Fast Algorithms for the Label Propagation Operator on Signed Graphs

Yubo Sun[1], **Zhongzhi Zhang**[1,2*]
[1]Research Institute of Intelligent Complex Systems, Fudan University
[2]College of Computer Science and Artificial Intelligence, Fudan University
ybsun25@m.fudan.edu.cn, zhangzz@fudan.edu.cn

Abstract

Diverse applications on signed networks, ranging from leader-follower opinion dynamics to graph semi-supervised learning, fundamentally reduce to solving the discrete Dirichlet boundary value problem. The computational bottleneck of this problem lies in evaluating the entries of the label propagation operator. While existing approximation methods based on absorbing random walks attempt to address this complexity, they often suffer from high variance and limited precision. To overcome these limitations, we propose MultiTreeQ, a novel and efficient framework designed to approximate the operator's entries with high accuracy. Specifically, MultiTreeQ is constructed based on the Signed Multi-Rooted Spanning Tree and leverages Rao-Blackwellized variance reduction to significantly improve estimation precision. Extensive experiments on real-world networks confirm that MultiTreeQ substantially outperforms existing baselines, scaling to massive graphs with up to 10 million nodes while delivering remarkable improvements in computational efficiency and precision. Our code is available at https://github.com/SunYubo23/label-propagation-operator.

1 Introduction

Many real-world systems contain both cooperative and competitive interactions. To capture this reality, signed graphs have become a standard tool [3, 13, 8]. Within this model, positive edges indicate friendly or aligned relationships, whereas negative edges capture conflict or opinion disagreement. With the growing focus on network dynamics such as group polarization and opinion formation [21, 40, 44, 49], node state inference on signed networks has become a critical frontier in data mining.

In the field of opinion dynamics, the classical DeGroot model [7, 50, 51] seeks to predict the steady state opinion distribution in a network containing stubborn leaders. Meanwhile, in the machine learning domain, Graph Semi-Supervised Learning based on Gaussian Random Fields [53] aims to infer the categories of massive unlabeled nodes using an extremely small set of labeled nodes. Despite their distinct application contexts, these models are mathematically equivalent when dealing with networks containing hard labels, namely, labeled (leader) nodes whose states cannot be altered. Fundamentally, they can all be unified and formulated as the Dirichlet boundary value problem on signed graphs. The computational core to solving this problem lies in evaluating how known information propagates from the labeled (leader) set L to the unlabeled (follower) set U. Algebraically, this strictly corresponds to calculating the global Label Propagation Operator $\boldsymbol{Q} = -\boldsymbol{L}_{U,U}^{-1}\boldsymbol{L}_{U,L}$, where $\boldsymbol{L}$ denotes the Laplacian matrix of the signed graph. In the leader-follower DeGroot model on signed graphs, each entry $\boldsymbol{Q}_{ij}$ quantifies the signed influence of leader j on follower i. Efficiently approximating these entries enables applications such as leader influence analysis and repeated opinion evaluations under different leader configurations.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, computing this operator efficiently on large-scale signed graphs poses severe challenges. Direct matrix inversion incurs a prohibitive cubic time complexity of $\mathcal{O}(|V_U|^3)$, where V_U denotes the set of unlabeled nodes. Although numerical methods such as CG and PCG can efficiently solve the underlying linear systems [37, 17, 6, 38], they are primarily designed to compute the solution for a given right-hand side rather than explicitly construct the propagation operator $\boldsymbol{Q}$. Consequently, applications requiring numerous operator entries or repeated evaluations under different label configurations may incur substantial computational costs. To circumvent these algebraic constraints, recent work [52] employs absorbing random walks to approximate the entries of $\boldsymbol{Q}$. Nevertheless, such random walk estimators inherently possess high variance, demanding large sample sizes to achieve target precision, which restricts their scalability. Additionally, while spanning tree formulations [27] address this in unsigned graphs, they are inapplicable to signed networks due to negative edges.

Motivated by these limitations, this paper presents MultiTreeQ, a novel and fast algorithmic framework designed to evaluate the entries of the signed label propagation operator $\boldsymbol{Q}$ with high precision. Our main contributions are summarized as follows:

- We introduce a topological structure termed the Signed Multi-Rooted Spanning Tree (SMRST), and establish the mathematical connection between the entries of $\boldsymbol{Q}$ and SMRSTs, bridging the theoretical gap caused by negative edges.
- Building upon this structure, we propose the MultiTreeQ algorithm. By utilizing the Rao-Blackwellized variance reduction technique, MultiTreeQ guarantees lower estimation variance compared to random walk baselines, thereby improving approximation precision.
- We conduct comprehensive experiments on real-world networks. Empirical results demonstrate that MultiTreeQ outperforms the random walk baseline in precision while maintaining comparable efficiency.

2 Related Work

In this section, we provide a brief review of existing work related to ours, including the following three aspects.

Gaussian Random Field. The Gaussian Random Field (GRF) model [19, 4] is a foundational paradigm for graph semi-supervised learning, minimizing a Laplacian-induced quadratic energy to enforce label smoothness [53]. Governed by a tunable hyperparameter, GRF relies on the cluster assumption that strongly connected nodes share similar representations [2, 48]. Its optimal solution yields a harmonic function where each unlabeled node converges to the weighted average of its neighbors [29]—a local averaging mechanism mathematically equivalent to steady-state consensus in opinion dynamics like the DeGroot model [7]. These Laplacian smoothing principles have profoundly influenced modern graph learning [16, 31]. While traditional embeddings enforce explicit smoothness and rely on inner-product decoders for downstream tasks like link prediction, modern graph neural networks internalize these principles via message-passing and employ MLPs to capture complex, non-linear structural dependencies.

Leader-Follower DeGroot Model. Rooted in mathematical sociology, the DeGroot model [7, 36, 34] provides a canonical framework for opinion diffusion in networked systems. In its leader-follower variant, networks are partitioned into stubborn leaders with fixed views and regular followers who iteratively update their opinions via a weighted average of their neighbors [50, 51]. At steady-state, every follower's opinion converges precisely to this local average. Remarkably, this consensus mechanism is mathematically identical to the harmonic function constraint within the Gaussian Random Field framework [53, 10]. Consequently, both paradigms yield the exact same algebraic solution $\boldsymbol{x}_U = -\boldsymbol{L}_{U,U}^{-1}\boldsymbol{L}_{U,L}\boldsymbol{x}_L$. This profound structural equivalence reveals that evaluating the label propagation operator $\boldsymbol{Q}$, which essentially encodes absorbing probabilities, forms the universal computational bottleneck across both domains. Overcoming the sheer complexity of this heavy matrix inversion on massive graphs has inspired a fundamental shift towards nearly-linear time Laplacian solvers [38] and, more critically, sublinear random walk-based sampling techniques [30, 5] to efficiently estimate the propagation entries.

Interpretation of the label propagation operator. On unsigned graphs, each entry $\boldsymbol{Q}_{ij}$ is interpreted as the absorption probability that a random walk originating from an unlabeled node i is captured by

a labeled node j [53]. Leveraging this probabilistic perspective, Zhou et al. [49] utilized random walk sampling to address the opinion maximization problem [9, 15] in social networks. In subsequent research, Liao et al. [27] proposed a rooted spanning tree interpretation for these entries and applied it to efficiently query resistance distances in graphs. In a recent study, Zhou et al. [52] generalized absorbing random walks to signed networks. They formulated $\boldsymbol{Q}_{ij}$ as the probability difference between positive and negative absorbing walks originating from node i and terminating at node j, utilizing truncation techniques for acceleration. However, this random walk estimator inherently exhibits high variance, and the applied truncation techniques render it biased, introducing additional approximation error.

3 Preliminaries

In this section, we introduce some useful notations and fundamental concepts for the convenience of description and analysis.

3.1 Graph and Matrix Definitions

We consider a signed graph $G = (V, E, \sigma)$, where V is the set of n nodes, $E \subseteq V \times V$ is the set of edges, and $\sigma : E \to \mathbb{R}$ assigns a sign to each edge, indicating positive or negative relationships. The signed adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ is defined as $\boldsymbol{W}_{ij} = \sigma(i, j)$ if $(i, j) \in E$, and $\boldsymbol{W}_{ij} = 0$ otherwise.

According to the sign of edges, a signed graph can be divided into two spanning subgraphs: the positive subgraph $G^+ = (V, E^+, \sigma)$ and the negative subgraph $G^- = (V, E^-, \sigma)$. The two subgraphs share the same node set V. G^+ contains all positive edges of G while G^- includes all negative edges. Let $\boldsymbol{W}^+$ be the adjacency matrix of G^+. $\boldsymbol{W}^-$ is defined as $\boldsymbol{W}^- = \boldsymbol{W}^+ - \boldsymbol{W}$. Notably, $\boldsymbol{W}^-$ is non-negative. The signed degree matrix $\boldsymbol{D}$ is a diagonal matrix with entries $\boldsymbol{D}_{ii} = \sum_j |\boldsymbol{W}_{ij}|$. The signed Laplacian matrix is then given by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} = \boldsymbol{D} - \boldsymbol{W}^+ + \boldsymbol{W}^-$.

3.2 Signed Random Walk

Assume a signed random walk s starts from node i_0. At each step, it transitions to a neighboring node according to edge weights, regardless of whether the edge is positive or negative, forming a walk $s = (i_0, i_1, \cdots, i_t)$. It can be observed that the transition probabilities are identical to those in the unsigned case; therefore, we define the transition probability matrix as $\boldsymbol{P} = \boldsymbol{D}^{-1}(\boldsymbol{W}^+ + \boldsymbol{W}^-) = \boldsymbol{D}^{-1}|\boldsymbol{W}|$. Unlike the unsigned case, we also track the sign of s, where $sgn(s) = sign(\prod_{u=1}^{t} \sigma(i_{u-1}, i_u))$ [24, 33]. Here, $sign(x)$ is the sign function for any nonzero x: $sign(x) = +1$ for $x > 0$, and $sign(x) = -1$ for $x < 0$. The signed random walk s is termed positive if $sgn(s) = +1$, and negative if $sgn(s) = -1$. A positive (negative) walk [3, 8, 14] contains an even (odd) number of negative edges.

3.3 Label Propagation Operator on Signed Graph

In Graph Semi-Supervised Learning, the node set V is partitioned into two disjoint subsets: the labeled node set V_L and the unlabeled node set V_U, such that $n = |V_L| + |V_U|$. Without loss of generality, we assume the nodes are ordered such that the unlabeled nodes appear first. To begin, we assume the label is binary: $y \in \{-1, +1\}$. Let $\boldsymbol{y}_L \in \mathbb{R}^{|V_L|}$ be the label vector for V_L. The goal of GSSL is to learn a prediction $\boldsymbol{f} \in \mathbb{R}^n$, partitioned as $\boldsymbol{f} = [\boldsymbol{f}_U^\top, \boldsymbol{f}_L^\top]^\top$ to infer the labels of V_U.

In signed graphs, structural balance theory [3] dictates that nodes connected by positive edges should belong to the same class, whereas those linked by negative edges should possess opposite signs. Consequently, the problem reduces to minimizing the following energy function:

$$\min_{\boldsymbol{f}_U} E(\boldsymbol{f}) = \min_{\boldsymbol{f}_U} \frac{1}{2} \sum_{i,j} |\boldsymbol{W}_{ij}| \left(\boldsymbol{f}_i - \text{sgn}(\boldsymbol{W}_{ij})\boldsymbol{f}_j\right)^2 = \min_{\boldsymbol{f}_U} \boldsymbol{f}^\top \boldsymbol{L}\boldsymbol{f},$$

where $\boldsymbol{L}$ is the signed graph Laplacian. We find that this formulation is isomorphic in form to that of unsigned graphs. To derive the harmonic function, we partition the weight adjacency matrix $\boldsymbol{W}$ and the Laplacian matrix $\boldsymbol{L}$ into the following block forms:

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_{U,U} & \boldsymbol{W}_{U,L} \\ \boldsymbol{W}_{L,U} & \boldsymbol{W}_{L,L} \end{bmatrix}, \quad \boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_{U,U} & \boldsymbol{L}_{U,L} \\ \boldsymbol{L}_{L,U} & \boldsymbol{L}_{L,L} \end{bmatrix}.$$

Differentiating the energy function with respect to $\boldsymbol{f}_U$ and setting the derivative to zero yields the harmonic property: $\boldsymbol{L}_{U,U}\boldsymbol{f}_U + \boldsymbol{L}_{U,L}\boldsymbol{y}_L = 0$. Solving this linear system provides the closed-form solution for the unlabeled nodes: $\boldsymbol{f}_U = -\boldsymbol{L}_{U,U}^{-1}\boldsymbol{L}_{U,L}\boldsymbol{y}_L$.

Let $\boldsymbol{Q} = -\boldsymbol{L}_{U,U}^{-1}\boldsymbol{L}_{U,L}$ denote the label propagation operator. Directly computing $\boldsymbol{Q} \in \mathbb{R}^{|V_U|\times|V_L|}$ involves matrix inversion and matrix multiplication, resulting in a prohibitive computational complexity of $O(|V_U|^3 + |V_U|^2|V_L|)$, which is infeasible for large-scale graphs.

4 Multi-Tree Approximation Framework

Although the signed random walk method [52] is applicable, we observe that random walks starting from different nodes may traverse highly overlapping paths, leading to substantial redundant computations. On unsigned graphs, spanning tree based approaches effectively avoid such redundancy by capturing global structural dependencies in a compact form. As a result, they have been widely used to compute various graph-related quantities including personalized PageRank [26], effective resistance [25], and the Kemeny constant [46]. In this section, we propose a novel spanning tree sampling algorithm to address the above limitation.

4.1 Signed Multi-Rooted Spanning Tree

To accurately capture the physical interpretation of $\boldsymbol{Q}$ in signed graphs, we define the sample space over Signed Multi-Rooted Spanning Tree (SMRST). We first introduce some concepts.

Given a signed graph $G = (V, E, \sigma)$ with labeled nodes V_L and unlabeled nodes V_U, an SMRST ϕ rooted at V_L is a directed spanning subgraph of G where each node $i \in V_U$ has exactly one outgoing edge and $j \in V_L$ acts as a root with zero out-degree. Moreover, each connected component of ϕ contains exactly one labeled node, which is referred to as the root of that component. Unlike unsigned graphs, an SMRST permits the existence of negative cycles (i.e., cycles with a negative sign product), which will be discussed in detail in the following. We use $c(\phi)$ to denote the number of negative cycles in ϕ.

We use $\mathcal{F}(V_L)$ to denote the set of all valid SMRSTs rooted at V_L. $\mathcal{F}_{ij}(V_L) \subseteq \mathcal{F}(V_L)$ is the subset where i is connected to the labeled root $j \in V_L$ and i is isolated from any negative cycles. We define the function $r_\phi : V_U \to V_L$ in a specific SMRST ϕ as the root of node $i \in V_U$. Note that $r_\phi(i) = 0$ if i belongs to a cycle. We further define the weight of an SMRST ϕ as $W(\phi) = 2^{c(\phi)} \prod_{e\in\phi} |\boldsymbol{W}_e|$. For notational simplicity, we let $\mathcal{F}$ and $\mathcal{F}_{ij}$ denote $\mathcal{F}(V_L)$ and $\mathcal{F}_{ij}(V_L)$, respectively, in the following. Let $\mathcal{P}_{i\to j}$ denote the unique path from i to j within $\mathcal{F}_{ij}$. We define the sign of this path, denoted by $\text{sgn}(\mathcal{P}_{i\to j})$, as 1 if $\mathcal{P}_{i\to j}$ contains an even number of negative edges, and -1 if it contains an odd number. Figure 1 illustrates a signed graph along with all its SMRSTs.

Remark 4.1. *We first clarify the directionality of SMRSTs. Note that the SMRSTs defined in our model are strictly oriented. Consider an unoriented version with c negative cycles. Since each undirected cycle inherently admits exactly two distinct directed orientations (i.e., clockwise and counter-clockwise), a single unoriented version corresponds to a set of 2^c distinct SMRSTs.*

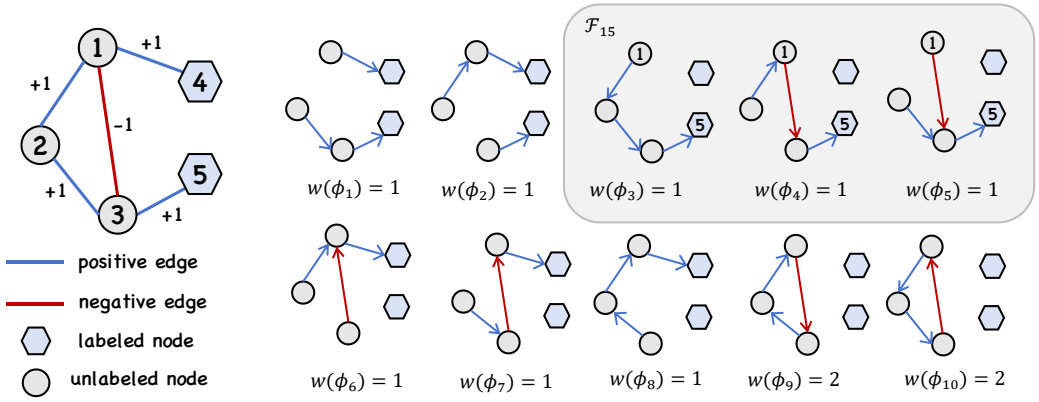
Since $\boldsymbol{L}_{U,L} = -\boldsymbol{W}_{U,L}$, we have $\boldsymbol{Q}_{ij} = \boldsymbol{e}_i^\top \boldsymbol{L}_{U,U}^{-1}\boldsymbol{W}_{U,L}\boldsymbol{e}_j = \sum_{k\in V_U}(\boldsymbol{L}_{U,U}^{-1})_{ik}\boldsymbol{W}_{kj}$. Applying Cramer's rule and leveraging the symmetry of the undirected Laplacian submatrix $\boldsymbol{L}_{U,U}$, the inverse matrix element can be expressed as $(\boldsymbol{L}_{U,U}^{-1})_{ik} = \frac{(-1)^{i+k}\det((\boldsymbol{L}_{U,U})_{-k,-i})}{\det(\boldsymbol{L}_{U,U})}$. Substituting this into the equation yields the foundational algebraic form:

$$\boldsymbol{Q}_{ij} = \frac{\sum_{k\in V_U}(-1)^{i+k}\det((\boldsymbol{L}_{U,U})_{-k,-i})\,\boldsymbol{W}_{kj}}{\det(\boldsymbol{L}_{U,U})} \tag{1}$$

To structurally interpret this algebraic form, we separately expand its denominator and numerator based on the SMRST framework (all missing proofs appear in Appendix C).

Theorem 4.2. *For a signed graph $G = (V, E, \sigma)$, the denominator in Equation* (1) *can be derived as:*

$$\det(\boldsymbol{L}_{U,U}) = \sum_{\phi\in\mathcal{F}} 2^{c(\phi)} \prod_{e\in\phi} |\boldsymbol{W}_e|. \tag{2}$$



(a) Signed Graph (b) All valid Signed Multi-Rooted Spanning Tree (SMRST) set $\mathcal{F}$

Figure 1: (a) A toy signed graph, where positive edges are marked in blue and negative edges in red. Nodes 1, 2, and 3 are unlabeled nodes, while nodes 4 and 5 are labeled nodes. (b) All valid SMRSTs. In each SMRST, nodes 4 and 5 serve as roots. For example, in ϕ_1, node 1 is rooted at node 4 and node 2, 3 are rooted at node 5. Moreover, ϕ_3, ϕ_4, ϕ_5 constitute the set $\mathcal{F}_{15}$, in which node 1 is rooted at node 5. The weights of ϕ_9 and ϕ_{10} are 2, as each contains exactly one negative cycle.

Similarly, the numerator can be evaluated by restricting the topological structure to SMRSTs that contain a directed path from the unlabeled node i to the labeled node j.

Theorem 4.3. *For a signed graph $G = (V, E, \sigma)$, the numerator in Equation* (1) *can be derived as:*

$$\sum_{k \in V_U} (-1)^{i+k} \det((\boldsymbol{L}_{U,U})_{-k,-i})\, \boldsymbol{W}_{kj} = \sum_{\phi \in \mathcal{F}_{ij}} sgn(\mathcal{P}_{i \to j}) 2^{c(\phi)} \prod_{e \in \phi} |\boldsymbol{W}_e|.$$

4.2 SNIS Estimator and Theoretical Guarantee

Thus, we establish the connection between the entries of matrix $\boldsymbol{Q}$ and SMRSTs. Based on this, the following theorem allows us to express the entry $\boldsymbol{Q}_{ij}$.

Theorem 4.4. *Given a signed graph $G = (V, E, \sigma)$, let V_L and V_U denote the labeled and unlabeled node sets, respectively. The matrix entry $\boldsymbol{Q}_{ij}$ for an unlabeled node $i \in V_U$ and a labeled node $j \in V_L$ is equivalent to:*

$$\boldsymbol{Q}_{ij} = \frac{\sum_{\phi \in \mathcal{F}} \mathbb{I}(r_\phi(i) = j) sgn(\mathcal{P}_{i \to j}) 2^{c(\phi)} \prod_{e \in \phi} |\boldsymbol{W}_e|}{\sum_{\phi \in \mathcal{F}} 2^{c(\phi)} \prod_{e \in \phi} |\boldsymbol{W}_e|}. \tag{3}$$

Proof. From the fundamental topological equivalence, we have established that:

$$\boldsymbol{Q}_{ij} = \frac{\sum_{\phi \in \mathcal{F}_{ij}} \text{sgn}(\mathcal{P}_{i \to j}) 2^{c(\phi)} \prod_{e \in \phi} |\boldsymbol{W}_e|}{\sum_{\phi \in \mathcal{F}} 2^{c(\phi)} \prod_{e \in \phi} |\boldsymbol{W}_e|}.$$

Recall that $\mathcal{F}_{ij}$ is the strictly defined subset of forests in $\mathcal{F}$ where the unlabeled node i is rooted at the specific labeled node j. We can naturally expand the summation in the numerator over the entire global forest space $\mathcal{F}$ by introducing an indicator function $\mathbb{I}(r_\phi(i) = j)$, which equals 1 if the root of i in forest ϕ is j, and 0 otherwise. Therefore, the restricted sum over $\mathcal{F}_{ij}$ is mathematically identical to the global sum weighted by this indicator function. This direct substitution yields Equation (3) and completes the proof. □

However, enumerating all SMRSTs in $\mathcal{F}$ is computationally impossible. This motivates us to approximate the result using a randomized sampling algorithm. We observe that the denominator in Equation (3) is essentially the sum of the weights of all valid SMRSTs in the graph. We can denote this global summation as a partition function $Z = \sum_{\phi \in \mathcal{F}} W(\phi)$.

By doing so, we naturally induce a target probability distribution $\mathbb{P}$ over the space $\mathcal{F}$. The probability of sampling a specific SMRST ϕ is strictly proportional to its weight: $\mathbb{P}(\phi) = \frac{W(\phi)}{Z}$. Furthermore,

we define a random variable $X_{ij}(\phi)$ to capture the path sign and label of node i in a given SMRST ϕ: $X_{ij}(\phi) = \mathbb{I}(r_\phi(i) = j)\text{sgn}(\mathcal{P}_{i\to j})$. Under this probabilistic framework, our original exact formulation in Equation (3) can be rewritten as a statistical expectation:

$$\boldsymbol{Q}_{ij} = \sum_{\phi\in\mathcal{F}} \mathbb{P}(\phi)\cdot X_{ij}(\phi) = \mathbb{E}_{\phi\sim\mathbb{P}}[X_{ij}(\phi)].$$

However, standard loop-erased random walks sample trees strictly proportional to their absolute edge weights $\prod_{e\in\phi}|\boldsymbol{W}_e|$ [20, 45, 32]. This omits the structural multiplier $2^{c(\phi)}$, leading to a systematically biased under-sampling of negative cycles. To correct this bias, we leverage importance sampling. We define the proposal distribution $\mathbb{P}_0$ over the space $\mathcal{F}$ based on edge weights as $\mathbb{P}_0(\phi) = \frac{1}{Z_0}\prod_{e\in\phi}|\boldsymbol{W}_e|$, where $Z_0 = \sum_{\phi\in\mathcal{F}}\prod_{e\in\phi}|\boldsymbol{W}_e|$ is the normalizing partition function. Then $\boldsymbol{Q}_{ij}$ can be reformulated as: $\boldsymbol{Q}_{ij} = \frac{\mathbb{E}_{\phi\sim\mathbb{P}_0}[X_{ij}(\phi)2^{c(\phi)}]}{\mathbb{E}_{\phi\sim\mathbb{P}_0}[2^{c(\phi)}]}$.

Then, we can construct the empirical Self-Normalized Importance Sampling (SNIS) estimator using l independent samples drawn from $\mathbb{P}_0$:

$$\widehat{\boldsymbol{Q}}_{ij} = \frac{\sum_{k=1}^{l} X_{ij}(\phi_k)2^{c(\phi_k)}}{\sum_{k=1}^{l} 2^{c(\phi_k)}}. \tag{4}$$

Let $c_{\max}$ denote the number of negative cycles in the graph. Crucially, $c(\phi)$ is strictly upper-bounded by $c_{\max}$ Then, the importance weights are bounded within $[1, 2^{c_{\max}}]$. This bounded property not only ensures asymptotic consistency but also allows us to establish a rigorous finite-sample error bound using concentration inequalities.

Theorem 4.5. *Let $\widehat{\boldsymbol{Q}}_{ij}$ be the SNIS estimator of $\boldsymbol{Q}_{ij}$ using l samples drawn from the proposal distribution $\mathbb{P}_0$. We define the maximum sample weight as $\alpha = 2^{c_{\max}}$ and the expected sample weight as $\beta = \mathbb{E}_{\phi\sim\mathbb{P}_0}[2^{c(\phi)}]$. For any error threshold $\epsilon > 0$ and confidence parameter $\delta \in (0,1)$, if the sample size l is chosen such that:$l \geq \left\lceil 2\left(\frac{\alpha}{\beta}\right)^2\left(\frac{\epsilon+2}{\epsilon}\right)^2\ln(4/\delta)\right\rceil$, then the approximation $\widehat{\boldsymbol{Q}}_{ij}$ of $\boldsymbol{Q}_{ij}$ satisfies the following relation with probability at least $1-\delta$: $\left|\widehat{\boldsymbol{Q}}_{ij} - \boldsymbol{Q}_{ij}\right| \leq \epsilon$.*

4.3 Rao-Blackwellized Variance Reduction

While the estimator $\widehat{\boldsymbol{Q}}_{ij}$ is asymptotically consistent, its variance can be significant in practice, particularly when labels are sparse. To address this, we propose a variance-reduced estimator by applying the Rao-Blackwell theorem. Specifically, let $T_i(\phi)$ denote the connected component containing node $i \in V_U$ in the forest obtained by removing all edges incident to V_L from ϕ. Instead of relying on the single absorption edge actually sampled, we compute the conditional expectation over all possible boundary edges.

Definition 4.6 (Boundary Set)**.** *For a given component $T_i \subseteq V_U$, let ∂T_i be the set of edges in the original graph G that connect any node $u \in T_i$ to any labeled node $j \in V_L$:*

$$\partial T_i = \{(u,j) \in E \mid u \in T_i, j \in V_L\}.$$

Given the internal structure T_i, the probability of being absorbed via a specific edge $(u,j) \in \partial T_i$ is proportional to its absolute weight $|\boldsymbol{W}_{uj}|$. Thus, the Rao-Blackwellized estimator for the k-th sample is formulated as:

$$\widehat{\boldsymbol{Q}}^{(k)}_{ij,RB} = \frac{\sum_{(u,j')\in\partial T_i}|\boldsymbol{W}_{uj'}|\cdot\text{sgn}(\mathcal{P}_{i\to u})\cdot\text{sgn}(\boldsymbol{W}uj')\cdot\mathbb{I}(j'=j)}{\sum_{(u,j')\in\partial T_i}|\boldsymbol{W}_{uj'}|}\cdot 2^{c(\phi_k)}.$$

By noticing that $|\boldsymbol{W}_{uj}|\cdot\text{sgn}(\boldsymbol{W}_{uj}) = \boldsymbol{W}_{uj}$, the estimator simplifies to:

$$\widehat{\boldsymbol{Q}}^{(k)}_{ij,RB} = \frac{\sum_{u\in T_i}\text{sgn}(\mathcal{P}_{i\to u})\boldsymbol{W}_{uj}}{\sum_{u\in T_i}\sum_{j'\in V_L}|\boldsymbol{W}_{uj'}|}\cdot 2^{c(\phi_k)}.$$

Theorem 4.7. *The Rao-Blackwellized estimator $\widehat{\boldsymbol{Q}}_{ij,RB}$ has the same expectation as the original SNIS estimator $\widehat{\boldsymbol{Q}}_{ij}$, satisfying $\mathbb{E}[\widehat{\boldsymbol{Q}}_{ij,RB}] = \mathbb{E}[\widehat{\boldsymbol{Q}}_{ij}]$. Moreover, its variance is no greater than that of the original estimator:* $\text{Var}(\widehat{\boldsymbol{Q}}_{ij,RB}) \leq \text{Var}(\widehat{\boldsymbol{Q}}_{ij})$.

Algorithm 1: MultiTreeQ($\mathcal{G}, \epsilon$)

Input : Signed graph $G = (V, E, \sigma)$, labeled set V_L, unlabeled set V_U, error hyperparameter ϵ

Output : Approximation matrix $\widehat{\boldsymbol{Q}}$

Initialize $\widehat{\boldsymbol{Q}} \leftarrow \mathbf{O}^{|V_U|\times|V_L|}$, $W_{\text{total}} \leftarrow 0, l \leftarrow \left\lceil 2(\alpha/\beta)^2 \left(\frac{\epsilon+2}{\epsilon}\right)^2 \ln(4/\delta) \right\rceil$

foreach $u \in V_U$ **do**

  $D[u] \leftarrow \sum_{j\in V_L} |W_{uj}|$

for $k = 1, 2, \cdots, l$ **do**

  $c(\phi) \leftarrow 0, T \leftarrow V_L$, $root[v] \leftarrow -1$ for $v \in V_U$

  foreach $u \in V_U \setminus T$ **do**

    $path \leftarrow \{u\}$

    while $u \notin T$ **do**

      $next \leftarrow$ a randomly selected neighbor of u based on $|W|$

      if $next \in path$ **then**

        $\mathcal{C} \leftarrow$ cycle formed by $next$

        if $\text{sgn}(\mathcal{C}) = 1$ **then**

          Erase $\mathcal{C}$ from $path$, $u \leftarrow next$

        else

          $c(\phi) \leftarrow c(\phi) + 1, T \leftarrow T \cup path$, $root[path] \leftarrow \text{null}$

          break

      else

        Append $next$ to $path$, $u \leftarrow next$

        if $u \in T$ **then**

          $root[path] \leftarrow root[u], T \leftarrow T \cup path$

  $w_k \leftarrow 2^{c(\phi)}, W_{\text{total}} \leftarrow W_{\text{total}} + w_k$

  foreach *valid tree branch* $T_c \subset \phi$ **do**

    $r \leftarrow$ an arbitrary reference node in T_c

    $S(u) \leftarrow \text{sgn}(\mathcal{P}_{r\to u}), \quad \forall u \in T_c$

    $D_{T_c} \leftarrow \sum_{u\in T_c} D[u]$

    foreach $j \in V_L$ *adjacent to* T_c **do**

      $N_{T_c,j} \leftarrow \sum_{u\in T_c} S(u) W_{uj}$

      foreach $i \in T_c$ **do**

        $\widehat{\boldsymbol{Q}}_{ij} \leftarrow \widehat{\boldsymbol{Q}}_{ij} + S(i)\frac{N_{T_c,j}}{D_{T_c}} w_k$

$\widehat{\boldsymbol{Q}} \leftarrow \widehat{\boldsymbol{Q}}/W_{\text{total}}$

return $\widehat{\boldsymbol{Q}}$

Proof. Let $\mathcal{G}$ denote the internal forest structures and negative-cycle information of all sampled SMRSTs, excluding their boundary-edge choices. Conditioned on $\mathcal{G}$, the importance weights $2^{c(\phi_k)}$ are fixed. Since the Rao-Blackwellized estimator averages over the possible boundary-edge choices, we have $\mathbb{E}[\widehat{\boldsymbol{Q}}_{ij} \mid \mathcal{G}] = \widehat{\boldsymbol{Q}}_{ij,RB}$. By the Law of Total Expectation, we obtain $\mathbb{E}[\widehat{\boldsymbol{Q}}_{ij}] = \mathbb{E}[\mathbb{E}[\widehat{\boldsymbol{Q}}_{ij} \mid \mathcal{G}]] = \mathbb{E}[\widehat{\boldsymbol{Q}}_{ij,RB}]$. The variance reduction follows directly from the Law of Total Variance: $\text{Var}(\widehat{\boldsymbol{Q}}_{ij}) = \mathbb{E}[\text{Var}(\widehat{\boldsymbol{Q}}_{ij} \mid \mathcal{G})] + \text{Var}(\mathbb{E}[\widehat{\boldsymbol{Q}}_{ij} \mid \mathcal{G}]) = \mathbb{E}[\text{Var}(\widehat{\boldsymbol{Q}}_{ij} \mid \mathcal{G})] + \text{Var}(\widehat{\boldsymbol{Q}}_{ij,RB}) \geq \text{Var}(\widehat{\boldsymbol{Q}}_{ij,RB})$, where equality holds if and only if $\text{Var}(\widehat{\boldsymbol{Q}}_{ij} \mid \mathcal{G}) = 0$ almost surely. □

Furthermore, since the error bounds in Theorem 4.5 derived via Hoeffding's inequality depend exclusively on the bounded range of the estimator rather than its variance, the theoretical error guarantees established for the original estimator remain strictly valid for the Rao-Blackwellized estimator $\widehat{\boldsymbol{Q}}_{ij,RB}$. In practice, the boundary terms $\boldsymbol{W}_{uj}$ and $\sum_{j'\in V_L} |\boldsymbol{W}_{uj'}|$ can be precomputed for all $u \in V_U$ in $O(|E|)$ time. Consequently, this Rao-Blackwellized approach significantly accelerates the convergence of the Monte Carlo process with negligible computational overhead.

4.4 Algorithm Design and Analysis

We then proceed to sample an SMRST according to the proposed distribution $\mathbb{P}_0$, and record the number of negative cycles as well as the sign of the path from node i to node j. Wilson's algorithm and its extension provide a method for uniformly sampling spanning trees [47, 27] and forests [42, 39, 41, 43] based on the loop-erased random walk. In our setting, however, negative cycles need to be preserved; therefore, we only erase loops when they are positive. Specifically, we sample an SMRST as follows:

- Initialize all nodes in V_L as absorbing states.
- For each non-absorbed node, perform a random walk according to edge weights. Along the walk, erase any encountered positive cycles, while preserving and counting negative cycles. If the walk reaches an absorbing node or a node on a negative cycle, all nodes along this walk are marked as absorbed. Repeat this process until all nodes are absorbed.

Algorithm 1 named MultiTreeQ presents our variance-reduced SMRST sampling method to efficiently estimate the entries of $\boldsymbol{Q}$. First, it precomputes the local boundary capacities for all unlabeled nodes (Lines 2 to 3), effectively marginalizing the final absorption step. Next, it samples an internal SMRST rooted at V_L based on the absolute edge weights (Lines 4 to 20). During this structural sampling phase, any positive cycles are erased (Lines 12 to 13), whereas negative cycles immediately terminate the walk and are tallied (Lines 14 to 16). Finally, instead of relying on a single sampled boundary edge, MultiTreeQ executes the Rao-Blackwellized aggregation (Lines 21 to 29). It aggregates the boundary influence signals across each intact tree branch and updates the corresponding entries in the operator matrix using the conditional expectation. We then proceed to analyze its computational efficiency, formally establishing the expected running time in Theorem 4.8.

Theorem 4.8. *Given a signed graph $G = (V, E, \sigma)$ with labeled nodes V_L and unlabeled nodes V_U. Let $\alpha = 2^{c_{\max}}$ and $\beta = \mathbb{E}[2^{c(\phi)}]$. If the error parameter ϵ is determined, then the expected time complexity of Algorithm 1 is bounded by $O(|E| + T)$, where the total sampling time T is:*

$$T = 2\left(\frac{\alpha}{\beta}\right)^2 \left(\frac{\epsilon+2}{\epsilon}\right)^2 \ln(4/\delta) \cdot \mathrm{Tr}\left((\boldsymbol{I} - \boldsymbol{P}_{UU})^{-1}\right).$$

5 Experiments

In this section, we conduct extensive experiments on various datasets to evaluate the accuracy, efficiency, and scalability of the two proposed algorithms. The hyperparameter configurations and downstream applications are provided in Appendix E.

5.1 Experimental Setup

Datasets. We conduct experiments on real-world networks publicly available from the Koblenz Network Collection [18], the SNAP repository [22] and Hugging Face [35]. We utilize both original signed graphs and modified signed graphs for our experiments. The modified signed graphs (marked with *) are generated from real unsigned graphs by randomly assigning a negative sign to each edge with a probability of 0.15. The statistics of all datasets are summarized in Table 1 and 2.

Equipment. All experiments were conducted on a Linux server with 36-core 2.10GHz Intel(R) Xeon(R) Platinum 8352V CPU and 256GB of RAM. Our algorithms and baselines are implemented in Julia. Given that our sampling algorithms and the CG algorithm can be parallelized efficiently, we use 72 computing cores to speed up the process.

Algorithms. Our proposed algorithm is MultiTreeQ (MTQ). To validate the effectiveness of the Rao-Blackwellization strategy, we evaluate its ablated version without Rao-Blackwellization ($\text{MTQ}_{\text{w/o-RB}}$). For baselines, we employ the state-of-the-art method based on signed random walks (RWQ) [52], alongside its variation without truncation ($\text{RWQ}_{\text{w/o-T}}$) to assess the impact of the truncation technique. To establish the ground truth, we utilize direct matrix inversion (Exact) for small datasets and the Conjugate Gradient (CG) method for medium datasets. For large-scale networks, obtaining exact values is computationally intractable due to time and memory constraints.

5.2 Accuracy Evaluation

We first evaluate the accuracy of our proposed algorithms on small and medium datasets. Specifically, we adopt the mean absolute error as the evaluation metric, and conduct experiments on seven small-scale datasets, each containing fewer than one million nodes. For each dataset, we run the experiments five times and report the mean and standard deviation. The results are presented in Table 1.

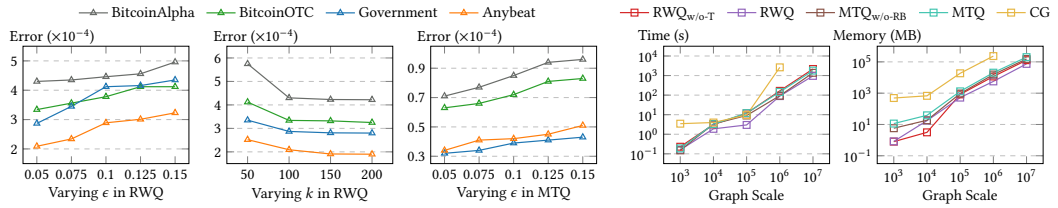
Table 1: Mean absolute error ($\times 10^{-4}$) of our algorithms on six small datasets.

Dataset	Node	Edge	$\text{RWQ}_{\text{w/o-T}}$	RWQ	$\text{MTQ}_{\text{w/o-RB}}$	MTQ
BitcoinAlpha	3,783	24,186	3.90±0.03	4.30±0.02	4.25±0.08	**0.96±0.01**
BitcoinOTC	5,881	35,592	3.24±0.04	3.34±0.04	3.38±0.02	**0.83±0.01**
Government*	7,057	89,455	2.76±0.03	2.87±0.03	2.89±0.01	**0.43±0.00**
WikiRfa	11,017	170,335	2.56±0.01	2.82±0.01	2.77±0.03	**0.48±0.00**
Anybeat*	12,645	49,132	2.00±0.02	2.09±0.02	2.05±0.01	**0.51±0.01**
Slashdot	82,140	549,202	2.11±0.01	2.52±0.02	2.51±0.02	**0.76±0.01**
Epinions	131,828	841,372	2.03±0.01	2.46±0.01	2.28±0.02	**0.42±0.01**

As shown in Table 1, across all datasets, our algorithm MTQ achieves the lowest approximation errors (below 10^{-4}), validating the effectiveness of the proposed estimator. This accuracy is primarily attributed to the implemented variance reduction technique. Following MTQ, $\text{RWQ}_{\text{w/o-T}}$ generally exhibits lower errors than $\text{MTQ}_{\text{w/o-RB}}$, as $\text{RWQ}_{\text{w/o-T}}$ functions as an unbiased estimator. Furthermore, we observe that the truncation strategy in RWQ introduces additional approximation error, which is particularly evident in specific datasets (Slashdot and Epinions). Furthermore, among all evaluated estimators, MTQ consistently exhibits the lowest variance, directly demonstrating the efficacy of our variance reduction technique.

5.3 Hyperparameter Sensitivity Analysis

In this subsection, we conduct a detailed sensitivity analysis of the key hyperparameters to investigate their impact on our algorithms. In our methods, these hyperparameters include the error tolerance ϵ in RWQ and MTQ, as well as the truncation length k in RWQ. Figure 2 illustrates the performance of our methods under different hyperparameter settings. We observe that decreasing ϵ improves accuracy due to the corresponding increase in sampling rounds. Additionally, increasing k initially reduces the estimation error, but this effect becomes negligible when $k > 150$. This indicates that almost all random walks on these graphs are absorbed within 150 steps.



(a) Hyperparameter Analysis. (b) Scalability Analysis.

Figure 2: (a) Mean absolute error ($\times 10^{-4}$) with respect to the parameters ϵ and k on four datasets. (b) Running time (s) and memory consumption (MB) with respect to the graph scale on synthetic signed graphs.

Furthermore, we performed a comprehensive trade-off analysis by varying the hyperparameter configurations and recording the corresponding changes in error and running time, as depicted in Figure 3. As observed, the baseline and ablated methods exhibit comparable performance profiles. In contrast, our proposed MTQ achieves a substantial reduction in approximation error. While this significant accuracy improvement incurs a slight increase in computational overhead, the additional running time is marginal. Overall, MTQ demonstrates a highly favorable trade-off, delivering superior effectiveness at a highly acceptable computational cost.

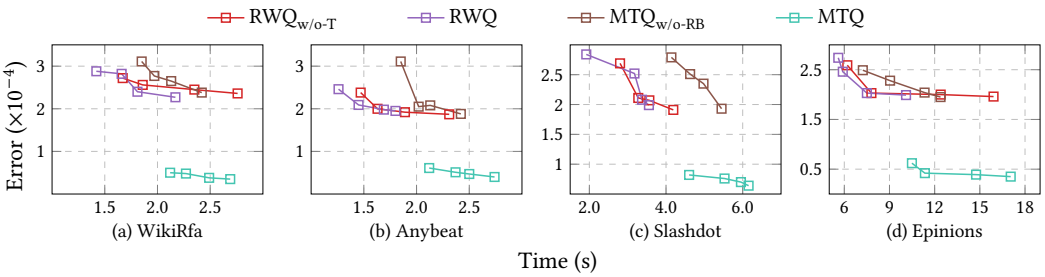


Figure 3: Scatter plots of mean absolute error ($\times 10^{-4}$) and total running time (s) under different parameter settings for our algorithms on four datasets.

5.4 Efficiency and Scalability

In this section, we analyze the efficiency and scalability of our algorithm. Table 2 illustrates the running time of our proposed algorithm across five large-scale datasets. We observe that RWQ$_{w/o-T}$ exhibits relatively slower performance. This is primarily because large-scale graphs often possess larger network diameters, which significantly decreases the absorption efficiency of standard random walks. However, our proposed truncation strategy effectively mitigates this issue, allowing the RWQ to achieve substantially faster execution times. Regarding the MTQ$_{w/o-RB}$ algorithm, the process of sampling the SMRST inherently acts as a mechanism for reusing random walk trajectories, which reduces the overall time consumption. Finally, the Rao-Blackwellized MTQ incurs a slight overhead for traversing sampled branches and aggregating boundary expectations.

Table 2: Running time (s) of our algorithms on five large datasets.

Dataset	Node	Edge	RWQ$_{w/o-T}$	RWQ	MTQ$_{w/o-RB}$	MTQ
TwitterSG	753,944	12,848,093	114.3	63.29	61.13	92.06
Youtube*	1,134,890	2,987,624	186.8	96.79	93.78	147.3
USPatents*	3,774,768	16,518,947	817.2	497.3	270.2	381.5
Dblp*	7,577,304	12,282,059	1678	892.5	823.6	1026
Livejournal*	10,690,276	112,307,385	3078	2145	2031	2540

To evaluate scalability, we generate synthetic signed graphs with varying node counts $|V|$ ranging from 10^3 to 10^7. The number of edges is strictly fixed at $|E| = 5|V|$. We then record the time and memory consumption of each algorithm as the graph size increases. As illustrated in Figure 2 (b), the CG baseline exhibits poor scalability. When the node count reaches 10^6, CG requires approximately 1,341 seconds and 235GB memory to converge, and it fails to process larger graphs due to prohibitive time and space overhead. In contrast, both the time and memory costs of our proposed RWQ and MTQ algorithms scale linearly with the graph size, firmly confirming our theoretical complexity analysis and demonstrating their capability to efficiently process massive networks.

6 Conclusion

This paper investigated the evaluation of the label propagation operator $\boldsymbol{Q}$ on signed graphs. We proposed two algorithms, RandomWalkQ and MultiTreeQ, which leverage truncation and Rao-Blackwellized sampling over the SMRST framework. Experiments confirm that our methods match the accuracy of numerical baselines while significantly improving efficiency and scalability on large-scale networks.

A current limitation is that the efficiency of our Rao-Blackwellized aggregation depends on the size of the labeled set V_L, which may pose challenges for datasets with an exceptionally large number of labels. Future work could address this by developing hierarchical sampling or grouping strategies to maintain performance. Additionally, extending these algorithms to dynamic signed graphs and higher-order structures like hypergraphs remains a promising direction.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62372112).

References

[1] Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *2006 47th annual IEEE symposium on foundations of computer science*, pages 475–486. IEEE, 2006.

[2] Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *International conference on computational learning theory*, pages 624–638. Springer, 2004.

[3] Dorwin Cartwright and Frank Harary. Structural balance: a generalization of heider's theory. *Psychological review*, 63(5):277, 1956.

[4] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[5] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International conference on machine learning*, pages 1725–1735. PMLR, 2020.

[6] Michael B Cohen, Rasmus Kyng, Gary L Miller, Jakub W Pachocki, Richard Peng, Anup B Rao, and Shen Chen Xu. Solving sdd linear systems in nearly $m \log^{1/2} n$ time. In *Proceedings of the 46th annual ACM symposium on Theory of computing*, pages 343–352, 2014.

[7] Morris H DeGroot. Reaching a consensus. *Journal of the American Statistical association*, 69(345):118–121, 1974.

[8] Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining*, pages 929–934. IEEE, 2018.

[9] Yucheng Dong, Min Zhan, Gang Kou, Zhaogang Ding, and Haiming Liang. A survey on the fusion process in opinion dynamics. *Information Fusion*, 43:57–65, 2018.

[10] Peter G Doyle and J Laurie Snell. *Random walks and electric networks*, volume 22. American Mathematical Soc., 1984.

[11] Albert M Erisman and William F Tinney. On computing certain elements of the inverse of a sparse matrix. *Communications of the ACM*, 18(3):177–179, 1975.

[12] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pages 409–426, 1994.

[13] Yaoping Hou, Jiongsheng Li, and Yongliang Pan. On the laplacian eigenvalues of signed graphs. *Linear and Multilinear Algebra*, 51(1):21–30, 2003.

[14] Mohammad Raihanul Islam, B Aditya Prakash, and Naren Ramakrishnan. Signet: Scalable embeddings for signed networks. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 157–169. Springer, 2018.

[15] Peng Jia, Anahita MirTabatabaei, Noah E Friedkin, and Francesco Bullo. Opinion dynamics and the evolution of social power in influence networks. *SIAM review*, 57(3):367–397, 2015.

[16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[17] Ioannis Koutis, Gary L Miller, and Richard Peng. A nearly-m log n time solver for sdd linear systems. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 590–598. IEEE, 2011.

[18] Jérôme Kunegis. Konect: the koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1343–1350, 2013.

[19] Hans Künsch. Gaussian markov random fields. *J. Fac. Sci. Univ. Tokyo Sect. IA Math*, 26(1):53–73, 1979.

[20] Gregory F. Lawler. A self-avoiding random walk. *Duke Mathematical Journal*, 47:655–693, 1980.

[21] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370, 2010.

[22] Jure Leskovec and Rok Sosič. SNAP: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology*, 8(1):1–20, 2016.

[23] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[24] Yanhua Li, Wei Chen, Yajun Wang, and Zhi-Li Zhang. Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In *Proceedings of the 6th ACM international conference on Web search and data mining*, pages 657–666, 2013.

[25] Meihao Liao, Rong-Hua Li, Qiangqiang Dai, Hongyang Chen, Hongchao Qin, and Guoren Wang. Efficient resistance distance computation: The power of landmark-based approaches. *Proceedings of the ACM on Management of Data*, 1(1):1–27, 2023.

[26] Meihao Liao, Rong-Hua Li, Qiangqiang Dai, and Guoren Wang. Efficient personalized pagerank computation: A spanning forests sampling based approach. In *Proceedings of the 2022 International Conference on Management of Data*, pages 2048–2061, 2022.

[27] Meihao Liao, Junjie Zhou, Rong-Hua Li, Qiangqiang Dai, Hongyang Chen, and Guoren Wang. Efficient and provable effective resistance computation on large graphs: An index-based approach. *Proceedings of the ACM on Management of Data*, 2(3):1–27, 2024.

[28] Lin Lin, Chao Yang, Juan C Meza, Jianfeng Lu, Lexing Ying, and Weinan E. Selinv—an algorithm for selected inversion of a sparse symmetric matrix. *ACM Transactions on Mathematical Software*, 37(4):1–19, 2011.

[29] Yang Liu, Jingfa Li, Shuyu Sun, and Bo Yu. Advances in gaussian random field generation: a review. *Computational Geosciences*, 23(5):1011–1047, 2019.

[30] László Lovász. Random walks on graphs. *Combinatorics, Paul erdos is eighty*, 2(1-46):4, 1993.

[31] Yao Ma and Jiliang Tang. *Deep learning on graphs*. Cambridge University Press, 2021.

[32] Philippe Marchal. Loop-erased random walks, spanning trees and hamiltonian cycles. *Electronic Communications in Probability*, 2000.

[33] Stefan Neumann and Pan Peng. Sublinear-time clustering oracle for signed graphs. In *International Conference on Machine Learning*, pages 16496–16528. PMLR, 2022.

[34] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[35] John Pougué-Biyong, Akshay Gupta, Aria Haghighi, and Ahmed El-Kishky. Learning stance embeddings from signed social graphs. *arXiv preprint arXiv:2201.11675*, 2022.

[36] Anton V Proskurnikov and Roberto Tempo. A tutorial on modeling and analysis of dynamic social networks. part i. *Annual Reviews in Control*, 43:65–79, 2017.

[37] 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.

[38] 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.

[39] Haoxin Sun, Yubo Sun, Xiaotian Zhou, and Zhongzhi Zhang. Fast computation and optimization for opinion-based quantities of Friedkin-Johnsen model. In *Proceedings of the 39th Conference on Neural Information Processing Systems*, volume 38, pages 42450–42472, 2025.

[40] Renjie Sun, Chen Chen, Xiaoyang Wang, Ying Zhang, and Xun Wang. Stable community detection in signed social networks. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):5051–5055, 2020.

[41] Yubo Sun, Haoxin Sun, Ahad N. Zehmakan, and Zhongzhi Zhang. Efficient approximation algorithms for resolvent subgraph centrality on signed graphs. *IEEE Transactions on Knowledge and Data Engineering*, 38(11):7416–7429, 2026.

[42] Yubo Sun, Haoxin Sun, and Zhongzhi Zhang. Scalable algorithms for forest-based centrality on large graphs. In *Proceedings of the ACM on Web Conference 2025*, pages 1331–1341, 2025.

[43] Yubo Sun, Haoxin Sun, and Zhongzhi Zhang. SigFJProp: Lightweight and scalable signed graph learning via opinion dynamics. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4544–4555, 2026.

[44] Ruo-Chun Tzeng, Bruno Ordozgoiti, and Aristides Gionis. Discovering conflicting groups in signed networks. In *Proceedings of the 34th Conference on Neural Information Processing Systems*, volume 33, pages 10974–10985, 2020.

[45] David Bruce Wilson. Generating random spanning trees more quickly than the cover time. In *Proceedings of the 28th annual ACM symposium on Theory of computing*, pages 296–303, 1996.

[46] Haisong Xia and Zhongzhi Zhang. Efficient approximation of Kemeny's constant for large graphs. *Proceedings of the ACM on Management of Data*, 2(3):1–26, 2024.

[47] Haisong Xia and Zhongzhi Zhang. Fast computation of Kemeny's constant for directed graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3472–3483, 2024.

[48] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Proceedings of the 17th Conference on Neural Information Processing Systems*, volume 16, 2003.

[49] Xiaotian Zhou, Haoxin Sun, Wanyue Xu, Wei Li, and Zhongzhi Zhang. Friedkin-Johnsen model for opinion dynamics on signed graphs. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[50] Xiaotian Zhou and Zhongzhi Zhang. Maximizing influence of leaders in social networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2400–2408, 2021.

[51] Xiaotian Zhou, Liwang Zhu, Wei Li, and Zhongzhi Zhang. A sublinear time algorithm for opinion optimization in directed social networks via edge recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3593–3602, 2023.

[52] Xinna Zhou, Xiaotian Zhou, Yuan Li, and Zhongzhi Zhang. Fast leader selection for opinion maximization in signed graphs. *Theoretical Computer Science*, page 115817, 2026.

[53] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on machine learning*, pages 912–919, 2003.

A Notations

We use $\mathbb{R}$ to represent the set of real numbers and regular lowercase letters like a, b, c to represent scalars in $\mathbb{R}$. Vectors are denoted with bold lowercase letters such as $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and matrices with bold uppercase letters like $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$. To denote specific elements in vectors and matrices, we use $\boldsymbol{a}_i$ to represent the ith element of vector $\boldsymbol{a}$ and use $\boldsymbol{A}_{ij}$ or $\boldsymbol{A}_{i,j}$ to represent the entry at position (i, j) in matrix $\boldsymbol{A}$. We denote submatrices by writing scalars in subscripts. For instance, we use $\boldsymbol{A}_{U,V}$ to denote the submatrix of $\boldsymbol{A}$ induced by row indices U and column indices V. In particular, we use the shorthand $\boldsymbol{A}_{-u,-v}$ to represent the submatrix obtained by removing the u-th row and v-th column. We use $\boldsymbol{a}^\top$ and $\boldsymbol{A}^\top$ to denote, respectively, transpose of vector $\boldsymbol{a}$ and matrix $\boldsymbol{A}$. And we use $\mathbf{1}$ to denote the vector of appropriate dimensions with all entries being ones and use $\boldsymbol{e}_i$ to represent the ith standard basis vector of appropriate dimensions. $\boldsymbol{I}$ denotes the identity matrix of appropriate dimensions.

B Proofs

B.1 Proof of Theorem 4.2

Proof. We first construct the incidence matrix $\boldsymbol{B} \in \mathbb{R}^{|V|\times|E|}$. For each edge $e = (u, v)$, the e-th column in $\boldsymbol{B}$ has exactly two non-zero entries at rows u and v: for a positive edge ($\boldsymbol{W}_{uv} > 0$), the entries are $+1$ and -1; for a negative edge ($\boldsymbol{W}_{uv} < 0$), the entries take identical signs (e.g., both $+1$). Letting $|\boldsymbol{W}_e|$ be the diagonal matrix of absolute edge weights, this specific sign assignment rigorously ensures the fundamental decomposition $\boldsymbol{L} = \boldsymbol{B}|\boldsymbol{W}_e|\boldsymbol{B}^\top$.

By partitioning the node set into V_L and V_U, we can decompose the incidence matrix into row blocks $\boldsymbol{B} = [\boldsymbol{B}_L^\top, \boldsymbol{B}_U^\top]^\top$. Consequently, the Laplacian submatrix corresponding to the unlabeled nodes naturally inherits the decomposition $\boldsymbol{L}_{U,U} = \boldsymbol{B}_U|\boldsymbol{W}_e|\boldsymbol{B}_U^\top$. Noting that a column in $\boldsymbol{B}_U$ retains exactly two non-zero entries if and only if its corresponding edge connects two unlabeled nodes within V_U, whereas edges incident to V_L are reduced to a single non-zero entry.

Applying the Cauchy-Binet formula, the determinant expands to the sum of squared determinants of square submatrices:

$$\det(\boldsymbol{L}_{U,U}) = \sum_S \det(\boldsymbol{B}_{U,S})^2 \prod_{e\in S} |\boldsymbol{W}_e|. \tag{5}$$

The edge set S induces a subgraph $G_S = (V_U, S)$. Let $\{C_1, C_2, \ldots, C_k\}$ be the weakly connected components of G_S. By applying appropriate permutation matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$, we can transform $\boldsymbol{B}_{U,S}$ into a block diagonal form:

$$\boldsymbol{H}_1\boldsymbol{B}_{U,S}\boldsymbol{H}_2 = \text{diag}(\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_k),$$

where each block $\boldsymbol{M}_i$ corresponds to the incidence relations within component C_i. Since $|\det(\boldsymbol{H}_1)| = |\det(\boldsymbol{H}_2)| = 1$, we have:

$$\det(\boldsymbol{B}_{U,S})^2 = \prod_{i=1}^{k} \det(\boldsymbol{M}_i)^2.$$

For $\det(\boldsymbol{B}_{U,S}) \neq 0$, each component $\boldsymbol{M}_i$ must be a square matrix of size $n_i \times n_i$. This implies each component has an equal number of nodes and edges ($|E(C_i)| = |V(C_i)|$).

If a component C_i contains a vertex v of degree 1 incident to edge e, the row for v in $\boldsymbol{M}_i$ has only one non-zero entry $\boldsymbol{M}_{v,e} = \pm 1$. Expanding the determinant along this row:

$$|\det(\boldsymbol{M}_i)| = |\boldsymbol{M}_{v,e}| \cdot |\det((\boldsymbol{M}_i)_{-v,-e})| = 1 \cdot |\det(((\boldsymbol{M}_i)_{-v,-e})|$$

Repeated application of this "pruning" removes all pendant trees. A non-singular $\boldsymbol{M}_i$ must reduce to either: a) An empty set (if C_i is a tree rooted at V_L): $|\det(\boldsymbol{M}_i)| = 1$ or b) A cycle core (if C_i is unicyclic within V_U): $|\det(\boldsymbol{M}_i)| = |\det(\boldsymbol{M}_{\text{cycle}})|$.

Let $\mathbf{M}_{\text{cycle}}$ be the $p \times p$ incidence matrix of a simple cycle. After row permutations, it can be expressed as:

$$\boldsymbol{M}_{\text{cycle}} = \begin{bmatrix} 1 & 0 & \dots & s \\ -1 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & -1 & 1 \end{bmatrix},$$

where $s = -\prod_{e \in \text{cycle}} \sigma(e)$. We apply Gaussian elimination by adding row j to row $j+1$ for $j = 1 \dots p-1$. The resulting upper triangular matrix has diagonal elements $\{1, 1, \dots, 1, 1+s\}$. Therefore,

$$\det(\boldsymbol{M}_{\text{cycle}}) = 1 + s = \begin{cases} 2, & \text{if the cycle is negative,} \\ 0, & \text{if the cycle is positive.} \end{cases}$$

For a selection S to satisfy $\det(\boldsymbol{B}_{U,S}) \neq 0$, every component must be either a rooted tree or a negative cycle. Then:

$$\det(\boldsymbol{B}_{U,S})^2 = \prod_{i \in \text{negative cycle}} 2^2 \cdot \prod_{j \in \text{Rooted tree}} 1^2 = 4^{c(S)}.$$

According to Remark 4.1, we obtain

$$\sum_S \det(\boldsymbol{B}_{U,S})^2 = \sum_S 4^{c(S)} = \sum_S 2^{c(S)} 2^{c(S)} = \sum_{\phi \in \mathcal{F}} 2^{c(\phi)}. \tag{6}$$

Combining Equation (5) and (6), we complete the proof. □

B.2 Proof of Theorem 4.3

Proof. Let $C_{ik} = (-1)^{i+k} \det((\boldsymbol{L}_{U,U})_{-k,-i})$ be the algebraic cofactor. Since $\boldsymbol{L}_{U,U} = \boldsymbol{B}_U |\boldsymbol{W}_e| \boldsymbol{B}_U^\top$, we can apply the Cauchy-Binet formula to the sub-matrix with row k and column i removed. For a selected edge set $S' \subset E$ with $|S'| = n-1$, we define two $(n-1) \times (n-1)$ sub-matrices of the incidence matrix:

- $\boldsymbol{B}_R = (\boldsymbol{B}_U)_{-k,S'}$: The incidence matrix with row k removed.
- $\boldsymbol{B}_C = (\boldsymbol{B}_U)_{-i,S'}$: The incidence matrix with row i removed.

The cofactor can be expanded as:

$$C_{ik} = \sum_{S', |S'|=n-1} (-1)^{i+k} \det(\boldsymbol{B}_R) \det(\boldsymbol{B}_C) \prod_{e \in S'} |\boldsymbol{W}_e|.$$

The non-vanishing of $\det(\boldsymbol{B}_R) \det(\boldsymbol{B}_C)$ necessitates the full column rank condition $\text{rank}(\boldsymbol{B}_R) = \text{rank}(\boldsymbol{B}_C) = n-1$. Topologically, this rank constraint dictates that S must be a forest devoid of positive cycles. Furthermore, nodes i and k must strictly belong to the same tree component $T_{ik} \subseteq S$; otherwise, the respective row deletions would yield an isolated component with a singular matrix block. Thus, S uniquely decomposes into T_{ik} and a set of independent frustrated unicyclic components U. Up to row and column permutations, the sub-matrices block-diagonalize as:

$$\boldsymbol{B}_R = \boldsymbol{M}_{R,T_{ik}} \oplus \boldsymbol{M}_U, \quad \boldsymbol{B}_C = \boldsymbol{M}_{C,T_{ik}} \oplus \boldsymbol{M}_U,$$

where $\oplus$ denotes the direct sum operation. Since the unicyclic block $\boldsymbol{M}_U$ is topologically isolated from nodes i and k, it remains identical in both matrices. By the Cycle Determinant Lemma, each negative cycle yields an absolute determinant of 2, contributing to the product as:

$$\det(\boldsymbol{B}_R) \det(\boldsymbol{B}_C) = \Big(\det(\boldsymbol{M}_{R,T_{ik}}) \det(\boldsymbol{M}_{C,T_{ik}}) \Big) \cdot [\det(\boldsymbol{M}_U)]^2,$$

where $[\det(\boldsymbol{M}_U)]^2 = (\pm 2)^{2c(S)} = 4^{c(S)}$. To evaluate the tree component block, we observe that T_{ik} contains a unique path $\mathcal{P}_{i \to k}$. Iterative Laplace expansion along the pendant vertices of $T_{ik} \setminus \mathcal{P}_{i \to k}$ evaluates to 1, reducing the determinant core exclusively to the incidence matrix of $\mathcal{P}_{i \to k}$. Let $p = |\mathcal{P}_{i \to k}|$. Deleting the row corresponding to i (in $\boldsymbol{M}_C$) and k (in $\boldsymbol{M}_R$) reduces the path incidence matrix to lower and upper bidiagonal forms, respectively. The product of their diagonal entries merges

the head and tail of each edge, yielding $\prod_{e\in\mathcal{P}_{i\to k}}(-\sigma(e)) = (-1)^p \text{sgn}(\mathcal{P}_{i\to k})$. The permutation sign resolves to $(-1)^p$ under the relabeled path indices, perfectly annihilating the parity factor since $(-1)^{2p} = 1$. Consequently, the isolated cofactor simplifies to:

$$C_{ik} = \sum_S \text{sgn}(\mathcal{P}_{i\to k}) 4^{c(S)} \prod_{e\in S} |\boldsymbol{W}_e|.$$

Multiplying by the external edge weight $\boldsymbol{W}_{kj} = \text{sgn}(\boldsymbol{W}_{kj})|\boldsymbol{W}_{kj}|$ acts as an oriented topological grafting. It merges e_{kj} onto T_{ik} to form a new spanning edge set $S' = S \cup \{e_{kj}\}$, extending the path sign to $\text{sgn}(\mathcal{P}_{i\to k})\text{sgn}(\boldsymbol{W}_{kj}) = \text{sgn}(\mathcal{P}_{i\to j})$. Summing over all intermediate nodes $k \in V_U$, we obtain:

$$\sum_{k\in V_U} C_{ik}\boldsymbol{W}_{kj} = \sum_{S'} \text{sgn}(\mathcal{P}_{i\to j}) 4^{c(S')} \prod_{e\in S'} |\boldsymbol{W}_e| \overset{(a)}{=} \sum_{\phi\in\mathcal{F}_{ij}} \text{sgn}(\mathcal{P}_{i\to j}) 2^{c(\phi)} \prod_{e\in\phi} |\boldsymbol{W}_e|.$$

Equation (a) follows by Remark 4.1, which completes the proof. □

B.3 Hoeffding's Inequality

Lemma B.1 (Hoeffding's inequality [12])**.** *Let $x_1, x_2, \cdots, x_n$ be l independent random variables satisfying $a \le x_i \le b$ for all $i = 1, 2, \cdots, n$. Let $x = \frac{1}{l}\sum_{i=1}^{l} x_i$. Then, for any $\epsilon > 0$,*

$$\mathbb{P}\{|x - \mathbb{E}(x)| \ge \epsilon\} \le 2\exp\left(-\frac{2l\epsilon^2}{(b-a)^2}\right).$$

B.4 Proof of Theorem 4.5

Proof. Let $V(\phi) = X_{ij}(\phi)2^{c(\phi)}$ and $W(\phi) = 2^{c(\phi)}$. By definition, the true matrix entry is $\boldsymbol{Q}_{ij} = \mathbb{E}[V]/\mathbb{E}[W]$ and the empirical estimator is $\widehat{\boldsymbol{Q}}_{ij} = \bar{V}/\bar{W}$, where $\bar{V}$ and $\bar{W}$ are the empirical sample means. Note that $\mathbb{E}[W] = \beta$. We analyze the deviation $|\widehat{\boldsymbol{Q}}_{ij} - \boldsymbol{Q}_{ij}|$ using the following algebraic identity:

$$\frac{a}{b} - \frac{A}{B} = \frac{B(a-A) + A(B-b)}{bB}. \tag{7}$$

By applying the triangle inequality, we obtain:

$$\left|\widehat{\boldsymbol{Q}}_{ij} - \boldsymbol{Q}_{ij}\right| = \left|\frac{\mathbb{E}[W](\bar{V} - \mathbb{E}[V]) + \mathbb{E}[V](\mathbb{E}[W] - \bar{W})}{\bar{W}\mathbb{E}[W]}\right| \le \frac{\mathbb{E}[W]|\bar{V} - \mathbb{E}[V]| + |\mathbb{E}[V]||\bar{W} - \beta|}{\bar{W}\beta}.$$

Given that the absolute value of the indicator and sign function is bounded as $|X_{ij}(\phi)| \le 1$, it follows that $|V(\phi)| \le W(\phi)$, which implies $|\mathbb{E}[V]| \le \mathbb{E}[W] = \beta$. Substituting this into the inequality, the error bound simplifies to:

$$\left|\widehat{\boldsymbol{Q}}_{ij} - \boldsymbol{Q}_{ij}\right| \le \frac{\beta|\bar{V} - \mathbb{E}[V]| + \beta|\bar{W} - \beta|}{\bar{W}\beta} = \frac{|\bar{V} - \mathbb{E}[V]| + |\bar{W} - \beta|}{\bar{W}}.$$

To ensure the total error is bounded by ϵ, we define a relative tolerance $\Delta = \frac{\epsilon\beta}{2+\epsilon}$. We assume that the following two events hold: $|\bar{V} - \mathbb{E}[V]| \le \Delta$ and $|\bar{W} - \beta| \le \Delta$. Under these conditions, the empirical denominator $\bar{W}$ is lower-bounded as:

$$\bar{W} \ge \beta - \Delta = \beta - \frac{\epsilon\beta}{2+\epsilon} = \frac{2\beta}{2+\epsilon}.$$

Consequently, the total estimation error is bounded as:

$$\left|\widehat{\boldsymbol{Q}}_{ij} - \boldsymbol{Q}_{ij}\right| \le \frac{\Delta + \Delta}{\frac{2\beta}{2+\epsilon}} = \frac{\frac{2\epsilon\beta}{2+\epsilon}}{\frac{2\beta}{2+\epsilon}} = \epsilon.$$

Next, we determine the sample size l required to satisfy these conditions using Hoeffding's Inequality. The variables are bounded by α: $V(\phi) \in [-\alpha, \alpha]$ (range 2α) and $W(\phi) \in [0, \alpha]$ (range α). The tail probabilities are:

$$\mathbb{P}\left(|\bar{V} - \mathbb{E}[V]| \geq \Delta\right) \leq 2\exp\left(-\frac{2l\Delta^2}{(2\alpha)^2}\right) = 2\exp\left(-\frac{l\Delta^2}{2\alpha^2}\right),$$
$$\mathbb{P}\left(|\bar{W} - \beta| \geq \Delta\right) \leq 2\exp\left(-\frac{2l\Delta^2}{\alpha^2}\right).$$

Applying the Union Bound, the total failure probability is at most $4\exp\left(-\frac{l\Delta^2}{2\alpha^2}\right)$. By setting this probability to δ and solving for l, we obtain:

$$l \geq 2\left(\frac{\alpha}{\beta}\right)^2\left(\frac{\epsilon+2}{\epsilon}\right)^2 \ln(4/\delta).$$

This confirms that the selected l ensures the error bound with probability at least $1-\delta$, completing the proof. □

B.5 Proof of Theorem 4.8

Proof. The execution time of the algorithm consists of the deterministic precomputation and the stochastic sampling phase. The precomputation of boundary capacities $D[u]$ requires a single pass over the neighborhood of V_U, taking strictly $O(|E|)$ time. Since Theorem 4.5 establishes the lower bound for the sample number l, it suffices to analyze the expected time required to sample and aggregate a single SMRST. The aggregation phase performs a linear traversal over the sampled SMRST components and their adjacent labeled nodes, costing at most $O(|V_U| + |E_L|)$ time per sample, where $|E_L|$ denotes the number of edges connecting to the labeled set. For the structural sampling phase, let t_i denote the absorption time of a positive loop erased random walk originating from node i. We can derive a strict upper bound for $\mathbb{E}[t_i]$ by considering the worst case scenario where the walk terminates solely upon reaching the labeled root set V_L, ignoring the possibility of early absorption by negative cycles. Thus, we have the upper bound:

$$\mathbb{E}[t_i] \leq \lim_{t\to\infty} \boldsymbol{e}_i^\top \sum_{k=0}^{t} \boldsymbol{P}_{UU}^k \boldsymbol{e}_i = (\boldsymbol{I} - \boldsymbol{P}_{UU})_{i,i}^{-1}.$$

Recall that for Wilson algorithm, the distribution of the sampled forest is independent of the node ordering. Initiating the random walk from an arbitrary sequence of nodes leaves the resulting measure unchanged. Therefore, the expected running time for generating the forest skeleton is equal to $\mathrm{Tr}\left((\boldsymbol{I} - \boldsymbol{P}_{UU})^{-1}\right)$. Because the graph is sparsely connected, the sampling time fundamentally dominates the $O(|V_U|)$ aggregation step. Summing over all l samples yields the stated overall time complexity, which completes the proof. □

C Space Complexity Analysis

In this section, we evaluate the space requirements of our proposed algorithms. Theoretically, both RandomWalkQ and MultiTreeQ produce a label propagation operator matrix $\boldsymbol{Q} \in \mathbb{R}^{|V_U|\times|V_L|}$. A naive dense implementation would require $O(|V_U| \cdot |V_L|)$ space, which is often prohibitive for large-scale networks where the number of labeled nodes can be substantial. However, our algorithms can be implemented far more efficiently by exploiting the sparsity of the sampling process. In the RandomWalkQ algorithm, each of the l independent random walks starting from an unlabeled node i terminates at exactly one labeled node j. This implies that for each Monte Carlo realization, we only need to update and store at most $|V_U|$ non zero entries. Similarly, in the MultiTreeQ algorithm, each sampled forest partition assigns each unlabeled node to a specific tree branch, and the expectations are computed based on the boundary connections of these branches.

While the Rao-Blackwellized aggregation increases the number of non zero updates per sample by marginalizing over all boundary edges, it significantly reduces the required sampling budget l due to its mathematically proven lower variance. By utilizing sparse matrix data structures to accumulate these results across the sampling rounds, the total space complexity is strictly bounded by $O(|V_U| \cdot l)$.

Given that the required number of samples l is determined solely by the error tolerance ϵ and is completely independent of the size of the labeled set $|V_L|$, this space complexity ensures that our algorithms remain highly scalable. Even for massive graphs with millions of potential roots, the memory consumption grows only linearly with the number of unlabeled nodes and the sampling budget, providing a significant advantage over traditional algebraic solvers that require storing dense intermediate inverse matrices.

D Computational Overhead of Rao-Blackwellized Aggregation

We provide a detailed analysis of the computational overhead introduced by Rao-Blackwellized (RB) aggregation in MTQ. The computational cost of MTQ consists of two main components: structural sampling and RB aggregation. The expected structural sampling cost is characterized by $\mathrm{Tr}((\boldsymbol{I} - \boldsymbol{P}_{UU})^{-1})$, which depends on both the proportion and distribution of labeled nodes. The RB aggregation cost is additionally influenced by the sizes of sampled components and the number of labeled nodes adjacent to each component.

RB aggregation cost. For a sampled SMRST ϕ, removing all edges incident to labeled nodes decomposes the unlabeled nodes into a collection of internal components $\mathcal{C}(\phi)$. For each component $C \in \mathcal{C}(\phi)$, let J_C denote the set of distinct labeled nodes adjacent to at least one node in C in the original graph, and define $a(C) = |J_C|$. The RB procedure marginalizes over the sampled terminal edge and updates all output columns corresponding to labeled nodes in J_C for every node $i \in C$. Therefore, each component C induces exactly $|C|a(C)$ candidate update-loop iterations. Components containing negative cycles contribute no update, for which we set $a(C) = 0$.

To characterize the additional computational overhead, we define

$$\kappa(\phi) = \frac{1}{|V_U|} \sum_{C \in \mathcal{C}(\phi)} |C|a(C). \tag{8}$$

Thus, $|V_U|\kappa(\phi)$ represents the total number of candidate RB update-loop iterations for a sample ϕ. We further define $\bar{\kappa} = \mathbb{E}_\phi[\kappa(\phi)]$, which measures the expected number of label-output columns updated per unlabeled node.

Effect of the labeled-node ratio. We next analyze how the RB aggregation cost depends on the proportion of labeled nodes. Let $\rho = |V_L|/|V|$ denote the labeled-node ratio, and suppose that labeled nodes are selected uniformly at random without replacement. For any edge, the probability that its endpoints belong to different partitions is

$$\frac{2|V_L||V_U|}{|V|(|V|-1)}. \tag{9}$$

Let $E_L = \{(u, j) \in E \mid u \in V_U, j \in V_L\}$ denote the set of boundary edges between labeled and unlabeled nodes. By linearity of expectation, we obtain

$$\mathbb{E}[|E_L|] = \frac{2|V_L||V_U|}{|V|(|V|-1)}|E| = 2\rho(1-\rho)\frac{|V|}{|V|-1}|E| \approx 2\rho(1-\rho)|E|. \tag{10}$$

Therefore, the expected number of boundary edges increases with ρ when $\rho < 0.5$ and decreases when $\rho > 0.5$.

Furthermore, we derive a deterministic upper bound for $\kappa(\phi)$. Let q_C denote the number of boundary edges incident to component C, and define $s_{\max} = \max_{C \in \mathcal{C}(\phi)} |C|$. Since each distinct adjacent labeled node must be connected to C through at least one boundary edge, we have $a(C) \le q_C$. Consequently,

$$\kappa(\phi) = \frac{1}{|V_U|} \sum_{C \in \mathcal{C}(\phi)} |C|a(C) \le \frac{s_{\max}}{|V_U|} \sum_{C \in \mathcal{C}(\phi)} q_C = \frac{s_{\max}|E_L|}{|V_U|}. \tag{11}$$

This bound indicates that the RB aggregation cost is jointly determined by the number of boundary edges and the sizes of sampled components. If the component structure were fixed, the boundary expansion cost would grow approximately linearly with ρ in the sparse-label regime. However,

increasing ρ also changes the sampled component structure. In particular, a larger labeled set generally shortens absorption paths and reduces the sizes of unlabeled components. Moreover, multiple boundary edges may connect a component to the same labeled node, resulting in only one output column. Consequently, $\bar{\kappa}$ need not increase monotonically with ρ and may instead decrease as component fragmentation becomes dominant.

Overall computational complexity. Let $H_L = \mathrm{Tr}((\boldsymbol{I} - \boldsymbol{P}_{UU})^{-1})$ denote the expected structural sampling cost. Combining structural sampling with RB aggregation, the expected running time of MTQ using l samples can be expressed as

$$\mathbb{E}[T_{\mathrm{MTQ}}] = \mathcal{O}\left(|E| + l\left[H_L + |V_U|(1+\bar{\kappa}) + |E_L|\right]\right). \tag{12}$$

Here, $|E|$ accounts for preprocessing, H_L characterizes the expected structural sampling cost, $|V_U|$ accounts for traversing and organizing sampled unlabeled components, $|V_U|\bar{\kappa}$ represents the expected RB expansion cost, and $|E_L|$ accounts for processing boundary information. This analysis reveals two competing effects as the labeled-node ratio increases: a larger labeled set typically decreases H_L by accelerating absorption, but may increase boundary-processing and RB-expansion costs through $|E_L|$ and $\bar{\kappa}$. Therefore, the overall computational cost of MTQ need not vary monotonically with the labeled-node ratio.

Empirical evaluation. To further investigate these effects, we conduct additional experiments on BitcoinAlpha. We generate a uniformly random permutation of all nodes and use its prefixes to construct nested labeled sets with five representative ratios $\rho \in \{1\%, 10\%, 30\%, 50\%, 70\%\}$. For each ratio, we generate $1{,}000$ shared SMRST samples for MTQ and its variant without RB aggregation, using a single Julia thread. We also numerically estimate H_L. The results are summarized in Table 3.

Table 3: Effect of the labeled-node ratio on structural sampling and RB aggregation on BitcoinAlpha. The runtime columns report the cost of processing $1{,}000$ shared samples, while Speedup compares the time required by MTQ without RB and MTQ to achieve the same target error.

ρ	$\lvert E_L\rvert$	$\mathbb{E}[\lvert E_L\rvert]$	H_L	Forest steps	Mean κ	RB updates	MTQ w/o RB	MTQ	Speedup
1%	507	481.1	4163.93	4124.22	20.901	78,274.0	0.473 s	0.770 s	$24.01\times$
10%	4,365	4,351.6	3692.47	3687.34	71.648	243,960.3	0.803 s	5.990 s	$2.31\times$
30%	10,158	10,161.3	2810.76	2810.91	34.557	91,507.5	0.643 s	3.710 s	$2.64\times$
50%	12,209	12,096.2	1970.21	1970.32	22.143	41,872.9	0.482 s	2.403 s	$3.73\times$
70%	10,219	10,161.3	1162.08	1162.59	16.860	19,135.7	0.217 s	1.101 s	$4.05\times$

As shown in Table 3, the observed number of boundary edges closely matches its expectation and peaks near $\rho = 50\%$, consistent with Eq. (10). As ρ increases from 1% to 70%, H_L decreases from 4163.93 to 1162.08, while the average number of forest steps decreases from 4124.22 to 1162.59, confirming the reduction in structural sampling cost. In contrast, the mean κ initially increases from 20.901 at $\rho = 1\%$ to 71.648 at $\rho = 10\%$, before decreasing to 16.860 at $\rho = 70\%$. This non-monotonic behavior indicates that component fragmentation eventually outweighs the growth in boundary connections.

The additional RB aggregation overhead also varies with the labeled-node ratio. In particular, the number of RB updates peaks at $\rho = 10\%$ and subsequently decreases as the sampled components become smaller. Although RB aggregation increases the cost of processing individual samples, it improves estimation precision and can reduce the total number of samples required to achieve a target error. As a result, the observed speedup is non-monotonic, decreasing to $2.31\times$ at $\rho = 10\%$ before increasing to $4.05\times$ at $\rho = 70\%$. These results demonstrate that the practical efficiency of RB aggregation depends on the interplay between sampling cost, boundary expansion, and variance reduction.

D.1 Comparison with Numerical Linear Algebra Methods

Numerical linear algebra methods provide an alternative approach to computing the label propagation operator on signed graphs. When the grounded signed Laplacian $\boldsymbol{L}_{UU}$ is positive definite and the Dirichlet problem is well posed, iterative methods such as Conjugate Gradient (CG) and Preconditioned Conjugate Gradient (PCG) can efficiently solve the linear system $\boldsymbol{L}_{UU}\boldsymbol{x}_U = -\boldsymbol{L}_{UL}\boldsymbol{y}_L$. These

methods are particularly suitable when only one or a few right-hand sides are required. However, obtaining numerous explicit entries of $\boldsymbol{Q} = -\boldsymbol{L}_{UU}^{-1}\boldsymbol{L}_{UL}$ may require solving multiple linear systems with different right-hand sides, resulting in additional computational costs.

Another class of numerical methods focuses on selected inversion, which computes specific entries of sparse matrix inverses using matrix factorizations [11, 28]. Their computational efficiency depends on the sparsity structure, factorization fill-in, and the requested inverse entries. While these methods provide an effective approach to obtaining selected inverse entries, constructing the corresponding entries of $\boldsymbol{Q}$ may involve additional computations associated with $\boldsymbol{L}_{UL}$.

In contrast, MTQ aims to construct a reusable approximation $\widehat{\boldsymbol{Q}}$ through shared signed multi-rooted spanning tree samples. Once constructed, this approximation supports repeated queries of individual entries $\boldsymbol{Q}_{ij}$, comparisons of different leaders' influence, and evaluations under different label vectors $\boldsymbol{y}_L$ without repeatedly solving linear systems. This property is particularly useful in downstream applications such as leader selection, where many candidate leaders must be evaluated. As demonstrated in Appendix E.2, MTQ achieves comparable optimization results to CG while reducing computational time in this setting. Therefore, MTQ and numerical linear algebra methods are complementary: iterative solvers are generally preferable for a small number of right-hand sides, whereas MTQ is designed for applications requiring numerous explicit operator entries or repeated evaluations on a fixed graph and labeled set.

E Experimental Details

E.1 Hyperparameter Setting

For RWQ, we set $\epsilon = 0.075$ and $\delta = 0.01$. For MTQ, we set $\epsilon = 0.15$ and $\delta = 0.01$, with the practical heuristic $\alpha/\beta = 2$. The number of samples l is determined according to Theorem 4.5 using this heuristic ratio. We further investigate the effects of these parameters in the hyperparameter sensitivity and accuracy-efficiency trade-off analyses. Following the common setting in graph semi-supervised learning (GSSL), where labeled data are typically scarce, we randomly select 1% of the nodes as the labeled set V_L and treat the remaining nodes as the unlabeled set V_U.

Truncation length in RWQ. In the RWQ baseline, the truncation length k controls the trade-off between computational efficiency and approximation accuracy. However, the original literature does not provide a rigorous criterion for selecting this parameter. In our experiments, we set k to the average expected absorption time across all unlabeled nodes, given by

$$k = \frac{1}{|V_U|}\mathbf{1}^\top(\boldsymbol{I} - \boldsymbol{P}_{UU})^{-1}\mathbf{1}.$$

From a spectral perspective, the absorption time is closely related to the Dirichlet spectral gap of the transient transition matrix $\boldsymbol{P}_{UU}$, whose leading eigenvalue λ_1 characterizes the slowest decay mode [23]. To avoid the computational overhead associated with evaluating the inverse or performing eigendecomposition on large-scale graphs, we use the topological volume ratio $k \approx \mathrm{Vol}(V)/\mathrm{Vol}(V_L)$ as a practical approximation, which scales as $\mathcal{O}(n/|V_L|)$ for graphs with relatively homogeneous degree distributions [1].

Practical choice of α/β in MTQ. Theorem 4.5 provides a fixed-budget sampling guarantee involving the graph-dependent ratio α/β, where $\alpha = 2^{c_{\max}}$ is the maximum importance weight and $\beta = \mathbb{E}_{\phi\sim P_0}[2^{c(\phi)}]$ is the expected importance weight. Computing this ratio exactly is generally intractable. Our initial motivation for setting $\alpha/\beta = 2$ was that the ratio equals 1 on structurally balanced graphs, where no negative cycles exist and $c(\phi) = 0$ for every sample. Since real-world signed graphs often contain relatively small proportions of negative edges, we heuristically choose 2 to account for deviations from the balanced case. We emphasize that this value is a practical heuristic rather than an empirical estimate or a theoretical upper bound on α/β.

To further investigate this choice, we sample $2{,}000$ SMRSTs on BitcoinAlpha and BitcoinOTC, with 1% of the nodes selected as labeled nodes. We estimate the maximum observed importance weight

and the mean importance weight as

$$\widehat{\alpha}_{\mathrm{obs}} = \max_{1\le k\le 2000} 2^{c(\phi_k)}, \qquad \widehat{\beta} = \frac{1}{2000}\sum_{k=1}^{2000} 2^{c(\phi_k)}.$$

The results are summarized in Table 4.

Table 4: Empirical statistics of importance weights based on 2,000 sampled SMRSTs.

Dataset	Negative-edge ratio	Max. $c(\phi)$	$\widehat{\alpha}_{\mathrm{obs}}$	$\widehat{\beta}$	$\widehat{\alpha}_{\mathrm{obs}}/\widehat{\beta}$
BitcoinAlpha	6.35%	6	64	6.03	10.613
BitcoinOTC	10.01%	9	512	26.34	19.438

As shown in Table 4, the observed ratios are substantially larger than the heuristic value of 2. Moreover, $\widehat{\alpha}_{\mathrm{obs}}$ is only the maximum weight observed among the sampled SMRSTs and may underestimate the theoretical maximum α. These results indicate that the practical choice $\alpha/\beta = 2$ should not be interpreted as an estimate of the theoretical ratio. Nevertheless, this heuristic yields satisfactory approximation accuracy in our experiments. The graph-dependent ratio α/β is retained in Theorem 4.5 to characterize the formal worst-case fixed-budget guarantee.

Adaptive stopping criterion. A fixed sampling budget based on the theoretical ratio α/β can be overly conservative, as the sampling variance varies across graphs and queried entries. To avoid relying on this difficult-to-compute ratio in practice, we additionally consider a batchwise adaptive stopping criterion based on the empirical Bernstein inequality.

Let $w(\phi) = 2^{c(\phi)}$ denote the importance weight and $X_{ij}(\phi)$ denote the signed indicator associated with the queried entry $\boldsymbol{Q}_{ij}$. Define $V_{ij}(\phi) = w(\phi)X_{ij}(\phi), W(\phi) = w(\phi)$. The target entry can then be expressed as $\boldsymbol{Q}_{ij} = \frac{\mathbb{E}[V_{ij}]}{\mathbb{E}[W]}$. After an initial burn-in period, we estimate the empirical variances of V_{ij} and W from the sampled SMRSTs. We then construct empirical Bernstein confidence intervals for their expectations and propagate these intervals to obtain a confidence interval for each requested $\boldsymbol{Q}_{ij}$.

Specifically, suppose the confidence intervals for $\mathbb{E}[V_{ij}]$ and $\mathbb{E}[W]$ are $[L_{V_{ij}}, U_{V_{ij}}]$ and $[L_W, U_W]$, respectively. Since $W(\phi) \ge 1$, the denominator has a strictly positive expectation. Provided that $L_W > 0$, a confidence interval for $\boldsymbol{Q}_{ij}$ is given by

$$[L_{Q_{ij}}, U_{Q_{ij}}] = \left[\min_{\substack{v\in\{L_{V_{ij}},U_{V_{ij}}\}\\ w\in\{L_W,U_W\}}} \frac{v}{w}, \max_{\substack{v\in\{L_{V_{ij}},U_{V_{ij}}\}\\ w\in\{L_W,U_W\}}} \frac{v}{w}\right].$$

Sampling proceeds in batches and terminates when $\max_{(i,j)\in\mathcal{S}} \frac{U_{Q_{ij}} - L_{Q_{ij}}}{2} \le \epsilon$, where $\mathcal{S}$ denotes the set of requested operator entries. This criterion determines the sampling budget adaptively according to the observed variance, avoiding the need to estimate the worst-case ratio α/β. The theoretical guarantee in Theorem 4.5 remains applicable to the original fixed-budget estimator, while the adaptive criterion provides an alternative practical sampling strategy.

E.2 Downstream Application: Leader Selection

To further demonstrate the practical utility of MTQ, we conduct a downstream experiment on leader selection for opinion maximization. In the leader-follower DeGroot model on signed graphs, a subset of nodes acts as stubborn leaders with fixed opinions, while the remaining nodes update their opinions through signed interactions. The equilibrium opinions of the followers are given by $\boldsymbol{x}_U = \boldsymbol{Q}\boldsymbol{y}_L$, where $\boldsymbol{Q}_{ij}$ quantifies the signed influence of leader j on follower i. Consequently, efficiently approximating $\boldsymbol{Q}$ enables repeated evaluation of opinion outcomes under different leader configurations.

We conduct the experiment on TwitterSG, randomly selecting 1% of the nodes as fixed leaders and dividing them equally into positive $(+1)$ and negative (-1) opinion groups. Our objective is to select

$K = 50$ negative leaders and switch their opinions from -1 to $+1$, thereby maximizing the number of followers with positive equilibrium opinions. Let L^- denote the set of negative leaders and $\boldsymbol{y}_L(S)$ denote the leader opinion vector after switching the opinions of leaders in $S \subseteq L^-$. The optimization problem is formulated as

$$\max_{S \subseteq L^-, |S|=K} F(S), \qquad F(S) = \sum_{i \in V_U} \mathbb{I}\left[(\boldsymbol{Q}\boldsymbol{y}_L(S))_i > 0\right]. \tag{13}$$

Here, $F(S)$ represents the number of followers whose equilibrium opinions are positive after the selected leaders change their opinions.

Exhaustively evaluating all $\binom{3770}{50}$ possible subsets is computationally infeasible. Following the greedy strategy commonly used for leader-selection problems, we initialize $S = \emptyset$ and iteratively select the remaining negative leader that maximizes the marginal gain $F(S \cup \{j\}) - F(S)$ until K leaders have been selected. Both CG and MTQ use exactly the same greedy procedure. Since signed influences can make F non-monotone, the greedy procedure serves as a scalable heuristic without guaranteeing global optimality. Table 5 reports the opinion gain and running time of CG and MTQ on TwitterSG.

Table 5: Performance comparison of CG and MTQ on the leader-selection task using TwitterSG. Gain denotes the number of followers with positive equilibrium opinions after leader selection.

Method	Gain	Runtime (s)
CG	46,880	1,183
MTQ	46,502	**463**

As shown in Table 5, MTQ achieves a gain of $46{,}502$, close to the $46{,}880$ obtained by CG, while reducing the running time from $1{,}183$ to 463 seconds, corresponding to a $2.55\times$ speedup. With a 1% leader ratio, TwitterSG contains approximately $7{,}539$ leaders and $3{,}770$ negative-leader candidates. CG requires solving thousands of linear systems with different right-hand sides to evaluate candidate-specific influence columns. In contrast, MTQ constructs a reusable approximation $\widehat{\boldsymbol{Q}}$ through shared forest samples, allowing subsequent marginal-gain evaluations to reuse the same operator without repeatedly solving linear systems. These results demonstrate the practical advantage of MTQ for downstream optimization tasks involving repeated influence queries.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction thoroughly detail the boundaries of our algorithm, highlighting both our theoretical guarantees and empirical findings.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the proposed approach are detailed in the conclusion of the manuscript.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We explicitly outline the necessary preconditions for each theoretical finding and supply full mathematical proofs in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All information required to reproduce the main results is provided in Section Experiments and Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to the code at the link: `https://github.com/SunYubo23/label-propagation-operator`. The data used in our experiments are all publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The full details can be found in Section Experiments and Appendix E.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We present the standard deviations of the approximation errors within our accuracy assessment.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We present the details of computer resources in Section 5.1 .

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work fully complies with the NeurIPS Code of Ethics. All datasets are publicly available with proper attribution, and the code is released via Github. No human subjects or sensitive data were involved.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special NeurIPS consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This manuscript focuses exclusively on foundational algorithmic improvements and theoretical guarantees for a matrix operator. As a generalized mathematical framework, it does not present direct or immediate societal consequences to discuss.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not involve the release of high-risk data or models such as pretrained language models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All third-party assets, including the benchmark datasets and software frameworks utilized in our empirical evaluations, are explicitly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our source code is available at `https://github.com/SunYubo23/label-propagation-operator`. It is accompanied by a detailed README file that documents the necessary environment dependencies and step-by-step instructions to reproduce the empirical results.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research focuses exclusively on theoretical modeling and algorithmic design. It does not involve human subjects, behavioral experiments, or any crowdsourced data collection.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: This research does not involve human subjects or behavioral experiments; therefore, Institutional Review Board approval and participant risk disclosures are not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Large Language Models are not utilized as a component of our core methodology or proposed graph algorithms.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.