derive truncation error bounds.

Our algorithms estimate $\boldsymbol{t}^\top \boldsymbol{x}_L^* = \frac{1}{2}\boldsymbol{t}^\top \sum_{\ell=0}^{L-1} \left(\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)\right)^\ell \mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}$ by adapting three techniques for random-walk probability estimation on graphs: random-walk sampling [39, 19], local push [2, 1], and the bidirectional method [32].

When $\mathbf{M}$ is RDD, the matrix $\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\left|\mathbf{A}_{\mathbf{M}}^\top\right|\right)$ is row substochastic, enabling us to interpret $\boldsymbol{t}^\top \boldsymbol{x}_L^*$ as an expectation over random walks that start from the distribution $|\boldsymbol{t}|/\|\boldsymbol{t}\|_1$ and transition according to $\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\left|\mathbf{A}_{\mathbf{M}}^\top\right|\right)$. Such random walks may terminate early, and the algorithm needs to record the signs of the entries in $\mathbf{A}_{\mathbf{M}}$ along the walk. This method extends the approach in [4], and we adopt a different sampling scheme and conduct variance analysis in some special cases to reduce the dependence on L in the complexity.

Based on local push methods, we formulate a `Push` primitive that estimates the vector $\boldsymbol{x}_L^* = \sum_{\ell=0}^{L-1} \left(\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)\right)^\ell \mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}$ through deterministic local computation. `Push` maintains coordinate variables initialized to $\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}$ and iteratively applies the linear operator $\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)$ to selected coordinates. Our analysis relies on invariant properties preserved by these operations, including an inequality variant that helps to handle negative entries. This approach yields closed-form accuracy guarantees for RDD systems and complexity bounds for special CDD systems. Combining these two aspects yield our result for special RCDD systems.

The bidirectional method combines random-walk sampling and local push from two directions. We adapt the `BiPPR` framework [32], performing `Push` from $\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}$ and exploiting the invariant property to construct an unbiased estimator for $\boldsymbol{t}^\top \boldsymbol{x}_L^*$, which can be sampled via random walks from $|\boldsymbol{t}|/\|\boldsymbol{t}\|_1$ when $\mathbf{M}$ is RDD. By balancing the costs of both components, we achieve improved dependence on L and ε, particularly for RCDD systems.

Crucially, we can transpose the expression for $\boldsymbol{t}^\top \boldsymbol{x}_L^*$ to obtain an equivalent summation with $\mathbf{A}_{\mathbf{M}}^\top$ replaced by $\mathbf{A}_{\mathbf{M}}$ and the roles of $\boldsymbol{b}$ and $\boldsymbol{t}$ exchanged. This allows us to alternatively apply `Push` from $\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{t}$ and random-walk sampling from $|\boldsymbol{b}|/\|\boldsymbol{b}\|_1$ when $\mathbf{M}$ is CDD, leading to symmetric algorithmic procedures and complexity results.

1.9 Paper Organization

The remainder of this paper is organized as follows. Section 2 introduces more related work. Section 3 elaborates on our formulation of $\boldsymbol{x}^*$ and p-norm gaps and proves Theorems 1 to 3. In Sections 4, 5, and 6, we present our algorithms based on random-walk sampling, local push, and the bidirectional method, respectively, and prove Theorems 4 to 8. After that, Sections 7 and 8 relate our problem and results to local computation of PageRank and effective resistance, respectively. In the full version of this paper [28], we provide more preliminaries, discussions on future directions, detailed explanations of the relationship between our `Push` primitive and the `ForwardPush` and `BackwardPush` algorithms, additional results, and all omitted proofs.

2 Other Related Work

A vast literature exists on nearly-linear-time Laplacian solvers and their extensions, including solvers for undirected Laplacian/SDD systems (e.g., [39, 40, 26]) and directed Laplacian/RDD/CDD systems (e.g., [13, 11, 25]). These solvers achieve nearly linear time complexity in $\mathrm{nnz}(\mathbf{M})$ with polylogarithmic dependence on $1/\varepsilon$ and the condition number. The development of global RDD/CDD solvers relies on reductions from solving RDD/CDD systems to solving Eulerian Laplacian systems, combined with efficient methods for computing PPR vectors with small α and the stationary distribution of random walks on graphs [13]. However, the global techniques from algorithmic linear algebra and known reductions to the Eulerian case do not directly apply to the local setting, revealing a fundamental distinction between global and local RDD/CDD solvers. In fact, the lower bounds established in [3] and this work demonstrate that local SDD solvers require polynomial dependence on $1/\varepsilon$ and $1/\gamma$, indicating a separation between global and local SDD solvers.

[16] develops probabilistic logspace solvers for certain classes of directed Laplacian systems. Their method also relies on approximating truncated Neumann series, but they bound the truncation error using spectral radius and Jordan normal form, which yield truncation parameters of at least n^2. Such a huge truncation parameter makes their algorithm and analysis inapplicable to the sublinear-time setting. As an aside, in the quantum regime, [41] presents algorithms for inverting well-conditioned matrices in quantum logspace, but their approaches are not directly applicable to our classical sublinear-time framework.

The idea of using random-walk sampling to solve linear systems dates back to the von Neumann-Ulam algorithm for approximating matrix inversion [20, 48]. The bidirectional method for estimating random-walk probabilities on graphs is first proposed in [33], which is inspired by property testing techniques [23, 27] and later simplified by the `BiPPR` framework [32].

The bidirectional idea has been widely applied to compute PageRank [5, 31, 32, 8, 49, 47, 6, 43], effective resistance [14, 52], heat kernel [8], and Markov Chain transition probability [5]. Among them, [47] proves that the simple `BiPPR` framework computes single-node PageRank on unweighted directed graphs in optimal time complexity (in terms of n and m). Their analysis relies on a new complexity bound of `BackwardPush`, which is parameterized by the PageRank value of the target node. Recently, [52] shows that the bidirectional technique can yield faster algorithms for constructing effective resistance sketch (as defined in [18]) on expander graphs, and [43] combines random-walk sampling with a novel randomized local push technique to improve the complexity of estimating single-node PageRank on directed graphs with bounded in-degree.

[37] uses the bidirectional method to estimate a single element in the product of a matrix power and a vector, which relates to our estimation of $\frac{1}{2}\boldsymbol{t}^\top \sum_{\ell=0}^{L-1} \left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)\right)^\ell \mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}$. However, they only derive an average-case complexity bound under some bounded-norm conditions and discuss its applications to solving PSD systems. In contrast, we conduct a more comprehensive study of this problem and apply it to solving RDD/CDD systems.

PageRank and PPR have been extensively studied and widely applied; we refer interested readers to the surveys [29, 21, 50]. More lower bounds for PageRank computation can be found in [8, 50, 6] and references therein. The recent work [6] conducts a comprehensive study of various types of PPR estimation problems using different graph access queries under both worst-case and average-case settings. For constant decay factors, they provide nearly tight complexity upper and lower bounds for achieving constant relative error guarantees when the target value is above a given threshold.

Effective resistance is ubiquitous in spectral graph theory [17, 34, 38, 25]. A line of work [35, 51, 14, 52] focuses on locally estimating single-pair effective resistance through multi-step random-walk probabilities. [10] studies this problem on non-expander graphs and establishes strong complexity lower bounds, though it does not explicitly give a lower bound on the spectral gap of the graph. Recently, [52] provides a lower bound on the relative error parameter for this problem. Besides, a line of work [30, 18, 52] studies the problem of constructing effective resistance sketches. Technically, [30] leverages random-walk sampling, [18] uses count sketches and SDD solvers, and [52] employs a bidirectional approach.

3 Formulation of $\boldsymbol{x}^*$ and the p-Norm Gaps

In this section, we prove Theorems 1, 2, and 3 and give further explanations on our formulations of $\boldsymbol{x}^*$ and the p-norm gaps.

First, we need the following lemma that upper bounds $\left\|\mathbf{D}_{\mathbf{M}}^{-1/q}\mathbf{A}_{\mathbf{M}}^\top\mathbf{D}_{\mathbf{M}}^{-1/p}\right\|_p$ for certain $\mathbf{M}$ and Hölder conjugates p, q. Its proof is given in the full version of this paper [28].

▶ **Lemma 12.** *The following hold:*

1. *If* $\mathbf{M}$ *is RDD, then* $\left\|\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right\|_\infty \le 1$.
2. *If* $\mathbf{M}$ *is CDD, then* $\left\|\mathbf{A}_{\mathbf{M}}^\top\mathbf{D}_{\mathbf{M}}^{-1}\right\|_1 \le 1$.
3. *If* $\mathbf{M}$ *is RCDD, then* $\left\|\mathbf{D}_{\mathbf{M}}^{-1/q}\mathbf{A}_{\mathbf{M}}^\top\mathbf{D}_{\mathbf{M}}^{-1/p}\right\|_p \le 1$ *for any Hölder conjugates* p, q.

To establish our formulation of $\boldsymbol{x}^*$ in Theorem 1, we use the following lemma, which characterizes the operator norm and Neumann series of certain restricted linear maps. The proof of this lemma is given in the full version of this paper [28].

▶ **Lemma 13.** *Suppose* $\mathbf{X} \in \mathbb{R}^{n\times n}$ *and* $\|\cdot\|$ *is the operator norm induced by some vector norm* $\|\cdot\|$. *If* $\|\mathbf{X}\| \le 1$, *then, for* $\bar{\mathbf{X}} := \frac{1}{2}(\mathbf{I}+\mathbf{X})$, *we have* $\left\|\bar{\mathbf{X}}|_{\mathrm{range}(\mathbf{I}-\mathbf{X})}\right\| < 1$ *and*

$$\left((\mathbf{I}-\mathbf{X})|_{\mathrm{range}(\mathbf{I}-\mathbf{X})}\right)^{-1} = \frac{1}{2}\left((\mathbf{I}-\bar{\mathbf{X}})|_{\mathrm{range}(\mathbf{I}-\mathbf{X})}\right)^{-1} = \frac{1}{2}\sum_{\ell=0}^{\infty}\left(\bar{\mathbf{X}}|_{\mathrm{range}(\mathbf{I}-\mathbf{X})}\right)^\ell.$$

Proof of Theorem 1. First assume that $\mathbf{M}$ is RDD. Applying Lemmas 12 and 13, we have $\left\|\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right\|_\infty \le 1$ and

$$\left((\mathbf{I}-\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top)|_{\mathrm{range}\left(\mathbf{I}-\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)}\right)^{-1} = \frac{1}{2}\sum_{\ell=0}^{\infty}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)\Big|_{\mathrm{range}\left(\mathbf{I}-\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)}\right)^\ell.$$

As $\boldsymbol{b} \in \operatorname{range}(\mathbf{M}) = \operatorname{range}\left(\mathbf{D_M} - \mathbf{A_M^\top}\right)$, we have $\mathbf{D_M^{-1}}\boldsymbol{b} \in \operatorname{range}\left(\mathbf{I} - \mathbf{D_M^{-1}}\mathbf{A_M^\top}\right)$, so $\boldsymbol{x}^*$ converges to

$$\frac{1}{2}\sum_{\ell=0}^{\infty}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D_M^{-1}}\mathbf{A_M^\top}\right)\right)^{\ell}\mathbf{D_M^{-1}}\boldsymbol{b} = \frac{1}{2}\sum_{\ell=0}^{\infty}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D_M^{-1}}\mathbf{A_M^\top}\right)\Big|_{\operatorname{range}\left(\mathbf{I}-\mathbf{D_M^{-1}}\mathbf{A_M^\top}\right)}\right)^{\ell}\mathbf{D_M^{-1}}\boldsymbol{b}$$
$$= \left(\left(\mathbf{I}-\mathbf{D_M^{-1}}\mathbf{A_M^\top}\right)\big|_{\operatorname{range}\left(\mathbf{I}-\mathbf{D_M^{-1}}\mathbf{A_M^\top}\right)}\right)^{-1}\mathbf{D_M^{-1}}\boldsymbol{b}.$$

Thus, we can check that

$$\mathbf{M}\boldsymbol{x}^* = \mathbf{D_M}\left(\mathbf{I}-\mathbf{D_M^{-1}}\mathbf{A_M^\top}\right)\left(\left(\mathbf{I}-\mathbf{D_M^{-1}}\mathbf{A_M^\top}\right)\big|_{\operatorname{range}\left(\mathbf{I}-\mathbf{D_M^{-1}}\mathbf{A_M^\top}\right)}\right)^{-1}\mathbf{D_M^{-1}}\boldsymbol{b} = \mathbf{D_M}\mathbf{D_M^{-1}}\boldsymbol{b} = \boldsymbol{b}.$$

Next, observe that for any Hölder conjugates p, q, we have

$$\mathbf{D_M^{-1/p}}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D_M^{-1/q}}\mathbf{A_M^\top}\mathbf{D_M^{-1/p}}\right)\right)^{\ell}\mathbf{D_M^{-1/q}}$$
$$= \mathbf{D_M^{-1/p}}\left(\frac{1}{2}\mathbf{D_M^{1/p}}\left(\mathbf{I}+\mathbf{D_M^{-1}}\mathbf{A_M^\top}\right)\mathbf{D_M^{-1/p}}\right)^{\ell}\mathbf{D_M^{-1/q}} = \left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D_M^{-1}}\mathbf{A_M^\top}\right)\right)^{\ell}\mathbf{D_M^{-1}}$$

for each $\ell \geq 0$, so

$$\boldsymbol{x}^* = \frac{1}{2}\mathbf{D_M^{-1/p}}\sum_{\ell=0}^{\infty}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D_M^{-1/q}}\mathbf{A_M^\top}\mathbf{D_M^{-1/p}}\right)\right)^{\ell}\mathbf{D_M^{-1/q}}\boldsymbol{b}$$

for any Hölder conjugates p, q.

When $\mathbf{M}$ is CDD, we consider the expression $\boldsymbol{x}^* = \frac{1}{2}\mathbf{D_M^{-1}}\sum_{\ell=0}^{\infty}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{A_M^\top}\mathbf{D_M^{-1}}\right)\right)^{\ell}\boldsymbol{b}$. We have $\left\|\mathbf{A_M^\top}\mathbf{D_M^{-1}}\right\|_1 \leq 1$ by Lemma 12, so applying the above arguments shows that $\boldsymbol{x}^*$ converges to $\mathbf{D_M^{-1}}\left(\left(\mathbf{I}-\mathbf{A_M^\top}\mathbf{D_M^{-1}}\right)\big|_{\operatorname{range}\left(\mathbf{I}-\mathbf{A_M^\top}\mathbf{D_M^{-1}}\right)}\right)^{-1}\boldsymbol{b}$ and

$$\mathbf{M}\boldsymbol{x}^* = \left(\mathbf{I}-\mathbf{A_M^\top}\mathbf{D_M^{-1}}\right)\mathbf{D_M}\mathbf{D_M^{-1}}\left(\left(\mathbf{I}-\mathbf{A_M^\top}\mathbf{D_M^{-1}}\right)\big|_{\operatorname{range}\left(\mathbf{I}-\mathbf{A_M^\top}\mathbf{D_M^{-1}}\right)}\right)^{-1}\boldsymbol{b} = \boldsymbol{b}.$$

So we have proved that $\boldsymbol{x}^*$ is well-defined and satisfies $\mathbf{M}\boldsymbol{x}^* = \boldsymbol{b}$ when $\mathbf{M}$ is RDD or CDD.

Next, assume that $\mathbf{M}$ is SDD. Then $\mathbf{M}$ is RCDD and Lemma 12 implies that $\left\|\widetilde{\mathbf{A}}_{\mathbf{M}}^\top\right\|_2 \leq 1$. Repeating the above arguments with $\mathbf{D_M^{-1/2}}\boldsymbol{b} \in \operatorname{range}\left(\mathbf{I}-\widetilde{\mathbf{A}}_{\mathbf{M}}^\top\right)$ gives

$$\boldsymbol{x}^* = \mathbf{D_M^{-1/2}}\left(\left(\mathbf{I}-\widetilde{\mathbf{A}}_{\mathbf{M}}^\top\right)\Big|_{\operatorname{range}\left(\mathbf{I}-\widetilde{\mathbf{A}}_{\mathbf{M}}^\top\right)}\right)^{-1}\mathbf{D_M^{-1/2}}\boldsymbol{b} = \mathbf{D_M^{-1/2}}\left(\widetilde{\mathbf{M}}\Big|_{\operatorname{range}(\widetilde{\mathbf{M}})}\right)^{-1}\mathbf{D_M^{-1/2}}\boldsymbol{b}.$$

Since $\mathbf{M}$ is symmetric, $\widetilde{\mathbf{M}}$ is also symmetric, so $\operatorname{range}(\widetilde{\mathbf{M}}) = \ker(\widetilde{\mathbf{M}})^\perp$. By the property of the pseudoinverse, we have $\boldsymbol{x}^* = \mathbf{D_M^{-1/2}}\left(\widetilde{\mathbf{M}}\Big|_{\ker(\widetilde{\mathbf{M}})^\perp}\right)^{-1}\mathbf{D_M^{-1/2}}\boldsymbol{b} = \mathbf{D_M^{-1/2}}\widetilde{\mathbf{M}}^+\mathbf{D_M^{-1/2}}\boldsymbol{b}$, finishing the proof. ◀

Recall that if $\mathbf{M}$ is RDD, then $1 - \left\|\mathbf{D_M^{-1}}\mathbf{A_M^\top}\right\|_\infty = \min_{j\in[n]}\left\{\frac{d_{\mathbf{M}}(j)-\sum_{k\neq j}|\mathbf{M}(j,k)|}{d_{\mathbf{M}}(j)}\right\} \geq 0$, and this quantity measures how strongly the diagonal entries dominate the off-diagonal entries in each row. However, this quantity can equal zero, making it unsuitable as a useful notion of "gap." In contrast, our Theorem 2 shows that the maximum p-norm gap

$\gamma_{\max}(\mathbf{M})$ is always strictly positive when $\mathbf{M}$ is RDD/CDD. As an example, $\gamma_\infty(\mathbf{M}) = 1 - \left\| \frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)|_{\mathrm{range}\left(\mathbf{I}-\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)} \right\|_\infty$, which refines the quantity $1 - \left\|\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right\|_\infty$ by replacing $\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top$ with $\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)$ and restricting the operator to range $\left(\mathbf{I} - \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)$.

Although the p-norm gaps only involve operator norms for the restricted linear maps, Theorem 3 shows that they are sufficient to bound the truncation error between $\boldsymbol{t}^\top \boldsymbol{x}_L^*$ and $\boldsymbol{t}^\top \boldsymbol{x}^*$. The proofs of Theorems 2 and 3 are given in the full version of this paper [28].

4 Random-Walk Sampling

This section presents a Monte Carlo algorithm for estimating $\boldsymbol{t}^\top \boldsymbol{x}^*$ via random-walk sampling. All our algorithms in this paper aim to estimate $\boldsymbol{t}^\top \boldsymbol{x}^*$ by approximating $\boldsymbol{t}^\top \boldsymbol{x}_L^*$. By Theorem 3 and our setting of L, it suffices to ensure that the estimate $\hat{x}$ satisfies that $\left|\hat{x} - \boldsymbol{t}^\top \boldsymbol{x}_L^*\right|$ is at most half the desired accuracy guarantee with probability at least $3/4$, and we will omit this matter in the following proofs.

We first focus on RDD systems and transfer the results to CDD systems by transposing the expression of $\boldsymbol{t}^\top \boldsymbol{x}_L^*$ at the end of this section. When $\mathbf{M}$ is RDD, $\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\left|\mathbf{A}_{\mathbf{M}}^\top\right|\right)$ is row substochastic and we can estimate

$$\boldsymbol{t}^\top \boldsymbol{x}_L^* = \frac{1}{2}\boldsymbol{t}^\top \sum_{\ell=0}^{L-1} \left(\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)\right)^\ell \mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}$$

by generating random-walk sampling from $|\boldsymbol{t}|/\|\boldsymbol{t}\|_1$. In each sample, we first sample a walk length ℓ from $[0, L-1]$ uniformly at random and sample a source coordinate v from the distribution $|\boldsymbol{t}|/\|\boldsymbol{t}\|_1$. Then we simulate a lazy random walk for ℓ steps starting from v according to the transition matrix $\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\left|\mathbf{A}_{\mathbf{M}}^\top\right|\right)$. Specifically, at each step from v, with probability $\frac{1}{2}$ the walk stays put at v, with probability $\frac{|\mathbf{A}_{\mathbf{M}}(u,v)|}{2d_{\mathbf{M}}(v)}$ the walk moves to each $u \in [n]$, and with the remaining probability $\frac{1}{2} - \sum_{u\in[n]} \frac{|\mathbf{A}_{\mathbf{M}}(u,v)|}{2d_{\mathbf{M}}(v)}$ the walk terminates, where $d_{\mathbf{M}}(v) := \mathbf{M}(v,v)$. Note that these probabilities lie in $[0,1]$ since $\mathbf{M}$ is RDD. Additionally, we keep track of the product of the signs of the initial entry in $\boldsymbol{t}$ and the entries in $\mathbf{A}_{\mathbf{M}}$ along the walk (where stay-put steps have sign 1), which is denoted as σ. After ℓ steps, if the walk has not terminated and is at coordinate v, we take the value $\sigma \cdot \frac{1}{2}\|\boldsymbol{t}\|_1 \cdot \frac{\boldsymbol{b}(v)}{d_{\mathbf{M}}(v)} \cdot L$ as the estimate of this sample. We repeat this process for n_{s} independent samples and return the average as the final estimate $\hat{x}$. We give a pseudocode for this approach in the full version of this paper [28].

We emphasize that our sampling scheme is different from the framework in [4], in that we first sample the walk length ℓ and then perform ℓ steps of the random walk, while they perform L steps in each sample and take the quantities obtained in each step into account. As it turns out, our scheme is easier to analyze and will save a factor of $\log L$ in the number of samples.

We first establish the unbiasedness of the sampling scheme, whose proof is given in the full version of this paper [28].

▶ **Lemma 14.** *Each sample described above gives an unbiased estimate of* $\boldsymbol{t}^\top \boldsymbol{x}_L^*$.

We can now prove Theorem 4 by applying the Hoeffding bound.

Proof of Theorem 4. We use random-walk sampling as described above. The algorithm returns $\hat{x}$ as the average of n_{s} independent samples, where each sampled value has absolute value at most $\frac{1}{2}\|\boldsymbol{t}\|_1 \left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_\infty L$. Thus, by Lemma 14 and the Hoeffding bound,

$\Pr\left\{\left|\hat{x} - \boldsymbol{t}^\top \boldsymbol{x}_L^*\right| \geq \frac{1}{2}\varepsilon \left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_\infty\right\}$ is upper bounded by

$$2\exp\left(-\frac{2n_s\left(\frac{1}{2}\varepsilon\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_\infty\right)^2}{\left(\|\boldsymbol{t}\|_1\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_\infty L\right)^2}\right) = 2\exp\left(-\frac{n_s\cdot\varepsilon^2}{2\|\boldsymbol{t}\|_1^2 L^2}\right).$$

To guarantee that this probability is at most $1/4$, we set $n_s := \Theta\left(\|\boldsymbol{t}\|_1^2 L^2/\varepsilon^2\right)$.

As each sample simulates at most L steps of random walk, and each step takes $O\big(f_{\text{row}}(\mathbf{M})\big)$ time, the time complexity is $O\big(f_{\text{row}}(\mathbf{M})L\cdot n_s\big) = O\left(f_{\text{row}}(\mathbf{M})\|\boldsymbol{t}\|_1^2 L^3/\varepsilon^2\right)$, as desired. ◀

The proof of Theorem 5 relies on a variance analysis when $\mathbf{A}_{\mathbf{M}}$, $\boldsymbol{b}$, and $\boldsymbol{t}$ are nonnegative. Additionally, Theorem 6 assumes that $\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_\infty$ is known and provides a relative error guarantee. Its proof relies on the Stopping Rule Theorem given in [15] for adaptively setting the number of samples in Monte Carlo estimation. The proofs of Theorems 5 and 6 are given in the full version of this paper [28].

On the other hand, if $\mathbf{M}$ is CDD, then $\mathbf{M}^\top$ is RDD and $\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}|\mathbf{A}_{\mathbf{M}}|\right)$ is row substochastic. As we can transpose the expression of $\boldsymbol{t}^\top \boldsymbol{x}_L^*$ to obtain

$$\boldsymbol{t}^\top \boldsymbol{x}_L^* = \frac{1}{2}\boldsymbol{b}^\top \mathbf{D}_{\mathbf{M}}^{-1}\sum_{\ell=0}^{L-1}\left(\frac{1}{2}\left(\mathbf{I} + \mathbf{A}_{\mathbf{M}}\mathbf{D}_{\mathbf{M}}^{-1}\right)\right)^\ell \boldsymbol{t} = \frac{1}{2}\boldsymbol{b}^\top \sum_{\ell=0}^{L-1}\left(\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}\right)\right)^\ell \mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{t},$$

our algorithms and results for RDD systems (except Theorem 5) applies to CDD systems by interchanging $\boldsymbol{t}$ with $\boldsymbol{b}$ and replacing $\mathbf{A}_{\mathbf{M}}^\top$ with $\mathbf{A}_{\mathbf{M}}$. This justifies our claim that we can derive symmetric results by replacing RDD/RDDZ by CDD/CDDZ, swapping $\boldsymbol{b}$ and $\boldsymbol{t}$ (except in $\boldsymbol{t}^\top \boldsymbol{x}^*$), and replacing $f_{\text{row}}(\mathbf{M})$ by $f_{\text{col}}(\mathbf{M})$ in the theorem statements. This argument also straightforwardly applies to our subsequent algorithms and results based on local push and the bidirectional method.

5 The Local Push Method

In this section, we adapt the local push methods to estimate $\boldsymbol{t}^\top \boldsymbol{x}_L^*$. We first describe our `Push` algorithm as a primitive that can be applied to both RDD and CDD systems. After that, we establish different properties of `Push` for RDD and CDD systems. This leads to our proof for Theorem 7.

For both RDD and CDD systems $\mathbf{M}\boldsymbol{x} = \boldsymbol{b}$, we describe our `Push` algorithm as a primitive that can be used for approximating the vector $2\boldsymbol{x}_L^* = \sum_{\ell=0}^{L-1}\left(\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)\right)^\ell \mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}$. The pseudocode is given in Algorithm 1.

In the `Push` algorithm, the initialization step sets the reserve and residue vectors to $\mathbf{0}$, except that $\boldsymbol{r}^{(0)}$ is set to be $\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}$, which requires $O\big(\|\boldsymbol{b}\|_0\big)$ time if we assume that we can scan through the nonzero entries of $\boldsymbol{b}$ in $O\big(\|\boldsymbol{b}\|_0\big)$ time. Next, the main loop iterates over levels ℓ from 0 to $L-2$. At each level ℓ, the algorithm performs a local push operation on each coordinate v whose residue $\boldsymbol{r}^{(\ell)}(v)$ exceeds the threshold $r_{\max}$ in absolute value. The push operation on v at level ℓ sets the reserve $\boldsymbol{p}^{(\ell)}(v)$ to $\boldsymbol{r}^{(\ell)}(v)$, increments $\boldsymbol{r}^{(\ell+1)}$ by $\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)\left(\boldsymbol{r}^{(\ell)}(v)\boldsymbol{e}_v\right)$, and sets $\boldsymbol{r}^{(\ell)}(v)$ to 0. In effect, the second step in the push operation increases $\boldsymbol{r}^{(\ell+1)}(v)$ by $\frac{1}{2}\boldsymbol{r}^{(\ell)}(v)$ and increases $\boldsymbol{r}^{(\ell+1)}(u)$ by $\frac{\mathbf{A}(v,u)}{2d_{\mathbf{M}}(u)}\cdot \boldsymbol{r}^{(\ell)}(v)$ for each $u \in [n]$ with $\mathbf{A}(v,u) \neq 0$, which can be done in $\left\|\mathbf{M}(\cdot,v)\right\|_0$ time given oracle row/column access to $\mathbf{M}$.

The following lemma gives the key invariant property of the `Push` algorithm. Its proof is based on induction and can be found in the full version of this paper [28].

Algorithm 1 $\texttt{Push}(\mathbf{M}, \boldsymbol{b}, L, r_{\max})$.

Input: oracle access to $\mathbf{M}$ and $\boldsymbol{b}$, truncation parameter L, threshold $r_{\max}$
Output: dictionaries $\boldsymbol{p}^{(\ell)}$ for reserves and $\boldsymbol{r}^{(\ell)}$ for residues, for $\ell = 0, 1, \dots, L-1$

$\boldsymbol{r}^{(\ell)} \leftarrow \mathbf{0}$, $\boldsymbol{p}^{(\ell)} \leftarrow \mathbf{0}$ for $\ell = 0, 1, \dots, L-1$
$\boldsymbol{r}^{(0)} \leftarrow \mathbf{D}_{\mathbf{M}}^{-1} \boldsymbol{b}$
for ℓ from 0 to $L-2$ **do**
  for each v with $\left|\boldsymbol{r}^{(\ell)}(v)\right| > r_{\max}$ **do**
    $\boldsymbol{p}^{(\ell)}(v) \leftarrow \boldsymbol{r}^{(\ell)}(v)$
    $\boldsymbol{r}^{(\ell+1)} \leftarrow \boldsymbol{r}^{(\ell+1)} + \frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^{\top}\right)\left(\boldsymbol{r}^{(\ell)}(v)\boldsymbol{e}_v\right)$
    $\boldsymbol{r}^{(\ell)}(v) \leftarrow 0$
return $\boldsymbol{p}^{(\ell)}$ and $\boldsymbol{r}^{(\ell)}$ for $\ell = 0, 1, \dots, L-1$

▶ **Lemma 15.** *The push operations preserve the following invariant:*

$$\sum_{\ell=0}^{L-1} \left(\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^{\top}\right)\right)^{\ell} \mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b} = \sum_{\ell=0}^{L-1} \boldsymbol{p}^{(\ell)} + \sum_{\ell=0}^{L-1}\sum_{\ell'=0}^{L-\ell-1} \left(\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^{\top}\right)\right)^{\ell} \boldsymbol{r}^{(\ell')}.$$

In light of this invariant, we use $\frac{1}{2}\boldsymbol{t}^{\top}\left(\sum_{\ell=0}^{L-1} \boldsymbol{p}^{(\ell)} + \boldsymbol{r}^{(L-1)}\right)$ as an estimate of $\boldsymbol{t}^{\top}\boldsymbol{x}_L^*$. Note that this quantity can be computed during the push process. The next lemma shows that if $\mathbf{M}$ is RDD, then the absolute error between this quantity and $\boldsymbol{t}^{\top}\boldsymbol{x}_L^*$ can be bounded.

▶ **Lemma 16.** *If* $\mathbf{M}$ *is RDD, then* $\left|\frac{1}{2}\boldsymbol{t}^{\top}\left(\sum_{\ell=0}^{L-1} \boldsymbol{p}^{(\ell)} + \boldsymbol{r}^{(L-1)}\right) - \boldsymbol{t}^{\top}\boldsymbol{x}_L^*\right| \le \frac{1}{2}\|\boldsymbol{t}\|_1 L^2 \cdot r_{\max}$.

Proof. By Lemma 15 and Equation (5), we have

$$\left|\frac{1}{2}\boldsymbol{t}^{\top}\left(\sum_{\ell=0}^{L-1} \boldsymbol{p}^{(\ell)} + \boldsymbol{r}^{(L-1)}\right) - \boldsymbol{t}^{\top}\boldsymbol{x}_L^*\right| = \frac{1}{2}\left|\sum_{\ell'=0}^{L-2}\sum_{\ell=0}^{L-\ell'-1} \boldsymbol{t}^{\top}\left(\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^{\top}\right)\right)^{\ell} \boldsymbol{r}^{(\ell')}\right|$$

$$\le \frac{1}{2}\sum_{\ell'=0}^{L-2}\sum_{\ell=0}^{L-\ell'-1} \|\boldsymbol{t}\|_1 \left\|\left(\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^{\top}\right)\right)^{\ell} \boldsymbol{r}^{(\ell')}\right\|_{\infty} \le \frac{1}{2}\|\boldsymbol{t}\|_1 L^2 \cdot r_{\max},$$

where we used $\left\|\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^{\top}\right)\right\|_{\infty} \le \frac{1}{2}\left(1 + \left\|\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^{\top}\right\|_{\infty}\right) \le 1$ for RDD $\mathbf{M}$ and $\left\|\boldsymbol{r}^{(\ell')}\right\|_{\infty} \le r_{\max}$ for any $\ell' \in [0, L-2]$ as guaranteed by the process of `Push`. ◀

We will also use the following inequality version of the invariant to bound the running time of the `Push` algorithm, whose proof is similar to that of the invariant equation and can be found in the full version of this paper [28].

▶ **Lemma 17.** *The push operations preserve the following inequality:*

$$\sum_{\ell=0}^{L-1} \left(\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\left|\mathbf{A}_{\mathbf{M}}^{\top}\right|\right)\right)^{\ell} \mathbf{D}_{\mathbf{M}}^{-1}|\boldsymbol{b}| \ge \sum_{\ell=0}^{L-1} \left|\boldsymbol{p}^{(\ell)}\right| + \sum_{\ell=0}^{L-1}\sum_{\ell'=0}^{L-\ell-1} \left(\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\left|\mathbf{A}_{\mathbf{M}}^{\top}\right|\right)\right)^{\ell} \left|\boldsymbol{r}^{(\ell')}\right|.$$

The next lemma bounds the complexity of the `Push` algorithm by a convoluted expression. We will shortly see how to simplify this expression for some special systems and settings.

▶ **Lemma 18.** *Suppose that we can scan through the nonzero entries of $\boldsymbol{b}$ in $O\big(\|\boldsymbol{b}\|_0\big)$ time. Then the complexity of the* `Push` *algorithm is bounded by*

$$O\left(\|\boldsymbol{b}\|_0 + \frac{1}{r_{\max}} \sum_{v\in[n]} \left\|\mathbf{M}(\cdot,v)\right\|_0 \cdot \boldsymbol{e}_v^\top \sum_{\ell=0}^{L-1}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D}_{\mathbf{M}}^{-1}\left|\mathbf{A}_{\mathbf{M}}^\top\right|\right)\right)^\ell \mathbf{D}_{\mathbf{M}}^{-1}|\boldsymbol{b}|\right).$$

Proof. Observe that each time a push operation is performed on a coordinate $v\in[n]$, the value of $\boldsymbol{e}_v^\top \sum_{\ell=0}^{L-1}\left|\boldsymbol{p}^{(\ell)}\right|$ is increased by at least $r_{\max}$. However, by Lemma 17, $\boldsymbol{e}_v^\top \sum_{\ell=0}^{L-1}\left|\boldsymbol{p}^{(\ell)}\right|$ is always upper bounded by $\boldsymbol{e}_v^\top \sum_{\ell=0}^{L-1}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D}_{\mathbf{M}}^{-1}\left|\mathbf{A}_{\mathbf{M}}^\top\right|\right)\right)^\ell \mathbf{D}_{\mathbf{M}}^{-1}|\boldsymbol{b}|$. Therefore, the total number of push operations performed on v is at most $\frac{1}{r_{\max}}\cdot \boldsymbol{e}_v^\top \sum_{\ell=0}^{L-1}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D}_{\mathbf{M}}^{-1}\left|\mathbf{A}_{\mathbf{M}}^\top\right|\right)\right)^\ell \mathbf{D}_{\mathbf{M}}^{-1}|\boldsymbol{b}|$. Since each push operation on v takes $O\left(\left\|\mathbf{M}(\cdot,v)\right\|_0\right)$ time, the lemma follows by summing the cost of the push operations over all $v\in[n]$ and adding the $O\big(\|\boldsymbol{b}\|_0\big)$ time for initialization. ◀

If $\mathbf{M}$ is CDD and the nonzero entries of $\mathbf{M}$ have absolute values of $\Omega(1)$, the next lemma shows that the complexity of `Push` can be simplified. Its proof is given in the full version of this paper [28].

▶ **Lemma 19.** *Suppose that $\mathbf{M}$ is CDD, the nonzero entries of $\mathbf{M}$ have absolute values of $\Omega(1)$, and we can scan through the nonzero entries of $\boldsymbol{b}$ in $O\big(\|\boldsymbol{b}\|_0\big)$ time. Then the complexity of the* `Push` *algorithm is $O\big(\|\boldsymbol{b}\|_0+\|\boldsymbol{b}\|_1 L/r_{\max}\big)$.*

Having established the properties of the `Push` algorithm for RDD and CDD systems, we can readily combine them to prove Theorem 7 for RCDD systems.

Proof of Theorem 7. We run `Push` with $r_{\max} := \varepsilon/L^2$ and use $\frac{1}{2}\boldsymbol{t}^\top\left(\sum_{\ell=0}^{L-1}\boldsymbol{p}^{(\ell)}+\boldsymbol{r}^{(L-1)}\right)$ as the result. Since $\mathbf{M}$ is RDD, by Lemma 16, the absolute error between the estimate and $\boldsymbol{t}^\top\boldsymbol{x}_L^*$ is upper bounded by $\frac{1}{2}\|\boldsymbol{t}\|_1 L^2\cdot r_{\max} \le \frac{1}{2}\varepsilon\|\boldsymbol{t}\|_1$. Since $\mathbf{M}$ is CDD and its nonzero entries have absolute values of $\Omega(1)$, by Lemma 19, the time complexity of the `Push` algorithm is $O\big(\|\boldsymbol{b}\|_0+\|\boldsymbol{b}\|_1 L/r_{\max}\big) = O\big(\|\boldsymbol{b}\|_0+\|\boldsymbol{b}\|_1 L^3/\varepsilon\big)$. This finishes the proof. ◀

6 The Bidirectional Method

This section combines the techniques of random-walk sampling and local push to develop bidirectional algorithms for estimating $\boldsymbol{t}^\top\boldsymbol{x}^*$, which leads to Theorem 8.

The framework of the bidirectional method for RDD systems is presented in Algorithm 2. First, we invoke the `Push` algorithm to obtain the reserve and residue vectors. Recall that the invariant equation of `Push` (Lemma 15) implies that

$$\boldsymbol{t}^\top\boldsymbol{x}_L^* - \frac{1}{2}\boldsymbol{t}^\top\left(\sum_{\ell=0}^{L-1}\boldsymbol{p}^{(\ell)}+\boldsymbol{r}^{(L-1)}\right) = \frac{1}{2}\boldsymbol{t}^\top\sum_{\ell'=0}^{L-2}\sum_{\ell=0}^{L-\ell-1}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)\right)^\ell \boldsymbol{r}^{(\ell')}$$
$$= \frac{1}{2}\boldsymbol{t}^\top\sum_{\ell=0}^{L-1}\left(\frac{1}{2}\left(\mathbf{I}+\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)\right)^\ell\left(\sum_{\ell'=0}^{\min(L-\ell-1,L-2)}\boldsymbol{r}^{(\ell')}\right).$$

So, instead of directly using $\frac{1}{2}\boldsymbol{t}^\top\left(\sum_{\ell=0}^{L-1}\boldsymbol{p}^{(\ell)}+\boldsymbol{r}^{(L-1)}\right)$ as an estimate of $\boldsymbol{t}^\top\boldsymbol{x}_L^*$, we estimate the right-hand side of the above equation using random-walk samplings from $|\boldsymbol{t}|/\|\boldsymbol{t}\|_1$ to reduce approximation error. We employ the same sampling scheme as described in Section 4 to obtain a walk length ℓ and the coordinate v reached by the random walk after ℓ steps, but each sampled value now involves the summation $\sum_{\ell'=0}^{\min(L-\ell-1,L-2)}\boldsymbol{r}^{(\ell')}(v)$. We take the average of the estimates across n_{s} independent samples and add it to $\frac{1}{2}\boldsymbol{t}^\top\left(\sum_{\ell=0}^{L-1}\boldsymbol{p}^{(\ell)}+\boldsymbol{r}^{(L-1)}\right)$ to obtain the final estimate $\hat{x}$.

We note that, compared to the sampling scheme in [14], our approach eliminates the need for using additional data structures to maintain the prefix sums of residues, since we can directly compute $\sum_{\ell'=0}^{\min(L-\ell-1,L-2)} \boldsymbol{r}^{(\ell')}(v)$ in $O(L)$ time per sample without increasing the asymptotic complexity.

Algorithm 2 bidirectional method for RDD systems.

Input: oracle access to $\mathbf{M}$, $\boldsymbol{b}$, and $\boldsymbol{t}$, truncation parameter L, threshold $r_{\max}$, number of samples n_s

Output: estimate $\hat{x}$ of $\boldsymbol{t}^\top \boldsymbol{x}^*$

```
r^(ℓ) and p^(ℓ) for ℓ = 0, 1, ..., L − 1 ← Push(M, b, L, r_max)
x̂ ← (1/2) tᵀ ( Σ_{ℓ=0}^{L−1} p^(ℓ) + r^(L−1) )
for j from 1 to n_s do
    ℓ ← uniformly random sample from [0, L − 1]
    v ← random sample according to distribution |t|/‖t‖_1
    σ ← sgn(t(v))
    for k from 1 to ℓ do
        simulate one step of the random walk from one of the following three
         possibilities:
          1. w.p. 1/2, v' ← v // stays put at v
          2. w.p. |A_M(u,v)| / (2 d_M(v)) for each u ∈ [n], v' ← u, σ ← σ · sgn(A_M(u,v)) // moves
         to u
          3. w.p. 1/2 − Σ_{u∈[n]} |A_M(u,v)| / (2 d_M(v)), σ ← 0, break the loop over k // terminates
        v ← v'
    x̂ ← x̂ + (1/n_s) · σ · (1/2) ‖t‖_1 L Σ_{ℓ'=0}^{min(L−ℓ−1, L−2)} r^(ℓ')(v)
return x̂
```

The next lemma establishes the unbiasedness of the bidirectional estimator.

▶ **Lemma 20.** *The sum of* $\frac{1}{2}\boldsymbol{t}^\top \left(\sum_{\ell=0}^{L-1} \boldsymbol{p}^{(\ell)} + \boldsymbol{r}^{(L-1)}\right)$ *and each sampled value in the bidirectional method described above gives an unbiased estimate of* $\boldsymbol{t}^\top \boldsymbol{x}_L^*$.

Proof. Following the proof of Lemma 14, we can show that the expectation of each sampled value is $\frac{1}{2}\boldsymbol{t}^\top \sum_{\ell=0}^{L-1} \left(\frac{1}{2}\left(\mathbf{I} + \mathbf{D}_{\mathbf{M}}^{-1}\mathbf{A}_{\mathbf{M}}^\top\right)\right)^{\ell} \left(\sum_{\ell'=0}^{\min(L-\ell-1,L-2)} \boldsymbol{r}^{(\ell')}\right)$. Combining this with the invariant equation in Lemma 15 completes the proof. ◀

Next, we prove Theorem 8 by proving the two stated complexity bounds separately in the following two lemmas. Their proofs are partly inspired by [14] and [52], respectively. The proof of Lemma 22 uses variance analysis and is given in the full version of this paper [28].

▶ **Lemma 21.** *Suppose the same assumptions as in Theorem 8. Then there exists a randomized algorithm that computes a* $\hat{x}$ *such that* $\Pr\left\{\left|\hat{x} - \boldsymbol{t}^\top \boldsymbol{x}^*\right| \le \varepsilon\right\} \ge \frac{3}{4}$ *in time* $O\left(\|\boldsymbol{b}\|_0\right)$ *plus*

$$O\left(f_{\mathrm{row}}(\mathbf{M})^{1/3} \|\boldsymbol{t}\|_1^{2/3} \|\boldsymbol{b}\|_1^{2/3} L^{7/3} \varepsilon^{-2/3}\right).$$

Proof. We use the bidirectional method as in Algorithm 2. Note that each sampled value equals $\sigma \cdot \frac{1}{2}\|\boldsymbol{t}\|_1 L \sum_{\ell'=0}^{\min(L-\ell-1,L-2)} \boldsymbol{r}^{(\ell')}(v)$ for some $\sigma \in \{0, \pm 1\}$ and $v \in [n]$ and the `Push` algorithm ensures that $\left\|\boldsymbol{r}^{(\ell')}\right\|_\infty \le r_{\max}$ for each $\ell' \in [0, L-2]$. Thus, the absolute value of each sampled value is at most $\frac{1}{2}\|\boldsymbol{t}\|_1 L^2 \cdot r_{\max}$. Using the Hoeffding bound, it follows that

$$\Pr\left\{\left|\hat{x} - \boldsymbol{t}^\top \boldsymbol{x}_L^*\right| \geq \frac{1}{2}\varepsilon\right\} \leq 2\exp\left(-\frac{2n_s(\frac{1}{2}\varepsilon)^2}{(\|\boldsymbol{t}\|_1 L^2 \cdot r_{\max})^2}\right) = 2\exp\left(-\frac{n_s \cdot \varepsilon^2}{2\|\boldsymbol{t}\|_1^2 L^4 \cdot r_{\max}^2}\right).$$

To guarantee that this probability is at most 1/4, we set $n_s := \Theta\left(\|\boldsymbol{t}\|_1^2 L^4 \cdot r_{\max}^2/\varepsilon^2\right)$, where $r_{\max}$ will be determined shortly.

By Lemma 19, the push cost is $O\left(\|\boldsymbol{b}\|_1 L/r_{\max}\right)$. We set $r_{\max} := \frac{\varepsilon^{2/3}\|\boldsymbol{b}\|_1^{1/3}}{f_{\mathrm{row}}(\mathbf{M})^{1/3}\|\boldsymbol{t}\|_1^{2/3} L^{4/3}}$, where $\|\boldsymbol{b}\|_1$ can be computed in $O(\|\boldsymbol{b}\|_0)$ time given our assumptions. Consequently, the cost for random-walk sampling and push both becomes $O\left(f_{\mathrm{row}}(\mathbf{M})^{1/3}\|\boldsymbol{t}\|_1^{2/3}\|\boldsymbol{b}\|_1^{2/3} L^{7/3}\varepsilon^{-2/3}\right)$, completing the proof. ◀

▶ **Lemma 22.** *Suppose the same assumptions as in Theorem 8. Then there exists a randomized algorithm that computes a $\hat{x}$ such that $\Pr\left\{\left|\hat{x} - \boldsymbol{t}^\top \boldsymbol{x}^*\right| \leq \varepsilon\right\} \geq \frac{3}{4}$ in time $O\big(\|\boldsymbol{b}\|_0\big)$ plus*

$$O\left(f_{\mathrm{row}}(\mathbf{M})^{1/2}\|\boldsymbol{t}\|_1\|\boldsymbol{b}\|_1^{1/2}\left\|\mathbf{D}_{\mathbf{M}}^{-1}\boldsymbol{b}\right\|_\infty^{1/2} L^{5/2}\varepsilon^{-1}\right).$$

Proof of Theorem 8. The theorem follows from Lemmas 21 and 22. ◀

7 Connections with PageRank Computation

This section discusses the connections between our framework and PageRank computation and presents proofs for Theorems 9 and 11.

As mentioned in Section 1, the PPR equations (1) and (2) and PageRank contribution equations (3) and (4) can be formulated as RDD/CDD systems. For example, by Equations (1) and (3), for a node $t \in V$, setting $\mathbf{M} = \mathbf{I} - (1-\alpha)\mathbf{A}_G^\top\mathbf{D}_G^{-1}, \boldsymbol{b} = \frac{\alpha}{n}\mathbf{1}, \boldsymbol{t} = \boldsymbol{e}_t$ or $\mathbf{M} = \mathbf{I} - (1-\alpha)\mathbf{D}_G^{-1}\mathbf{A}_G, \boldsymbol{b} = \alpha\boldsymbol{e}_t, \boldsymbol{t} = \frac{1}{n}\mathbf{1}$ in our formulation both yield $\boldsymbol{t}^\top\boldsymbol{x}^* = \boldsymbol{\pi}_{G,\alpha}(t)$; by Equation (2), for nodes $s,t \in V$, setting $\mathbf{M} = \mathbf{D}_G - (1-\alpha)\mathbf{A}_G^\top, \boldsymbol{b} = \alpha\boldsymbol{e}_s, \boldsymbol{t} = \boldsymbol{e}_t$ yields $\boldsymbol{t}^\top\boldsymbol{x}^* = \boldsymbol{\pi}_{G,\alpha}(s,t)/d_G^+(t)$. It is worth noting that on Eulerian graphs, Equations (2) and (4) are RCDD systems, and on undirected graphs, they are SDD systems.

By the definition of the p-norm gaps, we have

$$\gamma_1\left(\mathbf{I} - (1-\alpha)\mathbf{A}_G^\top\mathbf{D}_G^{-1}\right) = 1 - \left\|\frac{1}{2}\left(\mathbf{I} + (1-\alpha)\mathbf{A}_G^\top\mathbf{D}_G^{-1}\right)\Big|_{\mathrm{range}\left(\mathbf{I}-(1-\alpha)\mathbf{A}_G^\top\mathbf{D}_G^{-1}\right)}\right\|_1$$
$$= 1 - \left\|\frac{1}{2}\left(\mathbf{I} + (1-\alpha)\mathbf{A}_G^\top\mathbf{D}_G^{-1}\right)\right\|_1 = 1 - \frac{1}{2}\big(1 + (1-\alpha)\big) = \frac{1}{2}\alpha,$$

where we used $\mathrm{range}\left(\mathbf{I} - (1-\alpha)\mathbf{A}_G^\top\mathbf{D}_G^{-1}\right) = \mathbb{R}^n$ since $\mathbf{I} - (1-\alpha)\mathbf{A}_G^\top\mathbf{D}_G^{-1}$ is invertible. Similarly, we have $\gamma_1\left(\mathbf{D}_G - (1-\alpha)\mathbf{A}_G^\top\right) = \gamma_\infty\left(\mathbf{I} - (1-\alpha)\mathbf{D}_G^{-1}\mathbf{A}_G\right) = \gamma_\infty\left(\mathbf{D}_G - (1-\alpha)\mathbf{A}_G\right) = \frac{1}{2}\alpha$. Thus, $\frac{1}{2}\alpha$ serves as a lower bound on the maximum p-norm gap of all these matrices involved in the PPR and PageRank contribution equations.

7.1 Results for PageRank Computation when $\mathbf{D}_G - (1-\alpha)\mathbf{A}_G^\top$ is RCDD

Our framework provides new insights and results for PageRank computation, in particular when the involved system is RCDD. Theorem 9 stated in the introduction is one such example, which shows that previous results for single-node PageRank computation on undirected graphs can be improved and generalized to Eulerian graphs.

To prove Theorem 9, we investigate the case when the matrix $\mathbf{D}_G - (1-\alpha)\mathbf{A}_G^\top$ in Equation (2) is RCDD. This matrix is CDD, and it is also RDD if $d_G^+(v) \geq (1-\alpha)d_G^-(v)$ holds for all $v \in V$. In particular, this condition holds when G is Eulerian or α is large enough. Now, by applying Theorem 6, we directly obtain the following result.

▶ **Theorem 23.** *For any unweighted graph G and decay factor α, suppose that $d^+_G(v) \geq (1-\alpha)d^-_G(v)$ holds for all $v \in V$. Then there exists a randomized algorithm that, given $t \in V$, δ^+_G, and accuracy parameter ε, computes an estimate of $\boldsymbol{\pi}_{G,\alpha}(t)$ within relative error ε with success probability at least $3/4$ in time*

$$\widetilde{O}\left(\frac{1}{\alpha\varepsilon^2}\cdot\frac{d^+_G(t)}{\delta^+_G}\cdot\frac{1}{n\boldsymbol{\pi}_{G,\alpha}(t)}\right),$$

where $\widetilde{O}$ hides polylog $\left(\frac{n}{\alpha\varepsilon}\right)$ *factors.*

Proof. Consider applying Theorem 6 to Equation (2) with $\boldsymbol{s} = \frac{1}{n}\mathbf{1}$ and $\boldsymbol{t} = \boldsymbol{e}_t$. Note that the corresponding $\left\|\mathbf{D}^{-1}_{\mathbf{M}}\boldsymbol{b}\right\|_\infty$ equals $\frac{\alpha}{n\delta^+_G}$, $L = \widetilde{O}(1/\alpha)$, $f_{\mathrm{row}}(\mathbf{M}) = O(1)$, $\|\boldsymbol{t}\|_1 = 1$, and the obtained $(1 \pm \varepsilon)$-multiplicative approximation of $\boldsymbol{t}^\top\boldsymbol{x}^* = \boldsymbol{\pi}_{G,\alpha}(t)/d^+_G(t)$ directly yields a $(1 \pm \varepsilon)$-multiplicative approximation of $\boldsymbol{\pi}_{G,\alpha}(t)$. Therefore, the time complexity is

$$\widetilde{O}\left(\frac{\alpha/(n\delta^+_G)}{\alpha^2\varepsilon^2\cdot\boldsymbol{\pi}_{G,\alpha}(t)/d^+_G(t)}\right) = \widetilde{O}\left(\frac{1}{\alpha\varepsilon^2}\cdot\frac{d^+_G(t)}{\delta^+_G}\cdot\frac{1}{n\boldsymbol{\pi}_{G,\alpha}(t)}\right),$$

as desired. ◀

To prove Theorem 9, we establish some lower bounds on PageRank values in the next lemma, which may be of independent interest. This lemma is partly inspired by [8, Lemma 5.13], [46, 45], and [47, Theorem 1.1], and we give its proof in the full version of this paper [28].

▶ **Lemma 24.** *For any weighted directed graph G and $t \in V$, we have*

$$\boldsymbol{\pi}_{G,\alpha}(t) \geq \max\left(\frac{\alpha}{n}, \frac{\alpha(1-\alpha)d^-_G(t)}{n\Delta^+_G}, \frac{\alpha(1-\alpha)d^-_G(t)^2}{n\left\|\mathbf{A}_G(\cdot,t)\right\|_\infty\|\mathbf{A}_G\|_{1,1}}, \frac{\alpha(1-\alpha)d^-_G(t)^2}{n\sqrt{n}\left\|\mathbf{A}_G(\cdot,t)\right\|_2\|\mathbf{A}_G\|_{\mathrm{F}}}\right),$$

where $\|\mathbf{A}_G\|_{1,1} := \sum_{u,v\in[n]}\left|\mathbf{A}_G(u,v)\right|$ and $\|\mathbf{A}_G\|_{\mathrm{F}} := \sqrt{\sum_{u,v\in V}\mathbf{A}_G(u,v)^2}$ denote the entrywise 1-norm and the Frobenius norm, respectively. If G is Eulerian, we further have $\boldsymbol{\pi}_{G,\alpha}(t) \geq \frac{d_G(t)}{n\Delta_G}$; if G is unweighted Eulerian, we further have $\boldsymbol{\pi}_{G,\alpha}(t) \geq \frac{\sqrt{1-\alpha}\cdot d_G(t)}{n\sqrt{m}}$.

Proof of Theorem 9. Theorem 23 gives the complexity bound $\widetilde{O}\left(\frac{1}{\alpha\varepsilon^2}\cdot\frac{d_G(t)}{\delta_G}\cdot\frac{1}{n\boldsymbol{\pi}_{G,\alpha}(t)}\right)$. By Lemma 24, on unweighted Eulerian graphs, we have

$$\begin{aligned}\boldsymbol{\pi}_{G,\alpha}(t) &\geq \max\left(\frac{\alpha}{n}, \frac{\alpha(1-\alpha)d^-_G(t)^2}{n\left\|\mathbf{A}_G(\cdot,t)\right\|_\infty\|\mathbf{A}_G\|_{1,1}}, \frac{d_G(t)}{n\Delta_G}, \frac{\sqrt{1-\alpha}\cdot d_G(t)}{n\sqrt{m}}\right)\\ &= \max\left(\frac{\alpha}{n}, \frac{\alpha(1-\alpha)d_G(t)^2}{nm}, \frac{d_G(t)}{n\Delta_G}, \frac{\sqrt{1-\alpha}\cdot d_G(t)}{n\sqrt{m}}\right).\end{aligned}$$

By plugging these lower bounds on $\boldsymbol{\pi}_{G,\alpha}(t)$ into the complexity bound, we obtain the desired results up to polylog $\left(\frac{n}{\alpha\varepsilon}\right)$ factors (where we omit the terms of $1/(1-\alpha)$ since we often consider the case when $\alpha \to 0$). These polylog $\left(\frac{n}{\alpha\varepsilon}\right)$ factors can be removed by using non-truncated random walks for sampling (cf. [45]), leading to the stated complexity bounds. ◀

7.2 A Lower Bound on the Accuracy Parameter for SDD Solvers

This subsection proves Theorem 11. To this end, we establish the following reduction from single-node PageRank computation on undirected graphs to solving SDD systems.

▶ **Lemma 25.** *Suppose that there exists a randomized algorithm that computes an estimate* $\hat{x}_t$ *such that* $\Pr\left\{\left|\hat{x}_t - \boldsymbol{x}^*(t)\right| \leq \varepsilon\|\boldsymbol{x}^*\|_\infty\right\} \geq \frac{3}{4}$ *for any SDD system* $\mathbf{S}\boldsymbol{x} = \boldsymbol{b}$ *in* $O\left(\gamma^{-\nu}\varepsilon^{-\tau}\right)$ *time. Then there exists a randomized algorithm that, given* δ_G*, estimates* $\boldsymbol{\pi}_{G,\alpha}(t)$ *on unweighted undirected graphs* G *within constant relative error and with success probability at least* $3/4$ *in time* $O\left(\left(d_G(t)/\delta_G\right)^\tau/\alpha^{\nu+\tau}\right)$.

To prove this lemma, we use the following upper bound on $\boldsymbol{\pi}_{G,\alpha}(t)$ on Eulerian graphs, whose proof is given in the full version of this paper [28].

▶ **Lemma 26.** *On any Eulerian graph* G *and* $v \in V$*, we have* $\boldsymbol{\pi}_{G,\alpha}(v) \leq \frac{d_G(v)}{n\delta_G}$.

Proof of Lemma 25. Consider the PageRank equation (2) with $\boldsymbol{s} = 1/n \cdot \mathbf{1}$. When G is undirected, the matrix $\mathbf{D}_G - (1-\alpha)\mathbf{A}_G^\top$ is SDD. By setting $\gamma := \alpha/2$ and $\varepsilon := \Theta\big(\alpha\delta_G/d_G(t)\big)$, the supposed algorithm can compute an estimate $\hat{x}_t$ such that $\left|\hat{x}_t - \frac{\boldsymbol{\pi}_{G,\alpha}(t)}{d_G(t)}\right| \leq \varepsilon \cdot \max_{v\in V}\left\{\frac{\boldsymbol{\pi}_{G,\alpha}(v)}{d_G(v)}\right\}$ w.p. at least $3/4$ in time $O\left(\gamma^{-\nu}\varepsilon^{-\tau}\right) = O\left(\left(d_G(t)/\delta_G\right)^\tau/\alpha^{\nu+\tau}\right)$. Using Lemma 26 and $\boldsymbol{\pi}_{G,\alpha}(t) \geq \alpha/n$, we have $\max_{v\in V}\left\{\frac{\boldsymbol{\pi}_{G,\alpha}(v)}{d_G(v)}\right\} \leq \frac{1}{n\delta_G} \leq \frac{\boldsymbol{\pi}_{G,\alpha}(t)}{\alpha\delta_G}$. Thus, with probability at least $3/4$, $\left|d_G(t)\cdot\hat{x}_t - \boldsymbol{\pi}_{G,\alpha}(t)\right| \leq \varepsilon \cdot d_G(t) \cdot \frac{\boldsymbol{\pi}_{G,\alpha}(t)}{\alpha\delta_G} = \Theta\big(\boldsymbol{\pi}_{G,\alpha}(t)\big)$, so $d_G(t)\cdot\hat{x}_t$ is an estimate of $\boldsymbol{\pi}_{G,\alpha}(t)$ within constant relative error, completing the proof. ◀

Proof of Theorem 11. [45] establishes a complexity lower bound of $\Omega\big(d_G(t)/\delta_G\big)$ for estimating $\boldsymbol{\pi}_{G,\alpha}(t)$ within constant relative error with constant success probability on unweighted undirected graphs, where α is constant and the bound holds for any possible combination of δ_G and $d_G(t)$. This lower bound applies to the number of queries to the graph structure. Therefore, combining this lower bound with Lemma 25 and noting that the reduction uses $\varepsilon := \Theta\big(\alpha\delta_G/d_G(t)\big) = \Omega(1/n)$ yield the desired lower bound of $\Omega(1/\varepsilon)$ for $\varepsilon = \Omega(1/n)$. ◀

8 Connections with Effective Resistance Computation

This section justifies the relationship between our framework and effective resistance computation on graphs in Lemma 27 and proves Corollary 10.

Recall that in the context of computing effective resistances, we assume that G is undirected and connected. In our framework, we set $\mathbf{M} = \mathbf{L}_G$ and $\boldsymbol{b} = \boldsymbol{t} = \boldsymbol{e}_s - \boldsymbol{e}_t$. By Theorem 2, $\gamma_{\max}(\mathbf{L}_G) = \gamma(\mathbf{L}_G)$, so a lower bound γ on the spectral gap $\gamma(\mathbf{L}_G)$ serves as a lower bound on the maximum p-norm gap $\gamma_{\max}(\mathbf{L}_G)$. The following lemma states that with this setting, the quantity $\boldsymbol{t}^\top\boldsymbol{x}^*$ that our algorithms approximate equals the effective resistance $R_G(s,t)$.

▶ **Lemma 27.** *When* $\mathbf{M} = \mathbf{L}_G$ *and* $\boldsymbol{b} = \boldsymbol{t} = \boldsymbol{e}_s - \boldsymbol{e}_t$*, we have* $\boldsymbol{t}^\top\boldsymbol{x}^* = R_G(s,t)$.

Lemma 27 can be proved using the results in [7], and we provide a different self-contained proof in the full version of this paper [28].

Now we can directly apply Theorems 4 and 8 to prove Corollary 10. The only remaining detail in the proof is to derive a better setting of L for the case $\boldsymbol{b} = \boldsymbol{t} = \boldsymbol{e}_s - \boldsymbol{e}_t$.

Proof of Corollary 10. Following the proof of Theorem 3, we have $\left|\boldsymbol{t}^\top\boldsymbol{x}_L^* - \boldsymbol{t}^\top\boldsymbol{x}^*\right| \leq \frac{1}{2\gamma}\cdot e^{-\gamma L}\cdot\left\|\mathbf{D}_\mathbf{M}^{-1/2}\boldsymbol{t}\right\|_2\left\|\mathbf{D}_\mathbf{M}^{-1/2}\boldsymbol{b}\right\|_2 = \frac{1}{2\gamma}\cdot e^{-\gamma L}\left(\frac{1}{d_G(s)} + \frac{1}{d_G(t)}\right)$ when $\mathbf{M} = \mathbf{L}_G$ and $\boldsymbol{b} = \boldsymbol{t} = \boldsymbol{e}_s - \boldsymbol{e}_t$. Thus, setting $L := \Theta\left(\frac{1}{\gamma}\log\left(\frac{1}{\gamma\varepsilon}\left(\frac{1}{d_G(s)} + \frac{1}{d_G(t)}\right)\right)\right)$ ensures that $\left|\boldsymbol{t}^\top\boldsymbol{x}_L^* - \boldsymbol{t}^\top\boldsymbol{x}^*\right| \leq \frac{1}{2}\varepsilon$. The corollary then follows by applying Theorems 4 and 8 with this setting of L and noting that $\left\|\mathbf{D}_\mathbf{M}^{-1}\boldsymbol{b}\right\|_\infty = 1/\min\big(d_G(s), d_G(t)\big)$ and $f_{\text{row}}(\mathbf{M}) = O(1)$ in this case. ◀

References

1 Reid Andersen, Christian Borgs, Jennifer T. Chayes, John E. Hopcroft, Vahab S. Mirrokni, and Shang-Hua Teng. Local computation of PageRank contributions. *Internet Mathematics*, 5(1):23–45, 2008. doi:10.1080/15427951.2008.10129302.

2 Reid Andersen, Fan R. K. Chung, and Kevin J. Lang. Using PageRank to locally partition a graph. *Internet Mathematics*, 4(1):35–64, 2007. doi:10.1080/15427951.2007.10129139.

3 Alexandr Andoni, Robert Krauthgamer, and Yosef Pogrow. On solving linear systems in sublinear time. *CoRR*, abs/1809.02995, 2018. doi:10.48550/arXiv.1809.02995.

4 Alexandr Andoni, Robert Krauthgamer, and Yosef Pogrow. On solving linear systems in sublinear time. In *Proceedings of the 10th Innovations in Theoretical Computer Science Conference*, volume 124, pages 3:1–3:19, 2019. doi:10.4230/LIPIcs.ITCS.2019.3.

5 Siddhartha Banerjee and Peter Lofgren. Fast bidirectional probability estimation in Markov models. In *Advances in Neural Information Processing Systems 28*, pages 1423–1431, 2015. URL: https://proceedings.neurips.cc/paper/2015/hash/ede7e2b6d13a41ddf9f4bdef84fdc737-Abstract.html.

6 Christian Bertram, Mads Vestergaard Jensen, Mikkel Thorup, Hanzhi Wang, and Shuyi Yan. Estimating random-walk probabilities in directed graphs. *CoRR*, abs/2504.16481, 2025. doi:10.48550/arXiv.2504.16481.

7 Enrico Bozzo. The Moore-Penrose inverse of the normalized graph Laplacian. *Linear Algebra and its Applications*, 439(10):3038–3043, 2013. doi:10.1016/j.laa.2013.08.039.

8 Marco Bressan, Enoch Peserico, and Luca Pretto. Sublinear algorithms for local graph-centrality estimation. *SIAM Journal on Computing*, 52(4):968–1008, 2023. doi:10.1137/19M1266976.

9 Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998. doi:10.1016/S0169-7552(98)00110-X.

10 Dongrun Cai, Xue Chen, and Pan Peng. Effective resistances in non-expander graphs. In *Proceedings of the 31st Annual European Symposium on Algorithms*, volume 274, pages 29:1–29:18, 2023. doi:10.4230/LIPIcs.ESA.2023.29.

11 Michael B. Cohen, Jonathan A. Kelner, Rasmus Kyng, John Peebles, Richard Peng, Anup B. Rao, and Aaron Sidford. Solving directed Laplacian systems in nearly-linear time through sparse LU factorizations. In *Proceedings of the 59th IEEE Symposium on Foundations of Computer Science*, pages 898–909, 2018. doi:10.1109/FOCS.2018.00089.

12 Michael B. Cohen, Jonathan A. Kelner, John Peebles, Richard Peng, Anup B. Rao, Aaron Sidford, and Adrian Vladu. Almost-linear-time algorithms for Markov chains and new spectral primitives for directed graphs. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, pages 410–419, 2017. doi:10.1145/3055399.3055463.

13 Michael B. Cohen, Jonathan A. Kelner, John Peebles, Richard Peng, Aaron Sidford, and Adrian Vladu. Faster algorithms for computing the stationary distribution, simulating random walks, and more. In *Proceedings of the 57th IEEE Symposium on Foundations of Computer Science*, pages 583–592, 2016. doi:10.1109/FOCS.2016.69.

14 Guanyu Cui, Hanzhi Wang, and Zhewei Wei. Mixing time matters: Accelerating effective resistance estimation via bidirectional method. In *Proceedings of the 31st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 177–188, 2025. doi:10.1145/3690624.3709298.

15 Paul Dagum, Richard M. Karp, Michael Luby, and Sheldon M. Ross. An optimal algorithm for Monte Carlo estimation. *SIAM Journal on Computing*, 29(5):1484–1496, 2000. doi:10.1137/S0097539797315306.

16 Dean Doron, François Le Gall, and Amnon Ta-Shma. Probabilistic logarithmic-space algorithms for Laplacian solvers. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 81, pages 41:1–41:20, 2017. doi:10.4230/LIPIcs.APPROX-RANDOM.2017.41.

17 Peter G. Doyle and J. Laurie Snell. *Random walks and electric networks*, volume 22. The Mathematical Association of America, 1984.

18 Rajat Vadiraj Dwaraknath, Ishani Karmarkar, and Aaron Sidford. Towards optimal effective resistance estimation. In *Advances in Neural Information Processing Systems 36*, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/b8e2046160a568145af6d42eeef199f4-Abstract-Conference.html.

19 Dániel Fogaras, Balázs Rácz, Károly Csalogány, and Tamás Sarlós. Towards scaling fully Personalized PageRank: Algorithms, lower bounds, and experiments. *Internet Mathematics*, 2(3):333–358, 2005. doi:10.1080/15427951.2005.10129104.

20 George E. Forsythe and Richard A. Leibler. Matrix inversion by a Monte Carlo method. *Mathematics of Computation*, 4(31):127–129, 1950. URL: https://www.ams.org/journals/mcom/1950-04-031/S0025-5718-1950-0038138-X/home.html.

21 David F. Gleich. Pagerank beyond the web. *SIAM Review*, 57(3):321–363, 2015. doi:10.1137/140976649.

22 Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002. doi:10.1007/S00453-001-0078-7.

23 Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. In *Studies in Complexity and Cryptography*, volume 6650, pages 68–75. Springer, 2011. doi:10.1007/978-3-642-22670-0_9.

24 Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. doi:10.1103/PhysRevLett.103.150502.

25 Arun Jambulapati, Sushant Sachdeva, Aaron Sidford, Kevin Tian, and Yibin Zhao. Eulerian graph sparsification by effective resistance decomposition. In *Proceedings of the 2025 ACM-SIAM Symposium on Discrete Algorithms*, pages 1607–1650, 2025. doi:10.1137/1.9781611978322.50.

26 Arun Jambulapati and Aaron Sidford. Ultrasparse ultrasparsifiers and faster Laplacian system solvers. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms*, pages 540–559, 2021. doi:10.1137/1.9781611976465.33.

27 Satyen Kale, Yuval Peres, and C. Seshadhri. Noise tolerance of expanders and sublinear expansion reconstruction. *SIAM Journal on Computing*, 42(1):305–323, 2013. doi:10.1137/110837863.

28 Tsz Chiu Kwok, Zhewei Wei, and Mingji Yang. On solving asymmetric diagonally dominant linear systems in sublinear time. *CoRR*, abs/2509.13891, 2025. doi:10.48550/arXiv.2509.13891.

29 Amy Nicole Langville and Carl Dean Meyer. Survey: Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380, 2003. doi:10.1080/15427951.2004.10129091.

30 Lawrence Li and Sushant Sachdeva. A new approach to estimating effective resistances and counting spanning trees in expander graphs. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms*, pages 2728–2745, 2023. doi:10.1137/1.9781611977554.ch102.

31 Peter Lofgren, Siddhartha Banerjee, and Ashish Goel. Bidirectional PageRank estimation: from average-case to worst-case. In *Proceedings of the 12th International Workshop on Algorithms and Models for the Web-Graph*, volume 9479, pages 164–176, 2015. doi:10.1007/978-3-319-26784-5_13.

32 Peter Lofgren, Siddhartha Banerjee, and Ashish Goel. Personalized PageRank estimation and search: a bidirectional approach. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, pages 163–172, 2016. doi:10.1145/2835776.2835823.

33 Peter Lofgren, Siddhartha Banerjee, Ashish Goel, and Seshadhri Comandur. FAST-PPR: Scaling Personalized PageRank estimation for large graphs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1436–1445, 2014. doi:10.1145/2623330.2623745.

34 László Lovász. Random walks on graphs: a survey. *Combinatorics, Paul Erdös is Eighty*, 2(1-46):4, 1993.

35 Pan Peng, Daniel Lopatta, Yuichi Yoshida, and Gramoz Goranci. Local algorithms for estimating effective resistance. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1329–1338, 2021. doi:10.1145/3447548.3467361.

36 Ronitt Rubinfeld, Gil Tamir, Shai Vardi, and Ning Xie. Fast local computation algorithms. In *Proceedings of Innovations in Computer Science*, pages 223–238, 2011. URL: http://conference.iiis.tsinghua.edu.cn/ICS2011/content/papers/36.html.

37 Nitin Shyamkumar, Siddhartha Banerjee, and Peter Lofgren. Sublinear estimation of a single element in sparse linear systems. In *Proceedings of the 54th Annual Allerton Conference on Communication, Control, and Computing*, pages 856–860, 2016. doi:10.1109/ALLERTON.2016.7852323.

38 Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011. doi:10.1137/080734029.

39 Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 81–90, 2004. doi:10.1145/1007352.1007372.

40 Daniel A. Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014. doi:10.1137/090771430.

41 Amnon Ta-Shma. Inverting well conditioned matrices in quantum logspace. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 881–890, 2013. doi:10.1145/2488608.2488720.

42 Shang-Hua Teng. The Laplacian Paradigm: Emerging algorithms for massive graphs. In *Proceedings of the 7th Annual Conference on Theory and Applications of Models of Computation*, volume 6108, pages 2–14, 2010. doi:10.1007/978-3-642-13562-0_2.

43 Mikkel Thorup, Hanzhi Wang, Zhewei Wei, and Mingji Yang. PageRank centrality in directed graphs with bounded in-degree. In *Proceedings of the 2026 ACM-SIAM Symposium on Discrete Algorithms*, 2026. To appear. arXiv preprint at https://arxiv.org/abs/2508.01257. doi:10.48550/arXiv.2508.01257.

44 Nisheeth K. Vishnoi. $L\mathbf{x} = \mathbf{b}$. *Foundations and Trends in Theoretical Computer Science*, 8(1-2):1–141, 2013. doi:10.1561/0400000054.

45 Hanzhi Wang. Revisiting local PageRank estimation on undirected graphs: Simple and optimal. In *Proceedings of the 30th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 3036–3044, 2024. doi:10.1145/3637528.3671820.

46 Hanzhi Wang and Zhewei Wei. Estimating single-node PageRank in $\tilde{O}\left(\min\left\{d_t, \sqrt{m}\right\}\right)$ time. *Proceedings of the VLDB Endowment*, 16(11):2949–2961, 2023. doi:10.14778/3611479.3611500.

47 Hanzhi Wang, Zhewei Wei, Ji-Rong Wen, and Mingji Yang. Revisiting local computation of PageRank: Simple and optimal. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 911–922, 2024. doi:10.1145/3618260.3649661.

48 Wolfgang R. Wasow. A note on the inversion of matrices by random walks. *Mathematical Tables and Other Aids to Computation*, 6(38):78–81, 1952. doi:10.2307/2002546.

49 Zhewei Wei, Ji-Rong Wen, and Mingji Yang. Approximating single-source Personalized PageRank with absolute error guarantees. In *Proceedings of the 27th International Conference on Database Theory*, volume 290, pages 9:1–9:19, 2024. doi:10.4230/LIPIcs.ICDT.2024.9.

50 Mingji Yang, Hanzhi Wang, Zhewei Wei, Sibo Wang, and Ji-Rong Wen. Efficient algorithms for Personalized PageRank computation: a survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(9):4582–4602, 2024. doi:10.1109/TKDE.2024.3376000.

51 Renchi Yang and Jing Tang. Efficient estimation of pairwise effective resistance. *Proceedings of the ACM International Conference on Management of Data*, 1(1):16:1–16:27, 2023. doi:10.1145/3588696.

52 Yichun Yang, Rong-Hua Li, Meihao Liao, and Guoren Wang. Improved algorithms for effective resistance computation on graphs. In *Proceedings of the 38th Conference on Learning Theory*, volume 291, pages 5892–5920, 2025. URL: https://proceedings.mlr.press/v291/yichun25a.html.