COMMUNICATION-OPTIMAL DISTRIBUTED GRAPH CLUSTERING UNDER DUPLICATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

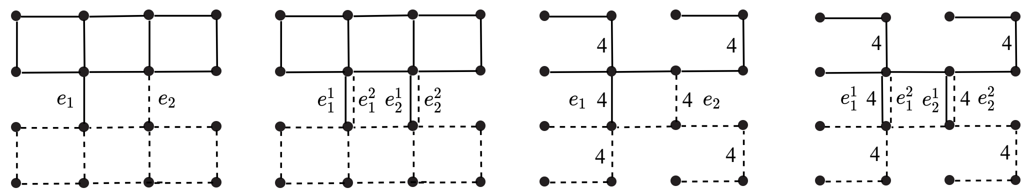
We consider the problem of clustering graph nodes over large-scale distributed graphs, when graph edges with possibly edge duplicates are observed distributively. Although edge duplicates across different sites appear to be beneficial at the first glance, in fact they could make the clustering task more complicated since potentially their processing would need extra computations and communications. We propose the first communication-optimal algorithms for two well-established communication models namely the message passing and the blackboard models. Specifically, given a graph on n nodes with edges observed at s sites, our algorithms achieve communication costs $\tilde{O}(ns)$ and $\tilde{O}(n+s)$ ($\tilde{O}$ hides a polylogarithmic factor), which almost match their lower bounds, $\Omega(ns)$ and $\Omega(n+s)$, in the message passing and the blackboard models respectively. The communication costs are asymptotically the same as those under non-duplication models, under a mild assumption on edge distribution. Our algorithms can also guarantee clustering quality nearly as good as that of centralizing all edges and then applying any standard clustering algorithm.

1 INTRODUCTION

Graph clustering is one of the most fundamental tasks in machine learning Giatsidis et al. (2014); Tian et al. (2014); Anagnostopoulos et al. (2016). Given a graph consisting of a node set and an edge set, graph clustering asks to partition graph nodes into clusters such that nodes within the same cluster are "densely-connected" by graph edges, while nodes in different clusters are "loosely-connected". Graph clustering on modern large-scale graphs imposes high computational and storage requirements, which are too expensive to obtain from a single machine. In contrast, distributed computing clusters and server storages are a popular and cheap way to meet the requirements. Distributed graph clustering has received considerable research interests, *e.g.*, Chen et al. (2016); Sun & Zanetti (2019); Zhu et al. (2019).

In an n-vertex distributed graph $G(V, E)$, each of s sites, S_i, holds a subset of edges $E_i \subseteq E$ on a common vertex set V and their union is $E = \cup_{i=1}^{s} E_i$. We consider two well-established models of communication, the *message passing* model and *blackboard* model, following the above work. In the former, there is a communication channel between every site and a distinguished coordinator. Each site can send a message to another site by first sending to the coordinator, who then forwards the message to the destination. In the latter, sites communicate with each other through a shared blackboard. The models can be further considered in two settings: edge sets of different sites are disjoint (non-duplication models) and they can have non-empty intersection (duplication models). The major objective is to minimize the communication cost that is usually measured by the total number of bits communicated.

A typical framework of distributed graph clustering is to employ graph sparsification tools to significantly reduce the size of edge sets of different sites while keeping structural properties. For example, spectral sparsifiers Spielman & Teng (2011) can sparsify an arbitrary graph while well approximating the spectral property of the original graph. Chen et al. (2016) proposed to compute spectral sparsifiers for the graphs at different sites and transmit them to the coordinator. Upon receiving all sparsifiers, the coordinator takes their union and applies a standard clustering algorithm, *e.g.*, Ng et al. (2001). However, all the existing methods that follow this framework such as Chen et al. (2016); Zhu et al. (2019) only work in non-duplication models. The assumption that edge sets



(a) A graph G w/o edge duplication (b) A graph G' w/ edge duplication (c) $H_1 \cup H_2 = (1+\epsilon)$-$SS(G)$ (d) $H_1' \cup H_2' \neq (1+\epsilon)$-$SS(G')$

Figure 1: An illustrating example for challenges in processing edge duplicates across sites. For all subfigures, edge weights are one unless stated explicitly and edges are distributed at two sites: solid edges are in site S_1 and dash edges are in S_2. (a): a graph G without edge duplication. The graph G' in (b) is similar to G but edge e_1 (and e_2) appears in both sites S_1 and S_2 as e_1^1 and e_1^2 (e_2^1 and e_2^2), respectively. (c) shows the decomposability. Each site S_i constructs a spectral sparsifier H_i of its local graph and their union is a spectral sparsifier of G. However, the decomposability does not work for G' as in (d). It is unknown how to process the two "duplicates" of e_1 and e_2, *e.g.*, e_1^1 and e_1^2 with different weights 4 and 1.

of different sites are disjoint is crucial to get the *decomposability* of spectral sparsifiers: the union of spectral sparsifiers of subgraphs at different sites is a spectral sparsifier of the distributed graph. Unfortunately, the decomposability does not work in duplication models. When edge sets of different sites have non-empty intersection, it is unclear how to process edge "duplicates" that are possible to have different edge weights after sparsification. See Figure 1 for a concrete example. To the best of our knowledge, none of the existing algorithms can perform distributed graph clustering in the more general duplication models with reasonable theoretical guarantees on both communication cost and clustering quality.

Instead of restoring the decomposability and turning to the framework, our algorithms are built based on the construction of spectral sparsifiers by graph spanners Koutis & Xu (2016). The adaptation of the algorithm to the duplication models need new algorithmic procedures such as weighted graph spanners and uniform sampling. Graph spanners are a type of graph sparsifiers that well approximate shortest-path distances in the original graph. Because computing graph spanners is an important building block of our algorithm, we also investigate the problem and obtain several interesting communication lower and upper bounds.

Our Contributions. We perform the first investigation of distributed graph clustering under duplication models. We propose communication-optimal (up to polylogarithmic factor) algorithms with communication cost $\tilde{O}(ns)$ and $\tilde{O}(n+s)$ in the message passing and blackboard with duplication models, respectively. Interestingly, the communication costs are (asymptotically) the same as the those in the non-duplication models under a mild assumption on edge distribution. It is guaranteed that the quality of our clustering results is nearly as good as the simple method of centralizing all edge sets at different sites and then applying a standard clustering algorithm, *e.g.*, Ng et al. (2001).

As side products, we also improve the poor understanding on communication complexity of constructing the important building block, graph spanners, in the blackboard model. Table 1 summarizes our findings. (See the definition of each type of spanners in Section 2.) In particular, the communication lower bounds for computing all the considered spanners under the non-duplication model are slightly lower than the duplication model. That means edge duplication presenting at different sites potentially brings more communications, unlike the problem of distributed clustering.

Related Work. There have been extensive research on graph clustering in the distributed setting, *e.g.*, Hui et al. (2007); Yang & Xu (2015); Chen et al. (2016); Sun & Zanetti (2017); Zhu et al. (2019). Yang & Xu (2015) proposed a divide and conquer method for distributed graph clustering. Chen et al. (2016) used spectral sparsifiers in graph clustering for two distributed communication models to reduce communication cost. Sun & Zanetti (2017) presented a computationally and communication efficient node degree based sampling scheme for distributed graph clustering. Zhu et al. (2019) studied distributed dynamic graph clustering based on the monotonicity property of graph sparsification. However, all these methods assume that there are no edge duplicates across different sites and do not work in the more general duplication setting. Graph spanners have been studied in the non-distributed model Thorup & Zwick (2005); Coppersmith & Elkin (2006); Cygan et al. (2013); Abboud & Bodwin (2016) and a few distributed models Censor-Hillel et al. (2018); Fernan-

Problem	Upper Bound	Lower Bound	
		Non-duplication	Duplication
$(2k-1)$-spanner	$\tilde{O}(s+n^{1+1/k})$	$\Omega(s+n^{1+1/k}\max\{1, \frac{\log s}{s^{(1+1/k)/2}}\})$	$\Omega(s+n^{1+1/k}\log s)$
$+2$ or 3-spanner	$\tilde{O}(s+n\sqrt{n+s})$	$\Omega(s+n^{3/2})$	$\Omega(s+n^{3/2}\log s)$
$+k$-spanner	$\tilde{O}(s+n\sqrt{n+s})$	$\Omega(s+n^{4/3-o(1)})$	$\Omega(s+n^{4/3-o(1)}\log s)$

Table 1: Communication complexity of computing graph spanners in the blackboard model, where n is the number of vertices in the input graph and s is the number of sites.

dez et al. (2020). Censor-Hillel et al. (2018) studied distributed constructions of pair-wise spanners that approximate distances only for some pairs of vertices in the CONGEST model. Fernandez et al. (2020) studied distributed construction of a serials of graph spanners in the message passing with and without duplication models. But, there exists no prior work considering such construction in the blackboard model, which has been a widely adopted communication model Braverman & Oshman (2015); Vempala et al. (2020); Dershowitz et al. (2021).

2 Definitions and Notations

A weighted undirected graph $G(V, E, W)$ consists of a vertex set V, an edge set E and a weight function W which assigns a weight $W(e)$ to each edge $e \in E$. W can be omitted from the presentation if it is clear from the context. Throughout the paper let $n = |V|$ and $m = |E|$ denote the number of vertices and the number of edges in G respectively, and s be the number of remote sites G is observed. Let w be the mamixum edge weight in G, *i.e.*, $w = \max_e W(e)$. We denote by $d_G(u, v)$ the *shortest-path distance* from u to v in G. A α-spanner and $+\beta$-spanner for G are a subgraph $H(V, E' \subseteq E)$ of G such that for every $u, v \in V$, $d_H(u, v) \leq \alpha * d_G(u, v)$ and $d_H(u, v) \leq d_G(u, v) + \beta$, respectively.

3 Distributed Graph Clustering

In this section, we state our distributed graph clustering algorithms in the message passing and blackboard with duplication models. We first discuss challenges introduced by edge duplicates presenting at different sites and then show how we overcome the challenges.

Definitions. Define the graph *Laplacian* of a graph G as $L = D - A$ where A is the adjacency matrix of G and D is the degree matrix, *i.e.*, a diagonal matrix with the i-th diagonal entry equal to the sum over the i-th row of A. A $(1+\epsilon)$-*spectral sparsifier* of G, denoted as $(1+\epsilon)$-$SS(G)$, is a (possibly re-weighted) subgraph H of G such that for every $x \in R^n$, the inequality

$$(1-\epsilon)x^T L_G x \leq x^T L_H x \leq (1+\epsilon)x^T L_G x$$

holds. Each edge e in G has *resistance* $R(e) = 1/W(e)$, and the *effective resistance* between any two vertices u and v in G, denoted as $R_G(u, v)$, is defined as the potential difference that has to be applied between them in order to drive one unit of current through the network G.

Challenges. Distributed graph clustering algorithms designed for non-duplication models cannot be easily extended to duplication models. We explain the fact using Chen et al. (2016) in the message passing model as an example: every site S_i constructs a spectral sparsifier of its local graph $G_i(V, E_i)$ as a synopsis H_i and then transmits H_i, instead of G_i, to the coordinator. Upon receiving H_i from all sites, the coordinator takes their union, $H = \cup_{i=1}^s H_i$ as the constructed structure. The algorithm is based on the decomposability property of spectral sparsifiers. To see this, for every $i \in [1, s]$, by definition of spectral sparsifiers, we have for every vector $x \in R^n$, $(1-\epsilon)x^T L_{G_i} x \leq x^T L_{H_i} x \leq (1+\epsilon)x^T L_{G_i} x$. Summing all inequalities for $i \in [1, s]$, we get that

$$(1-\epsilon)\sum_{i\in[1,s]} x^T L_{G_i} x \leq \sum_{i\in[1,s]} x^T L_{H_i} x \leq (1+\epsilon)\sum_{i\in[1,s]} x^T L_{G_i} x.$$

In the non-duplication model, it is easy to check that $\sum_{i=1}^s L_{G_i} = L_G$ by the definition of Laplacian matrix. Then the above inequality is equivalent to

$$(1-\epsilon)x^T L_G x \leq x^T L_H x \leq (1+\epsilon)x^T L_G x, \tag{1}$$

which concludes that H is a $(1+\epsilon)$-spectral sparsifier of G. Under the duplication model, however, it is clear that $\sum_{i=1}^{s} L_{G_i} \neq L_G$ and thus Inequality (1) does not hold any longer. In other words, the structure H constructed using the same principle is not a spectral sparsifier of G. See Figure 1 for an illustrating example.

Proposed Method. Restoring the decomposability of spectral sparsifiers in the duplication models appears to be quite challenging. We avoid it by asking every site cooperates to construct a spectral sparsifier of the distributed graph in the coordinator, who can then get clustering results by any standard clustering algorithm. A standard method of computing spectral sparsifiers Spielman & Srivastava (2011) is to sample each edge in the input graph with a probability proportional to its effective resistance and then include the sampled edges (after appropriate weight rescaling) into the sparsifier. But, when there are duplicated edges across different sites, an edge (u, v) may get sampled more than once at different sites, thereby resulting in multiple edges of possibly different weights between u and v, *e.g.*, edges e_1^1 and e_1^2 in Figure 1. It is unclear how to process these edges to guarantee the resulting structure is always a spectral sparsifier. As in Figure 1, simply taking union by summing edge weights does not produce a valid spectral sparsifier.

Instead of using the classic sampling method, we propose to make use of the fact that spectral sparsifiers can be constructed by graph spanners Koutis & Xu (2016) to compute spectral sparsifiers in the coordinator. The connection between spectral sparsifiers and graph spanners allows us to convert spectral sparsification to graph spanner construction and uniform sampling under duplication models. In the followings, we first introduce the algorithm of Koutis & Xu (2016) and then discuss how to adapt the algorithm in the message-passing and blackboard under duplication models.

The algorithm of Koutis & Xu (2016). Given a weighted graph, their algorithm first determines a set of edges that has small effective resistance through graph spanners. Specifically, it constructs a t-bundle $\log n$-spanner $J = J_1 \cup J_2 \cup \cdots \cup J_t$, that is, a sequence of $\log n$-spanners J_i for each graph $G_i = G - \cup_{j=1}^{i-1} J_j$ with $1 \le i \le t = O(\epsilon^{-2} \log n)$. Intuitively, it peels off a spanner J_i from the graph G_i to get G_{i+1} before computing the next spanner J_{i+1}, *i.e.*, J_1 is a spanner of G, J_2 is a spanner of $G - J_1$, *etc*. The t-bundle spanner guarantees that each *non-spanner* edge (edge not in the spanner) has t edge-disjoint paths between its endpoints in the spanner (and thus in G), serving as a certificate for its small effective resistance. The algorithm then uniformly samples each non-spanner edge with a fixed constant probability, *e.g.*, 0.25 and scales the weight of each sampled edge proportionally, *e.g.*, by 4 to preserve the edge's expectation. By the matrix concentration bounds, it is guaranteed that the spanner together with the sampled non-spanner edges are a moderately sparse spectral sparsifier, in which the number of edges has been reduced by a constant factor. The desirable spectral sparsifier can be obtained by repeating the process until we get a sufficient sparsity, which happens after logarithmic iterations.

Weighted Graph Spanners. An important building block in Koutis & Xu (2016) is the construction of graph spanners of stretch factor $\log n$, which can be used to construct the t-bundle $\log n$-spanner. Unfortunately, there is no algorithm that can generate such a spanner under the duplication models. Fernandez et al. (2020) developed an algorithm for constructing $(2k-1)$-spanners in unweighted graphs under the message passing with duplication model through the implementation of the greedy algorithm Althofer et al. (1993). But the algorithm does not work in weighted graphs, where the greedy algorithm would need to process the edges in nondecreasing order of their weights. This seems to be a notable obstacle in both the message passing model and the blackboard model.

In this paper, we first propose an algorithm for constructing $(4k-2)$-spanners in weighted graphs under the message passing with duplication model. We are able to overcome the challenge in weighted graphs at the expense of a reduced stretch factor $4k-2$. However, this is sufficient for the construction of $\log n$-spanners in weighted graphs by setting the parameter $k = O(\log n)$.

Specifically, we divide the range of edge weights $[1, w]$ into logarithmic intervals, where the maximum edge weight w is assumed to be polynomial in n [1]. Then we process edges in each logarithmic scale $[2^{i-1}, 2^i)$, where $1 \le i \le log_2(nw)$, as follows. Each site S_j in order decides which of its edge $e \in E_j$ of weight in $[2^{i-1}, 2^i)$ to include into the current spanner H. If including the edge e results in a cycle of at most $2k-1$ edges, then the shortest distance between e's endpoints in the current spanner is guaranteed to be less than $(4k-2)W(e)$ (see our proof below). Thus the edge

[1] This is a common and practical assumption for modern graphs.

Algorithm 1 $Spanner(G, k)$: $(4k-2)$-spanners under duplication models

Input: Graph $G(V, E, W)$ and a parameter $k > 1$
Output: Spanner H
1: $H \leftarrow \emptyset$
2: **for** $i \in [1, log_2(nw)]$ **do**
3: **for** each site S_j **do**
4: **for** each edge $e \in E_j$ of weight in $[2^{i-1}, 2^i)$ **do**
5: **if** $(V, H \cup \{e\})$ does not contain a cycle of $\leq 2k$ edges **then**
6: $H \leftarrow H \cup \{e\}$
7: **end if**
8: **end for**
9: Transmit the updated H to the next site
10: **end for**
11: **end for**
12: **return** H;

can be discarded. Otherwise, we update the current spanner H by including e. After completing processing of E_j, S_j forwards the updated spanner H to the next site. The algorithm is summarized in Algorithm (Alg.) 1.

Theorem 1. *Given a weighted graph and a parameter $k > 1$, Alg. 1 constructs a $(4k-2)$-spanner using communication cost $\tilde{O}(sn^{1+1/k})$ in the message passing with or without duplication model.*

Proof. We first prove that the stretch factor is $4k-2$. For each edge $(u, v) \in E$, if $(u, v) \notin H$, it must be that including the edge (u, v) would close a cycle of length $\leq 2k$. That is, there exists a path P of $\leq 2k-1$ edges between u and v in H. Since we process edges in logarithmic scale, the edge weights in P cannot be larger than $2W(u, v)$. Thus the path length of P is at most $(4k-2)W(e)$. Therefore, the output H is a $(4k-2)$-spanner.

We then prove the communication cost. By construction, the output graph H has girth (the minimum number of edges in a cycle contained in the graph) larger than $2k$. It is well known that a graph with girth larger than $2k$ have $O(n^{1+1/k})$ edges Althofer et al. (1993). Then H always has $O(n^{1+1/k})$ edges throughout the processing of each logarithmic interval. Thus the total communication cost is $\tilde{O}(sn^{1+1/k})$. The algorithm works for both with and without duplication settings, which do not affect the communication complexity. □

Alg. 1 can be extended to the blackboard model with the following modification: In Line 9, if site S_j does change H by adding some edge(s), it transmits the updated spanner H to the blackboard, instead of the next site; otherwise, it sends a special marker of one bit to the blackboard to indicate that it has completed the processing. The results are summarized in Theorem 2. In Section 4, we will show that the communication cost can be reduced to $2k-1$ in unweighted graphs.

Theorem 2. *The communication complexity of constructing a $(4k-2)$-spanner in weighted graphs under the blackboard with or without duplication model is $\tilde{O}(s + n^{1+1/k})$. In unweighted graph, the stretch factor can be reduced to $2k-1$.*

Constructing t-bundle $\log n$-spanner. Recall that a t-bundle $\log n$-spanner $J = J_1 \cup J_2 \cup \cdots \cup J_t$, where J_i is a $\log n$-spanner for graph $G_i = G - \cup_{j=1}^{i-1} J_j$, for $1 \leq i \leq t$. When $i = 1$, $G_1 = G$ is a distributed graph with each site S_j having edge set E_j. We can use Alg. 1 with $k = (2 + \log n)/4$ to compute a $\log n$-spanner J_1 of G_1. For $2 \leq i \leq t$, $G_j = G_{j-1} - J_j$ is again a distributed graph: each site S_j knows which of its edges E_j was included in $J_1, J_2, \cdots, J_{i-1}$ and those edges are excluded from its edge set $E_j - J_1 - J_2 - \cdots - J_{i-1}$. Therefore, the construction of a t-bundle $\log n$-spanner invokes Alg. 1 for t times. Because of $t = O(\epsilon^{-2} \log n)$ and Theorems 1 and 2, the total communication costs in the message passing and blackboard with duplication models are $\tilde{O}(sn)$ and $\tilde{O}(s + n)$, respectively.

Uniform Sampling. After the spanner construction, the algorithm of Koutis & Xu (2016) then uniformly samples each non-spanner edge with a fixed probability, *e.g.*, 0.25 and scales the weight of each sampled edge proportionally, *e.g.*, by 4. We observe that sampling with a fixed probability

Algorithm 2 *Light-SS* under duplication models

Input: $G(V, E), \epsilon \in (0, 1)$, and probability r_e for each edge e
Output: H with updated r'_e for each edge $e \in H$
1: $G_1 \leftarrow G$; $J \leftarrow \emptyset$
2: **for** $i \in [1, 24 \log^2 n/\epsilon^2]$ **do**
3: $\quad J_i \leftarrow Spanner(G_i, (2 + \log n)/4)$
4: $\quad G_{i+1} \leftarrow G_i - J_i$
5: **end for**
6: $H \leftarrow J$; $r'_e \leftarrow r_e$
7: **for** each site S_i **do**
8: $\quad$ **for** each edge $e \in E_i - J$ **do**
9: $\quad\quad$ Sample the edge e with probability p_e such that $1 - (1 - p_e * r_e)^s = 0.25$; if e is sampled, adds e to H with a new weight $4W(e)$ and set r'_e to $p_e * r_e$
10: $\quad$ **end for**
11: $\quad$ **if** it is the last iteration of the for-loop in Line 2 of Alg. 3 **then**
12: $\quad\quad$ Transmit the sampled edges to the coordinator
13: $\quad$ **end if**
14: **end for**
15: **return** H;

Algorithm 3 $(1 + \epsilon)$-*SS* under duplication models

Input: $G(V, E)$, probability r_e for each edge e, and parameters $\epsilon \in (0, 1)$ and $\rho > 1$
Output: H
1: $G_0 \leftarrow G$
2: **for** $i \in [1, \lceil \log \rho \rceil]$ **do**
3: $\quad G_i \leftarrow$ *Light-SS*$(G_{i-1}, \epsilon/\lceil \log \rho \rceil, r_e)$
4: **end for**
5: $H \leftarrow G_{\lceil \log \rho \rceil}$ {H is already transmitted to and known by the coordinator}
6: **return** H;

is much more friendly to edge duplicates as compared to sampling with a varied probability used in traditional methods such as Fung et al. (2011). For example in Figure 1, if the duplicates e_1^1 and e_1^2 of e_1 are both sampled (under a fixed probability 0.25), they still have the same weight $4W(e_1)$ and are edge duplicates again in the next iteration. If one of them, say e_1^1, is not sampled, it is removed from the (local) graph at site S_1 and will not formulate duplicates with e_1^2 at site S_2. In contrast, non-uniform sampling could result in sampled edges of rather different weights, which may not be even considered as duplicates. However, uniform sampling under duplication models is still very challenging: if a fixed probability is used for every edge, an edge with d duplicates across different sites is processed/sampled for d times, each at one of the d sites, and thus has a higher probability being sampled than another edge with smaller duplicates. This results in a non-uniform sampling.

To achieve the uniform sampling, we suppose that the probability of an edge e residing at each of the sites is a know value r_e. If we set the probability of random sampling at each site as p_e, then the probability that the edge is not sampled at each site is $1 - p_e * r_e$. It can be derived that the probability that e is sampled by *at least* one site is $p = 1 - (1 - p_e * r_e)^s$. Since the values of r_e and s are known, we can tune the value of p_e to get the expected sampling probability $p = 0.25$. At some site, if e is sampled and added to H, we update its presenting probability as $p_e * r_e$, which will be used in the next iteration. Otherwise (if e is not sampled), it is discarded and will not participate in the next iteration. See the details in Algorithms 2 and 3.

The main algorithm, Alg. 3 computes $(1 + \epsilon)$-spectral sparsifier in $\lceil \log \rho \rceil$ iterations of *Light-SS*, where ρ is a sparsification parameter. The communication cost of *Light-SS* is composed of the cost for the bundle spanner construction and the cost for non-spanner edge sampling. If the sampled edges are transmitted to the coordinator, the communication cost $\tilde{O}(m)$ could be prohibitively large. To see this, the number of edges in the output G_i after each iteration is only reduced by a constant factor because the uniform sampling removes $3/4$ of the non-spanner edges in expectation. To improve the communication cost, we keep sampled edges in each iteration at local sites and do not transmit them to the coordinator except for the very last iteration. Then similar to the input graph G, the output G_i for each iteration $i \in [1, \lceil \log \rho \rceil - 1]$ are also a distributed graph with possible edge duplication. Edge duplicates come from two sources: either the edge is included into the bundle spanner, or the edge is sampled by more than one site. In this way, the communication cost of *Light-SS* (except for the last iteration) contains only the cost of constructing the bundle spanner. In the last iteration, the number of sampled edges must be small $\tilde{O}(n)$, which is also the communication cost of their transmission. Therefore, the communication costs of Alg. 3 in the message passing and blackboard under duplication models are $\tilde{O}(n+s)$ and $\tilde{O}(ns)$, respectively. Putting all together, our results for distributed spectral sparsification under duplication models are summarized in Theorem 3 with its formal proof deferred to Appendix A.

Theorem 3 (Spectral Sparsification under Duplication Models)**.** *For a distributed graph G and parameters $\epsilon \in (0,1)$ and $\rho = O(\log n)$, Alg. 3 can construct a $(1+\epsilon)$-spectral sparsifier for G of expected size $\tilde{O}(n)$ using communication cost $\tilde{O}(ns)$ and $\tilde{O}(n+s)$ in the message passing and blackboard with duplication models respectively, with probability at least $1 - n^{-c}$ for constant c.*

Clustering in the Sparsifier. After obtaining the spectral sparsifier of the distributed graph, the coordinator applies a standard clustering algorithm such as Ng et al. (2001) in the sparsifier to get the clustering results. We can guarantee a clustering quality nearly as good as the simple method of centralizing all graph edges and then performing a clustering algorithm. Before formally stating the results, we define a few notations.

For every node set S in a graph G, let its *volume* and *conductance* be $vol_G(S) = \sum_{u\in S, v\in V} W(u,v)$ and $\phi_G(S) = (\sum_{u\in S, v\in V-S} W(u,v))/vol_G(S)$, respectively. Intuitively, a small value of conductance $\phi(S)$ implies that nodes in S are likely to form a cluster. A collection of subsets $A_1, \cdots, A_k$ of nodes is called a *(k-way) partition* of G if (1) $A_i \cap A_j = \emptyset$ for $1 \le i \ne j \le k$; and (2) $\cup_{i=1}^{k} A_i = V$. The *k-way expansion constant* is defined as $\rho(k) = \min_{partition A_1,\cdots,A_k} \max_{i\in[1,k]} \phi(A_i)$. A lower bound on $\Upsilon_G(k) = \lambda_{k+1}/\rho(k)$ implies that G has exactly k well-defined clusters Peng et al. (2015), where λ_{k+1} is the $k+1$ smallest eigenvalue of the normalized Laplacian matrix. For any two sets X and Y, their symmetric difference is defined as $X\Delta Y = (X-Y)\cup(Y-X)$.

Theorem 4. *For a distributed graph G with $\Upsilon_G(k) = \Omega(k^3)$ and an optimal partition $P_1, \cdots, P_k$ achieving $\rho(k)$ for some positive integer k, there exists an algorithm that can output partition $A_1, \cdots, A_k$ at the coordinator such that for every $i \in [1,k]$, $vol(A_i \Delta P_i) = O(k^3 \Upsilon^{-1} vol(P_i))$ holds with probability at least $1 - n^{-c}$ for constant c. The communication costs in the message passing and blackboard with duplication models are $\tilde{O}(ns)$ and $\tilde{O}(n+s)$, respectively.*

To the best of our knowledge, this is the first algorithm for performing distributed graph clustering in the message passing and blackboard with edge duplication models. Remarkably, we can show that the communication costs are *optimal*, almost matching the communication lower bounds $\Omega(ns)$ and $\Omega(n+s)$, respectively. It is interesting to see that the communication costs incurred under duplication models are asymptotically the same as those under non-duplication models. In other words, edge duplication does not incur more communications in the graph clustering task, unlike other problems such as graph spanner construction as we will show in Section 4. Although we make an assumption on the edge distribution probability, we conjecture that when the assumption is relaxed, *i.e.*, graph edges are presenting at different sites arbitrarily, the communication upper bounds remain the same in duplication models. We leave the study as an important future work.

4 Spanner Construction in the Blackboard Model

Because computing graph spanners is an important building block of our algorithm, we also study the construction of graph spanners in the blackboard models with and without edge duplication. This, unfortunately, has not been investigated by prior work yet. We prove several interesting communication upper and lower bounds for typical graph spanners as summarized in Table 1. Due to limit of space, we only describe the general $(2k-1)$-spanners in this section and move the additive spanners to the Appendix. We start with the duplication model, followed by the non-duplication model. The lower bounds obtained in Theorems 5 and 6 hold in both weighted and unweighted graphs and the rest results are on unweighted graphs.

Duplication Model. In Section 3, we have provided the communication upper bound, $\tilde{O}(s + n^{1+1/k})$, of constructing $(2k-1)$-spanners in unweighted graphs in Theorem 2. We now show that the communication lower bound is $\Omega(s + n^{1+1/k} \log s)$.

Theorem 5. *The communication lower bound of constructing a $(2k-1)$-spanner in the blackboard with duplication model is $\Omega(s + n^{1+1/k} \log s)$.*

Proof. To prove this, we target a more general statement that works for every spanner.

Lemma 1. *Suppose there exists an n-vertex graph F of size $f(n)$ such that F is the only spanner of itself or no proper subgraph F' of F is a spanner. Then the communication complexity of computing a spanner in the blackboard with duplication model is $\Omega(s + f(n) \log s)$ bits.*

Proof. Our proof is based on the reduction from the Multiparty Set-Disjointness problem ($DISJ_{m,s}$) to graph spanner computation. In $DISJ_{m,s}$, s players receive inputs $X_1, X_2, \cdots, X_s \subseteq \{1, \cdots, m\}$ and their goal is to determine whether or not $\cap_{i=1}^{s} X_i = \emptyset$. Now we construct a distributed graph G from the graph F and an instance of $DISJ_{f(n),s}$ as follows. We add edge e_j in F to site i if $j \notin X_i$ for $1 \leq j \leq f(n)$. If the coordinator outputs F as the spanner, we report $\cap_{i=1}^{s} X_i = \emptyset$; otherwise we report $\cap_{i=1}^{s} X_i \neq \emptyset$. It can be seen that the coordinator outputs F iff all its edges appear at some site, which is the case $\cap_{i=1}^{s} X_i = \emptyset$. Finally, according to the communication lower bound of $DISJ_{m,s}$ in the blackboard model Braverman & Oshman (2015), $\Omega(s + m \log s)$, the communication complexity of computing a spanner is $\Omega(s + f(n) \log s)$. ☐

For the lower bound of $(2k-1)$-spanners, the Erdos's girth conjecture states that there exists a family of graphs F of girth $2k+1$ and size $\Omega(n^{1+1/k})$ Erdos (1964). This implies that there exists only one $(2k-1)$-spanner of F, that is F itself. It is because the deletion of any edge in F would result in that the distance between the endpoints of the edge becomes at least $2k$. Then by Lemma 1, we get the lower bound $\Omega(s + n^{1+1/k} \log s)$. ☐

Non-Duplication Model. In the non-duplication model, we prove a lower bound via a reduction from the lower bound for the duplication model.

Theorem 6. *The communication complexity of constructing a $(2k-1)$-spanner in the blackboard without duplication model is $\Omega(s + n^{1+1/k} \max\{1, s^{-1/2-1/(2k)} \log s\})$.*

Proof. We can construct an instance of the $(2k-1)$-spanner problem without duplication on s sites and n vertices from an instance of the $(2k-1)$-spanner problem with duplication on s sites and $n/\sqrt{s}$ vertices. Specifically, we construct a graph G' with no duplication by replacing each vertex v by a set of vertices S_v of size $\sqrt{s}$. Since there are at most s copies of an edge (u, v) in the original graph G across the s sites, we can assign each server's copy to a distinct edge $(u', v') \in S_u \times S_v$ in G'. See Fig. 2 for an illustrating example of the construction. Then we apply an algorithm for the without duplication model, *e.g.*, the algorithm in Theorem 2, to get a $(2k-1)$-spanner H' of G'. Finally, the coordinator computes a $(2k-1)$-spanner H of G by including an edge (u, v) in H if there is at least one edge between S_u and S_v in H'.

To show the constructed H is a $(2k-1)$-spanner of G, let us consider an edge $(u, v) \in G$. By construction, there must be an edge $(u', v') \in S_u \times S_v$ in G'. Because H' is a $(2k-1)$-spanner of G', it contains a path P' of length at most $(2k-1) \cdot W(u, v)$ between u' and v'. For every edge (x', y') in P' where $x' \in S_x, y' \in S_y$, we have included an edge (x, y) in H. Therefore, there exists a path P of length at most $(2k-1) \cdot W(u, v)$ between u and v in H and thus H is a $(2k-1)$-spanner of G. Since the lower bound in the duplication model is $\Omega(s + n^{1+1/k} \log s)$ (Theorem 5), we have that the lower bound for the non-duplication model is $\Omega(s + (n/\sqrt{s})^{1+1/k} \log s) = \Omega(s + n^{1+1/k} s^{-1/2-1/(2k)} \log s)$.

Since representing the result itself needs $\Omega(n^{1+1/k})$, combining this with the above result get the final lower bound, $\Omega(s + n^{1+1/k} \max\{1, s^{-1/2-1/(2k)} \log s\})$. ☐

Discussions. We highlight several interesting observations from Table 1 and Table 2 in the Appendix.

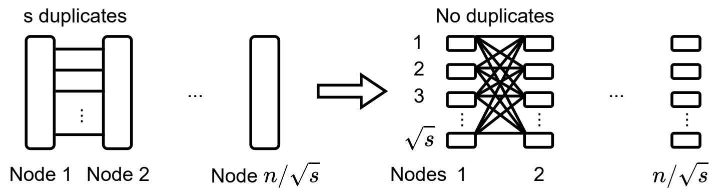


Figure 2: Converting a graph with duplication on s sites and $n/\sqrt{s}$ vertices into a graph without duplication on s sites and n vertices

1. We demonstrate that for graph spanner constructions, the blackboard model is powerful to significantly reduce the communication complexity compared to the message passing model. For instance in duplication models, computing the $(2k-1)$-spanners incurs communication cost $\tilde{O}(sn^{1+1/k})$ in the message passing model but only $\tilde{O}(s+n^{1+1/k})$ in the blackboard model. This is not necessarily the case for all computing problems. For example, for computing the sum of bit vectors modulo two Phillips et al. (2016) and estimating large moments Woodruff & Zhang (2012), the complexities are the same in both communication models.

2. To trade better communication bounds, spanners constructed in a distributed manner may include more edges than the smallest number of edges required in a centralized model. For example in $+2$-spanners and 3-spanners, the number of edges in the constructed structure is $n\sqrt{n+s}$, which is slightly larger than the optimal size $n\sqrt{n}$ in a centralized model. It is still open to investigate how to reduce the communication cost while maintaining an optimal number of edges in the spanner.

3. For constructing $(2k-1)$-spanners, the upper bound $\tilde{O}(s+n^{1+1/k})$ with a logarithmic factor hidden is very close to the lower bound $\Omega(s+n^{1+1/k}\log s)$. There is a small gap between the upper bound $\tilde{O}(s+n\sqrt{n+s})$ and lower bound $\Omega(s+n^{3/2}\log s)$ for $+2$ or 3-spanners. The gap is larger in $+k$-spanners (for $k>2$) where the lower bound becomes $\Omega(s+n^{4/3-o(1)}\log s)$. But this problem also happens in the message passing model. The construction of $+k$-spanners often involves more complex operations and might not be easy to adapt to distributed models.

5 Conclusions and Future Work

In this paper, we propose the first algorithms that can perform distributed graph clustering with edge duplication in the two well-established communication models, the message passing model and the blackboard model. We show the optimality of the achieved communication costs while maintaining a clustering quality nearly as good as a naive centralized method. This work complements the previous work on distributed graph clustering under non-duplication models and provides a complete set of tools for various clustering implications.

As the future work, we will study how to achieve the optimal communication complexity for distributed graph clustering while relaxing the assumption made. Furthermore, most of the existing work concentrate on global clustering but ignore local clustering which only returns the cluster of a given seed vertex. We will devise a local clustering method that hopefully enjoys communication cost not dependent on the size of the input graph and is more communication-efficient than traditional global graph clustering methods.

Cut sparsifiers are another type of graph sparsifiers and they can approximately preserve all the graph cut values in the original graph. Although spectral sparsifiers are also cut sparsifiers, the latters might have smaller number of edges. Because the algorithm of Koutis & Xu (2016) can be generalized to cut sparsifiers, it is promising to adapt the techniques in this work to the new problem. Finally, it is an intriguing open problem to improve the upper bounds or lower bounds and close their gap in both duplication and non-duplication models.

References

A. Abboud and G. Bodwin. The 4/3 additive spanner exponent is tight. In *Proceedings of ACM STOC Conference*, pp. 351–361, 2016.

I. Althofer, G. Das, D.P. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete Computational Geometry*, 9:81–100, 1993.

A. Anagnostopoulos, J. Lacki, S. Lattanzi, S. Leonardi, and M. Mahdian. Community detection in evolving graphs. In *Proceedings of NIPS Conference*, pp. 3522–3530, 2016.

Mark Braverman and Rotem Oshman. The communication complexity of number-in-hand set disjointness with no promise. In *Electron. Colloquium Comput. Complex.*, volume 22, pp. 2, 2015.

K. Censor-Hillel, T. Kavitha, A. Paz, and A. Yehudayoff. Distributed construction of purely additive spanners. *Distributed Computing*, 31(3):223–240, 2018.

J. Chen, H. Sun, D.P. Woodruff, and Q. Zhang. Communication-optimal distributed clustering. In *Proceedings of NIPS Conference*, pp. 3720–3728, 2016.

D. Coppersmith and M. Elkin. Sparse sourcewise and pairwise preservers. *SIAM Journal on Discrete Mathematics*, 20(2):463–501, 2006.

M. Cygan, F. Grandoni, and T. Kavitha. On pairwise spanners. In *Proceedings of STACS Conference*, pp. 209–220, 2013.

Nachum Dershowitz, Rotem Oshman, and Tal Roth. The communication complexity of multiparty set disjointness under product distributions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1194–1207, 2021.

P.G. Doyle and J.L. Snell. Random walks and electric networks. *https://arxiv.org/abs/math/0001057*, 2000.

P. Erdos. Extremal problems in graph theory. *Theory of Graphs and Its Applications*, pp. 29–36, 1964.

M.V. Fernandez, D.P. Woodruff, and T. Yasuda. Graph spanners in the message-passing model. In *Proceedings of ITCS Conference*, 2020.

W.S. Fung, R. Hariharan, N. J.A. Harvey, and D. Panigrahi. A general framework for graph sparsification. In *Proceedings of STOC Conference*, pp. 71–80, 2011.

C. Giatsidis, F.D. Malliaros, D.M. Thilikos, and M. Vazirgiannis. CORECLUSTER: A degeneracy based graph clustering framework. In *Proceedings of AAAI Conference*, pp. 44–50, 2014.

N. Harvey. Matrix concentration and sparsification. In *Workshop on Randomized Numerical Linear Algebra: Theory and Practise*, 2012.

P. Hui, E. Yoneki, S.Y. Chan, and J. Crowcroft. Distributed community detection in delay tolerant networks. In *Proceedings of 2nd ACM/IEEE International Workshop on Mobility in Evolving Internet Architecture*, 2007.

I. Koutis and S.C. Xu. Simple parallel and distributed algorithms for spectral graph sparsification. *ACM Transactions on Parallel Computing*, 3(2):14, 2016.

J.R. Lee, S.O. Gharan, and L. Trevisan. Multiway spectral partitioning and higher-order Cheeger inequalities. *Journal of the ACM*, 61(6):37, 2014.

A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In *Proceedings of NIPS Conference*, pp. 849–856, 2001.

R. Peng, H. Sun, and L. Zanetti. Partitioning well-clustered graphs: spectral clustering works! In *Proceedings of COLT Conference*, pp. 1423–1455, 2015.

J.M. Phillips, E. Verbin, and Q. Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. *SIAM Journal on Computing*, 45(1):174–196, 2016.

D.A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

D.A. Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.

H. Sun and L. Zanetti. Distributed graph clustering and sparsification. *https://arxiv.org/abs/1711.01262*, 2017.

H. Sun and L. Zanetti. Distributed graph clustering and sparsification. *ACM Transactions on Parallel Computing*, 6(3):17, 2019.

M. Thorup and U. Zwick. Approximate distance oracles. *Journal of the ACM*, 52(1):1–24, 2005.

F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu. Learning deep representations for graph clustering. In *Proceedings of AAAI Conference*, pp. 1293–1299, 2014.

J.A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

Santosh S Vempala, Ruosong Wang, and David P Woodruff. The communication complexity of optimization. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1733–1752. SIAM, 2020.

D.P. Woodruff and Q. Zhang. Tight bounds for distributed functional monitoring. In *Proceedings of STOC Conference*, pp. 941–960, 2012.

W. Yang and H. Xu. A divide and conquer framework for distributed graph clustering. In *Proceedings of ICML Conference*, pp. 504–513, 2015.

C. Zhu, T. Zhu, K.-Y. Lam, S. Han, and J. Bi. Communication-optimal distributed dynamic graph clustering. In *Proceedings of AAAI Conference*, pp. 5957–5964, 2019.

A Distributed Graph Clustering

In this section, we will provide the missing proof of several theorems for distributed graph clustering, including Theorems 2, 3, and 4.

A.1 Proof of Theorem 2

Theorem 2. *The communication complexity of constructing a* $(4k-2)$*-spanner in weighted graphs under the blackboard with or without duplication model is* $\tilde{O}(s + n^{1+1/k})$*. In unweighted graphs, the stretch factor can be reduced to* $2k-1$*.*

Proof. Considering weighted graphs, we first prove that the stretch factor of the structure H output by Alg. 1 (after adaptations described in the main text) is $4k-2$. For each edge $(u, v) \in E$, if $(u, v) \notin H$, it must be that including the edge (u, v) would close a cycle of length $\leq 2k$. That is, there exists a path P of $\leq 2k-1$ edges between u and v in H. Since we process edges in logarithmic scale, the edge weights in P cannot be larger than $2W(u, v)$. Thus the path length of P is at most $(4k-2)W(e)$. Therefore, the output H is a $(4k-2)$-spanner.

We then prove the communication cost. By construction, the output graph H has girth (the minimum number of edges in a cycle contained in the graph) larger than $2k$. It is well known that a graph with girth larger than $2k$ have $O(n^{1+1/k})$ edges Althofer et al. (1993). Then H always has $O(n^{1+1/k})$ edges throughout the processing of all logarithmic intervals. In addition, each site, if it does not modify H, needs to transmit a special marker of one bit to indicate that it has completed the processing. Therefore, the total communication cost is $\tilde{O}(s + n^{1+1/k})$. The algorithm works for both with and without duplication settings.

Algorithm 4 $(2k-1)$-spanners in the blackboard model with duplication model

Input: Unweighted graph $G(V, E)$ and a parameter $k > 1$
Output: Spanner H

$H \leftarrow \emptyset$
for each site S_i **do**
  for each edge $e = (u, v) \in E_i$ **do**
    if $d_H(u, v) > (2k-1)d_G(u, v)$ **then**
      $H = H \cup \{e\}$
    end if
  end for
  if H is updated in the above for-loop **then**
    Transmit the updated H to the blackboard
  else
    Transmit a special marker to the blackboard to indicate the completion of the processing
  end if
end for
return H;

Finally, we prove the properties in the setting of unweighted graphs. The algorithm is provided in Alg. 4. By construction, for every edge (u, v), $d_H(u, v) \leq (2k-1)d_G(u, v)$. Then the stretch factor is $2k-1$.

For communication cost, the output graph has girth larger than $2k$ by construction. Furthermore, each site needs to transmit a special marker of one bit when it does not update the current spanner H. Therefore, the total communication cost is $\tilde{O}(s + n^{1+1/k})$. □

A.2 Proof of Theorem 3

We start by defining a few notations. Suppose that P is a path connecting the two endpoints of an edge e, the *stretch* of e over P is equal to $\alpha_P(e) = W(e)\sum_{e' \in P}(1/W(e'))$. The notation $L_A \preceq L_B$ means that for every vector $x \in R^n$, $x^T L_A x \leq x^T L_B x$, while $L_A \preceq^{\{0,1\}} L_B$ means that for every vector $x \in \{0,1\}^n$, $x^T L_A x \leq x^T L_B x$. The Laplacian matrix L_G^e of an edge e in G is the Laplacian matrix of the subgraph of G containing only the edge e. It is zero elsewhere except a 2×2 submatrix. We will also use the following variant Harvey (2012) of a matrix concentration bound by Tropp (2012).

Theorem 7. *Harvey (2012) Let $Y_1, \cdots, Y_k$ be independent positive semi-definite matrices of size $n \times n$. Let $Y = \sum_{i=1}^k Y_i$ and $Z = E[Y]$. Suppose for every $i \in [1, k]$, $Y_i \preceq SZ$, where S is a scalar. Then for all $\epsilon \in [0, 1]$, $Pr[\sum_{i=1}^k Y_i \preceq (1-\epsilon)Z] \leq n \cdot exp(-\epsilon^2/2S)$, and $Pr[\sum_{i=1}^k Y_i \succeq (1+\epsilon)Z] \leq n \cdot exp(-\epsilon^2/3S)$.*

Theorem 3 (Spectral Sparsification under Edge Duplication)**.** *For a distributed graph G and parameters $\epsilon \in (0, 1)$ and $\rho = O(\log n)$, Alg. 3 can construct a $(1+\epsilon)$-spectral sparsifier for G of expected size $\tilde{O}(n)$ using communication cost $\tilde{O}(ns)$ and $\tilde{O}(n+s)$ in the message passing and blackboard with duplication models respectively, with probability at least $1 - n^{-c}$ for constant c.*

Proof. The communications happen for logarithmic iterations of distributed spanner constructions and during transmitting sampled edges in the last iteration. Constructing a t-bundle $\log n$-spanner involves computation of a $\log n$-spanner for $t = O(\epsilon^{-2}\log n)$ times, thereby incurring communication cost of $\tilde{O}(n+s)$ and $\tilde{O}(ns)$ under the message passing and blackboard models, respectively. As we will prove shortly, the output sparsifier has size $\tilde{O}(n)$. Because the sampled edges in the last iteration are a part of the output, It also has size $\tilde{O}(n)$. Therefore, the total communication costs are $\tilde{O}(n+s)$ and $\tilde{O}(ns)$, respectively.

It is easy to see by simple mathematical calculations that the probability of sampling a non-spanner edge across sites is 0.25. Then the proof that the output is a spectral sparsifier of the input graph follows directly from Koutis & Xu (2016). For self-containedness, we provide the proof below.

We first prove that $R_G(e) \leq \log n/t \cdot W(e)$ and $W(e) \cdot L_G^e \preceq \log n/t \cdot L_G$ for $t = O(\epsilon^{-2}\log n)$. By construction, for every edge $e \in G - J$, there are t edge-disjoint paths $P_1, \cdots, P_t$ between the two endpoints of e in J, such that for every $i \in [1,t]$, $\alpha_{P_i}(e) \leq \log n$. By definition, for every $i \in [1,t]$, we have that

$$\alpha_{P_i}(e) = W(e)\sum_{e\in P_i}(1/W(e)) \leq \log n. \tag{2}$$

According to the formula for resistors connected in series, for every path P_i with $i \in [1,t]$, the effective resistance between the two endpoints of e in P_i is equal to

$$R_{P_i}(e) = \sum_{e\in P_i} R(e) = \sum_{e\in P_i}(1/W(e)) \tag{3}$$

Combining Equations (2) and (3), we have that for every $i \in [1,t]$, $R_{P_i}(e) \leq \log n/W(e)$. According to the formula for resistors connected in parallel, for a set of edge-disjoint paths $\{P_1, \cdots, P_t\}$ between e's two endpoints, and let P be the union of these paths $P = \cup_{i=1}^t P_i$, the effective resistance between e's two endpoints in P is equal to $R_P(e) = (\sum_{i=1}^t (R_{P_i}(e))^{-1})^{-1} \leq \log n/t \cdot W(e)$. According to the Rayleigh's monotonicity law Doyle & Snell (2000), for any subgraph H of G and any edge $e \in G$, $R_G(e) \leq R_H(e)$ holds. Therefore,

$$R_G(e) \leq R_P(e) \leq \log n/t \cdot W(e). \tag{4}$$

By Spielman & Srivastava (2011), we have

$$L_G^e \preceq R_G(e) L_G. \tag{5}$$

By combining Equation (5) with Equation (4), we have that

$$W(e) \cdot L_G^e \preceq \log n/t \cdot L_G. \tag{6}$$

Next, we prove that the output H of *Light-SS* is a $(1+\epsilon)$-spectral sparsifier. For every edge $e \in G-J$, let X_e be the random variable defined as

$$X_e = \begin{cases} 4W(e)L_G^e, & \text{with probability } 0.25 \\ 0, & \text{otherwise} \end{cases}$$

For every $i \in [1, (\lfloor \epsilon^2/(6\log n) \rfloor)^{-1}]$, let $J_i = \lfloor \epsilon^2/(6\log n)\rfloor J$, which implies that

$$L_{J_i} = \lfloor \epsilon^2/(6\log n)\rfloor L_J.$$

We then apply Theorem 7 to the random matrix

$$Y = \sum_{e\in G-J} X_e + \sum_{i=1}^{(\lfloor \epsilon^2/(6\log n)\rfloor)^{-1}} L_{J_i} = \sum_{e\in G-J} X_e + L_J.$$

Note that

$$\begin{aligned} E(Y) &= E(\sum_{e\in G-J} X_e + L_J) = \sum_{e\in G-J} E(X_e) + L_J \\ &= \sum_{e\in G-J} L_G^e + L_J = L_G. \end{aligned}$$

By the definition of $X(e)$ and Equation (6), for every $e \in G - J$ we have that

$$X(e) \preceq 4W(e) \cdot L_G^e \preceq \epsilon^2/(6\log n) \cdot L_G.$$

Furthermore, by definition of J_i and the fact that $L_J \preceq L_G$, we have for every $i \in [1, (\lfloor \epsilon^2/(6\log n)\rfloor)^{-1}]$,

$$L_{J_i} = \lfloor \epsilon^2/(6\log n)\rfloor \cdot L_J \preceq \epsilon^2/(6\log n) \cdot L_G.$$

Now the condition of Theorem 7 is satisfied with $S = \epsilon^2/(6\log n)$. Therefore, the inequality

$$(1-\epsilon)L_G \preceq L_H \preceq (1+\epsilon)L_G \tag{7}$$

holds with probability at least $1 - 1/2n \cdot exp(-3 \log n) = 1 - 1/2n^{-2}$.

Finally, we prove that the property of the main algorithm, Alg. 3. By the property of *Light-SS* proved above and Induction, for every $i \in [1, \lceil \log \rho \rceil]$, the event that the inequality

$$(1 - \epsilon/\lceil \log \rho \rceil)^i L_G \preceq L_{G_i} \preceq (1 + \epsilon/\lceil \log \rho \rceil)^i L_G$$

holds and the expected size is

$$O(ni \log^2 n \log^2 \rho/\epsilon^2 + m/2^i),$$

happens with probability at least $(1 - 1/n^2)^i$. Since *SS* outputs $H = G_{\lceil \log \rho \rceil}$ as the final spectral sparsifier, the expected size becomes $O(n \log^3 n \log^3 \rho/\epsilon^2 + m/\rho)$. Because $\epsilon \in (0, 1)$ and $\rho = O(\log n)$, the expected size is $\tilde{O}(n)$. The desirable properties hold with probability at least $1 - n^{-c}$ for constant c. □

A.3 Proof of Theorem 4

For every node set $S \subseteq V$ in G, let its *volume* and *conductance* be $vol_G(S) = \sum_{u \in S, v \in V} W(u, v)$ and $\phi_G(S) = (\sum_{u \in S, v \in V - S} W(u, v))/vol_G(S)$, respectively. Intuitively, a small value of conductance $\phi(S)$ implies that nodes in S are likely to form a cluster. A collection of subsets $A_1, \cdots, A_k$ of nodes is called a *(k-way) partition* of G if (1) $A_i \cap A_j = \emptyset$ for $1 \leq i \neq j \leq k$; and (2) $\cup_{i=1}^k A_i = V$. The *k-way expansion constant* is defined as $\rho(k) = \min_{partition A_1, \cdots, A_k} \max_{i \in [1,k]} \phi(A_i)$. Let the *normalized Laplacian matrix* of G be $\mathcal{L}_G = D_G^{-1/2} L_G D_G^{-1/2}$ and its eigenvalues are $\lambda_1(\mathcal{L}_G) \leq \cdots \leq \lambda_n(\mathcal{L}_G)$. The high-order Cheeger inequality shows that $\lambda_k/2 \leq \rho(k) \leq O(k^2)\sqrt{\lambda_k}$ Lee et al. (2014). A lower bound on $\Upsilon_G(k) = \lambda_{k+1}/\rho(k)$ implies that, G has exactly k well-defined clusters Peng et al. (2015). It is because a large gap between λ_{k+1} and $\rho(k)$ guarantees the existence of a k-way partition $A_1, \cdots, A_k$ with bounded $\phi(A_i) \leq \rho(k)$, and that any $(k+1)$-way partition $A_1, \cdots, A_{k+1}$ contains a subset A_i with significantly higher conductance $\rho(k+1) \geq \lambda_{k+1}/2$ compared with $\rho(k)$. For any two sets X and Y, the symmetric difference of X and Y is defined as $X \Delta Y = (X - Y) \cup (Y - X)$. To prove Theorem 4, we will use the following lemma and theorems.

Lemma 2. *Chen et al. (2016) Let H be a $(1 + \epsilon)$-spectral sparsifier of $G(V, E)$ for some $\epsilon \leq 1/3$. For all node sets $S \subseteq V$, the inequality $0.5 \cdot \phi_G(S) \leq \phi_H(S) \leq 2 \cdot \phi_G(S)$ holds.*

Theorem 8. *Peng et al. (2015) Given a graph G with $\Upsilon_G(k) = \Omega(k^3)$ and an optimal partition $S_1, \cdots, S_k$ achieving $\rho(k)$ for some positive integer k, the spectral clustering algorithm can output partition $A_1, \cdots, A_k$ such that, for every $i \in [1, k]$, the inequality $vol(A_i \Delta S_i) = O(k^3 \Upsilon^{-1} vol(S_i))$ holds.*

Theorem 4. *For a distributed graph G with $\Upsilon_G(k) = \Omega(k^3)$ and an optimal partition $P_1, \cdots, P_k$ achieving $\rho(k)$ for some positive integer k, there exists an algorithm that can output partition $A_1, \cdots, A_k$ at the coordinator such that for every $i \in [1, k]$, $vol(A_i \Delta P_i) = O(k^3 \Upsilon^{-1} vol(P_i))$ holds with probability at least $1 - n^{-c}$ for constant c. The communication costs in the message passing and blackboard with duplication models are $\tilde{O}(ns)$ and $\tilde{O}(n + s)$, respectively.*

Proof. The algorithm starts by distributively constructing a spectral sparsifier H of G in the coordinator using our algorithms in Theorem 3 and then applies a standard graph clustering algorithm, *e.g.*, spectral clustreing Ng et al. (2001) in H to get the clustering results. The communication costs directly follow from Theorem 3 since the final clustering step does not incur communications. The rest of the proof follows from Chen et al. (2016) and we present it for the sake of self-containedness.

We prove that if G satisfies that $\Upsilon_G(k) = \Omega(k^3)$, H also satisfies that $\Upsilon_H(k) = \Omega(k^3)$. By the definition of Υ, it suffices to prove that $\rho_H(k) = \Theta(\rho_H(k))$ and $\lambda_{k+1}(\mathcal{L}_H) = \Theta(\lambda_{k+1}(\mathcal{L}_G))$. The former follows from that for every $i \in [1, k]$, the inequality

$$0.5 \cdot \phi_G(S_i) \leq \phi_H(S_i) \leq 2 \cdot \phi_G(S_i)$$

holds, according to Lemma 2. According to the definition of $(1 + \epsilon)$-spectral sparsifier and simple math, it holds for every vector $x \in R^n$ that

$$\begin{aligned}(1 - \epsilon)x^T D_G^{-1/2} L_G D_G^{-1/2} x &\leq x^T D_G^{-1/2} L_H D_G^{-1/2} x \\ &\leq (1 + \epsilon)x^T D_G^{-1/2} L_G D_G^{-1/2} x.\end{aligned}$$

Algorithm 5 *BFS* in the blackboard model with duplication model

Input: Graph $G(V, E)$ and root vertex $u \in V$
Output: BFS tree T
1: $T \leftarrow \{u\}; A \leftarrow \{u\}; C \leftarrow \emptyset$
2: **while** $C \neq V$ **do**
3:     **for** each site S_i **do**
4:         S_i transmits its edges $(v, w) \in E_i$ such that $v \in A$ and $w \notin C$ and (v, w) was not transmitted by sites S_j for $j < i$
5:         If such an edge cannot be found, S_i transmits a special marker to indicate the completion of its processing
6:         The coordinator includes the received edges (v, w) into T and maintains $N = \{w \mid (v, w)\}$
7:     **end for**
8:     $C \leftarrow C \cup A$; $A \leftarrow N$
9: **end while**
10: **return** T;

By the definition of normalized graph Laplacian $\mathcal{L}_G$, and the fact that for every vector $y \in R^n$,

$$0.5 \cdot y^T D_G^{-1} y \leq y^T D_H^{-1} y \leq 2 y^T D_G^{-1} y,$$

we have that for every $i \in [1, n]$,

$$\lambda_i(\mathcal{L}_H) = \Theta(\lambda_i(\mathcal{L}_G)),$$

which implies that $\lambda_{k+1}(\mathcal{L}_H) = \Theta(\lambda_{k+1}(\mathcal{L}_G))$. Then we can apply the spectral clustering algorithm in H to get the desirable properties, according to Theorem 8. $\square$

B GRAPH SPANNERS

B.1 BFS IN THE BLACKBOARD MODEL

Here we discuss an important building block for graph spanner construction: growing a breath first search (BFS) tree from a root vertex in a distributed graph. We observe that the communication complexity of computing a BFS tree from a given vertex in the blackboard model with or without duplication is $\tilde{O}(n + s)$, which is significantly smaller than $\tilde{O}(ns)$ in the message passing model Fernandez et al. (2020). This can be achieved by a simple distributed protocol where the blackboard maintains a partial BFS tree and an active set A of vertices, both initialized to be the root vertex. In each iteration, each of the sites S_i in order transmits edges of vertices in A pointing to a vertex that has never been in the active set and has not transmitted previously by site S_j for $j < i$. If such an edge cannot be found, S_i submits a special marker to indicate the completion of its processing. At the end of each iteration, the current active set A is updated to be the other endpoints of the transmitted edges. Since the edge linking each vertex to the BFS tree is transmitted at most once and each site has to transmit a special marker if it cannot add a new edge, the incurred communication cost is $\tilde{O}(n + s)$. The formal pseudo-code can be found in Alg. 5.

Theorem 9. *The communication complexity of building a BFS tree in the blackboard with or without duplication model is $\tilde{O}(n + s)$.*

Proof. We prove it for the more general duplication model. In Alg. 5, the edge linking each vertex to the BFS tree is transmitted at most *once*. This is because in Line 4, site S_i would not transmit an edge (v, w) if the edge is already transmitted previously by some site S_j for $j < i$. This is possible in the blackboard model since edges sent to the blackboard by one site are visible to all other sites. Furthermore, each site has to transmit one bit of information if it does not send any edge. Therefore, the total communication cost is $\tilde{O}(n + s)$. $\square$

B.2 +2-SPANNERS AND 3-SPANNERS

Upper Bound. We show that the communication complexity of constructing +2-spanners in the blackboard *without* edge duplication is $\tilde{O}(s + n\sqrt{n + s})$ (Theorem 10). It is achieved by a simple distributed algorithm as provided in Alg. 6. First, we aim to include all the edges of vertices with

Algorithm 6 +2-spanners in the blackboard without duplication model

Input: Graph $G(V, E)$
Output: Spanner H
$H \leftarrow \emptyset$
for each site S_i **do**
  for each vertex u **do**
    if the sum of the number of u's edges in the blackboard and the number of u's edges in E_i is no larger than $\sqrt{n+s}$ **then**
      S_i transmits all its edges $(u, v) \in E_i$ and then the coordinator includes these edges into H
    else
      S_i transmits a special marker to inform the completion of its processing
    end if
  end for
end for
The coordinator samples $\tilde{O}(n/\sqrt{n+s})$ vertices R uniformly at random with replacement from all the vertices V
for each sampled vertex $v \in R$ **do**
  The coordinator grows a BFS tree from v and includes edges in the BFS tree into H
end for
return H;

degree at most $\sqrt{n+s}$ in the spanner. However, the vertex degrees are not given directly. A naive method is that each site transmits all vertex degrees to the blackboard who then takes their sum. But this incurs very costly communication $\tilde{O}(ns)$. Our solution is that each site S_i in order transmits each vertex u's edges in its edge set E_i if the sum of the number of u's edges in the blackboard and the number of u's edges in E_i is no larger than $\sqrt{n+s}$. This only incurs communication cost $\tilde{O}(s+n\sqrt{n+s})$, instead of $\tilde{O}(ns)$. Next, the coordinator samples $\tilde{O}(n/\sqrt{n+s})$ vertices uniformly at random with replacement from all the vertices and let the sampled set be R. It then grows a BFS tree from each sampled vertex in R using Alg. 5, and includes edges of the BFS trees in the spanner. Our results are summarized in Theorem 10.

Theorem 10. *The communication complexity of constructing a* $+2$*-spanner in the blackboard with or without duplication model is* $\tilde{O}(s + n\sqrt{n+s})$.

Proof. We first prove the distance surplus $+2$. Consider the collection C of (immediate) neighbors of vertices of degree at least $\sqrt{n+s}$ in G. The event X that the sample set R contains at least one vertex from each set of neighbors in C happens with probability at least $1 - o(1)$. This can be obtained by a direct application of a well-known sampling fact, Lemma 3 with $U = V$ and $t = \sqrt{n+s}$.

Lemma 3 (Lemma 8 in Fernandez et al. (2020))**.** *Let* C *be a collection of sets over a ground set* U *each of size at least* t*. If we sample* $|U|/t \cdot \log |C/\delta|$ *elements from* U *uniformly with replacement, with probability at least* $1 - \delta$ *we sample at least one element from each set in* C*.*

Consider the shortest path P between two vertices u, v in G. If all edges on P are present in H, then $d_H(u, v) = d_G(u, v)$ and the distance surplus trivially holds. Otherwise, let $(u', v') \in P$ be a missing edge in H. We know both u' and v' have degree at least $\sqrt{n+s}$ as otherwise all their edges are included in H. Suppose the event X occurs and $x \in R$ is a neighbor of u'. Then we have

$$\begin{aligned} d_H(u, v) &\leq d_H(u, x) + d_H(x, v) \\ &= d_G(u, x) + d_G(x, v) \\ &\leq d_G(u, u') + 1 + d_G(u', v) + 1 \\ &= d_G(u, v) + 2. \end{aligned}$$

The first and third inequalities follow from the triangle inequality. The second equality holds since edges in the BFS tree rooted at x are included in H. The last equality holds because u' lies on the shortest path P.

We now prove the communication cost. Our method of including all edges of vertices with degree at most $\sqrt{n+s}$ incurs communication cost $\tilde{O}(s + n\sqrt{n+s})$. Constructing all the $\tilde{O}(n/\sqrt{n+s})$ BFS trees requires communication cost $\tilde{O}(n\sqrt{n+s})$ since growing each BFS tree incurs $\tilde{O}(n+s)$ communication according to Theorem 9. Therefore, the total communication cost is $\tilde{O}(s+n\sqrt{n+s})$. We point out that this method can be adapted to the duplication model. The modification is in the implementation of including all the edges of vertices with degree at most $\sqrt{n+s}$. When edge duplicates across sites are allowed, each site S_i checks whether the number of u's edges in the blackboard and the number of distinct edges associated with u in E_i of S_i is no larger than $\sqrt{n+s}$. If so, it only transmits the distinct edges of u to the blackboard, excluding any edges already in the blackboard. It is easy to see that both the correctness and the communication cost are not affected by this adaption. □

Since the $+2$-spanner construction algorithm immediately gives a 3-spanner construction algorithm in unweighted graphs, communication upper bounds of computing 3-spanners follow from Theorem 10, as shown in Corollary 1.

Corollary 1. *The communication complexity of constructing a* 3*-spanner in the blackboard with or without duplication model is* $\tilde{O}(s + n\sqrt{n+s})$.

Lower Bound. We first consider the duplication model and then will extend to the non-duplicaiton model. For the duplication model, we can prove the communication lower bound of computing $+2$-spanners is $\Omega(s + n^{3/2}\log s)$. It is because the size lower bound of $+2$-spanners is $\Omega(n^{3/2}\log s)$ as well as Lemma 1. There is only a small gap between the upper bound and the lower bound with an approximation ratio of $\sqrt{(n+s)/n}$.

The lower bound $\Omega(s + n^{3/2})$ for constructing 3-spanners in the duplication model can also be derived using Lemma 1. See Theorem 11 for the formal presentation. Computing 3-spanners is weaker than computing $+2$-spanners and thus might enjoy better lower bounds. However, how to achieve a tighter bound for 3-spanners remain open in both the message passing and the blackboard models.

Theorem 11. *The communication complexity of constructing a* $+2$*-spanner or a* 3*-spanner in the blackboard with duplication model is* $\Omega(s + n^{3/2}\log s)$.

Proof. According to Lemma 4, there is a graph G on n vertices with size $\Theta(n^{1.5})$ and girth at least 6. Since removing any edge in G increases the distance from its endpoints to at least 5, the only $+2$-spanner (3-spanner, respectively) of G is G itself. Then by applying $f(n) = n^{1.5}$ in Lemma 1, we get the desired lower bound $\Omega(s + n^{3/2}\log s)$.

Lemma 4 (Lemma 5 in Fernandez et al. (2020))**.** *For every* n*, there is a family of graphs on* n *vertices with* $\Theta(n^{1.5})$ *edges and girth at least* 6*.*

□

We now consider lower bounds in the non-duplication model, where we cannot use the technique in Lemma 1. But, we provide a weaker and similar lemma, Lemma 5, for the non-duplication model. By incorporating Lemma 5 into the analysis of Theorems 11, we get the lower bound of computing $+2$-spanner or 3-spanners, $\Omega(s + n^{3/2})$, as shown in Theorem 12.

Lemma 5. *Suppose there exists an* n*-vertex graph* F *of size* $f(n)$ *such that* F *is the only spanner of itself or no proper subgraph* F' *of* F *is a spanner. Then the communication complexity of computing a spanner in the blackboard without duplication model is* $\Omega(s + f(n))$ *bits.*

Proof. First, representing the result itself needs $f(n)$ bits. Next, each site needs to transmit at least one bit of information to inform the completion of its processing if it does not transmit some edge(s). Therefore, the total communication cost of constructing a spanner is $\Omega(s + f(n))$ bits. □

Theorem 12. *The communication complexity of constructing a* $+2$*-spanner or a* 3*-spanner in the blackboard without duplication model is* $\Omega(s + n^{3/2})$.

Problem	Upper Bound		Lower Bound	
	Non-duplication	Duplication	Non-duplication	Duplication
$(2k-1)$-spanner	$\tilde{O}(ks^{1-2/k}n^{1+1/k}+snk)$	$\tilde{O}(sn^{1+1/k})$	$\Omega(ks^{1/2-1/(2k)}n^{1+1/k}+sn)$	$\Omega(sn^{1+1/k})$
$+2$ or 3-spanner	$\tilde{O}(\sqrt{s}n^{3/2}+sn)$	$\tilde{O}(sn^{3/2})$	$\Omega(\sqrt{s}n^{3/2}+sn)$	$\Omega(sn^{3/2})$
$+k$-spanner	$\tilde{O}(\sqrt{s/k}n^{3/2}+snk)$	$\tilde{O}(sn^{3/2})$	$\Omega(n^{4/3-o(1)}+sn)$	$\Omega(sn^{4/3-o(1)})$

Table 2: Communication complexity of computing graph spanners in the message passing model, where n is the number of vertices in the input graph and s is the number of sites.

B.3 $+k$-SPANNERS

The communication upper bound of constructing $+k$-Spanners, $\tilde{O}(s+n\sqrt{n+s})$, immediately follows from the upper bound of constructing $+2$-spanners, because $+2$-spanners are valid $+k$-spanners. In the duplication and non-duplication models, the communication lower bounds $\Omega(s+n^{4/3-o(1)}\log s)$ and $\Omega(s+n^{4/3-o(1)})$ can be obtained by using the size lower bound $\Omega(n^{4/3-o(1)})$ of $(+k)$-spanners Abboud & Bodwin (2016) and Lemma 1 and 5, respectively.

Theorem 13. *The communication complexity of constructing a $+k$-spanner in the blackboard with or without duplication model is $\tilde{O}(s+n\sqrt{n+s})$.*

Theorem 14. *The communication complexity of constructing a $+k$-spanner in the blackboard with duplication model is $\Omega(s+n^{4/3-o(1)}\log s)$.*

Theorem 15. *The communication complexity of constructing a $+k$-spanner in the blackboard without duplication model is $\Omega(s+n^{4/3-o(1)})$.*

B.4 PRIOR RESULTS IN THE MESSAGE PASSING MODEL

Table 2 provides the communication complexity of constructing graph spanners in the message passing model Fernandez et al. (2020).