Scalable Graph Coreset Selection via Greedy Sampling

Zhaiming Shen ZSHEN49@GATECH.EDU
School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, USA

Alexander Cloninger ACLONINGER@UCSD.EDU
Department of Mathematics and Halicioğlu Data Science Institute, University of California, San Diego, La Jolla, CA 92093, USA

Abstract

Sampling representative nodes from large graphs is fundamental to graph signal processing and network analysis, yet existing methods require access to the full graph Laplacian, making them impractical at scale. We propose a simple and effective column-selective graph sampling algorithm based on a minimum inner product greedy selection rule. At each iteration, the algorithm accesses only a small random subset of Laplacian columns, requiring no eigendecomposition or global graph traversal, making it well-suited for large-scale graphs where the full Laplacian cannot be stored in memory. We analyze the algorithm under the stochastic block model and show that, when the degree distribution is balanced across nodes, the algorithm achieves sampling proportional to cluster size, and that the resulting mean estimate is controlled for band-limited graph signals in the Paley-Wiener space, with the error decaying as inter-cluster connectivity weakens. Numerical experiments on both synthetic and real-world data validate the effectiveness of the proposed method.

Keywords: coreset selection, graph signal estimation, greedy sampling, scalability

1. Introduction

Graphs provide a natural representation for relational data arising in diverse domains, including social networks (Newman et al., 2002; Myers et al., 2014), biological systems (Pavlopoulos et al., 2011; Aittokallio and Schwikowski, 2006), and signal processing (Ortega et al., 2018; Mateos et al., 2019), where nodes encode entities and edges encode pairwise interactions. A problem that arises frequently in such settings is estimating the aggregate behavior of a function defined on a graph — specifically, computing the sum or average of function values across all nodes. For example, in elections, opinion polls are designed to estimate the views of a networked population toward candidates (Lippmann, 2017). In environmental monitoring, tracking regional averages of temperature, humidity, and water quality enables early intervention against forest fires and water contamination (Yu et al., 2005). In healthcare, measuring population-level indicators such as average blood pressure, weight, and cholesterol informs health policy and disease prevention strategies (Choi et al., 2017).

In many applications, however, it is infeasible or costly to collect observations at every node, making it desirable to identify a *small, representative subset of nodes*, named *coreset*, that captures the essential structure of the graph. For example, deploying sensors across every node in a network may be cost-prohibitive. Similarly, collecting health measurements such as blood pressure from individuals in remote or hard-to-reach areas can be logistically challenging and expensive. This motivates the problem of coreset selection from a large

graph such that a weighted sum of function values over the subset faithfully approximates the sum over the entire graph.

Coreset selection methods can be broadly classified into deterministic and random sampling approaches. Deterministic methods (Anis et al., 2014; Narang et al., 2013; Marques et al., 2015; Sakiyama et al., 2019; Cloninger and Mhaskar, 2021; Vahidian et al., 2020) select vertices sequentially such that a target cost function is optimized at each step, whereas random methods (Puy et al., 2018; Perraudin et al., 2018) sample vertices according to a precomputed probability distribution. In recent years, coreset methods have also been studied for better training deep learning models and graph neural networks (GNNs). These include Ding et al. (2024); Mirzasoleiman et al. (2020); Campbell and Broderick (2018); Balcilar et al. (2021). A common limitation across these methods is their reliance on global spectral information or pairwise similarities computed over the full graph. This makes them ill-suited for large-scale settings where only partial graph access is available at each iteration, which is a practical constraint that arises frequently when graphs are too large to be processed all at once.

To address this limitation, we propose a greedy graph corset sampling method that selects nodes by iteratively choosing the candidate with the minimum inner product between a random subset of Laplacian columns and a running residual vector. The algorithm is inspired by Orthogonal Matching Pursuit (OMP) (Davis et al., 1994; Pati et al., 1993), as well as the recent compressive sensing based local clustering approaches (Lai and Mckenzie, 2020; Lai and Shen, 2023; Shen et al., 2023; Shen and Kang, 2025). The proposed method has two key properties that distinguish it from existing methods. First, it accesses only a small random subset of nodes per iteration, requiring no eigendecomposition and no global graph traversal. Second, the minimum inner product selection rule implicitly promotes diversity across clusters: by choosing the node whose Laplacian column is most orthogonal to the residual, the algorithm avoids resampling nodes from already-represented clusters.

The main contributions of this work are as follows.

- We propose a simple yet effective greedy coreset sampling method for graphs that requires only local graph information at each iteration, without ever needing access to the full graph.
- We theoretically establish that under the stochastic block model (SBM) with balanced degree distribution, the proposed algorithm achieves sampling proportional to cluster size in expectation, without requiring any explicit knowledge of cluster labels or sizes.
- We further theoretically establish that this proportional sampling guarantees accurate estimation of band-limited graph signals defined in the Paley-Wiener space.
- We evaluate the proposed method on both synthetic and real-world graphs under the practical setting where only a partial subgraph is accessible at each iteration.

Organization. The rest of the paper is structured as follows. Section 2 reviews the necessary background on graph signal processing and Paley-Wiener spaces on graphs. Section 3 presents the proposed sampling algorithm and discusses its computational and memory efficiency. Section 4 establishes the theoretical guarantees of the proposed method under the stochastic block model. Section 5 reports experimental results on both synthetic and real-world datasets. Section 6 concludes with a summary and directions for future work.

2. Preliminaries

2.1. Graph Laplacian and its Spectral Decomposition

Let $G = (V, E)$ be an undirected, unweighted graph with node set $V = \{1, \ldots, n\}$ and edge set $E \subseteq V \times V$. The adjacency matrix $A \in \{0, 1\}^{n \times n}$ has entries $A_{ij} = 1$ if $(i, j) \in E$ and $A_{ij} = 0$ otherwise. The degree of node i is $d_i = \sum_{j=1}^{n} A_{ij}$, collected into the diagonal degree matrix $D = \mathrm{diag}(d_1, \ldots, d_n)$. The *unormalized graph Laplacian* is defined as $L := D - A$. The columns of L play a central role in the proposed algorithm: the i-th column $L_{:,i}$ encodes the local connectivity structure of node i, with the diagonal entry reflecting its degree and the off-diagonal entries indicating its neighbors.

Since L is real symmetric, it admits an eigendecomposition:

$$L = U \Lambda U^\top = \sum_{k=0}^{n-1} \lambda_k u_k u_k^\top, \tag{1}$$

where $U = [u_0 \mid u_1 \mid \cdots \mid u_{n-1}]$ is the orthonormal matrix of eigenvectors and $\Lambda = \mathrm{diag}(\lambda_0, \ldots, \lambda_{n-1})$ contains the eigenvalues sorted in ascending order $0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1}$. The smallest eigenvector is always $u_0 = \frac{1}{\sqrt{n}}\mathbf{1}$ with eigenvalue $\lambda_0 = 0$. The number of zero eigenvalues equals the number of connected components of G.

2.2. Graph Signal Processing and Paley-Wiener Space

A *graph signal* is a function $f : V \to \mathbb{R}$, represented as a vector $f \in \mathbb{R}^n$ with $f(i)$ denoting the signal value at node i. The *graph Fourier transform* (GFT) of f is:

$$\hat{f} = U^\top f, \qquad \hat{f}_k = u_k^\top f, \tag{2}$$

with inverse $f = U\hat{f} = \sum_{k=0}^{n-1} \hat{f}_k u_k$. The eigenvalue λ_k plays the role of graph frequency: signals concentrated on low-frequency eigenvectors (small λ_k) vary slowly across the graph, while high-frequency components (large λ_k) vary rapidly between adjacent nodes.

The *Paley-Wiener space* on the graph is the space of bandlimited signals with graph frequency below ω:

$$PW_\omega(L) = \left\{ f \in \mathbb{R}^n : \hat{f}_k = 0 \text{ whenever } \lambda_k > \omega \right\}. \tag{3}$$

For ω chosen in the spectral gap $(\lambda_{K-1}, \lambda_K)$, the space $PW_\omega(L)$ equals the span of the first K eigenvectors: $PW_\omega(L) = \mathrm{span}\{u_0, \ldots, u_{K-1}\}$.

3. Scalable Graph Coreset Selection

3.1. Proposed Method

The proposed algorithm adopts a greedy strategy to iteratively construct a coreset by selecting one node each iteration. We describe the procedure below and summarize it in Algorithm 1. The following steps repeat until T iterations.

1. It maintains a residual vector $\mathbf{b} \in \mathbb{R}^n$, initialized to zero, which tracks the cumulative influence of already-selected nodes.

Algorithm 1 Greedy Graph Coreset Selection (GGCS)

Require: unnormalized graph Laplacian $L \in \mathbb{R}^{n\times n}$, number of iterations T
Ensure: Sampled node index set $\mathcal{X}$
1: $\mathbf{b} \leftarrow \mathbf{0} \in \mathbb{R}^n$, $\mathcal{X} \leftarrow \emptyset$
2: **for** $t = 1, \dots, T$ **do**
3: Sample a small candidate set $\mathcal{C}_t \subset \{1, \dots, n\}$ uniformly at random
4: Compute inner products: $\mathbf{a} = \left| L_{:,\mathcal{C}_t}^\top \mathbf{b} \right| \in \mathbb{R}^{|\mathcal{C}_t|}$
5: Find minimum inner product value: $m^* = \min_{j\in\mathcal{C}_t} \mathbf{a}_j$
6: Collect all minimizers: $\mathcal{I} = \{j \in \mathcal{C}_t : \mathbf{a}_j = m^*\}$ and break ties uniformly at random: $i^* \leftarrow \text{UniformSample}(\mathcal{I})$
7: Update selected set: $\mathcal{X} \leftarrow \mathcal{X} \cup \{i^*\}$ and update residual: $\mathbf{b} \leftarrow \mathbf{b} - L_{:,i^*}$
8: Eliminate selected column: $L_{:,i^*} \leftarrow \boldsymbol{\infty}$
9: **end for**

2. At each iteration, a small candidate set $\mathcal{C}_t$ nodes is drawn uniformly at random.

3. Computes the absolute inner product between each candidate column of L and the current residual $\mathbf{b}$, measuring how aligned each candidate node is with the accumulated selection.

4. The node with the minimum inner product, indicating the least redundancy with previously selected nodes, is chosen, with ties broken uniformly at random.

5. The selected node is added to the coreset $\mathcal{X}$, the residual is updated by subtracting the corresponding column of L, and the selected column is set to ∞ to prevent reselection.

3.2. Scalability via Partial Graph Access

A central advantage of the proposed algorithm over spectral and deterministic methods is that it *never accesses the full graph Laplacian.* At each iteration t, it draws a random candidate set $\mathcal{C}_t$ of size $s \ll n$ and reads only the s columns $L_{:,\mathcal{C}_t}$. No eigendecomposition, no pairwise kernel matrix, and no global graph traversal is required. Note that although the entire column $L_{:i}$ is sampled, it only encodes local information about which nodes share an edge with node i. This can be readily accessed in relational databases without incurring substantial overhead or memory usage.

Time cost. At each iteration t, the computational work consists of three steps. Column retrieval reads s columns of L, each with at most $d_{\max}$ nonzero entries, at cost $O(s \cdot d_{\max})$. Inner product computation evaluates $\mathbf{a} = |L_{:,\mathcal{C}_t}^\top \mathbf{b}|$ at the same cost $O(s \cdot d_{\max})$, since $\mathbf{b} \in \mathbb{R}^n$ but each column has $O(d_{\max})$ nonzeros. The residual update $\mathbf{b} \leftarrow \mathbf{b} - L_{:,i^*}$ reads one column with the worse case cost $O(d_{\max})$. The total time cost over T iterations is therefore $O(T \cdot s \cdot d_{\max})$.

Memory cost. The memory requirement is determined by what must be held *simultaneously*, not what is accessed in total. At any point in time the algorithm stores: (i). The

residual vector $\mathbf{b} \in \mathbb{R}^n$ with $O(n)$. (ii). The current column batch $L_{:,\mathcal{C}_t}$ with $O(s \cdot d_{\max})$ for sparse L. (iii). The inner product vector $\mathbf{a} \in \mathbb{R}^s$ with $O(s)$.

The s columns in $\mathcal{C}_t$ are read, their inner products computed, and then discarded — the full Laplacian L is *never stored in memory*. The peak memory cost is therefore $O(n + s \cdot d_{\max}) = O(n)$, where the $O(n)$ term from $\mathbf{b}$ dominates when $s \cdot d_{\max} \leq n$, which holds whenever the candidate batch is much smaller than the full graph.

The key advantage. The memory cost $O(n)$ is the critical scalability gain: the algorithm requires memory linear in the number of nodes, not quadratic. This makes it applicable to graphs where n is large enough that even storing L is infeasible.

4. Theoretical Analysis

Let us present the theoretical analysis of Algorithm 1 under the stochastic block model Assumptions 1–3. It is worthwhile to note that Assumptions 2–3 are the key structural conditions under which the proposed algorithm achieves cluster-size-proportional sampling. Due to space constraints, we defer these assumptions to Appendix A.

4.1. Balanced Cluster Proportion

The key quantity governing Algorithm 1 is the inner product between Laplacian columns, which equals the (i, j) entry of L^2. Our first Lemma 1 characterizes the inner product between columns i and j of the Laplacian depending on whether they belong to the same cluster or to different clusters.

Lemma 1 (Exact L^2 Entry and SBM Expectations) *For any $i \neq j$:*

$$(L^2)_{ij} = |\mathcal{N}(i) \cap \mathcal{N}(j)| - (d_i + d_j)\,\mathbf{1}[(i, j) \in E], \tag{4}$$

where $\mathcal{N}(i) = \{k \neq i : (i, k) \in E\}$ and $d_i = |\mathcal{N}(i)|$. Under Assumptions 1–2:

$$\mathbb{E}[(L^2)_{ij}] = \begin{cases} -n_a p_a^2 \left(1 + O\left(\dfrac{q_{\max}}{p_a}\right)\right) & i, j \in V_a, \\ \displaystyle\sum_{c \neq a,b} n_c q_{ac} q_{bc} + O(n q_{\max}^2) = O(n q_{\max}^2) & i \in V_a,\ j \in V_b,\ a \neq b. \end{cases} \tag{5}$$

Algorithm 1 maintains a residual $\mathbf{b}_t = -\sum_{s=1}^{t} L_{:i_s}$ and selects the node with minimum $|L_{:j}^\top \mathbf{b}_t|$ from a random candidate set. The following Theorem 2 derives an exact expression for the expected inner product at each iteration and establishes that the algorithm samples nodes in proportion to cluster size based on Lemma 1.

Theorem 2 (Sampling Proportional to Cluster Size) *Under Assumptions 1–3, for any candidate node $j \in V_a$ and any deterministic selected set $\mathcal{X}_t = \{i_1, \ldots, i_t\}$ at t-th iteration, it satisfies at $(t+1)$-th iteration:*

$$\mathbb{E}\left[L_{:,j}^\top \mathbf{b}_t \,\middle|\, \mathcal{X}_t\right] = m_a^{(t)} \cdot n_a p_a^2 + \eta_a^{(t)}, \tag{6}$$

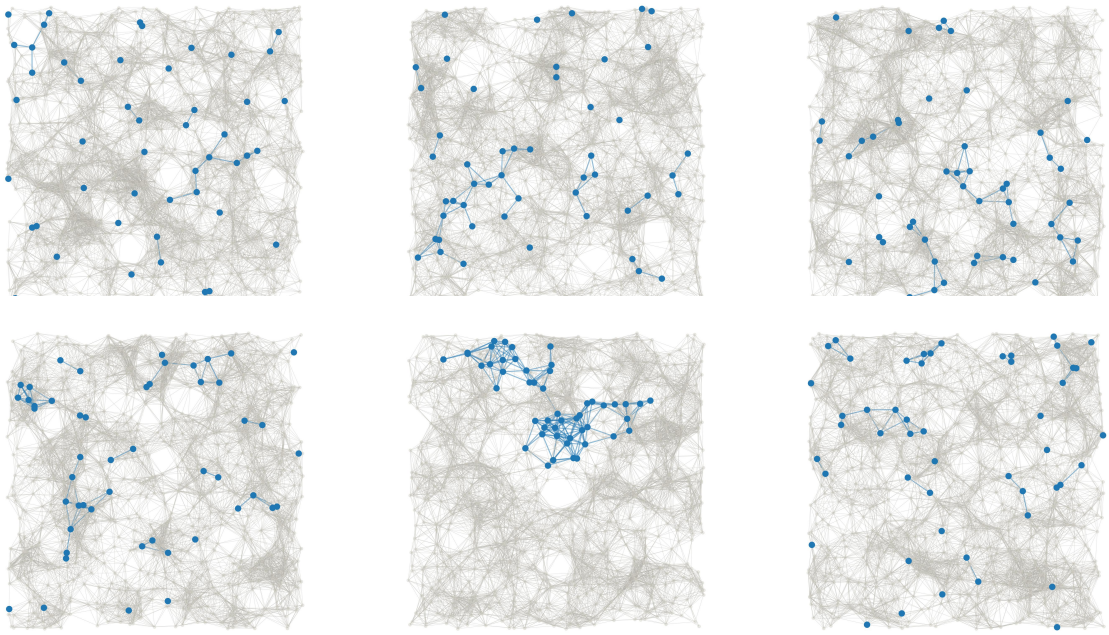

Figure 1: Examples of various sampling schemes for 50 nodes on random geometric graph generated under case (*ii*). (Top row from left to right: GGCS, Random Node, Page Rank. Bottom row from left to right: Degree-based, MHRW, uniform).

where $m_a^{(t)} = |\mathcal{X}_t \cap V_a|$ *and the remainder satisfies:*

$$|\eta_a^{(t)}| \leq m_a^{(t)} \cdot O(np_a q_{\max}) + \sum_{c \neq a} m_c^{(t)} \cdot O(nq_{\max}^2). \tag{7}$$

Consequently, $|\eta_a^{(t)}| = o(m_a^{(t)} n_a p_a^2)$. *Furthermore, suppose that* $\frac{m_a^{(t)}}{n_a}$ *is the smallest among all* $a = 1, \cdots, K$ *up to tolerance* $|\eta_a^{(t)}|$, *then in expectation Algorithm 1 selects a column in* $V_a \cap \mathcal{C}_{t+1}$ *in the* $(t+1)$*-th iteration.*

Theorem 2 shows that $|L_{:,j}^\top \mathbf{b}_t|$ is determined to leading order by $m_a^{(t)} \cdot n_a p_a^2$, with a multiplicative correction $O(q_{\max}/p_a)$ that vanishes under Assumption 2. Intuitively, Theorem 2 guarantees that Algorithm 1 always favors selecting a node from the cluster that is currently most underrepresented, thereby maintaining sampling proportional to cluster size.

4.2. Mean Estimation of Graph Signal on Coreset

Let us now connect the balanced cluster proportion to the signal estimation quality, showing that proportional coverage is sufficient for good mean estimation of band-limited functions.

Write $L = L^{\text{block}} + E$, where L^{block} is the block diagonal matrix encoding all intra-cluster information and E captures the inter-cluster connections. Our goal is to bound the absolute error between the sample mean $\hat{\mu}_S := \frac{1}{T}\sum_{i \in S} f(i)$ and the population mean $\mu := \frac{1}{n}\sum_{i \in V} f(i)$.

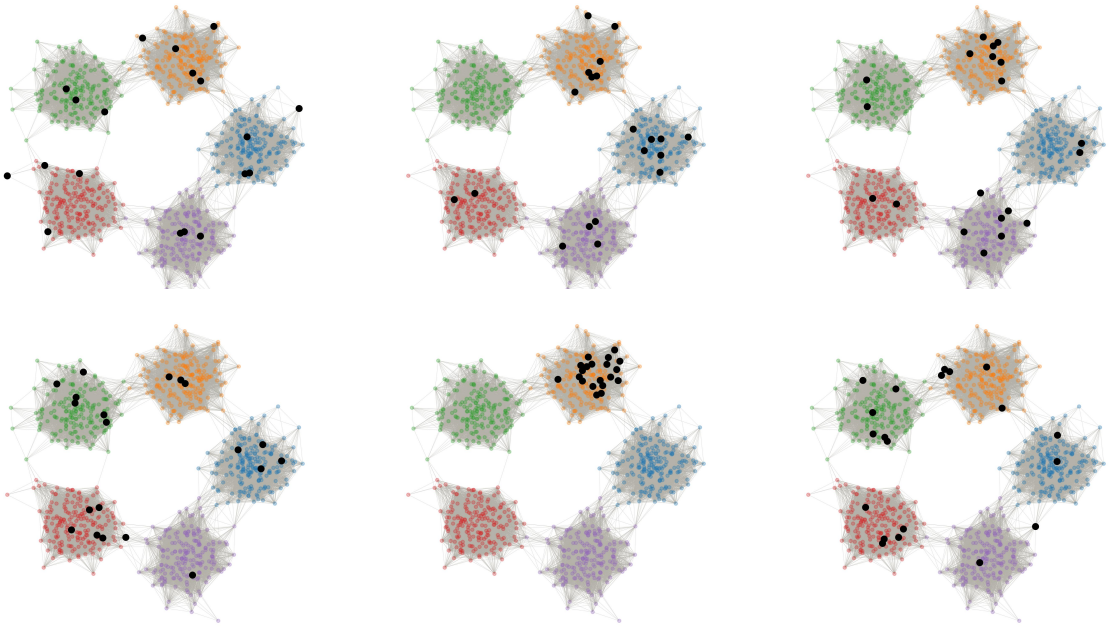

Figure 2: Examples of various sampling schemes for 20 nodes on random geometric graph generated under case (*iii*). (Top row from left to right: GGCS, Random Node, Page Rank. Bottom row from left to right: Degree-based, MHRW, uniform).

Theorem 3 (Mean Estimation Error for Band-limited Signals) *Let $f \in PW_\omega(L)$ for $\omega \in (\lambda_{K-1}, \lambda_K)$, so that $f = \sum_{k=0}^{K-1} \hat{f}_k u_k$. Let S be a coreset of size T obtained from Algorithm 1 with per-cluster counts satisfying $m_a/T = n_a/n + \epsilon_a$ with $|\epsilon_a| \leq \epsilon$ for all $a \in \{1, \ldots, K\}$. Let $\Delta = \lambda_K - \lambda_{K-1}$ denote the spectral gap of L. Then:*

$$|\hat{\mu}_S - \mu| \leq \sqrt{K}\, \|f\|_2 \left(\epsilon\sqrt{2K} + \frac{4\|E\|_2}{\Delta \cdot \sqrt{T}} \right). \tag{8}$$

The first term reflects cluster imbalance in the coreset and the second reflects eigenvector perturbation due to cross-cluster edges.

Theorem 3 shows that the coreset mean estimate is accurate provided that the sampled cluster proportions are balanced and the inter-cluster connections are sparse.

5. Experiments

Datasets. For the synthetic dataset, we consider three types of synthetic random graphs: (*i*) SBM$(n, K, \{n_a\}_{a=1}^K, p, q)$ with n nodes partitioned into K clusters of sizes $n_1, \ldots, n_K$. (*ii*) Random geometric graph in 2D, which places n nodes uniformly at random in $[0, 1]^2$, connects any two nodes within Euclidean distance 0.1. (*iii*) Random geometric graph in 2D, but with a more clean spatial clustering structure.

For the real-world datasets, we consider: (*iv*) Cora (Sen et al., 2008), which is a citation digraph 2708 nodes, and contains 7 different classes. We treat it as an undirected graph and

Table 1: Average $\epsilon(S)$ and $\mathcal{B}(S)$ on sampling 100 nodes from the SBM graph with three clusters of sizes $n_1 = 2000$, $n_2 = 1000$, $n_3 = 500$, and intra-cluster edge probabilities $p_1, p_2, p_3 = 0.01, 0.02, 0.04$, and $q_{ij} = 0.001$ for $i, j = 1, 2, 3$. The signal f is defined in (9).

	Random Node	Page Rank	Degree Based	MHRW	Uniform	GGCS (ours)
$\epsilon(S)$	0.061	0.057	0.087	0.17	0.081	**0.024**
$\mathcal{B}(S)$	0.037	0.032	0.033	0.10	0.033	**0.030**

take the largest connected component with 2485 nodes. (v) Amazon Photo (Shchur et al., 2018) is a co-purchase graph derived from Amazon, consisting of 7650 nodes spanning 8 product categories, where edges connect items frequently bought together.

Benchmark Methods. Deferred to Appendix C.

Testing Function We evaluate the proposed coreset selection algorithm using *Band-limited Function*. A band-limited function defined on a graph is a random linear combination of the first K eigenvectors of L:

$$f = \sum_{k=0}^{K-1} \alpha_k u_k, \quad \alpha_k \sim \mathcal{U}(0,1) \text{ i.i.d.}, \tag{9}$$

where $\mathcal{U}(0,1)$ denotes the uniform distribution on $[0,1]$. We multiply the function value by $\sqrt{n}$ for all cases to avoid small values of f.

Evaluation Metrics. Given a sampled coreset S of size T, we evaluate the following metrics:

- *Mean absolute error*: The primary metric is the absolute error between the sample mean $\hat{\mu}_S = \frac{1}{T}\sum_{i\in S} f(i)$ and the population mean $\mu = \frac{1}{n}\sum_{i\in V} f(i)$:

$$\epsilon(S) = \left|\hat{\mu}_S - \mu\right|. \tag{10}$$

- *Cluster balance error*: We report the per-cluster sampling count $m_a = |S \cap V_a|$ and compare against the proportional target Tn_a/n. The cluster balance error is:

$$\mathcal{B}(S) = \frac{1}{K}\sum_{a=1}^{K} \left|\frac{m_a}{T} - \frac{n_a}{n}\right|. \tag{11}$$

5.1. Results

For synthetic data, under (i), we summarize the results for different sampling schemes in Table 1 and Table 4. Under (ii) and (iii), we summarize the results in Table 2, and we also illustrate different sampling schemes in Figure 1 and Figure 2. Figure 1 shows that nodes selected by competing methods tend to cluster together or form elongated paths, whereas

Table 2: Average results on sampling 50 nodes from the random geometric graph with $n = 1000$. The signal f is defined in (9). The mean absolute error $\epsilon(S)$ is calculated under case (ii), and the class imbalance error $\mathcal{B}$(S) is calculated under case (iii). For the first row, the average true $|\mu|$ equals to 0.477.

	Random Node	Page Rank	Degree Based	MHRW	Uniform	GGCS (ours)
$\epsilon(S)$	0.084	0.086	0.084	0.50	0.071	**0.067**
$\mathcal{B}(S)$	0.042	0.043	0.045	0.30	0.047	**0.038**

Table 3: Average Cluster Balance Error $\mathcal{B}(S)$ on Cora with different sampling sizes S.

	Random Node	Page Rank	Degree Based	MHRW	Uniform	GGCS (ours)
$\|S\| = 20$	0.058	0.063	0.059	0.180	0.060	**0.056**
$\|S\| = 30$	0.049	0.049	0.049	0.173	0.051	**0.047**
$\|S\| = 50$	0.038	0.038	0.040	0.157	0.038	**0.036**
$\|S\| = 100$	0.027	0.026	0.028	0.122	0.027	**0.024**
$\|S\| = 300$	0.014	0.015	0.016	0.073	0.015	**0.012**

those selected by our method are more evenly distributed across the graph. Figure 2 further confirms that our method achieves more balanced selection across clusters compared to the baselines.

For real-world data, we summarize the class balance errors for different sampling sizes in Table 3 and Table 5. We compute the cluster balance error using the true class labels, regardless of the nodes' spatial locations. Under both (iv) and (v), our method consistently achieves lower error across all sampling sizes. It is worth noting that $\mathcal{B}(S)$ increases with the size of S by definition in Equation (11): for a fixed graph, the denominators n_a and n are constants, so a larger sampling size T naturally yields a larger error. The code to reproduce our experimental results is available at https://github.com/zzzms/GreedyGraphCoresetSelection.

6. Conclusion and Future Work

We presented a scalable graph coreset algorithm based on minimum inner product selection over random column subsets of the Laplacian, requiring only linear memory in the number of nodes and never accessing the full graph. Under the stochastic block model with balanced degree condition, we proved that the algorithm achieves sampling proportional to cluster size without knowledge of cluster labels, and that this proportional coverage guarantees the error for mean estimation for any band-limited graph signal in the Paley-Wiener space defined by the spectral gap of the Laplacian, with the error decaying as inter-cluster connectivity weakens. Several future research directions are worth pursuing: extending the theoretical analysis to incorporate the squared Laplacian and relaxing the balanced degree assumption; generalizing the framework to weighted or degree-corrected graphs; and adapting the algorithm to dynamic graphs in a streaming setting.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable feedback and constructive suggestions that improved the quality of the paper.

References

Lada A Adamic, Rajan M Lukose, Amit R Puniyani, and Bernardo A Huberman. Search in power-law networks. *Physical Review E*, 64(4):046135, 2001.

Tero Aittokallio and Benno Schwikowski. Graph-based methods for analysing networks in cell biology. *Briefings in Bioinformatics*, 7(3):243–255, 2006.

Aamir Anis, Akshay Gadde, and Antonio Ortega. Towards a sampling theorem for signals on arbitrary graphs. In *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3864–3868. IEEE, 2014.

Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Analyzing the expressive power of graph neural networks in a spectral perspective. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

Trevor Campbell and Tamara Broderick. Bayesian coreset construction via greedy iterative geodesic ascent. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 698–706. PMLR, 2018.

Kehui Chen and Jing Lei. Network cross-validation for determining the number of communities in network data. *Journal of the American Statistical Association*, 113(521):241–251, 2018.

Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. GRAM: Graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 787–795, 2017.

Alex Cloninger and Hrushikesh N Mhaskar. A low discrepancy sequence on graphs. *Journal of Fourier Analysis and Applications*, 27(5):76, 2021.

Geoffrey M Davis, Stephane G Mallat, and Zhifeng Zhang. Adaptive time-frequency decompositions. *Optical Engineering*, 33(7):2183–2191, 1994.

Mucong Ding, Yinhan He, Jundong Li, and Furong Huang. Spectral greedy coresets for graph neural networks. *arXiv preprint arXiv:2405.17404*, 2024.

Christian Hübler, Hans-Peter Kriegel, Karsten Borgwardt, and Zoubin Ghahramani. Metropolis algorithms for representative subgraph sampling. In *Proceedings of the 2008 IEEE International Conference on Data Mining (ICDM)*, pages 283–292. IEEE, 2008.

Ming-Jun Lai and Daniel Mckenzie. Compressive sensing for cut improvement and local clustering. *SIAM Journal on Mathematics of Data Science*, 2(2):368–395, 2020.

Ming-Jun Lai and Zhaiming Shen. A compressed sensing based least squares approach to semi-supervised local cluster extraction. *Journal of Scientific Computing*, 94(3):63, 2023.

Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 631–636. ACM, 2006.

Walter Lippmann. *Public Opinion*. Routledge, 2017.

Antonio G Marques, Santiago Segarra, Geert Leus, and Alejandro Ribeiro. Sampling of graph signals with successive local aggregations. *IEEE Transactions on Signal Processing*, 64(7):1832–1843, 2015.

Gonzalo Mateos, Santiago Segarra, Antonio G Marques, and Alejandro Ribeiro. Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Processing Magazine*, 36(3):16–43, 2019.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 6950–6960. PMLR, 2020.

Seth A Myers, Aneesh Sharma, Pankaj Gupta, and Jimmy Lin. Information network or social network? The structure of the Twitter follow graph. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 493–498, 2014.

Sunil K Narang, Akshay Gadde, and Antonio Ortega. Signal processing techniques for interpolation in graph structured data. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5445–5449. IEEE, 2013.

Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99(suppl_1):2566–2572, 2002.

Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.

Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44. IEEE, 1993.

Georgios A Pavlopoulos, Maria Secrier, Charalampos N Moschopoulos, Theodoros G Soldatos, Sophia Kossida, Jan Aerts, Reinhard Schneider, and Pantelis G Bagos. Using graph theory to analyze biological networks. *BioData Mining*, 4(1):10, 2011.

Nathanael Perraudin, Benjamin Ricaud, David I Shuman, and Pierre Vandergheynst. Global and local uncertainty principles for signals on graphs. *APSIPA Transactions on Signal and Information Processing*, 7(1):1–26, 2018.

Gilles Puy, Nicolas Tremblay, Rémi Gribonval, and Pierre Vandergheynst. Random sampling of bandlimited signals on graphs. *Applied and Computational Harmonic Analysis*, 44(2):446–475, 2018.

Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. Little Ball of Fur: A Python library for graph sampling. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 3133–3140. ACM, 2020.

Akie Sakiyama, Yuichi Tanaka, Toshihisa Tanaka, and Antonio Ortega. Eigendecomposition-free sampling set selection for graph signals. *IEEE Transactions on Signal Processing*, 67(10):2679–2692, 2019.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

Zhaiming Shen and Sung Ha Kang. Advancing local clustering on graphs via compressive sensing: Semi-supervised and unsupervised methods. In *Proceedings of the 1st Conference on Topology, Algebra, and Geometry in Data Science (TAG-DS 2025)*, volume 321, pages 126–146. PMLR, 2025.

Zhaiming Shen, Ming-Jun Lai, and Sheng Li. Graph-based semi-supervised local clustering with few labeled nodes. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

Michael PH Stumpf, Carsten Wiuf, and Robert M May. Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proceedings of the National Academy of Sciences*, 102(12):4221–4224, 2005.

Daniel Stutzbach, Reza Rejaie, Nick Duffield, Subhabrata Sen, and Walter Willinger. On unbiased sampling for unstructured peer-to-peer networks. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, pages 27–40. ACM, 2006.

Saeed Vahidian, Baharan Mirzasoleiman, and Alexander Cloninger. Coresets for estimating means and mean square error with limited greedy samples. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 350–359. PMLR, 2020.

Liyang Yu, Neng Wang, and Xiaoqiao Meng. Real-time forest fire detection with wireless sensor networks. In *Proceedings of the 2005 International Conference on Wireless Communications, Networking and Mobile Computing*, volume 2, pages 1214–1217. IEEE, 2005.

Appendix A. Model Assumptions

We model the graph G as a draw from the Stochastic Block Model with parameters n, K, $\{n_a\}_{a=1}^K$, $\{p_a\}_{a=1}^K$, $\{q_{ab}\}_{a,b=1,a\neq b}^K$. Our theoretical analysis requires the following assumptions.

Assumption 1 (Stochastic Block Model):
The node set V is partitioned into K disjoint clusters $V_1, \ldots, V_K$ of sizes $n_1, \ldots, n_K$ with $\sum_{a=1}^{K} n_a = n$. Edges are drawn independently with $\mathbb{P}[(i,j) \in E] = p_a$ if $i, j \in V_a$, and $\mathbb{P}[(i,j) \in E] = q_{ab}$ if $i \in V_a$, $j \in V_b$, $a \neq b$.

Assumption 2 (Sparse Inter-Cluster Connectivity):
The inter-cluster edge probabilities satisfy $q_{\max} = o(p_{\min})$, where $q_{\max} := \max_{a,b=1,\cdots,K}\{q_{ab}\}$ and $p_{\min} := \min_{a=1,\cdots,K}\{p_a\}$.

Assumption 3 (Balanced Intra-Cluster Degree):
There exists a sequence $\omega_n \to \infty$ such that the intra-cluster edge probabilities satisfy:

$$n_a p_a = \omega_n \qquad \forall\, a \in \{1, \ldots, K\}, \tag{12}$$

so $p_a = \omega_n / n_a$. This keeps the expected intra-cluster degree equal across all clusters while allowing ω_n to grow with n. Typical choices include $\omega_n = \Theta(\log n)$ (sparse regime) or $\omega_n = \Theta(\sqrt{n})$ (moderately dense regime).

Appendix B. Deferred Proofs and Other Useful Lemmas

In the subsequent subsections, we first establish an exact expression for the Laplacian column inner products (Lemma 1), then characterize the result for balanced cluster proportion (Theorem 2), and finally prove the error bound for the mean value estimation of band-limited functions defined on a graph using the selected coreset (Theorem 3).

B.1. Proof of Lemma 1

Proof [Proof of Lemma 1] Equation (4): Split $\sum_k L_{ik}L_{kj}$ into $k = i$, $k = j$, and $k \neq i, j$, giving $-d_i \mathbf{1}[(i,j) \in E]$, $-d_j \mathbf{1}[(i,j) \in E]$, and $\sum_{k \neq i,j} \mathbf{1}[(i,k) \in E]\mathbf{1}[(j,k) \in E] = |\mathcal{N}(i) \cap \mathcal{N}(j)|$ respectively.

Equation (5) Case 1 ($i, j \in V_a$): Each of the $n_a - 2$ within-cluster nodes contributes p_a^2 and each of the n_c cross-cluster nodes contributes q_{ac}^2 to $\mathbb{E}[|\mathcal{N}(i) \cap \mathcal{N}(j)|]$, giving $(n_a - 2)p_a^2 + \sum_{c \neq a} n_c q_{ac}^2$. The edge correction is $p_a \cdot 2((n_a - 1)p_a + \sum_{c \neq a} n_c q_{ac})$. Subtracting and collecting the leading terms: $(n_a - 2)p_a^2 - 2(n_a - 1)p_a^2 = -n_a p_a^2 + O(p_a^2)$, with the cross-cluster residual $O(np_a q_{\max})$, giving the first case of Equation (5).

Equation (5) Case 2 ($i \in V_a$, $j \in V_b$, $a \neq b$): The common neighbor sum splits as $(n_a - 1)p_a q_{ab} + (n_b - 1)q_{ab}p_b + \sum_{c \neq a,b} n_c q_{ac} q_{bc}$. The degree correction is $q_{ab}((n_a - 1)p_a + \sum_{c \neq a} n_c q_{ac} + (n_b - 1)p_b + \sum_{c \neq b} n_c q_{bc})$. The $O(\omega_n q_{ab})$ terms from $c = a$ and $c = b$ cancel exactly, leaving $\sum_{c \neq a,b} n_c q_{ac} q_{bc} + O(nq_{\max}^2) = O(nq_{\max}^2)$, and $O(nq_{\max}^2) = o(n_{\min} p_{\min}^2)$ since $q_{\max} = o(p_{\min})$. ∎

B.2. Proof of Theorem 2

Proof [Proof of Theorem 2] Since $L_{:,j}^\top \mathbf{b}_t = -\sum_{s=1}^{t} (L^2)_{ji_s}$ and $\mathcal{X}_t$ is deterministic, taking conditional expectation over G and splitting by cluster:

$$\mathbb{E}\Big[L_{:,j}^\top \mathbf{b}_t \mid \mathcal{X}_t\Big] = -\sum_{i_s \in V_a} \mathbb{E}[(L^2)_{ji_s}] - \sum_{c \neq a} \sum_{i_s \in V_c} \mathbb{E}[(L^2)_{ji_s}].$$

Substituting $\mathbb{E}[(L^2)_{ji_s}] = -n_a p_a^2 + O(np_a q_{\max})$ for $i_s \in V_a$ and $\mathbb{E}[(L^2)_{ji_s}] = O(nq_{\max}^2)$ for $i_s \in V_c$, $c \neq a$, from (5) gives (6)–(7).

Furthermore, suppose $\frac{m_a^{(t)}}{n_a}$ is the smallest among all $a = 1, \cdots, K$ up to tolerance $|\eta_a^{(t)}|$, i.e.,

$$\frac{m_a^{(t)}}{n_a} + \frac{|\eta_a^{(t)}|}{\omega_n^2} < \frac{m_b^{(t)}}{n_b} - \frac{|\eta_b^{(t)}|}{\omega_n^2}.$$

Multiplying the above equation both sides by $\omega_n^2 = n_a^2 p_a^2 = n_b^2 p_b^2$, gives

$$\mathbb{E}\Big[L_{:,j_a}^\top \mathbf{b}_t \,\Big|\, \mathcal{X}_t\Big] < \mathbb{E}\Big[L_{:,j_b}^\top \mathbf{b}_t \,\Big|\, \mathcal{X}_t\Big]$$

for any $j_a \in V_a$ and $j_b \in V_b$. Therefore, Algorithm 1 selects a node from V_a in the $(t+1)$-th iteration. ■

B.3. Proof of Theorem 3

Proof [Proof of Theorem 3] Let us define $\delta_k(S) := \frac{1}{T}\sum_{i\in S} u_k(i) - \frac{1}{n}\sum_{i\in V} u_k(i)$. Since $f \in PW_\omega(L)$, $\hat{f}_k = 0$ for $k \geq K$, so the estimation error reduces to:

$$\hat{\mu}_S - \mu = \sum_{k=0}^{K-1} \hat{f}_k\, \delta_k(S) = \sum_{k=1}^{K-1} \hat{f}_k\, \delta_k(S). \tag{13}$$

Notice that $\delta_0(S) = 0$, as u_0 is a constant vector across all nodes. Also, notice that for $k \geq 1$, $\delta_k(S) = \frac{1}{T}\sum_{i\in S} u_k(i)$ as $\frac{1}{n}\sum_{i\in V} u_k(i) = 0$.

Step 1: Decompose u_k via Wedin sinΘ Lemma. Write each eigenvector as:

$$u_k = \tilde{u}_k + r_k, \tag{14}$$

where $\tilde{u}_k = \sum_{j=1}^K \Phi_{jk} u_j^{\text{block}}$ is the rotated block-diagonal approximation (exactly cluster-constant within each cluster) and $r_k = u_k - \tilde{u}_k$ is the perturbation residual. By Lemma 4:

$$\sum_{k=0}^{K-1} \|r_k\|_2^2 = \|U_K - U_K^{\text{block}}\Phi\|_F^2 \leq \frac{8K\,\|E\|_2^2}{\Delta^2}. \tag{15}$$

Step 2: Bound the discrepancy $\delta_k(S)$. Substituting (14) into $\delta_k(S)$:

$$\delta_k(S) = \frac{1}{T}\sum_{i\in S} \tilde{u}_k(i) + \frac{1}{T}\sum_{i\in S} r_k(i) \triangleq \tilde{\delta}_k(S) + \rho_k(S). \tag{16}$$

For $\tilde{\delta}_k(S)$: since $\tilde{u}_k$ is exactly cluster-constant with value $\tilde{c}_k^{(a)}$ for $i \in V_a$:

$$|\tilde{\delta}_k(S)| = \left|\sum_{a=1}^K \tilde{c}_k^{(a)} \epsilon_a\right| \leq \epsilon \sum_{a=1}^K |\tilde{c}_k^{(a)}| \leq \epsilon\sqrt{K}\,\|\tilde{u}_k\|_\infty \leq \epsilon\sqrt{K},$$

where the last step uses $\|\tilde{u}_k\|_\infty \leq \|\tilde{u}_k\|_2 = 1$.

For $\rho_k(S)$: by the Cauchy-Schwarz inequality and $|S \cap V_a| \le T$:

$$|\rho_k(S)| = \left|\frac{1}{T}\sum_{i\in S} r_k(i)\right| \le \frac{1}{T}\sum_{i\in S}|r_k(i)| \le \frac{\sqrt{T}\,\|r_k\|_2}{T} = \frac{\|r_k\|_2}{\sqrt{T}},$$

Therefore, we get

$$|\delta_k(S)| \le \epsilon\sqrt{K} + \frac{\|r_k\|_2}{\sqrt{T}}. \tag{17}$$

Step 3: Combine via Cauchy-Schwarz. Applying Cauchy-Schwarz to (13):

$$\begin{aligned}
|\hat{\mu}_S - \mu| &\le \left(\sum_{k=1}^{K-1} \hat{f}_k^2\right)^{1/2} \left(\sum_{k=1}^{K-1} \delta_k(S)^2\right)^{1/2} \\
&\le \sqrt{2K}\,\|f\|_2 \sqrt{\left(\epsilon^2(K-1) + \frac{8\|E\|_2^2}{\Delta^2\cdot T}\right)} \\
&\le \sqrt{K}\,\|f\|_2 \left(\epsilon\sqrt{2K} + \frac{4\|E\|_2}{\Delta\cdot\sqrt{T}}\right),
\end{aligned}$$

which gives (8). ■

B.4. Wedin sinΘ Lemma

Lemma 4 (Wedin sinΘ Lemma) *Let $L = L_{\text{block}} + E$ where L_{block} is the block diagonal Laplacian under perfect cluster separation ($q = 0$) and E contains the cross-cluster edge contributions. Let U_K and U_K^{block} be the matrices of the first K eigenvectors of L and L_{block} respectively, and let $\Delta = \lambda_K - \lambda_{K-1}$ be the spectral gap of L. Then there exists an orthogonal matrix $\Phi \in \mathbb{R}^{K\times K}$ such that:*

$$\|U_K - U_K^{\text{block}}\Phi\|_F \le \frac{2\sqrt{2K}\,\|E\|_2}{\Delta}. \tag{18}$$

Proof We refer to Lemma 7 in Chen and Lei (2018) for a formal proof. ■

Appendix C. Other Details and Deferred Results on Experiments

C.1. Benchmark Methods.

We compare our method against baseline sampling schemes, including Random Node (Stumpf et al., 2005), Page Rank (Leskovec and Faloutsos, 2006), Degree-based (Adamic et al., 2001), Metropolis-Hastings Random Walk (Hübler et al., 2008; Stutzbach et al., 2006), and uniform node sampling. For convenience, implementations of these baselines are available via the Python library of Rozemberczki et al. (2020).

All simulations are repeated over 100 independent runs. For synthetic data, each run generates a random graph and a random sample from that generated graph. For real data, each run generates a random sample from a fixed graph.

C.2. Additional Experimental Results

Table 4: Average results on sampling 100 nodes from the SBM graph with three clusters of sizes $n_1 = 1000$, $n_2 = 1000$, $n_3 = 1000$, and intra-cluster edge probabilities $p_1, p_2, p_3 = 0.01, 0.01, 0.01$, and $q_{ij} = 0.001$ for $i, j = 1, 2, 3$. The signal f is defined in (9).

	Random Node	Page Rank	Degree Based	MHRW	Uniform	GGCS (ours)
$\epsilon(S)$	0.053	0.054	0.048	0.102	0.051	**0.046**
$\mathcal{B}(S)$	0.037	0.038	0.039	0.096	0.040	**0.035**

Table 5: Average Cluster Balance Error $\mathcal{B}(S)$ on Amazon Photo dataset with different sampling sizes S.

	Random Node	Page Rank	Degree Based	MHRW	Uniform	GGCS (ours)
$\lvert S\rvert = 20$	0.054	0.055	0.066	0.180	0.056	**0.050**
$\lvert S\rvert = 30$	0.047	0.047	0.057	0.171	0.044	**0.041**
$\lvert S\rvert = 50$	0.036	0.036	0.048	0.157	0.035	**0.032**
$\lvert S\rvert = 100$	0.025	0.026	0.043	0.142	0.024	**0.022**
$\lvert S\rvert = 300$	0.014	0.015	0.034	0.113	0.014	**0.012**