Dynamic Correlation Clustering in Sublinear Update Time

Vincent Cohen-Addad [*1] **Silvio Lattanzi** [*1] **Andreas Maggiori** [*2] **Nikos Parotsidis** [*1]

Abstract

We study the classic problem of correlation clustering in dynamic node streams. In this setting, nodes are either added or randomly deleted over time, and each node pair is connected by a positive or negative edge. The objective is to continuously find a partition which minimizes the sum of positive edges crossing clusters and negative edges within clusters. We present an algorithm that maintains an $O(1)$-approximation with $O(\text{polylog}\, n)$ amortized update time. Prior to our work Behnezhad et al. (2023b) achieved a 5-approximation with $O(1)$ expected update time in edge streams which translates in node streams to an $O(D)$-update time where D is the maximum possible degree. Finally we complement our theoretical analysis with experiments on real world data.

1. Introduction

Clustering is a cornerstone of contemporary machine learning and data analysis. A successful clustering algorithm partitions data elements so that similar items reside within the same group, while dissimilar items are separated. Introduced in 2004 by Bansal, Blum and Chawla (Bansal et al., 2004), the correlation clustering objective offers a natural approach to model this problem. Due to its concise and elegant formulation, this problem has drawn significant interest from researchers and practitioners, leading to applications across diverse domains. These include ensemble clustering identification (Bonchi et al., 2013), duplicate detection (Arasu et al., 2009), community mining (Chen et al., 2012), disambiguation tasks (Kalashnikov et al., 2008), automated labeling (Agrawal et al., 2009; Chakrabarti et al., 2008), and many more.

In the correlation clustering problem we are given a graph where each edge has either a positive or negative label, and where a positive edge (u, v) indicates that u, v are similar elements (and a negative edge (u, v) indicates that u, v are dissimilar), the objective is to compute a partition of the graph that minimizes the number of negative edges within clusters plus positive edges between clusters. Since the problem is NP-hard, researchers have focused on designing approximation algorithms.

The algorithm proposed by Cao et al. (2024) achieves an approximation ratio of $1.43 + \epsilon$, improving upon the previous $1.73 + \epsilon$ and $1.994 + \epsilon$ achieved by Cohen-Addad et al. (2023; 2022b). Prior to these developments, the best approximation guarantee of 2.06 was attained by the algorithm of Chawla et al. (2015).[1]

These above approaches are linear-programming-based: they require to solve a linear program and then provide a rounding algorithm. The best known "combinatorial" algorithm is due to a recent local search algorithm of Cohen-Addad et al. (2024) achieving a $1.845 + \varepsilon$-approximation. Prior to this, the best known "combinatorial" algorithm was the celebrated pivot algorithm of Ailon et al. (2008) which consists in repeatedly creating a cluster by picking a random unclustered node, and clustering it with all its positive unclustered neighbors. The versatility of the scheme has led the pivot algorithm to be used in a variety of contexts, and in particular for designing dynamic algorithms (Behnezhad et al., 2019; 2022; 2023a).

Dynamic algorithms hold a key position in algorithm design due to their relevance in handling real-world, evolving datasets. Consequently, substantial research has focused on crafting clustering algorithms expressly designed for dynamic environments (including streaming, online, and distributed settings) (Lattanzi & Vassilvitskii, 2017; Fichtenberger et al., 2021; Jaghargh et al., 2019; Cohen-Addad et al., 2019; Guo et al., 2021; Cohen-Addad et al., 2022a; Assadi & Wang, 2022; Lattanzi et al., 2021; Behnezhad et al., 2022; 2023a; Bateni et al., 2023).

[*]Equal contribution [1]Google Research [2]Columbia University. Correspondence to: Andreas Maggiori <am6292@columbia.edu>.

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

[1]Note also that there is also a version of the problem (Bansal et al., 2004) where the objective is to maximize the number of positive edges whose both endpoints are in the same cluster plus the number of negative edges across clusters. Similarly, if the input is weighted an $O(\log n)$ approximation has been shown by Demaine et al. (Demaine et al., 2006).

While the classical approach is to design variations of the Pivot algorithm of Ailon et al. (2008), Cohen-Addad et al. (2021) provide an alternative approach based on a notion called *agreement*, which entails the calculation of the positive neighborhood similarity of pairs of nodes. While that approach is initially used in the context of distributed correlation clustering, it has been also used in (Assadi & Wang, 2022) to design a static algorithm with $n \operatorname{polylog} n$ time complexity. In the latter setting the algorithm does not read the entire input, otherwise a time complexity of $n \operatorname{polylog} n$ would be impossible for dense graphs. However the algorithm has access to the graph through queries. We formalize that model in Definition 1 and ask the following natural question: in which settings can correlation clustering be solved in $o(n^2)$ time?

Our Contribution In this paper, our focus lies on the dynamic case, where nodes are inserted adversarially and / or deleted randomly over time. This setting has already been studied for other clustering problems in (Epasto et al., 2015) and serves as a bridge between the fully adversarial and random input models. Our objective is to maintain an $O(1)$-approximate solution at any point in time, while paying as little computation time as possible upon modification of the input (node insertion or deletion). Unfortunately, the approximation algorithms mentioned above are computationally expensive and cannot be re-executed each time the input changes.

The best known bound for the fully dynamic setting is due to Behnezhad et al. (2023a) who provided a 5-approximation in $O(m)$ total update time for adversarial edge insertions and deletions, where m is the total number of positive edges of the graph. Of course, for node updates, the above approach can be used to achieve a 5-approximation with $O(D)$ update time, where D is the maximum positive degree of a node throughout the vertex sequence. Note that for dense graphs this is equivalent to a $\Theta(n^2)$ algorithm. We ask whether it is possible to go beyond this bound if we are given indirect access to the graph through queries and we do not need to read the entire input.

We answer the above question positively. More precisely, we provide the first algorithm which achieves a constant factor approximation model with poly-logarithmic update time per node insertion/deletion . We also complement our theoretical result with experiments showing the effectiveness of our algorithm in practice.

2. Problem Definition and the Database Model of Computation

The *disagreements minimization* version of the correlation clustering problem receives as input a complete signed undirected graph $G = (V, E, s)$ where each edge $e = \{u, v\}$ is assigned a sign $s(e) \in \{`+', `-'\}$ and the goal is to find a partition of the nodes such that the number of '$-$' edges inside the same cluster and '$+$' edges in between clusters is minimized. For simplicity we denote the set of '$+$' and '$-$' edges by E^+ and E^- respectively. A *clustering* is a partition of the nodes $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ and the cost of that clustering is $|\{u, v\} \in E^+ : u \in C_i, v \in C_j, i \neq j| + |\{u, v\} \in E^- : \exists i : u, v \in C_i|$

Note that a complete undirected signed graph $G = (V, E, s)$ can be converted into a non-signed undirected graph $G = (V, E)$ where for each pair of nodes $\{u, v\}$ there is an edge between them in G if and only if $s(\{u, v\}) = `+'$. Thus the absence of an edge between two nodes corresponds to a negative edge in the original signed graph and the presence of an edge to a positive edge in the original graph. For simplicity, throughout the paper we work with the non-signed equivalent definition of the correlation clustering problem.

The cost of a clustering $\mathcal{C}$ becomes:

$$|\{u, v\} \in E : u \in C_i, v \in C_j, i \neq j| + \\ |\{u, v\} \notin E : \exists i : u, v \in C_i|$$

In our setting, nodes arrival are adversarial and node deletions are random. More precisely, at each time t an adversary can decide either to add to the graph an adversarially chosen node or to delete a random node from the graph. Upon arrival a node reveals all the edges to previously arrived nodes and upon deletion all edges of the node are deleted. We denote by u_t the node that arrived or left at time t and by G_t the graph structure after the first t node arrivals/deletions. We also denote by V_t, E_t the set of nodes and edges of graph G_t and by n, m the total number of nodes and edges respectively that appeared throughout the dynamic stream. Note that $\forall t$ we have $|V_t| \leqslant n$ and $|E_t| \leqslant m$.

We denote by OPT_t the cost of an optimal correlation clustering solution for graph G_t and by ALG_t the cost of a dynamic algorithm solution at time t. We say that an algorithm maintains a $c-$approximate solution if $\forall t$, input graphs and node streams we have $ALG_t \leqslant c \cdot OPT_t$.

We now formally define the computation model that we study in the paper. For an unsigned graph $G = (V, E)$ we denote by V, E the set of nodes and edges respectively, and for a node $u \in V$ we use $N_G(u)$ to denote the neighborhood of u in G. This model was considered by Assadi and Wang (Assadi & Wang, 2022) for designing sublinear algorithms for correlation clustering.

Definition 1 (Database model (Assadi & Wang, 2022)). *Given a graph $G = (V, E)$ we have access to the graph structure through the following queries which have a cost* *of* $O(\log|V|)$ *:*

1. *Degree queries:* $\forall u \in V$ *we get its degree* $|N_G(u)|$
2. *Edge queries:* $\forall u, v \in V$ *we get whether* $(u, v) \in E$
3. *Neighborhood sample queries:* $\forall u$*, we get a node* $v \in N_G(u)$ *uniformly at random from set* $N_G(u)$

Note that all these queries are easily implementable in the classical computational model where the graph is stored in the same processing unit as the one we use to compute our clustering solution. Thus, other than permitting us to avoid reading and storing the graph locally the Database model is strictly harder than the classical RAM computational model. Our goal is to maintain a constant approximation with respect to OPT_t using only $\text{polylog}\, n$ amortized update time (queries and computational operations). Note that for a dense graph this is sublinear in the time of reading the entire input sequence.

3. Algorithm and Techniques

Our approach draws inspiration from the Agreement algorithm, initially presented in (Cohen-Addad et al., 2021). In particular, we leverage their key insight that to obtain a constant factor approximation it is enough to cluster together nodes with similar neighborhoods. Essentially, it is enough to focus on identifying near-clique structures. A second key idea that we use comes from Assadi & Wang (2022) where it is noted that to discover these dense substructures one does not need to examine the entire neighborhood of each node but it is possible to carefully sub-sample the edges of the graph to obtain a sparser structure. We build upon these two ideas along with developing several new techniques to obtain a constant factor approximation algorithm for dynamic graph with sublinear complexity.

The section is structured as follows: (1) we introduce the Agreement algorithm of (Cohen-Addad et al., 2021) along with its useful properties; (2) we describe the challenges in applying that algorithm on a dynamic graph; (3) we describe a *notification* procedure which is the base of our algorithm; and (4) provide the pseudocode of our algorihtm.

3.1. The Agreement Algorithm

Before describing the Agreement algorithm of (Cohen-Addad et al., 2021) we need to introduce two central notions to quantify the similarity between the neighborhood of two nodes.

Definition 2 (Agreement)**.** *Two nodes* u, v *are in* ϵ*-agreement in* G *if*

$$|N_G(u) \triangle N_G(v)| < \epsilon \max\{|N_G(u)|, |N_G(v)|\}$$

where $\triangle$ *denotes the symmetric difference of two sets.*

Definition 3 (Heaviness)**.** *A node is called* ϵ-heavy *if it is in* ϵ*-agreement with more than a* $(1-\epsilon)$*-fraction of its neighbors. Otherwise it is called* ϵ-light*.*

When ϵ is clear from the context we will simply say that two nodes are or are not in agreement and that a node is heavy or light.

The Agreement algorithm uses the agreement and heaviness definitions to compute a solution to the correlation clustering problem, as described in Algorithm 1. We call the output of Algorithm 1 the agreement decomposition of the graph G.

Algorithm 1 AGREEMENTALGORITHM(G)

Create a graph $\tilde{G}$ from G by discarding all edges whose endpoints are not in ϵ-agreement.
Discard all edges of $\tilde{G}$ between light nodes of G.
Compute the connected components of $\tilde{G}$, and output them as the solution.

At a high level, the first two steps of Algorithm 1 can be characterized as a filtering procedure which ensures that two nodes with similar neighborhoods end up in the same connected component of $\tilde{G}$ and consequently in the same cluster of the final partitioning. The main lemma which helps bounding the approximation ratio of the Agreement algorithm and which will be also used to analyze the performance of our algorithm is the following: [2]

Lemma 4 (rephrased from (Cohen-Addad et al., 2021))**.** *Let* $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ *be a clustering solution for graph* $G = (V, E)$ *and* ϵ *a small enough constant. If the following properties hold:*

1. $\forall i \in \{1, 2, \ldots, k\}$ *and* $u \in C_i$ *such that* $|C_i| > 1$ *we have* $|N_G(u) \cap C_i| \geqslant 3/4|C_i|$
2. $\forall e = (u, v) \in E$ *such that* $u \in C_i$*,* $v \in C_j$ *and* $i \neq j$ *then either* u *and* v *are not in* ϵ*-agreement or both nodes are not* ϵ*-heavy.*

Then the cost of $\mathcal{C}$ *is a constant factor approximation to that of the optimal correlation clustering solution for graph* G

For ϵ small enough Cohen-Addad et al. (2021) prove that Algorithm 1 satisfies both properties of Lemma 4 and therefore produces a constant factor approximation.

3.2. Challenges of Dynamic Agreement

Our goal is to design a dynamic version of the Agreement algorithm which consistently maintains a sparse

[2] We note that Lemma 4 is not explicitly stated in (Cohen-Addad et al., 2021) but it is a combination of lemmas 3.5, 3.6, 3.7 and 3.8 in the latter paper

graph $\tilde{G}$ whose induced clustering satisfy both conditions of Lemma 4 while only spending $O(\text{polylog}\, n)$ update time upon node insertions and random deletions.

We briefly describe what are the main challenges that we face in such endeavour:

1. computing the agreement between two nodes or the heaviness of a node may take time $\Theta(n)$;
2. the number of agreement calculations performed by Algorithm 1 is equal to the number of edges in our graph; and
3. since the total complexity that we aim is $O(n\, \text{polylog}\, n)$, the graph $\tilde{G}$ that we maintain should be both sparse at any point and stable (do not change significantly between consecutive times).

From a high level perspective, we solve those issues as follows. First, instead of computing exactly whether two nodes are in ϵ-agreement and whether a node is ϵ-heavy, we design two stochastic procedures, namely PROBABILISTICAGREEMENT(u, v, ϵ) and HEAVY(u, ϵ), which only need a sample of logarithmic size of the two neighborhoods to answer correctly, with high probability, those questions. We defer the description of those procedures in Appendix E. Second, for each dense substructure, we maintain dynamically a random set of $O(\text{polylog}\, n)$ heavy nodes. We call that sample the *anchor* set and show that the connections of those nodes are enough to recover a good clustering. Finally, to efficiently maintain our sparse graph $\tilde{G}$ we design a message-passing procedure, which we call Notify, to communicate events across neighboring nodes. Roughly, this procedure propagates information about the arrival or deletion of a node u to a $O(\text{polylog}\, n)$-size randomly chosen subset of nodes within small hop distance from u. Whenever a node receives a "notification", we either add, with some probability, this node to the *anchor* set and we revisit the agreement between that node and nodes already in the anchor set.

While similar ideas to resolve the first and second challenges have been already explored in the sublinear static algorithm of Assadi & Wang (2022) it is important to observe that applying the same principles in the dynamic sublinear setting is highly non-trivial. Indeed, in this setting it is not even clear if one can even maintain a good approximation of the degrees of nodes in sublinear time[3].

3.3. Notify Procedure

A central sub-procedure in our algorithm, which allow us to keep track of evolving clusters is the Notify procedure. We believe that it is of independent interest and we devote Section 3.3 entirely to its description. As mentioned before the Notify procedure is responsible to propagate the information of node arrivals and deletions to $O(\text{polylog}\, n)$ nodes. We distinguish between different types of notifications depending on how many nodes did the notification propagate through from the "source" node that initiated the notify procedure.

Notifications are subdivided in categories depending on their type, which could be $\text{Type}_i, i = 0, 1, 2$. A central definition in our algorithm and in its analysis is the "interesting event" definition.

Definition 5. *We say that u participates in an "interesting event" either when u arrives or u receives a Type_0 or Type_1 notification.*

To simplify the description of the algorithm. We denote by $d(u)$ the current degree of a node u and we define the function $l(x) = \lfloor \log(x) \rfloor, \forall x > 0$. With a slight abuse of notation for a node u we denote by l_u the quantity $l(d(u))$. Note that for all nodes u we have that $2^{l_u} \leqslant d(u) < 2^{l_u+1}$.

Each node u stores sets $I_u^0, I_u^1, I_u^2, \dots, I_u^{\log n}$ and $B_u^0, B_u^1, B_u^2, \dots, B_u^{\log n}$. Set I_u^i contains u's last neighborhood sample when its degree was in $[2^i, 2^{i+1})$ and set B_u^i contains all nodes v such that $u \in I_v^i$.

Each time a node u participates in an "interesting event" or receives a Type_2 notification it stores a random sample of size $O(\log n)$ of its neighborhood in $I_u^{l_u}$. Further, u enters the sets $B_v^{l_u}$ for all nodes $v \in I_u^{l_u}$ of its sample. Those nodes are responsible to notify u when they get deleted. Thus, upon u's deletion all nodes in $\bigcup_i B_u^i$ are notified.

This notification strategy has two key properties, that are: (1) u gets notified and participates in an "interesting event" when a constant fraction of its neighborhood gets deleted; and (2) w.h.p. u gets notified when its 2-hop neighborhood changes substantially. Interestingly, the notification strategy does this while maintaining the expected complexity bounded by $O(\text{polylog}\, n)$ each time the *Notify* procedure is called. Algorithm 2 contains the pseudocode of the notification procedure

3.4. Our Dynamic Algorithm Pseudocode

Our Algorithm 3 contains 4 procedures: the Notify(v_t, ϵ) procedure which we described in Section 3.3, the Clean(u, ϵ, t) procedure, the Anchor(u, ϵ, t) procedure and the Connect(u, ϵ, t) procedure. Before describing the last three we introduce some auxiliary notation.

We denote by $\mathcal{I}_t$ the set of nodes that participated in an "interesting event" at round t. We also denote by Φ a dynamically changing (across the execution of our algorithm) subset of the nodes which we call anchor set. We avoid the

[3]While we are able to circumvent the degree computation in our algorithm, we note that this is an interesting open problem.

Algorithm 2 Notify(u, ϵ)

if u arrived **then**
  $I_u^{l_u} \leftarrow$ sample $10^{10} \log n/\epsilon$ random neighbors
  $\forall v \in I_u^{l_u}$: $B_v^{l_u} \leftarrow B_v^{l_u} \cup \{u\}$
  $\forall v \in I_u^{l_u}$: u sends v a Type_0 notification
else if u was deleted **then**
  $\forall v \in \bigcup_i B_u^i$: u sends v a Type_0 notification
end if
for $i = 0, 1, 2$ **do**
  for all w that received a Type_i notification **do**
    $\forall v \in I_w^{l_w}$: $B_v^{l_w} \leftarrow B_v^{l_w} \setminus \{w\}$
    $I_w^{l_w} \leftarrow$ sample $10^{10} \log n/\epsilon$ random neighbors
    $\forall v \in I_w^{l_w}$: $B_v^{l_w} \leftarrow B_v^{l_w} \cup \{w\}$
    if $i \in \{0, 1\}$ **then**
      $\forall v \in I_w^{l_w}$: w sends v a Type_{i+1} notification
    end if
  end for
end for

subscript t in the set Φ notation as it is always clear for the context at what time we are referring to. When a node u is deleted, then, we also apply $\Phi \leftarrow \Phi \setminus \{u\}$. We maintain a sparse graph $\tilde{G}_t$ with the same node set of G_t and with edge set $\tilde{E}_t \subseteq E_t$. We start our algorithm with $\tilde{G}_0$ being an empty graph. The backbone of our sparse solution $\tilde{G}_t$ is the anchor set nodes Φ. Indeed, we have that $\forall e = (u, v) \in \tilde{E}_t$ either u or v are in Φ. Moreover, for any node u we denote by Φ_u the subset of the nodes in Φ connected to u in our sparse solution $\tilde{G}_t$.

Algorithm 3 Dynamic Agreement (DA)

on arrival/deletion of v_t **do**
  Notify(v_t, ϵ)
  for all $u \in \mathcal{I}_t$ **do**
    Clean(u, ϵ, t)
    Anchor(u, ϵ, t)
    Connect(u, ϵ, t)
  end for
end on

After the arrival or deletion of node v_t and the propagation of notifications through the Notify(v_t, ϵ) procedure, for all nodes u in an important event we do the following.

First we call the Clean(u, ϵ, t) procedure which is responsible to delete edges between nodes which are not in ϵ-agreement anymore and delete from the anchor set nodes that lost too many edges in our sparse solution. These operations enable our algorithm to refine clusters which became too sparse or refine cluster assignment for nodes that are not anymore in ϵ-agreement.

Then we call the Anchor(u, ϵ, t) where if u is heavy then with probability $\min\{1, \frac{10^7 \log n}{\epsilon |N_{G_t}(u)|}\}$ we add this node to Φ. If u is added in Φ then we calculate agreements with all of its neighbors and whenever u is in agreement with a neighbor v we add edge (u, v) to our sparse solution $\tilde{G}_t$. As mentioned previously the anchor nodes are representative nodes of clusters. Thus, the Anchor(u, ϵ, t) procedure allows us to update the set of anchor nodes so that they behave approximately like a uniform sample of each cluster.

Finally, we initiate the *Connect* procedure. The purpose of this step is to add some redundant information so that the clustering is stable and also ensure that nodes that are inserted lately are guaranteed to be connected to some anchor node of their cluster.

Algorithm 4 Connect(u, ϵ, t)

Let J_u be a random sample of size $^{10^5 \log n}/_{\epsilon}$ from $N_{G_t}(u)$.
for $w \in J_u$ **do**
  for $r \in \Phi_w$ **do**
    If r is heavy and in agreement with u then add the edge (r, w) to $\tilde{G}_t$
  end for
end for

Algorithm 5 Anchor(u, ϵ, t)

$X_u \sim$ Bernoulli($\min\{^{10^7 \log n}/_{\epsilon |N_{G_t}(u)|}, 1\}$)
if $u \in \Phi$ **then**
  Delete all edges (u, v) where $v \notin \Phi$ from $\tilde{G}_t$
end if
if u is heavy in G_t and $X_u = 1$ **then**
  For every neighbor v, if u and v are in agreement add edge (u, v) in our sparse solution $\tilde{G}_t$.
end if
if $X_u = 1$ **then**
  Add u to Φ at the end of iteration t
else if $u \in \Phi$ and $X_u = 0$ **then**
  Delete u from Φ at the end of iteration t
end if

Algorithm 6 Clean(u, ϵ, t)

for $w \in \Phi_u$ **do**
  if w not in agreement with u or w is not heavy **then**
    Delete edge (w, u) from $\tilde{G}_t$
  end if
  if w lost more than an ϵ fraction of its edges in $\tilde{G}_t$ from when it entered Φ **then**
    Delete w from Φ and all edges between w and its neighbors in $\tilde{G}_t$
  end if
end for

4. Overview of our Analysis

In this section we give a high level description of our proof strategy, the full proof is available in the Appendix. We first present the high-level ideas on how to prove Theorem 6 and then we bound the running time.

Theorem 6. *For each time t the Dynamic Agreement algorithm outputs an $O(1)-$approximate clustering with probability at least $1 - 5/n$.*

4.1. Overview of the Correctness Proof

To prove the correctness of our algorithm we show that at any time t: 1) for every cluster C that is identified by the offline Agreement algorithm (which is known to be constant factor approximate) our algorithm w.h.p. forms a cluster that is a superset of C, and 2) every cluster C' detected by our algorithm is a dense cluster w.h.p., meaning that $\forall u \in C'$, $|N_{G_t}(u) \cap C'| \geqslant (1 - c' \cdot \epsilon)|C'|$ for small enough ϵ, and a constant $c' \ll 1/\epsilon$. Combining the above two facts and union bounding over all time t, we get that all clusters detected by the offline agreement algorithm are found, and all detected clusters are very dense. So using the fact that the offline agreement algorithm is a constant approximation algorithm we can show that also our algorithm is. The formal proof of the two facts requires the introduction of several concepts and probabilistic events, and is deferred to Appendix A and Appendix B. Here we give a high-level overview of our proof strategy. For the remainder of this section, we call a cluster computed by the offline agreement algorithm a good cluster.

All good clusters are detected. As discussed in Section 3 the key idea of the algorithm is to design a sampling strategy and a notify procedure to keep track of the good clusters efficiently. The key idea is to not identify a good cluster only at the time that is formed, but to design a strategy to track the most important events that affect any node in the graph during the execution of the algorithm and to maintain the clustering structure through connections to the anchor set nodes.

Let a cluster C be a good cluster at time t. In our analysis, we analyze the last $\frac{\epsilon}{c}|C|$ interesting events involving nodes of cluster C. Let $L \subset C$ be the set of nodes involved in these last interesting events, we further subdivide L into L_1 containing the half of L that participated earlier in interesting events, and L_2 containing those that participated later. We also denote $R = C \setminus L$. Intuitively we show that every node connects to another node in the anchor set either in the first or second part of our analysis.

Let t_u be the last time in which a node $u \in L$ participates in an "interesting event". Our notification procedure ensures that w.h.p. u's neighborhood does not change significantly after its last participation in an "interesting event", i.e., $N_{G_{t_u}}(u) \simeq N_{G_{t'}}(u)$, $\forall t' \in [t_u, t]$ (see Lemma 14 and the preceding discussion). At the same time, we know that at time t, u belongs to the good cluster C and by the properties of the agreement decomposition (see Appendix G) we have that $N_{G_t}(u) \simeq C$. Combining the last two observations we can conclude that $N_{G_{t_u}}(u) \simeq C$. This line of arguments can be extended to all nodes $v \in N_{G_{t_u}}(u) \cap R$ which by the definition of R do not participate in an "interesting event" after time t_u. Thus, $\forall v \in N_{G_{t_u}}(u) \cap R$, it holds that: $N_{G_{t_u}}(v) \simeq C$. In addition, R contains almost all nodes of C, thus $N_{G_{t_u}}(u) \cap R \simeq C$. Combining all these observations we can conclude that u is in agreement with almost all of its neighbors at time t_u, and therefore it is heavy. In addition, if u enters in the anchor set, it remains there until time t. This is proved in Theorem 17 of Appendix A.

We are ready to prove that our algorithm finds a cluster $C' \supseteq C$ at time t for every good cluster C detected by the offline agreement algorithm. As described in Section 3, the clusters that our algorithms form are determined by the connected components in our sparse solution graph $\tilde{G}_t$. To this end we argue that each node of C is connected to a node in C that is in the anchor set in $\tilde{G}_t$. We show how this is true for each of the three sets R, L_1, L_2. For R we recall that each node in $L_1 \subset L$ is heavy and enters the anchor set with probability $O(\frac{\log n}{\epsilon |N_{G_{t_u}}(u)|})$ at time t_u.

Given that $|L_1| = \frac{c}{2\epsilon}|C|$ for $c \ll \epsilon$, we can show that each $v \in R$ has a neighbor $v' \in L_1$ that enters the anchor set w.h.p., and v' connects to v in $\tilde{G}_t$ during the $Anchor(v', \epsilon, t_{v'})$ procedure. This is formally proved in Lemma 20 of Appendix A. Similarly, each node in L_1 has a neighbor in L_2 that enters the anchor set w.h.p.. Finally, we prove that each node v in L_2 is connected to a node v' in the anchor set in L_1 w.h.p.. In fact, since most pairs of nodes in $L_1 \cup L_2$ are in agreement w.h.p. there are many common neighbors $w \in R$ such that w is in agreement with both v' and v, which implies that the $Connect(v, \epsilon, t_v)$ will connect v to v'. See Lemma 22 of Appendix A for the formal proof. Hence, all nodes in C get connected to a node in the anchor set (which as we claimed above remains in the anchor set until time t). To conclude the argument we also note that the nodes in the anchor set are connected to each other because they share most of their neighbors. This is proved in Theorem 23 of Appendix A.

All found clusters are good. To prove that all clusters identified by our algorithm are good clusters, we follow a proof strategy similarly to (Cohen-Addad et al., 2021). Roughly speaking, we show that each connected component C' of $\tilde{G}_t$ has diameter 4, which follows by observing that all nodes in the anchor set are within distance 2 from each other, and that each other node is connected to a node in the anchor set. Then, due to the transitivity of the agreement property, it follows that all nodes in the connected component C' are in agreement with each other. This last claim, then further implies that for each $v \in C'$ it holds that $|N_{G_t} \cap C'| \geqslant (1 - c\epsilon)|C'|$ which makes C' a good cluster. See Appendix B for the formal proof.

4.2. Overview of Running Time Bound

We observe that our algorithm is correct even when deletions occur adversarially, but this unfortunately does not hold for the analysis of its running time. Notice that for insertions the running time is bounded by the $O(\text{polylog}\, n)$ forward notifications sent as a result of the insertion of a new node. On the other hand, the deletion of a node u may cause all nodes in $\bigcup_i B_u^i$ to receive a notification, and this in turn causes those nodes to send forward notifications. The issue arises in that there is no bound in the size of $|\bigcup_i B_u^i|$ when deletions occur adversarially. In fact, we can construct an instance where $|\bigcup_i B_u^i| \in \widetilde{\Theta}(n)$ for $\widetilde{O}(n)$ many deletions. Thankfully, in the case where the deletions appear in a random order, this cannot happen as we can nicely bound the expectation of $|\bigcup_i B_u^i|$ to be $O(\text{polylog}\, n)$ for a node u chosen uniformly at random. We devote Appendix C in the formal proof of the running time analysis.

5. Experimental Evaluation

We conduct two sets of experiments. We first evaluate the performance of our algorithm to the same set of real-world graphs that were used in (Lattanzi et al., 2021). Then, we investigate how the running time of our algorithm scales with the size of the input.

5.1. Baselines and Datasets

We compare our algorithm to SINGLETONS where its output always consists of only singleton clusters and PIVOT-DYNAMIC (Behnezhad et al., 2023a) which is a dynamic variation of the Pivot algorithm (Ailon et al., 2008) for edge streams with $O(1)$ update time. While PIVOT-DYNAMIC guarantees a constant factor approximation, SINGLETONS does not have any theoretical guarantees. Nevertheless, it has been observed in (Lattanzi et al., 2021) that sparse graphs tend to not have a good correlation clustering structure, and often the clustering that consists of only singleton clusters is a competitive solution.

We consider five real-world datasets. For the first set of experiments, we use four graphs from SNAP (Leskovec & Krevl, 2014) that include a Social network (musae-facebook), an email network (email-Enron), a collaboration network (ca-AstroPh), and a paper citation network (cit-HepTh).

In the second set of experiments where we investigate the runtime of our algorithm with respect to the size of the input we use Drift (Vergara et al., 2012; Rodriguez-Lujan et al., 2014) from the UCI Machine Learning Repository (Dua & Graff, 2017). The dataset contains $13,910$ points embedded in a space of 129 dimensions. Each point corresponds to a node in our graph and we add a positive connection between two nodes if the euclidean distance of the corresponding points is below a certain threshold. The lower we set that threshold the denser the graph becomes.

All graphs are formatted so as to be undirected and without parallel edges. In addition, we create the node streams of node additions and deletions as follows: we first create a random arrival sequence for all the nodes. Subsequently in between any two additions, with probability p, we select at random a node of the current graph and delete it. If all nodes have already arrived then at each time we randomly select one of those and delete it.

5.2. Setup and Experimental Details

Our code is written in Python 3.11.5 and is available at `https://github.com/andreasr27/DCC`. We set the deletion probability in between any two node arrivals to be 0.2. The agreement parameter is set to $\epsilon = 0.2$, as this setting exhibited the best behavior in (Cohen-Addad et al., 2021) and (Lattanzi et al., 2021).

In addition, we set the number of samples in our procedures to a small constant. More precisely, all our subroutines use a random sample of size 2 and the probability of a node joining the anchor set is set to $20/d_u$ where d_u denotes its degree in the current graph. Here we deviate from the numbers we use in theory as we observe that, in practice, for sparse graphs only running time is affected. We note that in the runtime calculation we do not include reading the input and calculating the quality of our clustering. We do this in an effort to best approximate the Database model in Definition 1 while reading the input we implement suitable data structures[4] where the graph is stored in the form of adjacency lists which permit node additions, deletions and getting a random sample in $O(1)$ expected time.

Solution quality. In the first set of experiments we run all three algorithms and plot their performance relative to SINGLETONS, that is, we plot the cost of solution produced by our algorithm or PIVOT-DYNAMIC divided by the cost of the solution of SINGLETONS. In all datasets, DYNAMIC-AGREEMENT consistently outperforms both PIVOT-DYNAMIC and SINGLETONS. For example, in Fig. 1 we plot the correlation clustering objective every 10 nodes additions/deletions in the node stream.

[4]e.g.: `https://leetcode.com/problems/insert-delete-getrandom-o1/description/`

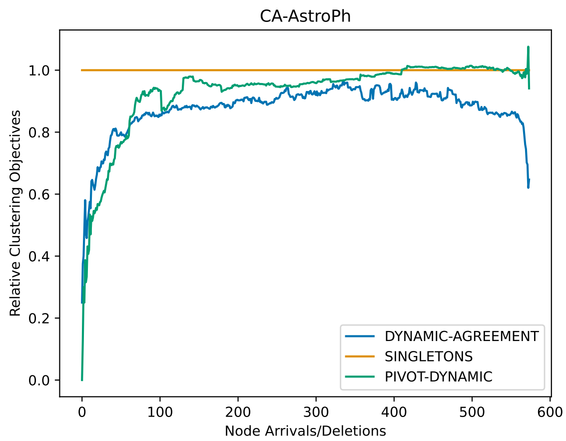

Figure 1. Correlation clustering objective relative to singletons

	DA	PD
MUSAE-FACEBOOK	2.27	0.1
EMAIL-ENRON	2.79	0.12
CIT-HEPTH	3.84	0.21
CA-ASTROPH	2.96	0.11

Table 1. RUNTIMES OF THE ALGORITHMS DYNAMIC-AGREEMENT (DA) AND PIVOT-DYNAMIC (PD).

We observe that after a constant fraction of all nodes have arrived, the clustering objective of our algorithm relative to SINGLETONS remains stable both for node additions and deletions. On the contrary the performance of PIVOT-DYNAMIC fluctuates and tends to increase especially in the last part of the sequence when all nodes have already arrived and the node stream contains predominantly deletions. A similar behaviour is observed in the other three datasets which are deferred to Appendix H. In Appendix H we also present a table which gives an estimate of the optimum offline solution based on the classical PIVOT algorithm of Ailon et al. (2008). In Section 5.2 we see that our algorithm is slower to the PIVOT-DYNAMIC implementation. This is something that we expect since PIVOT-DYNAMIC is extremely efficient for sparse graphs.

Running time. As mentioned previously, we constructed a graph using the Drift dataset by associating points with nodes and adding positive edges between nodes if the Euclidean distance of the corresponding points is less than a threshold. Different thresholds lead to the creation of distinct graphs. Now we relate the density of the graph (average node degree) with the runtimes and clustering quality of both DYNAMIC-AGREEMENT and PIVOT-DYNAMIC. Section 5.2 shows the average relative clustering quality of each algorithm over the entire node stream and their running times.

	Relative Objective		Running Time	
Density	DA	PD	DA	PD
253.36	0.696	0.610	12.48	0.28
114.87	0.605	0.551	14.54	0.16
69.74	0.522	0.504	15.82	0.11
52.17	0.389	0.416	14.10	0.09
42.25	0.320	0.375	12.85	0.07

Table 2. Summary of our scalability experiment for the algorithms DYNAMIC-AGREEMENT (DA) and PIVOT-DYNAMIC (PD), on the Drift dataset at varying densities.

We observe that both DYNAMIC-AGREEMENT and PIVOT-DYNAMIC outperform SINGLETONS, as expected, since SINGLETONS excels in sparse graphs and offers poor quality solutions in denser graphs. Additionally, the running time of DYNAMIC-AGREEMENT remains stable when the density of the graphs increases confirming that the algorithm's running time is not affected by the graph density. On the other hand, the running time of PIVOT-DYNAMIC increases linearly with the density. This suggests that for very large and dense graphs, where even reading the entire input is prohibitive DYNAMIC-AGREEMENT scales smoothly, further validating the theory.

Experimental summary. We observe that the newly proposed algorithm computes high quality solutions for both sparse and dense graphs. This is in contrast with comparison methods that fail to produce good solutions in at least one of the two settings. We also show that even if the runtime of our algorithm is higher compared to the competitor, its runtime does not increase with the density of the input graph. This is inline with our theoretical results and is achieved via our sampling and notify strategies.

Conclusion and Future Work

We provide the first sublinear time algorithm for correlation clustering in dynamic streams where nodes are added adversarially and deleted randomly. Our algorithm is based on new and carefully defined sampling and notification strategies that can be of independent interest. We also show experimentally that our algorithm provides high quality solution both for dense and sparse graph outperforming previously known algorithms.

A very interesting open question is to extend our result in the general setting where both nodes' addition and deletion is adversarial. One possible way of achieving the later would be to use a different sparse-dense decomposition which is more stable and requires less updates so that it is maintained approximately, e.g., the one proposed by Assadi & Wang (2022).

Reducing the amortize update time in our setting is another very interesting and natural question.

Impact Statement

This paper presents work whose goal is to provide a theoretical foundation to machine learning heuristics often used in practice. We do not feel that any particular societal consequences of our work should be highlighted here.

References

Agrawal, R., Halverson, A., Kenthapadi, K., Mishra, N., and Tsaparas, P. Generating labels from clicks. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, pp. 172–181, 2009.

Ailon, N., Charikar, M., and Newman, A. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM (JACM)*, 55(5), 2008.

Arasu, A., Ré, C., and Suciu, D. Large-scale deduplication with constraints using dedupalog. In *Proceedings of the 25th International Conference on Data Engineering*, pp. 952–963, 2009.

Assadi, S. and Wang, C. Sublinear time and space algorithms for correlation clustering via sparse-dense decompositions. In *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS)*, pp. 10:1–10:20, 2022.

Bansal, N., Blum, A., and Chawla, S. Correlation clustering. *Machine learning*, 56(1):89–113, 2004.

Bateni, M., Esfandiari, H., Fichtenberger, H., Henzinger, M., Jayaram, R., Mirrokni, V., and Wiese, A. Optimal fully dynamic k-center clustering for adaptive and oblivious adversaries. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2677–2727. SIAM, 2023.

Behnezhad, S., Derakhshan, M., Hajiaghayi, M., Stein, C., and Sudan, M. Fully dynamic maximal independent set with polylogarithmic update time. In Zuckerman, D. (ed.), *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pp. 382–405. IEEE Computer Society, 2019. doi: 10.1109/FOCS.2019.00032. URL https://doi.org/10.1109/FOCS.2019.00032.

Behnezhad, S., Charikar, M., Ma, W., and Tan, L. Almost 3-approximate correlation clustering in constant rounds. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pp. 720–731. IEEE, 2022. doi: 10.1109/FOCS54457.2022.00074. URL https://doi.org/10.1109/FOCS54457.2022.00074.

Behnezhad, S., Charikar, M., Ma, W., and Tan, L. Single-pass streaming algorithms for correlation clustering. In Bansal, N. and Nagarajan, V. (eds.), *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pp. 819–849. SIAM, 2023a. doi: 10.1137/1.9781611977554.ch33. URL https://doi.org/10.1137/1.9781611977554.ch33.

Behnezhad, S., Charikar, M., Ma, W., and Tan, L. Single-pass streaming algorithms for correlation clustering. In *Proceedings of the 34th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 819–849, 2023b.

Bonchi, F., Gionis, A., and Ukkonen, A. Overlapping correlation clustering. *Knowledge and information systems*, 35(1):1–32, 2013.

Cao, N., Cohen-Addad, V., Lee, E., Li, S., Newman, A., and Vogl, L. Understanding the cluster lp for correlation clustering. In *STOC'24*, 2024.

Chakrabarti, D., Kumar, R., and Punera, K. A graph-theoretic approach to webpage segmentation. In *Proceedings of the 17th International World Wide Web Conference (WWW)*, pp. 377–386, 2008.

Chawla, S., Makarychev, K., Schramm, T., and Yaroslavtsev, G. Near optimal lp rounding algorithm for correlation clustering on complete and complete k-partite graphs. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 219–228, 2015.

Chen, Y., Sanghavi, S., and Xu, H. Clustering sparse graphs. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 2204–2212, 2012.

Cohen-Addad, V., Hjuler, N., Parotsidis, N., Saulpic, D., and Schwiegelshohn, C. Fully dynamic consistent facility location. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Cohen-Addad, V., Lattanzi, S., Mitrovic, S., Norouzi-Fard, A., Parotsidis, N., and Tarnawski, J. Correlation clustering in constant many parallel rounds. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pp. 2069–2078, 2021.

Cohen-Addad, V., Lattanzi, S., Maggiori, A., and Parotsidis, N. Online and consistent correlation clustering. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 4157–4179, 2022a.

Cohen-Addad, V., Lee, E., and Newman, A. Correlation clustering with sherali-adams. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pp. 651–661. IEEE, 2022b. doi: 10.1109/FOCS54457.2022.00068. URL https://doi.org/10.1109/FOCS54457.2022.00068.

Cohen-Addad, V., Lee, E., Li, S., and Newman, A. Handling correlated rounding error via preclustering: A 1.73-approximation for correlation clustering. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pp. 1082–1104. IEEE, 2023. doi: 10.1109/FOCS57990.2023.00065. URL https://doi.org/10.1109/FOCS57990.2023.00065.

Cohen-Addad, V., Lolck, D. R., Pilipczuk, M., Thorup, M., Yan, S., and Zhang, H. Combinatorial correlation clustering. In *STOC'24*, 2024.

Demaine, E. D., Emanuel, D., Fiat, A., and Immorlica, N. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2-3):172–187, 2006.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Epasto, A., Lattanzi, S., and Sozio, M. Efficient densest subgraph computation in evolving graphs. In *Proceedings of the 24th international conference on world wide web*, pp. 300–310, 2015.

Fichtenberger, H., Lattanzi, S., Norouzi-Fard, A., and Svensson, O. Consistent k-clustering for general metrics. In *Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2660–2678. SIAM, 2021.

Guo, X., Kulkarni, J., Li, S., and Xian, J. Consistent k-median: Simpler, better and robust. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, pp. 1135–1143, 2021.

Jaghargh, M. R. K., Krause, A., Lattanzi, S., and Vassilvtiskii, S. Consistent online optimization: Convex and submodular. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2241–2250, 2019.

Kalashnikov, D. V., Chen, Z., Mehrotra, S., and Nuray-Turan, R. Web people search via connection analysis. *IEEE Transactions on Knowledge and Data Engineering*, 20(11):1550–1565, 2008.

Lattanzi, S. and Vassilvitskii, S. Consistent k-clustering. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1975–1984, 2017.

Lattanzi, S., Moseley, B., Vassilvitskii, S., Wang, Y., and Zhou, R. Robust online correlation clustering. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Leskovec, J. and Krevl, A. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, 2014.

Rodriguez-Lujan, I., Fonollosa, J., Vergara, A., Homer, M., and Huerta, R. On the calibration of sensor arrays for pattern recognition using the minimal number of experiments. *Chemometrics and Intelligent Laboratory Systems*, 130:123–134, 2014.

Vergara, A., Vembu, S., Ayhan, T., Ryan, M. A., Homer, M. L., and Huerta, R. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166:320–329, 2012.

A. Finding Dense Clusters

In this section we prove that our algorithm correctly finds the non-singleton clusters of G_t that are also found by the AGREEMENTALGORITHM(G_t) when the agreement and heaviness parameters are set to a small enough value. That is, let C be a non-singleton cluster that is found by AGREEMENTALGORITHM($G_{t_{current}}$) at time $t_{current}$ when the agreement and heaviness parameters are set to $\epsilon/10^{14}$. It can be proven that cluster C is extremely dense, i.e., it forms almost a clique, with very few outgoing edges. In Theorem 23 we prove that at time $t_{current}$ our algorithm outputs a cluster C' that contains all nodes of C, i.e., $C' \supseteq C$. While the latter may not seem surprising per se (note that a trivial algorithm which clusters all nodes in the same partition also achieves that property), our approach consists in a delicate argument where we prove that for all non-trivial clusters C found by AGREEMENTALGORITHM($G_{t_{current}}$) our algorithm always constructs a cluster $C' \supseteq C$ and at the same time in Appendix B all clusters C' constructed by our algorithm (which may not be constructed by AGREEMENTALGORITHM($G_{t_{current}}$)) are very dense. In order to simplify the notation we use t_c instead of $t_{current}$ in our formulas.

The first challenge in the current section is to prove that the notify procedure correctly samples enough nodes of C, "close enough" to time $t_{current}$, so that many of these nodes enter the anchor set and help us reconstruct C. Note that the latter ensures that enough nodes from a specific cluster enter the anchor set, but does not ensure that these clusters are correctly found. Indeed, the last arriving node may not enter the anchor set and at the same time the "interesting events" that its arrival generates may not induce any node to join the anchor set. However, to get a constant factor approximation, we still need to ensure that the last node of C is clustered correctly with the rest of the cluster. In order to ensure the latter, our algorithm uses the CONNECT(v, ϵ) procedure. CONNECT(v, ϵ) constructs a sample of v's neighborhood and for every node w in that sample, it checks whether v is in ϵ-agreement with any node $r \in \Phi_w$, i.e., any node that is in the anchor set of and if so v is connected to r.

In the current section we make great use of the relation between C and $N_{G_{t_c}}(u)$ for a node $u \in C$, indeed since C is a dense cluster found by AGREEMENTALGORITHM($G_{t_{current}}$) at time $t_{current}$, informally we have that $N_{G_{t_c}}(u) \approx C$ (see the properties of the Agreement decomposition in Appendix G).

Consider the last $\lfloor \epsilon/10^4 |C| \rfloor$ nodes of C that participate in an "interesting event" and denote those nodes by u_i for $i = 1, 2, \ldots, \lfloor \epsilon/10^4 |C| \rfloor$. For each u_i denote by t_i the last time that this node participated in an "interesting event", so that $t_1 \leqslant t_2 \leqslant \ldots \leqslant t_{\lfloor \epsilon/10^4 |C| \rfloor}$. If two nodes u_i, u_j participated in an "interesting event" after the same arrival or deletion in the node streams then we order with respect to the type of notification received, that is, $t_i \leqslant t_j$ if u_i just arrived or received a notification of a lower type than u_j. Ties are broken arbitrarily but consistently. Note that both nodes u_i and times t_i for $i = 1, 2, \ldots, \lfloor \epsilon/10^4 |C| \rfloor$ are random variables which depend on the internal randomness of the Notify procedure.

We now introduce some auxiliary notation:

1. $L = \{u_1, u_2, \ldots, u_{\lfloor \epsilon/10^4 |C| \rfloor}\}$ and $R = C \setminus L$.
2. $L_1 = \{u_1, u_2, \ldots, u_{\lfloor \epsilon/2 \cdot 10^4 |C| \rfloor}\}$.
3. $L_2 = \{u_{\lfloor \epsilon/2 \cdot 10^4 |C| \rfloor + 1}, u_{\lfloor \epsilon/2 \cdot 10^4 |C| \rfloor + 2}, \ldots, u_{\lfloor \epsilon/10^4 |C| \rfloor}\}$.
4. $L^i = \{u_1, u_2, \ldots, u_i\}$.
5. We denote as t' the earliest time that all nodes of R have arrived.

Again, note that since the u_i's and t_i's are random variables then also R, L, L_1, L_2, L^i and t' are random variables.

We start by some simple observations where we argue that after time t' all nodes of C that have already arrived form a dense subgraph.

Observation 7. *$\forall u \in C$ we have:*

$$(1 - \epsilon/10^{13})|C| \leqslant |N_{G_{t_c}}(u)| \leqslant (1 + \epsilon/10^{13})|C|$$

Proof. From Property 6 and Property 10 in Appendix G the result is immediate. □

Observation 8. *$\forall u \in C$ we have:*

$$|N_{G_{t_c}}(u) \cap C| \geqslant (1 - \epsilon/10^{13})|C|$$

Proof. Follows immediately from Property 4. □

Observation 9. *For each $u \in C$ that arrived at time $t \in [t', t_c]$ we have:*

$$|N_{G_t}(u)| \geqslant |N_{G_t}(u) \cap C| \geqslant (1 - 2\epsilon/10^4)|C|$$

Proof. The left-hand side inequality is obvious. For the right inequality note that at time t' all nodes in R have arrived. Thus,

$$\begin{aligned}|N_{G_t}(u) \cap C| &\geqslant |N_{G_{t_c}}(u) \cap C| - |L| \geqslant (1 - \epsilon/10^{13})|C| - \epsilon/10^4|C| \\ &\geqslant (1 - 2\epsilon/10^4)|C|\end{aligned}$$

Where in the second inequality we use Observation 8 and $|L| = \lfloor \epsilon/10^4|C| \rfloor$. □

Observation 10. *For each $u \in C$ that has already arrived before time $t \in [t', t_c]$ we have:*

$$|C \setminus N_{G_t}(u)| \leqslant 2\epsilon/10^4|C| \leqslant \epsilon/10^3|N_{G_t}(u)|$$

Proof. The left inequality is immediate from Observation 9 and the second one holds for ϵ small enough again using Observation 9. □

Observation 10 informally states that the neighborhood of any node $v \in C$ between times t' and t_c contains C almost in its entirety. Let $t \in [t', t_c]$ be a time when node v has already arrived, then we have that: $N_{G_t}(v) \supseteq C'$ and $C' \simeq C$. Ideally, we would like all nodes $v \in C$ to have almost their final neighborhood between times t' and t_c so that the Anchor procedure correctly reconstructs the cluster C. That is, we would like to also have $N_{G_t}(v) \subseteq C''$ and $C'' \simeq C$. While the latter may be not true in general, we prove that with high probability at every time t_i it is true for u_i and a large part of its neighborhood.

We introduce auxiliary notation to formalize these claims.

Definition 11. *For a node $v \in C$ and time t we define the following events:*

$$\begin{aligned}M_v^t &= \{|N_{G_t}(v) \setminus N_{G_{t_c}}(v)| < \epsilon/8\cdot10^4 |N_{G_t}(v)|\} \\ A_v^t &= \{\{v \in \mathcal{I}_t\} \vee \{v \text{ received a } Type_2 \text{ notification at time } t\}\}\end{aligned}$$

M_v^t is true if at least a $(1 - \epsilon/8\cdot10^4)$ fraction of v's neighborhood at time t does not get deleted until time t_c. Also, note that if A_v^t is true then v samples with replacement $10^{10} \log n/\epsilon$ neighbors at time t. By $\overline{M_v^t}$ and $\overline{A_v^t}$ we denote the complementary events of M_v^t and A_v^t respectively.

The rest of the section is devoted in arguing that with high probability at each time t_i, both u_i and nearly all its neighbors possess almost their "final" neighborhood. The crux of our analysis is based on the following two observations (stated informally for the moment). For all $v \in R \cup L_i$: (1) If $M_v^{t_i}$ is true, then $N_{G_{t_i}}(v) \simeq C$; and (2) The event $\overline{M_v^{t_i}} \wedge A_v^{t_i}$ happens with low probability.

In the following lemma we prove structural properties of the neighborhood $N_{G_{t_i}}(v)$ of a node when v when $M_v^{t_i}$ is true.

Lemma 12. *$\forall v \in R \cup L^i$ we have that if $M_v^{t_i}$ is true then:*

$$|N_{G_{t_i}}(v)| < (1 + \epsilon/2\cdot10^4)|C| \tag{1}$$

$$|N_{G_{t_i}}(v) \setminus C| < \epsilon/10^4|C| < \epsilon/10^3|N_{G_{t_i}}(v)| \tag{2}$$

Proof. 1. For the first statement we prove the contrapositive, i.e., we argue that $|N_{G_{t_i}}(v)| \geqslant (1 + \epsilon/2\cdot10^4)|C|$ implies $\overline{M_v^{t_i}}$. We distinguish between the cases:

- $|N_{G_{t_i}}(v)| \geqslant \frac{|N_{G_{t_c}}(v)|}{(1-\epsilon/(8\cdot10^4))}$, then

$$\begin{aligned}|N_{G_{t_i}}(v) \setminus N_{G_{t_c}}(v)| &\geqslant |N_{G_{t_i}}(v)| - |N_{G_{t_c}}(v)| \\ &\geqslant \epsilon/(8\cdot10^4)|N_{G_{t_i}}(v)|\end{aligned}$$

- $|N_{G_{t_i}}(v)| < \frac{|N_{G_{t_c}}(v)|}{(1-\epsilon/(8\cdot 10^4))}$, in that case from Observation 7 we have the bound $|N_{G_{t_c}}(v)| \leqslant (1+\epsilon/10^{13})|C|$

$$\begin{aligned}
&|N_{G_{t_i}}(v) \setminus N_{G_{t_c}}(v)| \\
&\geqslant |N_{G_{t_i}}(v)| - |N_{G_{t_c}}(v)| \\
&\geqslant (1+\epsilon/(2\cdot 10^4))|C| - |N_{G_{t_c}}(v)| \\
&\geqslant \left(\frac{1+\epsilon/(2\cdot 10^4)}{1+\epsilon/10^{13}} - 1\right)|N_{G_{t_c}}(v)| \\
&\geqslant \left(\frac{1+\epsilon/(2\cdot 10^4)}{1+\epsilon/10^{13}} - 1\right)\left(1-\epsilon/(8\cdot 10^4)\right)|N_{G_{t_i}}(v)| \\
&\geqslant \left(\epsilon/(2\cdot 10^4) - \epsilon/10^{13}\right)\left(\frac{1-\epsilon/(8\cdot 10^4)}{1+\epsilon/10^{13}}\right)|N_{G_{t_i}}(v)| \\
&\geqslant \epsilon/(8\cdot 10^4)|N_{G_{t_i}}(v)|
\end{aligned}$$

Where in the second inequality we used our assumption that $|N_{G_{t_i}}(v)| \geqslant (1+\epsilon/2\cdot 10^4)|C|$, the third one from Observation 7, the fourth one from the fact that we are in the second case and the last one holds for ϵ small enough.

2. From the first part of the current lemma we have that $|N_{G_{t_i}}(v)| < (1+\epsilon/2\cdot 10^4)|C|$ and from Observation 9 we have that $|N_{G_{t_i}}(v) \cap C| \geqslant (1-2\epsilon/10^4)|C|$, consequently:

$$\begin{aligned}
|N_{G_{t_i}}(v) \setminus C| &= |N_{G_{t_i}}(v)| - |N_{G_{t_i}}(v) \cap C| \\
&< (1+\epsilon/(2\cdot 10^4))|C| - (1-2\epsilon/10^4)|C| \\
&< 3\epsilon/10^4|C| \\
&< 3\epsilon/10^4\frac{|N_{G_{t_i}}(v)|}{(1-2\epsilon/10^4)} \\
&< \epsilon/10^3|N_{G_{t_i}}(v)|
\end{aligned}$$

Where the second inequality holds for ϵ small enough and proves the first inequality of (2) in the current lemma, the third inequality uses again Observation 9 and the last inequality holds for ϵ small enough.

□

By Combining Lemma 12 and Observation 10 we can argue that $\forall v \in R \cup L^i$ if $M_v^{t_i}$ is true then $N_{G_{t_i}}(v) \simeq C$. Consequently for two neighboring nodes $v, u \in R \cup L^i$: $M_v^{t_i} \wedge M_u^{t_i}$ implies that $N_{G_{t_i}}(v) \simeq N_{G_{t_i}}(u) \simeq C$ and consequently u and v are in agreement. We formalize the latter in the following lemma.

Lemma 13. *For all neighboring nodes $v, u \in R \cup L^i$: if $M_v^{t_i} \wedge M_u^{t_i}$ is true then v and u are in $\epsilon/10$-agreement.*

Proof. Since both $M_u^{t_i}$ and $M_v^{t_i}$ are true, from Lemma 12 we have that:

$$|N_{G_{t_i}}(u) \setminus C| < \epsilon/10^3|N_{G_{t_i}}(u)|$$

and

$$|N_{G_{t_i}}(v) \setminus C| < \epsilon/10^3|N_{G_{t_i}}(v)|$$

Also from Observation 10 it holds that:

$$|C \setminus N_{G_{t_i}}(u)| \leqslant \epsilon/10^3|N_{G_{t_i}}(u)|$$

and

$$|C \setminus N_{G_{t_i}}(v)| \leqslant \epsilon/10^3|N_{G_{t_i}}(v)|$$

To argue that $|N_{G_{t_i}}(u) \triangle N_{G_{t_i}}(v)| \leqslant \epsilon/10 \max\{|N_{G_{t_i}}(u)|, N_{G_{t_i}}(v)|\}$, we bound both $|N_{G_{t_i}}(u) \setminus N_{G_{t_i}}(v)|$ and $|N_{G_{t_i}}(v) \setminus N_{G_{t_i}}(u)|$ by $\epsilon/10^2 \max\{|N_{G_{t_i}}(u_i)|, N_{G_{t_i}}(w)|\}$. From the latter the lemma follows since $2\epsilon/10^2 < \epsilon/10$ for ϵ small enough. We upper bound the size of $N_{G_{t_i}}(u) \setminus N_{G_{t_i}}(v)$:

$$\begin{aligned} |N_{G_{t_i}}(u) \setminus N_{G_{t_i}}(v)| &\leqslant |N_{G_{t_i}}(u) \setminus C| + |C \setminus N_{G_{t_i}}(v)| \\ &\leqslant \epsilon/10^3 |N_{G_{t_i}}(u)| + \epsilon/10^3 |N_{G_{t_i}}(v)| \\ &\leqslant \epsilon/10^2 \max\{|N_{G_{t_i}}(u)|, N_{G_{t_i}}(v)|\} \end{aligned}$$

and the upper bound of $|N_{G_{t_i}}(v) \setminus N_{G_{t_i}}(u)|$ follows the same line of arguments. □

We know prove that the probability of a node $v \in R \cup L_i$ both sampling its neighborhood at time t_i and having a neighborhood which is very different than C is very low. The following lemma is crucial, as it subsequently help us argue that most of the neighboring nodes of u_i at time t_i have almost their final neighborhoods.

Lemma 14. *$\forall v \in C$ and time t we have that:*

$$\mathbf{Pr}[\overline{M_v^t} \wedge A_v^t \wedge \{v \in R \cup L^i\} \wedge \{t \geqslant t_i\}] < 1/n^{10^4}$$

Proof. Let $d_{t'}$ denote the degree of node v at time t' and $l_{t'} = l(d_{t'})$. In this proof we write $I_{v,t'}$ to denote the neighborhood sample stored in $I_v^{l_t}$ at time t'. Note that $I_{v,t'}$ and $I_{v,t''}$ for $t' < t''$ may be different if for example v received a Type_2 notification between those times and updated its neighborhood sample.

Also, for all $w \in N_{G_t}(v) \setminus N_{G_{t_c}}(v)$ let t_w be the time when w gets deleted from the node stream and let w_j be the j-th element to be deleted in $N_{G_t}(v) \setminus N_{G_{t_c}}(v)$. For $j = 1, 2, \ldots, |N_{G_t}(v) \setminus N_{G_{t_c}}(v)|$, we define event $E_j = \{w_j \notin I_{v,t_w}\}$.

We argue that $\{v \in R \cup L^i\} \wedge \{t \geqslant t_i\}$ implies $\bigcap_{1 \leqslant j \leqslant |N_{G_t}(v) \setminus N_{G_{t_c}}(v)|} E_j$. Indeed, $\overline{E_j}$ implies that $v \in B_w^{l_t}$ at time t_w. Consequently, upon its deletion, w would send a Type_0 notification to v and v would participate in an "interesting event" between times $[t_i + 1, t_c)$. The latter implies that $v \notin R \cup L_i$.

Thus:

$$\begin{aligned} \mathbf{Pr}[\overline{M_v^t} \wedge A_v^t \wedge \{v \in R \cup L^i\} \wedge \{t \geqslant t_i\}] &\leqslant \mathbf{Pr}[\overline{M_v^t} \wedge A_v^t \wedge \bigcap_{1 \leqslant j \leqslant |N_{G_t}(v) \setminus N_{G_{t_c}}(v)|} E_j] \\ &\leqslant \mathbf{Pr}[A_v^t \wedge \bigcap_{1 \leqslant j \leqslant \epsilon/8 \cdot 10^4 d_t} E_j] \\ &= \prod_{1 \leqslant j \leqslant \epsilon/8 \cdot 10^4 d_t} \mathbf{Pr}[A_v^t \wedge E_j \mid \bigcap_{j' < j} \{A_v^t \wedge E_{j'}\}] \end{aligned}$$

where in the second inequality we used that $|N_{G_t}(v) \setminus N_{G_{t_c}}(v)| \geqslant \epsilon/8 \cdot 10^4 |N_{G_t}(v)|$. Note that since:

$$(1 - 1/2 \cdot d_t)^{\left(10^{10} \log n/\epsilon\right) \cdot (\epsilon/8 \cdot 10^4 d_t)} < 1/n^{10^4}$$

It is enough to argue that:

$$\mathbf{Pr}[A_v^t \wedge E_j \mid \bigcap_{j' < j} \{A_v^t \wedge E_{j'}\}] < (1 - 1/2 \cdot d_t)^{10^{10} \log n/\epsilon}$$

Let $\tilde{t}_{w_{j'}}, j' < j$, be the random variable denoting the last time before $t_{w_{j'}}$ that $I_v^{l_t}$ was updated. Since $t < t_{w_j}$:

1. A_v^t implies that:
   (a) $t \leqslant \tilde{t}_{w_j} < t_{w_j}$; and

(b) $w_j \in N_{G_{\tilde{t}_{w_j}}}(v)$

2. By the definition of $\tilde{t}_{w_j}$: $|N_{G_{\tilde{t}_{w_j}}}(v)| < 2 \cdot d_t$

Let r_i be the i-th random sample used to construct $I_v^{l_t}$ at time t_w, i.e., $I_{w_j, t_{w_j}}$. By the principle of deferred decisions it is enough to decide whether r_i is equal to w_j at time t_{w_j}. Note that given $\bigcap_{j'<j} \{A_v^t \wedge E_{j'}\}$, r_i is a uniform at random sample from a set that contains w_j and has at most $2 \cdot d_t$ elements, i.e., set $N_{G_{\tilde{t}_{w_j}}}(v) \setminus \bigcup_{j'<j} w_{j'}$. Thus:

$$\mathbf{Pr}[\{r_i = v\} \mid \bigcap_{j'<j} \{A_v^t \wedge E_{j'}\}] \leqslant (1 - 1/2 \cdot d_t)$$

and the proof is concluded by noting that $I_{w_j, t_{w_j}} = \bigcup_{i \leqslant 10^{10} \log n/\epsilon} r_i$ and r_i's are independent. □

As already mentioned, we prove that for time t_i most of u_i's neighboring nodes have almost their final neighborhood. Towards that goal, for every t_i we define a random set which contains all neighboring nodes of u_i whose neighborhood at time t_i is "very" different from their final one.

Definition 15. *For t_i we define the random set:* $D_{t_i} = \{v : v \in N_{G_{t_i}}(u_i) \cap (R \cup L^i) \wedge \overline{M_v^{t_i}}\}$.

We prove that the size of this random set is small with high probability.

Lemma 16. $\forall t_i$ *we have that:*

$$\mathbf{Pr}[|D_{t_i}| \geqslant \epsilon/10^5 |N_{G_{t_i}}(u_i)|] < 1/n^{10^3}$$

Proof. In this proof when we write I_{u_i} we refer to u_i's neighborhood sample constructed via the connect procedure at that specific time t_i. Note that:

$$\begin{aligned}\mathbf{Pr}[|D_{t_i}| \geqslant \epsilon/10^5 |N_{G_{t_i}}(u_i)|] &= \mathbf{Pr}[\{|D_{t_i}| \geqslant \epsilon/10^5 |N_{G_{t_i}}(u_i)|\} \wedge \{D_{t_i} \cap I_{u_i} = \emptyset\}] \\ &\quad + \mathbf{Pr}[\{|D_{t_i}| \geqslant \epsilon/10^5 |N_{G_{t_i}}(u_i)|\} \wedge \{D_{t_i} \cap I_{u_i} \neq \emptyset\}]\end{aligned}$$

We bound each of the two terms in the right hand side independently.

For the first term we have:

$$\begin{aligned}\mathbf{Pr}[\{|D_{t_i}| \geqslant \epsilon/10^5 |N_{G_{t_i}}(u_i)|\} \wedge \{D_{t_i} \cap I_{u_i} = \emptyset\}] &\leqslant (1 - \epsilon/10^5)^{10^{10} \log n/\epsilon} \\ &\leqslant 1/n^{10^5}\end{aligned}$$

since $N_{G_{t_i}}(u_i) \setminus D_{t_i}$ is at least a $(1 - \epsilon/10^5)$ fraction of $N_{G_{t_i}}(u_i)$.

For the second term we have:

$$\begin{aligned}\mathbf{Pr}[\{|D_{t_i}| \geqslant \epsilon/10^5 |N_{G_{t_i}}(u_i)|\} \wedge \{D_{t_i} \cap I_{u_i} \neq \emptyset\}] &= \sum_{v \in C} \mathbf{Pr}[\{v \in N_{G_{t_i}}(u_i)\} \wedge \{v \in D_{t_i}\} \wedge \{v \in I_{u_i}\}] \\ &\leqslant \sum_{v \in C} \mathbf{Pr}[\{v \in N_{G_{t_i}}(u_i)\} \wedge \{v \in D_{t_i}\} \wedge \{v \in A_v^{t_i}\}] \\ &\leqslant \sum_{v \in C} \mathbf{Pr}[\{v \in D_{t_i}\} \wedge A_v^{t_i}] \\ &\leqslant \sum_{v \in C} \mathbf{Pr}[\{v \in R \cup L^i\} \wedge \{\overline{M_v^{t_i}}\} \wedge A_v^{t_i}] \\ &\leqslant n \cdot (1/n^{10^4}) \\ &\leqslant 1/n^{10^4 - 1}\end{aligned}$$

Where in the first inequality note that $v \in I_{u_i}$ implies $A_v^{t_i}$ (since v receives a notification from u_i), in the second inequality we used that $D_{t_i} \subseteq N_{G_{t_i}}(u_i)$, in the third inequality we used the definition of set D_{t_i} and in the fourth inequality we used Lemma 14.

Combining the two upper bounds concludes the proof. □

The next lemma argues that if D_{t_i} is small and $N_{G_{t_i}}(u_i) \simeq C$ then u_i is heavy at time t_i.

Theorem 17. *If* $\{|D_{t_i}| < \epsilon/10^5 |N_{G_{t_i}}(u_i)|\} \wedge M_{u_i}^{t_i}$ *is true then* u_i *is* $\epsilon/10$*-heavy at time* t_i*. In addition, if* u_i *also enters the anchor set at time* t_i *then it remains in it at least until time* t_c*.*

Proof. By the definition of D_{t_i} we have that $\forall w \in N_{G_{t_i}}(u_i) \cap (R \cup L^i) \setminus D_{t_i}$ the event $M_w^{t_i}$ is true. Thus, since $M_{u_i}^{t_i}$ is also true from Lemma 13 we can conclude that u_i and w are in $\epsilon/10$-agreement.

We now argue that u_i is in $\epsilon/10$ agreement with at least a $1 - \epsilon/10$ fraction of its neighborhood at time t_i. For this, note that:

$$\begin{aligned}
|N_{G_{t_i}}(u_i) \setminus \big(N_{G_{t_i}}(u_i) \cap (R \cup L^i) \setminus D_{t_i}\big)| &\leqslant |N_{G_{t_i}}(u_i) \setminus \big(N_{G_{t_i}}(u_i) \cap (R \cup L^i)\big)| + |D_{t_i}| \\
&= |N_{G_{t_i}}(u_i) \setminus (R \cup L^i)| + |D_{t_i}| \\
&\leqslant |N_{G_{t_i}}(u_i) \setminus C| + |C \setminus (R \cup L^i)| + |D_{t_i}| \\
&\leqslant |N_{G_{t_i}}(u_i) \setminus C| + |C \setminus R| + |D_{t_i}| \\
&= |N_{G_{t_i}}(u_i) \setminus C| + |L| + |D_{t_i}| \\
&\leqslant \epsilon/10^3 |N_{G_{t_i}}(u_i)| + \epsilon/10^4 |C| + |D_{t_i}| \\
&\leqslant \epsilon/10^3 |N_{G_{t_i}}(u_i)| + 5\epsilon/10^4 |N_{G_{t_i}}(u_i)| + |D_{t_i}| \\
&\leqslant \epsilon/10^3 |N_{G_{t_i}}(u_i)| + 5\epsilon/10^4 |N_{G_{t_i}}(u_i)| + \epsilon/10^5 |N_{G_{t_i}}(u_i)| \\
&\leqslant \epsilon/10 |N_{G_{t_i}}(u_i)|
\end{aligned}$$

Where in the sixth inequality we used (2) of Lemma 12 (since $M_{u_i}^{t_i}$ is true), in the seventh inequality we used that $|D_{t_i}| < \epsilon/10^5 |N_{G_{t_i}}(u_i)|$ and the last inequality holds for ϵ small enough.

To prove the second claim of the theorem, we note that if u_i enters the anchor set at time t_i then to exit that set before time t_c at least an ϵ-fraction of its neighborhood at time t_i needs to call the $Clean(\cdot, \epsilon)$ procedure and consequently participate in an "interesting event". That is, for ϵ small enough, at least $\epsilon(1 - \epsilon/10)|N_{G_{t_i}}(u_i)| \geqslant \epsilon/2 |N_{G_{t_i}}(u_i)|$ nodes in $N_{G_{t_i}}(u_i)$ need to participate in an "interesting event". However, such nodes can be at most:

$$\begin{aligned}
|N_{G_{t_i}}(u_i) \setminus C| + |L \setminus (R \cup L^i)| &\leqslant |N_{G_{t_i}}(u_i) \setminus C| + |L| \\
&\leqslant \epsilon/10^3 |N_{G_{t_i}}(u_i)| + \epsilon/10^4 |C| \\
&\leqslant \epsilon/10^3 |N_{G_{t_i}}(u_i)| + \epsilon/10^3 |N_{G_{t_i}}(u_i)| \\
&\leqslant 2 \cdot \epsilon/10^3 |N_{G_{t_i}}(u_i)|
\end{aligned}$$

which is smaller than $\epsilon/2 |N_{G_{t_i}}(u_i)|$. In the second inequality we used the cardinality of L and in the third inequality we used (2) of Lemma 12. □

Note that $A_{u_i}^{t_i}$ is always true from the definition of u_i and t_i. Combining Lemma 14 and Lemma 16 we deduce that event $\{|D_{t_i}| < \epsilon/10^5 |N_{G_{t_i}}(u_i)|\} \wedge M_{u_i}^{t_i}$ happens with high probability. Thus, from Theorem 17 we conclude that u_i is $\epsilon/10$-heavy at time t_i with high probability.

The proof proceeds arguing that the following events happen with high probability:

- For every node $v \in R$ there exists a node $u_i \in L_1$ such that: (1) u_i enters the anchor set at time t_i; (2) u_i is in agreement with node v at time t_i; and (3) u_i does not exit the anchor set before time t_c. Consequently edge (u_i, v) is added to our sparse solution graph $\tilde{G}_{t_i}$ and remains in our sparse solution at least until time t_c.

- Similarly, we argue that for every node $v \in L_1$ there exists a node $u_i \in L_2$ such that (1), (2) and (3) hold. So that edge (u_i, v) is added to our sparse solution graph $\tilde{G_{t_i}}$ and remains in our sparse solution at least until time t_c.
- The last part of the proof argues how the *Connect*$(\cdot, \epsilon)$ procedure clusters nodes in L_2 that do not enter the anchor set with the rest of C. At time t_i when $u_i \in L_2$ participates in an "interesting event" most nodes in R have their neighborhood similar to their final, i.e., at time t_c, neighborhood, consequently they are in agreement with u_i. In addition to that, u_i is in agreement with almost all nodes in L_1, therefore there are many triangles of the form u_i, v, w where $v \in R$, $w \in \Phi_v \cap L_1$ and all three nodes are in agreement. The *Connect*(u_i, ϵ) connects u_i to the rest of the cluster with high probability using the edge (u_i, w) in one of those triangles.

To prove the aforementioned points, we define the following random set:

Definition 18. *For $v \in C$ we define the random set:*

$$T_v = \left\{ i \in \{1, 2, \ldots, \lfloor \epsilon/10^4 |C| \rfloor\} : \overline{M_v^{t_i}} \wedge \{(u_i, v) \in E_{t_i}\} \right\}$$

For a node $v \in C$, Definition 18 captures which times during the last $\lfloor \epsilon/10^4 |C| \rfloor$ "interesting events", node v was connected to u_i and its neighborhood was different from its final one. We proceed by proving that for every node in $v \in R \cup L_1$, $|T_v|$ is small with high probability.

Lemma 19. *Let $v \in R \cup L_1$, then*

$$\mathbf{Pr}[|T_v| \geqslant \epsilon/10^5 |C|] < 1/n^{10^2}$$

Proof. In this proof, for a node $u_i \in L$ when we write I_{u_i} we refer to u_i's neighborhood sample constructed via the connect procedure at time t_i. For short we write $\left[\epsilon/10^4 |C|\right]$ instead of $\{1, 2, \ldots, \lfloor \epsilon/10^4 |C| \rfloor\}$. We first argue that event $\bigcap_{i \in [\epsilon/10^4 |C|]} M_{u_i}^{t_i}$ happens with high probability.

$$\begin{aligned}
\mathbf{Pr}[\bigcap_{i \in [\epsilon/10^4 |C|]} M_{u_i}^{t_i}] &= 1 - \mathbf{Pr}[\exists i \in \left[\epsilon/10^4 |C|\right] : \overline{M_{u_i}^{t_i}}] \\
&\geqslant 1 - \sum_{i \in [\epsilon/10^4 |C|]} \mathbf{Pr}[\overline{M_{u_i}^{t_i}}] \\
&\geqslant 1 - \sum_{i \in [\epsilon/10^4 |C|]} \mathbf{Pr}[\overline{M_{u_i}^{t_i}} \wedge A_{u_i}^{t_i}] \\
&\geqslant 1 - n \cdot 1/n^{10^4} \\
&\geqslant 1 - 1/n^{10^3}
\end{aligned}$$

where in the first inequality we used the union bound, in the second inequality we used that $A_{u_i}^{t_i}$ is always true by definition of u_i and t_i and in the third inequality we used Lemma 14. Thus using

$$\begin{aligned}
\mathbf{Pr}[|T_v| \geqslant \epsilon/10^5 |C|] = \mathbf{Pr}[\{|T_v| \geqslant \epsilon/10^5 |C|\} \wedge \bigcap_{i \in [\epsilon/10^4 |C|]} M_{u_i}^{t_i}] + \mathbf{Pr}[\{|T_v| \geqslant \epsilon/10^5 |C|\} \wedge \overline{\bigcap_{i \in [\epsilon/10^4 |C|]} M_{u_i}^{t_i}}] \\
\leqslant \mathbf{Pr}[\{|T_v| \geqslant \epsilon/10^5 |C|\} \wedge \bigcap_{i \in [\epsilon/10^4 |C|]} M_{u_i}^{t_i}] + 1/n^{10^3}
\end{aligned}$$

We focus on upper bounding the first term of the right hand side. Again, using the law of total probability we have that:

$$\begin{aligned}
\mathbf{Pr}[\{|T_v| \geqslant \epsilon/10^5 |C|\} \wedge \bigcap_{i \in [\epsilon/10^4 |C|]} M_{u_i}^{t_i}] &= \mathbf{Pr}[\{|T_v| \geqslant \epsilon/10^5 |C|\} \wedge \bigcap_{i \in [\epsilon/10^4 |C|]} M_{u_i}^{t_i} \wedge \{\forall i \in T_v \; v \notin I_{u_i}\}] \\
&+ \mathbf{Pr}[\{|T_v| \geqslant \epsilon/10^5 |C|\} \wedge \bigcap_{i \in [\epsilon/10^4 |C|]} M_{u_i}^{t_i} \wedge \{\exists i \in T_v : \; v \in I_{u_i}\}]
\end{aligned}$$

For the first term note that for ϵ small enough and using (1) of Lemma 12 we have that: if $\bigcap_{i\in[\epsilon/10^4|C|]} M_{u_i}^{t_i}$ is true then $\forall i \in [\epsilon/10^4|C|]$ we have that $|N_{G_{t_i}}(u_i)| < 2|C|$. Now consider the following random process: each time our algorithm samples u_i's neighborhood to create I_{u_i} we sample uniformly at random the set $N_{G_{t_i}}(u_i) \cup \{r_{|N_{G_{t_i}}(u_i)|+1}, r_{|N_{G_{t_i}}(u_i)|+2}, \ldots, r_{2|C|}\}$ to create sets $\tilde{I}_{u_i}$ where r_j nodes only exist for analysis purposes. Now note that:

$$\begin{aligned}
&\mathbf{Pr}[\{|T_v| \geqslant \epsilon/10^5|C|\} \wedge \bigcap_{i\in[\epsilon/10^4|C|]} M_{u_i}^{t_i} \wedge \{\forall i \in T_v :\ v \notin I_{u_i}\}] \leqslant \\
&\mathbf{Pr}[\{|T_v| \geqslant \epsilon/10^5|C|\} \wedge \bigcap_{i\in[\epsilon/10^4|C|]} M_{u_i}^{t_i} \wedge \{\forall i \in T_v :\ v \notin \tilde{I}_{u_i}\}] \leqslant \\
&(1 - 1/(2|C|))^{(10^{10}\log n/\epsilon)\cdot(\epsilon/10^5|C|)} \leqslant \\
&(1 - 1/(2|C|))^{10^5 \log n|C|} \leqslant \\
&(1/n)^{10^5/2} \leqslant \\
&1/n^{10^4}
\end{aligned}$$

We continue by upper bounding $\mathbf{Pr}[\{|T_v| \geqslant \epsilon/10^5|C|\} \wedge \bigcap_{i\in[\epsilon/10^4|C|]} M_{u_i}^{t_i} \wedge \{\exists i \in T_v :\ v \in I_{u_i}\}]$ as follows:

$$\begin{aligned}
\mathbf{Pr}[\{|T_v| \geqslant \epsilon/10^5|C|\} \wedge \bigcap_{i\in[\epsilon/10^4|C|]} M_{u_i}^{t_i} \wedge \{\exists i \in T_v :\ v \in I_{u_i}\}] &\leqslant \sum_{i\in\{1,2,\ldots,\epsilon/10^4|C|\}} \mathbf{Pr}[i \in T_v, v \in I_{u_i}] \\
&\leqslant \sum_{i\in\{1,2,\ldots,\epsilon/10^4|C|\}} \mathbf{Pr}[\tilde{M_v^{t_i}} \wedge A_v^{t_i}] \\
&\leqslant n \cdot (1/n^{10^4}) \\
&\leqslant 1/n^{10^4-1}
\end{aligned}$$

Where in the first inequality we used the union bound, in the second inequality we used the fact that event $\{v \in I_{u_i}\}$ implies $A_v^{t_i}$, and in the third one we used Lemma 14.

The proof follows by noting that $1/n^{10^3} + 1/n^{10^4} + 1/n^{10^4-1} < 1/n^{10^2}$ ☐

We continue proving that, with high probability, every node in R is selected by a node in L_1 which enters the anchor set and remains in the anchor set until at least time t_c.

Lemma 20. *For every node $v \in R$ let*

$$X_v = \{\exists i \in [\epsilon/(2\cdot 10^4)|C|] : \{u_i \in L_1 \textit{ enters the anchor set }\} \wedge \{(v, u_i) \in E_{t_{i'}}, \forall i' \in [i, t_c]\}\}$$

then $\mathbf{Pr}[X_v] > 1 - 1/n^{10}$.

Proof. Let $i \in \{1, 2, \ldots, \lfloor \epsilon/(2\cdot 10^4)|C|\rfloor\}$ be such that $\{|D_{t_i}| < \epsilon/10^5|N_{G_{t_i}}(u_i)|\} \wedge M_v^{t_i} \wedge M_{u_i}^{t_i} \wedge \{(u_i, v) \in E_{t_i}\}$ is true, we then know from Theorem 17 that if u_i enters the anchor set at that time t_i, edge (u_i, v) is added to our sparse solution and remains in it at least until time t_c. Thus, it is useful to define the following set:

$$Y_v = \{i \in \{1, 2, \ldots, \lfloor \epsilon/(2\cdot 10^4)|C|\rfloor :\ \{|D_{t_i}| < \epsilon/10^5|N_{G_{t_i}}(u_i)|\} \wedge M_v^{t_i} \wedge M_{u_i}^{t_i} \wedge \{(u_i, v) \in E_{t_i}\}\}$$

We proceed by proving that with high probability $|Y_v|$ is "large".

Note that from Observation 8 we have that:

$$|\{i \in \{1, 2, \ldots, \lfloor \epsilon/(2\cdot 10^4)|C|\rfloor :\ (u_i, v) \in E_{t_i}\}| \geqslant \epsilon/(2\cdot 10^4)|C| - \epsilon/10^{13}|C| > \epsilon/(3\cdot 10^4)|C|$$

where the second inequality holds for ϵ small enough.

For simplicity let:

$$\mathcal{R} = \bigcap_{i \in [\epsilon/(2\cdot 10^4)|C|]} \{\{|D_{t_i}| < \epsilon/10^5 |N_{G_{t_i}}(u_i)|\} \wedge M_{u_i}^{t_i}\} \wedge \{|T_v| < \epsilon/10^5 |C|\}$$

Note that $\mathcal{R}$ implies that:

$$|Y_v| \geqslant \epsilon/(3\cdot 10^4)|C| - \epsilon/10^5|C| > \epsilon/(4\cdot 10^4)|C|$$

where the second inequality holds for ϵ small enough. We continue upper bounding the probability that $\mathcal{R}$ does not occur as follows:

$$\begin{aligned}
\mathbf{Pr}[\overline{\mathcal{R}}] &\leqslant \sum_{i \in [\epsilon/(2\cdot 10^4)|C|]} \mathbf{Pr}[\{|D_{t_i}| \geqslant \epsilon/10^5 |N_{G_{t_i}}(u_i)|\}] \\
&\quad + \sum_{i \in [\epsilon/(2\cdot 10^4)|C|]} \mathbf{Pr}[\overline{M_{u_i}^{t_i}}] \\
&\quad + \mathbf{Pr}[\{|T_v| \geqslant \epsilon/10^5|C|\}] \\
&\leqslant n \cdot (1/n^{10^4}) + n \cdot (1/n^{10^3}) + 1/n^{10^2} \\
&\leqslant 3/n^{10^2}
\end{aligned}$$

where in the first inequality we use the union bound and in the second we use Lemma 14, Lemma 16 and Lemma 19.

Note that $\mathcal{R}$ also implies that u_i enters the anchor set at time t_i with probability at least $\min\{^{10^7 \log n}/_{2\epsilon|C|}, 1\}$ where we used (1) in Lemma 12 for ϵ small enough. In addition, note that under any realization of the random variables $|N_{G_{t_1}}(u_1)|, |N_{G_{t_2}}(u_2)|, \ldots$ the randomness of the *Anchor* procedure is independent from the randomness of the *Connect* procedure. Consequently:

$$\begin{aligned}
\mathbf{Pr}[\overline{X_v}] &= \mathbf{Pr}[\overline{X_v} \cap \overline{\mathcal{R}}] + \mathbf{Pr}[\overline{X_v} \cap \mathcal{R}] \\
&\leqslant \mathbf{Pr}[\overline{\mathcal{R}}] + \mathbf{Pr}[\overline{X_v} \cap \mathcal{R}] \\
&\leqslant 3/n^{10^2} + \mathbf{Pr}[\overline{X_v} \cap \mathcal{R}] \\
&\leqslant 3/n^{10^2} + \mathbf{Pr}[\overline{X_v} \cap \{|Y_v| > \epsilon/(4\cdot 10^4)|C|\}] \\
&\leqslant 3/n^{10^2} + (1 - \min\{^{10^7 \log n}/_{2\epsilon|C|}, 1\})^{\epsilon/(4\cdot 10^4)|C|} \\
&\leqslant 3/n^{10^2} + 1/n^{10^2} \\
&< 1/n^{10}
\end{aligned}$$

□

Using the same line of arguments as Lemma 20 in Lemma 21 we argue that every node in L_1 is selected by a node in L_2 which enters the anchor set and remains in the anchor set until at least time t_c. We state the lemma and omit the proof.

Lemma 21. *For every node $v \in L_1$ let*

$$X_v = \{\exists i \in \{\epsilon/(2\cdot 10^4)|C|+1, \ldots, \epsilon/10^4|C|\} : \{u_i \in L_2 \textit{ enters the anchor set }\} \wedge \{(v, u_i) \in E_{t_{i'}}, \forall i' \in [i, t_c]\}\}$$

then $\mathbf{Pr}[X_v] > 1 - 1/n^{10}$.

Proof. Omitted as it follows the same line of reasoning with the proof of Lemma 20. □

We underline that while the essence of both Lemma 20 and Lemma 21 could be summarized in a single lemma, in the last part of this section we use the facts that Lemma 20 refers to how nodes in R get into the cluster through nodes in L_1 and Lemma 21 refers to how nodes in L_1 get into the cluster through nodes in L_2.

The last part of the section is devoted in arguing that through the *Connect*$(\cdot, \epsilon)$ procedure, with high probability, all nodes in L_2 get connected to some node in L_1 in our sparse solution.

Lemma 22. *For every node $u_j \in L_2$ let*

$$X_{u_j} = \left\{\exists i \in [\epsilon/(2\cdot 10^4)|C|] : \{u_i \in L_1 \textit{ enters the anchor set }\} \wedge \{(u_j, u_i) \in E_{t_{i'}}, \forall i' \in [t_j, t_c]\right\}$$

then

$$\mathbf{Pr}[X_{u_j}] \geqslant 1 - 1/n^{10}$$

Proof. The proof proceeds by arguing that:

1. u_j is connected to almost all nodes R;
2. let $w \in R$, then $N_{G_{t_j}}(u_j) \cap N_{G_{t_j}}(w) \cap L_1$ is a sufficiently large fraction of L_1;
3. with high probability almost all nodes in $N_{G_{t_j}}(u_j) \cap N_{G_{t_j}}(w) \cap L_1$ have almost their final neighborhood; and
4. with high probability one of those nodes is in the anchor set.

It is useful to define the following event:

$$\mathcal{R} = \bigcap_{i\in[\epsilon/(2\cdot 10^4)|C|]} \left\{\{|D_{t_i}| < \epsilon/10^5 |N_{G_{t_i}}(u_i)|\} \wedge M_{u_i}^{t_i}\right\}$$

We upper bound the probability that $\mathcal{R}$ does not occur as follows:

$$\begin{aligned}\mathbf{Pr}[\overline{\mathcal{R}}] \leqslant & \sum_{i\in[\epsilon/(2\cdot 10^4)|C|]} \mathbf{Pr}[\{|D_{t_i}| \geqslant \epsilon/10^5 |N_{G_{t_i}}(u_i)|\}] \\ & + \sum_{i\in[\epsilon/(2\cdot 10^4)|C|]} \mathbf{Pr}[\overline{M_{u_i}^{t_i}}] \\ & \leqslant n\cdot(1/n^{10^4}) + n\cdot(1/n^{10^3}) \\ & \leqslant 1/n^{10^2}\end{aligned}$$

where in the first inequality we use the union bound and in the second we use Lemma 14 and Lemma 16.

Let $w \in N_{G_{t_j}}(u_j) \cap R$, from Observation 7 we have that:

$$\begin{aligned}|N_{G_{t_j}}(u_j) \cap N_{G_{t_j}}(w) \cap L_1| &\geqslant |L_1| - \epsilon/10^{13}|C| - \epsilon/10^{13}|C| \\ &\geqslant \epsilon/(2\cdot 10^4)|C| - \epsilon/10^{13}|C| - \epsilon/10^{13}|C| \\ &\geqslant \epsilon/(4\cdot 10^4)|C|\end{aligned}$$

Moreover, note that event $\mathcal{R}$ implies that at least $(1-\epsilon/10^5)|N_{G_{t_j}}(u_j)|$ neighbors of u_j at time t_j have almost their final neighborhood. Consequently, $\mathcal{R}$ implies the following bound for all $w \in N_{G_{t_j}}(u_i) \cap R$:

$$\begin{aligned}|v \in N_{G_{t_j}}(w) \cap N_{G_{t_j}}(u_j) \cap L_1 :\ M_v^{t_j}| &> \epsilon/(4\cdot 10^4)|C| - \epsilon/10^5 |N_{G_{t_j}}(u_j)| \\ &> \epsilon/(4\cdot 10^4)|C| - 2\epsilon/10^5|C| \\ &> \epsilon/10^5|C|\end{aligned}$$

where in the second inequality we used (1) in Lemma 12 for ϵ small enough.

As in the proof of Lemma 20, $\mathcal{R}$ implies that u_i enters the anchor set at time t_i with probability at least $\min\{^{10^7 \log n}/_{2\epsilon|C|}, 1\}$ (using (1) in Lemma 12 for ϵ small enough). In addition, under any realization of the random variables $|N_{G_{t_1}}(u_1)|, |N_{G_{t_2}}(u_2)|, \ldots$ the randomness of the *Anchor* procedure is independent from the randomness of the other procedures of our algorithm. Consequently, $\forall w \in N_{G_{t_j}}(u_j) \cap R$ let $H_w = \left\{\{v \in N_{G_{t_j}}(w) \cap N_{G_{t_j}}(u_j) \cap L_1 :\ M_v^{t_j}\} \cap \Phi = \emptyset\right\}$, we have:

$$\begin{aligned}\mathbf{Pr}[\mathcal{R} \cap H_w] &\leqslant (1 - \min\{^{10^7 \log n}/_{2\epsilon|C|}, 1\})^{\epsilon/10^5|C|} \\ &\leqslant e^{-10^7 \log n/2\epsilon|C| \cdot \epsilon/10^5|C|} \\ &\leqslant 1/n^{50}\end{aligned}$$

From Observation 9 we have that $|N_{G_{t_j}}(u_j) \cap C| > (1 - 2\epsilon/10^4)|C|$, consequently:

$$\begin{aligned}|N_{G_{t_j}}(u_j) \cap R| &\geqslant |N_{G_{t_j}}(u_j) \cap C| - |C \setminus R| \\ &= |N_{G_{t_j}}(u_j) \cap C| - |L| \\ &\geqslant (1 - 2\epsilon/10^4)|C| - \epsilon/10^4|C| \\ &\geqslant (1 - 3\epsilon/10^4)|C|\end{aligned}$$

Again, using (1) in Lemma 12 for ϵ small enough, $\mathcal{R}$ implies that

$$|N_{G_{t_j}}(u_j) \cap R| \geqslant (1 - 3\epsilon/(2 \cdot 10^4))|N_{G_{t_j}}(u_j)| > (1 - 4\epsilon/10^4)|N_{G_{t_j}}(u_j)|$$

where the second inequality holds for ϵ small enough.

We are now ready to upper bound $\mathbf{Pr}[\overline{X_{u_j}} \cap \mathcal{R}]$, note that J_{u_j} denotes the random sample constructed via the Connect procedure of u_j's neighborhood at time t_j.

$$\begin{aligned}\mathbf{Pr}[\overline{X_{u_j}} \cap \mathcal{R}] &\leqslant \mathbf{Pr}[\overline{X_{u_j}} \cap \mathcal{R} \cap \{J_{u_j} \cap N_{G_{t_j}}(u_j) \cap R = \emptyset\}] \\ &+ \mathbf{Pr}[\overline{X_{u_j}} \cap \mathcal{R} \cap \{J_{u_j} \cap N_{G_{t_j}}(u_j) \cap R \neq \emptyset\}] \\ &\leqslant (4\epsilon/10^4)^{10^5 \log n/\epsilon} + \mathbf{Pr}[\mathcal{R} \bigcap \bigcup_{w \in J_{u_j} \cap N_{G_{t_j}}(u_j) \cap R} H_w] \\ &\leqslant (4\epsilon/10^4)^{10^5 \log n/\epsilon} + n \cdot (1/n^{50}) \\ &\leqslant 1/n^{20}\end{aligned}$$

We conclude again using the law of total probability:

$$\begin{aligned}\mathbf{Pr}[\overline{X_{u_j}}] &= \mathbf{Pr}[\overline{X_{u_j}} \cap \mathcal{R}] + \mathbf{Pr}[\overline{X_{u_j}} \cap \overline{\mathcal{R}}] \\ &\leqslant \mathbf{Pr}[\overline{X_{u_j}} \cap \mathcal{R}] + \mathbf{Pr}[\overline{\mathcal{R}}] \\ &\leqslant 1/n^{20} + 1/n^{10^2} \\ &\leqslant 1/n^{10}\end{aligned}$$

□

Theorem 23. *With probability at least $(1 - 1/n^8)$ all nodes of C are clustered together at time t_c by our algorithm.*

Proof. As in previous lemmas we define the following event:

$$\mathcal{R} = \bigcap_{i \in [\epsilon/10^4|C|]} \{\{|D_{t_i}| < \epsilon/10^5 |N_{G_{t_i}}(u_i)|\} \wedge M_{u_i}^{t_i}\}$$

for which $\mathbf{Pr}[\overline{\mathcal{R}}] \leqslant 1/n^{10^2}$.

From Theorem 17 we have that $\mathcal{R}$ implies that $\forall u_i \in L$ that joins the anchor set at time t_i:

1. they remain there until at least time t_c; and
2. all edges (v, u_i) where $v \in R$ added in our sparse solution are not deleted until at least time t_c.

We now argue that $\mathcal{R}$ also implies that any two nodes $u_i, u_j \in L$ which joined the anchor set at times t_i and t_j respectively belong to the same connected component of our sparse solution at time t_c. To that end, let W_{t_i} and W_{t_j} denote the set of nodes that are in agreement respectively with u_i and u_j at times t_i and t_j.

$$\begin{aligned}|W_{t_i}| &\geqslant (1-\epsilon/10)|N_{G_{t_i}}(u_i)| \\ &> (1-\epsilon/10)(1-2\epsilon/10^4)|C|> \\ &> (1-2\epsilon/10)|C|\end{aligned}$$

where in the second inequality we used Observation 9.

In addition, from Observation 9 we have that $|N_{G_{t_i}}(u_i) \cap C|> (1-2\epsilon/10^4)|C|$, consequently:

$$\begin{aligned}|N_{G_{t_i}}(u_i) \cap R| &\geqslant |N_{G_{t_i}}(u_i) \cap C| - |C \setminus R| \\ &= |N_{G_{t_i}}(u_i) \cap C| - |L| \\ &\geqslant (1-2\epsilon/10^4)|C| - \epsilon/10^4|C| \\ &\geqslant (1-3\epsilon/10^4)|C|\end{aligned}$$

Consequently, combining the latter two inequalities and the fact that $W_{t_i} \subseteq N_{G_{t_i}}(u_i)$:

$$\begin{aligned}|W_{t_i} \cap R| &> (1-2\epsilon/10-3\epsilon/10^4)|C| \\ &> (1-3\epsilon/10)|C|\end{aligned}$$

where the second inequality holds for ϵ small enough.

Thus:

$$\begin{aligned}|W_{t_i} \cap W_{t_j} \cap R| &> (1-6\epsilon/10)|C| \\ &\geqslant 1\end{aligned}$$

where the second inequality holds for ϵ small enough.

To conclude, let $C_{correct}$ be the event that all nodes in C are clustered together by our algorithm at time t_c. Then:

$$\begin{aligned}\mathbf{Pr}[\overline{C_{correct}}] &= \mathbf{Pr}\left[\overline{C_{correct}} \cap \left\{\mathcal{R} \bigcap_{v \in C} X_v\right\}\right] + \mathbf{Pr}\left[\overline{C_{correct}} \cap \overline{\left\{\mathcal{R} \bigcap_{v \in C} X_v\right\}}\right] \\ &\leqslant \mathbf{Pr}\left[\overline{C_{correct}} \cap \left\{\mathcal{R} \bigcap_{v \in C} X_v\right\}\right] + \mathbf{Pr}\left[\overline{\left\{\mathcal{R} \bigcap_{v \in C} X_v\right\}}\right] \\ &\leqslant 0 + \mathbf{Pr}\left[\overline{\left\{\mathcal{R} \bigcap_{v \in C} X_v\right\}}\right] \\ &\leqslant \mathbf{Pr}\left[\left\{\overline{\mathcal{R}} \cup \bigcup_{v \in C} \overline{X_v}\right\}\right] \\ &\leqslant 0 + 1/n^{10^2} + n \cdot 1/n^{10} \\ &< 1/n^8\end{aligned}$$

where in the first inequality we used the law of total probability, in the third we used that $\mathcal{R} \bigcap_{v \in C} X_v$ implies $C_{correct}$ and in the fifth one we used Lemma 20, Lemma 21 and Lemma 22. □

B. All Found Clusters are Dense.

The goal of this section is to prove that all clusters found by our algorithm are dense, i.e., any node u that belongs to a cluster C is connected to almost all nodes in C in graph G_t. We underline that a cluster C found by our algorithm is always induced by a connected component of the sparse solution $\widetilde{G_t}$ and our goal is to prove the main Theorem 39 of this section which states that $\forall u \in C$, $|N_{G_t}(u) \cap C| \geqslant (1 - 541080\epsilon)|C|$ for a small enough ϵ.

Similarly to the notation of Appendix A we denote by t_c the current time and by C a cluster found by our algorithm at that time. For all $u \in C$ we denote by t_u the last time before $t_c + 1$ that u participated in an "interesting event", note that t_u (similarly to the definition of times t_i in Appendix A) is a random variable. We denote by C_{Φ_t} the subset of nodes in C that are in the anchor set at time t, i.e., $\Phi_t \cap C$. We avoid the subscript t in the set C_Φ notation as it will always be clear for the context at what time we are referring to. Equivalently, we denote by $C_{\overline{\Phi}}$ the rest of the nodes in C, i.e., $C \setminus C_\Phi$, at time t.

Initially we prove the two crucial lemmas, these are Lemma 25 and Lemma 27. In both lemmas we use the properties of our notification procedure and argue that with high probability for a node u after time t_u (and at least until time t_c):

- Lemma 25: u does not lose more than a very small fraction of its neighborhood after time t_u.
- Lemma 27 u's neighborhood does not increase with many nodes of "small" degree.

We proceed to the formal statement and proof of these two lemmas and start with Lemma 25 where we actually prove something slightly stronger that what was mentioned in the previous paragraph. To facilitate the description of the next lemmas, similarly to Appendix A, we define the following:

Definition 24. *For a node $v \in C$ and t we define the following events:*

$$\begin{aligned}
T_i^{v,t} &= \{v \textit{ received a } Type_i \textit{ notification at time } t\} \\
T_i^{v,t,t'} &= \{v \textit{ received a } Type_i \textit{ notification during the interval } (t,t']\} \\
\mathcal{T}^{v,t,t'} &= \{v \textit{ participated in an "interesting event" during the interval } (t,t']\} \\
A_v^t &= \{\{v \in \mathcal{I}_t\} \vee \{v \textit{ received a } Type_2 \textit{ notification at time } t\}\} \\
M_v^{t,t'} &= \{|N_{G_t}(v) \setminus N_{G_{t'}}(v)| \leqslant \epsilon/10^5 |N_{G_t}(v)|\}
\end{aligned}$$

Lemma 25. *Let $u \in V$ and times t', t where $t < t'$. Then:*

$$\mathbf{Pr}\left[A_v^t \wedge \overline{T_0^{v,t,t'}} \wedge \overline{M_v^{t,t'}}\right] < 1/n^{10^4}$$

Proof. The proof of the current lemma follows the same line of arguments as the proof of Lemma 14 and it is omitted. □

To facilitate the description of the Lemma 27 we define the following random variable.

Definition 26. *For every node $v \in V$ and times t, t' where $t < t'$ and a positive integer d:*

$$P_{d,t,t'}^v = \big(N_{G_{t'}}(v) \setminus N_{G_t}(v)\big) \cap \big\{w : \exists t'' \in (t,t'] : |N_{G_{t''}}(w)| < 10^2 d\big\}$$

In other words $P_{d,t,t'}^v$ contains all neighbors of v that arrived between times t and t' whose degree at some point between those times is "small".

In Lemma 27 we argue that $|P_{d,t_v,t'}^v| < \epsilon/10^2 d$ with high probability. The intuition behind the latter statement is that if for some t $|P_{d,t,t'}^v| \geqslant \epsilon/10^2 d$ then, with high probability, v participates in an "interesting event" at a time $t'' \in (t,t']$.

Lemma 27. *For every node $v \in V$, time t' and a positive integer d:*

$$\mathbf{Pr}[\{|P_{d,t_v,t'}^v| > \epsilon/10^2 d\}] < 3/n^{10^3}$$

Proof. We define the following event:

$$\mathcal{R} = \bigcap_{u \in V, t_1, t_2 : t_1 < t_2} \left\{ \overline{A_u^{t_1}} \vee T_0^{u,t_1,t_2} \vee M_u^{t_1,t_2} \right\}$$

for which, using Lemma 25 and a union bound, we get: $\mathbf{Pr}[\overline{\mathcal{R}}] \leqslant 1/n^{10^4-3}$. Using the law of total probability, we have:

$$\begin{aligned} \mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\}\right] &= \mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\} \wedge \mathcal{R}\right] + \mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\} \wedge \overline{\mathcal{R}}\right] \\ &\leqslant \mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\} \wedge \mathcal{R}\right] + \mathbf{Pr}\left[\overline{\mathcal{R}}\right] \\ &\leqslant \mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\} \wedge \mathcal{R}\right] + 1/n^{10^4-3} \end{aligned}$$

Thus, in the rest of the proof we focus on upper bounding the term $\mathbf{Pr}\left[\left\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\right\} \wedge \mathcal{R}\right]$.

To that end, for each node $u \in V$ let t_u^{arrival} be its arrival time and note that $\forall u \in P^v_{d,t_v,t'}$ we have that $t_u^{\text{arrival}} \in (t, t']$. We define the following sets:

$$\begin{aligned} S &= \left\{u \in P^v_{d,t_v,t'} : |N_{G_{t_v^{\text{arrival}}}}(v)| \leqslant 10^3 d\right\} \\ L &= \left\{u \in P^v_{d,t_v,t'} : |N_{G_{t_v^{\text{arrival}}}}(v)| > 10^3 d\right\} \end{aligned}$$

The set S contains nodes of $P^v_{d,t_v,t'}$ whose degree when they arrived was relatively "small" and on the contrary L, which is equal to $P^v_{d,t_v,t'} \setminus S$, contains nodes whose degree on arrival was "large". Since $|P^v_{d,t_v,t'}| = |L| + |S|$, we have that event $\left\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\right\} \wedge \{|S| < \epsilon/10^3 d\}$ implies $\{|L| > 9\epsilon/10^3 d\}$. Using the total law of probability we have:

$$\begin{aligned} &\mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\} \wedge \mathcal{R}\right] \\ &= \mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\} \wedge \mathcal{R} \wedge \{|S| \geqslant \epsilon/10^3 d\}\right] + \mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\} \wedge \mathcal{R} \wedge \{|S| < \epsilon/10^3 d\}\right] \\ &\leqslant \mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\} \wedge \mathcal{R} \wedge \{|S| \geqslant \epsilon/10^3 d\}\right] + \mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\} \wedge \mathcal{R} \wedge \{|L| > 9\epsilon/10^3 d\}\right] \\ &\leqslant \mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\} \wedge \{|S| \geqslant \epsilon/10^3 d\}\right] + \mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\} \wedge \mathcal{R} \wedge \{|L| > 9\epsilon/10^3 d\}\right] \end{aligned}$$

We bound each of the two terms separately. For the first term: $\forall u \in S$ let I_u be the sample constructed by u at arrival and note that u sends a Type_0 notification to all nodes in I_u. Since $\forall u \in S \quad t_u^{\text{arrival}} > t_v$ we have that $\{\forall u \in S : v \notin I_u\}$. Thus:

$$\begin{aligned} \mathbf{Pr}\left[\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\} \wedge \{|S| \geqslant \epsilon/10^3 d\}\right] &\leqslant \mathbf{Pr}\left[\{\forall u \in S : v \notin I_u\} \wedge \{|S| \geqslant \epsilon/10^3 d\}\right] \\ &\leqslant (1 - {}^{1}\!/_{10^3 d})^{(10^{10} \log n/\epsilon) \cdot (\epsilon d/10^3)} \\ &\leqslant 1/n^{10^4} \end{aligned}$$

Where in the second inequality we use that $\forall u, u' \in S$ events $\{v \notin I_u\}$ and $\{v \notin I_{u'}\}$ are independent.

We now turn our attention to the second term. Note that $\forall u \in L$ event $A_u^{t_u^{\text{arrival}}}$ is always true and for each node $u \in L$ denote by $\widehat{t}_u \in (t_u, t']$ the last time when $|N_{G_{\widehat{t}_u}}(u)| < 10^2 d$.

$$\begin{aligned} |N_{G_{t_u^{\text{arrival}}}}(u) \setminus N_{G_{\widehat{t}_u}}(u)| &\geqslant |N_{G_{t_u^{\text{arrival}}}}(u)| - |N_{G_{\widehat{t}_u}}(u)| \\ &> 10^3 d - 10^2 d \\ &= 9 \cdot 10^2 d \end{aligned}$$

where in the second inequality we used the definition of set L.

Note that we just argued that event $A_u^{t_u^{\text{arrival}}} \wedge \overline{M_u^{t_u^{\text{arrival}}, \widehat{t}_u}}$ is always true. Consequently $\mathcal{R}$ implies that $u \in L$ will keep getting notifications of Type_0 until its degree is close enough to its degree at time $\widehat{t}_u$, in other words, there exists a time

$\widetilde{t}_u \in (t_u^{\text{arrival}}, \widehat{t}_u]$ such that $T_0^{u,\widetilde{t}_u} \wedge M_u^{\widetilde{t}_u,\widehat{t}_u}$ is true. At time $\widetilde{t}_u$, u receives a Type_0 notification and its degree can be upper bounded as follows:

$$\begin{aligned}|N_{G_{\widetilde{t}_u}}(u)| &\leqslant |N_{G_{\widetilde{t}_u}}(u) \setminus N_{G_{\widehat{t}_u}}(v)| + |N_{G_{\widehat{t}_u}}(v)| \\ &< \epsilon/10^5 |N_{G_{\widetilde{t}_u}}(u)| + 10^2 d\end{aligned}$$

where we used the fact that $M_u^{\widetilde{t}_u,\widehat{t}_u}$ is true. Thus, for ϵ small enough it holds that

$$\begin{aligned}&|N_{G_{\widetilde{t}_u}}(u)| < \frac{10^2 d}{1-\epsilon/10^5} \Rightarrow \\ &|N_{G_{\widetilde{t}_u}}(u)| < 10^3 d\end{aligned}$$

Similarly to the arguments when we were bounding the first term of the sum, $\forall u \in L$ let I_u be the sample constructed by u at time $\widetilde{t}_u$, and note that u sends a Type_1 notification to all nodes in I_u. Since $\forall u \in S \quad \widetilde{t}_u > t_v$ we have that $\{\forall u \in L : v \notin I_u\}$.

$$\begin{aligned}\mathbf{Pr}\left[\left\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\right\} \wedge \mathcal{R} \wedge \left\{|L| > 9\epsilon/10^3 d\right\}\right] &\leqslant \mathbf{Pr}\left[\{\{\forall u \in L : v \notin I_u\}\} \wedge \left\{|L| > 9\epsilon/10^3 d\right\}\right] \\ &\leqslant (1 - {}^{1}\!/_{10^3 d})^{(10^{10} \log n/\epsilon)\cdot(9\epsilon d/10^3)} \\ &\leqslant 1/n^{10^3}\end{aligned}$$

Combining the previous bounds, we conclude the proof of the lemmas as follows:

$$\mathbf{Pr}\left[\left\{|P^v_{d,t_v,t'}| > \epsilon/10^2 d\right\}\right] < 1/n^{10^4} + 1/n^{10^3} + 1/n^{10^4-3} < 3/n^{10^3}$$

$\square$

We now define the following event which will be a crucial for our arguments in the rest of the section

Definition 28.

$$\mathcal{R} = \bigcap_{u \in C, t > t_u} M_u^{t_u,t} \bigcap_{u \in C, d \in [1,n]} \{|P^v_{d,t_v,t_c}| \leqslant \epsilon/10^2 d\}$$

definition From Lemma 25, Lemma 27, and using the union bound we get the following:

Observation 29.

$$\mathbf{Pr}[\mathcal{R}] > 1 - 1/n^{10^2}$$

In addition, by the definition of event $\mathcal{R}$:

Observation 30. *$\mathcal{R}$ implies that for all $v \in C$ and $t \in (t_v, t_c]$:*

1. $|N_{G_{t_v}}(v) \setminus N_{G_t}(v)| \leqslant \epsilon/10^5 |N_{G_{t_v}}(v)|$; *and*
2. $|\left(N_{G_t}(v) \setminus N_{G_{t_v}}(v)\right) \cap \left\{u : |N_{G_t}(u)| < 10^2 |N_{G_t}(v)|\right\}| \leqslant \epsilon/10^2 |N_{G_t}(v)|$

As in the previous Appendix A, we will focus on how the neighborhood of a node $v \in C_\Phi$ can change after the last time it participated in an "interesting event", i.e., time t_v. From Observation 30 we know that with high probability v does not lose more than a small fraction of its neighborhood until time t_c. At the same time, its neighborhood may increase drastically, with many nodes of high degree. Thus, in an approximate sense, we have that $N_{\widetilde{G}_t}(v) \supseteq N_{\widetilde{G}_{t_v}}(v)$. The next Lemma 31 and Corollary 32 argue that for all $t \in (t_v, t_c]$ such that v is in agreement with another node (a new edge adjacent to v may be added to our sparse solution in that case) we have that $N_{G_t}(v) \simeq N_{G_{t_v}}(v)$.

Lemma 31. *Let $u \in \mathcal{C}$ and $t > t_u$ a time when u is in ϵ-agreement with v, and either u or v is ϵ-heavy. Then $\mathcal{R}$ implies that:*

$$(1+10\epsilon)|N_{G_{t_u}}(u)| \geqslant |N_{G_t}(u)| \geqslant (1-\epsilon/10^5)|N_{G_{t_u}}(u)|$$

Proof. The right hand side is implied immediately by (1) of Observation 30. We prove the left hand side for the case when v is ϵ-heavy and omit the case where u is ϵ-heavy as it is proven similarly. Since v is ϵ-heavy and in ϵ-agreement with u, from Property 2 and Property 7, we can deduce that u is 5ϵ-heavy, i.e., it is in 5ϵ-agreement with at least a $(1-5\epsilon)$ fraction of its neighborhood at time t. Thus, again using Property 7, we have that at least $(1-5\epsilon)|N_{G_t}(u)|$ neighboring nodes of u have degree at most $\frac{|N_{G_t}(u)|}{(1-5\epsilon)} < 2|N_{\widetilde{G}_t}(u)|$ for ϵ small enough. Due to (2) of Observation 30, from those nodes at most $\epsilon/10^2|N_{G_t}(u)|$ of them could have arrived after time t_v. Consequently, $|N_{G_t}(u) \setminus N_{G_{t_v}}(u)| \leqslant \epsilon/10^2|N_{G_t}(u)| + 5\epsilon|N_{G_t}(u)| < 6\epsilon|N_{G_t}(u)|$. Using that $|N_{G_t}(u)| - |N_{G_{t_u}}(u)| \leqslant |N_{G_t}(u) \setminus N_{G_{t_u}}(u)|$ we conclude that:

$$|N_{G_t}(u)| \leqslant \frac{|N_{G_{t_u}}(u)|}{1-6\epsilon} < (1+10\epsilon)|N_{G_{t_u}}(u)|$$

where the second inequality holds for ϵ small enough. ☐

Corollary 32. *Let $u \in \mathcal{C}$ and $t \geqslant t_u$ a time when u is in ϵ-agreement with v, and either u or v is ϵ-heavy. Then $\mathcal{R}$ implies that:*

1. $|N_{G_{t_u}}(u) \setminus N_{G_t}(u)| \leqslant \epsilon/10^5|N_{G_{t_u}}(u)| < \epsilon/10^4|N_{G_t}(u)|$; *and*
2. $|N_{G_t}(u) \setminus N_{G_{t_u}}(u)| \leqslant 6\epsilon|N_{G_t}(u)| < 7\epsilon|N_{G_{t_u}}(u)|$

Proof. If $t = t_u$ then the claim trivially holds. In the following we assume that $t > t_u$.

The left hand side of the first inequality is immediate from (1) of Observation 30. For the right hand side of the first inequality note that from Lemma 31 we have $|N_{G_{t_u}}(u)| < 1/(1-\epsilon/10^5)|N_{G_t}(u)|$. Thus, for ϵ small enough we get $(\epsilon/10^5)\cdot(1/(1-\epsilon/10^5)) < \epsilon/10^4$.

For the left hand side of the second inequality it suffices to repeat the arguments of Observation 30 and for the right hand side, again using Observation 30, and the fact that $6\epsilon(1+10\epsilon) < 7\epsilon$ for ϵ small enough. ☐

The next Lemma 33 concentrates on how the neighborhood of a node $u \in C_\Phi$ in our sparse solution changes after time t_u. At a high level, we argue that $N_{\widetilde{G}_{t_u}}(u) \simeq N_{\widetilde{G}_t}(u)$.

Lemma 33. *Let $u \in C_\Phi$ and $t > t_u$. Then $\mathcal{R}$ implies that:*

1. $N_{\widetilde{G}_{t_u}}(u) \subseteq N_{G_{t_u}}(u)$
2. $|N_{G_{t_u}}(u) \setminus N_{\widetilde{G}_{t_u}}(u)| < \epsilon|N_{G_{t_u}}(u)|$
3. $|N_{\widetilde{G}_{t_u}}(u)| \geqslant (1-\epsilon)|N_{G_{t_u}}(u)|$
4. $|N_{\widetilde{G}_{t_u}}(u) \setminus N_{\widetilde{G}_t}(u)| < \epsilon|N_{\widetilde{G}_{t_u}}(u)|$
5. $|N_{\widetilde{G}_t}(u) \setminus N_{\widetilde{G}_{t_u}}(u)| < 8\epsilon|N_{G_{t_u}}(u)|$
6. $|N_{\widetilde{G}_t}(u) \setminus N_{\widetilde{G}_{t_u}}(u)| < 12\epsilon|N_{\widetilde{G}_{t_u}}(u)|$
7. $|N_{\widetilde{G}_t}(u)| < (1+12\epsilon)|N_{\widetilde{G}_{t_u}}(u)|$
8. $(1-\epsilon)|N_{\widetilde{G}_{t_u}}(u)| < |N_{\widetilde{G}_t}(u)|$

Proof. We prove each statement as follows:

1. The neighborhood of a node in our sparse solution is always a subset of its true neighborhood.

2. Since $u \in C_\Phi$, u is ϵ-heavy at time t_u and in ϵ-agreement with at least an ϵ fraction of its neighborhood at that time. All edges (u, v) where $v \in N_{G_{t_u}}(u)$ is in ϵ-agreement with u are added to our sparse solution at that time.

3. Again, due to u being ϵ-heavy at time t_u.

4. Due to the $Clean(\cdot, \epsilon)$ procedure u cannot lose more than an ϵ fraction of its neighborhood in our sparse solution at time t_u and remain in the anchor set.

5. W.l.o.g. we assume that time t was the last time u's neighborhood in the sparse solution increased, and let v be its last new neighbor in the sparse solution. Note that since $t > t_u$, v must be in ϵ-agreement with u at time t and either u or v must be ϵ-heavy. We have:

$$\begin{aligned}|N_{\widetilde{G}_t}(u) \setminus N_{\widetilde{G}_{t_u}}(u)| &\leqslant |N_{\widetilde{G}_t}(u) \setminus N_{G_{t_u}}(u)| + |N_{G_{t_u}}(u) \setminus N_{\widetilde{G}_{t_u}}(u)| \\ &\leqslant |N_{G_t}(u) \setminus N_{G_{t_u}}(u)| + |N_{G_{t_u}}(u) \setminus N_{\widetilde{G}_{t_u}}(u)| \\ &< 7\epsilon |N_{G_{t_u}}(u)| + \epsilon |N_{G_{t_u}}(u)| \\ &\leqslant 8\epsilon |N_{G_{t_u}}(u)|\end{aligned}$$

Where in the second inequality we used the fact that $N_{\widetilde{G}_t}(u) \subseteq N_{G_t}(u)$ is always true, and in the third we used both (2) of the current lemma and (2) of Corollary 32.

6. It is immediate by combining (3) and (5), since for ϵ small enough it holds that $8\epsilon/(1-\epsilon) < 12\epsilon$.

7. It is immediate from (6).

8. It is immediate from (4).

□

The following Lemma 35, Lemma 36 and Corollary 37 pave the road to the main Theorem 39 of the current section by arguing that for any two nodes $u, v \in C_\Phi$ at time $t \geqslant \max\{t_u, t_v\}$ their neighborhood in the sparse solution is very similar, i.e., $N_{\widetilde{G}_t}(u) \simeq N_{\widetilde{G}_t}(v)$. To that end Lemma 35 proves that if $N_{\widetilde{G}_t}(u) \cap N_{\widetilde{G}_t}(v) \neq \emptyset$ then $N_{\widetilde{G}_t}(u) \simeq N_{\widetilde{G}_t}(v)$, Lemma 36 continues arguing that $N_{\widetilde{G}_t}(u) \cap N_{\widetilde{G}_t}(v)$ is always non-empty and Corollary 37 concludes that indeed $N_{\widetilde{G}_t}(u) \simeq N_{\widetilde{G}_t}(v)$.

Before proceeding to Lemma 35 we state a useful set inequality.

Inequality 34. *Let A, B, C and D be four sets, then:*

$$|A \cap B| \geqslant |C \cap D| - |C \setminus A| - |D \setminus B|$$

Proof. Let $s \in C \cap D$. It is enough to prove that the latter implies: $s \in (A \cap B) \cup (C \setminus A) \cup (D \setminus B)$. We consider the following cases:

1. If s is in $A \cap B$, the claim is satisfied.

2. If s is not in $A \cap B$, then either s is not in set A or s is not in set B.
   (a) If s is not in A, then s must be in $C \setminus A$.
   (b) If s is not in B, then s must be in $D \setminus B$.

□

Lemma 35. *Let $u, v \in C_\Phi$ and $t' \geqslant \max\{t_u, t_v\}$ such that $N_{\widetilde{G}_{t'}}(u) \cap N_{\widetilde{G}_{t'}}(v)$ is non-empty. Then $\mathcal{R}$ implies that: $|N_{\widetilde{G}_{t'}}(u) \cap N_{\widetilde{G}_{t'}}(v)| \geqslant (1 - 80\epsilon) \max\{|N_{\widetilde{G}_{t'}}(u)|, |N_{\widetilde{G}_{t'}}(v)|\}$.*

Proof. W.l.o.g. we assume that $t_v \geqslant t_u$, i.e., v entered the anchor set after u. Let $t \geqslant t_v$ be the minimum time such that $N_{\widetilde{G}_t}(u) \cap N_{\widetilde{G}_t}(v) \neq \emptyset$. We distinguish between the following two scenarios

1. At time t a node w participates in an "interesting" event and gets connected to both nodes u and v; or
2. At time t, v participates in an "interesting event" and gets connected to a node w which was already connected to node u. Note that in this case we have $t = t_v$.

We prove that in both scenarios u and v have a very large neighborhood overlap in our sparse solution.

Case 1 We have that u and v are both in ϵ-agreement with w and that either w is ϵ-heavy or both u and v are ϵ-heavy. From the latter observation and using Property 7 we have that:

$$|N_{G_t}(u) \cap N_{G_t}(v)| \geqslant (1 - 5\epsilon) \max\{|N_{G_t}(u)|, |N_{G_t}(v)|\} \tag{3}$$

that is $N_{G_t}(u) \simeq N_{G_t}(v)$.

We proceed arguing that $N_{G_{t_u}}(u) \simeq N_{G_{t_v}}(v)$.

$$\begin{aligned}
|N_{G_{t_u}}(u) \cap N_{G_{t_v}}(v)| &\geqslant |N_{G_t}(u) \cap N_{G_t}(v)| - |N_{G_t}(u) \setminus N_{G_{t_u}}(u)| - |N_{G_t}(v) \setminus N_{G_{t_v}}(v)| \\
&\geqslant |N_{G_t}(u) \cap N_{G_t}(v)| - 7\epsilon |N_{G_{t_u}}(u)| - 7\epsilon |N_{G_{t_v}}(v)| \\
&\geqslant (1 - 5\epsilon) \max\{|N_{G_t}(u)|, |N_{G_t}(v)|\} - 7\epsilon |N_{G_{t_u}}(u)| - 7\epsilon |N_{G_{t_v}}(v)| \\
&\geqslant (1 - 5\epsilon)(1 - \epsilon/10^5) \max\{|N_{G_{t_u}}(u)|, |N_{G_{t_v}}(v)|\} - 7\epsilon |N_{G_{t_u}}(u)| - 7\epsilon |N_{G_{t_v}}(v)| \\
&\geqslant (1 - 20\epsilon) \max\{|N_{G_{t_u}}(u)|, |N_{G_{t_v}}(v)|\}
\end{aligned}$$

Where in the first inequality we use Inequality 34, in the second we use (2) of Corollary 32, in the third we use Eq. (3) of the current proof, in the fourth Lemma 31 and the last one holds for ϵ small enough.

In the same manner we proceed arguing that $N_{\widetilde{G}_{t_u}}(u) \simeq N_{\widetilde{G}_{t_v}}(v)$ by lower bounding $|N_{\widetilde{G}_{t_u}}(u) \cap N_{\widetilde{G}_{t_v}}(v)|$ as follows:

$$\begin{aligned}
|N_{\widetilde{G}_{t_u}}(u) \cap N_{\widetilde{G}_{t_v}}(v)| &\geqslant |N_{G_{t_u}}(u) \cap N_{G_{t_v}}(v)| - |N_{G_{t_u}}(u) \setminus N_{\widetilde{G}_{t_u}}(u)| - |N_{G_{t_v}}(v) \setminus N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant |N_{G_{t_u}}(u) \cap N_{G_{t_v}}(v)| - \epsilon |N_{G_{t_u}}(u)| - \epsilon |N_{G_{t_v}}(v)| \\
&\geqslant |N_{G_{t_u}}(u) \cap N_{G_{t_v}}(v)| - \epsilon/(1-\epsilon) |N_{\widetilde{G}_{t_u}}(u)| - \epsilon/(1-\epsilon) |N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant (1 - 20\epsilon) \max\{|N_{G_{t_u}}(u)|, |N_{G_{t_v}}(v)|\} - \epsilon/(1-\epsilon) |N_{\widetilde{G}_{t_u}}(u)| - \epsilon/(1-\epsilon) |N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant (1 - 20\epsilon) \max\{|N_{G_{t_u}}(u)|, |N_{G_{t_v}}(v)|\} - 3\epsilon |N_{\widetilde{G}_{t_u}}(u)| - 3\epsilon |N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant (1 - 20\epsilon)(1 - \epsilon) \max\{|N_{\widetilde{G}_{t_u}}(u)|, |N_{\widetilde{G}_{t_v}}(v)|\} - 3\epsilon |N_{\widetilde{G}_{t_u}}(u)| - 3\epsilon |N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant (1 - 30\epsilon) \max\{|N_{\widetilde{G}_{t_u}}(u)|, |N_{\widetilde{G}_{t_v}}(v)|\}
\end{aligned}$$

Where in the first inequality we use Inequality 34, in the second inequality we use (2) of Lemma 33, in the third inequality we use (3) of Lemma 33, in the fourth one we used the lower bound on $|N_{G_{t_u}}(u) \cap N_{G_{t_v}}(v)|$ that we just proved in the current lemma, in the fifth and the seventh holds for ϵ small enough and, in the sixth inequality we use again (3) of Lemma 33, and the fifth and seventh inequality hold for ϵ small enough.

We can now finish the first case by lower bounding the $|N_{\widetilde{G}_{t'}}(u) \cap N_{\widetilde{G}_{t'}}(v)|$.

$$\begin{aligned}
|N_{\widetilde{G}_{t'}}(u) \cap N_{\widetilde{G}_{t'}}(v)| &\geqslant |N_{\widetilde{G}_{t_u}}(u) \cap N_{\widetilde{G}_{t_v}}(v)| - |N_{\widetilde{G}_{t_u}}(u) \setminus N_{\widetilde{G}_{t'}}(u)| - |N_{\widetilde{G}_{t_v}}(v) \setminus N_{\widetilde{G}_{t'}}(v)| \\
&\geqslant |N_{\widetilde{G}_{t_u}}(u) \cap N_{\widetilde{G}_{t_v}}(v)| - \epsilon |N_{\widetilde{G}_{t_u}}(u)| - \epsilon |N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant (1 - 30\epsilon) \max\{|N_{\widetilde{G}_{t_u}}(u)|, |N_{\widetilde{G}_{t_v}}(v)|\} - \epsilon |N_{\widetilde{G}_{t_u}}(u)| - \epsilon |N_{\widetilde{G}_{t_v}}(v)|| \\
&\geqslant \frac{1 - 30\epsilon}{1 + 12\epsilon} \max\{|N_{\widetilde{G}_{t'}}(u)|, |N_{\widetilde{G}_{t'}}(v)|\} - \frac{\epsilon}{1-\epsilon} |N_{\widetilde{G}_{t'}}(u)| - \frac{\epsilon}{1-\epsilon} |N_{\widetilde{G}_{t'}}(v)| \\
&\geqslant (1 - 60\epsilon) \max\{|N_{\widetilde{G}_{t'}}(u)|, |N_{\widetilde{G}_{t'}}(v)|\}
\end{aligned}$$

Where in the first inequality we use Inequality 34, in the second we use (4) of Lemma 33, in the third we use the lower bound on $N_{\widetilde{G}_{t_u}}(u) \cap N_{\widetilde{G}_{t_v}}(v)|$ that we proved in the current lemma, in the fourth inequality we use both (7) of Lemma 33 and (8) of Lemma 33, and the last inequality holds for ϵ small enough.

Case 2 We now turn our attention to the second case. In that case w.l.o.g. we assume that u was connected to w in our sparse solution before time t and that at time t, v was inserted in the anchor set and got connected with node w with which they are in ϵ-agreement at time t. Consequently, $|N_{G_t}(v) \cap N_{G_t}(w)| \geqslant (1-\epsilon)\max\{|N_{G_t}(v)|, |N_{G_t}(w)|\}$.

We can also assume that w does not participate in an "interesting event" at time t since this situation is already covered by the first case.

Let $t'' = \max\{t_u, t_w\}$ and note that at that time u and w were in ϵ-agreement and one of them is ϵ-heavy, thus we have that $|N_{G_{t''}}(u) \cap N_{G_{t''}}(w)| \geqslant (1-\epsilon)\max\{|N_{G_{t''}}(u)|, |N_{G_{t''}}(w)|\}$

The goal is to argue that $N_{\widetilde{G}_t}(u) \approx N_{\widetilde{G}_t}(v)$, and after that use Lemma 33 to deduce that $N_{\widetilde{G}_{t'}}(u) \approx N_{\widetilde{G}_{t'}}(v)$.

Towards this goal, we initially prove that $N_{G_{t''}}(w) \approx N_{G_t}(w)$. Note that in both times, t and t'' node w was in ϵ-agreement with another node and got connected to it. We distinguish between two cases:

(a) $t'' = t_u > t_w$; and

(b) $t'' = t_w$

In (a) we prove that $N_{G_{t''}}(w) \simeq N_{G_{t_w}}(w)$ and $N_{G_t}(w) \simeq N_{G_{t_w}}(w)$ to conclude that $N_{G_{t''}}(w) \approx N_{G_t}(w)$.

$$\begin{aligned}
|N_{G_{t''}}(w) \cap N_{G_{t_w}}(w)| &\geqslant \max\{|N_{G_{t''}}(w)|, |N_{G_{t_w}}(w)|\} - |N_{G_{t_w}}(w) \setminus N_{G_{t''}}(w)| - |N_{G_{t''}}(w) \setminus N_{G_{t_w}}(w)| \\
&\geqslant \max\{|N_{G_{t''}}(w)|, |N_{G_{t_w}}(w)|\} - (7\epsilon + \epsilon/10^4)\max\{|N_{G_{t''}}(w)|, |N_{G_{t_w}}(w)|\} \\
&\geqslant (1-8\epsilon)\max\{|N_{G_{t''}}(w)|, |N_{G_{t_w}}(w)|\}
\end{aligned}$$

Where in the second inequality we use (1) of Corollary 32 and (2) of Corollary 32, and the third inequality holds for ϵ small enough.

Similarly, we have

$$\begin{aligned}
|N_{G_t}(w) \cap N_{G_{t_w}}(w)| &\geqslant \max\{|N_{G_t}(w)|, |N_{G_{t_w}}(w)|\} - |N_{G_{t_w}}(w) \setminus N_{G_t}(w)| - |N_{G_t}(w) \setminus N_{G_{t_w}}(w)| \\
&\geqslant \max\{|N_{G_t}(w)|, |N_{G_{t_w}}(w)|\} - (7\epsilon + \epsilon/10^4)\max\{|N_{G_{t''}}(w)|, |N_{G_{t_w}}(w)|\} \\
&\geqslant (1-8\epsilon)\max\{|N_{G_t}(w)|, |N_{G_{t_w}}(w)|\}
\end{aligned}$$

We now argue that $N_{G_{t''}}(w) \simeq N_{G_t}(w)$ as follows:

$$\begin{aligned}
|N_{G_t}(w) \cap N_{G_{t''}}(w)| &\geqslant |N_{G_{t_w}}(w)| - |N_{G_{t_w}}(w) \setminus N_{G_t}(w)| - |N_{G_{t_w}}(w) \setminus N_{G_{t''}}(w)| \\
&\geqslant (1-8\epsilon)\max\{|N_{G_t}(w)|, |N_{G_{t''}}(w)|\} - (8\epsilon + 8\epsilon)\max\{|N_{G_t}(w)|, |N_{G_{t''}}(w)|\} \\
&\geqslant (1-24\epsilon)\max\{|N_{G_t}(w)|, |N_{G_{t''}}(w)|\}
\end{aligned}$$

Case (b) is simpler and we can immediately use (1) of Corollary 32 and (2) of Corollary 32 to argue that $N_{G_{t''}}(w) \simeq N_{G_t}(w)$, as follows:

$$\begin{aligned}
|N_{G_t}(w) \cap N_{G_{t''}}(w)| &\geqslant \max\{|N_{G_t}(w)|, |N_{G_{t''}}(w)|\} - |N_{G_t}(w) \setminus N_{G_{t''}}(w)| - |N_{G_{t''}}(w) \setminus N_{G_t}(w)| \\
&\geqslant \max\{|N_{G_t}(w)|, |N_{G_{t''}}(w)|\} - (7\epsilon + \epsilon/10^4)\max\{|N_{G_{t''}}(w)|, |N_{G_{t_w}}(w)|\} \\
&\geqslant (1-8\epsilon)\max\{|N_{G_t}(w)|, |N_{G_{t''}}(w)|\}
\end{aligned}$$

Where the third inequality holds for ϵ small enough.

Thus, in both cases (a) and (b) we have that:

$$|N_{G_t}(w) \cap N_{G_{t''}}(w)| \geqslant (1-24\epsilon)\max\{|N_{G_t}(w)|, |N_{G_{t''}}(w)|\}$$

We continue arguing that $N_{G_{t''}}(u) \approx N_{G_t}(v)$. Indeed we have

$$\begin{aligned}
|N_{G_t}(v) \cap N_{G_{t''}}(u)| &\geqslant |N_{G_t}(w) \cap N_{G_{t''}}(w)| - |N_{G_t}(w) \setminus N_{G_t}(v)| - |N_{G_{t''}}(w) \setminus N_{G_{t''}}(u)| \\
&\geqslant (1-24\epsilon)\max\{|N_{G_t}(w)|, |N_{G_{t''}}(w)|\} - \epsilon|N_{G_t}(w)| - \epsilon|N_{G_{t''}}(w)| \\
&\geqslant (1-26\epsilon)\max\{|N_{G_t}(w)|, |N_{G_{t''}}(w)|\} \\
&\geqslant \frac{1-26\epsilon}{1-\epsilon}\max\{|N_{G_t}(v)|, |N_{G_{t''}}(u)|\} \\
&\geqslant (1-27\epsilon)\max\{|N_{G_t}(v)|, |N_{G_{t''}}(u)|\}
\end{aligned}$$

Where in the second inequality we used the lower bound on $|N_{G_t}(w) \cap N_{G_{t''}}(w)|$ that we proved in the current lemma and the fact that at times t and t'', w is in ϵ-agreement with nodes v and u respectively and got connected to them in our sparse solution. In the fourth inequality we again used the latter fact and the last inequality holds for ϵ small enough.

We now argue why $N_{G_{t_u}}(u) \approx N_{G_{t_v}}(v)$. Note that $t = t_v$ and that u does not participate in an "interesting event" after t''.

$$\begin{aligned}
|N_{G_{t_v}}(v) \cap N_{G_{t_u}}(u)| &\geqslant |N_{G_t}(v) \cap N_{G_{t''}}(u)| - |N_{G_t}(v) \setminus N_{G_{t_v}}(v)| - |N_{G_{t''}}(u) \setminus N_{G_{t_u}}(u)| \\
&= |N_{G_t}(v) \cap N_{G_{t''}}(u)| - |N_{G_{t''}}(u) \setminus N_{G_{t_u}}(u)| \\
&\geqslant |N_{G_t}(v) \cap N_{G_{t''}}(u)| - 6\epsilon|N_{G_{t''}}(u)| \\
&\geqslant (1-6\epsilon)|N_{G_t}(v) \cap N_{G_{t''}}(u)| \\
&\geqslant (1-6\epsilon)(1-27\epsilon)\max\{|N_{G_t}(v)|, |N_{G_{t''}}(u)|\} \\
&\geqslant (1-\epsilon/10^5)(1-6\epsilon)(1-27\epsilon)\max\{|N_{G_t}(v)|, |N_{G_{t_u}}(u)|\} \\
&= (1-\epsilon/10^5)(1-6\epsilon)(1-27\epsilon)\max\{|N_{G_{t_v}}(v)|, |N_{G_{t_u}}(u)|\} \\
&\geqslant (1-40\epsilon)\max\{|N_{G_{t_v}}(v)|, |N_{G_{t_u}}(u)|\}
\end{aligned}$$

where in the first inequality we use Eq. (3), the second inequality holds since either $t'' > t_u$ and we use (2) of Corollary 32 or $t'' = t_u$ and in that case $|N_{G_{t''}}(u) \setminus N_{G_{t_u}}(u)| = 0$, in the fourth inequality we use the lower bound on $|N_{G_t}(v) \cap N_{G_{t''}}(u)|$ that we proved in the current lemma, in the fifth inequality we use Lemma 31, and the last inequality holds for ϵ small enough.

Using what we just proved, i.e., $|N_{G_{t_v}}(v) \cap N_{G_{t_u}}(u)| \geqslant (1-40\epsilon)\max\{|N_{G_{t_v}}(v)|, |N_{G_{t_u}}(u)|\}$, we now lower bound $|N_{\widetilde{G}_{t_u}}(u) \cap N_{\widetilde{G}_{t_v}}(v)|$ using the same arguments as in the first case.

$$
\begin{aligned}
|N_{\widetilde{G}_{t_u}}(u) \cap N_{\widetilde{G}_{t_v}}(v)| &\geqslant |N_{G_{t_u}}(u) \cap N_{G_{t_v}}(v)| - |N_{G_{t_u}}(u) \setminus N_{\widetilde{G}_{t_u}}(u)| - |N_{G_{t_v}}(v) \setminus N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant |N_{G_{t_u}}(u) \cap N_{G_{t_v}}(v)| - \epsilon |N_{G_{t_u}}(u)| - \epsilon |N_{G_{t_v}}(v)| \\
&\geqslant |N_{G_{t_u}}(u) \cap N_{G_{t_v}}(v)| - \epsilon/(1-\epsilon) |N_{\widetilde{G}_{t_u}}(u)| - \epsilon/(1-\epsilon) |N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant (1-40\epsilon) \max\{|N_{G_{t_u}}(u)|, |N_{G_{t_v}}(v)|\} - \epsilon/(1-\epsilon) |N_{\widetilde{G}_{t_u}}(u)| - \epsilon/(1-\epsilon) |N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant (1-40\epsilon) \max\{|N_{G_{t_u}}(u)|, |N_{G_{t_v}}(v)|\} - 3\epsilon |N_{\widetilde{G}_{t_u}}(u)| - 3\epsilon |N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant (1-40\epsilon)(1-\epsilon) \max\{|N_{\widetilde{G}_{t_u}}(u)|, |N_{\widetilde{G}_{t_v}}(v)|\} - 3\epsilon |N_{\widetilde{G}_{t_u}}(u)| - 3\epsilon |N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant (1-50\epsilon) \max\{|N_{\widetilde{G}_{t_u}}(u)|, |N_{\widetilde{G}_{t_v}}(v)|\}
\end{aligned}
$$

And conclude the proof of the second case, in a similar manner to the first case, that is, proving $N_{\widetilde{G}_t}(u) \approx N_{\widetilde{G}_t}(v)$.

$$
\begin{aligned}
|N_{\widetilde{G}_{t'}}(u) \cap N_{\widetilde{G}_{t'}}(v)| &\geqslant |N_{\widetilde{G}_{t_u}}(u) \cap N_{\widetilde{G}_{t_v}}(v)| - |N_{\widetilde{G}_{t'}}(u) \setminus N_{\widetilde{G}_{t_u}}(u)| - |N_{\widetilde{G}_{t'}}(v) \setminus N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant |N_{\widetilde{G}_{t_u}}(u) \cap N_{\widetilde{G}_{t_v}}(v)| - 12\epsilon |N_{\widetilde{G}_{t_u}}(u)| - 12\epsilon |N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant (1-50\epsilon) \max\{|N_{\widetilde{G}_{t_u}}(u)|, |N_{\widetilde{G}_{t_v}}(v)|\} - 12\epsilon |N_{\widetilde{G}_{t_u}}(u)| - 12\epsilon |N_{\widetilde{G}_{t_v}}(v)| \\
&\geqslant (1-74\epsilon)/(1+12\epsilon) \max\{|N_{\widetilde{G}_{t'}}(u)|, |N_{\widetilde{G}_{t'}}(v)|\} \\
&\geqslant (1-80\epsilon) \max\{|N_{\widetilde{G}_{t'}}(u)|, |N_{\widetilde{G}_{t'}}(v)|\}
\end{aligned}
$$

Where in the second inequality we use (6) of Lemma 33, in the third inequality we use the lower bound on $N_{\widetilde{G}_{t_u}}(u) \cap N_{\widetilde{G}_{t_v}}(v)|$ that we proved in the current lemma, in the fourth inequality we use (7) of Lemma 33 and the last inequality holds for ϵ small enough.

□

Lemma 36. *$\mathcal{R}$ implies that $\forall u, v \in C_\Phi$: $N_{\widetilde{G}_{t_c}}(u) \cap N_{\widetilde{G}_{t_c}}(v) \neq \emptyset$.*

Proof. The proof follows very similar arguments to the (Cohen-Addad et al., 2021) that we repeat here for completeness. Let $d(x, y)$ be the distance of two nodes x, y in our sparse solution at time t_c. Suppose the lemma is not true, then we have that $d(u, v) > 2$. Towards a contradiction also assume u, v to be the minimum distance nodes in C_Φ such that $d(u, v) > 2$. Note that for any edge in our sparse solution $\widetilde{G}_{t_c}$, one of its endpoints is in C_Φ. If $d(u, v) \geqslant 5$, let $P = \langle u, u', u'', \ldots, v\rangle$ be the shortest u-v path in $\widetilde{G}_{t_c}$, thus either u' or u'' must be in C_Φ, contradicting the minimality assumption regarding the distance between u and v. If $d(u, v) \leqslant 4$, then either $d(u, v) = 4$ or $d(u, v) = 3$. We end up in a contradiction in both cases:

1. If $d(u, v) = 4$ then $\exists v', w, u'$ such that $v' \in N_{\widetilde{G}_{t_c}}(v) \cap N_{\widetilde{G}_{t_c}}(w)$, $w \in C_\Phi$ and $u' \in N_{\widetilde{G}_{t_c}}(w) \cap N_{\widetilde{G}_{t_c}}(u)$.
2. If $d(u, v) = 3$ then $\exists w, u'$ such that $w \in N_{\widetilde{G}_{t_c}}(v) \cap N_{\widetilde{G}_{t_c}}(u')$, $u' \in N_{\widetilde{G}_{t_c}}(w) \cap N_{\widetilde{G}_{t_c}}(u)$ and either w or u are in C_Φ. Assume that $w \in C_\Phi$ (and the case where $u' \in C_\Phi$ is similar)

In both cases, from Lemma 35 we have:

$$
\begin{aligned}
|N_{\widetilde{G}_{t_c}}(u) \cap N_{\widetilde{G}_{t_c}}(v)| &\geqslant |N_{\widetilde{G}_{t_c}}(w)| - |N_{\widetilde{G}_{t_c}}(w) \setminus N_{\widetilde{G}_{t_c}}(u)| - |N_{\widetilde{G}_{t_c}}(w) \setminus N_{\widetilde{G}_{t_c}}(v)| \\
&\geqslant |N_{\widetilde{G}_{t_c}}(w)| - 80\epsilon |N_{\widetilde{G}_{t_c}}(u)| - 80\epsilon |N_{\widetilde{G}_{t_c}}(v)| \\
&\geqslant (1-80\epsilon) \max\{|N_{\widetilde{G}_{t_c}}(u)|, |N_{\widetilde{G}_{t_c}}(v)|\} - 80\epsilon |N_{\widetilde{G}_{t_c}}(u)| - 80\epsilon |N_{\widetilde{G}_{t_c}}(v)| \\
&\geqslant (1-240\epsilon) \max\{|N_{\widetilde{G}_{t_c}}(u)|, |N_{\widetilde{G}_{t_c}}(v)|\} \\
&> 0
\end{aligned}
$$

Where the last inequality holds for ϵ small enough. Thus, we ended in a contradiction. ☐

Corollary 37. *Let $u, v \in C_\Phi$, $\mathcal{R}$ implies that* $|N_{\widetilde{G}_{t_c}}(u) \cap N_{\widetilde{G}_{t_c}}(v)| \geqslant (1 - 80\epsilon) \max\{|N_{\widetilde{G}_{t_c}}(u)|, |N_{\widetilde{G}_{t_c}}(v)|\}$.

Proof. It is immediate from Lemma 35 and Lemma 36. ☐

Before proceeding to the last theorems of this section we state some useful set inequalities.

Observation 38. *For sets A, B, C and positive reals α, β:*

1. *If* $|A \triangle B| \leqslant \epsilon \max\{|A|, |B|\}$ *then* $|A \cap B| \geqslant (1 - \epsilon) \max\{|A|, |B|\}$.
2. *If* $|A \cap B| \geqslant (1 - \epsilon) \max\{|A|, |B|\}$ *then for ϵ small enough:*
$$|A \triangle B| \leqslant 2\epsilon \max\{|A|, |B|\} \leqslant \frac{2\epsilon}{1-\epsilon} \min\{|A|, |B|\} \leqslant 3\epsilon \min\{|A|, |B|\}$$
3. *If* $|A \triangle B| \leqslant \epsilon \max\{|A|, |B|\}$ *then for ϵ small enough* $|A \triangle B| \leqslant 3\epsilon \min\{|A|, |B|\}$
4. $|A \triangle C| \leqslant |A \triangle B| + |B \triangle C|$.
5. $\alpha \min\{|A|, |B|\} + \beta \min\{|B|, |C|\} \leqslant (\alpha + \beta) \max\{|A|, |C|\}$

We now proceed proving the main Theorem 39 of this section. Note that on Theorem 39 we also argue that $\forall v \in C$ their degree when they last participated in an "interesting event" is lower bounded by the $|C|/2$. The latter property will be crucial in Appendix C.

Theorem 39. *$\mathcal{R}$ implies that cluster C is dense and $\forall v \in C$:*

$$|N_{G_{t_c}}(v) \cap C| \geqslant (1 - 541080\epsilon)|C|$$

and

$$|N_{G_{t_u}}(v)| \geqslant |C|/2.$$

Proof. Note that for node $v \in C_{\overline{\Phi}}$ its neighborhood in $\widetilde{G}_{t_c}$ is a subset of C_Φ. To relate the neighborhood of v inside C in G_{t_c} we construct an auxiliary graph which acts as a bridge between $\widetilde{G}_{t_c}$ and G_{t_c}. We construct a graph $\widehat{G}_{t_c}$ so that:

1. $\forall u \in C$ and $u' \in C_\Phi$ $N_{\widehat{G}_{t_c}}(u) \simeq N_{\widetilde{G}_{t_c}}(u')$;
2. $\widehat{G}_{t_c} \subseteq G_{t_c}$;
3. C is a connected component in $\widehat{G}_{t_c}$; and
4. $\forall u \in C$ $|N_{\widehat{G}_{t_c}}(v) \cap C| \geqslant (1 - 541080\epsilon)|C|$

The existence of such a graph suffices to demonstrate the current theorem. For each $v \in C_{\overline{\Phi}}$ let $\widehat{t}_v$ be the last time before t_c that v was in ϵ-agreement with a node $u \in C_\Phi$, one of either u or v was heavy and edge (u, v) is added to the sparse solution and remains until time t_c. Note that $\widehat{t}_v \geqslant t_v$, otherwise we would have that $v \notin C$ and $\widehat{t}_v \geqslant t_u$ since edge (u, v) remains in our sparse solution until time t_c.

We now prove that the neighborhood $N_{G_{\widehat{t}_v}}(v)$ is almost the same to the neighborhood of $N_{\widetilde{G}_{t_c}}(u')$ for any $u' \in C_\Phi$. We first derive the following inequalities:

(a) $|N_{G_{\widehat{t}_v}}(v) \triangle N_{G_{\widehat{t}_v}}(u)| \leqslant \epsilon \max\{|N_{G_{\widehat{t}_v}}(v)|, |N_{G_{\widehat{t}_v}}(u)|\} \leqslant 3\epsilon \min\{|N_{G_{\widehat{t}_v}}(v)|, |N_{G_{\widehat{t}_v}}(u)|\}$

(b) $|N_{G_{\widehat{t}_v}}(u) \triangle N_{G_{t_u}}(u)| \leqslant 8\epsilon \max\{|N_{G_{\widehat{t}_v}}(u)|, |N_{G_{t_u}}(u)|\} \leqslant 24\epsilon \min\{|N_{G_{\widehat{t}_v}}(u)|, |N_{G_{t_u}}(u)|\}$

(c) $|N_{G_{t_u}}(u) \triangle N_{\widetilde{G}_{t_u}}(u)| \leqslant \epsilon \max\{|N_{G_{t_u}}(u)|, |N_{\widetilde{G}_{t_u}}(u)|\} \leqslant 3\epsilon \min\{|N_{G_{t_u}}(u)|, |N_{\widetilde{G}_{t_u}}(u)|\}$

(d) $|N_{\widetilde{G}_{t_u}}(u) \triangle N_{\widetilde{G}_{t_c}}(u)| \leqslant 13\epsilon \max\{|N_{\widetilde{G}_{t_u}}(u)|, |N_{\widetilde{G}_{t_c}}(u)|\} \leqslant 39\epsilon \min\{|N_{\widetilde{G}_{t_u}}(u)|, |N_{\widetilde{G}_{t_c}}(u)|\}$

(e) $|N_{\widetilde{G}_{t_c}}(u) \triangle N_{\widetilde{G}_{t_c}}(u')| \leqslant 160\epsilon \max\{|N_{\widetilde{G}_{t_c}}(u)|, |N_{\widetilde{G}_{t_c}}(u')|\} \leqslant 240\epsilon \min\{|N_{\widetilde{G}_{t_c}}(u)|, |N_{\widetilde{G}_{t_c}}(u')|\}$

The right hand side of each inequality in (a), (b), (c), (d) and (e) uses Observation 38 and the left hand side of (a) uses that u and v are in ϵ-agreement at time $\widehat{t}_v$, (b) uses Corollary 32, (c) uses (1) of Lemma 33 and (2) of Lemma 33, (d) uses (4) of Lemma 33 and (6) of Lemma 33 and (e) uses Corollary 37. By using the triangle inequality of Observation 38 iteratively we can conclude that:

$$|N_{G_{\widehat{t}_v}}(v) \cap N_{\widetilde{G}_{t_c}}(u')| \geqslant (1 - 1113\epsilon) \max\{|N_{G_{\widehat{t}_v}}(v)|, ||N_{\widetilde{G}_{t_c}}(u')|\}$$

With the same line of reasoning for a node $u \in C_\Phi$ we can define $\widehat{t}_u$ as the last time that an edge (u, v) where $v \in C$ was added to our sparse solution and remained until at least time t_c. Inequalities (b), (c), (d) and (e) remain true and following the same arguments we conclude that:

$\forall u, u' \in C_\Phi$:

$$|N_{G_{\widehat{t}_u}}(u) \cap N_{\widetilde{G}_{t_c}}(u')| \geqslant (1 - 1113\epsilon) \max\{|N_{G_{\widehat{t}_u}}(u)|, ||N_{\widetilde{G}_{t_c}}(u')|\}$$

Consequently, $\forall u \in C$ and $u' \in C_\Phi$:

$$|N_{G_{\widehat{t}_u}}(u) \cap N_{\widetilde{G}_{t_c}}(u')| \geqslant (1 - 1113\epsilon) \max\{|N_{G_{\widehat{t}_u}}(u)|, |N_{\widetilde{G}_{t_c}}(u')|\} \tag{*}$$

Let $d = \max\{|N_{\widetilde{G}_{t_c}}(u')|: u' \in C_\Phi\}$. Using Eq. (*) we can conclude that for ϵ small enough:

$$\frac{d}{2} \leqslant (1 - 1113\epsilon)d \leqslant |N_{G_{\widehat{t}_u}}(u)| \leqslant \frac{d}{(1 - 1113\epsilon)} \leqslant 2d \tag{**}$$

Also note that since $N_{\widetilde{G}_{t_c}}(u') \subseteq C$ we also have that $\forall v \in C_{\overline{\Phi}}$ and $u' \in C_\Phi$:

$$|(N_{G_{\widehat{t}_v}}(v) \cap C) \cap N_{\widetilde{G}_{t_c}}(u')| \geqslant (1 - 1113\epsilon) \max\{|N_{G_{\widehat{t}_v}}(v) \cap C|, ||N_{\widetilde{G}_{t_c}}(u')|\} \tag{***}$$

and using the same arguments as in the case of $|N_{G_{\widehat{t}_u}}(u)|$ we can also lower and upper bound $|N_{G_{\widehat{t}_v}}(v) \cap C|$ as follows:

$$\frac{d}{2} \leqslant (1 - 1113\epsilon)d \leqslant |N_{G_{\widehat{t}_v}}(v) \cap C| \leqslant \frac{d}{(1 - 1113\epsilon)} \leqslant 2d \tag{****}$$

To facilitate the description of the intermediate graph $\widehat{G}_{t_c}$ for each $u \in C$ let $E_u = \{(u, v) : v \in N_{G_{\widehat{t}_u}}(u) \cap C\}$ be the set of edges between u and nodes in $N_{G_{\widehat{t}_u}}(u) \cap C$. Note that all these edges also exist in G_{t_c}. We define $\widehat{G}_{t_c} = (C, \bigcup_{u \in C} E_u)$ to be the graph with set of nodes to be C and edges $\bigcup_{u \in C} E_u$. Let N_u be the neighborhood of u in that graph, then:

1. $N_u \supseteq N_{G_{\widehat{t}_u}}(u) \cap C$
2. $\widehat{G}_{t_c} \subseteq G_{t_c}$
3. $N_u = (N_{G_{\widehat{t}_u}}(u) \cap C) \cup \{v \in C : \{t_v^{\text{arrival}} > \widehat{t}_u\} \wedge \{(v \in N_{G_{t_c}}(u)\}\}$

While (1) and (2) are immediate for (3) we further elaborate. Note that for node $u \in C$ we can construct its neighborhood in $\widehat{G}_{t_c}$ as follows: first add all edges to nodes in $N_{G_{\widehat{t}_u}}(u) \cap C$ and then for every node $v \in N_{G_{t_c}}(u) \cap (C \setminus N_{G_{\widehat{t}_u}}(u))$ such that $\widehat{t}_v > \widehat{t}_u$ add edge (u, v). Then (3) follows by noting that sets $\{v \in N_{G_{t_c}}(u) \cap (C \setminus N_{G_{\widehat{t}_u}}(u)) : \widehat{t}_v > \widehat{t}_u\}$ and $\{v \in C : \{t_v^{\text{arrival}} > \widehat{t}_u\} \wedge \{(v \in N_{G_{t_c}}(u)\}\}$ are equal. We now argue that $N_u \simeq (N_{G_{\widehat{t}_u}}(u) \cap C)$. For each node $v \in C$ such that $t_v^{\text{arrival}} > \widehat{t}_u$ and $v \in N_{G_{t_c}}(u)$ it holds that:

1. $v \in N_{G_{t_c}}(u) \setminus N_{G_{\widehat{t}_u}}(u)$; and
2. $|N_{G_{\widehat{t}_v}}(v)| < 2d$ where $\widehat{t}_v > t_u$

For the second point note that $\widehat{t}_v \geqslant t_v^{\text{arrival}}$, $t_v^{\text{arrival}} > \widehat{t}_u$ and $\widehat{t}_u \geqslant t_u$. From these two points note that $\mathcal{R}$ implies that $|\{v \in C : \{t_v^{\text{arrival}} > \widehat{t}_u\} \wedge \{(v \in N_{G_{t_c}}(u)\}| < \epsilon/10^2 d$ and from Eq. (****) we have that $|N_{G_{\widehat{t}_u}}(u) \cap C| \geqslant \frac{d}{2}$. Combining these facts:

$$\begin{aligned} |N_u \triangle (N_{G_{\widehat{t}_u}}(u) \cap C)| &= |N_u \setminus (N_{G_{\widehat{t}_u}}(u) \cap C)| \\ &= |\{v \in C : \{t_v^{\text{arrival}} > \widehat{t}_u\} \wedge \{(v \in N_{G_{t_c}}(u)\}\}| \\ &\leqslant \epsilon/10^2 d \\ &\leqslant 2\epsilon/10^2 |N_{G_{\widehat{t}_u}}(u) \cap C| \\ &\leqslant 2\epsilon/10^2 \min\{|N_u|, |N_{G_{\widehat{t}_u}}(u) \cap C|\} \end{aligned}$$

Also applying Observation 38 to Eq. (***) we get that $\forall u \in C_{\overline{\Phi}}$ and $u' \in C_\Phi$:

$$|(N_{G_{\widehat{t}_u}}(u) \cap C) \triangle N_{\widetilde{G}_{t_c}}(u')| \leqslant 3339\epsilon \min\{|N_{G_{\widehat{t}_u}}(u) \cap C|, ||N_{\widetilde{G}_{t_c}}(u')|\}$$

Thus, again from Observation 38 and combining:

1. $|(N_{G_{\widehat{t}_u}}(u) \cap C) \triangle N_{\widetilde{G}_{t_c}}(u')| \leqslant 3339\epsilon \min\{|N_{G_{\widehat{t}_u}}(u) \cap C|, ||N_{\widetilde{G}_{t_c}}(u')|\}$; and
2. $|N_u \triangle (N_{G_{\widehat{t}_u}}(u) \cap C)| \leqslant 2\epsilon/10^2 \min\{|N_u|, |N_{G_{\widehat{t}_u}}(u) \cap C|\}$

We conclude that $\forall u \in C$ and $u' \in C_\Phi$:

$$\begin{aligned} |N_u \triangle N_{\widetilde{G}_{t_c}}(u')| &\leqslant 3(3339 + 2\epsilon/10^2) \min\{|N_u|, |N_{\widetilde{G}_{t_c}}(u')|\} \\ &\leqslant 10020\epsilon \min\{|N_u|, |N_{\widetilde{G}_{t_c}}(u')|\} \end{aligned} \tag{*****}$$

Moreover, applying the latter inequality for any $u, w \in C$ and $u' \in C_\Phi$ we conclude that:

$$\begin{aligned} |N_u \triangle N_w| &\leqslant 3(10020 + 10020) \min\{|N_u|, |N_w|\} \\ &\leqslant 60120\epsilon \min\{|N_u|, |N_w|\} \\ &\leqslant 60120\epsilon \max\{|N_u|, |N_w|\} \end{aligned}$$

and

$$|N_u \cap N_w| \geqslant 60120\epsilon \max\{|N_u|, |N_w|\}$$

Thus, $\forall u, w \in C$:

1. u and w are connected; and
2. If u and w are connected in $\widehat{G}_{t_c}$ then they are in 60120ϵ-agreement

Thus, C is a cluster in the agreement decomposition of graph $\widehat{G}_{t_c}$ when agreement and heaviness parameters are set to 60120ϵ. Using Property 4 we can deduce that $\forall u \in C$

$$|N_u \cap C| \geqslant (1 - 9 \cdot 60120\epsilon)|C| = (1 - 541080\epsilon)|C|$$

We now proceed lower bounding $|N_{G_{t_u}}(u)|$. For a node $u \in C$ and $u' \in C_\Phi$ using (c), (d), (e) and Observation 38 we can deduce that:

$$\begin{aligned} |N_{G_{t_u}}(u) \triangle N_{\widetilde{G}_{t_c}}(u')| &\leqslant 3 \cdot (3 \cdot (3 + 39) + 240)\epsilon \min\{|N_{G_{t_u}}(u)|, |N_{\widetilde{G}_{t_c}}(u')|\} \\ &\leqslant 1098\epsilon \min\{|N_{G_{t_u}}(u)|, |N_{\widetilde{G}_{t_c}}(u')|\} \end{aligned}$$

Combining the latter with Eq. (*****) we get:

$$\begin{aligned}|N_{G_{t_u}}(u) \triangle N_u| &\leqslant 3 \cdot (10020 + 1098)\epsilon \min\{|N_{G_{t_u}}(u)|, |N_u|\} \\ &\leqslant 33354\epsilon \min\{|N_{G_{t_u}}(u)|, |N_u|\}\end{aligned}$$

We are now ready to deduce that:

$$\begin{aligned}|N_{G_{t_u}}(u)| &\geqslant (1 - 33354\epsilon)|N_u| \\ &\geqslant (1 - 33354\epsilon)|N_u \cap C| \\ &\geqslant (1 - 33354\epsilon)(1 - 541080\epsilon)|C| \\ &\geqslant (1 - 574434\epsilon)|C| \\ &\geqslant |C|/2\end{aligned}$$

where in the third inequality we used that C is dense and the last inequality holds for ϵ small enough. □

C. Runtime Analysis

In this section we compute the complexity of our algorithm and prove that, on expectation, the time spent per each node insertion or deletion is $O(\operatorname{polylog} n)$. Note that the ϵ used in Algorithm 3 and in all proofs is a small enough constant which gets "absorbed" in the big $O(\cdot)$ notation. We introduce some auxiliary random variable definitions:

1. W_t denotes the complexity of operations at time t;
2. L is the size of the largest connected component of $\widetilde{G}_{t-1}$; and
3. Q_t is the total number of agreement and heaviness calculations at time t.

Note that edge additions are at least as many as edge deletions and each edge addition is preceded by a heaviness and agreement calculation between its endpoints. Thus, the total complexity of Algorithm 3 is upper bounded by a constant times the total complexity of agreement and heaviness calculations. Since from Appendix E each agreement and heaviness calculation requires $O(\log n)$ time we have that:

$$\begin{aligned}\mathbf{E}\left[\sum_t W_t\right] &\leqslant O(\log n) \cdot \mathbf{E}\left[\sum_t Q_t\right] \\ &\leqslant O(\log n) \cdot \sum_t \mathbf{E}\left[Q_t\right]\end{aligned}$$

The focus in the rest of the section will be to prove that $\mathbf{E}\left[Q_t\right] = O(\operatorname{polylog} n)$. We start by arguing that the number of nodes that participate in an "interesting event" at time t is, on expectation, at most $\operatorname{polylog} n$.

Lemma 40.

$$\mathbf{E}\left[|\mathcal{I}_t|\right] = O(\operatorname{polylog} n)$$

Proof. In the case of a node addition it is always true that $|\mathcal{I}_t| \leqslant \left(10^{10} \log n/\epsilon\right)^4 = O(\operatorname{polylog} n)$. To argue that the same bound holds on expectation in the case of a node deletion we introduce some auxiliary notation. For a node $v \in G_{t'}$ we use $I_v^{i,t'}$ and $B_v^{i,t'}$ to denote the sets I_v^i and B_v^i respectively at the end of iteration t'. At time t we have that:

$$\begin{aligned}|\mathcal{I}_t| &\leqslant \left|\bigcup_i B_{v_t}^{i,t-1}\right| \cdot \left(10^{10} \log n/\epsilon\right)^3 \\ &\leqslant \sum_i |B_{v_t}^{i,t-1}| \cdot \Theta\left(\operatorname{polylog} n\right)\end{aligned}$$

Also, our algorithm maintains the following invariant:

$$\sum_{v \in G_{t'},i} |I_v^{i,t'}| = \sum_{v \in G_{t'},i} |B_v^{i,t'}| = O(n \operatorname{polylog} n), \ \forall t'$$

Thus, at time t if we select uniformly at random a node and delete it, we have that:

$$\mathbf{E}\left[\sum_i |B_{v_t}^{i,t}|\right] = O(\operatorname{polylog} n)$$

□

The main Theorem 41 of this section requires a delicate coupling argument. We give a proof sketch before proceeding to the formal proof. Similarly to Appendix B let t_u denote the last time strictly before t that u participated in an "interesting event". We remind the reader an important event definition of Appendix B which will be used also in this section.

Definition 28.

$$\mathcal{R} = \bigcap_{u \in C} M_u^{t_u,t-1} \bigcap_{u \in C, d \in [1,n]} \{|P^u_{d,t_u,t-1}| \leqslant \epsilon/10^2 d\}$$

Let C_u be the connected component u belongs to in $\widetilde{G}_{t-1}$ then from Theorem 39 we have that: $\mathcal{R}$ implies that $\forall v \in C_u$ it holds that $|N_{G_{t_u}}(v)| \geqslant |C_u|/2$.

Since event $\mathcal{R}$ happens with high probability and $|C_u| \leqslant n$, we can argue that $\mathbf{E}[N_{G_{t_u}}(u)|] \geqslant |C_u|/3$. Consequently for a connected component C of $\widetilde{G}_{t-1}$ we can upper bound, on expectation, the number of nodes in the anchor set as follows:

$$\begin{aligned}\sum_{u \in C} {}^{10^7 \log n}/_{\epsilon |N_{G_{t_u}}(u)|} &\simeq \sum_{u \in C} {}^{3 \cdot 10^7 \log n}/_{\epsilon |C|} \\ &= {}^{3 \cdot 10^7 \log n}/_{\epsilon} \\ &= O(\log n)\end{aligned}$$

Consequently $\mathbf{E}[L] = O(\log n)$. In the last step of our proof we argue that $\mathbf{E}[W_t] = O(\log n)\mathbf{E}[|\mathcal{I}_t|]\mathbf{E}[L] = O(\operatorname{polylog} n)$

Theorem 41.

$$\mathbf{E}\left[\sum_t W_t\right] = O(\operatorname{polylog} n)$$

Proof. As already mentioned in the beginning of the section it is enough to argue that $\mathbf{E}[Q_t] = O(\operatorname{polylog} n)$. We use Φ to denote the set of anchor set nodes at time $t-1$ and by L the maximum number of anchor set nodes in any connected component of our sparse solution at time $t-1$.

For a node $v \in \widetilde{G}_t$, on expectation, the number of agreement and heaviness calculations for each subroutine is upper bounded as follows:

1. Connect(v, ϵ, t): $10^7 \log n/\epsilon \cdot L = L \cdot O(\log n)$
2. Anchor(v, ϵ, t): $\min\left\{\{|N_{G_u}(t)| \cdot \frac{10^7 \log n}{\epsilon |N_{G_u}(t)|}, |N_{G_u}(t)|\}\right\} = O(\log n)$
3. Clean(v, ϵ, t): L

Let $W_{t,v}$ be the total number of agreement and heaviness calculations required by those three procedures, then:

$$\mathbf{E}[W_{t,v}] = O(\log n) \cdot \mathbf{E}[L]$$

We continue upper bounding $\mathbf{E}[L] \leqslant O(\log n)$.

From the law of total expectation we have:

$$\begin{aligned}\mathbf{E}[L] &= \mathbf{E}[L \mid \mathcal{R}] \cdot \mathbf{Pr}[R] + \mathbf{E}[L \mid \overline{\mathcal{R}}] \cdot \mathbf{Pr}[\overline{R}] \\ &\leqslant \mathbf{E}[L \mid \mathcal{R}] + n \cdot \mathbf{Pr}[\overline{R}] \\ &\leqslant \mathbf{E}[L \mid \mathcal{R}] + n \cdot 1/n^{10^2} \\ &< \mathbf{E}[L \mid \mathcal{R}] + 1/n^{99}\end{aligned}$$

where in the first inequality we used that $\mathbf{Pr}[R] \leqslant 1$ and $L \leqslant n$ are always true and in the second inequality we used Observation 29. We now concentrate on bounding the first term of the summation. We remind the reader that u_i is the i-th node in our node stream.

From the principled of deferred decisions we can construct an instance of $\widetilde{G}_{t-1}$ as follows: we first run Notify(v_1, ϵ), Notify(v_2, ϵ), $\ldots$, Notify(v_{t-1}, ϵ) sequentially and calculate $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_{t-1}$. Note that at this point for each node u in G_{t-1}, the random variable t_u is realized, indeed $t_u = \max\{i \in [1, t-1] : u \in \mathcal{I}_i\}$. Moreover Φ is not realized, since we did not sample any Bernoulli variable from our Anchor procedure yet. We now run the "for loop" of our algorithm which contains the Clean, Connect and Anchor procedures, sequentially for $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_{t-1}$. Let F denote a realization of the first step and let X_u be the Bernoulli variables used in the Anchor(u, ϵ, t_u) procedures for all u's.

Towards a coupling argument we now describe a second stochastic procedure to construct a "sparse graph" $\widehat{G}_{t-1}$: we again run the Notify(v_1, ϵ), Notify(v_2, ϵ), $\ldots$, Notify(v_{t-1}, ϵ) sequentially and calculate $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_{t-1}$. In the second step we run the "for loop" of our algorithm sequentially for $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_{t-1}$ assuming that all the Bernoulli variables which were sampled by the Anchor procedures are 1. As a final step we use the variables X_u (also used by the first procedure) and delete from the anchor set all nodes for which $X_u = 0$ and also delete all edges between nodes u, v such that both X_u and X_v are 0.

Let $\mathcal{F}$ be the set of possible realizations of the first step conditioned on event $\mathcal{R}$. From the law of total expectation it suffices to prove that $\forall F \in \mathcal{F}$ we have:

$$\mathbf{E}[L \mid F \wedge \mathcal{R}] = O(\log n)$$

Let $\widehat{\mathcal{C}}$ denote the set of connected components of the sparse graph constructed by the second stochastic procedure at the second step. Note that given F, $\widehat{\mathcal{C}}$ is a deterministic set. Similarly let $\mathcal{C}$ be the random variable denoting the set of connected components of $\widetilde{G}_{t-1}$. Note that the following is always true: $\forall C \in \mathcal{C}$ there exists a set $C' \in \widehat{\mathcal{C}}$ such that $C \subseteq C'$. Now, for all sets $C' \in \widehat{\mathcal{C}}$ let $L_{C'}$ be the random variable denoting the number of anchor set nodes in set C' after the third step. We have that $\max_{C' \in \widehat{\mathcal{C}}} L_{C'} \geqslant L$. Consequently, it is enough to bound the following quantity:

$$\mathbf{E}[\max_{C' \in \widehat{\mathcal{C}}} L_{C'} \mid F \wedge \mathcal{R}] = O(\log n)$$

Note that Theorem 39 also applies to the clustering $\widehat{\mathcal{C}}$. Let $C' \in \widehat{\mathcal{C}}$ then $\forall u \in C'$:

$$\mathbf{E}[X_u \mid F \wedge \mathcal{R}] \leqslant {}^{2 \cdot 10^7 \log n}/_{\epsilon |C'|}$$

$$\mathbf{E}[L_{C'} \mid F \wedge \mathcal{R}] = \mathbf{E}[\sum_{u \in C'} X_u \mid F \wedge \mathcal{R}] \leqslant {}^{2 \cdot 10^7 \log n}/_{\epsilon}$$

Using Chernoff we can get that:

$$\mathbf{Pr}[L_{C'} > 2 \cdot \frac{2 \cdot 10^7 \log n}{\epsilon} \mid F \wedge \mathcal{R}] \leqslant 1/n^{10}$$

Let $T = \{\forall C' \in \widehat{\mathcal{C}} : L_{C'} \leqslant 2 \cdot \frac{2 \cdot 10^7 \log n}{\epsilon}\}$, we can conclude that:

$$\begin{aligned}
\mathbf{E}[\max_{C' \in \widehat{\mathcal{C}}} L_{C'} \mid F \wedge \mathcal{R}] &= \mathbf{E}[\max_{C' \in \widehat{\mathcal{C}}} L_{C'} \mid T \wedge F \wedge \mathcal{R}] \cdot \mathbf{Pr}[T] + \mathbf{E}[\max_{C' \in \widehat{\mathcal{C}}} L_{C'} \mid \overline{T} \wedge F \wedge \mathcal{R}] \cdot \mathbf{Pr}[\overline{T}] \\
&\leqslant 2 \cdot \frac{2 \cdot 10^7 \log n}{\epsilon} + n \cdot \mathbf{Pr}[\overline{T}] \\
&\leqslant 2 \cdot \frac{2 \cdot 10^7 \log n}{\epsilon} + n \cdot n \cdot 1/n^{10} \\
&= O(\operatorname{polylog} n)
\end{aligned}$$

where in the first inequality we used the definition of event T and that $\max_{C' \in \widehat{\mathcal{C}}} L_{C'} \leqslant n$ and in the second inequality used a union bound on the sets of $\widehat{\mathcal{C}}$.

We now conclude the proof. Note that while $W_{t,v}$ only depends on the random choices of our algorithm up until time $t-1$, $\mathcal{I}_t$ is independent of those. Consequently:

$$\begin{aligned}
\mathbf{E}[Q_t] &= \mathbf{E}\left[\sum_{v \in V} \mathbb{1}\{v \in \mathcal{I}_t\} W_{t,v}\right] \\
&= \sum_{v \in V} \mathbf{E}[\mathbb{1}\{v \in \mathcal{I}_t\} W_{t,v}] \\
&= \sum_{v \in V} \mathbf{E}[\mathbb{1}\{v \in \mathcal{I}_t\}] \, \mathbf{E}[W_{t,v}] \\
&\leqslant \sum_{v \in V} \mathbf{E}[\mathbb{1}\{v \in \mathcal{I}_t\}] \, O(\log n) \mathbf{E}[L] \\
&\leqslant O(\log n) \mathbf{E}[L] \sum_{v \in V} \mathbf{E}[\mathbb{1}\{v \in \mathcal{I}_t\}] \\
&\leqslant O(\log n) \mathbf{E}[L] \, \mathbf{E}\left[\sum_{v \in V} \mathbb{1}\{v \in \mathcal{I}_t\}\right] \\
&\leqslant O(\log n) \mathbf{E}[L] \, \mathbf{E}[|\mathcal{I}_t|] \\
&\leqslant O(\operatorname{polylog} n)
\end{aligned}$$

where in the last inequality we used Lemma 40.

□

D. Efficient Clustering Computation from Sparse Solutions

In previous sections we prove that our algorithm computes a series of sparse graphs $\widetilde{G}_1, \widetilde{G}_2, \ldots, \widetilde{G}_n$ such that with high probability:

1. a total update time complexity of $\Theta(n \operatorname{polylog} n)$ is required; and
2. $\forall t$, the connected components of $\widetilde{G}_t$ define constant factor approximation correlation clustering for graph G_t.

This section aims to demonstrate how we can compute and output the connected components of any sparse graph $\widetilde{G}_t$ in $\Theta(n)$ time. Note that $\widetilde{G}_t$ contains at most $\Theta(n \operatorname{polylog} n)$ edges. Consequently, a naive approach of computing connected components by traversing all edges of $\widetilde{G}_t$ would result in a time complexity of $\Theta(n \operatorname{polylog} n)$.

It is worth emphasizing that the complexity of the offline algorithm proposed in (Assadi & Wang, 2022) is also $\Theta(n \operatorname{polylog} n)$. Therefore, if $\Theta(n \operatorname{polylog} n)$ time is required just to compute the connected components of $\widetilde{G}_t$, simply rerunning the algorithm from (Assadi & Wang, 2022) would be an equivalent solution our algorithm; rendering our effort in maintaining a sequence of sparse graphs meaningless.

Following the notation of the previous sections, let $\widetilde{G} = (V, \widetilde{E})$ be a sparse graph maintained by our algorithm, Φ the set of anchor nodes and $\Phi_v = N_{\widetilde{G}}(v) \cap \Phi$ the anchor set nodes which are connected to v in our sparse solution $\widetilde{G}$. To ease notation, we write $u \sim v$ if and only if u and v belong to the same connected component of $\widetilde{G}$.

We will describe a procedure that constructs a function $f : V \to \mathbb{N}$. f will encode our clustering solution as follows: two nodes $u, v \in V$ are in the same cluster if and only if $f(u) = f(v)$. We then argue that: if $\forall u, v \in \Phi$ which are in the same connected of $\widetilde{G}$ it holds that $N_{\widetilde{G}}(u) \cap N_{\widetilde{G}}(v) \neq \emptyset$ then f's clustering coincides with the connected components of $\widetilde{G}$. Note that from Appendix B we can use Lemma 36 to prove that the latter property holds with high probability. To ease notation, We write $u \sim v$ if and only if u and v belong to

Algorithm 7 ComputeConnectedComponents($\widetilde{G}$)

```
Initialization: ∀v ∈ V: f(v) ← −1, ID ← 0, Q ← Φ
  while Q ≠ ∅ do
    Let u be any node in Q
    f(u) ← ID
    T ← {u}
    for all v ∈ N_G̃(u) do
      if f(v) = −1 then
        No conflict phase
        f(v) ← ID
        T ← T ∪ {v}
      else
        Resolving conflict phase
        for all v′ ∈ T do
          f(v′) ← f(v)
        end for
        break from for loop
      end if
    end for
    ID ← ID + 1
    Q ← Q \ T
  end while
  for all v ∈ V \ Φ such that f(v) = −1 do
    Let u be a node in Φ_v
    f(v) ← f(u)
  end for
```

Lemma 42. *If for $\widetilde{G}$ and Φ:*

1. $V = \bigcup_{v \in \Phi} N_{\widetilde{G}}(v)$*; and*
2. $\forall u, v \in \Phi$ *such that* $u \sim v$, $N_{\widetilde{G}}(u) \cap N_{\widetilde{G}}(v) \neq \emptyset$

then $f(u) = f(v)$ *if and only if* $u \sim v$.

Proof. Note that after the while loop is terminated all nodes in Φ are assigned a value different than -1. In addition, $\forall v \in V \setminus \Phi$ there exists a node $u \in \Phi_v$ such that $f(v) = f(u)$. Thus, it suffices to argue that for any two nodes $u, v \in \Phi$: $f(u) = f(v)$ if and only if $u \sim v$. Let u_i be the node selected from Q in the beginning of the while's loop i-th iteration, f_i the f function assignment after the termination of the i-th iteration and Φ_i all the nodes in $u \in \Phi$ such that $f_i(u) \neq -1$. Note that for any node u if $f_i(u) \neq -1$ then $\forall j > i$: $f_j(u) \neq -1$ and $f_j(u) = f_i(u)$. Consequently, $\Phi_i \subset \Phi_{i+1}, \forall i$. We prove by induction the following statements in tandem:

1. if $f(u_i) = i$ then $\forall j \geqslant i$ and $v \in N_{\widetilde{G}}(u)$ $f_j(v) = k$; and

2. $\forall u, v \in \Phi_i$: $f_i(u) = f_i(v)$ if and only if $u \sim v$.

For $i = 1$ note that the algorithm updates the f function only in the No conflict phase for all nodes in $N_{\widetilde{G}}(u_1)$ with the same $ID = 1$. Thus, both statements hold. For $i > 1$ we consider two cases:

1. If f is updated only in the No conflict phase then all nodes in $N_{\widetilde{G}}(u_i)$ receive the same $ID = i$. Since $ID = i$ is first used in iteration i all nodes that received an ID in previous iterations maintain their previous ID (which is smaller than i). To conclude this case, we need to argue that $\forall u \in \Phi_{i-1}$ $u \not\sim u_i$. Towards a contradiction assume that such a node exists. Let $u \in \Phi_{i-1}$ be such that $f_{i-1}(u) = j < i$ and $u \sim u_i$. By definition, u_j is the first node to receive $ID = j$ and since $\sim$ is a transitive relation $u_j \sim u_i$. By our inductive hypothesis $\forall v \in N_{\widetilde{G}}(u_j)$ $f_{i-1}(v) = j$ and by the conditions of the current lemma, $\exists w \in N_{\widetilde{G}}(u_j) \cap N_{\widetilde{G}}(u_i)$. However, this is a contradiction because f would have been updated also in Resolving conflict phase.

2. If f is updated also in the Resolving conflict phase then we have that $f_i(u_i) = j < i$ and it suffices to argue that there exists a node $u \in \Phi_{i-i}$ such that $f_{i-1}(u) = j$ it holds that $u \sim u_i$. Let $v \in N_{\widetilde{G}}(u_i)$ be a node such that $f_{i-1}(v) = j \neq -1$. Note that such a node exists since f is also updated in the Resolving conflict phase. W.l.o.g. assume that the value of f at v was updated at time $j' \in [j, i-1]$. Then, either $v \in \Phi_{i-i}$ or $v \in N_{\widetilde{G}}(u_{j'})$ where for $u_{j'} \in \Phi_{i-1}$ it holds that $f_{i-1}(u_{j'}) = j$. In both cases since $\sim$ is a transitive relation we conclude that a node $u \in \Phi_{i-i}$ such that $f_{i-1}(u) = j$ and $u \sim u_i$ always exists.

□

Lemma 43. *Algorithm 7 has complexity $O(|V|)$.*

Proof. The final for loop has complexity at most $\Theta(|V|)$. To bound the complexity of the while loop we simply note that for every node v its f function value changes at most twice. Consequently, operations of the form "$f(\cdot) \leftarrow$" are at most $2|V|$. □

E. Agreement and Heaviness Calculation with $O(\log n)$ Sampled Nodes

In this section we design the ProbabilisticAgreement(u, v, ϵ) and Heavy(u, ϵ) procedures.

These two procedure are used to test if two nodes u and v are in ϵ-agreement and if a node u is ϵ-heavy. The idea is that if we let some slack on how much in agreement and how much heavy a node is then $O(\text{polylog}\, n)$ samples of each node's neighborhood are enough to design a ProbabilisticAgreement(u, v, ϵ) procedure that, w.h.p. answers affirmatively if the two nodes are indeed in 0.1ϵ-agreement and at the same time answers negatively if these two nodes are not in ϵ-agreement. Consequently, if two nodes are in an ϵ'-agreement for ϵ' between 0.1ϵ and ϵ then ProbabilisticAgreement(u, v, ϵ) may answer positively or negatively. We do the same for the Heavy(u, ϵ) procedure.

We start by proving some useful inequalities regarding the neighborhood of two nodes in case they are indeed in ϵ-agreement and in case they are not. Subsequently we analyze the properties of ProbabilisticAgreement(u, v, ϵ) and finally we do the same for the Heavy(u, ϵ) procedure.

Algorithm 8 PROBABILISTICAGREEMENT(u, v, ϵ)

```
Initialization: k = 300 log n/ε.
for i = 1 to k do
    Draw a random neighbor r_i of u and a random neighbor s_i of v.
    Let x_i = 1{r_i ∈ N(u) \ N(v)}, y_i = 1{r_i ∈ N(v) \ N(u)}
end for
if Σ_i x_i/k < 0.4ε and Σ_i y_i/k < 0.4ε then
    Output " YES"
else
    Output "NO"
end if
```

Algorithm 9 HEAVY(u, ϵ)

Initialization: $k = 1200 \log n/\epsilon$.
for $i = 1$ to k **do**
  Draw a random neighbor v_i of u.
  Let $x_i = \mathbb{1}\{\text{ProbabilisticAgreement}(u, v_i, \epsilon) = \text{“No”}\}$
end for
if $\sum_i x_i/k < 1.2\epsilon$ **then**
  Output “ Yes”
else
  Output “No”
end if

Proposition 44. *Let u, v be two nodes which are in ϵ-agreement then $\frac{|N(v)|}{1-\epsilon} \geqslant |N(u)| \geqslant (1-\epsilon)|N(v)|$.*

Proof. Assume w.l.o.g. that $|N(u)| > |N(v)|$. Then $|N(u)| - |N(v)| \leqslant |N(u) \setminus N(v)| \leqslant |N(u) \triangle N(v)| \leqslant \epsilon \max\{|N(u)|, |N(v)|\}$ □

Proposition 45. *Let u, v be two nodes which are in ϵ-agreement then $\frac{|N(u)\setminus N(v)|}{|N(u)|} \leqslant 1.1\epsilon$ and $\frac{|N(v)\setminus N(u)|}{|N(v)|} \leqslant 1.1\epsilon$.*

Proof. $|N(u) \setminus N(v)| \leqslant |N(u) \triangle N(v)| \leqslant \epsilon \max\{|N(u)|, |N(v)|\} \leqslant \epsilon \frac{|N(u)|}{1-\epsilon} \leqslant 1.1|N(u)|$ for ϵ small enough. □

Proposition 46. *Let u, v be two nodes which are not in ϵ-agreement then either $\frac{|N(u)\setminus N(v)|}{|N(u)|} \geqslant 0.5\epsilon$ or $\frac{|N(v)\setminus N(u)|}{|N(v)|} \geqslant 0.5\epsilon$.*

Proof. Since $|N(v) \setminus N(u)| + |N(u) \setminus N(v)| = |N(u) \triangle N(v)| \geqslant \epsilon \max\{|N(u)|, |N(v)|\}$ then either $|N(v) \setminus N(u)| > 0.5\epsilon|N(v)|$ or $|N(u) \setminus N(v)| > 0.5\epsilon|N(u)|$. □

Observation 47. *Let u, v be two nodes which share an edge, s, t two nodes chosen uniformly at random from $N(u)$ and $N(v)$ respectively and $X_s = \mathbb{1}\{s \in N(u) \setminus N(v)\}$, $X_t = \mathbb{1}\{s \in N(v) \setminus N(u)\}$. We have the following:*

1. *If u, v are in agreement then $\mathbf{Pr}[X_s = 1] \leqslant 1.1\epsilon$ and $\mathbf{Pr}[X_t = 1] \leqslant 1.1\epsilon$.*
2. *If u, v are not in agreement then either $\mathbf{Pr}[X_s = 1] \geqslant 0.5\epsilon$ or $\mathbf{Pr}[X_t = 1] \geqslant 0.5\epsilon$.*

Theorem 48. *(Chernoff bound) Let $X_1, X_2, \ldots, X_k$ be k i.i.d. random variables in $[0, 1]$. Let $X = \sum_i X_i/k$. Then:*

1. *For any $\delta \in [0, 1]$ and $U \geqslant E[X]$ we have*
$$\mathbf{Pr}[X \geqslant (1+\delta)U] \leqslant \exp\bigl(-\delta^2 Uk/3\bigr)$$
2. *For any $\delta > 0$ and $U \leqslant E[X]$ we have*
$$\mathbf{Pr}[X \leqslant (1-\delta)U] \leqslant \exp\bigl(-\delta^2 Uk/2\bigr)$$

Lemma 49. *If Algorithm 8 outputs “YES” then u, v are in ϵ-agreement with probability greater than $1 - 1/n^3$.*

Proof. We proceed by upper bounding the probability that the algorithm outputs “YES” and u, v are not in agreement. Since u, v are not in agreement by observation 47 we have that either $E[\sum_i X_i/k] > 0.5\epsilon$ or $E[\sum_i Y_i/k] > 0.5\epsilon$. W.l.o.g. assume that $E[\sum_i X_i/k] > 0.5\epsilon$. By using the second inequality of the Chernoff bound with $U = 0.5\epsilon$ and $\delta = 0.2$. We have
$$\mathbf{Pr}[\sum_i X_i/k < (1-\delta)U] \leqslant \exp\bigl(-0.2^2 \cdot 0.5\epsilon \cdot 300 \log n/\epsilon \cdot 0.5\bigr) = \exp(-3 \cdot \log n) = \frac{1}{n^3}$$
□

Lemma 50. *If u, v are in 0.1ϵ-agreement then Algorithm 8 outputs “YES” with probability greater than $1 - 1/n^3$.*

Proof. We will upper bound the probability that u, v are in 0.1ϵ-agreement and the algorithm outputs "NO". Note that since u, v are in 0.1ϵ-agreement by observation 47 we have that $E[\sum_i X_i/k] < 1.1 \cdot 0.1\epsilon = 0.11\epsilon$ and $E[\sum_i Y_i/k] \leqslant 1.1 \cdot 0.1\epsilon = 0.11\epsilon$. The algorithm outputs "NO" if either $\sum_i X_i/k \geqslant 0.4\epsilon$ or $\sum_i Y_i/k \geqslant 0.4\epsilon$. We bound the probability of the first event using the first inequality of the Chernoff bound with $U = 0.4\epsilon$ and $\delta = 2$ ($> 0.4/0.11 - 1$). We have

$$\mathbf{Pr}[\sum_i X_i/k \geqslant (1+\delta)U] \leqslant \exp\bigl(-2^2 \cdot 0.11\epsilon \cdot 300 \log n/\epsilon \cdot 0.334\bigr) < \exp(-44 \cdot \log n) = 1/n^{44}$$

Overall the probability that the algorithm outputs "NO" by the union bound is upper bounded by $1/n^{44} + 1/n^{44} < 1/n^3$ $\square$

Lemma 51. *Let u be a node and v a random neighbor of u. Let $X_v = \mathbb{1}\{\text{ProbabilisticAgreement}(u, v, \epsilon) = \text{"No"}\}$. We have the following:*

1. *If u is in $\underline{0.1\epsilon\text{-agreement}}$ with a $\underline{(1-\epsilon)\text{-fraction}}$ of its neighborhood then $\mathbf{Pr}[X_u = 1] \leqslant 1.1\epsilon$.*
2. *If u is not in $\underline{\epsilon\text{-agreement}}$ with a $\underline{\epsilon\text{-fraction}}$ of its neighborhood then $\mathbf{Pr}[X_u = 1] > 3\epsilon$.*

Proof. Let A_{uv} be the event that u and v are in 0.1ϵ-agreement and B_{uv} be the event that u and v are not in ϵ-agreement. For (1) we have:

$$\mathbf{Pr}[X_u = 1] \ = \mathbf{Pr}[X_u = 1 \mid A_{uv}] \cdot \mathbf{Pr}[A_{uv}] + \mathbf{Pr}[X_u = 1 \mid \widetilde{A_{uv}}] \cdot \mathbf{Pr}[\widetilde{A_{uv}}] \leqslant 1/n^3 \cdot 1 + 1 \cdot \epsilon < 1.1\epsilon$$

and for (2) we have that:

$$\mathbf{Pr}[X_u = 1] \ = \mathbf{Pr}[X_u = 1 \mid B_{uv}] \cdot \mathbf{Pr}[B_{uv}] + \mathbf{Pr}[X_u = 1 \mid \widetilde{B_{uv}}] \cdot \mathbf{Pr}[\widetilde{B_{uv}}] \geqslant (1 - 1/n^3) \cdot 1 + 1 \cdot (1-\epsilon) > 3\epsilon$$

$\square$

Lemma 52. *If u is in 0.1ϵ-agreement with at least a $(1-\epsilon)$-fraction of its neighborhood, then Algorithm 9 outputs "Yes", i.e., that the node is heavy, with probability greater than $1 - 1/n^3$.*

Proof. We will upper bound the probability that Algorithm 9 outputs "No". Note that from lemma 51 $E[\sum_i X_i/k] > 1.1\epsilon$. By using the first inequality of the Chernoff bound with $U = 1.1\epsilon$ and $\delta = 1/11$. We have:

$$\mathbf{Pr}[\sum_i X_i/k \geqslant (1+\delta)U] \leqslant \exp\bigl(-1/11^2 \cdot 1.1\epsilon \cdot 1200 \log n/\epsilon \cdot 1/3\bigr) < \exp(-3.63 \cdot \log n) < 1/n^3$$

$\square$

Lemma 53. *If u is not in ϵ-agreement with at least a ϵ-fraction of its neighborhood, then Algorithm 9 outputs "No", i.e., that the node is not heavy, with probability greater than $1 - 1/n^3$.*

Proof. We will upper bound the probability that Algorithm 9 outputs "Yes". Note that from lemma 51 $E[\sum_i X_i/k] > 3\epsilon$. By using the second inequality of the Chernoff bound with $U = 3\epsilon$ and $\delta = 0.6$. We have:

$$\mathbf{Pr}[\sum_i X_i/k \leqslant (1-\delta)U] \leqslant \exp\bigl(-0.6^2 \cdot 3\epsilon \cdot 1200 \log n/\epsilon \cdot 1/2\bigr) < \exp(-648 \cdot \log n) < 1/n^3$$

$\square$

F. Final Theorem

We are now ready to prove that the clustering produced by our algorithm for every t is a constant factor approximation to the optimal correlation clustering solution for graph G_t. To this end note that by Lemma 4 it is enough to argue that with high probability:

1. all dense enough clusters found by the agreement algorithm on each graph G_t are also identified by our algorithm and all their nodes are clustered together; and
2. all clusters that are found by our dynamic agreement algorithm are dense enough

For (1) we use Theorem 23 and for (2) we use Theorem 39.

Theorem 6. *For each time t the Dynamic Agreement algorithm outputs an $O(1)-$approximate clustering with probability at least $1 - 5/n$.*

Proof. Theorem 23 assumes the following event: for all pair of nodes u, v that are in $\epsilon/10^{14}$-agreement and all nodes u' that are $\epsilon/10^{14}$-heavy the ProbabilisticAgreement(u, v, ϵ) and Heavy(u', ϵ) procedures of Appendix E output "Yes" and Theorem 39 assumes that: for all or all pair of nodes u, v that are not in ϵ-agreement and all nodes u' that are not ϵ-heavy the ProbabilisticAgreement(u, v, ϵ) and Heavy(u', ϵ) procedures of Appendix E output "No". Using Lemma 49, Lemma 50, Lemma 52 and Lemma 53 we bound the probability that this event does not happen by $n^2/n^3 + n^2/n^3 + n/n^3 + n/n^3 < 4/n$. Upper bounding over all times and possible clusters and using Theorem 23 and Theorem 39 we conclude that the probability of the dynamic algorithm to output at each time a $O(1)$-approximation is at least:

$$1 - n^3/n^{10^2} - n^3/n^{10^2} - 4/n > 1 - 5/n$$

□

G. Structural Properties of the *Agreement* Decomposition

Let G be the graph and let $\mathcal{C}$ be the clustering produced by AGREEMENTALGORITHM(G). Let u, v be two nodes which belong to the same non-trivial cluster C of $\mathcal{C}$, then for ϵ small enough the following properties hold, which were shown in (Cohen-Addad et al., 2022a).

Property 1 $|N_G(u) \cap C| \geqslant (1 - 3\epsilon)|N_G(u)|$

Property 2 $|N_G(u) \setminus C| < 3\epsilon|N_G(u)|$

Property 3 $|C| \geqslant (1 - 3\epsilon)|N_G(u)|$

Property 4 $|N_G(u) \cap C| \geqslant (1 - 9\epsilon)|C|$

Property 5 $|C \setminus N_G(u)| < 9\epsilon|C|$

Property 6 $|N_G(u)| \geqslant (1 - 9\epsilon)|C|$

Property 7 $|N_G(u) \cap N_G(v)| \geqslant (1 - 5\epsilon) \max\{|N_G(u)|, |N_G(v)|\}$

Property 8 $|N_G(v)|(1 - 5\epsilon) \leqslant |N_G(u)| \leqslant \frac{|N_G(v)|}{1-5\epsilon}$

Property 9 $|C \setminus N_G(u)| < 9\epsilon|C| < \frac{9\epsilon}{1-9\epsilon}|N_G(u)|$

Property 10 $|N_G(u) \setminus C| < 3\epsilon|N_G(u)| < \frac{3\epsilon}{1-3\epsilon}|C|$

Property 11 $N_G(u) \cap N_G(v) \neq \emptyset$

H. Additional Experiments

Here we have the experiments for the rest of the datasets presented in Section 5. Moreover, we present the performance of each dataset/algorithm pair when we restrict the node stream to only additions and we calculate the objective after all nodes have arrived. The correlation clustering objective value of each algorithm is divided by the performance of SINGLETONS. Since PIVOT is (on expectation) a 3-approximation we note that the solution achieved by all other algorithms is at most a multiplicative factor 3 away from the optimum offline solution.

Table 3. Performance on the entire graph when node stream contains only additions

Dataset	AGREE-STATIC	PIVOT-DYNAMIC	SINGLETONS	PIVOT
musae-facebook	0.97	1.13	1.00	1.19
email-Enron	0.95	1.08	1.00	1.25
cit-HepTh	1.00	1.20	1.00	1.22
ca-AstroPh	0.99	1.10	1.00	0.98

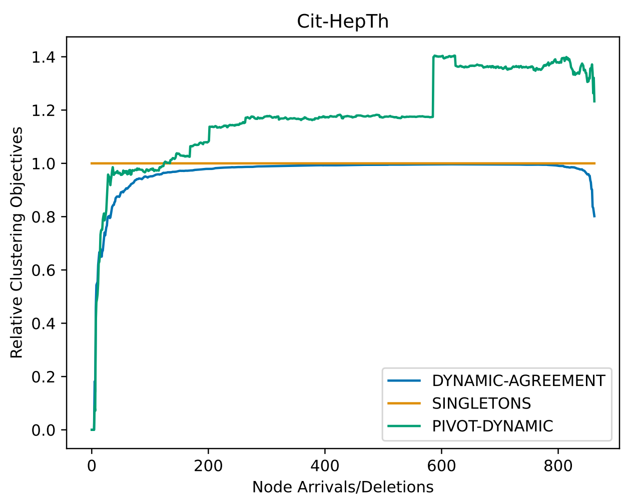

Figure 2. Correlation clustering objective relative to singletons

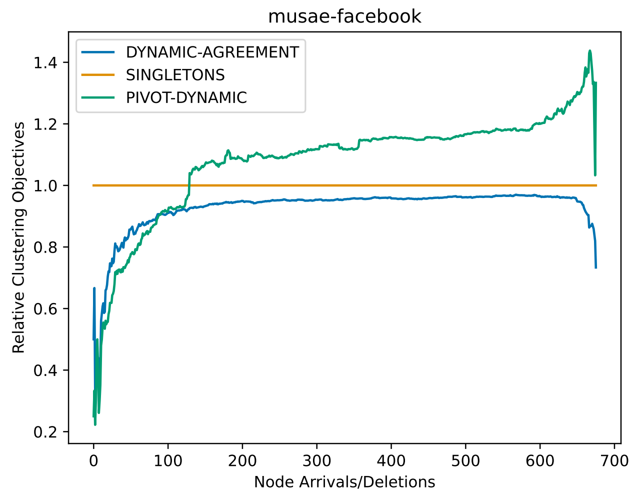

Figure 3. Correlation clustering objective relative to singletons

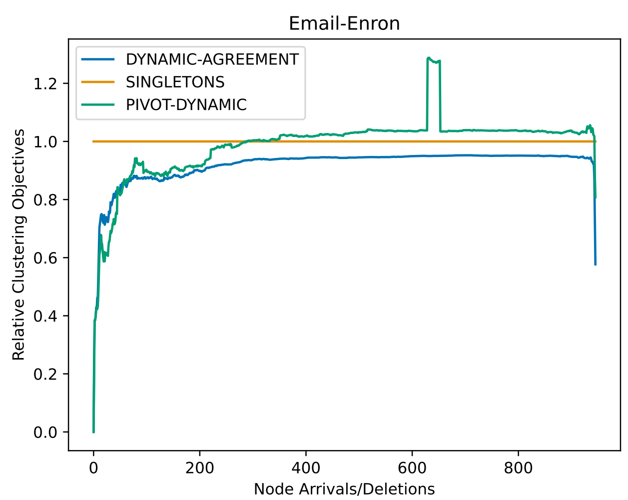

Figure 4. Correlation clustering objective relative to singletons