Dynamic Spectral Clustering with Provable Approximation Guarantee

Steinar Laenen [1] **He Sun** [1]

Abstract

This paper studies clustering algorithms for dynamically evolving graphs $\{G_t\}$, in which new edges (and potential new vertices) are added into a graph, and the underlying cluster structure of the graph can gradually change. The paper proves that, under some mild condition on the cluster-structure, the clusters of the final graph G_T of n_T vertices at time T can be well approximated by a dynamic variant of the spectral clustering algorithm. The algorithm runs in amortised update time $O(1)$ and query time $o(n_T)$. Experimental studies on both synthetic and real-world datasets further confirm the practicality of our designed algorithm.

1. Introduction

For any graph $G = (V, E)$ and parameter $k \in \mathbb{N}$ as input, the objective of graph clustering is to partition the vertex set of G into k clusters such that vertices within each cluster are better connected than to the rest of the graph. Since large-scale graphs are commonly used to model practical datasets, designing efficient graph clustering algorithms is an important problem in machine learning and related fields.

In practice, these large-scale graphs usually evolve over time: not only are new vertices and edges added into a graph, but the graph's clusters could also change gradually, resulting in a new cluster-structure in the long term. Instead of periodically running a clustering algorithm from scratch, it is important to design algorithms that can quickly identify and return the new clusters in dynamically evolving graphs.

In this paper we study clustering for dynamically evolving graphs, and obtain the following results. As the first and conceptual contribution, we propose a model for dynamic graph clustering. In contrast to the classical model for dynamic graph algorithms (Thorup, 2007; Beimel et al., 2022), our proposed model considers not only edge insertions but also vertex insertions; as such the underlying graph can gradually form a new cluster-structure with a different number of clusters from the initial graph.

As the second and algorithmic result, we design a randomised graph clustering algorithm that works in the above-mentioned model, and our result is as follows:

Theorem 1.1 (Informal statement of Theorem 4.6)**.** *Let $G_1 = (V_1, E_1)$ be a graph of n_1 vertices and $k = \widetilde{O}(1)$ clusters.*[1] *Assume that new edges, which could be adjacent to new vertices, are added to G_t at each time t to obtain G_{t+1}, and there are $O(\mathrm{poly}(n_1))$ added edges in total at time $T = O(\mathrm{poly}(n_1))$ to form G_T of n_T vertices and k' clusters. Then, there is a randomised algorithm such that the following hold with high probability:*

- *The initial k clusters of G_1 can be approximately computed in $\widetilde{O}(|E_1|)$ time.*
- *The new k' clusters of G_T can be approximately computed with amortised update time $O(1)$ and query time $o(n_T)$.*

To examine the result, we notice that, although the number of clusters k in G_1 can be identified with the classical eigen-gap heuristic (Ng et al., 2001; von Luxburg, 2007), computing an eigen-gap is expensive and cannot be directly applied to determine the change of k in dynamically evolving graphs. Our result shows that the new number of clusters k' can be computed by a dynamic clustering algorithm with sublinear query time. Secondly, as the running time of a clustering algorithm is at least linear in the number of edges in G_T and it takes $\Omega(n_T)$ time to output the cluster membership of all the vertices, obtaining an $o(n_T)$ amortised query time[2] is significant. To the best of our knowledge, our work presents the first such result with respect to theoretical guarantees of the output clusters, and time complexity.

Our algorithm not only achieves strong theoretical guarantees, but also works very well in practice. For instance, for input graphs with 300,000 vertices and up to 490,000,000 edges generated from the stochastic block model, our algo-

[1]School of Informatics, University of Edinburgh, United Kingdom. Correspondence to: Steinar Laenen <steinar.laenen@ed.ac.uk>, He Sun <h.sun@ed.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]We use $\widetilde{O}(n)$ to represent $O(n \cdot \log^c(n))$ for constant c.

[2]Throughout the paper we use T to denote query time, and t as arbitrary time throughout the sequence of graphs $\{G_t\}$.

rithm runs more than 100 times faster than repeated execution of spectral clustering on the updated graphs, while obtaining a comparable clustering result.

1.1. Overview of the Algorithm

For any input graph G_1 with a well-defined cluster structure, we first construct a *cluster-preserving sparsifier* H_1 of G_1, which is a sparse subgraph of G_1 that maintains its cluster-structure, and employ spectral clustering on H_1 to obtain the initial k clusters of G_1. After this, with a new edge arriving at every time t, our designed algorithm applies two components to track the cluster-structure of G_t.

The first component is a dynamic algorithm that maintains a cluster-preserving sparsifier H_t for G_t. Our designed algorithm is based on sampling edges with probability proportional to the degrees of their endpoints, and these edges get resampled if their degrees have significantly changed. We show that H_t always preserves the cluster-structure of G_t, and the algorithm's amortised update time complexity is $O(1)$.

Our second component is an algorithm that dynamically maintains a *contracted graph* $\widetilde{G}_t$ of G_t, and this contracted graph is used to sketch the cluster-structure of G_t. For the first input graph G_1 and the output of spectral clustering on H_1, our initial contracted graph $\widetilde{G}_1$ consists of k super vertices with self-loops: these super vertices correspond to the k clusters in G_1, and are connected by edges with weight equal to the cut values of the corresponding clusters in H_1. After that, when new edges (and potentially new vertices) arrive over time, our algorithm updates $\widetilde{G}_t$ such that (new) clusters are represented by either the same super vertices, newly added vertices, or a combination of both. The algorithm further updates the edge weights between the super vertices. With slight increase in the number of vertices of $\widetilde{G}_t$ over time, we prove that the cluster-structure in G_t is approximately preserved in $\widetilde{G}_t$. In particular, when new clusters are formed in G_t, this new cluster-structure of G_t can be identified by the eigen-gap of $\widetilde{G}_t$'s Laplacian matrix. See Figure 1 for the illustration of our approach.

1.2. Related work

Our work directly relates to a number of works on incremental spectral clustering algorithms (e.g., (Dhanjal et al., 2014; Martin et al., 2018; Ning et al., 2007)). These works usually rely on analysing the change of approximate eigenvectors and don't show the approximation guarantee of the returned clusters. Many works along this direction further employ matrix perturbation theory in their analysis, requiring that the total number of vertices in a graph is fixed.

Our work is also linked to related dynamic graph algorithm problems (e.g., (Bernstein et al., 2022; Saranurak & Wang, 2019)). However, most works in dynamic graph algorithms focus on the design of dynamic algorithms in a *general* graph, while for dynamic clustering one needs to assume the presence of cluster-structures in the initial and final graphs, such that the algorithm's performance can be rigorously analysed. Nevertheless, some of our presented techniques, like the adaptive sampling, are inspired by the dynamic graph algorithms literature.

2. Preliminaries

2.1. Notation

Let $G = (V, E, w)$ be an undirected graph with $|V| = n$ vertices, $|E| = m$ edges, and weight function $w : V \times V \to \mathbb{R}_{\geqslant 0}$. For any edge $e = \{u, v\} \in E$, we write $w_G(u, v)$ or $w_G(e)$ to express the weight of e. For a vertex $u \in V$, we denote its *degree* by $\deg_G(u) \triangleq \sum_{v \in V} w_G(u, v)$, and the volume for any $S \subseteq V$ is defined as $\mathrm{vol}_G(S) \triangleq \sum_{u \in S} \deg_G(u)$. For any $S, T \subset V$, we define the *cut value* between S and T by $w_G(S, T) \triangleq \sum_{e \in E_G(S,T)} w_G(e)$, where $E_G(S, T)$ is the set of edges between S and T. Moreover, for any $S \subset V$, the conductance of S is defined as

$$\Phi_G(S) \triangleq \frac{w_G(S, V \setminus S)}{\min\{\mathrm{vol}_G(S), \mathrm{vol}_G(V \setminus S)\}}$$

if $S \neq \emptyset$, and $\Phi_G(S) = 1$ if $S = \emptyset$. For any integer $k \geqslant 2$, we call subsets of vertices $A_1, \ldots, A_k$ a k-way partition of G if $\bigcup_{i=1}^{k} A_i = V$ and $A_i \cap A_j = \emptyset$ for different i and j. We define the *k-way expansion* of G by

$$\rho_G(k) \triangleq \min_{\text{partitions } A_1,\ldots,A_k} \max_{1 \leqslant i \leqslant k} \Phi_G(A_i).$$

Our analysis is based on the spectral properties of graphs, and we list the basics of spectral graph theory. For a graph $G = (V, E, w)$, let $D_G \in \mathbb{R}^{n \times n}$ be the diagonal matrix defined by $D_G(u, u) = \deg_G(u)$ for all $u \in V$. We denote by $A_G \in \mathbb{R}^{n \times n}$ the *adjacency matrix* of G, where $A_G(u, v) = w_G(u, v)$ for all $u, v \in V$. The *normalised Laplacian matrix* of G is defined as $\mathcal{L}_G \triangleq I - D_G^{-1/2} A_G D_G^{-1/2}$, where I is the $n \times n$ identity matrix. The normalised Laplacian $\mathcal{L}_G$ is symmetric and real-valued, and has n real eigenvalues which we write as $0 = \lambda_1(\mathcal{L}_G) \leqslant \ldots \leqslant \lambda_n(\mathcal{L}_G) \leqslant 2$; we use $f_i \in \mathbb{R}^n (1 \leqslant i \leqslant n)$ to express the eigenvector of $\mathcal{L}_G$ corresponding to λ_i.

Lemma 2.1 (Higher-order Cheeger inequality, (Lee et al., 2014))**.** *There is an absolute constant $C_{2.1}$ such that it holds for any graph G and $k \geqslant 2$ that*

$$\frac{\lambda_k(\mathcal{L}_G)}{2} \leqslant \rho_G(k) \leqslant C_{2.1} \cdot k^3 \sqrt{\lambda_k(\mathcal{L}_G)}. \qquad (1)$$

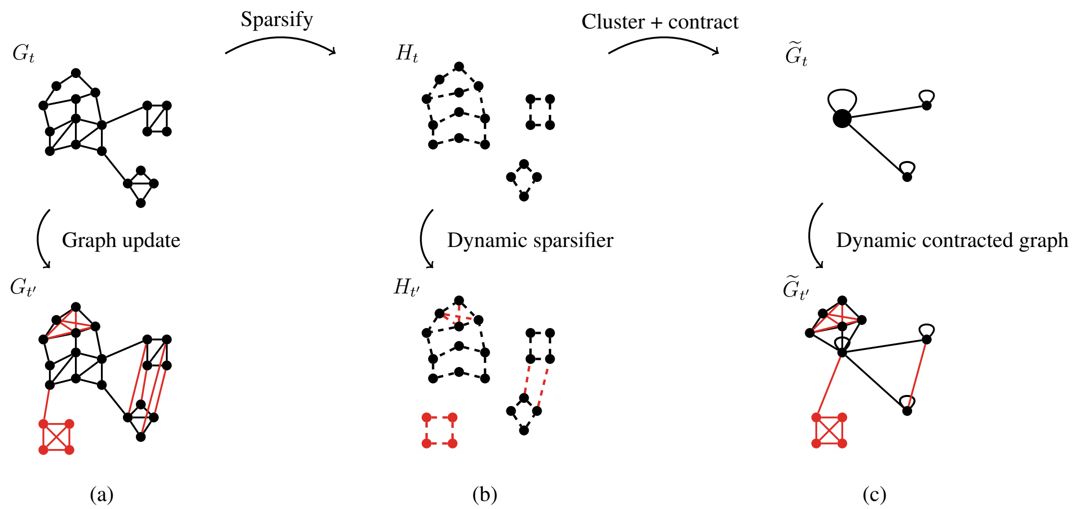

Figure 1. Illustration of our technique. The black and red edges in Figure (a) are the edges in G_t and the added ones in $G_{t'}$; the dashed black and red edges in Figure (b) are the ones added in H_t and $H_{t'}$; the black and red edges in Figure (c) are the ones in $\widetilde{G}_t$ and $\widetilde{G}_{t'}$.

2.2. Spectral Clustering

Spectral clustering is a popular clustering algorithm used in practice (Ng et al., 2001), and it can be described with a few lines of code (Algorithm 1).

Algorithm 1 SpectralClustering(G, k)

1: **Input:** Graph $G = (V, E, w)$, number of clusters $k \in \mathbb{N}$
2: **Output:** Partitioning $P_1, \ldots, P_k$
3: Compute eigenvectors $f_1, \ldots, f_k$ of $\mathcal{L}_G$
4: **for** $u \in V$ **do**
5: $\quad F(u) \leftarrow \frac{1}{\sqrt{\deg_G(u)}} \cdot (f_1(u), \ldots, f_k(u))^\intercal$
6: **end for**
7: $P_1, \ldots, P_k \leftarrow k$-means$(\{F(u)\}_{u \in V}, k)$
8: **Return** $P_1, \ldots, P_k$

To analyse the performance of spectral clustering, we examine the scenario in which there is a large gap between $\lambda_{k+1}(\mathcal{L}_G)$ and $\rho_G(k)$. By the higher-order Cheeger inequality, a low value of $\rho_G(k)$ ensures that V can be partitioned into k clusters, each of which has conductance at most $\rho_G(k)$; on the other hand, a large value of $\lambda_{k+1}(\mathcal{L}_G)$ implies that any $(k+1)$ partition of V would introduce some $A \subset V$ with $\Phi_G(A) \geqslant \rho_G(k+1) \geqslant \lambda_{k+1}(\mathcal{L}_G)/2$. Based on this, Peng et al. (2017) introduced the parameter

$$\Upsilon_G(k) \triangleq \frac{\lambda_{k+1}(\mathcal{L}_G)}{\rho_G(k)}, \tag{2}$$

and showed that a large value of $\Upsilon_G(k)$ is sufficient to guarantee a good performance of spectral clustering. They further showed that, for a graph G with m edges, spectral clustering runs in $O(m \cdot \log^{\beta} m)$ time for constant $\beta \in \mathbb{R}^+$.

For convenience of notation, we always order the output of spectral clustering by $P_1, \ldots, P_k$ such that $\mathrm{vol}_G(P_1) \leqslant \ldots \leqslant \mathrm{vol}_G(P_k)$.

2.3. Model for Dynamic Graph Clustering

We assume that the initial graph $G_1 = (V_1, E_1)$ with n_1 vertices satisfies $\lambda_{k+1}(\mathcal{L}_{G_1}) = \Omega(1)$ and $\rho_{G_1}(k) = O(k^{-8} \log^{-2\gamma}(n_1))$ for some constant $\gamma \in \mathbb{R}^+$. This condition is similar to lower bounding $\Upsilon_{G_1}(k)$, and ensures that the initial input graph G_1 has k well-defined clusters. After this, the underlying graph is updated through an edge insertion at each time, and let $G_t = (V_t, E_t)$ be the graph constructed at time t. We assume that every edge insertion introduces at most one new vertex; as such the underlying graph is always connected, and the number of vertices $n_t \triangleq |V_t|$ could increase over time. We further assume that, after every $\Theta(\log^{\gamma}(n_t))$ steps, there is time t' such that $G_{t'} = (V_{t'}, E_{t'})$ presents a well-defined structure of k' clusters, which is characterised by $\lambda_{k'+1}(\mathcal{L}_{G_{t'}}) = \Omega(1)$ and $\rho_{G_{t'}}(k') = O(k'^{-8} \cdot \log^{-2\gamma}(n_{t'}))$ for some $k' \in \mathbb{N}$.

Notice that, since both the number of vertices n_t in time t and the number of clusters could change, our above-defined *dynamic gap assumption* allows the underlying graph to gradually form a new cluster structure, e.g., $O(\log^{\gamma}(n_1))$ newly added vertices and their adjacent edges could initially form a small new cluster which gradually "grows" into a large one. On the other side, our assumption prevents the disappearance of the underlying graph's cluster-structure throughout the edge updates, which would make the objective function of a clustering algorithm ill-defined.

3. Dynamic Cluster-Preserving Sparsifiers

A graph sparsifier is a sparse representation of an input graph that inherits certain properties of the original dense graph, and their efficient construction plays a key role in designing a number of nearly-linear time graph algorithms. However, typical constructions of graph sparsifiers are based on fast Laplacian solvers, making them difficult to implement in practice. To overcome this, Sun & Zanetti (2019) studied a variant of graph sparsifiers for graph clustering, and introduced the notion of a cluster-preserving sparsifier:

Definition 3.1 (Cluster-preserving sparsifier)**.** Let $G = (V, E)$ be any graph with k clusters, and $\{S_i\}_{i=1}^k$ a k-way partition of G corresponding to $\rho_G(k)$. We call a re-weighted subgraph $H = (V, F \subset E, w_H)$ a cluster-preserving sparsifier of G if (i) $\Phi_H(S_i) = O(k \cdot \Phi_G(S_i))$ for $1 \leqslant i \leqslant k$, and (ii) $\lambda_{k+1}(\mathcal{L}_H) = \Omega(\lambda_{k+1}(\mathcal{L}_G))$.

To examine the two conditions of Definition 3.1, notice that graph $G = (V, E)$ has exactly k clusters if (i) G has k disjoint subsets $S_1, \ldots, S_k$ of low conductance, and (ii) any $(k+1)$-way partition of G would include some $A \subset V$ of high conductance, which would be implied by a lower bound on $\lambda_{k+1}(\mathcal{L}_G)$ due to (1). With the well-known eigen-gap heuristic and theoretical analysis on spectral clustering (Peng et al., 2017), these two conditions ensure that the k optimal clusters in G have low conductance in H as well.

3.1. The SZ Algorithm

We first present the algorithm in (Sun & Zanetti, 2019) for constructing a cluster-preserving sparsifier; we call it the SZ algorithm for simplicity. Given any input graph $G = (V, E)$, the algorithm computes

$$p_u(v) \triangleq \min\left\{C \cdot \frac{1}{\lambda_{k+1}(\mathcal{L}_G)} \cdot \frac{\log n}{\deg_G(u)}, 1\right\}$$

$$p_v(u) \triangleq \min\left\{C \cdot \frac{1}{\lambda_{k+1}(\mathcal{L}_G)} \cdot \frac{\log n}{\deg_G(v)}, 1\right\},$$

for every $e = \{u, v\}$, where $C \in \mathbb{R}^+$ is some constant. Then, the algorithm samples $e = \{u, v\}$ with probability $p_e \triangleq p_u(v) + p_v(u) - p_u(v) \cdot p_v(u)$, and sets the weight of every sampled $e = \{u, v\}$ in H as $w_H(u, v) \triangleq 1/p_e$. By setting F as the set of the sampled edges, the algorithm returns $H = (V, F, w_H)$. Sun & Zanetti (2019) proved that, with high probability, H has $\widetilde{O}(n)$ edges and is a cluster-preserving sparsifier of G.

On the other side, while Definition 3.1 shows that the optimal clusters S_i $(1 \leqslant i \leqslant k)$ of G have low conductance in H, it doesn't build the connection *from* the vertex sets of low conductance in H *to* the ones in G. In this paper, we prove that such a connection holds as well; this allows us to apply spectral clustering on H, and reason about the conductance of its returned clusters in G.

Lemma 3.2. *Let G be a graph with $\Upsilon_G(k) = \Omega(k)$ for some $k \in \mathbb{N}$ with optimal clusters $\{S_i\}_{i=1}^k$, and H its cluster preserving sparsifier. Let $\{P_i\}_{i=1}^k$ be the output of spectral clustering on H, and without loss of generality let the optimal correspondence of P_i be S_i for any $1 \leqslant i \leqslant k$. Then, it holds with high probability for any $1 \leqslant i \leqslant k$ that*

$$\mathrm{vol}_G(P_i \triangle S_i) = O\left(\frac{k^2}{\Upsilon_G(k)}\right) \cdot \mathrm{vol}_G(S_i),$$

$$\Phi_G(P_i) = O\left(\Phi_G(S_i) + \frac{k^2}{\Upsilon_G(k)}\right),$$

where $A \triangle B \triangleq (A \setminus B) \cup (B \setminus A)$.

3.2. Construction of Dynamic Cluster-Preserving Sparsifiers

Now we design an algorithm that constructs a cluster-preserving sparsifier under edge and vertex insertions, and our algorithm works as follows. Initially, for the input G_1 with n_1 vertices, a well-defined structure of k clusters and

$$\tau \geqslant \frac{C}{\lambda_{k+1}(\mathcal{L}_{G_1})} \tag{3}$$

for some constant $C \in \mathbb{R}^+$, we run the SZ algorithm and obtain a cluster-preserving sparsifier of G_1. In addition to storing the sparsifier H_1 of G_1, the algorithm employs the vector $\mathbf{sp}_1^*$ to store the values $\log n_1 / \deg_{G_1}(u)$ for every vertex u, which are used to sample adjacent edges of vertex u. See Algorithms 2 and 3 for formal description.

Algorithm 2 SampleEdge(e, G, τ)

1: **Input:** edge $e = \{u, v\}$, graph $G = (V, E)$ of n vertices, parameter $\tau \in \mathbb{R}^+$
Output: edge e' with weight $w(e')$
$p(u, v) \leftarrow p_u(v) + p_v(u) - p_u(v) \cdot p_v(u)$
Sample e with probability $p(u, v)$
2: **if** e is sampled **then**
3: $e' \leftarrow e$, $w(e') \leftarrow 1/p(u, v)$
4: **else**
5: $e' \leftarrow \emptyset$, $w(e') \leftarrow 0$
6: **end if**
7: **Return** e', $w(e')$

Next, given the graph G_t currently constructed at time t, its sparsifier H_t, and edge insertion $e = \{u, v\}$, the algorithm compares for every vertex w the parameter $\log n_{t+1} / \deg_{G_{t+1}}(w)$ with $\mathbf{sp}_t^*(w)$, the quantity used to sample the adjacent edges of w the last time, and checks whether the two values change significantly. If it is the case, then the used sampling probability is too far from the "correct" one when running the static SZ algorithm on G_{t+1}, and hence we resample all the edges adjacent to w with

Algorithm 3 $\mathsf{StaticSZSparsifier}(G, \tau)$

Input: $G = (V, E)$ of n vertices, parameter $\tau \in \mathbb{R}^+$
Output: Cluster preserving sparsifier $H = (V, F, w_H)$, degree list $\mathbf{sp}^*$
$F \leftarrow \emptyset$
for $e \in E$ **do**
  $e', w(e') \leftarrow \mathsf{SampleEdge}(e, G, \tau)$
  $F \leftarrow F \cup e', w_H(e) \leftarrow w(e')$
end for
$\mathbf{sp}^* \leftarrow \left\{ \frac{\log n}{\deg_G(u)} \mid u \in V \right\}$
Return $H, \mathbf{sp}^*$

Algorithm 4 $\mathsf{UpdateSparsifier}(G_t, H_t, \mathbf{sp}^*_t, e, \tau)$

Input: $G_t = (V_t, E_t)$, $H_t = (V_t, F_t, w_{H_t})$, $\mathbf{sp}^*_t$, incoming edge $e = \{u, v\}$, parameter τ
Output: $H_{t+1} = (V_{t+1}, F_{t+1}, w_{H_{t+1}})$, $\mathbf{sp}^*_{t+1}$
$V_{\mathrm{new}} \leftarrow \{u, v\} \setminus V_t$
$G_{t+1} \leftarrow (V_t \cup V_{\mathrm{new}}, E_t \cup e)$
$H_{t+1} \leftarrow (V_t \cup V_{\mathrm{new}}, F_t, w_{H_t})$
$\mathbf{sp}^*_{t+1} \leftarrow \mathbf{sp}^*_t$
if $V_{\mathrm{new}} \neq \emptyset$ **then**
  $e', w(e') \leftarrow \mathsf{SampleEdge}(e, G_{t+1}, \tau)$
  $F_{t+1} \leftarrow F_{t+1} \cup e', w_{H_{t+1}}(e) \leftarrow w(e')$
  if $u \in V_{\mathrm{new}}$ **then**
    $\mathbf{sp}^*_{t+1}(u) \leftarrow \frac{\log n_{t+1}}{\deg_{G_{t+1}}(u)}$
  end if
  if $v \in V_{\mathrm{new}}$ **then**
    $\mathbf{sp}^*_{t+1}(u) \leftarrow \frac{\log n_{t+1}}{\deg_{G_{t+1}}(v)}$
  end if
end if
$V_{\mathrm{doubled}} \leftarrow \left\{ \hat{v} \in V_{t+1} \setminus V_{\mathrm{new}} \mid \frac{\log n_{t+1}}{\deg_{G_{t+1}}(\hat{v})} > 2 \cdot \mathbf{sp}^*_t(\hat{v}) \text{ or } \frac{\log n_{t+1}}{\deg_{G_{t+1}}(\hat{v})} < \frac{\mathbf{sp}^*_t(\hat{v})}{2} \right\}$
if $|V_{\mathrm{doubled}}| > 0$ **then**
  for $\hat{u} \in V_{\mathrm{doubled}}$ **do**
    $F_{t+1} \leftarrow F_{t+1} \setminus E_{H_{t+1}}(\hat{u})$
    for $\hat{e} \in E_{G_{t+1}}$ adjacent to $\hat{u}$ **do**
      $\hat{e}', w(\hat{e}') \leftarrow \mathsf{SampleEdge}(\hat{e}, G_{t+1}, \tau)$
      $F_{t+1} \leftarrow F_{t+1} \cup \hat{e}', w_{H_{t+1}}(\hat{e}) \leftarrow w(\hat{e}')$
    end for
    $\mathbf{sp}^*_{t+1}(\hat{u}) \leftarrow \frac{\log n_{t+1}}{\deg_{G_{t+1}}(\hat{u})}$
  end for
else
  $e', w(e') \leftarrow \mathsf{SampleEdge}(e, G_{t+1}, \tau)$
  $F_{t+1} \leftarrow F_{t+1} \cup e', w_{H_{t+1}}(e) \leftarrow w(e')$
end if
Return $H_{t+1}, \mathbf{sp}^*_{t+1}$

the right sampling probability. Otherwise, we simply use the values stored in $\mathbf{sp}^*_t$ to sample the upcoming edge e, and include it in H_{t+1} if e is sampled. See Algorithm 4 for formal description[3], and Theorem 3.3 for its performance:

Theorem 3.3. *Let $G_1 = (V_1, E_1)$ be a graph with n_1 vertices and a well-defined structure of $k = \widetilde{O}(1)$ clusters, and $\{G_t\}$ the sequence of graphs of $\{n_t\}$ vertices constructed sequentially through an edge insertion at each time. Assuming graph G_T at time $T = O(\mathrm{poly}(n_1))$ has a well-defined structure of $\widetilde{O}(1)$ clusters and $n_T = O(\mathrm{poly}(n_1))$, Algorithm 4 returns a cluster-preserving sparsifier $H_T = (V_T, F_T, w_{H_T})$ of G_T with high probability, and $|F_T| = \widetilde{O}(n_T)$. The algorithm's amortised running time is $O(1)$ per edge update.*

4. Dynamic Spectral Clustering Algorithm

This section presents our main dynamic spectral clustering algorithm, and is organised as follows: In Section 4.1, we present the construction and update procedure of a contracted graph, which is the data structure that summarises the cluster structure of an underlying input graph and allows for quick updates to the clusters. The properties of dynamic contracted graphs are analysed in Section 4.2. We present the main algorithm and analyse its performance in Section 4.3.

4.1. Construction and Update of Contracted Graphs

For any input graph $G_t = (V_t, E_t)$ of n_t vertices, its dynamic cluster-preserving sparsifier $H_t = (V_t, F_t, w_{H_t})$, and its k clusters $P_1, \ldots, P_k$ returned from running spectral clustering on H_t, we apply Algorithm 5 to construct a contracted graph $\widetilde{G}_t = (\widetilde{V}_t, \widetilde{E}_t, w_{\widetilde{G}_t})$ of G_t. Notice that we introduce the set of *non-contracted vertices* $\widetilde{V}_t^{\mathrm{nc}} = \emptyset$, which will be used later.

Lemma 4.1. *The algorithm $\mathsf{ContractGraph}(H_t, \mathcal{P})$ returns $\widetilde{G}_t = (\widetilde{V}_t, \widetilde{E}_t, w_{\widetilde{G}_t})$ in $O(|F_t|)$ time.*

Next we discuss how the contracted graph is updated under edge and vertex insertions. Given the graph $G_t = (V_t, E_t)$ with n_t vertices that satisfies $\lambda_{k+1}(\mathcal{L}_{G_t}) = \Omega(1)$ and $\rho_{G_t}(k) = O(k^{-8} \log^{-2\gamma}(n_t))$ for some constant $\gamma \in \mathbb{R}^+$, its cluster-preserving sparsifier $H_t = (V_t, F_t, w_{H_t})$, the corresponding contracted graph $\widetilde{G}_t = (\widetilde{V}_t, \widetilde{E}_t, w_{\widetilde{G}_t})$, and the upcoming edge insertion $e = \{u, v\}$, we construct $\widetilde{G}_{t+1}$ from $\widetilde{G}_t$ as follows:

[3]Notice that, since $\lambda_{k+1}(\mathcal{L}_{G_t}) = \Omega(1)$ for any graph G_t exhibiting a well-defined structure of k clusters and it holds for G_T at time $T = O(\mathrm{poly}(n_1))$ that $n_T = O(\mathrm{poly}(n_1))$, i.e., $\log n_T = O(\log n_1)$, by setting C to be a sufficiently large constant, $\tau \cdot \log n_1$ is the right parameter for defining the sampling probability at time $T = O(\mathrm{poly}(n_1))$.

Algorithm 5 ContractGraph$(H_t, \mathcal{P})$

Input: Cluster preserving sparsifier $H_t = (V_t, F_t, w_{H_t})$, partition $\mathcal{P} = \{P_1, \ldots P_k\}$
Output: Contracted graph $\widetilde{G}_t = (\widetilde{V}_t, \widetilde{E}_t, w_{\widetilde{G}_t})$
Let p_i be a representative super vertex for each cluster $P_i \in \mathcal{P}$.
$\widetilde{V}_t^{\mathrm{c}} \leftarrow \{p_i \mid P_i \in \mathcal{P}\}$, $\widetilde{V}_t^{\mathrm{nc}} \leftarrow \emptyset$
$\widetilde{V}_t \leftarrow \widetilde{V}_t^{\mathrm{nc}} \cup \widetilde{V}_t^{\mathrm{c}}$
$\widetilde{E}_t \leftarrow \emptyset$
for $\{p_i, p_j\} \in \widetilde{V}_t^{\mathrm{c}} \times \widetilde{V}_t^{\mathrm{c}}$ **do**
    $\widetilde{E}_t \leftarrow \widetilde{E}_t \cup \{p_i, p_j\}$
    $w_{\widetilde{G}_t}(p_i, p_j) \leftarrow w_{H_t}(P_i, P_j)$
end for
Return $\widetilde{G}_t = (\widetilde{V}_t, \widetilde{E}_t, w_{\widetilde{G}_t})$

- If either u or v is a new vertex, the algorithm adds the vertex to $\widetilde{G}_t$ as a non-contracted vertex. The algorithm sets $V_{\mathrm{new}} = \{u, v\} \setminus V_t$, and $V_{t+1} = V_t \cup V_{\mathrm{new}}$.
- For every existing vertex $w \in \{u, v\} \setminus V_{\mathrm{new}}$ that belongs to some P_i, the algorithm checks whether $\deg_{G_{t+1}}(w) > 2 \cdot \deg_{G_r}(w)$, where $\deg_{G_r}(w)$ for $r \leqslant t$ is the degree of w when the contracted graph was constructed. If it is the case, the algorithm pulls w out of p_i, and adds it to $\widetilde{V}_{t+1}$, i.e., the uses a single vertex in $\widetilde{G}_{t+1}$ to represent w.
- The algorithm adjusts the edge weights in the contracted graph based on the type of the vertices. For instance, the algorithm sets $w_{\widetilde{G}_{t+1}}(u, v) = 1$ if both of u and v are non-contracted vertices, and decreases the value of $w_{\widetilde{G}_{t+1}}(P_u, P_u)$ if vertex u pulls out of $P_u \in \mathcal{P}$.

See Algorithm 7 in the appendix for the formal description of the algorithm UpdateContractedGraph$(G_t, \widetilde{G}_t, e)$.

Lemma 4.2. *The amortised time complexity of* UpdateContractedGraph$(G_t, \widetilde{G}_t, e)$ *is* $O(1)$.

4.2. Properties of the Contracted Graph

Now we analyse the properties of the contracted graph. Since the amortised time complexity for every edge update (Theorem 3.3 and Lemma 4.2) remains valid when we consider a sequence of edge updates at every time, without loss of generality let $G_{t'} = (V_t \cup V_{\mathrm{new}}, E_t \cup E_{\mathrm{new}})$ be the graph after a sequence of edge updates from $G_t = (V_t, E_t)$ with n_t vertices, and $\widetilde{G}_{t'}$ be the contracted graph of $G_{t'}$ constructed by sequentially running UpdateContractedGraph for each $e \in E_{\mathrm{new}}$. We assume that $|E_{\mathrm{new}}| \leqslant \log^{\gamma}(n_t)$ for some $\gamma \in \mathbb{R}^{+}$.

We first prove that the clusters returned by spectral clustering on H_t also have low conductance in G_t. Notice that, as the underlying graph G_t could be dense over time, running a clustering algorithm on its sparsifier H_t with $\widetilde{O}(n_t)$ edges is crucial to achieve the algorithm's quick update time.

Lemma 4.3. *It holds with high probability that* $\Phi_{H_t}(P_i) = O\left(k^2 \cdot \rho_{G_t}(k)\right)$ *and* $\Phi_{G_t}(P_i) = O\left(k^2 \cdot \rho_{G_t}(k)\right)$ *for all* $P_i \in \mathcal{P}$.

Next, we define the event $\mathcal{E}_1$ that

$$\Phi_{H_t}(P_i) = O\left(k^{-6} \cdot \log^{-2\gamma}(n_t)\right)$$

and

$$\Phi_{G_t}(P_i) = O\left(k^{-6} \cdot \log^{-2\gamma}(n_t)\right)$$

hold for all $P_i \in \mathcal{P}$. By the fact that $\lambda_{k+1}(G_t) = \Omega(1)$, $\rho_{G_t}(k) = O(k^{-8} \cdot \log^{-2\gamma}(n_t))$ and Lemma 4.3, $\mathcal{E}_1$ holds with high probability. We further define the event $\mathcal{E}_2$ that

$$(1/2) \cdot \deg_{G_t}(u) \leqslant \deg_{H_t}(u) \leqslant (3/2) \cdot \deg_{G_t}(u)$$

hold for all $u \in V_t$, and know from the proof of Theorem 3.3 that $\mathcal{E}_2$ holds with high probability. In the following we assume that both of $\mathcal{E}_1$ and $\mathcal{E}_2$ happen.

Next, we study the relationship between the cluster-structure in $G_{t'}$ and the one in $\widetilde{G}_{t'}$. Recall that the number of vertices in $\widetilde{G}_{t'}$ is much smaller than the one in $G_{t'}$. We first prove that there are ℓ disjoint vertex sets of low conductance in $G_{t'}$ if and only if there are ℓ such vertex sets in $\widetilde{G}_{t'}$.

Lemma 4.4. *The following statements hold:*

- *If* $\rho_{G_{t'}}(\ell) \leqslant \log^{-\alpha}(n_{t'})$ *holds for some* $\ell \in \mathbb{N}$ *and* $\alpha > 0$, *then* $\rho_{\widetilde{G}_{t'}}(\ell) = \max\left\{O\left(\log^{-0.9\alpha}(n_{t'})\right), O\left(k^{-6} \cdot \log^{-\gamma}(n_{t'})\right)\right\}$.
- *If* $\rho_{\widetilde{G}_{t'}}(\ell) \leqslant \log^{-\delta}(n_{t'})$ *holds for some* $\ell \in \mathbb{N}$ *and* $\delta > 0$, *then* $\rho_{G_{t'}}(\ell) = \max\left\{O\left(\log^{-\delta}(n_{t'})\right), O\left(k^{-6} \cdot \log^{-\gamma}(n_{t'})\right)\right\}$.

Secondly, we show that there is a close connection between $\lambda_{\ell+1}(\mathcal{L}_{G_{t'}})$ and $\lambda_{\ell+1}(\mathcal{L}_{\widetilde{G}_{t'}})$ for any $\ell \in \mathbb{N}$.

Lemma 4.5. *The following statements hold:*

- *If* $\lambda_{\ell+1}(\mathcal{L}_{\widetilde{G}_{t'}}) = \Omega(1)$ *for some* $\ell \in \mathbb{N}$, *then* $\lambda_{\ell+1}\left(\mathcal{L}_{G_{t'}}\right) = \Omega\left(\log^{-\alpha}(n_{t'})/\ell^{6}\right)$ *for constant* $\alpha > 0$.
- *If* $\lambda_{\ell+1}\left(\mathcal{L}_{G_{t'}}\right) = \Omega(1)$ *holds for some* $\ell \in \mathbb{N}$, *then* $\lambda_{\ell+1}(\mathcal{L}_{\widetilde{G}_{t'}}) = \Omega(1)$.

Lemmas 4.4 and 4.5 imply that the cluster-structures in $G_{t'}$ and $\widetilde{G}_{t'}$ are approximately preserved.

4.3. Main Algorithm

Our main algorithm consists of the preprocessing stage, update stage, and query stage. They are described as follows:

Preprocessing Stage. For the initial input graph $G_1 = (V_1, E_1)$, we apply (i) $\mathsf{StaticSZSparsifier}(G_1, \tau)$ to obtain $H_1 = (V_1, F_1, w_{H_1})$, (ii) $\mathsf{SpectralClustering}(H_1, k)$ to obtain initial partition $\mathcal{P} = \{P_1, \ldots P_k\}$, and (iii) $\mathsf{ContractGraph}(H_1, \mathcal{P})$ to obtain $\widetilde{G}_1 = (\widetilde{V}_1, \widetilde{E}_1)$.

Update Stage. When a new edge arrives at time t, we apply Algorithm 4 and the update procedure of the contracted graph (Section 4.1) to dynamically maintain H_t and $\widetilde{G}_t$.

Query Stage. When a query for a new clustering starts at time T, the algorithm performs the following operations, where γ is the constant satisfying $\gamma > \beta$ and $\gamma > 0.9\alpha$:

- For r being the last time at which $\widetilde{G}_t$ is recomputed, the algorithm checks if $T - r \leqslant \log^{\gamma}(n_r)$, i.e., the number of added edges after the last reconstruction of the contracted graph is less than $\log^{\gamma}(n_r)$. If it is the case, then the algorithm runs spectral clustering on the contracted graph $\widetilde{G}_T$.
- Otherwise, the algorithm runs spectral clustering on H_T. It also recomputes $\widetilde{G}_T$, by first computing $\widetilde{G}_{r'}$, where r' is the last time at which the dynamic gap assumption holds, and updating $\widetilde{G}_{r'}$ to $\widetilde{G}_T$ with the edge updates between time r' and T.

See Algorithm 6 for formal description.

Algorithm 6 $\mathsf{QuerySpecClustering}(G_T, H_T, \widetilde{G}_T, \gamma, \ell)$

```
1: Input: Graphs G_T, H_T, and G̃_T, γ ∈ ℝ⁺, and ℓ ∈ ℕ
2: Output: Partition 𝒫 = {P_1, ... P_ℓ}
3: Let r be the last time at which G̃_T is recomputed.
4: if T − r ⩽ log^γ(n_r) then
5:     P_1, ..., P_ℓ ← SpectralClustering(G̃_T, ℓ)
6:     Return {P_1, ..., P_ℓ}
7: else
8:     P_1, ..., P_ℓ ← SpectralClustering(H_T, ℓ)
9:     Recompute G̃_{r'}, where r' is the last time at which
       the dynamic gap assumption holds
10:    Update G̃_{r'} to G̃_T with the edge updates between
       time r' and T
11:    Return {P_1, ..., P_ℓ}
12: end if
```

Theorem 4.6. *Let $G_1 = (V_1, E_1)$ be a graph with n_1 vertices and $k = \widetilde{O}(1)$ clusters, and $\{G_t\}$ the sequence of graphs of $\{n_t\}$ vertices constructed through an edge insertion at each time satisfying the dynamic gap assumption. Assume that G_T at query time T has ℓ clusters, i.e., $\lambda_{\ell+1}(\mathcal{L}_{G_T}) = \Omega(1)$ and $\rho_{G_T}(\ell) = O(\ell^{-1}\log^{-\alpha}(n_T))$ for $\alpha \in \mathbb{R}^{+}$. Then, with high probability Algorithm 6 returns $P_1, \ldots P_\ell$ with $\Phi_{G_T}(P_i) = O\left(\ell \cdot \log^{-0.9\alpha}(n_T)\right)$ for every $1 \leqslant i \leqslant \ell$. The algorithm's running time for returning the clusters of G_1 is $\widetilde{O}(|E_1|)$. Afterwards, the algorithm's amortised update time is $O(1)$, and amortised query time is $o(n_T)$.*

Proof. The algorithm's running time and approximation guarantee on G_1 follows from (Macgregor & Sun, 2022), so we only need to analyse the dynamic update stage. We first analyse the conductance of every output P_i. Notice that, if Lines 4–6 of Algorithm 6 are executed, then by Lemmas 3.2, 4.4 and 4.5 the approximation guarantee holds. Otherwise, Lines 7–12 are executed, then by the dynamic gap condition and Lemma 3.2 the approximation guarantee holds as well.

Next, we prove the running time guarantee. The $O(1)$ amortised update time of H_t and $\widetilde{G}_t$ follows by Theorem 3.3 and Lemma 4.2. For the query at time T, notice that if Lines 4–6 are executed, then the query time is at most $O(|\widetilde{V}_T|^3) = O((k + \log^{\gamma}(n_T))^3) = \widetilde{O}(1)$. Note, the super vertices are used as sketches to quickly update the cluster assignment of each vertex; otherwise, Lines 7–12 are executed, and the query time is dominated by spectral clustering's time complexity of $O\left(n_T \cdot \log^{\beta}(n_T)\right)$. Since this only happens every $\log^{\gamma}(n_r) = O(\log^{\gamma}(n_T))$ edge updates, the amortised query time is $O(n_T \cdot \log^{\beta-\gamma}(n_T)) = o(n_T)$.

Finally, we show that the number of clusters ℓ can be identified with our claimed time complexity. Notice that, if Lines 4–6 of the algorithm are executed, then by Lemmas 4.4 and 4.5 we can detect the spectral gap in G_T using $\widetilde{G}_T$; hence we can choose ℓ in $o(n_t)$ time. Otherwise, Lines 7–12 are executed. In this case, we run spectral clustering with different values of ℓ' and find the correct value of ℓ (The same procedure is done to recompute $\widetilde{G}_{r'}$). Since there are $\widetilde{O}(1)$ clusters in total, we achieve the same query time guarantee. □

5. Experiments

We experimentally evaluate the performance of our algorithm on synthetic and real-world datasets. We report the clustering accuracy of all tested algorithms using the Adjusted Rand Index (ARI) (Rand, 1971), and compute the average and standard deviation over 10 independent runs. Algorithms were implemented in Python 3.12.1 and experiments were performed using a Lenovo ThinkPad T15G, with an Intel(R) Xeon(R) W-10855M CPU@2.80GHz processor and 126 GB RAM. Our code can be downloaded from https://github.com/steinarlaenen/Dynamic-Spectral-Clustering-With-Provable-Approximation-Guarantee.

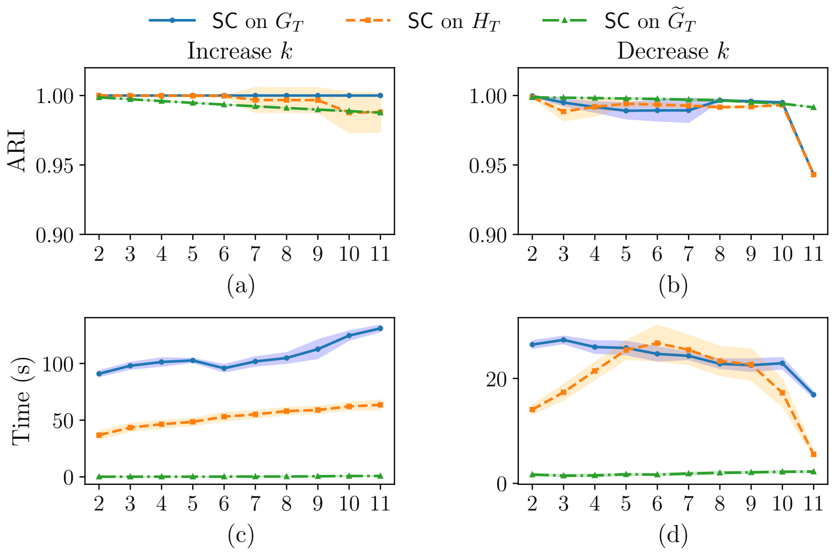

Figure 2. Results on the two versions of our dynamic SBM. Figures (a) and (b) report the average ARI score at each time T for the clustering results on G_T, H_T, and $\widetilde{G}_T$; Figures (c) and (d) report the running time in seconds at each time T. Shaded regions indicate the standard deviation.

5.1. Results on Synthetic Data

We study graphs generated from the stochastic block model (SBM), and introduce two dynamic extensions to generate new clusters and merge existing clusters.

SBM with increasing number of clusters. We generate the first graph G_1 based on the standard SBM, and set $k = 10$ and the number of vertices in each cluster $\{S_i\}_{i=1}^k$ as $n_k = 10,000$. For every pair of $u \in S_i$ and $v \in S_j$ we include edge $\{u, v\}$ with probability p if $i = j$, and with probability q if $i \neq j$.

To update the graph, we generate a batch of edge updates in two steps: first, we randomly select a subset $Q \subset V(G_1)$ such that $|Q| = n_{\text{new}} = 400$, and for any $u, v \in Q$ we include edge $e = \{u, v\}$ in the graph with probability r_1; setting r_1 sufficiently large ensures that the set Q forms a new cluster in the graph. Second, for any $u, v \in V(G_1)$ we include edge $e = \{u, v\}$ with probability s. The edges sampled from these two processes form one edge update batch. We sample 10 such batches (ensuring no new clusters overlap), each inducing a new cluster and additional noise.

To cluster each G_T, we run spectral clustering (SC) on three graphs:

1. We run spectral clustering on the full graph G_T.
2. We construct the contracted graph $\widetilde{G}_1$ at time $T = 1$, and incrementally update $\widetilde{G}_1$ using the procedure described in Section 4.1. Then, we run spectral clustering on each $\widetilde{G}_T$.
3. We construct a cluster-preserving sparsifier H_1 using Algorithm 3, which we dynamically update using Algorithm 4 with sampling parameter $\tau = 3$, and cluster each subsequent H_T.

At each time T, we run spectral clustering with $k = 10 + T - 1$ on all three graphs, and report the running times and ARI scores. We set $p = 0.1$, $q = 0.01$, $r_1 = 0.95$, and $s = 0.00001$, and plot the results in the left plots of Figure 2. We can see that at every time T, spectral clustering on G_T returns the perfect clustering, and spectral clustering on $\widetilde{G}_T$ and H_T returns marginally worse clustering results. On the running time, we see that running spectral clustering on G_T, H_T and $\widetilde{G}_T$ takes around 100 seconds, 50 seconds, and less than 1 second respectively. This highlights that our algorithm returns nearly-optimal clusters with much faster running time than running spectral clustering on G_T or H_T.

Next, we compare the spectral gaps of $\mathcal{L}_{G_T}$ and $\mathcal{L}_{\widetilde{G}_T}$ for every T, and Table 1 reports that these gaps are well preserved. This demonstrates that, as what we prove earlier, the new cluster-structure of G_T can be indeed identified from $\widetilde{G}_T$.

Table 1. Spectral gaps in $\mathcal{L}_{G_T}$ and $\mathcal{L}_{\widetilde{G}_T}$ for SBM with increasing number of clusters. We report $\lambda_{k+T}(\mathcal{L}_{G_T})/\lambda_{k+T-1}(\mathcal{L}_{G_T})$ and $\lambda_{k+T}(\mathcal{L}_{\widetilde{G}_T})/\lambda_{k+T-1}(\mathcal{L}_{\widetilde{G}_T})$ at each time T.

T	2	3	4	5	6	7	8	9	10	11
G_T	6.3	5.8	5.8	5.7	5.7	5.7	5.6	5.6	5.6	5.5
$\widetilde{G}_T$	9.3	9.2	9.0	8.7	8.5	8.1	7.8	7.5	7.1	6.8

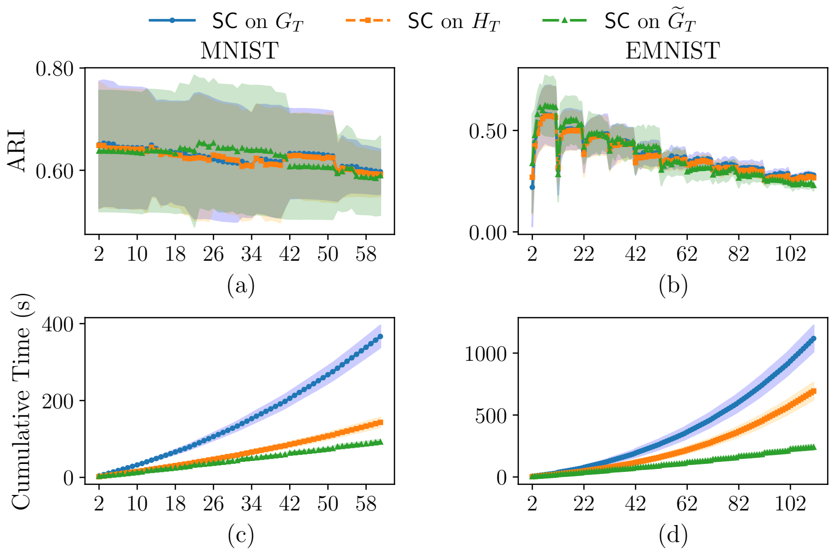

Figure 3. Results on MNIST and EMNIST. Figures (a) and (b) report the average ARI scores at each time T for the clustering results on G_T, H_T, and $\widetilde{G}_T$; Figures (c) and (d) report the average cumulative running time in seconds at each time T. Shaded regions indicate the standard deviation.

SBM with decreasing number of clusters. We set $k = 25$, and the first graph G_1 is generated based on the standard SBM with parameters p and q. For clusters $\{S_i\}_{i=1}^{5}$ we set $|S_i| = 20,000$, and for $\{S_i\}_{i=6}^{25}$ we set $|S_i| = 500$; hence there are 5 large and 20 small clusters.

To update the graph, we generate a batch of edge updates as follows: we randomly choose two clusters S_i and S_j such that $|S_i| = |S_j| = 500$, and for any $u \in S_i$ and $v \in S_j$ we include edge $e = \{u, v\}$ in the graph with probability r_2. Setting r_2 sufficiently large ensures that clusters S_i and S_j merge. Similarly as before, for any $u, v \in V(G_1)$ we also include edge $e = \{u, v\}$ with probability s. All the edges sampled by these two processes form a single batch update. We sample 10 such batches, and there are $k = 15$ clusters at final time $T = 11$. At each time T, we run spectral clustering with $k = 25 - T + 1$ on all three graphs. We set $p = 0.1$, $q = 0.001$, $r_2 = 0.95$, and $s = 0.00001$, and plot the results in the right plots of Figure 2.

Similar to the SBM with increasing number of clusters, at every time T, spectral clustering on all three graphs returns similar results. We further see that spectral clustering on $\widetilde{G}_T$ has lower running time than the one on G_T and H_T. The spectral gaps in G_T and $\widetilde{G}_T$ are reported in Table 2.

Table 2. Spectral gaps in $\mathcal{L}_{G_T}$ and $\mathcal{L}_{\widetilde{G}_T}$ for SBM with decreasing number of clusters. We report $\lambda_{k-T+2}(\mathcal{L}_{G_T})/\lambda_{k-T+1}(\mathcal{L}_{G_T})$ and $\lambda_{k-T+2}(\mathcal{L}_{\widetilde{G}_T})/\lambda_{k-T+1}(\mathcal{L}_{\widetilde{G}_T})$ at each time T.

T	2	3	4	5	6	7	8	9	10	11
G_T	4.5	4.4	4.2	4.0	4.0	3.9	3.8	3.8	3.8	3.6
$\widetilde{G}_T$	8.3	8.0	7.5	7.4	7.4	7.0	6.8	6.3	5.8	5.4

5.2. Results on Real-World Data

We further evaluate our algorithm on the MNIST dataset (Lecun et al., 1998), which consists of 10 classes of handwritten digits and has $70,000$ images, and the "letter" subset of the EMNIST dataset (Cohen et al., 2017), which consists of 26 classes of handwritten letters and has $145,600$ images. We construct a k-nearest neighbour graph for each dataset, and set $k = 100$ (resp. $k = 200$) for MNIST (resp. EMNIST).

We select four classes (clusters) at random; the chosen vertices and adjacent edges in the k-nearest neighbour graph form G_1. To construct the sequence of updates, we select one new cluster (resp. two) at random for MNIST (resp. EMNIST), and add the edges inside the new cluster as well as the ones between the new and existing clusters. We randomly partition these new edges into 10 batches of equal size, and add these to the graph sequentially. We recompute $\widetilde{G}_T$ after one class (resp. two) is streamed for MNIST (resp. EMNIST), and report the results in Figure 3. The update/reconstruction time is included in the running time.

Our experiments on real-world data further confirm that, as the size of the underlying graph and its number of clusters increase over time, our designed algorithm has much lower running time compared with repeated execution of spectral clustering, while producing comparable clustering results.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

This work is supported by an EPSRC Early Career Fellowship (EP/T00729X/1). Part of this work was done when He Sun was visiting the Simons Institute for the Theory of Computing in Fall 2023.

References

Beimel, A., Kaplan, H., Mansour, Y., Nissim, K., Saranurak, T., and Stemmer, U. Dynamic algorithms against an adaptive adversary: generic constructions and lower bounds. In *54th Annual ACM Symposium on Theory of Computing (STOC'22)*, pp. 1671–1684, 2022.

Bernstein, A., van den Brand, J., Gutenberg, M. P., Nanongkai, D., Saranurak, T., Sidford, A., and Sun, H. Fully-dynamic graph sparsifiers against an adaptive adversary. In *49th International Colloquium on Automata, Languages, and Programming (ICALP'22)*, pp. 20:1–20:20, 2022.

Chung, F. and Lu, L. Concentration inequalities and martingale inequalities: a survey. *Internet mathematics*, 3(1): 79–127, 2006.

Cohen, G., Afshar, S., Tapson, J., and Schaik, A. V. Emnist: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926, 2017.

Dhanjal, C., Gaudel, R., and Clémençon, S. Efficient eigenupdating for spectral graph clustering. *Neurocomputing*, 131:440–452, 2014.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, J. R., Gharan, S. O., and Trevisan, L. Multiway spectral partitioning and higher-order Cheeger inequalities. *Journal of the ACM*, 61(6):1–30, 2014.

Macgregor, P. and Sun, H. A tighter analysis of spectral clustering, and beyond. In *39th International Conference on Machine Learning (ICML'22)*, pp. 14717–14742, 2022.

Martin, L., Loukas, A., and Vandergheynst, P. Fast approximate spectral clustering for dynamic networks. In *35th International Conference on Machine Learning (ICML'18)*, pp. 3420–3429, 2018.

Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 15 (NeurIPS'01)*, pp. 849–856, 2001.

Ning, H., Xu, W., Chi, Y., Gong, Y., and Huang, T. Incremental spectral clustering with application to monitoring of evolving blog communities. In *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*, pp. 261–272, 2007.

Peng, R., Sun, H., and Zanetti, L. Partitioning Well-Clustered Graphs: Spectral Clustering Works! *SIAM Journal on Computing*, 46(2):710–743, 2017.

Rand, W. M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

Saranurak, T. and Wang, D. Expander decomposition and pruning: Faster, stronger, and simpler. In *30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*, pp. 2616–2635, 2019.

Sun, H. and Zanetti, L. Distributed graph clustering and sparsification. *ACM Transactions on Parallel Computing*, 6(3):17:1–17:23, 2019.

Thorup, M. Fully-dynamic min-cut. *Combinatorica*, 27(1): 91–127, 2007.

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

von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing volume*, 17(4):395–416, 2007.

A. Omitted Details from Section 3

The section presents the details omitted from Section 3, and is organised as follows. We prove Lemma 3.2 in Section A.1, and prove Theorem 3.3 in Section A.2.

A.1. Proof of Lemma 3.2

We first prove a structure theorem. We define the vectors $\chi_1, \ldots \chi_k$ to be the indicator vectors of the optimal clusters $S_1, \ldots, S_k$ in G, where $\chi_i(u) = 1$ if $u \in S_i$, and $\chi_i = 0$ otherwise. We further use $\bar{g}_1, \ldots, \bar{g}_k$ to denote the indicator vectors of the optimal clusters $S_1, \ldots S_k$ in G, normalised by the degrees in H, i.e.,

$$\bar{g}_i \triangleq \frac{D_H^{\frac{1}{2}} \chi_i}{\| D_H^{\frac{1}{2}} \chi_i \|}. \tag{4}$$

Theorem A.1. *Let $S_1, \ldots, S_k$ be a k-way partition of G achieving $\rho_G(k)$, and $\Upsilon_G(k) = \Omega(k)$, and $\{f_i\}_{i=1}^k$ be first k eigenvectors of $\mathcal{L}_H$ and let $\{\bar{g}_i\}_{i=1}^k$ be defined as in (4) above. Then, the following statements hold:*

1. *For any $i \in [k]$, there is $\widehat{f}_i \in \mathbb{R}^n$, which is a linear combination of $f_1, \ldots, f_k$, such that $\|\bar{g}_i - \widehat{f}_i\|^2 = O\left(k/\Upsilon_G(k)\right)$.*
2. *There are vectors $\widehat{g}_1, \ldots, \widehat{g}_k$, each of which is a linear combination of $\bar{g}_1, \ldots, \bar{g}_k$, such that $\sum_{i=1}^k \|f_i - \widehat{g}_i\|^2 = O\left(k^2/\Upsilon_G(k)\right)$.*

Proof. Let $\widehat{f}_i = \sum_{j=1}^k \langle \bar{g}_i, f_j \rangle f_j$, and we write $\bar{g}_i$ as a linear combination of the vectors $f_1, \ldots, f_n$ by $\bar{g}_i = \sum_{j=1}^n \langle \bar{g}_i, f_j \rangle f_j$. Since $\widehat{f}_i$ is a projection of $\bar{g}_i$, we have that $\bar{g}_i - \widehat{f}_i$ is perpendicular to $\widehat{f}_i$ and

$$\left\| \bar{g}_i - \widehat{f}_i \right\|^2 = \|\bar{g}_i\|^2 - \left\| \widehat{f}_i \right\|^2 = \left(\sum_{j=1}^n \langle \bar{g}_i, f_j \rangle^2 \right) - \left(\sum_{j=1}^k \langle \bar{g}_i, f_j \rangle^2 \right) = \sum_{j=k+1}^n \langle \bar{g}_i, f_j \rangle^2.$$

Now, let us consider the quadratic form

$$\begin{aligned} \bar{g}_i^\intercal \mathcal{L}_H \bar{g}_i &= \left(\sum_{j=1}^n \langle \bar{g}_i, f_j \rangle f_j^\intercal \right) \mathcal{L}_H \left(\sum_{j=1}^n \langle \bar{g}_i, f_j \rangle f_j \right) \\ &= \sum_{j=1}^n \langle \bar{g}_i, f_j \rangle^2 \lambda_j(\mathcal{L}_H) \\ &\geqslant \lambda_{k+1}(\mathcal{L}_H) \left\| \bar{g}_i - \widehat{f}_i \right\|^2 \\ &= \Omega\left(\lambda_{k+1}(\mathcal{L}_G)\right) \left\| \bar{g}_i - \widehat{f}_i \right\|^2, \end{aligned} \tag{5}$$

where the second to last inequality follows by the fact that $\lambda_i(\mathcal{L}_H) \geqslant 0$ holds for any $1 \leqslant i \leqslant n$, and the last inequality follows because H is a cluster preserving sparsifier of G. This gives us that

$$\begin{aligned} \bar{g}_i^\intercal \mathcal{L}_H \bar{g}_i &= \sum_{(u,v) \in E_H} w_H(u,v) \left(\frac{\bar{g}_i(u)}{\sqrt{\deg_H(u)}} - \frac{\bar{g}_i(v)}{\sqrt{\deg_H(v)}} \right)^2 \\ &= \sum_{(u,v) \in E_H} w_H(u,v) \left(\frac{\chi_i(u)}{\sqrt{\mathrm{vol}_H(S_i)}} - \frac{\chi_i(v)}{\sqrt{\mathrm{vol}_H(S_i)}} \right)^2 \\ &= \frac{w_H(S_i, V \setminus S_i)}{\mathrm{vol}_H(S_j)} \\ &= O\left(k \cdot \rho_G(k)\right), \end{aligned} \tag{6}$$

where the last line holds because H is a cluster preserving sparsifier of G. Combining (5) with (6), we have that

$$\left\|\bar{g}_i - \widehat{f}_i\right\|^2 \leqslant \frac{\bar{g}_i^\intercal \mathcal{L}_H \bar{g}_i}{\Omega\left(\lambda_{k+1}(\mathcal{L}_G)\right)} \leqslant \frac{O\left(k \cdot \rho_G(k)\right)}{\Omega\left(\lambda_{k+1}(\mathcal{L}_G)\right)} = O\left(\frac{k}{\Upsilon_G(k)}\right),$$

which proves the first statement of the theorem.

Now we prove the second statement. We define for any $1 \leqslant i \leqslant k$ that $\widehat{g}_i = \sum_{j=1}^{k} \langle f_i, \bar{g}_j \rangle \bar{g}_j$, and have that

$$\begin{aligned}
\sum_{i=1}^{k} \|f_i - \widehat{g}_i\|^2 &= \sum_{i=1}^{k} \left(\|f_i\|^2 - \|\widehat{g}_i\|^2\right) \\
&= k - \sum_{i=1}^{k}\sum_{j=1}^{k} \langle \bar{g}_j, f_i \rangle^2 \\
&= \sum_{j=1}^{k} \left(1 - \sum_{i=1}^{k} \langle \bar{g}_j, f_i \rangle^2\right) \\
&= \sum_{j=1}^{k} \left(\|\bar{g}_j\|^2 - \left\|\widehat{f}_j\right\|^2\right) \\
&= \sum_{j=1}^{k} \left\|\bar{g}_j - \widehat{f}_j\right\|^2 \\
&= \sum_{j=1}^{k} O\left(\frac{k}{\Upsilon_G(k)}\right) \\
&= O\left(\frac{k^2}{\Upsilon_G(k)}\right),
\end{aligned}$$

where the last inequality follows by the first statement of Theorem A.1. □

Proof Sketch of Lemma 3.2. The proof follows Theorem 1.2 of (Peng et al., 2017) and Theorem 2 of (Macgregor & Sun, 2022), which imply that every returned cluster P_i $(1 \leqslant i \leqslant k)$ from spectral clustering on G satisfies that

$$\mathrm{vol}_G(P_i \triangle S_i) = O\left(k \cdot \frac{\mathrm{vol}_G(S_i)}{\Upsilon_G(k)}\right)$$

and

$$\Phi_G(P_i) = O\left(\Phi_G(S_i) + \frac{k}{\Upsilon_G(k)}\right),$$

where S_i is the optimal correspondence of P_i in G. Since H is a cluster-preserving sparsifier of G, we know that $\rho_H(k) = O(k \cdot \rho_G(k))$ and $\lambda_{k+1}(\mathcal{L}_H) = \Omega(\lambda_{k+1}(\mathcal{L}_G))$, which implies that

$$\Upsilon_H(k) = \frac{\lambda_{k+1}(\mathcal{L}_H)}{\rho_H(k)} = \frac{\Omega(\lambda_{k+1}(\mathcal{L}_G))}{O(k \cdot \rho_G(k))} = \Omega\left(\frac{1}{k} \cdot \Upsilon_G(k)\right). \tag{7}$$

On the other side, compared with their work, we need to apply the bottom k eigenvectors of $\mathcal{L}_H$ instead of $\mathcal{L}_G$ to run spectral clustering. As such, combining (7) with the adjusted structure theorem (Theorem A.1) one can prove Lemma 3.2 using the proof technique from (Macgregor & Sun, 2022) and (Peng et al., 2017). □

A.2. Proof of Theorem 3.3

Let

$$E_{\text{resampled}} \triangleq e \bigcup \left\{\{u, v\} \in E_{G_{t+1}} \mid u \in V_{\text{doubled}}\right\}$$

be the set of all the edges that have been (re)-sampled by Algorithm 4, and

$$E_{\mathrm{old}} \triangleq E_{t+1} \setminus E_{\mathrm{resampled}}.$$

Moreover, let

$$p_u^{(t+1)}(v) \triangleq \min\left\{\frac{\tau \cdot \log(n_{t+1})}{\deg_{G_{t+1}}(u)}, 1\right\}$$

be the "ideal" sampling probability of an edge $\{u, v\}$ if one runs the SZ algorithm from the scratch on G_{t+1}, and let

$$q^{(t+1)}(u, v) \triangleq p_u^{(t+1)}(v) + p_v^{(t+1)}(u) - p_u^{(t+1)}(v) \cdot p_v^{(t+1)}(u)$$

be the probability that edge e is sampled if one runs the SZ algorithm from scratch at time $t+1$. For any edge $\{u, v\}$, we use

- $\widetilde{q}(u, v) \triangleq q^{(r)}(u, v)$
- $\widetilde{p}_u(v) \triangleq p_u^{(r)}(v)$
- $\widetilde{p}_v(u) \triangleq p_v^{(r)}(u)$

for some $1 \leqslant r \leqslant t+1$ to denote the sampling probability last used for edge $\{u, v\}$ throughout the sequence of edge updates. Hence, we have $\widetilde{q}(u, v) = q^{(t+1)}(u, v)$ if $\{u, v\} \in E_{\mathrm{resampled}}$, and $\widetilde{q}(u, v) = q^{(r)}(u, v)$ for some $1 \leqslant r \leqslant t+1$ if edge $\{u, v\} \in E_{\mathrm{old}}$. By the algorithm description (Line 16 in Algorithm 4), we know that

$$\frac{\tau \cdot \log(n_{t+1})}{2 \cdot \deg_{G_{t+1}}(u)} \leqslant \frac{\tau \cdot \log(n_r)}{\deg_{G_r}(u)} \leqslant \frac{2 \cdot \tau \cdot \log(n_{t+1})}{\deg_{G_{t+1}}(u)}. \tag{8}$$

The following two concentration inequalities will be used in our analysis.

Lemma A.2 (Bernstein's Inequality (Chung & Lu, 2006))**.** *Let $X_1, \ldots, X_n$ be independent random variables such that $|X_i| \leqslant M$ for any $i \in \{1, \ldots, n\}$. Let $X = \sum_{i=1}^n X_i$, and $R = \sum_{i=1}^n \mathbb{E}\left[X_i^2\right]$. Then, it holds that*

$$\mathbb{P}\left[|X - \mathbb{E}\left[X\right]| \geqslant t\right] \leqslant 2\exp\left(-\frac{t^2}{2(R + Mt/3)}\right).$$

Lemma A.3 (Matrix Chernoff Bound (Tropp, 2012))**.** *Consider a finite sequence $\{X_i\}$ of independent, random, PSD matrices of dimension d that satisfy $\|X_i\| \leqslant R$. Let $\mu_{\min} \triangleq \lambda_{\min}(\mathbb{E}\left[\sum_i X_i\right])$ and $\mu_{\max} \triangleq \lambda_{\max}(\mathbb{E}\left[\sum_i X_i\right])$. Then, it holds that*

$$\mathbb{P}\left[\lambda_{\min}\left(\sum_i X_i\right) \leqslant (1-\delta)\mu_{\min}\right] \leqslant d\left(\frac{\mathrm{e}^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu_{\min}/R}$$

for $\delta \in [0, 1]$, and

$$\mathbb{P}\left[\lambda_{\max}\left(\sum_i X_i\right) \geqslant (1+\delta)\mu_{\max}\right] \leqslant d\left(\frac{\mathrm{e}^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu_{\max}/R}$$

for $\delta \geqslant 0$.

We first prove the following result on the relationship of cut values between G_{t+1} and H_{t+1}.

Lemma A.4. *Let G_{t+1} be a graph, and H_{t+1} the sparsifier returned by Algorithm 4. Suppose for every $\{u, v\} \in E_{t+1}$ that $\widetilde{p}_u(v) < 1$, then it holds for any non-empty subset $A \subset V_{t+1}$ that*

$$\begin{aligned}
&\mathbb{P}\left[|w_{H_{t+1}}(A, V_{t+1} \setminus A) - w_{G_{t+1}}(A, V_{t+1} \setminus A)| \geqslant \frac{1}{2} \cdot w_{G_{t+1}}(A, V_{t+1} \setminus A)\right] \\
&\leqslant 2 \cdot \exp\left(\frac{-\tau \cdot \log n_{t+1} \cdot w_{G_{t+1}}(A, V_{t+1} \setminus A)}{10 \cdot \mathrm{vol}_{G_{t+1}}(A)}\right)
\end{aligned}$$

Proof. For any edge $e = \{u, v\}$, we define the random variable Y_e by

$$Y_e \triangleq \begin{cases} \frac{1}{\widetilde{q}(u,v)} & \text{with probability } \widetilde{q}(u,v) \\ 0 & \text{otherwise.} \end{cases}$$

We also define

$$Z \triangleq \sum_{e \in E_{G_{t+1}}(A, V_{t+1} \setminus A)} Y_e,$$

and have that

$$\mathbb{E}[\,Z\,] = \sum_{e=\{u,v\} \in E_{G_{t+1}}(A, V_{t+1} \setminus A)} \mathbb{E}[\,Y_e\,] = \sum_{e=\{u,v\} \in E_{G_{t+1}}(A, V_{t+1} \setminus A)} \widetilde{q}(u,v) \cdot \widetilde{q}(u,v)^{-1} = w_{G_{t+1}}(A, V_{t+1} \setminus A).$$

To prove a concentration bound on this degree estimate, we apply the Bernstein inequality (Lemma A.2), for which we need to bound the second moment

$$R \triangleq \sum_{e=\{u,v\} \in E_{G_{t+1}}(A, V_{t+1} \setminus A)} \mathbb{E}[\,Y_e^2\,].$$

We get that

$$\begin{aligned} R &= \sum_{e=\{u,v\} \in E_{G_{t+1}}(A, V_{t+1} \setminus A)} \widetilde{q}(u,v) \cdot \left(\frac{1}{\widetilde{q}(u,v)} \right)^2 = \sum_{e=\{u,v\} \in E_{G_{t+1}}(A, V_{t+1} \setminus A)} \frac{1}{\widetilde{q}(u,v)} \\ &\leqslant \sum_{e=\{u,v\} \in E_{G_{t+1}}(A, V_{t+1} \setminus A)} \frac{1}{\widetilde{p}_u(v)} \qquad (9) \\ &= \sum_{e=\{u,v\} \in E_{G_{t+1}}(A, V_{t+1} \setminus A)} \frac{2 \cdot \deg_{G_{t+1}}(u)}{\tau \cdot \log(n_{t+1})} \qquad (10) \\ &\leqslant \frac{2 \cdot \Delta_{G_{t+1}}(A)}{\tau \cdot \log(n_{t+1})} \cdot \sum_{e=\{u,v\} \in E_{G_{t+1}}(A, V_{t+1} \setminus A)} 1 \\ &= \frac{2 \cdot \Delta_{G_{t+1}}(A) \cdot w_{G_{t+1}}(A, V_{t+1} \setminus A)}{\tau \cdot \log(n_{t+1})}, \end{aligned}$$

where $\Delta_{G_{t+1}}(A) \triangleq \max_{u \in A} \deg_{G_{t+1}}(u)$, (9) holds since $\widetilde{q}(u,v) = \widetilde{p}_u(v) + \widetilde{p}_v(u) - \widetilde{p}_u(v) \cdot \widetilde{p}_v(u) \geqslant \widetilde{p}_u(v)$, and (10) holds because of (8).

Note, by (8), for any edge $e = \{u, v\} \in E_{G_{t+1}}(A, V_{t+1} \setminus A)$ we have that

$$0 \leqslant Y_e = \frac{1}{\widetilde{q}(u,v)} \leqslant \frac{1}{\widetilde{p}_u(v)} \leqslant \frac{2 \cdot \Delta_{G_{t+1}}(A)}{\tau \cdot \log n_{t+1}}.$$

Then, by applying Bernstein's inequality, we have that

$$\begin{aligned} \mathbb{P}\left[|Z - \mathbb{E}[\,Z\,]| \geqslant \frac{1}{2}\mathbb{E}[\,Z\,] \right] &\leqslant 2 \cdot \exp\left(- \frac{w_{G_{t+1}}(A, V_{t+1} \setminus A)^2/4}{\frac{\Delta_{G_{t+1}}(A) \cdot w_{G_{t+1}}(A, V_{t+1} \setminus A)}{\tau \cdot \log(n_{t+1})} + \frac{\Delta_{G_{t+1}}(A) \cdot w_{G_{t+1}}(A, V_{t+1} \setminus A)}{3 \cdot \tau \cdot \log(n_{t+1})}} \right) \qquad (11) \\ &= 2 \cdot \exp\left(- \frac{\tau \cdot \log(n_{t+1}) \cdot 3 \cdot w_{G_{t+1}}(A, V_{t+1} \setminus A)}{16 \cdot \Delta_{G_{t+1}}(A)} \right) \qquad (12) \\ &\leqslant 2 \cdot \exp\left(- \frac{\tau \cdot \log(n_{t+1}) \cdot w_{G_{t+1}}(A, V_{t+1} \setminus A)}{10 \cdot \mathrm{vol}_{G_{t+1}}(A)} \right), \qquad (13) \end{aligned}$$

which proves the lemma. □

Proof of Theorem 3.3. We first analyse the number of edges in H_{t+1}, i.e., the size of F_{t+1}. We have that

$$\sum_{u\in V_{t+1}}\sum_{e=\{u,v\}\in E_{G_{t+1}}}\widetilde{p}_u(v)\leqslant\sum_{u\in V_{t+1}}\sum_{e=\{u,v\}\in E_{G_{t+1}}}\frac{2\cdot\tau\cdot\log n_{t+1}}{\deg_{G_{t+1}}(u)}=2\cdot\tau\cdot n_{t+1}\cdot\log n_{t+1},$$

where the first inequality holds by (8). Therefore, it holds by the Markov inequality that the number of edges $\{u,v\}$ with $\widetilde{p}_u(v)\geqslant 1$ is $O\left(\tau\cdot n_{t+1}\log n_{t+1}\right)$. Without loss of generality, we assume that these edges are included in F_{t+1}, and we assume for the remaining part of the proof that it holds that $\widetilde{p}_u(v)<1$.

We now show that the degrees of the vertices in G_{t+1} are approximately preserved in H_{t+1}. Let u be an arbitrary vertex of G_{t+1}. Observing that $\mathrm{vol}_{G_{t+1}}(u)=w_{G_{t+1}}(u,V\setminus u)=\deg_{G_{t+1}}(u)$ and $w_{H_{t+1}}(u,V_{t+1}\setminus u)=\deg_{H_{t+1}}(u)$, by Lemma A.4 it holds that

$$\begin{aligned}\mathbb{P}\left[\left|\deg_{H_{t+1}}(u)-\deg_{G_{t+1}}(u)\right|\geqslant\frac{1}{2}\deg_{G_{t+1}}(u)\right]&=2\exp\left(-(1/10)\cdot\tau\cdot\log n_{t+1}\right)\\&=2\exp\left(-(1/10)\cdot(\log n_{t+1}\cdot C)/\lambda_{k+1}(\mathcal{L}_{G_{t+1}})\right)\\&=o(1/n_{t+1}^2).\end{aligned}$$

Hence, by taking C to be sufficiently large and the union bound, it holds with high probability that the degrees of all the vertices in G_{t+1} are preserved in H_{t+1} up to a constant factor. Throughout the rest of the proof, we assume this is the case. This implies for any subset $A\subseteq V_{t+1}$ that $\mathrm{vol}_{H_{t+1}}(A)=\Theta(\mathrm{vol}_{G_{t+1}}(A))$.

Secondly, we prove it holds that $\Phi_{H_{t+1}}(S_i)=O(k\cdot\Phi_{G_{t+1}}(S_i))$ for any $1\leqslant i\leqslant k$, where $S_1,\ldots,S_k$ are the optimal clusters corresponding to $\rho_{G_{t+1}}(k)$. For any $1\leqslant i\leqslant k$, it holds that

$$\mathbb{E}[\,w_{H_{t+1}}(S_i,V_{t+1}\setminus S_i)\,]=\sum_{\substack{e=\{u,v\}\in E_{t+1}\\u\in S_i,v\notin S_i}}\widetilde{q}(u,v)\cdot\frac{1}{\widetilde{q}(u,v)}=w_{G_{t+1}}(S,V_{t+1}\setminus S_i).$$

Hence, by Markov's inequality and the union bound, it holds with constant probability that $w_{H_{t+1}}(S_i,V_{t+1}\setminus S_i)=O(k\cdot w_{G_{t+1}}(S_i,V_{t+1}\setminus S_i))$. Therefore, it holds with constant probability that

$$\rho_{H_{t+1}}(k)\leqslant\max_{1\leqslant i\leqslant k}\Phi_{H_{t+1}}(S_i)=\max_{1\leqslant i\leqslant k}O(k\cdot\Phi_{G_{t+1}}(S_i))=O(k\cdot\rho_{G_{t+1}}(k)).$$

Next, we prove that $\lambda_{k+1}(\mathcal{L}_{H_{t+1}})=\Omega(\lambda_{k+1}(\mathcal{L}_{G_{t+1}}))$. Let $\overline{\mathcal{L}}_{G_{t+1}}$ be the projection of $\mathcal{L}_{G_{t+1}}$ on its top $n_{t+1}-k$ eigenspaces, and notice that $\overline{\mathcal{L}}_{G_{t+1}}$ can be written as

$$\overline{\mathcal{L}}_{G_{t+1}}=\sum_{i=k+1}^{n_{t+1}}\lambda_i(\mathcal{L}_{G_{t+1}})\cdot f_if_i^{\intercal}$$

where $f_1,\ldots,f_{n_{t+1}}$ are the eigenvectors of $\mathcal{L}_{G_{t+1}}$. Let $\overline{\mathcal{L}}_{G_{t+1}}^{-1/2}$ be the square root of the pseudoinverse of $\overline{\mathcal{L}}_{G_{t+1}}$. We prove that the top $n_{t+1}-k$ eigenvalues of $\mathcal{L}_{G_{t+1}}$ are preserved, which implies that $\lambda_{k+1}(\mathcal{L}_{H_{t+1}})=\Theta(\lambda_{k+1}(\mathcal{L}_{G_{t+1}}))$.

To prove this, for each edge $e=\{u,v\}\in E_{G_{t+1}}$ we define a random matrix $X_e\in\mathbb{R}^{n_{t+1}\times n_{t+1}}$ by

$$X_e=\begin{cases}w_{H_{t+1}}(u,v)\cdot\overline{\mathcal{L}}_{G_{t+1}}^{-1/2}b_eb_e^{\intercal}\overline{\mathcal{L}}_{G_{t+1}}^{-1/2} & \text{if } e=\{u,v\}\text{ is sampled by the algorithm}\\0 & \text{otherwise,}\end{cases}$$

where $b_e\triangleq\chi_u-\chi_v$ is the edge indicator vector and $\chi_v\in\mathbb{R}^n$ is defined by

$$\chi_v(a)\triangleq\begin{cases}\frac{1}{\sqrt{\deg_{G_{t+1}}(v)}} & \text{if } a=v\\0 & \text{otherwise.}\end{cases}$$

Notice that

$$\sum_{e\in E_{G_{t+1}}}X_e=\sum_{\substack{e=\{u,v\}\\e\in E_{G_{t+1}}}}w_{H_{t+1}}(u,v)\cdot\overline{\mathcal{L}}_{G_{t+1}}^{-1/2}b_eb_e^{\intercal}\overline{\mathcal{L}}_{G_{t+1}}^{-1/2}=\overline{\mathcal{L}}_{G_{t+1}}^{-1/2}\mathcal{L}_{H'_{t+1}}\overline{\mathcal{L}}_{G_{t+1}}^{-1/2},$$

where

$$\mathcal{L}_{H'_{t+1}} \triangleq \sum_{e \in E_{G_{t+1}}} w_{H_{t+1}}(u,v) \cdot b_e b_e^{\intercal}$$

is $\mathcal{L}_{H_{t+1}}$ normalised with respect to the degree of the vertices in G_{t+1}. We prove that, with high probability, the top $n_{t+1}-k$ eigenvalues of $\mathcal{L}_{H'_{t+1}}$ and $\mathcal{L}_{G_{t+1}}$ are approximately the same. Then, to finish the proof, we also show that this is the case for the top $n_{t+1}-k$ eigenvalues of $\mathcal{L}_{H_{t+1}}$ and $\mathcal{L}_{H'_{t+1}}$, from which we get that $\lambda_{k+1}(\mathcal{L}_{H_{t+1}}) = \Omega\left(\lambda_{k+1}(\mathcal{L}_{G_{t+1}})\right)$.

First, from (8) we get that for any edge e it holds that

$$\widetilde{q}(u,v) \leqslant \widetilde{p}_u(v) + \widetilde{p}_v(u) \leqslant 2 \cdot \left(\frac{\tau \cdot \log(n_{t+1})}{\deg_{G_{t+1}}(u)} + \frac{\tau \cdot \log(n_{t+1})}{\deg_{G_{t+1}}(v)} \right), \tag{14}$$

and

$$\widetilde{q}(u,v) \geqslant \frac{1}{2} \cdot (\widetilde{p}_u(v) + \widetilde{p}_v(u)) \geqslant \frac{1}{4} \cdot \left(\frac{\tau \cdot \log(n_{t+1})}{\deg_{G_{t+1}}(u)} + \frac{\tau \cdot \log(n_{t+1})}{\deg_{G_{t+1}}(u)} \right). \tag{15}$$

We start by calculating the first moment of $\sum_{e \in E_{G_{t+1}}} X_e$, and have that

$$\begin{aligned}
\mathbb{E}\left[\sum_{e \in E_{G_{t+1}}} X_e \right] &= \sum_{\substack{e=\{u,v\} \\ e \in E_{G_{t+1}}}} \widetilde{q}(u,v) \cdot w_{H_{t+1}}(u,v) \cdot \overline{\mathcal{L}}_{G_{t+1}}^{-1/2} b_e b_e^{\intercal} \overline{\mathcal{L}}_{G_{t+1}}^{-1/2} \overline{\mathcal{L}}_{G_{t+1}}^{-1/2} \\
&= \sum_{\substack{e=\{u,v\} \\ e \in E_{G_{t+1}}}} \widetilde{q}(u,v) \cdot \frac{1}{\widetilde{q}(u,v)} \cdot \overline{\mathcal{L}}_{G_{t+1}}^{-1/2} b_e b_e^{\intercal} \overline{\mathcal{L}}_{G_{t+1}}^{-1/2} \\
&= \overline{\mathcal{L}}_{G_{t+1}}^{-1/2} \mathcal{L}_{G_{t+1}} \overline{\mathcal{L}}_{G_{t+1}}^{-1/2}.
\end{aligned}$$

Moreover, for any sampled $e = \{u, v\}$ we have that

$$\begin{aligned}
\|X_e\| &\leqslant w_{H_{t+1}}(u,v) \cdot b_e^{\intercal} \overline{\mathcal{L}}_{G_{t+1}}^{-1/2} \overline{\mathcal{L}}_{G_{t+1}}^{-1/2} b_e \\
&= \frac{1}{\widetilde{q}(u,v)} \cdot b_e^{\intercal} \overline{\mathcal{L}}_{G_{t+1}}^{-1/2} \overline{\mathcal{L}}_{G_{t+1}}^{-1/2} b_e \\
&\leqslant \frac{1}{\widetilde{q}(u,v)} \cdot \frac{1}{\lambda_{k+1}(\mathcal{L}_{G_{t+1}})} \cdot \|b_e\|^2 \\
&\leqslant \frac{4\lambda_{k+1}(\mathcal{L}_{G_{t+1}})}{C \cdot \log n_{t+1} \cdot \left(\frac{1}{\deg_{G_{t+1}}(u)} + \frac{1}{\deg_{G_{t+1}}(v)} \right)} \cdot \frac{1}{\lambda_{k+1}(\mathcal{L}_{G_{t+1}})} \cdot \left(\frac{1}{\deg_{G_{t+1}}(u)} + \frac{1}{\deg_{G_{t+1}}(v)} \right) \qquad (16) \\
&= \frac{4}{C \cdot \log n_{t+1}},
\end{aligned}$$

where the second inequality follows by the min-max theorem of eigenvalues, and (16) holds by (15). Now we apply the matrix Chernoff bound (Lemma A.3) to analyse the eigenvalues of $\sum_{e \in E_{G_{t+1}}} X_e$. We set $\lambda_{\max}\left(\mathbb{E}\left[\sum_{e \in E_{G_{t+1}}} X_e\right]\right) = \lambda_{\max}\left(\overline{\mathcal{L}}_{G_{t+1}}^{-1/2} \mathcal{L}_{G_{t+1}} \overline{\mathcal{L}}_{G_{t+1}}^{-1/2}\right) = 1$, $R = \frac{4}{C \cdot \log n_{t+1}}$ and $\delta = 1/2$, and have that

$$\mathbb{P}\left[\lambda_{\max}\left(\sum_{e \in E_{G_{t+1}}} X_e \right) \geqslant \frac{3}{2} \right] \leqslant n_{t+1} \cdot \left(\frac{e^{1/2}}{(1+1/2)^{3/2}} \right)^{C \cdot \log n_{t+1}/4} = O(1/n_{t+1}^c)$$

for some constant c. Therefore we get that

$$\mathbb{P}\left[\lambda_{\max}\left(\sum_{e \in E_{G_{t+1}}} X_e \right) < \frac{3}{2} \right] = 1 - O(1/n_{t+1}^c). \tag{17}$$

Similarly, since $\lambda_{\min}\left(\mathbb{E}\left[\sum_{e\in E_{G_{t+1}}} X_e\right]\right) = \lambda_{\min}\left(\overline{\mathcal{L}}_{G_{t+1}}^{-1/2}\mathcal{L}_{G_{t+1}}\overline{\mathcal{L}}_{G_{t+1}}^{-1/2}\right) = 1$, the other side of the matrix Chernoff bound gives us that

$$\mathbb{P}\left[\lambda_{\min}\left(\sum_{e\in E_{G_{t+1}}} X_e\right) > \frac{1}{2}\right] = 1 - O(1/n_{t+1}^c). \tag{18}$$

Combining (17) and (18), it holds with probability $1 - O(1/n_{t+1}^c)$ for any non-zero $x \in \mathbb{R}^{n_{t+1}}$ in the space spanned by $f_{k+1}, \ldots, f_{n_{t+1}}$ that

$$\frac{x^\intercal \overline{\mathcal{L}}_{G_{t+1}}^{-1/2}\mathcal{L}'_{H_{t+1}}\overline{\mathcal{L}}_{G_{t+1}}^{-1/2}x}{x^\intercal x} \in (1/2, 3/2).$$

Since $\dim(\operatorname{span}\{f_{k+1}, \ldots, f_{n_{t+1}}\}) = n_{t+1} - k$, there exist $n_{t+1} - k$ orthogonal vectors whose Rayleigh quotient with respect to $\mathcal{L}'_{H_{t+1}}$ is $\Omega\left(\lambda_{k+1}\left(\mathcal{L}_{G_{t+1}}\right)\right)$. The Courant-Fischer Theorem implies that $\lambda_{k+1}(\mathcal{L}'_{H_{t+1}}) = \Omega\left(\lambda_{k+1}\left(\mathcal{L}_{G_{t+1}}\right)\right)$.

It only remains to show that $\lambda_{k+1}(\mathcal{L}_{H_{t+1}}) = \Omega(\lambda_{k+1}(\mathcal{L}'_{H_{t+1}}))$, which implies that $\lambda_{k+1}(\mathcal{L}_{H_{t+1}}) = \Omega\left(\lambda_{k+1}\left(\mathcal{L}_{G_{t+1}}\right)\right)$. By definition of $\lambda_{k+1}(\mathcal{L}'_{H_{t+1}})$, we have that

$$\mathcal{L}_{H_{t+1}} = D_{H_{t+1}}^{-1/2} D_{G_{t+1}}^{1/2} \mathcal{L}'_{H_{t+1}} D_{G_{t+1}}^{1/2} D_{H_{t+1}}^{-1/2}.$$

Therefore, for any $x \in \mathbb{R}^{n_{t+1}}$ and $y \triangleq D_{G_{t+1}}^{1/2} D_{H_{t+1}}^{-1/2} x$, it holds that

$$\frac{x^\intercal \mathcal{L}_{H_{t+1}} x}{x^\intercal x} = \frac{y^\intercal \mathcal{L}'_{H_{t+1}} y}{x^\intercal x} = \Omega\left(\frac{y^\intercal \mathcal{L}'_{H_{t+1}} y}{y^\intercal y}\right),$$

where the final guarantee follows from the fact that the degrees in H_{t+1} are preserved up to a constant factor. The conclusion of the theorem follows from the Courant-Fischer Theorem.

Finally, it remains to analyse the amortised update time of the algorithm. Notice that, if one only needs to sample the incoming edge at time $t+1$, then the update time is $O(1)$. Otherwise, all the edges adjacent to some vertex w need to be resampled, and the running time for this step is $O(\deg_{G_{t+1}}(w))$. However, this means that either $\deg_{G_{t+1}}(w) > 2\cdot\deg_{G_t}(w)$ or $\log(n_{t+1}) > 2\cdot\log(n_t)$. In the first case, this only occurs at most every $\deg_{G_t}(w)$ edge updates, which results in the amortised update time of $O(1)$. The second case only happens after every n_t^2 vertex additions, and in the worst case we only have to resample all the edges in present in G_t every n_t^2 edge updates, which again leads to the amortised update time of $O(1)$. □

B. Omitted Details from Section 4

This section contains the omitted details from Section 4, and is organised as follows. In Section B.1 we introduce additional notation to analyse our constructed contracted graphs. In Section B.2 we present the omitted proofs for Lemmas 4.1, 4.2, 4.3, 4.4, and 4.5, and we formally describe the UpdateContractedGraph procedure.

B.1. Notation

For any subset $A \subset V_{t'}$, let $\widetilde{A} \triangleq A \cap \widetilde{V}_{t'}^{\mathrm{nc}}$ be the representation of A among the non-contracted vertices of $\widetilde{G}_{t'}$. Recall that for any subset of vertices $A \subset V_{t'}$, we use $A^{(t)} \triangleq A \cap V_t$ to denote the set of vertices present at time t. Let

$$E_{\text{added}} \triangleq E_{\text{new}} \cup \left\{\{u,v\} \in E_t \;\middle|\; \deg_{G_{t'}}(u) > 2\cdot\deg_{G_r}(u) \text{ or } \deg_{G_{t'}}(v) > 2\cdot\deg_{G_r}(v)\right\}$$

be the set of edges that have been directly added into $\widetilde{G}_t$, where $\deg_{G_r}(w)$ for $r \leqslant t$ is the degree of w used to construct the contracted graph. These edges are the ones directly added as new edges or their endpoints are pulled out from clusters in $\widetilde{G}_t$. For a subset $B \subset \widetilde{V}_{t'}$, let $\hat{B}$ be the representation of the set B in $G_{t'}$, i.e.,

$$\hat{B} \triangleq B^{\mathrm{nc}} \bigcup \left(\bigcup_{p_i \in B^{\mathrm{c}}} P_i^{(t')}\right),$$

where $P_i^{(t')} \triangleq P_i \setminus (P_i \cap \widetilde{V}_{t'}^{\mathrm{nc}})$, $B^{\mathrm{nc}} \triangleq B \cap V_{t'}^{\mathrm{nc}}$, and $B^{\mathrm{c}} \triangleq B \cap V_{t'}^{\mathrm{c}}$. One can see $P_i^{(t')}$ as the vertices in P_i that are still represented by the respective super vertex in $\widetilde{G}$.

B.2. Omitted Proofs

Our analysis is based on approximation guarantee of spectral clustering. The following result, which can be shown easily by combining the proof technique of (Peng et al., 2017) and the one of (Macgregor & Sun, 2022), will be used in our analysis.

Lemma B.1. *There is an absolute constant $C_{B.1} \in \mathbb{R}_{>0}$, such that the following holds: Let G be a graph with k optimal clusters $\{S_i\}_{i=1}^k$, and $\Upsilon_G(k) \geqslant C_{B.1} \cdot k$. Let $\{P_i\}_{i=1}^k$ be the output of spectral clustering and, without loss of generality, the optimal correspondence of P_i is S_i for any $1 \leqslant i \leqslant k$. Then, it holds for any $1 \leqslant i \leqslant k$ that*

$$\mathrm{vol}_G(P_i \triangle S_i) \leqslant \frac{k \cdot C_{B.1}}{3\Upsilon_G(k)} \cdot \mathrm{vol}_G(S_i),$$

where $A \triangle B$ for any sets A and B is defined by $A \triangle B \triangleq (A \setminus B) \cup (B \setminus A)$. It also holds that

$$\Phi_G(P_i) = O\left(\Phi_G(S_i) + \frac{k}{\Upsilon_G(k)}\right).$$

Moreover, these $P_1, \ldots, P_k$ can be computed in nearly-linear time.

Proof of Lemma 4.1. The running time of the algorithm is dominated by computing the total weight $w_{H_t}(P_i, P_j)$ between every $P_i, P_j \in \mathcal{P}$ (Lines 7–10), which takes $O(|F_t|)$ time as there are $|F_t|$ edges in H_t. □

Proof of Lemma 4.2. The running time of the update operation is dominated by the case in which a vertex is pulled out from a contracted vertex (Lines 7–22). It's easy to see that, if this does not happen, then the running time is $O(1)$ as the edge is just added into $\widetilde{G}_t$.

Let $\{u, v\}$ be the added edge, and we assume Lines 7–22 are triggered. The running time for this case is $O(\deg_{G_{t+1}}(u) + \deg_{G_{t+1}}(v))$, since at least one of u and v is pulled out from their respective contracted vertices and all the adjacent edges are placed into the contracted graph. Notice that this only happens if $\deg_{G_{t+1}}(u) > 2 \cdot \deg_{G_t}(u)$ or $\deg_{G_{t+1}}(v) > 2 \cdot \deg_{G_t}(v)$. Since at least $\deg_{G_t}(u)$ or $\deg_{G_t}(v)$ edge insertions are needed before running Lines 7–22, the amortised per edge update time is $O(1)$. □

Proof of Lemma 4.3. Notice by Lemma B.1 we know it holds with high probability for all $1 \leqslant i \leqslant k$ that $\Phi_{H_t}(P_i) = O\left(k \cdot \rho_{H_t}(k)\right)$. By applying Theorem 3.3, it holds with high probability that $\Phi_{H_t}(P_i) = O\left(k^2 \cdot \rho_{G_t}(k)\right)$. By Lemma 3.2, we also have with high probability that $\Phi_{G_t}(P_i) = O\left(k^2 \cdot \rho_{G_t}(k)\right)$. This proves the statement. □

The next lemma shows that, starting from G_t and H_t, one can easily construct a cluster preserving sparsifier of $G_{t'}$.

Lemma B.2. *Let $H'_{t'} \triangleq (V_{t'}, F_t \cup E_{\mathrm{added}}, w_{H'_{t'}})$ be a graph, where*

$$w_{H'_{t'}}(e) \triangleq \begin{cases} 1 & e \in E_{\mathrm{added}} \\ w_{H_t}(e) & e \in F_t \setminus E_{\mathrm{added}} \\ 0 & \textit{otherwise.} \end{cases}$$

Then, it holds with high probability that $H'_{t'}$ is a cluster preserving sparsifier of $G_{t'}$.

Proof. First, for any $e \in E_{\mathrm{added}}$ we know that it is included in $H'_{t'}$ with probability 1. For any other edge $e = \{u, v\} \in F_t \setminus E_{\mathrm{added}}$, we know by the construction of H_t using the dynamic cluster-preserving sparsifier that the parameter used to sample e from the perspective of u is

$$\frac{\tau \cdot \log(n_t)}{2 \cdot \deg_{G_t}(u)} \leqslant \frac{\tau \cdot \log(n_r)}{\deg_{G_r}(u)} \leqslant \frac{2 \cdot \tau \cdot \log(n_t)}{\deg_{G_t}(u)},$$

for some $1 \leqslant r \leqslant t$. We also know by construction that for any $e = \{u, v\} \in F_t \setminus E_{\mathrm{added}}$ that

$$\deg_{G_{t'}}(u) \leqslant 2 \cdot \deg_{G_t}(u).$$

Algorithm 7 UpdateContractedGraph$(G_t, \widetilde{G}_t, e)$

Input: Graph $G_t = (V_t, E_t)$, contracted graph $\widetilde{G}_t = (\widetilde{V}_t, \widetilde{E}_t, w_{\widetilde{G}_t})$, incoming edge $e = \{u, v\}$.
Output: Contracted graph $\widetilde{G}_{t+1} = (\widetilde{V}_{t+1}, \widetilde{E}_{t+1}, w_{\widetilde{G}_{t+1}})$
$V_{\mathrm{new}} \leftarrow \{u, v\} \setminus V_t$
$G_{t+1} \leftarrow (V_t \cup V_{\mathrm{new}}, E_t \cup e)$
$\widetilde{G}_{t+1} \leftarrow (\widetilde{V}_t \cup V_{\mathrm{new}}, \widetilde{E}_t, w_{\widetilde{G}_t}) = (\widetilde{V}_{t+1}, \widetilde{E}_{t+1}, w_{\widetilde{G}_{t+1}})$
$\widetilde{V}^{\mathrm{nc}}_{t+1} \leftarrow \widetilde{V}^{\mathrm{nc}}_{t+1} \cup V_{\mathrm{new}}$
for $w \in \{u, v\} \setminus V_{\mathrm{new}}$ **do**
    Let G_r be the graph at time r when the contracted graph is constructed, and $H_r = (V_r, F_r, w_{H_r})$ the cluster preserving sparsifier at time r.
    if $w \notin \widetilde{V}^{\mathrm{nc}}_{t+1}$ and $\deg_{G_{t+1}}(w) > 2 \cdot \deg_{G_r}(w)$ **then**
        Let p_j be the super node such that $w \in P_j$
        $\widetilde{V}^{\mathrm{nc}}_{t+1} \leftarrow \widetilde{V}^{\mathrm{nc}}_{t+1} \cup w$
        $\widetilde{E}_{t+1} \leftarrow \widetilde{E}_{t+1} \cup E_{G_{t+1}}(w, \widetilde{V}^{\mathrm{nc}}_{t+1})$
        for $\hat{v} \in \widetilde{V}^{\mathrm{nc}}_{t+1}$ adjacent to w **do**
            $w_{\widetilde{G}_{t+1}}(p_j, \hat{v}) \leftarrow w_{\widetilde{G}_{t+1}}(p_j, \hat{v}) - 1$
        end for
        for $\{w, p_i\} \in w \times \widetilde{V}^{\mathrm{c}}_{t+1}$ **do**
            $\widetilde{E}_{t+1} \leftarrow \widetilde{E}_{t+1} \cup \{w, p_i\}$
            $w_{\widetilde{G}_{t+1}}(w, p_i) \leftarrow w_{G_{t+1}}(w, P_i^{(t+1)})$
            $w_{\widetilde{G}_{t+1}}(p_i, p_j) \leftarrow w_{\widetilde{G}_{t+1}}(p_i, p_j) - w_{H_r}(w, P_i^{(t+1)})$
        end for
    end if
end for
if $u \in \widetilde{V}^{\mathrm{nc}}_{t+1}$ and $v \in \widetilde{V}^{\mathrm{nc}}_{t+1}$ **then**
    $\widetilde{E}_{t+1} \leftarrow \widetilde{E}_{t+1} \cup \{u, v\}$
else if $u \in \widetilde{V}^{\mathrm{nc}}_{t+1}$ or $v \in \widetilde{V}^{\mathrm{nc}}_{t+1}$ **then**
    Without loss of generality, let $u \in \widetilde{V}^{\mathrm{nc}}_{t+1}$ and $v \notin \widetilde{V}^{\mathrm{nc}}_{t+1}$. Let p_j be the supernode such that $v \in P_j$
    $\widetilde{E}_{t+1} \leftarrow \widetilde{E}_{t+1} \cup \{u, p_j\}$
    $w_{\widetilde{G}_{t+1}}(u, p_j) \leftarrow w_{\widetilde{G}_{t+1}}(u, p_j) + 1$
else
    Let p_i and p_j be the supernodes such that $u \in P_i$ and $v \in P_j$
    $w_{\widetilde{G}_{t+1}}(p_i, p_j) \leftarrow w_{\widetilde{G}_{t+1}}(p_i, p_j) + 1$
end if
Return $G_{t+1} = (\widetilde{V}_{t+1}, \widetilde{E}_{t+1}, w_{\widetilde{G}_{t+1}})$

Finally, it holds that $\log(n_t) \leqslant \log(n_{t'}) \leqslant 2\log(n_t)$. From this we get that e is sampled from vertex u with the following parameter

$$\frac{\tau \cdot \log(n_{t'})}{4 \cdot \deg_{G_{t'}}(u)} \leqslant \frac{\tau \cdot \log(n_r)}{\deg_{G_r}(u)} \leqslant \frac{4 \cdot \tau \cdot \log(n_{t'})}{\deg_{G_{t'}}(u)}.$$

Following almost the same analysis as the proof of Theorem 3.3, it holds with high probability that $H'_{t'}$ is a cluster preserving sparsifier of $G_{t'}$. □

Our next lemma proves several useful properties about the contracted graph as it is updated.

Lemma B.3. *The following statements hold:*

(C1) It holds for any subset $B \subset V_{t'} \setminus \widetilde{V}^{\mathrm{nc}}_{t'}$ that $\mathrm{vol}_{G_{t'}}(B) \leqslant 2 \cdot \mathrm{vol}_{G_t}(B)$.

(C2) Suppose for a subset $A \subset V_{t'}$ with $\mathrm{vol}_{G_{t'}}(A) \leqslant \mathrm{vol}(G_{t'})/2$ we have that $\Phi_{G_t}(A^{(t)}) \geqslant 1/c_1$ and $\Phi_{G_{t'}}(A) \leqslant \log^{-\varepsilon}(n_{t'})$ for any positive c_1, ε such that $4 \cdot c_1 \leqslant \log^{\varepsilon}(n_{t'})$, then it holds that

$$\Phi_{\widetilde{G}_{t'}}(\widetilde{A}) \leqslant \frac{21 \cdot c_1}{\log^{\varepsilon}(n_{t'})}.$$

(C3) For any super node $p_i \in \widetilde{V}^{\mathrm{c}}_t$, it holds that

$$\Phi_{\widetilde{G}_t}(p_i) = O\left(k^{-6} \cdot \log^{-2\gamma}(n_t)\right),$$

and

$$\Phi_{\widetilde{G}_{t'}}(p_i) = O\left(k^{-6} \cdot \log^{-\gamma}(n_t)\right).$$

Informally speaking, Property (C1) of Lemma B.3 shows that the volume of any vertex set $B \subset V_{t'}$ that are not directly represented in $\widetilde{G}_{t'}$ remains approximately the same in G_t and $G_{t'}$; Property (C2) states that, if the conductance of any set $A \subset V_{t'}$ in $G_{t'}$ becomes much lower than the one in G_t, then its representative set $\widetilde{A} \subset \widetilde{V}_{t'}$ has low conductance; Property (C3) further shows that the conductance of all the contracted vertices doesn't change significantly over time.

Proof of Lemma B.3. For (C1), by construction we have that for any $u \in V_{t'} \setminus \widetilde{V}^{\mathrm{nc}}_{t'}$ it holds that $\deg_{G_{t'}}(u) \leqslant 2 \cdot \deg_{G_t}(u)$, from which the statement follows.

Next, we prove (C2). The following two claims will be used in our analysis.

Claim B.3.1. It holds that $\mathrm{vol}_{E_{\mathrm{new}}}(A) \geqslant \frac{\mathrm{vol}_{G_t}(A^{(t)}) \cdot \log^{\varepsilon}(n_{t'})}{2 \cdot c_1}$.

Proof. Assume by contradiction that $\mathrm{vol}_{E_{\mathrm{new}}}(A) < \frac{\mathrm{vol}_{G_t}(A^{(t)}) \cdot \log^{\varepsilon}(n_{t'})}{2 \cdot c_1}$. We have that

$$\begin{aligned}
\Phi_{G_{t'}}(A) &= \frac{w_{G_t}(A^{(t)}, V_t \setminus A^{(t)}) + w_{E_{\mathrm{new}}}(A, V_t \setminus A)}{\mathrm{vol}_{G_t}(A^{(t)}) + \mathrm{vol}_{E_{\mathrm{new}}}(A)} \\
&\geqslant \frac{w_{G_t}(A^{(t)}, V_t \setminus A^{(t)})}{\mathrm{vol}_{G_t}(A^{(t)}) + \mathrm{vol}_{E_{\mathrm{new}}}(A)} \\
&\geqslant \frac{1}{2} \min\left\{ \frac{w_{G_t}(A^{(t)}, V_t \setminus A^{(t)})}{\mathrm{vol}_{G_t}(A^{(t)})}, \frac{w_{G_t}(A^{(t)}, V_t \setminus A^{(t)})}{\mathrm{vol}_{E_{\mathrm{new}}}(A)} \right\} \\
&\geqslant \frac{1}{2} \min\left\{ \Phi_{G_t}(A^{(t)}), \frac{\mathrm{vol}_{G_t}(A^{(t)})}{c_1 \cdot \mathrm{vol}_{E_{\mathrm{new}}}(A)} \right\} \\
&> \frac{1}{\log^{\varepsilon}(n_{t'})},
\end{aligned}$$

where on the last line we used the contradictory assumption. This contradicts the condition of $\Phi_{G_{t'}}(A) \leqslant \log^{-\varepsilon}(n_{t'})$, and hence the statement holds. □

Notice that this claim implies that

$$\begin{aligned}\mathrm{vol}_{G_{t'}}(A) &= \mathrm{vol}_{G_t}(A^{(t)}) + \mathrm{vol}_{E_{\mathrm{new}}}(A) \\ &\geqslant \left(1 + \frac{\log^{\varepsilon}(n_{t'})}{2 \cdot c_1}\right) \cdot \mathrm{vol}_{G_t}(A^{(t)}) \\ &\geqslant \left(1 + \frac{\log^{\varepsilon}(n_{t'})}{4 \cdot c_1} + \frac{\log^{\varepsilon}(n_{t'})}{4 \cdot c_1}\right) \cdot \mathrm{vol}_{G_t}(A^{(t)}) \\ &\geqslant \left(2 + \frac{\log^{\varepsilon}(n_{t'})}{4 \cdot c_1}\right) \cdot \mathrm{vol}_{G_t}(A^{(t)}) \end{aligned} \tag{19}$$

where the last inequality follows from the fact that $4 \cdot c_1 \leqslant \log^{\varepsilon}(n_{t'})$.

Claim B.3.2. It holds that $\mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A}) \geqslant \frac{\log^{\varepsilon}(n_{t'})}{4 \cdot c_1} \cdot \mathrm{vol}_{G_t}(A^{(t)})$.

Proof. Assume by contradiction that $\mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A}) < \frac{\log^{\varepsilon}(n_{t'})}{4 \cdot c_1} \cdot \mathrm{vol}_{G_t}(A^{(t)})$. Then, it holds that

$$\begin{aligned}\mathrm{vol}_{G_{t'}}(A) &= \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A}) + \mathrm{vol}_{G_{t'}}(A \setminus \widetilde{A}) \\ &\leqslant \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A}) + 2 \cdot \mathrm{vol}_{G_t}(A \setminus \widetilde{A}) \\ &< \frac{\log^{\varepsilon}(n_{t'})}{4 \cdot c_1} \cdot \mathrm{vol}_{G_t}(A^{(t)}) + 2 \cdot \mathrm{vol}_{G_t}(A^{(t)}),\end{aligned}$$

where the first inequality holds by statement (C1). Hence, we reach a contradiction with (19), and the claim holds. □

Now we are ready to prove statement (C2). We have that

$$\begin{aligned}\Phi_{\widetilde{G}_{t'}}(\widetilde{A}) &= \frac{w_{\widetilde{G}_{t'}}(\widetilde{A}, \widetilde{V}_{t'} \setminus \widetilde{A})}{\mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A})} \\ &\leqslant \frac{w_{G_{t'}}(A, V_t \setminus A) + w_{G_{t'}}(A \setminus \widetilde{A}, \widetilde{A})}{\mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A})} \\ &\leqslant \frac{\log^{-\varepsilon}(n_{t'}) \cdot \mathrm{vol}_{G_{t'}}(A) + \mathrm{vol}_{G_{t'}}(A \setminus \widetilde{A})}{\mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A})} \\ &\leqslant \frac{\mathrm{vol}_{G_{t'}}(A)}{\log^{\varepsilon}(n_{t'}) \cdot \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A})} + \frac{8 \cdot c_1 \cdot \mathrm{vol}_{G_t}(A^{(t)})}{\mathrm{vol}_{G_t}(A^{(t)}) \cdot \log^{\varepsilon}(n_{t'})} && (20) \\ &\leqslant \frac{\mathrm{vol}_{G_t}(A^{(t)}) + \mathrm{vol}_{E_{\mathrm{new}}}(A)}{\log^{\varepsilon}(n_{t'}) \cdot \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A})} + \frac{8 \cdot c_1}{\log^{\varepsilon}(n_{t'})} \\ &\leqslant \frac{3 \cdot \mathrm{vol}_{G_t}(A^{(t)}) + \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A})}{\log^{\varepsilon}(n_{t'}) \cdot \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A})} + \frac{8 \cdot c_1}{\log^{\varepsilon}(n_{t'})} && (21) \\ &\leqslant \frac{3 \cdot \mathrm{vol}_{G_t}(A^{(t)}) \cdot c_1 \cdot 4}{\log^{2\varepsilon}(n_{t'}) \cdot \mathrm{vol}_{G_t}(A^{(t)})} + \frac{\mathrm{vol}_{\widetilde{G}_{t'}}(A)}{\log^{\varepsilon}(n_{t'}) \cdot \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A})} + \frac{8 \cdot c_1}{\log^{\varepsilon}(n_{t'})} && (22) \\ &\leqslant \frac{12 \cdot c_1}{\log^{2\varepsilon}(n_{t'})} + \frac{1}{\log^{\varepsilon}(n_{t'})} + \frac{8 \cdot c_1}{\log^{\varepsilon}(n_{t'})} \\ &\leqslant \frac{1 + 20 \cdot c_1}{\log^{\varepsilon}(n_{t'})} \leqslant \frac{21 \cdot c_1}{\log^{\varepsilon}(n_{t'})},\end{aligned}$$

where (20) holds by Claim B.3.2 and the fact that $\mathrm{vol}_{G_{t'}}(A \setminus \widetilde{A}) \leqslant 2 \cdot \mathrm{vol}_{G_t}(A \setminus \widetilde{A}) \leqslant 2 \cdot \mathrm{vol}_{G_t}(A^{(t)})$ by statement (C1). (21) holds because by construction $\mathrm{vol}_{E_{\mathrm{new}}}(A) \leqslant \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A}) + \mathrm{vol}_{G_{t'}}(A \setminus \widetilde{A}) \leqslant \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{A}) + 2 \cdot \mathrm{vol}_{G_t}(A^{(t)})$, and (22) holds because of Claim B.3.2.

Finally, we prove statement (C3). For any $P_i \in \mathcal{P}$, we have by construction that

$$\Phi_{\widetilde{G}_t}(p_i) = \Phi_{H_t}(P_i) = O\left(k^{-6} \cdot \log^{-2\gamma}(n_t)\right), \tag{23}$$

where the last equality holds by Lemma 4.3. This proves the first part of the statement. Next, notice that for any $p_i \in \widetilde{V}_t^{\mathrm{c}}$, because G_t is connected and each P_i has almost identical volume as the corresponding optimal S_i in G_t (Lemma 3.2), by construction it holds that

$$\mathrm{vol}_{\widetilde{G}_t}(p_i) = \Omega(k^6 \cdot \log^{2\gamma}(n_t)), \tag{24}$$

and

$$\mathrm{vol}_{\widetilde{G}_t}(\widetilde{V}_t \setminus p_i) = \Omega(k^6 \cdot \log^{2\gamma}(n_t)). \tag{25}$$

Taking this into account, we get that

$$\begin{aligned} w_{\widetilde{G}_{t'}}(p_i, \widetilde{V}_{t'} \setminus p_i) &\leqslant w_{\widetilde{G}_t}(p_i, \widetilde{V}_t \setminus p_i) + |E_{\mathrm{new}}| + w_{G_{t'}}\left(P_i \cap \widetilde{V}_{t'}^{\mathrm{nc}}, P_i \setminus (P_i \cap \widetilde{V}_{t'}^{\mathrm{nc}})\right) \\ &\leqslant w_{\widetilde{G}_t}(p_i, \widetilde{V} \setminus p_i) + \log^\gamma(n_t) + \mathrm{vol}_{G_{t'}}\left(P_i \cap \widetilde{V}_{t'}^{\mathrm{nc}}\right) \\ &\leqslant \Phi_{\widetilde{G}_t}(p_i) \cdot \min\{\mathrm{vol}_{\widetilde{G}_t}(p_i), \mathrm{vol}_{\widetilde{G}_t}(\widetilde{V} \setminus p_i)\} + \log^\gamma(n_t) + 2 \cdot \log^\gamma(n_t) \qquad (26) \\ &= O\left(k^{-6} \cdot \log^{-2\gamma}(n_t)\right) \cdot \min\{\mathrm{vol}_{\widetilde{G}_t}(p_i), \mathrm{vol}_{\widetilde{G}_t}(\widetilde{V} \setminus p_i)\} + 3 \cdot \log^\gamma(n_t), \qquad (27) \end{aligned}$$

where (26) holds because $\mathrm{vol}_{G_{t'}}(P_i \cap \widetilde{V}_{t'}^{\mathrm{nc}}) \leqslant 2 \cdot \log^\gamma(n_t)$ as every vertex that is pulled out of p_i needs to at least double in degree, so adding $|E_{\mathrm{new}}|$ edges ensures at most $2 \cdot |E_{\mathrm{new}}|$ volume can be pulled out of p_i, (27) holds because of (23). Moreover, we also have that

$$\begin{aligned} \min\{\mathrm{vol}_{\widetilde{G}_{t'}}(p_i), \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{V}_{t'} \setminus p_i)\} &\geqslant \min\{\mathrm{vol}_{\widetilde{G}_t}(p_i) - 2 \cdot \log^\gamma(n_t), \mathrm{vol}_{\widetilde{G}_t}(\widetilde{V}_t \setminus p_i)\} \qquad (28) \\ &= \Omega\left(\min\{\mathrm{vol}_{\widetilde{G}_t}(p_i), \mathrm{vol}_{\widetilde{G}_t}(\widetilde{V}_t \setminus p_i)\}\right), \qquad (29) \end{aligned}$$

where (28) holds because $\mathrm{vol}_{\widetilde{G}_{t'}}(p_i) \geqslant \mathrm{vol}_{\widetilde{G}_t}(p_i) - \mathrm{vol}_{G_{t'}}(P_i \cap \widetilde{V}_{t'}^{\mathrm{nc}}) \geqslant \mathrm{vol}_{\widetilde{G}_{t'}}(p_i) - 2 \cdot \log^\gamma(n_t)$, and (29) holds because of (24) and (25). Combining (27) and (29), we have for any $p_i \in \widetilde{V}_t^{\mathrm{c}}$ that

$$\Phi_{\widetilde{G}_{t'}}(p_i) = \frac{w_{\widetilde{G}_{t'}}(p_i, \widetilde{V}_{t'} \setminus p_i)}{\min\{\mathrm{vol}_{\widetilde{G}_{t'}}(p_i), \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{V}_{t'} \setminus p_i)\}} = O\left(k^{-6} \cdot \log^{-\gamma}(n_t)\right),$$

which proves the second part of statement (C3). □

Corollary B.4. *Suppose for a subset $A \subset V_{t'}$ with $\mathrm{vol}_{G_{t'}}(A) \leqslant \mathrm{vol}(G_{t'})/2$, it holds that $\Phi_{\widetilde{G}_{t'}}(\widetilde{A}) > (21 \cdot c_1) \cdot \log^{-\varepsilon}(n_{t'})$ and $\Phi_{G_{t'}}(A) \leqslant \log^{-\varepsilon}(n_{t'})$ for any positive c_1, ε satisfying $4 \cdot c_1 \leqslant \log^{\varepsilon}(n_{t'})$. Then, it holds that $\Phi_{G_t}(A^{(t)}) < 1/c_1$.*

Proof of Corollary B.4. Assume by contradiction that $\Phi_{G_t}(A^{(t)}) \geqslant 1/c_1$. Then, by statement (C2) in Lemma B.3 and the fact that $\Phi_{G_{t'}}(A) \leqslant \log^{-\varepsilon}(n_{t'})$, it holds that $\Phi_{\widetilde{G}_{t'}}(\widetilde{A}) \leqslant (21 \cdot c_1) \cdot \log^{-\varepsilon}(n_{t'})$, which is a contradiction. Hence, it holds that $\Phi_{G_t}(A^{(t)}) < 1/c_1$. □

Before analysing the spectral gap in the contracted graph $\widetilde{G}_{t'}$ with respect to the spectral gap in the full graph $G_{t'}$, we show that for any small subset of vertices $A \subset V$ with a low value of $\Phi_{G_{t'}}(A)$, the conductance of its corresponding set in the contracted graph $\Phi_{\widetilde{G}_{t'}}(\widetilde{A})$ is low as well.

Lemma B.5. *Let $C \subset V_{t'}$ be a subset of vertices such that $\mathrm{vol}_{G_{t'}}(C) \leqslant k^6 \cdot \log^{2\gamma}(n_t)$ and $\Phi_{G_{t'}}(C) \leqslant \log^{-\varepsilon}(n_{t'})$ for some constant $\varepsilon > 0$. Then, it holds that*

$$\Phi_{\widetilde{G}_{t'}}(\widetilde{C}) = O(\log^{-0.9\varepsilon}(n_{t'})).$$

Proof. We prove this by contradiction. Assume by contradiction that

$$\Phi_{\widetilde{G}_{t'}}(\widetilde{C}) > \frac{21}{4} \cdot \log^{0.1\varepsilon}(n_{t'}) \cdot \log^{-\varepsilon}(n_{t'}) = \frac{21}{4} \cdot \log^{-0.9\varepsilon}(n_{t'}).$$

Setting $c_1 \triangleq (1/4) \cdot \log^{0.1\varepsilon}(n_{t'})$, it holds by Corollary B.4 that

$$\Phi_{G_t}(C^{(t)}) < 4 \cdot \log^{-0.1\varepsilon}(n_{t'}). \tag{30}$$

We will show that $C^{(t)}$ can be used to create a $(k+1)$-partition in G_t with low outer conductance, contradicting with the fact that $\lambda_{k+1}(G_t) = \Omega(1)$.

Let $S_1, \ldots S_k$ be the optimal clusters in G_t corresponding to $\rho_{G_t}(k)$. Given that G_t is a connected graph and $\rho_{G_t}(k) = O\left(k^{-8} \cdot \log^{-2\gamma}(n_t)\right)$, it holds that $\mathrm{vol}_{G_t}(S_i) = \Omega\left(k^8 \cdot \log^{2\gamma}(n_t)\right)$ for all $1 \leqslant i \leqslant k$. We then create the following $(k+1)$-partition:

$$\mathcal{A} \triangleq C^{(t)} \cup \left\{S_1 \setminus C^{(t)}, \ldots, S_k \setminus C^{(t)}\right\},$$

which is a valid partition as we know that $\mathrm{vol}_{G_t}(C^{(t)}) \leqslant \mathrm{vol}_{G_{t'}}(C) \leqslant k^6 \cdot \log^{2\gamma}(n_t)$ by the conditions of the lemma. Now we will compute the conductance of each cluster in $\mathcal{A}$.

First of all, we have from (30) that

$$\Phi_{G_t}(C^{(t)}) < 4 \cdot \log^{-0.1\varepsilon}(n_{t'}). \tag{31}$$

Second, for any cluster $S_j \setminus C^{(t)}$ we have that

$$\Phi_{G_t}(S_j \setminus C^{(t)}) = \frac{w_{G_t}(S_j \setminus C^{(t)}, V_t \setminus (S_j \setminus C^{(t)}))}{\min\{\mathrm{vol}_{G_t}(S_j \setminus C^{(t)}), \mathrm{vol}_{G_t}(V_t \setminus (S_j \setminus C^{(t)}))\}}.$$

Our proof is by the following case distinction:

Case 1: $\min\{\mathrm{vol}_{G_t}(S_j \setminus C^{(t)}), \mathrm{vol}_{G_t}(V_t \setminus (S_j \setminus C^{(t)}))\} = \mathrm{vol}_{G_t}(V_t \setminus (S_j \setminus C^{(t)}))$.

$$\begin{aligned}
\Phi_{G_t}(S_j \setminus C^{(t)}) &= \frac{w_{G_t}(S_j \setminus C^{(t)}, V_t \setminus (S_j \setminus C^{(t)}))}{\mathrm{vol}_{G_t}(V_t \setminus (S_j \setminus C^{(t)}))} \\
&\leqslant \frac{w_{G_t}(S_j, V_t \setminus S_j) + w_{G_t}(C^{(t)}, V_t \setminus C^{(t)})}{\mathrm{vol}_{G_t}(V_t \setminus S_j) + \mathrm{vol}_{G_t}(C^{(t)} \cap S_j)} \\
&\leqslant 2 \cdot \max\left\{\frac{w_{G_t}(S_j, V_t \setminus S_j)}{\mathrm{vol}_{G_t}(V_t \setminus S_j)}, \frac{w_{G_t}(C^{(t)}, V_t \setminus C^{(t)})}{\mathrm{vol}_{G_t}(V_t \setminus S_j)}\right\} \\
&\leqslant 2 \cdot \max\left\{\Phi_{G_t}(S_j), \Phi_{G_t}(C^{(t)})\right\} \qquad (32)\\
&\leqslant \max\{O\left(k^{-8} \cdot \log^{-2\gamma}(n_t)\right), 4 \cdot \log^{-0.1\varepsilon}(n_{t'})\},
\end{aligned}$$

where for (32) it holds that $\min\{\mathrm{vol}_{G_t}(S_j), \mathrm{vol}_{G_t}(V_t \setminus S_j)\} = \mathrm{vol}_{G_t}(V_t \setminus S_j)$ because we know that $\mathrm{vol}(G_t)/2 \geqslant \mathrm{vol}_{G_t}(V_t \setminus (S_j \setminus C^{(t)})) \geqslant \mathrm{vol}_{G_t}(V_t \setminus S_j)$, and we also know that $\mathrm{vol}_{G_t}(V_t \setminus S_j) \geqslant \mathrm{vol}_{G_t}(C^{(t)})$.

Case 2: $\min\{\mathrm{vol}_{G_t}(S_j \setminus C^{(t)}), \mathrm{vol}_{G_t}(V_t \setminus (S_j \setminus C^{(t)}))\} = \mathrm{vol}_{G_t}(S_j \setminus C^{(t)})$.

$$\begin{aligned}
\Phi_{G_t}(S_j \setminus C^{(t)}) &= \frac{w_{G_t}(S_j \setminus C^{(t)}, V_t \setminus (S_j \setminus C^{(t)}))}{\mathrm{vol}_{G_t}(S_j \setminus C^{(t)})} \\
&\leqslant \frac{w_{G_t}(S_j, V_t \setminus S_j) + w_{G_t}(C^{(t)}, V_t \setminus C^{(t)})}{\mathrm{vol}_{G_t}(S_j) - \mathrm{vol}_{G_t}(C^{(t)})} \\
&\leqslant \frac{w_{G_t}(S_j, V_t \setminus S_j) + w_{G_t}(C^{(t)}, V_t \setminus C^{(t)})}{\Omega\left(\mathrm{vol}_{G_t}(S_j)\right)} \qquad (33)\\
&= O\left(\Phi_{G_t}(S_j)\right) + O\left(\Phi_{G_t}(C^{(t)})\right) \qquad (34)\\
&= 2 \cdot \max\left\{O\left(k^{-8} \cdot \log^{-2\gamma}(n_t)\right), O\left(\log^{-0.1\varepsilon}(n_{t'})\right)\right\} \qquad (35)
\end{aligned}$$

where (33) holds because $\mathrm{vol}_{G_t}(S_i) = \Omega\left(k^8 \cdot \log^{2\gamma}(n_t)\right)$ and $\mathrm{vol}_{G_t}(C^{(t)}) = O\left(k^6 \cdot \log^{2\gamma}(n_t)\right)$, (34) holds because $w_{G_t}(S_j, V_t \setminus S_j) \leqslant \Phi_{G_t}(S_j) \cdot \mathrm{vol}_{G_t}(S_j)$ and $w_{G_t}(C^{(t)}, V_t \setminus C^{(t)}) \leqslant \Phi_{G_t}(C^{(t)}) \cdot \mathrm{vol}_{G_t}(C^{(t)})$.

Combining both cases, we have for every $1 \leqslant j \leqslant k$ that

$$\Phi_{G_t}(S_j \setminus C^{(t)}) = 2 \cdot \max\left\{O\left(k^{-8} \cdot \log^{-2\gamma}(n_t)\right), O\left(\log^{-0.1\varepsilon}(n_{t'})\right)\right\}. \tag{36}$$

Therefore, by combining (31) and (36), we have shown that

$$\rho_{G_t}(k+1) \leqslant \max_{A_j \in \mathcal{A}} \Phi_{G_t}(A_j) = 2 \cdot \max\left\{O\left(k^{-8} \cdot \log^{-2\gamma}(n_t)\right), O\left(\log^{-0.1\varepsilon}(n_{t'})\right)\right\},$$

which contradicts the fact that $\rho_{G_t}(k+1) \geqslant \frac{\lambda_{k+1}(\mathcal{L}_{G_t})}{2} = \Omega(1)$. Hence, the statement of the lemma follows. □

Proof of Lemma 4.4. We first prove the first statement. Let $\mathcal{S} = S_1, \ldots, S_\ell$ be a set of clusters that achieve $\rho_{G_{t'}}(\ell)$. For ease of notation we set

$$\mathcal{S}_{\mathrm{small}} \triangleq \mathcal{S}^{(t')}_{\mathrm{small}}\left(k^6 \cdot \log^{2\gamma}(n_t)\right)$$

to be the clusters in $\mathcal{S}$ with volume at most $k^6 \cdot \log^{2\gamma}(n_t)$, and similarly

$$\mathcal{S}_{\mathrm{large}} \triangleq \mathcal{S}^{(t')}_{\mathrm{large}}\left(k^6 \cdot \log^{2\gamma}(n_t)\right).$$

We will use the partition $\mathcal{S}$, which has low outer conductance in $G_{t'}$, to create an r-way partition in $\widetilde{G}_{t'}$ with low r-way expansion. We construct this r-way partition, denoted by $\mathcal{R}$, as follows:

$$\mathcal{R} \triangleq \left\{\widetilde{S}_1, \ldots, \widetilde{S}_{\ell_1}, p_1, \ldots, p_{k-1}, p_k^*\right\}$$

where $\ell_1 \triangleq |\mathcal{S}_{\mathrm{small}}|$, and we define

$$p_k^* \triangleq p_k \cup \left(\widetilde{V}^{\mathrm{nc}}_{t'} \setminus \bigcup_{S_j \in \mathcal{S}_{\mathrm{small}}} \widetilde{S}_j\right)$$

to be the union of the super node p_k with the leftover non-contracted vertices which do not belong to any $\widetilde{S}_j$. We start by showing that $\mathcal{R}$ has low r-way expansion:

- By Lemma B.5, we know that for every $S_j \in \mathcal{S}_{\mathrm{small}}$, it holds that $\Phi_{\widetilde{G}_{t'}}(\widetilde{S}_j) = O\left(\log^{-0.9\alpha}(n_{t'})\right)$.
- By Property (C3) of Lemma B.3, we know that for every super node $p_i \in \{p_{k_1}, \ldots p_{k-1}\}$ it holds that $\Phi_{\widetilde{G}_{t'}}(p_i) = O\left(k^{-6} \cdot \log^{-\gamma}(n_{t'})\right)$.
- Finally, for p_k^* we know that

$$\Phi_{\widetilde{G}_{t'}}(p_k^*) = \frac{w_{\widetilde{G}_{t'}}(p_k^*, \widetilde{V}_{t'} \setminus p_k^*)}{\min\left\{\mathrm{vol}_{\widetilde{G}_{t'}}(p_k^*), \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{V}_{t'} \setminus p_k^*)\right\}}. \tag{37}$$

We split the computation of this conductance into two cases.

Case 1: Suppose $\min\left\{\mathrm{vol}_{\widetilde{G}_{t'}}(p_k^*), \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{V}_{t'} \setminus p_k^*)\right\} = \mathrm{vol}_{\widetilde{G}_{t'}}(p_k^*)$. Then, we have that

$$
\begin{aligned}
\Phi_{\widetilde{G}_{t'}}(p_k^*) &= \frac{w_{\widetilde{G}_{t'}}(p_k^*, \widetilde{V}_{t'} \setminus p_k^*)}{\mathrm{vol}_{\widetilde{G}_{t'}}(p_k^*)} \\
&\leqslant \frac{w_{\widetilde{G}_t}(p_k, \widetilde{V}_t \setminus p_k) + \log^{\gamma}(n_{t'}) + \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{V}_{t'}^{\mathrm{nc}})}{\mathrm{vol}_{\widetilde{G}_t}(p_k) - \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{V}_{t'}^{\mathrm{nc}})} && (38) \\
&\leqslant \frac{\Phi_{\widetilde{G}_t}(p_k) \cdot \mathrm{vol}_{\widetilde{G}_t}(p_k) + 3 \cdot \log^{\gamma}(n_{t'})}{\Omega\left(\mathrm{vol}_{\widetilde{G}_t}(p_k)\right)} && (39) \\
&= \frac{O\left(k^{-6} \cdot \log^{-2\gamma}(n_t)\right) \cdot \mathrm{vol}_{\widetilde{G}_{t'}}(p_k) + 3 \cdot \log^{\gamma}(n_{t'})}{\Omega\left(\mathrm{vol}_{\widetilde{G}_t}(p_k)\right)} && (40) \\
&= \frac{O\left(k^{-6} \cdot \log^{-2\gamma}(n_t) \cdot \mathrm{vol}_{\widetilde{G}_{t'}}(p_k)\right)}{\Omega\left(\mathrm{vol}_{\widetilde{G}_t}(p_k)\right)} && (41) \\
&= O\left(k^{-6} \cdot \log^{-2\gamma}(n_t)\right),
\end{aligned}
$$

where (38) holds because $|E_{\mathrm{new}}| \leqslant \log^{\gamma}(n_t)$ is the maximum amount of weight that can be added between p_k and its complement, (39) holds because $\mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{V}_{t'}^{\mathrm{nc}}) \leqslant 2 \cdot |E_{\mathrm{new}}|$ is the maximum volume of non-contracted vertices that can be added to $\widetilde{G}_{t'}$ and (40) holds because of statement (C3) of Lemma B.3, and (41) holds since $\mathrm{vol}_{\widetilde{G}_{t'}}(p_k) \geqslant \mathrm{vol}(G_t)/k \geqslant n_t/k$.

Case 2: Suppose $\min\left\{\mathrm{vol}_{\widetilde{G}_{t'}}(p_k^*), \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{V}_{t'} \setminus p_k^*)\right\} = \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{V}_{t'} \setminus p_k^*)$. Then it holds that the conductance of p_k^* is upper bounded by the maximum conductance of every other cluster in $\mathcal{R}$, i.e.,

$$
\begin{aligned}
\Phi_{\widetilde{G}_{t'}}(p_k^*) &= \frac{w_{\widetilde{G}_{t'}}(p_k^*, \widetilde{V}_{t'} \setminus p_k^*)}{\mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{V}_{t'} \setminus p_k^*)} \\
&\leqslant \frac{\sum_{S_j \in \mathcal{S}_{\mathrm{small}}} w_{\widetilde{G}_{t'}}(\widetilde{S}_j, \widetilde{V}_{t'} \setminus \widetilde{S}_j) + \sum_{j=1}^{k-1} w_{\widetilde{G}_{t'}}(p_j, \widetilde{V}_{t'} \setminus p_j)}{\sum_{S_j \in \mathcal{S}_{\mathrm{small}}} \mathrm{vol}_{\widetilde{G}_{t'}}(\widetilde{S}_j) + \sum_{j=1}^{k-1} \mathrm{vol}_{\widetilde{G}_{t'}}(p_j)} \\
&\leqslant \max\left\{\max_{S_j \in \mathcal{S}_{\mathrm{small}}}\left\{\Phi_{\widetilde{G}_{t'}}(\widetilde{S}_j)\right\}, \max_{p_j \in \{p_1, \ldots, p_{k-1}\}}\left\{\Phi_{\widetilde{G}_{t'}}(p_j)\right\}\right\} \\
&= \max\left\{O\left(\log^{-0.9\alpha}(n_{t'})\right), O\left(k^{-6} \cdot \log^{-\gamma}(n_{t'})\right)\right\},
\end{aligned}
$$

where the last inequality follows by the mediant inequality.

Combining the two cases, we have that

$$
\Phi_{\widetilde{G}_{t'}}(p_k^*) = \max\left\{O\left(\log^{-0.9\alpha}(n_{t'})\right), O\left(k^{-6} \cdot \log^{-\gamma}(n_{t'})\right)\right\}.
$$

We have so far analysed the conductance of each of the clusters in the partition $\mathcal{R}$, and have shown that

$$
\rho_{\widetilde{G}_{t'}}(r) = \max\left\{O\left(\log^{-0.9\alpha}(n_{t'})\right), O\left(k^{-6} \cdot \log^{-\gamma}(n_{t'})\right)\right\}. \tag{42}
$$

Before reaching the final contradiction, we prove the following claim.

Claim B.5.1. It holds that $r \geqslant \ell$.

Proof. Assume by contradiction that $r < \ell$. In this case, we know that $r = |\mathcal{S}_{\mathrm{small}}| + |\mathcal{P}|$, and $\ell = |\mathcal{S}|$. Therefore, the condition of $r < \ell$ gives us that $|\mathcal{S}_{\mathrm{small}}| + |\mathcal{P}| < |\mathcal{S}|$, which implies that $|P| < |\mathcal{S}| - |\mathcal{S}_{\mathrm{small}}| = |\mathcal{S}_{\mathrm{large}}|$. This means that the number of large clusters in $\mathcal{S}$ is greater than the number of clusters in $\mathcal{P}$.

It therefore holds that $|\mathcal{S}_{\mathrm{large}}| > |\mathcal{P}| = k$. Furthermore, since it holds that for every $S_j \in \mathcal{S}_{\mathrm{large}}$ that $\mathrm{vol}_{G_{t'}}(S_j) > k^6 \cdot \log^{2\gamma}(n_t)$, and the number of new edges is $|E_{\mathrm{new}}| \leqslant \log^{\gamma}(n_t)$, it also holds that

$$\Phi_{G_t}(S_j) = O\left(\Phi_{G_{t'}}(S_j)\right) = \max\left\{O\left(\log^{-\alpha}(n_{t'})\right), k^6 \cdot \log^{\gamma}(n_t)\right\}.$$

This means that $\mathcal{S}_{\mathrm{large}}$ is a set of $|\mathcal{S}_{\mathrm{large}}| \geqslant k+1$ disjoint subsets in G_t with low conductance, which contradicts the higher-order Cheeger inequality and proves the claim. □

Combining (42) with Claim B.5.1 gives us that

$$\rho_{\widetilde{G}_{t'}}(\ell) = \max\left\{O\left(\log^{-0.9\alpha}(n_{t'})\right), O\left(k^{-6} \cdot \log^{-\gamma}(n_{t'})\right)\right\},$$

and this proves the first statement.

Next we prove the second statement. Let $A_1, \ldots A_\ell$ be the partition such that $\Phi_{\widetilde{G}_{t'}}(A_i) = O\left(\log^{-\delta}(n_{t'})\right)$ for every $1 \leqslant i \leqslant \ell$. Recall that $\hat{A}_i$ is the representation of the set A_i in the full graph $G_{t'}$, i.e.,

$$\hat{A}_i \triangleq A_i^{\mathrm{nc}} \bigcup \left(\bigcup_{p_j \in A_i^{\mathrm{c}}} P_j^{(t')} \right),$$

where $P_j^{(t')} = P_j \setminus (P_j \cap \widetilde{V}_{t'}^{\mathrm{nc}})$, $A_i^{\mathrm{nc}} \triangleq A_i \cap V_{t'}^{\mathrm{nc}}$, and $A_i^{\mathrm{c}} \triangleq A_i \cap V_{t'}^{\mathrm{c}}$. One can see $P_j^{(t')}$ as the vertices in P_j that have not been pulled out into the contracted graph yet.

Notice that, when $A_i^{\mathrm{c}} = \emptyset$, it holds by construction that $\Phi_{G_{t'}}(A_i) = \Phi_{\widetilde{G}_{t'}}(A_i) \leqslant \log^{-\delta}(n_{t'})$. So we only look at the case where $A_i^{\mathrm{c}} \neq \emptyset$. Without loss of generality, we assume that A_i^c does not contain all the contracted nodes $p_1, \ldots, p_k$. If it did, then

$$\Phi_{G_{t'}}(\hat{A}_i) = \Phi_{G_{t'}}\left(\bigcup_{A_j^{\mathrm{c}} = \emptyset} A_j \right) \leqslant \log^{-\delta}(n_{t'}).$$

Therefore, given that for any $1 \leqslant i \leqslant \ell$ it holds that $\mathrm{vol}_{G_{t'}}(A_i^{\mathrm{nc}}) \leqslant 2 \cdot |E_{\mathrm{new}}| \leqslant 2 \cdot \log^{\gamma}(n_t)$, and for any $1 \leqslant j \leqslant k$ it holds that $\mathrm{vol}_{G_{t'}}(P_i^{(t')}) = \Omega(k^6 \cdot \log^{2\gamma}(n_t))$, we get that

$$\Phi_{G_{t'}}(\hat{A}_i) = O\left(\Phi_{G_{t'}}\left(\bigcup_{p_j \in A_i^{\mathrm{c}}} P_j^{(t')} \right) \right) = O\left(k^{-6} \cdot \log^{-\gamma}(n_{t'})\right),$$

where the last line holds because of property (C3) of Lemma B.3. This proves the second statement. □

Proof of Lemma 4.5. We first prove the first statement, and we will prove this by contradiction. Assume by contradiction that $\lambda_{\ell+1}(\mathcal{L}_{G_{t'}}) < C \cdot \frac{\log^{-\alpha}(n_{t'})}{(\ell+1)^6}$ for some constant C. Then, by the higher-order Cheeger inequality, there exists an optimal $(\ell+1)$-way partition $\mathcal{S} = \{S_1, \ldots S_{\ell+1}\}$ such that for all $1 \leqslant i \leqslant \ell+1$

$$\Phi_{G_{t'}}(S_i) \leqslant \rho_{G_{t'}}(\ell+1) \leqslant C_{2.1} \cdot (\ell+1)^3 \cdot \sqrt{\lambda_{\ell+1}(G_{t'})} = O\left(\log^{-0.5\alpha}(n_{t'})\right).$$

By Lemma 4.4, it then holds that $\rho_{\widetilde{G}_{t'}}(\ell+1) = \max\left\{O\left(\log^{-0.45\alpha}(n_{t'})\right), O\left(k^{-6} \cdot \log^{-\gamma}(n_{t'})\right)\right\}$, which contradicts the fact that

$$\rho_{\widetilde{G}_{t'}}(\ell+1) \geqslant \frac{\lambda_{\ell+1}(\mathcal{L}_{\widetilde{G}_{t'}})}{2} = \Omega(1),$$

from which the first statement of the lemma follows.

Next we prove the second statement. We prove this by analysing the spectrum of $\mathcal{L}_{\widetilde{G}_{t'}}$ with respect to $\mathcal{L}_{G_{t'}}$ through $\mathcal{L}_{H'_{t'}}$. As proven in Lemma B.2, $H'_{t'}$ is a cluster preserving sparsifier of $G_{t'}$, and therefore we know that

$$\lambda_{\ell+1}(\mathcal{L}_{H'_{t'}}) = \Omega(\lambda_{\ell+1}(\mathcal{L}_{G_{t'}})). \tag{43}$$

Our next analysis is inspired by the work on meta graphs of Macgregor and Sun (Macgregor & Sun, 2022). We will analyse the spectrum of $\mathcal{L}_{H'_{t'}}$ with respect to the spectrum of $\mathcal{L}_{\widetilde{G}_{t'}}$, and for simplicity we denote $H'_{t'} \triangleq H$, $\widetilde{G}_{t'} \triangleq \widetilde{G}$, and $n_{t'} \triangleq n$. For every vertex $u_j \in V(\widetilde{G})$ in the contracted graph, we associate it with a non-empty group of vertices $A_j \subset V(H)$ as follows: for all $u_j \in \widetilde{V}_{t'}^{\mathrm{nc}}$, we associate u_j with its unique corresponding single vertex $v \in V(H)$, and for every $u_j = p_r \in \widetilde{V}_{t'}^{\mathrm{c}}$ for some r, we associate it with its corresponding vertices in the cluster $P_r^{(t')} \subset V(H)$. Then, let $\chi_j \in \mathbb{R}^n$ be the indicator vector for the vertices $A_j \subset V(H)$ corresponding to the vertex $u_j \in V(\widetilde{G})$.

We define $\widetilde{n} = |V(\widetilde{G})|$, and let the eigenvalues of $\mathcal{L}_{\widetilde{G}}$ be $\gamma_1 \leqslant \gamma_2 \leqslant \ldots \leqslant \gamma_{\widetilde{n}}$ with corresponding eigenvectors $g_1, g_2, \ldots, g_{\widetilde{n}} \in \mathbb{R}^{\widetilde{n}}$. We further define vectors $\bar{g}_1, \ldots \bar{g}_{\widetilde{n}}$ which will represent the eigenvectors $g_1, \ldots g_{\widetilde{n}}$ of the normalised Laplacian $\mathcal{L}_{\widetilde{G}}$, but blown up to size $\mathbb{R}^n$. Formally, we define

$$\bar{g}_i \triangleq \sum_{j=1}^{\widetilde{n}} \frac{D_H^{\frac{1}{2}} \chi_j}{\|D_H^{\frac{1}{2}} \chi_j\|} g_i(j).$$

We can readily check that these vectors form an orthonormal basis. First,

$$\begin{aligned}
\bar{g}_i \bar{g}_i^{\intercal} &= \sum_{j=1}^{\widetilde{n}} \sum_{u \in A_j} \left(\frac{\sqrt{d_H(u)}}{\sqrt{\mathrm{vol}_H(A_j)}} g_i(j) \right)^2 \\
&= \sum_{j=1}^{\widetilde{n}} g_i(j)^2 \sum_{u \in A_j} \frac{d_H(u)}{\mathrm{vol}_H(A_j)} \\
&= \sum_{j=1}^{\widetilde{n}} g(j)^2 = 1.
\end{aligned}$$

And similarly for any $i_1 \neq i_2$,

$$\begin{aligned}
\bar{g}_{i_1} \bar{g}_{i_2}^{\intercal} &= \sum_{j=1}^{\widetilde{n}} \sum_{u \in A_j} \frac{d_H(u)}{\mathrm{vol}_H(A_j)} g_{i_1}(j) g_{i_2}(j) \\
&= \sum_{j=1}^{\widetilde{n}} g_{i_1}(j) g_{i_2}(j) = 0.
\end{aligned}$$

We also get the useful property that for the eigenvalues $\lambda_1, \ldots, \lambda_n$ of $\mathcal{L}_H$ and $\gamma_1, \ldots, \gamma_{\widetilde{n}}$ of the contracted Laplacian $\mathcal{L}_{\widetilde{G}}$, it holds that $\lambda_i \leqslant 2 \cdot \nu_i$. In particular,

$$\begin{aligned}
\bar{g}_i \mathcal{L}_H \bar{g}_i^{\intercal} &= \sum_{x=1}^{\widetilde{n}} \sum_{y=x}^{\widetilde{n}} \sum_{u \in A_x} \sum_{v \in A_y} w_H(u,v) \left(\frac{\bar{g}_i(u)}{\sqrt{d_H(u)}} - \frac{\bar{g}_i(v)}{\sqrt{d_H(v)}} \right)^2 \\
&= \sum_{x=1}^{\widetilde{n}} \sum_{y=x}^{\widetilde{n}} w_H(A_x, A_y) \left(\frac{g_i(x)}{\sqrt{\mathrm{vol}_H(A_x)}} - \frac{g_i(y)}{\sqrt{\mathrm{vol}_H(A_y)}} \right)^2 \\
&= 2 \cdot g_i \mathcal{L}_{\widetilde{G}} g_i^{\intercal}.
\end{aligned}$$

Therefore we have an i-dimensional subspace X_i such that

$$\max_{x \in X_i} \frac{x^{\intercal} \mathcal{L}_H x}{x^{\intercal} x} = 2 \cdot \gamma_i,$$

from which it follows by the Courant-Fischer theorem that $\lambda_i \leqslant 2 \cdot \gamma_i$. Combining this with (43), we get that

$$\lambda_{\ell+1}(\mathcal{L}_{\widetilde{G}_{t'}}) \geqslant \frac{1}{2} \cdot \lambda_{\ell+1}(\mathcal{L}_{H'_{t'}}) = \Omega\left(\lambda_{\ell+1}(\mathcal{L}_{G_{t'}})\right) = \Omega(1),$$

which proves the lemma. □