Tiered Sampling: An Efficient Method for Counting Sparse Motifs in Massive Graph Streams

LORENZO DE STEFANI, Brown University
ERISA TEROLLI, Max Planck Institute for Informatics
ELI UPFAL, Brown University

We introduce Tiered Sampling, a novel technique for estimating the count of sparse motifs in massive graphs whose edges are observed in a stream. Our technique requires only a single pass on the data and uses a memory of fixed size M, which can be magnitudes smaller than the number of edges.

Our methods address the challenging task of counting sparse motifs—sub-graph patterns—that have a low probability of appearing in a sample of M edges in the graph, which is the maximum amount of data available to the algorithms in each step. To obtain an unbiased and low variance estimate of the count, we partition the available memory into tiers (layers) of reservoir samples. While the base layer is a standard reservoir sample of edges, other layers are reservoir samples of sub-structures of the desired motif. By storing more frequent sub-structures of the motif, we increase the probability of detecting an occurrence of the sparse motif we are counting, thus decreasing the variance and error of the estimate.

While we focus on the designing and analysis of algorithms for counting 4-cliques, we present a method which allows generalizing Tiered Sampling to obtain high-quality estimates for the number of occurrence of any sub-graph of interest, while reducing the analysis effort due to specific properties of the pattern of interest.

We present a complete analytical analysis and extensive experimental evaluation of our proposed method using both synthetic and real-world data. Our results demonstrate the advantage of our method in obtaining high-quality approximations for the number of 4 and 5-cliques for large graphs using a very limited amount of memory, significantly outperforming the single edge sample approach for counting sparse motifs in large scale graphs.

CCS Concepts: • **Mathematics of computing** → *Graph enumeration*; • **Information systems** → *Data stream mining*; • **Human-centered computing** → *Social networks*; • **Theory of computation** → *Dynamic graph algorithms*;

Additional Key Words and Phrases: Graph motif mining, reservoir sampling, stream computing

ACM Reference format:
Lorenzo De Stefani, Erisa Terolli, and Eli Upfal. 2021. Tiered Sampling: An Efficient Method for Counting Sparse Motifs in Massive Graph Streams. *ACM Trans. Knowl. Discov. Data* 15, 5, Article 79 (May 2021), 52 pages.
https://doi.org/10.1145/3441299

Erisa Terolli part of this work done while visiting Brown University.
This work is supported in part by NSF awards RI-NSF 1813444 and CCF-NSF 1740741.
Authors' addresses: L. De Stefani and E. Upfal, Brown University, Department of Computer Science, 115 Waterman St, Providence, RI 02906; emails: {lorenzo, eli}@cs.brown.edu; E. Terolli, Max Planck Institute for Informatics, Stuhlsatzenhausweg 4, Saarbrücken, 66123, Germany; email: eterolli@mpi-inf.mpg.de.

1556-4681/2021/05-ART79 $15.00
https://doi.org/10.1145/3441299

1 INTRODUCTION

Counting motifs (sub-graphs with a given pattern) in large graphs is a fundamental primitive in graph mining with numerous practical applications including link prediction and recommendation [21], community detection [7], topic mining [13], spam and anomaly detection [6, 14, 22], protein interaction networks analysis [24], and analysis of temporal patterns [23].

Computing the exact count of motifs in massive, Web-scale networks is often impractical or even infeasible. Furthermore, many interesting networks, such as social networks, are continuously growing. Hence, there is limited value in maintaining an exact count. Rather, the goal is to have, *at any time*, a high-quality approximation of the quantity of interest. To obtain a scalable and efficient solution for massive size graphs, we focus on the well-studied model of one-pass stream computing. Our algorithms use a memory of fixed size M, where M is significantly smaller than the size of the input graph. The input graph is observed as a *stream* of edges in an *arbitrary* order, and the algorithm has only one pass on the input. The goal of the algorithm is to compute, at any given time, an unbiased and low-variance estimate of the count of motif occurrences in the graph observed up to that time.

Given its theoretical and practical importance, the problem of counting motifs in graph streams has received a lot of attention in the literature, with particular emphasis on the approximation of the number of 3-cliques (triangles) [20, 30, 33]. A standard approach to this problem is to sample up to M edges uniformly at random, using a fixed sampling probability or, more efficiently, reservoir sampling. A count of the number of motifs in the sample, extrapolated (normalized) appropriately, gives an unbiased estimate for the number of occurrences in the entire graph. The variance (and error) of this method depends on the expected number of occurrences in the sample. In particular, for sparse motifs that are unlikely to appear many times in the sample, this method exhibits high variance (higher than the actual count), which makes it useless for counting. Note that when the input graph is significantly larger than the memory size M, a motif that is unlikely to appear in a random sample of M edges may still have a large count in the graph. Also, as we attempt to count larger structures than triangles, these structures are more likely to be sparse in the graph. It is therefore important to obtain efficient methods for counting *sparser* motifs in massive-scale graph streams.

In this work, we introduce the concept of Tiered Sampling in stream computing. To obtain an unbiased and low variance estimate for the amount of sparse motif in massive-scale graphs, we partition the available memory to tiers (layers) of reservoir samples. The base tier is a standard reservoir sample of individual edges, while other tiers are reservoir samples of sub-structures of the desired motif. This strategy significantly improves the probability of detecting occurrences of the motif.

Assume that we count motifs with k edges. If all the available memory is used to store a sample of the edges, we would need $k-1$ of the motif's edges to be in the sample when the last edge of the motif is observed on the stream. The probability of this event decreases exponentially in k. Assume now that we use part of the available memory to store a sample of the observed occurrences of a fixed "*prototype*" sub-motif with $k/2$ edges. We are more likely to observe such motifs (we only need $k/2-1$ of their edges to be in the edge sample when the last edge is observed in the stream), and they are more likely to stay in the second reservoir sample since they are still relatively sparse. We now observe a full motif when the current edge in the stream completes an occurrence of the motif with edges and with a prototype sub-motifs in the two reservoirs. For an appropriate choice of parameters and with fixed total memory size, this event has a significantly higher probability than observing the $k-1$ edges in the edge sample. To obtain an unbiased estimate of the count, the number of observed occurrences needs to be carefully normalized by the probabilities of observing each of the components.

In this work, we make the following main contributions:

- —We introduce the *Tiered Sampling* framework for counting sparse motifs in large-scale graph streams using multi-layer reservoir samples.
- —We develop and fully analyze two algorithms for counting the number of 4-cliques in a graph using two-tier reservoir sampling.
- —For the purpose of comparison, we analyze a standard (single-tier) reservoir sample algorithm for counting 4-cliques.
- —We verify the advantage of our multi-layer algorithms by analytically comparing their performance to a standard (single-tier) reservoir sample algorithm for 4-cliques counting problem on random Barábasi–Albert graphs.
- —We develop the ATS4C technique, which allows to adaptively adjust the sub-division of memory space among the two tiers according to the properties of the graph being considered.
- —We conduct an extensive experimental evaluation of our algorithms for counting 4-cliques on massive graphs with up to hundreds of millions of edges. We show the quality of the achieved estimations by comparing them with the actual ground truth value. Our algorithms are also extremely scalable, showing update times in the order of hundreds of microseconds for graphs with billions of edges. Further, we show that our methods consistently outperform alternative approaches based on edge sampling.
- —We present a generalization of the Tiered Sampling approach to count any arbitrary sub-graph of interest. We show how to greatly simplify the analysis while retaining high-quality estimates. As an example, we obtain TS5C, which estimates the number of 5-cliques in a graph stream using an edge reservoir sample and a second reservoir sample of the 4-cliques observed on the stream.

Paper organization: In Section 2, we introduce the notation used in the presentation and some fundamental concepts used in this work. In Section 3, we discuss methods and results related to our work from the literature. In Section 4, we introduce the Tiered Sampling approach and its application to the problem of counting 4-cliques in a graph steam. In Section 4.1 (resp., Section 4.2), we present algorithm TS4C_1 (resp., TS4C_2) and we study its statistical properties. In Section 5, we delve into the comparison of the methods based on the Tiered Sampling approach with methods for counting 4-cliques using a single reservoir sample of edges observed on the stream, such as algorithm FourEst which we present and analyze in Section 5.1. We compare analytically the variance of the proposed methods (Section 5.2), and their performance for random Barábasi–Albert graphs [5] (Section 5.3). In Section 6, we introduce a variation of the previous Tiered Sampling algorithms, in which the partition of the memory among the two samples (*tiers*) being used can be adjusted dynamically to adapt to the properties of the graph of interest. We showcase the quality of the estimators provide by Tiered Sampling algorithms and their benefit with respect to edge-sampling approaches in Section 7 via extensive experimental evaluation on many real-world graphs of size ranging from millions to several hundred million edges. Finally, in Section 8, we discuss how to generalize the Tiered Sampling to any sub-graph of interest. In particular, we show how an opportune simplified analysis may ease such generalization by reducing the effort of the analysis. As an example, we present and evaluate algorithm TS5C, which yields an estimate of the number of 5-cliques in a graph stream.

2 PRELIMINARIES

For any (discrete) time step $t \geq 0$, we denote the graph observed up to and including time t as $G^{(t)} = (V^{(t)}, E^{(t)})$, where $V^{(t)}$ (resp., $E^{(t)}$) denotes the set of vertices (resp., edges) of $G^{(t)}$. At time

$t = 0$ we have $V^{(t)} = E^{(t)} = \emptyset$. For any $t > 0$, at time $t + 1$, we receive one single edge $e_{t+1} = (u, v)$ from a stream, where u, v are two distinct vertices. $G^{(t+1)}$ is thus obtained by *inserting the new edge*: $E^{(t+1)} = E^{(t)} \cup \{(u, v)\}$; if either u or v do not belong to $V^{(t)}$, they are added to $V^{(t+1)}$. Edges can be added just once (we discuss a generalization for *multigraphs* at the end of Section 2) in an arbitrary adversarial order, i.e., as to cause the worst outcome for the algorithm. However, we assume that the adversary has *no access to the random bits* used by the algorithm.

This work explores the idea of storing a sample of *prototype* sub-motifs in order to enhance the count of a sparse motif. For concreteness we focus on estimating the counts of 4-cliques and 5-cliques. Given a graph $G^{(t)} = (V^{(t)}, E^{(t)})$, a *k-clique* in $G^{(t)}$ is a *set* of $\binom{k}{2}$ (distinct) edges connecting a set of k (distinct) vertices.

Problem definition. We study the 4-Clique Counting Problem in Graph Edges Streams, which requires to compute, at each time $t \geq 0$ an estimation of $|C_k^{((t))}|$.

We denote by $C_k^{(t)}$ the set of *all* k-cliques in $G^{(t)}$.

Reservoir Sampling: Our work makes use of the *reservoir sampling scheme* [34]. Consider a stream of elements e_i observed in discretized time steps. Given a fixed sample size $M > 0$, for any time step t the reservoir sampling scheme allows to maintain a uniform sample $\mathcal{S}$ of size $\min\{M, t\}$ of the t elements observed on the stream:

- If $t \leq M$, then the element $e_t = (u, v)$ on the stream at time t is deterministically inserted in $\mathcal{S}$.
- If $t > M$, then the sampling mechanism flips a biased coin with heads probability M/t. If the outcome is "*heads*", it chooses an element e_i uniformly at random from those currently in $\mathcal{S}$ to be replaced by e_t. Otherwise, $\mathcal{S}$ is not modified.

When using reservoir sampling for estimating the number of occurrences of a sub-graph of interest, it is necessary to compute the probability of multiple edges elements being in $\mathcal{S}$ at the same time.

Lemma 2.1 (Lemma 4.1 [33]). *For any time step t and any positive integer $k \leq t$, let B be any subset of size $|B| = k \leq \min\{M, t\}$ of the element observed on the stream. Then,* at the end *of time step t (i.e., after updating the sample at time t), we have* $\Pr(B \subseteq \mathcal{S}) = 1$ *if* $t \leq M$, *and* $\Pr(B \subseteq \mathcal{S}) = \prod_{i=0}^{k-1} \frac{M-i}{t-i}$ *otherwise.*

Evaluation: In our experimental analysis (Sections 5.3 and 7), we measure the accuracy of the obtained estimator *through the evolution of the graph* in terms of their ***Mean Average Percentage Error* (MAPE)** [16]. The MAPE measures the relative error of an estimator (in this case, $\varkappa^{(t)}$) with respect to the ground truth (in this case, $|C_k^{(t)}|$) averaged over t time steps, that is:

$$MAPE = \frac{1}{t} \sum_{i=1}^{t} \frac{|\varkappa^{(t)} - |C_k^{(t)}||}{|C_k^{(t)}|.}$$

Multigraphs: Our approach can be extended to count the number of subgraphs of a *multigraph* represented as a stream of edges. Using a formalization analogous to that discussed for graphs, for any (discrete) time instant $t \geq 0$, let $G^{(t)} = (V^{(t)}, \mathcal{E}^{(t)})$ be the multigraph observed up to and including time t, where $\mathcal{E}^{(t)}$ is now a *bag* of edges between vertices of $V^{(t)}$. The multigraph evolves through a series of edge additions according to a process similar to the one described for graphs. The definition of the occurrence of a sub-graph (e.g., a 3, 4, or 5-clique) in a multigraph is the same as in a graph. As before, we denote with $C_k^{(t)}$ the set of *all* k-cliques in $G^{(t)}$, but now this set may contain multiple k-cliques with the same set of vertices, although each of these triangles

will be a different set of edges among those vertices, i.e., a subset of the bag $\mathcal{E}^{(t)}$ which differ by at least one element. The problem of 4-clique counting in multigraph edge streams is defined exactly in the same way as for graph edge streams. For the sake of simplicity, in the remainder of the presentation we focus on the analysis of graph edges streams.

3 RELATED WORK

Counting subgraphs in large networks is a well-studied problem in data mining which was originally brought to attention in the seminal work [24] on the analysis of protein interaction networks. In particular, many contributions in the literature have focused on the *triangle counting problem*, that is, the counting of the number of 3-cliques, including exact algorithms, MapReduce algorithms [27, 29], and streaming algorithms [1, 19, 30, 33].

Previous works in the literature on counting graph motifs [4, 31] can also be used to estimate the number of cliques in large graphs. Other recent works on *graphlets* (i.e., small subgraphs) counting introduced randomized [17, 28] and MapReduce [15] algorithms. However, these require prior information on the graph such as its degeneracy (for [17]) or the vertex degree ordering (for [15]) or the vertex degrees [28]. In [10], Bressan *et al.* present and compare *Monte Carlo* and *Color Coding* approaches for obtaining estimates of the count of graphlets with five or more vertices in the static setting. These approaches are not, however, used in the streaming setting.

The idea of using sub-structures of a graph motif in order to improve the estimation of its frequency in a massive graph has been previously explored in literature. In [9], Bordino *et al.* proposed a data stream algorithm that estimates the number of occurrences of a given subgraph by sampling its "*prototypes*" (i.e., sub-structures). While this approach is shown to be effective in estimating the number of occurrences of motifs with three and four edges, it requires multiple passes through the graph stream and further knowledge on the properties of the graph. In [18], Seshadhri and Pinar presented an algorithm that effectively and efficiently approximates the frequencies of all 4-vertex subgraphs by sampling paths of length three. However, this algorithm requires prior knowledge of the degrees of all the vertices in the graph and cannot be used in the streaming setting. In [2], reservoir sampling is used to develop the "*graph priority sampling*" framework for counting subgraphs. In a recent work [17], Jain and Seshadhri propose a clique counting method based on Túran's Theorem that requires knowledge of the degeneracy of the observed graph. [11] show an application of the color-coding scheme for the static setting which uses *colorful trees* as *prototypes* to guide the sampling phase. Some alternative approaches focus on counting specific graphs such as *butterfly* sub-graphs in bipartite graph streams [32].

In this work, we present a sampling-based, one-pass algorithm for insertion only streams to approximate the global number of cliques found in large graphs. Furthermore, our algorithms do not require any further information on the properties of the observed graph.

Using a strategy similar to our Tiered Sampling approach, Seshadhri and Pinar propose in [19] a one-pass streaming algorithm for triangle counting. This algorithm uses a first reservoir sample for edges that are then used to generate a stream of *wedges* (i.e., paths of length two) stored in a second, dedicated, reservoir sample. This approach appears to be not worthwhile for triangle counting as it is consistently outperformed by a simpler strategy based on a single reservoir presented in [33]. This is due to the fact that, as in most large graphs of interest wedges are much more frequent than edges, it is generally not worth devoting a large fraction of the available memory space to maintaining wedges over edges.

In [3], a similar approach, based on the use of multiple reservoir sample levels, is used to develop a sampling-based streaming algorithm for approximate bipartite projection in streaming bipartite networks: first, they maintain a reservoir of sampled bipartite edges with sampling weights that

favor the selection of high similarity nodes. Second, arriving edges generate a stream of *similarity updates* based on their adjacency with the current sample. These updates are aggregated in a second reservoir sample-based stream to yield the final unbiased estimate.

This work extends and completes a preliminary version of results presented in [12].

4 TIERED SAMPLING APPLICATION TO 4-CLIQUE COUNTING

In this section, we present TS4C$_1$ and TS4C$_2$, two applications of our Tiered Sampling approach for counting the number of 4-cliques in an undirected graph observed as an edge stream. Rather than counting a 4-cliques *only* when the currently observed edge completes a 4-clique with five other edges maintained in an edge sample (similarly to what successfully done for 3-cliques in [33]), we increase the probability of observing a clique by using a 3-cliques (i.e., triangles) reservoir sample. Our two Tiered Sampling algorithms partition the available memory into two samples: an edges reservoir sample and a triangles reservoir sample. TS4C$_1$ attempts in each step to construct a 4-cliques using the currently observed edge, two edges from the edge reservoir sample and one triangle from the triangle reservoir sample. TS4C$_2$ attempts at each step to construct a 4-clique from the currently observed edge and two triangles from the triangle reservoir sample. That is, both algorithms use triangle sub-patterns of a 4-clique as *prototype* sub-graphs used to aid in the detection of an entire 4-clique. At each time step t, both algorithms maintain a *running estimation* $\varkappa^{(t)}$ of $|C_4^{(t)}|$. Clearly $\varkappa^{(0)} = 0$ and the estimator is increased every time a 4-clique is "*detected*" on the stream. The two algorithms also maintains a counter $\tau^{(t)}$ for the number of triangles observed in the stream up to time t. This value is used by the reservoir sampling scheme which manages the triangle reservoir.

4.1 Algorithm TS4C$_1$

Algorithm TS4C$_1$ maintains an *edges (resp., triangles) reservoir sample* $\mathcal{S}_e$ (resp., $\mathcal{S}_\Delta$) of *fixed* size M_e (resp., M_Δ). From Lemma 2.1, for any t the probability of any edge e (resp., triangle T) observed on the stream Σ (resp., observed by the TS4C$_1$) to be included in S_e (resp., S_Δ) is M_e/t (resp. $M_\Delta/\tau^{(t)}$). We denote as $\mathcal{S}_e^{(t)}$ (resp., $\mathcal{S}_\Delta^{(t)}$) the set of edges (resp., triangles) in $\mathcal{S}_e$ (resp., $\mathcal{S}_\Delta$) *before* any update to the sample(s) occurring at step t. We denote with $\mathcal{N}_u^{\mathcal{S}_e^{(t)}}$ the *neighborhood* of u with respect to the edges in $\mathcal{S}_e^{(t)}$, that is $\mathcal{N}_u^{\mathcal{S}_e^{(t)}} = \{v \in V^{\mathcal{S}_e^{(t)}} : (u,v) \in \mathcal{S}_e^{(t)}\}$.

Let $e_t = (u,v)$ be the edge observed on the stream at time t. At each step TS4C$_1$ executes three main tasks:

— *Estimation update:* TS4C$_1$ invokes the function Update 4-cliques to detect any 4-clique completed by (u,v) (see Figure 1 for an example). That is, the algorithm verifies whether in triangle reservoir $\mathcal{S}_\Delta$ there exists any triangle T which includes u (resp., v). Note that since each edge is observed just once and the edge (u,v) is being observed for the first time no triangle in $\mathcal{S}_\Delta$ can include both u and v. For any such triangle $T' = \{u,w,z\}$ (or $\{v,w,z\}$) the algorithm checks whether the edges (v,w) and (v,z) (resp., (u,w) and (u,z)) are currently in $\mathcal{S}_e$. When such conditions are meet, we say that a 4-clique is "*observed on the stream.*" The algorithm then uses ProbClique to compute the *exact* probability p of the observation based on the timestamps of all its edges. The estimator $\varkappa$ is then increased by $p^{-1}/2$.

— *Triangle sample update:* using UpdateTriangles the algorithm verifies whether the edge e_t completes any triangle with the edges in $\mathcal{S}_e^{(t)}$. If that is the case, we say that a new triangle T^* is *observed on the stream.* The counter τ is increased by one and the new triangle is a candidate for inclusion in $\mathcal{S}_\Delta$ with probability M_Δ/τ^t.

ALGORITHM 1: TS4C$_1$ - Tiered Sampling for 4-Clique counting

Input: Insertion-only edge stream Σ, integers M, M_Δ
$\mathcal{S}_e \leftarrow \emptyset$, $\mathcal{S}_\Delta \leftarrow \emptyset$, $t \leftarrow 0$, $t_\Delta \leftarrow 0$, $\sigma \leftarrow 0$
for each element (u, v) from Σ **do** ▷ Process each edge (u,v) coming from the stream in discretized timesteps
  $t \leftarrow t + 1$
  UPDATE4CLIQUES(u, v) ▷ Update the 4-clique estimator by considering the new 4-cliques closed by edge (u,v)
  UPDATETRIANGLES(u, v) ▷ Update the triangles reservoir with the new triangles formed by edge (u,v)
  SAMPLEEDGE$((u, v), t)$ ▷ Update the edges sample with the edge (u,v) according to RS scheme

function UPDATE4CLIQUES$((u, v), t)$
  for each triangle $(u, w, z) \in \mathcal{S}_\Delta$ **do** ▷ For each triangle with vertices u, w and z
    if $(v, w) \in \mathcal{S}_e \wedge (v, z) \in \mathcal{S}_e$ **then** ▷ If triangle (u, w, z) forms a 4-clique (u,v,w,z) with two edges in $\mathcal{S}_e$
      $p \leftarrow$ PROBCLIQUE$((u, w, z), (v, w), (v, z))$ ▷ Calculate probability of observing 4-clique (u,v,w,z)
      $\sigma \leftarrow \sigma + p^{-1}/2$ ▷ Update the 4-clique estimator

  for each triangle $(v, w, z) \in \mathcal{S}_\Delta$ **do** ▷ For each triangle with vertices v, w and z
    if $(u, w) \in \mathcal{S}_e \wedge (u, z) \in \mathcal{S}_e$ **then** ▷ In case triangle (v,w,z) forms a 4-clique (u,v,w,z) with two edges in $\mathcal{S}_e$
      $p \leftarrow$ PROBCLIQUE$((v, w, z), (u, w), (u, z))$ ▷ Calculate probability of observing 4-clique (u,v,w,z)
      $\sigma \leftarrow \sigma + p^{-1}/2$ ▷ Update the 4-clique estimator

function UPDATETRIANGLES$((u, v), t)$
  $\mathcal{N}^{\mathcal{S}}_{u,v} \leftarrow \mathcal{N}^{\mathcal{S}}_u \cap \mathcal{N}^{\mathcal{S}}_v$
  for each element w from $\mathcal{N}^{\mathcal{S}}_{u,v}$ **do** ▷ For each triangle (u,v,w)
    $t_\Delta \leftarrow t_\Delta + 1$ ▷ Increment the number of observed triangles
    SAMPLETRIANGLE(u, v, w) ▷ Update the triangles sample with the triangle (u,v,w) according to RS Scheme

— *Edge sample update:* the algorithm updates the edge sample $\mathcal{S}_e$ according to the Reservoir Sampling scheme described in Section 2.

Each time a 4-clique is observed on the stream, TS4C$_1$ uses PROBCLIQUE to compute the *exact probability* of the observation. Such computation is all but trivial as it is influenced by both the order according to which the edges of the 4-clique were observed on the stream and by the number of triangles observed on the stream $\tau^{(t)}$. The analysis proceeds using a (somehow tedious) analysis of *all* the 5! possible orderings of the first five edges of the clique observed on the stream. Before presenting the analysis for computing the *exact probability* of observing a 4-clique on the stream in Lemma 4.2, we introduce Lemma 4.1

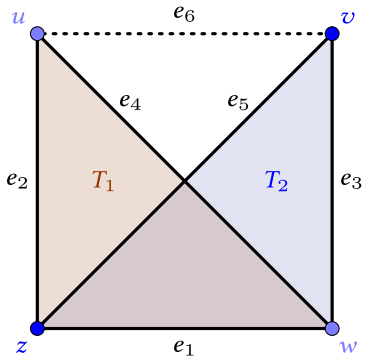


Fig. 1. Detection of a 4-clique using triangles.

Lemma 4.1. *Let $\lambda \in C_4^{(t)}$ with $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ using Figure 1 as reference. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. λ can be observed by TS4C$_1$ at time t_6 either as a combination of triangle $T_1 = \{e_1, e_2,$ or $e_4\}$ and edges $e_3 = (v, w)$ and $e_5 = (v, z)$, or as a combination of triangle $T_2 = \{e_1, e_3, e_5\}$ and edges $e_2 = (u, w)$ and $e_4 = (u, z)$.*

Proof. As presented in Algorithm 1, TS4C$_1$ can detect λ only when its last edge is observed on the stream (hence, t_6). When e_6 is observed, the algorithm first evaluates whether there is any triangle in $\mathcal{S}_\Delta^{(t_6)}$ that shares one of the two endpoint u or v from e_6. Since at this step (i.e., the execution of function Update4Cliques) the triangle sample is yet to be updated based on the observation of e_6, the only triangle sub-structures of λ which may have been observed on the stream, and thus included in $\mathcal{S}_\Delta^{(t_6)}$ are $T_1 = \{e_1, e_2, e_4\}$ and $T_2 = \{e_1, e_3, e_5\}$. If any of these is indeed in $\mathcal{S}_\Delta^{(t_6)}$, TS4C$_1$ proceeds to check whether the remaining two edges required to complete λ (resp., e_3, e_5 for T_1, or e_2, e_4 for T_2) are in $\mathcal{S}_e$. λ is thus observed either once or twice depending on which just one or both of these conditions are verified. □

Lemma 4.2. *Let $\lambda \in C_4^{(t)}$ with $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ using Figure 1 as reference. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. Let $t_{1,2,4} = \max\{t_1, t_2, t_4, M_e + 1\}$. The probability p_λ of λ being observed on the stream by TS4C$_1$ using the triangle $T_1 = \{e_1, e_2, e_4\}$ and the edges e_3, e_5, is computed by the* Prob-Clique *function as:*

$$p_\lambda = \frac{M_e}{t_{1,2,4}-1}\frac{M_e-1}{t_{1,2,4}-2}\min\left\{1, \frac{M_\Delta}{\tau^{(t_6)}}\right\}p',$$

where

$$p' = \begin{cases} 1 & \text{if } t_6 \leq M_e \\ \frac{M_e}{t_6-1}\frac{M_e-1}{t_6-2} & \text{if } \min\{t_3, t_5\} > t_{1,2,4} \\ \frac{M_e-1}{t_6-2}\frac{M_e-2}{t_{1,2,4}-3}\frac{t_{1,2,4}-1}{t_6-1} & \text{if } \max\{t_3, t_5\} > t_{1,2,4} > \min\{t_3, t_5\} \\ \frac{M_e-2}{t_{1,2,4}-3}\frac{M_e-3}{t_{1,2,4}-4}\frac{t_{1,2,4}-1}{t_6-1}\frac{t_{1,2,4}-2}{t_6-2} & \text{otherwise.} \end{cases}$$

Proof. Let us define the event $E_\lambda = \lambda$*is observed on the stream by TS4C$_1$ using triangle* $T_1 = \{e_1, e_2, e_4\}$ *and edges* e_3, e_5. Further let $E_{T_1} = T_1 \in \mathcal{S}_\Delta^{(t_6)}$, and $E_{3,5} = \{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)}$. Given the definition of TS4C$_1$ we have:

$$E_\lambda = E_{T_1} \wedge E_{3,5},$$

and hence:

$$p_\lambda = \Pr(E_\lambda) = \Pr(E_{T_1} \wedge E_{3,5}) = \Pr(E_{3,5}|E_{T_1}) \Pr(E_{T_1}).$$

In order to study $\Pr(E_{T_1})$ we shall introduce event $E_{S(T_1)}$ ="*triangle T_1 is observed on the stream by TS4C$_1$*." From the definition of TS4C$_1$, we know that T_1 is observed on the stream iff when the last edge of T_1 is observed on the stream at $\max\{t_1, t_2, t_4\}$ the remaining two edges are in the edge sample. Applying Bayes's rule of total probability we have:

$$\Pr(E_{T_1}) = \Pr\left(E_{T_1}|E_{S(T_1)}\right) \Pr\left(E_{S(T_1)}\right),$$

and thus:

$$p_\lambda = \Pr(E_{3,5}|E_{T_1}) \Pr\left(E_{T_1}|E_{S(T_1)}\right) \Pr\left(E_{S(T_1)}\right). \tag{1}$$

Let $t_{1,2,4} = \max\{t_1, t_2, t_4, M+1\}$, in order for T_1 to be observed by TS4C$_1$ it is required that when the last edge of T_1 is observed on the stream at $t_{1,2,4}$ its two remaining edges are kept in $\mathcal{S}_e$. From Lemma 2.1 we have:

$$\Pr\left(E_{S(T_1)}\right) = \frac{M_e}{t_{1,2,4}-1} \frac{M_e-1}{t_{1,2,4}-2}, \tag{2}$$

$$\Pr\left(E_{T_1}|E_{S(T_1)}\right) = \frac{M_e}{\tau^{t_6}}. \tag{3}$$

Let us now consider $\Pr(E_{3,5}|E_{T_1})$. While the content of $\mathcal{S}_\Delta$ itself does not influence the content of $\mathcal{S}_e$, the fact that T_1 is maintained in $\mathcal{S}_\Delta^{(t_6)}$ implies that is has been observed on the stream at a previous time and hence, that two of its edges have been maintained in $\mathcal{S}_e$ *at least* until the last of its edges has been observed on the stream. We thus have $\Pr(E_{3,5}|E_{T_1}) = \Pr(E_{3,5}|E_{S(T_1)})$. In order to study $p' = \Pr(E_{3,5}|E_{S(T_1)})$ it is necessary to distinguish the possible (5!) different arrival orders for edges e_1, e_2, e_3, e_4 and e_5. However in an efficient analysis we reduce the number of cases to be considered to just four:

- $t_6 \leq M + e$: in this case *all* the edges observed on the stream up until t_6 are deterministically inserted in $\mathcal{S}_e$ and thus $p' = 1$.
- $\min\{t_3, t_5\} > t_{1,2,4}$: in this cases both edges e_3 and e_5 are observed after *all* the edges composing T_1 have already been observed on the stream. As for any $t > t_{1,2,4}$ the event $E_{S(T_1)}$ does not imply that any of the edges of T_1 is *still* in $\mathcal{S}_e$ we have:

  $$p' = \Pr(E_{3,5}|E_{T_1}) = \Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)}\right),$$

  and, thus, from Lemma 2.1:

  $$p' = \frac{M_e}{t_6-1} \frac{M_e-1}{t_6-2}.$$

- $\max\{t_3, t_5\} > t_{1,2,4} > \min\{t_3, t_5\}$: in this case, only one of the edges e_3, e_5 is observed after all the edges in T_1 are observed. We need to therefore take into consideration that the two edges of T_1 are kept in S_e until $t_{1,2,4}$. Let $e_{3,5}^M$ (resp., $e_{3,5}^m$) denote the last (resp., first) edge observed on the stream between e_3 and e_5.

$$\begin{aligned} p' &= \Pr\left(e_{3,5}^M \in \mathcal{S}_e^{(t_6)} | e_{3,5}^m \in \mathcal{S}_e^{(t_6)}\right) \Pr\left(e_{3,5}^m \in \mathcal{S}_e^{(t_6)}\right), \\ &= \frac{M_e-1}{t_6-2} \Pr\left(e_{3,5}^m \in \mathcal{S}_e^{(t_6)} | e_{3,5}^m \in \mathcal{S}_e^{(t_{1,2,4})}\right) \Pr\left(e_{3,5}^m \in \mathcal{S}_e^{(t_{1,2,4})}\right), \\ &= \frac{M_e-1}{t_6-2} \frac{t_{1,2,4}-1}{t_6-1} \frac{M_e-2}{t_{1,2,4}-3}. \end{aligned}$$

— $t_{1,2,4} > \max\{t_2, t_4\}$: in all the remaining cases both e_3 and e_5 are observed before the last edge of T_1 has been observed. Hence:

$$p' = \Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | \{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_{1,2,4})}\right) \Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_{1,2,4})}\right),$$
$$= \frac{M_e - 2}{t_{1,2,4} - 3} \frac{M_e - 3}{t_{1,2,4} - 4} \frac{t_{1,2,4} - 1}{t_6 - 1} \frac{t_{1,2,4} - 2}{t_6 - 2}.$$

The lemma follows combining the result for the values of p' with Equations (2) and (3) in Equation (1). □

In our proofs, we carefully account for the fact that, as we use reservoir sampling [34], the presence of an edge (resp., of a triangle) in $\mathcal{S}_e$ (resp., $\mathcal{S}_\Delta$) is *not independent* from the concurrent presence of another edge (resp., triangle) in $\mathcal{S}_\Delta$. Further, we account for the fact that whether a triangle can be in $\mathcal{S}_\Delta$ *only if* it was previously detected by TS4C$_1$ (using the UpdateTriangles function from Algorithm 1), which itself is dependent on which edges are held in the sample when the last edge of the triangle itself is observed on the stream. In order to account for such dependencies, it is necessary to break down the order of arrival of the edges themselves.

The following corollary provides a lower bound to the probability according to which a 4-clique is observed by TS4C$_1$. Although generally loose, this result will provide simplification and insight in the analysis of the variance of the estimate obtained using TS4C$_1$.

Corollary 4.1. *Let $\lambda \in C_4^{(t)}$ defined as in the statement of Lemma 4.2. The probability p_λ of λ being observed on the stream by TS4C$_1$ using the triangle $T_1 = \{e_1, e_2, e_4\}$ and the edges e_3, e_5 is evaluated by the ProbClique function as:*

$$p_\lambda \geq \min\left\{1, \left(\frac{M_e}{t-1}\right)^4 \frac{M_\Delta}{\tau^{(t)}}\right\}.$$

Analysis of the TS4C$_1$ estimator. We now present the analysis of the estimations obtained using TS4C$_1$. First, we show their *unbiasedness*, and we then provide a bound on their variance. In the following, we denote as t_Δ the first time step at for which the number of triangles seen by TS4C$_1$ exceeds M_Δ.

Theorem 4.3. *The estimator $\varkappa$ returned by TS4C$_1$ is unbiased, that is $\varkappa^{(t)} = |C_k^{(t)}|$ if $t \leq \min\{M_e, t_\Delta\}$. Further, $\mathbb{E}[\varkappa^{(t)}] = |C_k^{(t)}|$ if $t \leq \min\{M_e, t_\Delta\}$.*

Proof of Theorem 4.3. From Lemma 4.1, we have that TS4C$_1$ can detect any 4-clique $\lambda \in C_4^{(t)}$ in exactly two ways: either using triangle T_1 and edges e_3, e_5, or by using triangle T_2 and edges e_2, e_4 (use Figure 1 as a reference).

For each $\lambda \in C_4^{(t)}$ let us consider the random variable δ_{λ_1} (resp. δ_{λ_2}) which takes value $p_{\lambda_1}^{-1}/2$ (resp., $p_{\lambda_2}^{-1}/2$) if the 4-clique λ is observed by TS4C$_1$ using triangle T_1 (resp., T_2) or zero otherwise. Let p_{λ_1} (resp., p_{λ_2}) denote the probability of such event: we then have $\mathrm{E}\left[\delta_{\lambda_1}\right] = \mathrm{E}\left[\delta_{\lambda_2}\right] = 1/2$.

From Lemma 4.2, we have that the estimator $\varkappa^{(t)}$ computed using TS4C$_1$ can be expressed as $\varkappa^{(t)} = \sum_{\lambda \in C_4^{(t)}} (\delta_{\lambda_1} + \delta_{\lambda_2})$. From the previous discussion and by applying linearity of expectation we thus have:

$$\mathrm{E}\left[\varkappa^{(t)}\right] = \mathrm{E}\left[\sum_{\lambda \in C_4^{(t)}} (\delta_{\lambda_1} + \delta_{\lambda_2})\right] = \sum_{\lambda \in C_4^{(t)}} (\mathrm{E}\left[\delta_{\lambda_1}\right] + \mathrm{E}\left[\delta_{\lambda_2}\right]) = \sum_{\lambda \in C_4^{(t)}} 1 = |C_4^{(t)}|.$$

Finally, let t^* denote the first step for which the number of triangles detected by TS4C$_1$ exceeds M_Δ. For $t \leq \min\{M_e, t^*\}$, the entire graph $G^{(t)}$ is maintained in $\mathcal{S}_e$ and all the triangles in $G^{(t)}$ are

stored in $\mathcal{S}_\Delta$. Hence all the cliques in $G^{(t)}$ are deterministically observed by TS4C$_1$ *in both ways* and we therefore have $\varkappa^{(t)} = |C_4^{(t)}|$. □

We now introduce an *upper bound* to the variance of the TS4C$_1$ estimations. We present here the most general result for $t \geq M_e, t_\Delta$. Note that if $t \leq \min\{M_e, t_\Delta\}$, from Theorem 4.3, $\varkappa^{(t)} = |C_k^{(t)}|$ and hence $\mathrm{Var}[\varkappa^{(t)}] = 0$. For $\min\{M_e, t_\Delta\} < t \leq \max\{M_e, t_\Delta\}$, $\mathrm{Var}[\varkappa^{(t)}]$ admits an upper bound similar to the one in Theorem 4.4.

Theorem 4.4. *For any time $t > \min\{M_e, t_\Delta\}$, he estimator $\varkappa$ returned by TS4C$_1$ satisfies*

$$\begin{aligned} Var\left[\varkappa^{(t)}\right] \leq |C_4^{(t)}| \left(c \left(\frac{t-1}{M_e}\right)^4 \left(\frac{\tau^{(t)}}{M_\Delta}\right) - 1 \right) + 2a^{(t)} \left(c \frac{t-1}{M_e} - 1 \right) \\ + 2b^{(t)} \left(c \left(\frac{t-1}{M_e}\right)^2 \left(\frac{1}{4}\frac{\tau^{(t)}}{M_\Delta} + \frac{3}{4}\frac{t-1}{M_e} \right) - 1 \right), \end{aligned} \tag{4}$$

where $a^{(t)}$ (resp., $b^{(t)}$) denotes the number of unordered pairs of 4-cliques which share one edge (resp., three edges) in $G^{(t)}$, and $c \geq \frac{M_e^3}{(M_e-1)(M_e-2)(M_e-3)}$.

Similarly to what done in the proof of Theorem 4.3, in the proof of our result on the bound of the variance of the estimate, we associate two random variables to each of the 4-cliques of the graph (one for each of the two possible ways of detecting such 4-clique). The proof relies on a careful analysis of the order of arrival of the edges shared between a pair of two 4-cliques in the stream (which we assume to be adversarial), as shown in Lemma A.1. When bounding the variance, we must consider not only pairs of 4-cliques that share edges, but also pairs of 4-cliques sharing no edges, since the respective presences of the parts used to detect them (i.e., edges and triangles) in the respective samples are not independent events.

In order to gain some insight on the variance bound in Theorem 4.4, note that the first, and dominant, term of (4) is given by the total number of 4-cliques in $G^{(t)}$ (i.e., $|C_4^{(t)}|$) multiplied by $(c(\frac{t-1}{M_e})^4(\frac{\tau^{(t)}}{M_\Delta}) - 1)$ which corresponds *approximately* to $c(p_\lambda)^{-1}$, where p_λ denotes the *lower bound* to the probability of TS4C$_1$ detecting a 4-clique as provided by Corollary 4.1. This suggest a natural relation between these quantities: as the probability of detecting a 4-clique decreases (resp., the total number of 4-cliques increases) the variance increases accordingly. For detailed proofs of this section we refer the reader to Appendix A.

Memory partition across layers. In most practical scenario we assume that a certain amount of total available memory M is available for algorithm TS4C$_1$. A natural question that arises is what is the best way of spitting the available memory between $\mathcal{S}_e$ and $\mathcal{S}_\Delta$. While different heuristics are possible, in our work we chose to assign the available space in such a way that the dominant first term of upper bound of TS4C$_1$ in Theorem 4.4 is minimized. For $0 < \alpha < 1$ let $M_e = \alpha M$ and $M_\Delta = (1-\alpha)M$. Then we have that $|C_4^{(t)}|(c(\frac{t-1}{\alpha M})^4(\frac{\tau^{(t)}-1}{(1-\alpha)M}) - 1)$ is minimized for $\alpha = 4/5$. This convenient splitting rule works well in most cases, and is used in most of the article. In Section 6, we discuss a more sophisticated *adaptive* dynamic allocation of the available memory among the two sample tiers, named, ATS4C. We present experimental results for this method in Section 7.2.

Concentration bound. Based on the bound on the variance in Theorem 4.4, we now show a concentration bound for the estimator $\varkappa^{(t)}$ returned by TS4C$_1$. The proof of Theorem 4.5 is based on the application of Chebyshev's inequality [26, Thm. 3.6].

Theorem 4.5. *Let $t > \min\{M_e, t_\Delta\}$ and assume $|C_4^{(t)}| > 0$. Let $a^{(t)}$ (resp., $b^{(t)}$) denote the number of unordered pairs of 4-cliques which share one edge (resp., three edges) in $G^{(t)}$, and*

$c \geq \frac{M_e^3}{(M_e-1)(M_e-2)(M_e-3)}$. *Further, let* $M_e = \alpha M$ *(resp.,* $M_e = (1-\alpha)\,M$*), for* $\alpha \in (0,1)$. *For any* $\varepsilon, \delta \in (0,1)$, *if*

$$M > \alpha^{-1} \max\left\{ \sqrt[5]{\frac{\alpha}{1-\alpha}\frac{3c(t-1)^4(\tau^{(t)})}{\delta\varepsilon^2|C_4^{(t)}|}}, \frac{6ca^{(t)}(t-1)}{\delta\varepsilon^2|C_4^{(t)}|^2}, \sqrt[3]{\frac{3cb^{(t)}c\,(t-1)^2\left(\alpha\tau^{(t)}+3(1-\alpha)(t-1)\right)}{2(1-\alpha)\delta\varepsilon^2|C_4^{(t)}|^2}} \right\},$$

then the estimator $\varkappa^{(t)}$ *returned by TS4C*$_1$ *satisfies:*

$$Pr\left(|\varkappa^{(t)} - |C_4^{(t)}| < \varepsilon|C_4^{(t)}|\right) > 1-\delta.$$

Proof. By Chebyshev's inequality it is sufficient to prove that

$$\frac{\mathrm{Var}\left[\varkappa^{(t)}\right]}{\varepsilon^2|C_4^{(t)}|^2} < \delta.$$

From Theorem 4.4, we can write:

$$\begin{aligned}
\frac{\mathrm{Var}\left[\tau^{(t)}\right]}{\varepsilon^2|C_4^{(t)}|^2} &\leq \frac{\left(|C_4^{(t)}|\left(c\left(\frac{t-1}{M_e}\right)^4\left(\frac{\tau^{(t)}}{M_\Delta}\right)-1\right)+2a^{(t)}\left(c\frac{t-1}{M_e}-1\right)+2b^{(t)}\left(c\left(\frac{t-1}{M_e}\right)^2\left(\frac{1}{4}\frac{\tau^{(t)}}{M_\Delta}+\frac{3}{4}\frac{t-1}{M_e}\right)-1\right)\right)}{\varepsilon^2|C_4^{(t)}|} \\
&< \frac{\left(|C_4^{(t)}|c\left(\frac{t-1}{M_e}\right)^4\left(\frac{\tau^{(t)}}{M_\Delta}\right)+2a^{(t)}\left(c\frac{t-1}{M_e}\right)+2b^{(t)}c\left(\frac{t-1}{M_e}\right)^2\left(\frac{1}{4}\frac{\tau^{(t)}}{M_\Delta}+\frac{3}{4}\frac{t-1}{M_e}\right)\right)}{\varepsilon^2|C_4^{(t)}|} \\
&= \frac{\left(|C_4^{(t)}|c\frac{(t-1)^4\tau^{(t)}}{\alpha^4(1-\alpha)M^5}+2a^{(t)}c\left(\frac{t-1}{\alpha M}\right)+2b^{(t)}c\left(\frac{t-1}{\alpha M}\right)^2\frac{\alpha\tau^{(t)}+3(1-\alpha)(t-1)}{4\alpha(1-\alpha)M}\right)}{\varepsilon^2|C_4^{(t)}|} \\
&= \frac{\left(|C_4^{(t)}|c\frac{(t-1)^4\tau^{(t)}}{\alpha^4(1-\alpha)M^5}+2a^{(t)}c\left(\frac{t-1}{\alpha M}\right)+2b^{(t)}c\frac{(t-1)^2\left(\alpha\tau^{(t)}+3(1-\alpha)(t-1)\right)}{4\alpha^3(1-\alpha)M^3}\right)}{\varepsilon^2|C_4^{(t)}|}.
\end{aligned}$$

Hence it is sufficient to impose the following three conditions:

Condition 1.

$$\frac{\delta}{3} > \frac{1}{\varepsilon^2|C_4^{(t)}|^2}|C_4^{(t)}|c\frac{(t-1)^4\,\tau^{(t)}}{\alpha^4\,(1-\alpha)\,M^5}\,\frac{\alpha}{1-\alpha}\frac{c\,(t-1)^4\,\tau^{(t)}}{\varepsilon^2|C_4^{(t)}|\alpha^5M^5},$$

which is verified for:

$$M > \alpha^{-1}\sqrt[5]{\frac{3\alpha}{1-\alpha}\frac{c\,(t-1)^4\,\tau^{(t)}}{\delta\varepsilon^2|C_4^{(t)}|}}.$$

Condition 2.

$$\frac{\delta}{3} > \frac{1}{\varepsilon^2|C_4^{(t)}|^2}2a^{(t)}c\left(\frac{t-1}{\alpha M}\right),$$

which is verified for:

$$M > \alpha^{-1}\frac{6a^{(t)}c\,(t-1)}{\delta\varepsilon^2|C_4^{(t)}|^2}.$$

ALGORITHM 2: TS4C$_2$ - Tiered Sampling for 4-Clique counting using 2 triangle sub-structures

function UPDATE4CLIQUES($(u, v), t$)
  for each $(u, w, z) \in \mathcal{S}_\Delta \wedge (v, w, z) \in \mathcal{S}_\Delta$ **do** ▷ For each 4-clique (u,v,w,z) formed by two triangles (u,w,z) and (v,w,z)
    $p \leftarrow$ PROBCLIQUE($(u, w, z), (v, w, z)$) ▷ Calculate the probability of observing 4-clique (u,v,w,z)
    $\sigma \leftarrow \sigma + p^{-1}$ ▷ Update the 4-clique estimator

Condition 3.

$$\frac{\delta}{3} > \frac{1}{\varepsilon^2 |C_4^{(t)}|^2} 2b^{(t)} c \frac{(t-1)^2 \left(\alpha\tau^{(t)} + 3(1-\alpha)(t-1)\right)}{4\alpha^3(1-\alpha)M^3},$$

which is verified for:

$$M > \alpha^{-1} \sqrt[3]{\frac{3b^{(t)} c (t-1)^2 \left(\alpha\tau^{(t)} + 3(1-\alpha)(t-1)\right)}{2\delta\varepsilon^2(1-\alpha)|C_4^{(t)}|^2}}.$$

The theorem follows. □

Note that for $t \leq \min\{M_e, t_\Delta\}$ all the edges (resp., triangles) observed up to time t can be maintained in $\mathcal{S}_e$ (resp., $\mathcal{S}_\Delta$), and, hence, we have $\varkappa^{(t)} = |C_k^{(t)}|$ and $\mathrm{Var}[\varkappa^{(t)}] = 0$.

Theorem 4.5, provides a bound on the relative ϵ-approximation of $|C_4^{(t)}|$ provided by TS4C$_1$. In particular, Theorem 4.5 suggest that while the variance of the estimator is bound to increase as $|C_4^{(t)}|$ increases (as stated in Theorem 4.4), for a given value δ, the size of available sample memory M required to achieve an ϵ-approximation for $|C_4^{(t)}|$ with probability at least $1 - \delta$ decreases for higher values of $|C_4^{(t)}|$.

4.2 Algorithm TS4C$_2$

While the version of TS4C$_1$ presented in Algorithm 1 detects 4-cliques by using one triangle sub-structure form $\mathcal{S}_\Delta$ and two edges from $\mathcal{S}_e$, it is possible to use different sub-structures to achieve the same goal. We now present a variation of TS4C$_1$, called TS4C$_2$, which detects a 4-clique when the current observed edge completes a 4-clique using *two triangles* currently in $\mathcal{S}_\Delta$.

TS4C$_1$ and TS4C$_2$ differ *only* in the *Estimation update* step, implemented by the function UPDATE4CLIQUES which determinates how the occurrences of 4-cliques are detected and the estimators are correspondingly updated. The pseudocode for TS4C$_2$ is presented in Algorithm 2:

- Whenever a new edge $e_t = (u, v)$ is observed on the stream at time T, TS4C$_2$ invokes the function UPDATE 4-CLIQUES (Algorithm 2) which verifies whether in triangle reservoir $\mathcal{S}_\Delta$ there exists any (unordered) pair of triangles $\{T_1, T_2\}$ such that $T_1 = \{u, w, z\}$ and $T_2 = \{v, w, z\}$, for $w, z \in V^{(t)}$. When such conditions are meet, we say that a 4-clique $\{u, v, w, z\}$ is "*observed on the stream*" by TS4C$_2$. The algorithm then uses PROBCLIQUE to compute the *exact* probability p of the observation based on the timestamps of all its edges according to the results of Lemma 4.7. The estimator $\varkappa$ is then increased by p^{-1}.
- TS4C$_2$ then proceeds in updating the triangle sample $\mathcal{S}_\Delta$ and the edge sample $\mathcal{S}_e$ following the same steps as TS4C$_1$ as described in Section 4.1.

The difference in the way 4-cliques are detected by TS4C$_2$ corresponds to a difference in the probability of such detections. Before introducing the lemma for calculating the probability of detecting a 4-clique using TS4C$_2$, we introduce the following technical Lemma 4.6 which states

that each 4-clique can be observed just once using TS4C$_1$. In order to simplify the presentation, in the following we use the following notation:

$$t^M_{1,2,\ldots,i} \triangleq \max\{t_1, t_2, \ldots, t_i\}$$
$$t^m_{1,2,\ldots,i} \triangleq \min\{t_1, t_2, \ldots, t_i\}$$

Lemma 4.6. *Let $\lambda \in C_4^{(t)}$ with $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ using Figure 1 as reference. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. λ can be observed by TS4C$_2$ at time t_6 only by a combination of two triangles $T_1 = \{e_1, e_2, e_4\}$ and $T_2 = \{e_1, e_3, e_5\}$.*

Proof. TS4C$_2$ can detect λ only when its last edge is observed on the stream (hence, t_6). When e_6 is observed, the algorithm first evaluates whether there are two triangles in $\mathcal{S}_\Delta^{(t_6)}$, that share two endpoints and the other endpoints are u and v respectively. Since at this step (i.e., the execution of function Update4Cliques) the triangle sample is jet to be updated based on the observation of e_6, the only triangle sub-structures of λ which may have been observed on the stream, and thus included in $\mathcal{S}_\Delta^{(t_6)}$ are $T_1 = \{e_1, e_2, e_4\}$ and $T_2 = \{e_1, e_3, e_5\}$. λ is thus observed if and only if both of the triangles T_1 and T_2 are found in $\mathcal{S}_\Delta^{(t_6)}$. □

Lemma 4.6 marks an important difference between TS4C$_2$ and TS4C$_1$ as for the latter there are multiple ways of detecting the same 4-clique as discussed in Section 4. Such difference impacts the analysis of the probability of detecting a 4-clique using TS4C$_2$:

Lemma 4.7. *Let $\lambda \in C_4^{(t)}$ with $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ using Figure 1 as reference. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. The probability p_λ of λ being observed on the stream by TS4C$_2$, is computed by ProbClique as:*

$$p_\lambda = \min\left\{1, \frac{M_e}{t^M_{1,3,5} - 1}\frac{M_e - 1}{t^M_{1,3,5} - 2}\right\} \min\left\{1, \frac{M_\Delta}{\tau^{(t_6)}}\frac{M_\Delta - 1}{\tau^{(t_6)} - 1}\right\} p',$$

where:

$$p' = \begin{cases} 1, & if\ t^M_{1,2,4} \leq M_e \\ \frac{M_e-2}{t_1-3}\frac{M_e-3}{t_1-4}, & if\ t_1 > t^M_{2,3,4,5} \\ \frac{M_e-2}{t_1-3}\frac{M_e-3}{t_1-4}, & if\ t^M_{3,5} > t_1 > \max\{t^m_{3,5}, t_2, t_4\} \\ \frac{M_e-2}{t^M_{2,4}-3}, & if\ t^M_{3,5} > t^M_{2,4} > \max\{t^m_{3,5}, t^m_{2,4}, t_1\} \\ \frac{M_e}{t_1-1}\frac{M_e-1}{t_1-2}, & if\ t^m_{3,5} > t_1 > t^M_{2,4} \\ \frac{M_e-1}{t^M_{2,4}-2} & if\ t^m_{3,5} > t^M_{2,4} > t_1 \\ \frac{M_e-1}{t^M_{2,4}-1}\frac{M_e-2}{t_1-3}\frac{t_1-1}{t_2-1}, & if\ t^M_{2,4} > t_1 > \max\{M_e, t^m_{2,4}, t^M_{3,5}\} \\ \frac{M_e-1}{t^M_{2,4}-1}\frac{M_e}{t_2-1} & if\ t^M_{2,4} > M_e > t_1 > \max\{t^m_{2,4}, t^M_{3,5}\} \\ \frac{t_3-1}{t^M_{2,4}-1}\frac{t_3-2}{t^M_{2,4}-2}\frac{M_e-2}{t^M_{3,5}-3}, & if\ t^M_{2,4} > t^M_{3,5} > \max\{M_e, t^m_{3,5}, t^m_{2,4}, t_1\} \\ \frac{M_e}{t^M_{2,4}-1}\frac{M_e-1}{t^M_{2,4}-2}, & if\ t^M_{2,4} > M_e > t^M_{3,5} > \max\{t^m_{3,5}, t^m_{2,4}, t_1\} \\ \frac{M_e}{t^M_{2,4}-1}\frac{M_e-1}{t^M_{2,4}-2}, & if\ t^m_{2,4} > t_1 > t^M_{3,5} \\ \frac{M_e-1}{t^M_{2,4}-2}\frac{t^M_{3,5}-1}{t^M_{2,4}-1}, & if\ t^m_{2,4} > t^M_{3,5} > \max\{M, t_1\} \\ \frac{M_e-1}{t^M_{2,4}-2}\frac{M_e}{t^M_{2,4}-1}, & if\ t^m_{2,4} > M_e > t^M_{3,5} > t_1 \end{cases}$$

The analysis presents several complications due to the interplay of the probabilities of observing each of the two triangles that share an edge. For a detailed proof of Lemma 4.2 and of the other results in this section (Theorems 4.8, 4.9, and 4.10), we refer the reader to Appendix B. The following corollary presents a *lower bound* to the probability of a 4-clique being observed by TS4C$_2$. While rather loose, especially for 4-cliques that are detected towards the beginning of the edge stream, it provides a useful intuition for the analysis of the variance of the estimator presented in Theorem 4.9.

Corollary 4.2. *Let $\lambda \in C_4^{(t)}$ and let p_λ denote the probability of λ being observed on the stream by TS4C$_2$. We have:*

$$p_\lambda \leq \left(\frac{M_e}{t-1}\right)^4 \left(\frac{M_\Delta}{\tau^{(t)}}\right)^2.$$

Analysis of TS4C$_2$ estimator

Theorem 4.8. *The estimator $\varkappa$ returned by TS4C$_2$ is unbiased, that is $\varkappa^{(t)} = |C_k^{(t)}|$ if $t \leq \min\{M_e, t_\Delta\}$. Further, $\mathbb{E}[\varkappa^{(t)}] = |C_k^{(t)}|$ if $t \leq \min\{M_e, t_\Delta\}$.*

The proof of 4.8 closely follows the steps of the proof of 4.3. The main differences are given by the fact that there is one unique possible way according to which 4-cliques are observed by TS4C$_2$ (Lemma 5.1), and the probability of such event is given by 4.6.

The analysis of the variance of TS4C$_2$ presents considerable differences with respect to the analysis of the variance of TS4C$_1$, which are due to the different strategies that the two algorithms use to detect 4-cliques. The key component of the analysis depends on the analysis of the covariance of all the possible pairs of the binary random variables that are each associated with the detection of one of the 4-cliques. As stated in Lemma 4.6, for TS4C$_2$ there is just a single possible way according to which a 4-clique can be detected. Thus, the number and the distribution of the random variables considered in the analysis of the variance of TS4C$_2$ are clearly different from those of the random variables considered in the previous section for the analysis of the variance of TS4C$_1$.

Theorem 4.9. *For any time $t > \min\{M_e, t_\Delta\}$, the estimator $\varkappa$ returned by TS4C$_2$ satisfies*

$$\begin{aligned} Var\left[\varkappa^{(t)}\right] \leq & |C_4^{(t)}| \left(c\left(\frac{t-1}{M_e}\right)^4 \left(\frac{\tau^{(t)}}{M_\Delta}\right)^2 - 1\right) + 2a^{(t)}\left(c\frac{t-1}{M_e} - 1\right) \\ & + 2b^{(t)}\left(c\left(\frac{t-1}{M_e}\right)^2 \left(\frac{1}{4}\frac{\tau^{(t)}}{M_\Delta} + \frac{3}{4}\frac{t-1}{M_e}\right) - 1\right), \end{aligned} \tag{5}$$

where $a^{(t)}$ (resp., $b^{(t)}$) denotes the number of unordered pairs of 4-cliques which share one edge (resp., three edges) in $G^{(t)}$, and $c \geq \frac{M_e^3}{(M_e-1)(M_e-2)(M_e-3)}$.

Note that the first, and dominant, term of Equation (5) is given by the total number of 4-cliques in $G^{(t)}$ (i.e., $|C_4^{(t)}|$) multiplied by $(c(\frac{t-1}{M_e})^4(\frac{\tau^{(t)}}{M_\Delta})^2 - 1)$ which corresponds *approximately* to $c(p_\lambda)^{-1}$, where p_λ denotes the *lower bound* to the probability of TS4C$_2$ detecting a 4-clique as provided by Corollary 4.2. This suggests a natural relation between these quantities: as the probability of detecting a 4-clique decreases (resp., the total number of 4-cliques increases), the variance increases accordingly. The bound on the variance of TS4C$_2$ is similar to the corresponding result for TS4C$_1$ in Theorem 4.9. While it appears that the leading term of the variance for TS4C$_2$ is higher than that those of TS4C$_2$ by a $\frac{\tau^{(t)}}{M_\Delta}$ factor, it should be noted that this is mostly a result of the simplification of the analysis which is, however, required to achieve a closed-form expression for the variance. In our experimental analysis on random (Section 5.3) and real-world graphs (Section 7), we observe

that TS4C$_1$ and TS4C$_2$ generally exhibit comparable performance in terms of the quality of the produced estimated and their variance.

Memory partition: Following the same criterion discussed in Section 4.1, we use $|M_e| = 2M/3$ and $|M_\Delta| = M/3$ as a general rule for assigning the available memory space between the two sample levels. Given the fact that there is a much higher emphasis on the role and the use of the triangle sub-patterns in TS4C$_2$ compared to TS4C$_1$, it should not be surprising that the preferred memory split for the former assigned a higher fraction of the available memory space to the triangle sample compared to the latter. We use this assignment in the remainder of the article and for the experimental evaluation of the performance of our algorithm.

Concentration bound: Based on the bound on the variance in Theorem 4.9, we now show a concentration bound for the estimator $\varkappa^{(t)}$ returned by TS4C$_2$ based on the application of Chebyshev's inequality [26, Thm. 3.6].

Theorem 4.10. *Let $t > \min\{M_e, t_\Delta\}$ and assume $|C_4^{(t)}| > 0$. Let $a^{(t)}$ (resp., $b^{(t)}$) denote the number of unordered pairs of 4-cliques which share one edge (resp., three edges) in $G^{(t)}$, and $c \geq \frac{M_e^3}{(M_e-1)(M_e-2)(M_e-3)}$. Further, let $M_e = \alpha M$ (resp., $M_e = (1-\alpha)\,M$), for $\alpha \in (0,1)$. For any $\varepsilon, \delta \in (0,1)$, if*

$$M > \alpha^{-1} \max \left\{ \sqrt[3]{\frac{\alpha}{1-\alpha} \frac{3c(t-1)^2(\tau^{(t)})^2}{\delta\varepsilon^2|C_4^{(t)}|}}, \frac{6ca^{(t)}(t-1)}{\delta\varepsilon^2|C_4^{(t)}|^2}, \sqrt[3]{\frac{6b^{(t)}c\,(t-1)^2\left(\alpha\tau^{(t)} + 3(1-\alpha)(t-1)\right)}{4(1-\alpha)\delta\varepsilon^2|C_4^{(t)}|^2}} \right\}$$

then the estimator $\varkappa^{(t)}$ returned by TS4C$_2$ satisfies

$$Pr\left(|\varkappa^{(t)} - |C_4^{(t)}| < \varepsilon|C_4^{(t)}|\right) > 1 - \delta.$$

As for Theorem 4.5, note that for $t \leq \min\{M_e, t_\Delta\}$ all the edges (resp., triangles) observed up to time t can be maintained in $\mathcal{S}_e$ (resp., $\mathcal{S}_\Delta$), and, hence, we have $\varkappa^{(t)} = |C_k^{(t)}|$ and $\mathrm{Var}[\varkappa^{(t)}] = 0$. The proof for Theorem 4.10 follows steps analogous to those discussed for Theorem 4.5.

Although the difference between TS4C$_1$ and TS4C$_2$ may appear of minor interest, our experimental analysis shows that it can lead to significantly different performances depending on the properties of the graph $G^{(t)}$. Intuitively, TS4C$_2$ emphasizes the importance of the triangle substructures compared to TS4C$_1$, thus resulting in better performance when the input graph is very sparse with a low number of occurrences of 3 and 4-cliques.

5 COMPARISON WITH SINGLE SAMPLE APPROACH

Given a certain fixed amount M of available memory space, our Tiered Sampling approach suggest that the user *allocates* such memory into multiple tiers of reservoir samples in order to exploit the sparsity of the sub-structures of the motif of interest. However, it is only natural to wonder how this approach compares to an alternative, somewhat simpler, strategy that maintains a *single* sample of edges and that relies *only* on the edges in the sample to detect occurrences of the motif in $G^{(t)}$.

To quantify the advantage of our Tiered Sampling approach, we construct and fully analyze algorithm FourEst that uses a single edges sample strategy. We thoroughly compare the performance achieved by TS4C$_1$ with the performances of an algorithm, named FourEst, that uses a single reservoir sample strategy. We analyze the estimator provided by FourEst compare the analytical bound on the variance of the estimator, and we then compare their performance on both synthetic and real-world data (in Section 7).

ALGORITHM 3: FourEst

Input: Edge stream Σ, integer $M \geq 6$
Output: Estimation of the number of 4-cliques $\varkappa$
$\mathcal{S}_e \leftarrow \emptyset$, $t \leftarrow 0$, $\varkappa \leftarrow 0$
for each element (u, v) from Σ **do** ▷ Process each edge (u,v) coming from the stream in discretized timesteps
  $t \leftarrow t + 1$
  Update4Cliques(u, v) ▷ Update the 4-clique estimator by considering the new 4-cliques closed by edge (u,v)
  SampleEdge$((u, v), t)$ ▷ Update edge sample with (u,v) according to RS scheme

function Update4Cliques$((u, v), t)$
  $\mathcal{N}_{u,v}^{\mathcal{S}} \leftarrow \mathcal{N}_u^{\mathcal{S}} \cap \mathcal{N}_v^{\mathcal{S}}$
  for each element (x, w) from $\mathcal{N}_{u,v}^{\mathcal{S}} \times \mathcal{N}_{u,v}^{\mathcal{S}}$ **do** ▷ For each 4-clique formed by (u, v) and 5 edges in the S_e
    if (x, w) in $\mathcal{S}_e$ **then** ▷ Calculate the probability of observing the new formed 4-clique
      if $t \leq M$ **then**
        $p \leftarrow 1$
      else
        $p \leftarrow min\{1, \frac{M(M-1)(M-2)(M-3)(M-4)}{(t-1)(t-2)(t-3)(t-4)(t-5)}\}$
      $\varkappa \leftarrow \varkappa + p^{-1}$ ▷ Increase 4-clique estimator by the inverse of the probability of observing the new 4-clique

5.1 Edge Sampling Approach—FourEst

FourEst (FOUR clique ESTimation) maintains a uniform random sample $\mathcal{S}$ of size M of the edges observed over the stream using the reservoir sampling scheme, in order to estimate the number of four cliques in $G^{(t)}$. This algorithm is a natural extension of the technique discussed in [33] for triangle counting. We refer the reader to Appendix C for the complete proofs of the results in this section.

At each time step t, FourEst maintains a *running estimation* $\varkappa^{(t)}$ of $|C_4^{(t)}|$. Clearly $\varkappa^{(0)} = 0$. Every time a new edge $e_t = (u, v)$ is observed on the stream, FourEst verifies whether $e_{(t)}$ completes any 4-cliques with the edges currently in $\mathcal{S}_e^{(t)}$. If that is the case, the estimator $\varkappa$ is increased by the reciprocal of the probability $p = \min\{1, \prod_{i=0}^{4} \frac{M-i}{t-1-i}\}$ of observing that same 4-clique (from Lemma 2.1) using FourEst. Finally, the algorithm updates $\mathcal{S}_e$ according to the reservoir sampling scheme discussed in Section 2. The pseudocode for FourEst is presented in Algorithm 3.

Analysis of the FourEst estimator: Before presenting the proof of the *unbiasedness* of the estimations obtained using FourEst in Theorem 5.2, we introduce Lemma 5.1 which characterizes the probability of a 4-clique being observed by algorithm FourEst. The proofs of the following results share the structure of the proofs of corresponding results for TS4C$_1$ and TS4C$_2$. We refer the reader to appendix C for the complete presentation of the proofs.

Lemma 5.1. *Let $\lambda \in C_4^{(t)}$ with $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. Assume, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. λ is observed*

by FourEst at time t_6 with probability:

$$p_\lambda = \begin{cases} 0 & if\ |M| < 5, \\ 1 & if\ t_6 \leq M+1, \\ \prod_{i=0}^{5} \frac{M-i}{t-i-1} & if\ t_6 > M+1. \end{cases} \tag{6}$$

Proof. Clearly FourEst can observe λ *only* at the time step at which the last edge e_6 is observed on the stream at t_6. Further, from its construction, FourEst observed a 4-clique λ if and only if when its large edge is observed on the stream its remaining five edges are kept in the edge reservoir $\mathcal{S}$. $\mathcal{S}$ is a uniform edge sample maintained by means of the reservoir sampling scheme. From Lemma 2.1 we have that the probability of any five elements observed on the stream *prior* to t_6 being in $\mathcal{S}$ at the beginning of step t_6 is given by:

$$\Pr\left(\{e_1, e_2, e_3, e_4, e_5, e_6\} \subseteq\right) = \begin{cases} 0 & if\ |M| < 5, \\ 1 & if\ t_6 \leq M+1, \\ \prod_{i=0}^{4} \frac{M-i}{t-i-1} & if\ t_6 > M+1. \end{cases}$$

The lemma follows. □

As for the Tiered Sampling algorithms, the estimator obtained using FourEst is unbiased.

Theorem 5.2. *Let $\varkappa^{(t)}$ the estimated number of 4-cliques in $G^{(t)}$ computed by FourEst using memory of size M. $\varkappa^{(t)} = |C_4^{(t)}|$ if $t \leq M+1$ and $\mathbb{E}[\varkappa^{(t)}] = |C_4^{(t)}|$ if $t > M+1$.*

We now show an *upper bound* to the variance of the FourEst estimations for $t > M$ (for $t \leq M$ we have $\varkappa^{(t)} = |C_4^{(t)}|$ and thus the variance of $\varkappa^{(t)}$ is zero), and a corresponding concentration bound.

Theorem 5.3. *For any time $t > M+1$, we have*

$$Var\left[\varkappa^{(t)}\right] \leq |C_4^{(t)}| \left(\left(\frac{t-1}{M}\right)^5 - 1\right) + a^{(t)}\left(\frac{t-1}{M} - 1\right) + b^{(t)}\left(\left(\frac{t-1}{M}\right)^3 - 1\right),$$

where $a^{(t)}$ (resp., $b^{(t)}$) denotes the number of unordered pairs of 4-cliques which share one edge (resp., three edges) in $G^{(t)}$. Thus, for any $\varepsilon, \delta \in (0,1)$, if

$$M > (t-1)\max\left\{\sqrt[5]{\frac{3}{\delta\varepsilon^2|C_4^{(t)}|}}, \frac{3a^{(t)}}{\delta\varepsilon^2|C_4^{(t)}|^2}, \sqrt[3]{\frac{3b^{(t)}}{\delta\varepsilon^2|C_4^{(t)}|^2}}\right\},$$

then $Pr(|\varkappa^{(t)} - |C_4^{(t)}| < \varepsilon|C_4^{(t)}|) > 1-\delta$.

5.2 Variance Comparison

Although the upper bounds obtained in Theorems 4.4, 4.9, and 5.3 cannot be compared directly, they still provide some useful insight on which algorithm may be performing better according to the properties of $G^{(t)}$.

Let us consider the first, dominant, terms of each of the variance bounds, that is $|C_4^{(t)}|((\frac{t-1}{M_e})^4 \frac{\tau^{(t)}}{M_\Delta} - 1)$ for TS4C$_1$, $|C_4^{(t)}|(c(\frac{t-1}{M_e})^4(\frac{\tau^{(t)}}{M_\Delta})^2 - 1)$ for TS4C$_2$, and $|C_4^{(t)}|((\frac{t-1}{M})^5 - 1)$ for FourEst. All the bounds share a similar dependence from the number of pairs of cliques that share one or three edges in the second and third term. Due to the fact that TS4C$_1$ splits the memory into two levels (in particular, with $M_e = 4M/5$) we have a higher overall contribution for these terms.

While TS4C$_1$ exhibits a slightly higher constant multiplicative term cost due to the splitting of the memory in the Tiered Sampling approach, the most relevant difference is however given by

the term $\frac{\tau^{(t)}}{M}$ appearing in the bound for TS4C$_1$ compared with an additional $\frac{t-1}{M}$ appearing in the bound for FourEst. Recall that $\tau^{(t)}$ denotes here the number of triangles *observed* by the algorithm up to time t. Due to the fact that the probability of observing a triangle decreases quadratically with respect to the size of the graph t, we expect that $\tau < t$ and, for sparser graphs for which 3 and 4-cliques are indeed "*rare patterns,*" we actually expect $\tau^{(t)} << t$. Therefore, under these circumstances, we would expect $M/5\tau >> M/t$.

This is the critical condition for the success of the Tiered Sampling approach. If the sub-structure selected as a tool for counting the motif of interest is not "*rare enough*" then there is no benefit in devoting a certain amount of the memory budget to maintaining a sample of occurrences of the sub-structure. Such a problem would, for instance, arise when using the Tiered Sampling approach for counting triangles using *wedges* (i.e., two-hop paths) as a sub-structure, as attempted in [19], as in most real-world graph the number of wedges is much greater of the number of edges themselves making them not suited to be used as a sub-structure.

5.3 Experimental Evaluation Over Random Graphs

In this section, we compare the performances of our Tiered Sampling algorithms of TS4C$_1$ and TS4C$_2$ with the performance of the single sample approach FourEst, on randomly generated graphs. In particular, we analyze random graph based on a variation[1] of the Barábasi–Albert random graph [5] model, which exhibits the same *scale-free* property observed in many real-world graphs of interest such as social networks. The graph are generated as follows: the initial graph is a *star graph* with $m + 1$ nodes and m edges where a node (i.e., the *center* of the star) is connected to the remaining m ones. Then, n vertices are added one at a time. When the i-th node is added, for $m + 1 \leq i \leq n$, it is connected to m vertices among the $i - 1$ ones already added which are chose with a probability that is *proportional* to the number of *links* (i.e., edges) that the existing nodes already have. In particular, the probability that the i-th new node is connected to node j, for $1 \leq j \leq i - 1$ is $p_i, j = d_j / \sum_{\ell=1}^{i-1} d_\ell$, where d_j denotes the degree of the j-th node before the insertion of the i-th node. The graph constructed at the end of this process has $n + m$ total vertices and nm total edges. The corresponding graph stream can be constructed by simulating the generating process of the random graph and by selecting randomly the order of the edges among the m that are generated for each node insertion. Such variation allows the study of a random preferential attachment graph without starting from a densely connected initial component.

In our experiments, we set $n = 20{,}000$ and we consider various values for m from 50 to 2,000 in order to compare the performances of the two approaches as the number of edges (and thus triangles) increases and the generated graph grows more dense. The algorithms use a memory space whose size corresponds to 5% of the number of edges in the graph nm. While FourEst devotes the entire available memory to maintaining an edges reservoir sample, the Tiered Sampling algorithms will split the available space between the edge sample $\mathcal{S}_e$ and the triangle samples ($\mathcal{S}_\Delta$) according to the *splitting criteria* discusses in the respective sections (i.e., for TS4C$_1$: $M_e = 4M/5$ and $M_\Delta = M/5$; for TS4C$_2$ $M_e = 2M/3$ and $M_\Delta = M/3$).

We compare the accuracy of Tiered Sampling and FourEst approaches on random graphs using the standard *MAPE* [16] as defined in Section 2.

In Figure 2, we compare the average MAPE of FourEst, TS4C$_1$ and TS4C$_2$ for Barábasi–Albert random graphs with 20,000 nodes and various values of m ranging from 50 to 2,000. In columns 2,

[1]We use the version provided by the *NetworkX* package https://networkx.github.io/documentation/networkx-1.9.1/reference/generated/networkx.generators.random_graphs.barabasi_albert_graph.html

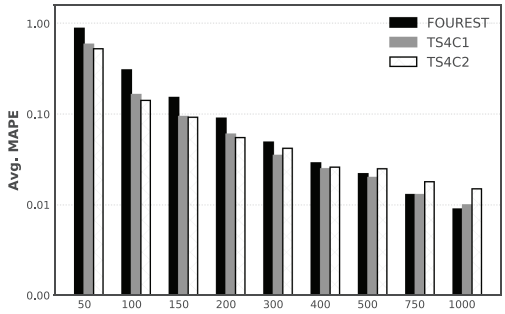


Fig. 2. Average MAPE on Barábasi–Albert random graphs, with $n = 20,000$ and various values of m.

Table 1. Reference for the Notation Used in This Work

Symbol	Explanation
Σ	Edge stream
t	Time Step, i.e., number of edges observed up to step t (included)
t_Δ	Number of triangles observed up to now
$G^{(t)}$	Graph observed up to time t (included)
e_t	Edge observed in the stream at time t.
$C_k^{(t)}$	Set of k-cliques in $G^{(t)}$
$\varkappa^{(t)}$	Estimation of number of 4-cliques at time t
$\mathcal{S}_e$	Uniform sample of edges
$\mathcal{S}_e^{(t)}$	Content of $\mathcal{S}_e$ at the beginning of time t
$\mathcal{S}_\Delta$	Sample of observed triangles
$\mathcal{S}_\Delta^{(t)}$	Content of $\mathcal{S}_\Delta$ at the beginning of time t
M	Memory Size
M_e	Edge sample memory size
M_Δ	Triangle sample memory size
$\tau^{(t)}$	Number of triangles seen by TS4C_1 (TS4C_2) up to time t
$\mathcal{N}_u^{\mathcal{S}_e^{(t)}}$	Neighborhood of u with respect to edges in $\mathcal{S}_e^{(t)}$
$\mathcal{N}_{u,v}^{\mathcal{S}}$	Common neighbors of u and v
α	Splitting coefficient

3, and 5 of Table 2, we present the average of the MAPEs of the ten runs for FourEst, TS4C_1, and TS4C_2. In column 4 (resp., 6) of Table 2, we report the percent reduction/increase in terms of the average MAPE obtained by TS4C_1 (resp., TS4C_2) with respect to FourEst.

Both Tiered Sampling algorithms consistently outperform FourEst for values of m up to 400, that is for fairly sparse graphs for which we expect 3 and 4-cliques to be rare patterns. The advantage over FourEst is particularly strong for values of m up to 200 with reductions of the average MAPE up to 30%. For denser graphs, i.e., $m \geq 750$, FourEst outperforms *both* Tiered Sampling algorithms. This in consistent with the intuition discussed in Section 5.2, as for denser graphs triangles are not "*rare enough*" to be worth saving over edges. Note however that in these cases the quality of *all* the estimators is very high (i.e., MAPE$\leq 1\%$). We can also observe that TS4C_2 outperforms TS4C_1 for $m \leq 200$. Vice versa, TS4C_1 outperforms TS4C_2 (and FourEst) for $300 \leq m \leq 750$. Since, as discussed in Section 4.2, TS4C_2 gives "*more importance*" to triangles, it works particularly well

Table 2. Comparison of MAPE of FourEst. $TS4C_1$ and $TS4C_2$ for Barábasi–Albert Graphs

m	FourEst	$TS4C_1$	Change $TS4C_1$	$TS4C_2$	Change $TS4C_2$
50	0.8775	0.5862	−33.19%	0.5222	−40.49%
100	0.3054	0.1641	−46.27%	0.1408	−53.90%
150	0.1521	0.0937	−38.34%	0.0917	−39.70%
200	0.0899	0.0599	−33.39%	0.0549	−38.95%
300	0.0486	0.0346	−28.80%	0.0417	−14.19%
400	0.0289	0.0249	−13.87%	0.0261	−9.32%
500	0.0221	0.0197	−11.28%	0.0239	8.08%
750	0.0134	0.0132	−1.57%	0.0181	34.90%
1000	0.0088	0.0099	13.14%	0.0146	66.36%

when the graph is very sparse, and triangles are particularly rare. As the graph grows denser (and the number of triangles increases), $TS4C_1$ performs better until, for highly dense graphs FourEst produces the best estimates.

6 ADAPTIVE TIERED SAMPLING ALGORITHM

An appropriate partition of the available memory between the layers used in the Tiered Sampling approach is crucial for the success of the algorithm. While assigning more memory to the triangle sample allows to maintain more sub-patterns, removing too much space from the edge sample reduces the probability of observing new triangles.

While in Section 4.1 (resp., Section 4.2), we provide a general rule according to which to decide how to split the available memory space for $TS4C_1$ (resp., $TS4C_2$), such partition may not always lead to the best possible results. For instance, if the graph being observed is very sparse, assigning a large portion of the memory to the triangles would result in a considerable waste of memory space due to the low probability of observing triangles. Further, as discussed in Section 5, depending on the properties of $G^{(t)}$ an approach based on simply maintaining a sample of the edges could perform better than the Tiered Sampling algorithms. As in the graph streaming setting these properties are generally not known a priori nor stable through the graph evolution, a fixed memory allocation policy appears not to be the ideal solution.

In this section, we present ATS4C, an *adaptive* variation of our $TS4C_2$ algorithm, which *dynamically* analyzes the properties of $G^{(t)}$ through time and consequently decides how to allocate the available memory.

Algorithm description: We present a step by step pseudocode description of ATS4C in Algorithm 4. ATS4C has two main "*execution regimens*": the "*initial regimen*" **(R1)** for which it behaves exactly as FourEst, and the "*stable regimen*" **(R2)** for which it behaves similarly to $TS4C_2$. **(R1)** is the initial regimen for ATS4C. Once the algorithm switches to **(R2)** it does not ever switch back to **(R1)**. ATS4C maintains an estimate $\varkappa^{(t)}$ of the number of 4-cliques observed on the stream up to time t. ATS4C increases $\varkappa$ each time a 4-clique is observed according to the same method discusses for FourEst while in **(R1)** (by invoking Update4CliqueFirstRegimen, line 13 of Algorithm 4), and according to the same method discussed for $TS4C_2$while in **(R3)** (by invoking Update4CliqueSecondRegimen, line 26 of Algorithm 4). Analogously to what done for the other algorithms discusses so far, towards guaranteeing that $\varkappa$ is an *unbiased estimator*, each time a 4-clique is detected ATS4C computes the probability p of such detection and increases the estimate by p^{-1}.

ALGORITHM 4: ATS4C - Adaptive Version of Tiered Sampling

Input: Insertion-only edge stream Σ, M

$\mathcal{S}_e \leftarrow \emptyset$, $\mathcal{S}_\Delta \leftarrow \emptyset$, $\mathcal{S}'_\Delta \leftarrow \emptyset$, $M'_\Delta \leftarrow 0$, $t \leftarrow 0$, $t_\Delta \leftarrow 0$, $\sigma \leftarrow 0$, $r \leftarrow 1$
for each element (u, v) from Σ **do**
  $t \leftarrow t + 1$
  if $r = 1$ **then** ▷ Operations while in **(R1)**
    if $t\%M = 0$ **then** ▷ Every M time steps evaluate whether to switch from **(R1)** to **(R2)**
      $\alpha \leftarrow$ SWITCH ▷ Compute recommended splitting coefficient
      if $\alpha > 0$ **then**
        $r \leftarrow 2$ ▷ Switch to **(R2)**
        $\mathcal{S}_\Delta \leftarrow$ CreateTriangleReservoir(α) ▷ Create and populate triangles sample $\mathcal{S}_\Delta$
        $\mathcal{S}_e \leftarrow$ Subsample($\alpha, \mathcal{S}_e$) ▷ Select uniformly at random a subset of size αM of $\mathcal{S}_e$
        Update4CliquesSecondRegimen($(u, v), t$) ▷ Update 4-clique estimate
    else
      Update4CliquesFirstRegimen($(u, v), t$) ▷ Remain in **(R1)**, update 4-clique estimate
  else if $r = 2$ **then** ▷ Operations while in **(R2)**
    if $t\%M = 0$ **then** ▷ Every M time steps update split proportions between S_e and S_Δ
      UpdateMemory ▷ Decide whet er to assign more space to S_Δ
    if $t'_\Delta < M_\Delta$ **then**
      UpdateTriangles($(u, v), t'_\Delta, S'_\Delta$)
    $p_\Delta \leftarrow min(1, \frac{M_\Delta}{t_\Delta})$, $p'_\Delta \leftarrow min(1, \frac{M'_\Delta}{t'_\Delta})$
    if $p_\Delta < p'_\Delta$ **then** ▷ Merge main triangle sample $\mathcal{S}_\Delta$ and temporary $\mathcal{S}'_\Delta$
      $S_{merged} \leftarrow \emptyset$
      for each $(x, y, z) \in S_\Delta$ **do**
        if FlipBiasedCoin($\frac{p'_\Delta}{p_\Delta}$) = heads **then** ▷ Select the triangles to be moved from $\mathcal{S}'_\Delta$ to $\mathcal{S}_\Delta$
          $S_{merged} \leftarrow S_{merged} \cup (x, y, z)$
      $S_\Delta \leftarrow S_{merged} \cup S'_\Delta$
    Update4CliquesSecondRegimen($(u, v), t$) ▷ Update 4-clique estimate
    UpdateTriangles($(u, v), S_\Delta$) ▷ Update triangles sample
  SampleEdge($(u, v), t$) ▷ Update edges sample
function Switch ▷ Decide whether to switch from **(R1)** to **(R2)**
  $p_s \leftarrow min(1, \frac{M}{t})^5$
  $p_\alpha \leftarrow min(1, \alpha\frac{M}{t})^4 \cdot min(1, (1-\alpha)\frac{M}{t_\Delta})^2$ **where** $a = argmax_{\alpha \in [\frac{2}{3}, 1]} p_\alpha$
  if $p_s < p_\alpha$ **then**
    return α ▷ ATS4C switches from **(R1)** to **(R2)**
  else
    return 0 ▷ ATS4C remains in **(R1)**

36: **function** CreateTriangleReservoir(α)
37: $i \leftarrow 1$
38: $M_\Delta = (1-\alpha)M$
39: **while** $|S_\Delta| < M_\Delta$ **do**
40: $(x,y)^{(i)} \leftarrow$ *Edge observed at time i*
41: **for each** element z from $\mathcal{N}^{S^{(i)}_{x,y}}$ **do**
42: $t_\Delta \leftarrow t_\Delta + 1$
43: $S_\Delta \leftarrow S_\Delta \cup (x,y,z)$
44: **while** $|S| > M - M_\Delta$ **do**
45: $(v,w) \leftarrow$ *random edge from S*
46: $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(v,w)\}$
47: **function** UpdateMemory
48: $t_\Delta^{(i+1)M} = 2t_\Delta^{(iM)} - t_\Delta^{((i-1)M)}$ ▷ Prediction of number of triangles at (i+1) step
49: $p_\alpha \leftarrow min(1, \alpha\frac{M}{t})^4 \cdot min(1, (1-\alpha)\frac{M}{t_\Delta^{(i+1)M}})^2$ **where** $a = argmax_{\alpha\in[\frac{2}{3},1]}p_\alpha$
50: **if** $\alpha < \frac{M_\Delta}{M}$ **then**
51: **for** $i \in [1, \frac{M_\Delta}{M} - \alpha]$ **do**
52: $(v,w) \leftarrow$ *random edge from S*
53: $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(v,w)\}$
54: $M_\Delta \leftarrow M_\Delta + 1, M'_\Delta \leftarrow M'_\Delta + 1$
55: **function** UpdateTriangles($(u,v),t$) ▷ Update $\mathcal{S}_\Delta$
56: $\mathcal{N}^{\mathcal{S}}_{u,v} \leftarrow \mathcal{N}^{\mathcal{S}}_u \cap \mathcal{N}^{\mathcal{S}}_v$
57: **for each** element w from $\mathcal{N}^{\mathcal{S}}_{u,v}$ **do**
58: $t_\Delta \leftarrow t_\Delta + 1$
59: SampleTriangle(u,v,w)
60: **function** Update4CliquesFirstRegimen($(u,v),t$))
61: Update $\varkappa$ using procedure Update4Cliques($(u,v),t$) of FourEst algorithm
62: **function** Update4CliquesSecondRegimen($(u,v),t$)
63: Update $\varkappa$ using procedure Update4Cliques($(u,v),t$) of TS4C$_2$ algorithm

While in **(R1)**, every M time steps ATS4C decides, based on the number of triangles observed so far, whether to switch from **(R1)** to **(R2)**. Recall that, from Lemma 2.1 (resp., Lemma 4.2), the probability of a 4-clique whose last edge is observed at time t being observed by TS4C$_2$ (resp., FourEst) is approximately $p_\alpha = (\min\{1, \alpha M/t\})^4(\min\{1,(1-\alpha)M/\tau^{(t)}\})^2$ (resp., $p_s = (\min\{1, M/t-1\})^5$), where $2/3 < \alpha < 1$ (resp., $0 < 1-\alpha < 1/3$) denotes the fraction of the available memory which is assigned to the edge sample (resp., triangle sample). SWITCH determines which partition of the available space between edge only sample and triangle sample would maximize the approximate probability of detecting a 4-clique using TS4C$_2$. That is, SWITCH computes $\alpha^* = \text{argmax}_{\alpha\in[2/3,1)}p_\alpha$ (line 31 Algorithm 4). SWITCH then evaluates if the approximate probability of detecting a clique using the FourEst approach is higher than that achievable using TS4C$_2$ while partitioning the available memory according to α^* (line 32). If that is the case (i.e., $p_s > p_{\alpha^*}$), SWITCH returns zero and ATS4C elects to remain in **(R1)** (line 11). Vice versa, if $p_s \leq p_{\alpha^*}$, SWITCH returns the value α^* and ATS4C transitions to **(R2)**): the triangle reservoir $\mathcal{S}_\Delta$ is assigned $(1-\alpha^*)M$ memory space, and it is filled with the triangles composed by the edges *currently* in the edge reservoir, using the reservoir sampling scheme.

Finally the edge reservoir $\mathcal{S}_e$ is constructed by selecting $\alpha^* M$ of the edges in the current sample uniformly at random, thus ensuring that $\mathcal{S}_e$ is an *uniform sample* (lines 7–11). Once ATS4C switches to **(R2)** it never goes back to **(R1)**.

While in **(R2)**, as long as $|\mathcal{S}_\Delta| < M/3$, every M time steps ATS4C evaluates whether it is opportune to assign a higher fraction of the available memory to $\mathcal{S}_\Delta$. Let $t = iM$, rather than just using the information of the number of triangles seen so far $\tau^{(iM)}$, ATS4C computes a "*prediction*" of the total number of triangles seen until $(i+1)M$ assuming that the number of triangles seen during the next M steps will equal the number of triangles seen during the last M steps (line 53), that is $\tilde{\tau}^{((i+1)M)} = 2\tau^{(iM)} - \tau^{((i-)M)}$. ATS4C then evaluates the partitioning of the available memory space among the two tiers which would maximize the approximate probability of detecting a 4-clique at time $(i+1)M$, i.e., $\alpha^* = \text{argmax}_{a \in [2/3,1)} (\min\{1, \alpha M/(i+1)M)\})^4 (\min\{1, (1-\alpha)M/\tilde{\tau}^{((i+1)M)}\})^2$. Let α denote the split being used by ATS4C at $t = iM$:

- If $\alpha^* > \alpha$, ATS4C determines that increasing the memory space devoted to the triangles sample would not improve the algorithm's likelihood of detecting 4-cliques. ATS4C continues its execution with no further operations (ATS4C never reduces the memory space assigned to $\mathcal{S}_\Delta$).
- Otherwise, if $\alpha^* < \alpha$, ATS4C removes $(\alpha - \alpha^*)M$ edges from $\mathcal{S}_e$ selected uniformly at random, thus ensuring that $\mathcal{S}_e$ is still an uniform sample of the edges observed on the stream, and the freed space is assigned to $\mathcal{S}_\Delta$. Let us denote this space as $\mathcal{S}'_\Delta$.

As ATS4C progresses and observes new triangles it fills $\mathcal{S}'_\Delta$ using the reservoir sampling scheme. That is $\mathcal{S}'_\Delta$ is used a *temporary buffer* to store observed triangle patterns. $\mathcal{S}_\Delta$ and $\mathcal{S}'_\Delta$ are then *merged* at the first time step for which the probability p_Δ of a triangle seen before the creation of $\mathcal{S}'_\Delta$ being in $\mathcal{S}_\Delta$ becomes lower than the probability $p_{\Delta'}$ of a triangle observed after the creation of $\mathcal{S}'_\Delta$ being in it. The merged triangle sample contains all the triangles in $\mathcal{S}'_\Delta$, while the triangles in $\mathcal{S}_\Delta$ are moved to it with probability $p_{\Delta'}/p_\Delta$. This ensures that after the merge all the triangles seen on the stream are kept in the triangle reservoir with probability $p_{\Delta'}$. After the merge, ATS4C operates the samples as described in TS4C$_2$. Finally, ATS4C increases the memory space for $\mathcal{S}_\Delta$ only if all the currently assigned space is used. Once $\mathcal{S}_\Delta$ is filled, ATS4C maintains it a reservoir sample for the triangle observed on the stream (invoking the UpdateTraingles function, line 55).

The analysis of ATS4C presents several additional challenges compared to those of our previous algorithms TS4C$_1$ and TS4C$_2$, due to the interplay between execution regimens. Still, as during **(R2)** (resp., **(R1)**) ATS4C behaves effectively as TS4C$_2$ (resp., FourEst), albeit with some further complication due to the adjustment of the assignment of memory between layer, whenever a 4-clique is observed, ATS4C may compute *exactly* the probability of observing such clique, by relying on the properties of TS4C$_2$ and FourEst.

Theorem 6.1. *The estimator $\varkappa$ returned by ATS4C is unbiased, that is $\varkappa^{(t)} = |C_k^{(t)}|$ if $t \leq M$. Further, $\mathbb{E}[\varkappa^{(t)}] = |C_k^{(t)}|$ if $t \geq M$.*

The proof of Theorem 6.1, follows a reasoning analogous to that discussed in similar results for the unbiasedness of TS4C$_2$ and FourEst, and relies on the fact that, as discussed in the algorithm description, whenever a 4-clique is observed, ATS4C can compute *exactly* the probability of observing such clique.

We show how the adaptive assignment of the available memory realized by ATS4C succeeds in combining the advantages of the single edge sample approach (e.g., FourEst), and of the multi-tier prototype-based approach of Tiered Sampling via experimental evaluation on real-world graphs presented in Section 7.2.

Table 3. Graphs Used in the Experiments

Graph	Nodes	Edges	Exact 4-clique count	Tiered Sampling estimate 4-clique count for M = 0.05|E|	Source
DBLP	986,324	3,353,618	40,675,407	41,750,428	[8]
Patent (Cit)	2,745,762	13,965,132	3,296,890	3,294,045	[33]
LastFM	681,387	30,311,117	46,201,534,449	49,189,084,815	[33]
Live Journal	5,363,186	49,514,271	16,121,317,106	16,035,963,528	[8]
Hollywood	1,917,070	114,281,101	728,184,767,782	727,002,104,249	[8]
Orkut	3,072,441	117,185,083	3,221,163,953	3,210,957,578	[25]
Twitter	25,080,769	100,000,000	19,920,704	20,562,810	[8, 33]

7 EXPERIMENTAL EVALUATION

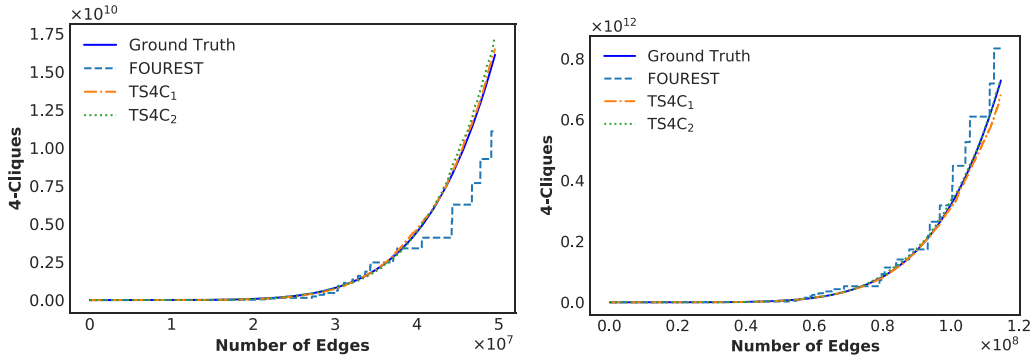
In this section, we evaluate through extensive experiments the performance of our proposed Tiered Sampling method when applied for counting 4-cliques and 5-cliques in large graphs observed as streams. We use several real-world graphs with size ranging from 10^6 to 10^8 edges (see Table 3 for a complete list). All graphs are treated as undirected. The edges are observed on the stream according to the values of the associated timestamps if available, or in random order otherwise. In order to evaluate the accuracy of our algorithms, we compute the "*ground truth*" exact number of 4-cliques (resp., 5-cliques) for each time step using an exact algorithm that maintains the entire $G^{(t)}$ in memory. Besides verifying the accuracy of the proposed Tiered Sampling methods, an important goal of the experimental analysis is to ascertain the benefits of our multi-tier method compared to approaches for which the estimate is achieved by maintaining simply a reservoir sample of the observed edges. Therefore, our main term of comparison will be the FourEst algorithm, which is a generalization of the TRIEST algorithm [33]. Our experiments are implemented in Python and their source code can be found in (https://github.com/erisaterolli/TieredSampling). The experiments were run on the Brown University CS department cluster[2], where each run employed a single core and used at most 4 GB of RAM.

The section is organized as follows: we first evaluate the performance of TS4C$_1$ and TS4C$_2$ and we compare them with the estimations obtained using FourEst. We then present practical examples that motivate the necessity for the adaptive version of our Tiered Sampling approach, and we show how our ATS4C manages to capture the best of the single and multi-level approach. Finally, we show how the Tiered Sampling approach can be generalized in order to count structures other than 4-cliques.

7.1 Counting 4-Cliques

We estimate the global number of 4-cliques on insertion-only streams, starting as empty graphs and for which an edge is added at each time step, using algorithms FourEst, TS4C$_1$, and TS4C$_2$. As discussed in Section 4.1 (resp., Section 4.2), in TS4C$_1$ (resp., TS4C$_2$) we split the total available memory space M as $|\mathcal{S}_e| = 4M/5$ and $|\mathcal{S}_\Delta| = M/5$ (resp., $|\mathcal{S}_e| = 2M/3$ and $|\mathcal{S}_\Delta| = M/3$). The experimental results show that these fixed memory splits perform well for most cases. We then experiment with an adaptive splitting mechanism that handles the remaining cases.

[2]https://cs.brown.edu/about/system/services/hpc/grid/



(a) Live Jouranal (b) Hollywood

Fig. 3. Comparison of $|C_4^{(t)}|$ estimates obtained using TS4C$_1$, TS4C$_2$, and FourEst with $M = 5 \times 10^5$.

Table 4. Average MAPE of Various Approaches For All Graphs with Available Memories of 1%, 2%, and 5% of the Size of the Graph

	M = 0.01				M = 0.02			
	FOUR EST	**TS4C1**	**TS4C2**	**TS Change**	**FOUR EST**	**TS4C1**	**TS4C2**	**TS Change**
DBLP	0.942	0.658	0.599	−36.00%	0.915	0.336	0.526	−63.28%
Holly wood	0.056	0.029	0.023	−59.00%	0.013	0.016	0.01	−23.08%
Last FM	0.07	0.20	0.196	65.00%	0.022	0.239	0.10	354.54%
Live Journal	0.295	0.079	0.148	−73.00%	0.098	0.02	0.025	−79.59%
Orkut	0.789	0.224	0.086	−89.00%	0.116	0.059	0.058	−50.00%
Patent Cit	0.954	0.767	0.188	−80.00%	0.473	0.304	0.141	−70.19%
Twitter	0.928	0.896	0.511	−45.00%	0.928	0.481	0.428	−53.88%

	M = 0.05			
	FOUR EST	**TS4C1**	**TS4C2**	**TS Change**
DBLP	0.114	0.069	0.105	−39.47%
Holly wood	0.004	0.006	0.002	−50.00%
Last FM	0.003	0.0304	0.032	866.67%
Live Journal	0.019	0.006	0.006	−68.42%
Orkut	0.013	0.021	0.008	−38.46%
Patent Cit	0.112	0.113	0.039	−65.18%
Twitter	0.205	0.076	0.088	−62.93%

In the **TS Change** column we report the decrease/increase in MAPE of the best performing Tiered Sampling algorithm among TS4C$_1$ and TS4C$_2$ compared to the single reservoir algorithm FourEst.

In Figure 3, we present the estimation obtained by averaging 10 runs of respectively TS4C$_1$, TS4C$_2$, and FourEst using total memory space $M = 5 \times 10^5$ for the LiveJournal and Hollywood graphs (i.e., respectively using less than 1% and 0.5% of the graph size). While the average of the runs for TS4C$_1$ and TS4C$_2$ are almost *indistinguishable* from the ground truth, the quality of the estimator obtained using FourEst considerably worsens as the graph size increases.

In Table 4, we report the average MAPE performance over five runs for TS4C$_1$, TS4C$_2$, and FourEst for graphs listed in Table 3. For each graph, we assign a different total memory space equal to a 1%, 2%, and 5% fraction of the total number of edges (i.e., of the size of the graph).

Table 5. Average Update Time For All Graphs Measured in Microseconds

Dataset	FourEst	TS4C$_1$	TS4C$_2$
DBLP	6.56	19.6	15.8
Hollywood	10.95	15.75	19.95
Lastfm	17.65	45.62	64.92
Live Journal	9.22	10.72	16.49
Orkut	5.55	8.93	7.45
PatentCit	9.25	11.95	12.12
Twitter	9.08	20.49	17.77

Except for LastFM, our Tiered Sampling algorithms clearly and consistently outperform FourEst for all the considered available memory sizes with the average MAPE reduced by up to 89%. While the performances of all algorithms are improved (in terms of MAPE reduction) as the size of the available memory increases, the improvement achieved by or Tiered Sampling algorithms with respect to FourEst appear to consistent for the various memory sizes being considered. While on most graphs TS4C$_1$ and TS4C$_2$ produce similar estimates (and, hence, MAPE), it can be noted that TS4C$_2$ clearly outperforms TS4C$_1$ in graphs for which the number of 4-cliques is low compared to the number of edges, such as Patent (Cit) and Twitter. In these graphs, triangles are "*rare enough*" so that assigning a greater fraction of the available memory to maintain their occurrences and relying on observed triangles to count 4-cliques leads to tighter estimates.

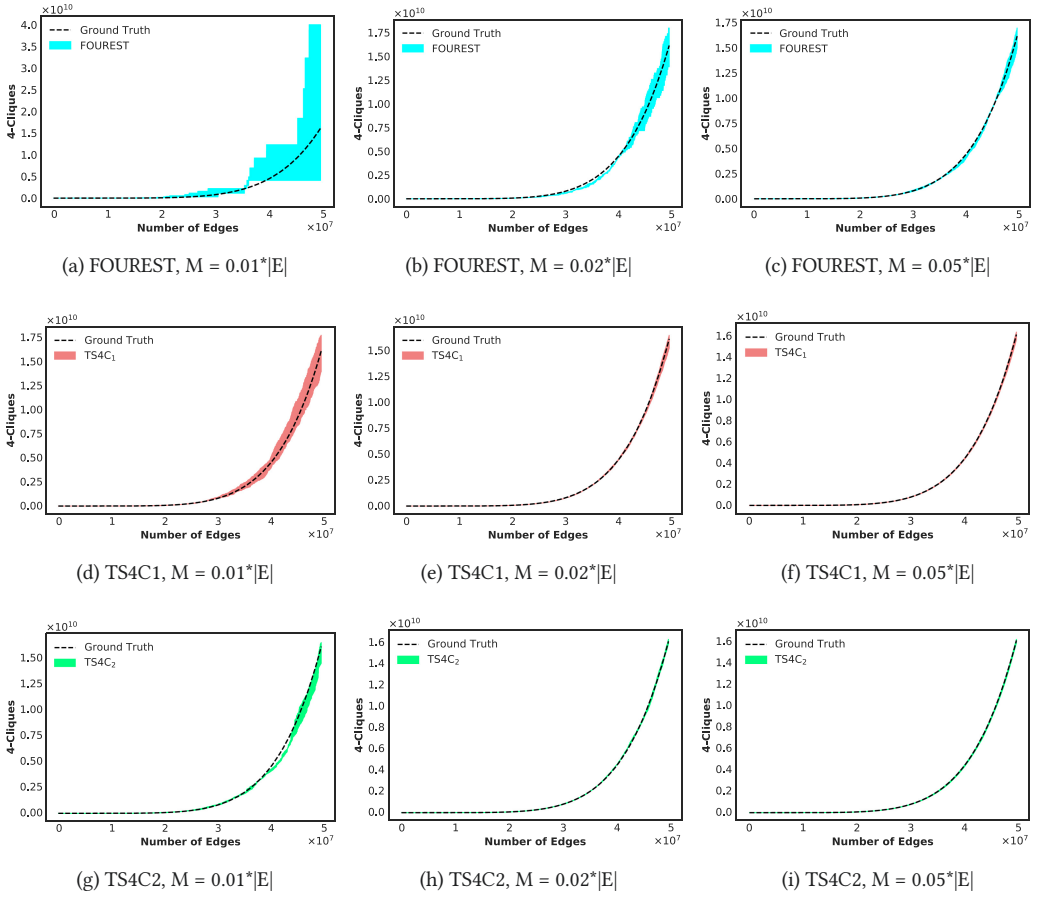
LastFM is the only graph for which FourEst (considerably) outperforms the Tiered Sampling algorithms. Such phenomenon is due to the high density of the graph $|E|/|V| > 500$ and to the fact that for the LastFM graph triangles are not a rare enough sub-structure to justify the choice of maintaining them over simple edges.

We analyze the variance reduction achieved using our Tiered Sampling algorithms by comparing the empirical variance observed over ten runs for all graphs using available memory M equal to 1%, 2%, and 5% of the size of the graphs. The results for Live Journal graph are reported in Figure 4. While for both TS4C$_1$ and TS4C$_2$ the minimum and maximum estimators are close to the ground truth throughout the evolution of the graph even in cases having a small amount of available memory, FourEst estimators exhibit very high variance especially towards the end of the stream and when the available memory is small. This trend is consistently observed for all the other graphs. As an illustrative example, we show the variance of all algorithms for Twitter with an available memory size of 5% of the total graph.

Despite the differences between TS4C$_1$ and TS4C$_2$, their variances appear to be strikingly similar for the evaluated graph, with TS4C$_2$ exhibiting slightly better performance, especially during the first half of the stream. This is due to the rarity of the triangle prototype pattern used in the detection of 4-cliques for the Hollywood graph: compared to TS4C$_1$, TS4C$_2$ devotes a higher fraction of the available memory to $\mathcal{S}_\Delta$ which leads to a higher likelihood of detecting 4-cliques and, hence, a lower empirical variance.

Our experiments not only verify that TS4C$_1$ and TS4C$_2$ yield high quality estimates which are in most cases (except for the LastFM graph) superior to the ones achievable using a single sample strategy, but also validate the general intuition underlying the Tiered Sampling approach.

Both Tiered Sampling algorithms are extremely scalable, showing average update times in the order of hundred microseconds for all graphs as shown in Table 5.



(a) FOUREST, M = 0.01*|E| (b) FOUREST, M = 0.02*|E| (c) FOUREST, M = 0.05*|E|

(d) TS4C1, M = 0.01*|E| (e) TS4C1, M = 0.02*|E| (f) TS4C1, M = 0.05*|E|

(g) TS4C2, M = 0.01*|E| (h) TS4C2, M = 0.02*|E| (i) TS4C2, M = 0.05*|E|

Fig. 4. Variance of various approaches for Live Journal with available memories of 1%, 2%, and 5% of the size of the graph.

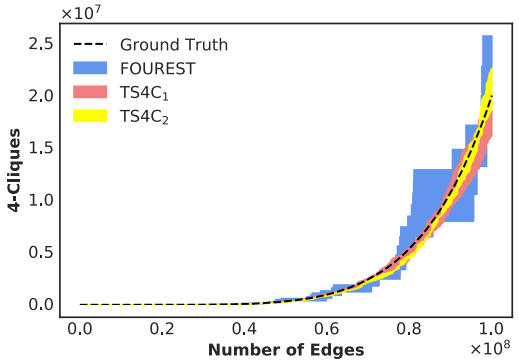


Fig. 5. Variance of various approaches for Twitter with available memory of 5% of the size of the graph.

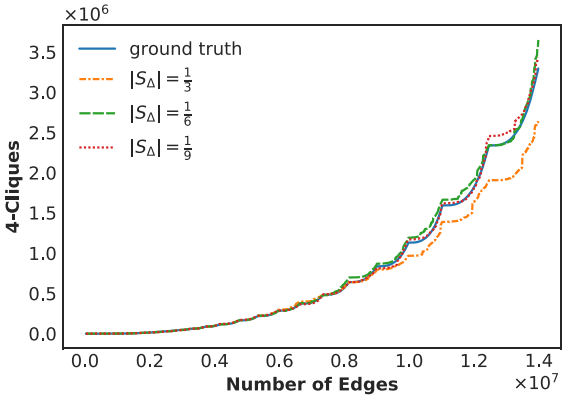


Fig. 6. $|C_4^{(t)}|$ estimations for Patent (Cit) using TS4C$_2$ with $M = 5 \times 10^5$ and different memory space assignments among tiers.

7.2 Adaptive Tiered Sampling

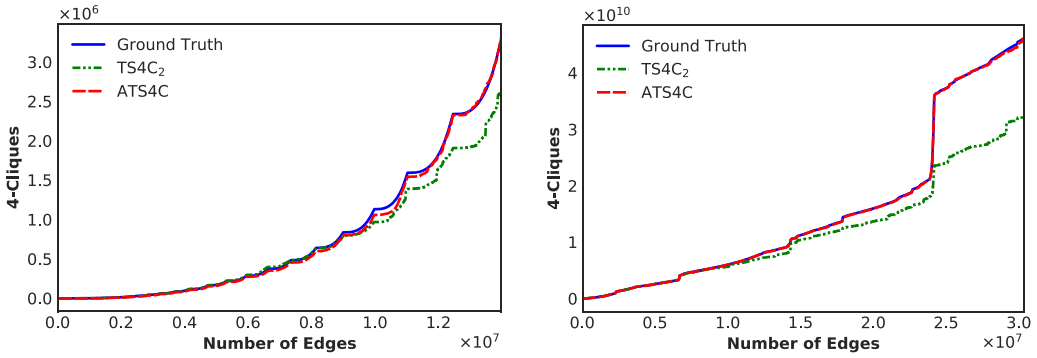
In Section 7.1, we showed that TS4C$_2$, allows to obtain high quality estimations for the number of 4-cliques outperforming in most cases both TS4C$_1$ and FourEst. These results were obtained splitting the available memory such that $|\mathcal{S}_e| = 2M/3$ and $|\mathcal{S}_\Delta| = M/3$. As discussed in Section 6, while this is a useful general rule, depending on the properties of the graph, different splitting rules may yield better results. We verify this fact by evaluating the performance of TS4C$_2$ when used to analyze the Patent(Cit) graph using different assignments of the total space $M = 5 \times 10^5$ to the two levels. The results in Figure 6 show that decreasing the space assigned to the triangle sample from $M/3$ to $M/9$ allows to achieve estimates that are closer to the ground truth leading to a 31% reduction of the average MAPE. Due to the sparsity of the Patent(Cit) graph, TS4C$_2$ observes a very small number of triangles for a large part of the stream. Assigning a large fraction of the memory space to $\mathcal{S}_\Delta$ is thus inefficient as the probability of observing new triangles is reduced, and the space assigned to $\mathcal{S}_\Delta$ is not fully used.

To overcome such difficulties, in Section 6, we introduced ATS4C, an *adaptive* version of TS4C$_2$, which allows to dynamically adjust the use of the available memory space based on the properties of the graph being observed. We experimentally evaluate the performance of ATS4C over ten runs on the Patent(Cit) and the LastFM graphs and we compare it with TS4C$_2$ and FourEst using $M = 5 \times 10^5$.

As shown in Figure 7, for both graphs, ATS4C produces estimates that are nearly indistinguishable from the ground truth. ATS4C clearly outperforms TS4C$_2$ (with $|\mathcal{S}_\Delta| = \frac{M}{3}$) on Patent(Cit) where triangles are sparse motifs achieving an ∼85% reduction of the average MAPE compared to TS4C$_2$. ATS4C returns high quality estimations even for the LastFM graph, for which the single level approach FourEst outperforms TS4C$_1$ and TS4C$_2$.

8 GENERALIZING THE TIERED SAMPLING APPROACH

While so far we focused on counting the number of occurrences of 4-cliques in a graph stream, the Tiered Sampling approach can be generalized towards estimating the counts of a wide variety of graphlets. In this section, we discuss a heuristic that allows generalizing the approach discussed so far to a general graphlet. As an example, we discuss an application of the Tiered Sampling approach that yields high-quality estimates of the number of 5-cliques in graph streams, named TS5C.



(a) Patent(Cit) (b) LastFM

Fig. 7. Comparison of $|C_4^{(t)}|$ estimates obtained using ATS4C, TS4C$_2$, and FourEst with $M = 5 \times 10^5$.

8.1 The Tiered Sampling Framework

Let $G' = (V', E')$ be the graph or pattern of interest for whom we aim to estimate the number of occurrences in the graph stream. Towards obtaining an algorithm $\mathcal{A}$, which provides such estimates, we apply the Tiered Sampling approach following the following steps:

(1) **Identify a "prototype" sub-pattern:** We identify a sub-pattern of G', henceforth denoted as $G'' = (V'', E'')$ such that $V'' \subseteq V'$ and $E'' \subseteq E'$. G'' will play a role analogous to that played by triangles towards counting 4-cliques in the stream. Such sub-pattern serves as a *prototype* to aid in the detection of instances of the graphlet of interest. Part of the memory available to algorithm $\mathcal{A}$ will be allocated to maintaining a reservoir sample of the instances of G'' observed on the stream by $\mathcal{A}$ so far, henceforth named $\mathcal{S}_P$, while the remaining available memory will be used as a uniform sample of the edges observed on the stream, henceforth named $\mathcal{S}_e$. Both samples are maintained by $\mathcal{A}$ according to the reservoir sampling mechanism, following steps analogous to those discussed for TS4C$_1$ and TS4C$_2$. Further, we assume that whenever an occurrence of G'' is selected to be included in $\mathcal{S}_P$, $\mathcal{A}$ maintains information on the time step at which it was detected.

Algorithm $\mathcal{A}$ operates similarly to TS4C$_1$ and TS4C$_2$: whenever an edge e is observed in the steam at time t $\mathcal{A}$ verifies whether an instance of G' can be obtained by combining e with one of the prototypes in $\mathcal{S}_P^{(t)}$ and with *exactly* $|E'| - |E''| - 1$ edges in $\mathcal{S}_e^{(t)}$. $\mathcal{A}$ maintains an estimate $\varkappa^{(t)}$ of the number of occurrences of G' in the stream up to time t which is updated each time an instance of G' is detected. Then $\mathcal{A}$ determines whether an instance of the prototype G'' can be detected combining e with $|E''| - 1$ edges in $\mathcal{S}_e^{(t)}$. Each detected occurrence G'' is submitted to the reservoir $\mathcal{S}_P$. Finally, $\mathcal{A}$ inserts e into the reservoir $\mathcal{S}_e$.

While G'' can be chosen arbitrarily among the sub-graphs of G' some properties appear to be desirable: (i) G'' should be *rare enough* so that it is worth maintaining instances of it in memory but not *overly rare* so that part of the available memory would not be utilized (unless using the adaptive version of Tiered Sampling); (ii) G'' should cover a significant fraction of the edges of G'', that is, it should be possible to complete an instance of G' by adding a small number of edges to G'; and (iii) G'' should be *connected*.

Property (i) is desirable since, as shown for the case of 4-cliques, under such circumstances $\mathcal{A}$ achieves efficient use of the available memory and the benefit of maintaining *rare* instances of the prototype towards detecting G'. Property (ii) is desirable as it allows for a simpler and less computational intensive detection of instances of G' by composing instances of G'' and a few remaining edges. Finally, property (iii) is desirable as connected sub-graphs are, in general, rarer than non-connected graphs with the same number of edges, and they allow for a simpler—less computationally intensive—algorithmic criteria for detecting occurrences of G' by completing occurrences of the prototype G''. While any subgraph satisfying these properties appears desirable to be used as a prototype, as a heuristic, choosing as G'' a clique sub-graph of G' satisfies the desired properties.

(2) **Bounding the probability of observing G':** Algorithm $\mathcal{A}$ maintains an estimator $\varkappa(t)$ of the number of occurrences of G' observed on the stream up to step t. Ideally, $\mathcal{A}$ would proceed by increasing $\varkappa(t)$ each time an occurrence of G' is detected on the stream by an amount which corresponds to the reciprocal of the probability of such occurrence being observed by $\mathcal{A}$. As discussed in Theorem 4.3 (resp., Theorem 4.8) for TS4C$_1$ (resp., TS4C$_2$), this does indeed guarantee $\varkappa(t)$ to be an *unbiased estimate* of the number of occurrences of G'. However, computing *exactly* the probability of observing G' and G'' is, in general, rather complicated as it requires breaking down a high number of sub-cases each tied to the order according to which the edges are observed on the stream and whose number does, therefore, grow *exponentially* with respect to the number of edges of G'. While it is possible to reduce the number of these cases by matching some common patterns, the analysis is still rather complex and not easily generalize from one sub-graph G' to another G'. Rather than computing the exact value, using an approximate of the probability according to whom $\mathcal{A}$ observes an instance of G' (resp., G'') is generally sufficient to obtain good quality, albeit generally non-unbiased, estimates for the count of occurrences of G' while considerably reducing the effort of the analysis.

Let t denote the time step at whom the last edge e of an occurrence of G', denoted as $\hat{G}' = (\hat{V}', \hat{E}')$, is observed on the stream. Based on the choice of the prototype, it will be possible for $\mathcal{A}$ to detect $\hat{G}'$ by composing an occurrence of the prototype G'' (whose edges has been previously observed on the stream), which may have been previously detected and maintained in the reservoir sample used to maintain instances of G'', with some edges currently stored in the edge reservoir sample, and e itself. Assume that $\mathcal{A}$ detects by composing an occurrence of G'', henceforth denoted as $\hat{G}'' = (\hat{V}'', \hat{E}'')$. stored in $\mathcal{S}_\Delta^{(t)}$, $|E'| - |E''| - 1$ edges of $\hat{G}''$ in $\mathcal{S}_e^{(t)}$, and the edge e itself. Let p denote the probability of $\mathcal{A}$ observing $\hat{G}'$ in such a way, $\mathcal{A}$ approximates p as

$$\begin{aligned}\tilde{p} &= \Pr\left(\hat{G}'' \in \mathcal{S}_\Delta\right) \Pr\left(\hat{V}' \setminus (\hat{V}'' \cup \{e\}) \subseteq S_e^{(t)}\right) \\ &= \Pr\left(\hat{G}'' \in \mathcal{S}_\Delta | \hat{G}'' \text{ was observed by } \mathcal{A}\right) \Pr\left(\hat{G}'' \text{ was observed by } \mathcal{A}\right) \Pr\left(\hat{V}' \setminus (\hat{V}'' \cup \{e\}) \subseteq \mathcal{S}_e^{(t)}\right).\end{aligned}$$

— $\mathcal{A}$ approximates $\Pr(\hat{G}'' \in \mathcal{S}_\Delta | \hat{G}''$ was observed by $\mathcal{A})$, as the probability that, conditioned on $\hat{G}''$ having been observed by $\mathcal{A}$, $\hat{G}''$ is stored in the the reservoir $\mathcal{S}_P$ at time t, that is:

$$\Pr\left(\hat{G}'' \in \mathcal{S}_\Delta | \hat{G}'' \text{ was observed by } \mathcal{A}\right) \approx \min\left\{1, \frac{|\mathcal{S}_P|}{\tau^{(t)}}\right\},$$

where $|\mathcal{S}_P|$ denotes the size of the fraction of the memory space allocated to $\mathcal{S}_P$ and $\tau^{(t)}$ denotes the number of instances of G'' detected by $\mathcal{A}$ up to time $t-1$.

— $\mathcal{A}$ approximates $\Pr(\hat{G}''$ was observed by $\mathcal{A})$ as the probability that when the last edge of $\hat{G}''$ arrived on the stream its remaining $|E''|-1$ were stored in $\mathcal{S}_e$. That is:

$$\Pr\left(\hat{G}'' \text{ was observed by } \mathcal{A}\right) \approx \left(\min\left\{1, \frac{|\mathcal{S}_e|}{t'-1}\right\}\right)^{|E''|-1},$$

where $|\mathcal{S}_e|$ denotes the size of the fraction of the memory space allocated to $\mathcal{S}_e$ and t' is the time step at which the last edge of $\hat{G}''$ was observed on the stream. The approximate used here is a simplified version of the exact value computed in Lemma 2.1.

— $\mathcal{A}$ approximates $\Pr(\hat{V}' \setminus (\hat{V}'' \cup \{e\}))$ as the probability of the $|E'|-|E''|-1$ edges in $\hat{V}' \setminus (\hat{V}'' \cup \{e\}$ being included in $\mathcal{S}_\urcorner$. That is:

$$\Pr\left(\hat{V}' \setminus (\hat{V}'' \cup \{e\})\right) \approx \left(\min\left\{1, \frac{|\mathcal{S}_e|}{t-1}\right\}\right)^{|E'|-|E''|-1}.$$

The approximate used here is a simplified version of the exact value computed in Lemma 2.1.

Hence, we have:

$$\tilde{p} = \min\left\{1, \frac{|\mathcal{S}_P|}{\tau^{(t)}}\right\}\left(\min\left\{1, \frac{|\mathcal{S}_e|}{t'-1}\right\}\right)^{|E''|-1}\left(\min\left\{1, \frac{|\mathcal{S}_e|}{t-1}\right\}\right)^{|E'|-|E''|-1}.$$

In general we have that $p \neq \tilde{p}$. In most cases $\tilde{p}$ is lower than than the actual value p.

(3) **Determining increments to the estimate $\varkappa$:** The computed value $\tilde{p}$ approximates the probability of $\mathcal{A}$ detecting $\hat{G}'$ specifically using the instance $\hat{G}''$ of the prototype completed with edges form $\mathcal{S}_e$. However, in general it will be possible for $\mathcal{A}$ to detect the same occurrence $\hat{G}'$ of the pattern of interest. A similar circumstance was discussed for TS4C$_1$, as the algorithm can detect a 4-clique in two possible ways (i.e., using two possible triangle sub-graphs). Correcting for such phenomena is of crucial importance towards avoiding overestimating the number of occurrences of G'. In order to do so, it is necessary to determine the number of possible ways for $\mathcal{A}$ to detect $\hat{G}'$ given the order of the edges on the stream. Let e denote the last edge of $\hat{G}'$ observed on the stream. If/when $\mathcal{A}$ detects $\hat{G}'$, it counts the number of possible distinct occurrences of the prototype G'' in the sub-graph $(\hat{V}', \hat{E}' \setminus \{e\})$ (i.e., the sub-graph of $\hat{G}'$ obtained by removing e). By construction of $\mathcal{A}$, such number, henceforth denoted as $c(\hat{G}')$ corresponds to the number of possible ways according to whom $\mathcal{A}$ may detect $\hat{G}'$ by combining e, an occurrence of the prototype G'', and exactly $|V'|-|V''|-1$ edges from $\mathcal{S}_e$. $c(\hat{G}')$ can be derived only from e and $\hat{G}'$ which are both known to $\mathcal{A}$ if/when $\hat{G}'$ is detected.

Whenever $\mathcal{A}$ observes an instance of G' it increases $\varkappa$ by $\tilde{p}^{-1}/c(\hat{G}')$. For each occurrence $\hat{G}'$ of G' let $\varkappa_{hatG'}$ be the random variable which corresponds increment to the estimate $\varkappa$ due to $\hat{G}'$. That is, $\varkappa = \sum_{\text{occurrences of G'}} \varkappa_{\hat{G}'}$. Towards $\varkappa$ being an unbiased estimate we

therefore desire $\mathrm{E}[\varkappa_{\hat{G}'}] = 1$. By construction of $\mathcal{A}$ we have:

$$\begin{aligned}
\mathrm{E}\left[\varkappa_{\hat{G}'}\right] &= \sum_{\text{ways for } \mathcal{A} \text{ to detect } \hat{G}'} \frac{\tilde{p}^{-1}}{c(\hat{G}')} \Pr\left(\mathcal{A} \text{ detects } \hat{G}' \text{ in the } i\text{-th way}\right) \\
&\approx \frac{1}{c(\hat{G}')} \sum_{\text{ways for } \mathcal{A} \text{ to detect } \hat{G}'} \tilde{p}^{-1}\tilde{p} \\
&= \frac{c(\hat{G}')}{c(\hat{G}')} \\
&= 1.
\end{aligned}$$

While, for different combinations of prototypes and edges (i.e., the *ways*) the probabilities of detecting $\hat{G}'$, using the approximate $\tilde{p}$ allows for a considerable simplification of the analysis (and, in turn, of the algorithm) while still providing good estimates. While the criteria used deciding how to increment $\varkappa$ aims to have it function as an unbiased estimate of the number of occurrences of G', due to the use of the approximate $\tilde{p}$, $\varkappa$ is not, in general, an actual unbiased estimator. Since, as previously discussed, we have that, in general, $\tilde{p} < p$, our estimator $\varkappa$ will generally *slightly underestimate* the number of occurrences of the pattern of interest.

(4) **Partition of the available memory space among reservoir tiers:** Following the heuristic already used for determining how to partition the available space discussed for $\mathrm{TS4C}_1$ and $\mathrm{TS4C}_2$, toward maximizing the probability of $\mathcal{A}$ detecting instances of G', we assign $|\mathcal{S}_P|$ and $|\mathcal{S}_\gamma|$ in such a way that $\tilde{p}$ is maximized, while setting t, t' and $\tau^{(t)}$ as constants.

By following the outlined steps, it is possible to deploy the Tiered Sampling approach to obtain an algorithm $\mathcal{A}$ that produces high-quality estimates of the number of occurrences of the desired pattern G'. While the construction of $\mathcal{A}$ previously discussed leads to obtaining an algorithm for which the memory space is statically divided among the edge-only reservoir and the prototype reservoir, a generalization of the adaptive algorithm ATS4C can be obtained following similar reasoning.

8.2 Using Tiered Sampling to Count 5-Cliques

To demonstrate the generality of our Tiered Sampling approach, we present TS5C, a one-pass counting algorithms for 5-clique in a stream. Following the steps outlined in the previous section, TS5C selects a 4-clique as a prototype sub-pattern to detect 5-cliques. The algorithm maintains two reservoir samples, one for edges and one for 4-cliques. When the currently observed edge completes a 4-clique with edges in the edge sample the algorithm attempts to insert it to the 4-cliques reservoir sample. A 5-clique is counted when the current observed edge completes a 5-clique using one 4-clique in the reservoir sample and 3 edges in the edge sample. Given the choice of the prototype, each occurrence of a 5-clique may be detected by TS5C in at most two different ways. The probability of each detection is approximated as

$$\tilde{p} = \min\left\{1, \frac{M_P}{\tau}\right\}\left(\min\left\{1, \frac{M_e}{t'-1}\right\}\right)^5\left(\min\left\{1, \frac{M_e}{t-1}\right\}\right)^3,$$

ALGORITHM 5: TS5C - Tiered Sampling for 5-Clique counting

Input: Insertion-only edge stream Σ, integers M, M_C
$\mathcal{S}_e \leftarrow \emptyset$, $\mathcal{S}_P \leftarrow \emptyset$, $t \leftarrow 0$, $\tau \leftarrow 0$, $\varkappa \leftarrow 0$
for each element (u, v) from Σ **do** ▷ Process each edge (u, v) coming from the stream in discretized timesteps
  $t \leftarrow t + 1$
  UPDATE5CLIQUES(u, v) ▷ Update the 5-clique estimator by considering the new 5-cliques closed by edge (u,v)
  UPDATE4CLIQUES(u, v) ▷ Update the 4-clique sample with the new 4-clique observed according to RS scheme
  SAMPLEEDGE$((u, v), t)$ ▷ Update the edges sample with the edge (u, v) according to RS scheme

function UPDATE5CLIQUES$((u, v), t)$
  for each 4-clique $(u, x, w, z, t') \in \mathcal{S}_P$ **do**
    if $(v, x) \in \mathcal{S}_e \wedge (v, w) \in \mathcal{S}_e \wedge (v, z) \in \mathcal{S}_e$ **then**
      $\tilde{p} \leftarrow \min\left\{1, \frac{M_P}{\tau}\right\} \left(\min\left\{1, \frac{M_e}{t-1}\right\}\right)^3 \left(\min\left\{1, \frac{M_e}{t'-1}\right\}\right)^5$ ▷ Approximate probability of detecting a 5-clique
      $\varkappa \leftarrow \varkappa + \tilde{p}^{-1}/2$ ▷ Estimate update accounting for two possible ways of detecting 5-cliques
  for each 4-clique $(v, x, w, z, t') \in \mathcal{S}_\Delta$ **do**
    if $(u, x) \in \mathcal{S}_e \wedge (u, w) \in \mathcal{S}_e \wedge (u, z) \in \mathcal{S}_e$ **then**
      $\tilde{p} \leftarrow \min\left\{1, \frac{M_P}{\tau}\right\} \left(\min\left\{1, \frac{M_e}{t-1}\right\}\right)^3 \left(\min\left\{1, \frac{M_e}{t'-1}\right\}\right)^5$
      $\varkappa \leftarrow \varkappa + \tilde{p}^{-1}/2$

function UPDATE4CLIQUES$((u, v), t)$
  $\mathcal{N}^{\mathcal{S}}_{u,v} \leftarrow \mathcal{N}^{\mathcal{S}}_u \cap \mathcal{N}^{\mathcal{S}}_v$
  for each pair (w, z) from $\mathcal{N}^{\mathcal{S}}_{u,v} \times \mathcal{N}^{\mathcal{S}}_{u,v}$ **do**
    if $(w, z) \in \mathcal{S}_e$ **then**
      $t_C \leftarrow t_C + 1$
      SAMPLE4CLIQUE(u, v, w, z, t)

where M_e (resp., M_P) denotes the memory space assigned to the edge-only (resp., the prototype/4-cliques) reservoir, t (resp., t') denotes the time step at which the last edge of the 5-clique (reps., prototype 4-clique being used) arrived on the stream, and $\tau^{(t)}$ denotes the number of 4-clique prototypes detected by TS5C up to time t. The approximate $\tilde{p}$ is obtained following the breakdown discussed in step (3) of Section 8.1:

- $\min\{1, \frac{M_P}{\tau}\}$ approximates the probability of the 4-clique prototype used to detect the occurrence of a 5-clique to be in $\mathcal{S}_P$ at time t conditioned on the fact that it was observed by TS5C;
- $(\min\{1, \frac{M_e}{t'-1}\})^5$ approximates the probability that the 4-clique prototype used to detect the occurrence of a 5-clique and whose last edge arrived on the stream at time t' was observed

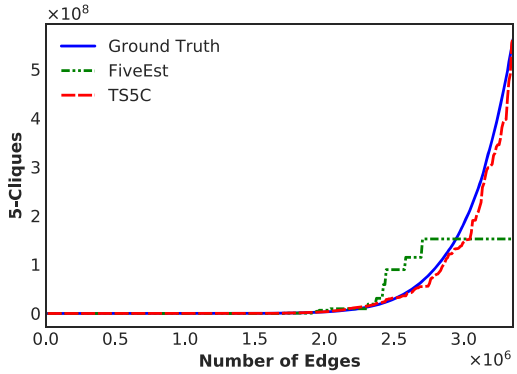


Fig. 8. Comparison of $|C_5^{(t)}|$ estimates for the DBLP graph obtained using TS5C and FiveEst with $M = 3 \times 10^5$. Out of 567,495,440 5-cliques that are present in the DBLP graph, the single reservoir algorithm FiveEst estimates a number of 153,979,072 and TS5C estimates of number of 565,173,908 5-cliques.

by TS5C (i., the probability that when the last edge of the 4-clique was observed on the stream the remaining 5 edges were included in $\mathcal{S}_e^{(t')}$);

— $(\min\{1, \frac{M_e}{t-1}\})^3$ approximates the probability that the remaining 3 edges used in the detection of the 5-clique are included in $\mathcal{S}_e^{(t)}$.

Other details of the construction of TS5C follow the blueprint outlined in the previous section. The pseudocode of TS5C is presented in Algorithm 5. The use of the approximate $\tilde{p}$ allows adapting the Tiered Sampling paradigm to the task of counting 5-cliques while considerably reducing the analysis effort. In contrast, the exact analysis would require a breakdown of a number of cases corresponding to all the possible ordering of arrival the 11 edges in a 5-clique.

While the use of approximate values of the probability of detecting instances of the pattern of interest leads to the estimate $\varkappa$ not being unbiased, in the following experimental evaluation, we show that the computed estimate is still very close to the ground truth value. Our experiments compare the performance of TS5C to that of a standard one-tier edge reservoir sample algorithm FiveEst (refer to Algorithm 6 in Appendix D), similar to FourEst. We evaluate the average performance over 10 runs of the two algorithms on the DBLP graph using $M = 3 \times 10^5$. The results are presented in Figure 8. TS5C clearly outperforms FiveEst, despite the latter providing an *unbiased estimate* (Theorem D.2), in obtaining much better estimations of the ground truth value $|C_5^{(t)}|$ achieving an $\sim 56\%$ reduction of the average MAPE.

9 CONCLUSIONS

In this work, we study the problem of counting sparse motifs in large-scale graphs streams using multi-layer (tiered) reservoir sampling. We developed Tiered Sampling, a novel technique for approximate counting sparse motifs in massive graphs whose edges are observed in a one-pass stream.

We fully analyze and demonstrate the advantage of our method in a specific application for counting the number the of 4-cliques in a graph using a two-sample approach. Through extensive experimental analysis, we show that the proposed algorithms produce high quality and low variance approximations for the number of 4-cliques for large graphs, both synthetic and real-world, with up to hundreds of millions of edges. We present both analytical proofs and experimental results, demonstrating the advantage of our method in counting sparse motifs compared to the standard methods of using just a single edge reservoir sample.

We present a simple process that allows generalizing the Tiered Sampling approach to provide estimates of the count for any subgraph of interest while considerably reducing the effort for the analysis by using opportune approximations. We showcase the effectiveness of this approach by presenting TS5C, an application of Tiered Sampling for counting the number of 5-cliques in a graph stream.

With the growing interest in discovering and analyzing large motifs in massive-scale graphs in social networks, genomics, and neuroscience, we expect to see further applications of our technique.

APPENDIX

A PROOFS OF TECHNICAL RESULTS FOR ALGORITHM TS4C$_1$

Lemma A.1. *Any pair (λ, γ) of distinct 4-cliques in $G^{(t)}$ can share either one, three or no edges. If λ and γ share three edges, those three edges compose a triangle.*

Proof. Suppose that λ and γ share exactly two distinct edges. This implies that they share at least three distinct vertices and, thus, must share the three edges connecting each pair out of said three vertices. This constitutes a contradiction. Suppose instead that λ and γ share four or five edges while being distinct. This implies that they must share four vertices, hence they cannot be distinct cliques. This leads to a contradiction. □

Lemma A.1, allows us to point out the various cases to be considered in the main proof of Theorem 4.4.

Proof of Theorem 4.4. Assume $|C_4^{(t)}| > 0$, otherwise TS4C$_1$ estimation is deterministically correct and has variance 0 and the thesis holds. For each $\lambda \in C_4^{(t)}$ let $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, without loss of generality let us assume the edges are disposed as in Figure 1. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. Let $t_{1,2,4} = \max\{t_1, t_2, t_4, M_e + 1\}$. Let us consider the random variable δ_{λ_1} (resp. δ_{λ_2}) which takes value $p_{\lambda_1}^{-1}/2$ (resp., $p_{\lambda_2}^{-1}/2$) if the 4-clique λ is observed by TS4C$_1$ using triangle $T_1 = \{e_1, e_2, e_4\}$ (resp., T_2) and edges e_3, e_5 (resp., e_2, e_4) or zero otherwise. Let p_{λ_1} (resp., p_{λ_2}) denote the probability of such event. From Lemma 4.2 we have:

$$\mathrm{Var}\left[\delta_{\lambda_1}\right] = \frac{p_{\lambda_1}^{-1}}{4} - \frac{1}{4} \leq \frac{1}{4}\left(\frac{\tau^{(t)}}{M_\Delta}\prod_{i=0}^{3}\frac{t-1-i}{M_e-i} - 1\right),$$

$$\mathrm{Var}\left[\delta_{\lambda_2}\right] = \frac{p_{\lambda_2}{}^{-1}}{4} - \frac{1}{4} \leq \frac{1}{4}\left(\frac{\tau^{(t)}}{M_\Delta}\prod_{i=0}^{3}\frac{t-1-i}{M_e-i} - 1\right).$$

Thus,

$$\begin{aligned}
\mathrm{Var}\left[\varkappa^{(t)}\right] &= \mathrm{Var}\left[\sum_{\lambda\in C_4^{(t)}} \delta_{\lambda_1}+\delta_{\lambda_2}\right] = \sum_{\lambda\in C_4^{(t)}}\sum_{\gamma\in C_4^{(t)}}\sum_{i\in\{1,2\}}\sum_{j\in\{1,2\}} \mathrm{Cov}\left[\delta_{\lambda_i},\delta_{\gamma_j}\right] \\
&= \sum_{\lambda\in C_4^{(t)}} \left(\mathrm{Var}\left[\delta_{\lambda_1}\right]+\mathrm{Var}\left[\delta_{\lambda_2}\right]\right) + \sum_{\lambda\in C_4^{(t)}} \left(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\lambda_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\lambda_1}\right]\right) \\
&\quad + \sum_{\substack{\lambda,\gamma\in C_4^{(t)}\\ \lambda\neq\gamma}} \left(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\gamma_1}\right]+\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\gamma_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\gamma_1}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\gamma_2}\right]\right) \\
&= \frac{|C_4^{(t)}|}{2}\left(\frac{\tau^{(t)}}{M_\Delta}\prod_{i=0}^{3}\frac{t-1-i}{M_e-i}-1\right) + \sum_{\lambda\in C_4^{(t)}} \left(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\lambda_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\lambda_1}\right]\right) \\
&\quad + \sum_{\substack{\lambda,\gamma\in C_4^{(t)}\\ \lambda\neq\gamma}} \left(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\gamma_1}\right]+\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\gamma_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\gamma_1}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\gamma_2}\right]\right) \\
&\leq \frac{|C_4^{(t)}|}{2}\left(\frac{\tau^{(t)}}{M_\Delta}c\left(\frac{t-1}{M_e}\right)^4-1\right) + \sum_{\lambda\in C_4^{(t)}} \left(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\lambda_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\lambda_1}\right]\right) \\
&\quad + \sum_{\substack{\lambda\in C_4^{(t)}\\ \lambda\neq\gamma}} \left(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\gamma_1}\right]+\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\gamma_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\gamma_1}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\gamma_2}\right]\right). \qquad (7)
\end{aligned}$$

We now proceed to analyze the various covariance terms appearing in Equation (7). In the following we refer to

—The second summation in Equation (7), $\sum_{\lambda\in C_4^{(t)}} \left(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\lambda_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\lambda_1}\right]\right)$, concerns the sum of the covariances of pairs of random variables each corresponding to one of the two possible ways of detecting a 4-clique using $\mathrm{TS4C}_1$. Let us consider one single element of the summation:

$$\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\lambda_2}\right] = \mathrm{E}\left[\delta_{\lambda_1}\delta_{\lambda_2}\right]-\mathrm{E}\left[\delta_{\lambda_1}\right]\mathrm{E}\left[\delta_{\lambda_2}\right] = \mathrm{E}\left[\delta_{\lambda_1}\delta_{\lambda_2}\right]-1/4.$$

Let us now focus on $\mathrm{E}\left[\delta_{\lambda_1}\delta_{\lambda_2}\right]$, according to the definition of δ_{λ_1} and δ_{λ_2} we have:

$$\begin{aligned}
\mathrm{E}\left[\delta_{\lambda_1}\delta_{\lambda_2}\right] &= \frac{p_{\lambda_1}^{-1}p_{\lambda_2}^{-1}}{4}\Pr\left(\delta_{\lambda_1}=p_{\lambda_1}^{-1}\wedge\delta_{\lambda_2}=p_{\lambda_2}^{-1}\right) \\
&= \frac{p_{\lambda_1}^{-1}p_{\lambda_2}^{-1}}{4}\Pr\left(\delta_{\lambda_1}=p_{\lambda_1}^{-1}|\delta_{\lambda_2}=p_{\lambda_2}^{-1}\right)\Pr\left(\delta_{\lambda_2}=p_{\lambda_2}^{-1}\right) \\
&\leq \frac{p_{\lambda_1}^{-1}p_{\lambda_2}^{-1}}{4}\Pr\left(\delta_{\lambda_2}=p_{\lambda_2}^{-1}\right) \\
&\leq \frac{p_{\lambda_1}^{-1}p_{\lambda_2}^{-1}}{4}p_{\lambda_2} \\
&\leq \frac{p_{\lambda_1}^{-1}}{4}.
\end{aligned}$$

We can therefore conclude:

$$\sum_{\lambda \in C_4^{(t)}} \left(\mathrm{Cov}\left[\delta_{\lambda_1}, \delta_{\lambda_2}\right] + \mathrm{Cov}\left[\delta_{\lambda_2}, \delta_{\lambda_1}\right] \right) \leq \frac{|C_4^{(t)}|}{2} \left(\frac{\tau^{(t)}}{M_\Delta} \prod_{i=0}^{3} \frac{t-1-i}{M_e - i} - 1 \right) \leq \frac{|C_4^{(t)}|}{2} \left(\frac{\tau^{(t)}}{M_\Delta} c \left(\frac{t-1}{M_e} \right)^4 - 1 \right). \tag{8}$$

—The third summation in Equation (7), includes the covariances of all $|C_4^{(t)}|(2|C_4^{(t)}| - 1)$ unordered pairs of random variables corresponding each to one of the two possible ways of counting distinct 4-cliques in $C_4^{(t)}$. In order to provide a significant bound it is necessary to divide the possible pairs of 4-cliques depending on how many edges they share (if any). From Lemma A.1, we have that any pair of 4-cliques λ and γ can share either one, three or no edges. In the remainder of our analysis, we shall distinguishing three group of pairs of 4-cliques based on how many edges they share. In the following, we present, without loss of generality, bounds for $\mathrm{Cov}[\delta_{\lambda_1}, \delta_{\gamma_2}]$. The results steadily holds for the other possible combinations $\mathrm{Cov}[\delta_{\lambda_1}, \delta_{\gamma_1}]$, $\mathrm{Cov}[\delta_{\lambda_2}, \delta_{\gamma_1}]$, $\mathrm{Cov}[\delta_{\lambda_2}, \delta_{\gamma_2}]$, $\mathrm{Cov}[\delta_{\gamma_1}, \delta_{\lambda_1}]$, $\mathrm{Cov}[\delta_{\gamma_1}, \delta_{\lambda_2}]$, $\mathrm{Cov}[\delta_{\gamma_2}, \delta_{\lambda_1}]$, and $\mathrm{Cov}[\delta_{\gamma_2}, \delta_{\lambda_2}]$

(1) λ and γ do not share any edge:

$$\mathrm{E}\left[\delta_{\lambda_1}\delta_{\gamma_2}\right] = \frac{p_{\lambda_1}^{-1}p_{\gamma_2}^{-1}}{4}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} \wedge \delta_{\gamma_2} = p_{\gamma_2}^{-1}\right) = \frac{p_{\lambda_1}^{-1}p_{\gamma_2}^{-1}}{4}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}\right)\Pr\left(\delta_{\lambda_2} = p_{\lambda_2}^{-1}\right).$$

The term $\Pr(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1})$ denotes the probability of TS4C$_1$ observing λ using T_1 and edges e_3 and e_5 conditioned of the fact that γ was observed by the algorithm using T_3 and edges g_3 and g_5. Note that as λ and γ do not share any edge, no edge of γ will be used by TS4C$_1$ to detect λ. Rather, if any edge of γ is included in $\S_e$ or if T_3 is included in $\mathcal{S}_\Delta$, this would lessen the probability of TS4C$_1$ detecting λ using $T1$, e_3 and e_5 as some of space in $\mathcal{S}_e$ or $\mathcal{S}_\Delta$ may be occupied by edges or triangle sub-structures for γ. Therefore we have $\Pr(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}) \leq \Pr(\delta_{\lambda_1} = p_{\lambda_1}^{-1})$ and thus:

$$\begin{aligned}
\mathrm{E}\left[\delta_{\lambda_1}\delta_{\gamma_2}\right] &= \frac{p_{\lambda_1}^{-1}p_{\gamma_2}^{-1}}{4}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}\right)\Pr\left(\delta_{\lambda_2} = p_{\lambda_2}^{-1}\right) \\
&\leq \frac{p_{\lambda_1}^{-1}p_{\gamma_2}^{-1}}{4}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1}\right)\Pr\left(\delta_{\lambda_2} = p_{\lambda_2}^{-1}\right) \\
&\leq \frac{p_{\lambda_1}^{-1}p_{\gamma_2}^{-1}}{4} p_{\lambda_1} p_{\lambda_2} \\
&\leq \frac{1}{4}.
\end{aligned}$$

We therefore have $\mathrm{Cov}[\delta_{\lambda_1}, \delta_{\gamma_2}] \leq 0$. Hence we can conclude that the contribution of the covariances of the pairs of random variables corresponding to 4-cliques that do not share any edge to the summation in Equation (7) is less or equal to zero.

(2) λ and γ share exactly one edge $e^* = \lambda \cap \gamma$ as shown in Figure 9 Let us consider the event $E* =$"$e^* \cap T_1 \cap E^{(t_{1,2,4}-1)} \in \mathcal{S}_e^{t_{1,2,4}}$, and $e^* \cap \{e_3, e_5\} \cap E^{(t_6-1)} \in S_e^{(t_6)}$". Clearly $\Pr(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}) \leq \Pr(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | E*)$. Recall from Lemma 4.2 that $\Pr(\delta_{\lambda_1} | E^*) = \Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*)\Pr(T_1 \in \mathcal{S}_\Delta^{(t_6)} | E*)\Pr(S(T_1) | E^*)$, where $S(T_1)$ denotes the event "T_1 is observed on the stream by TS4C$_1$." By applying the law of total probability e have that

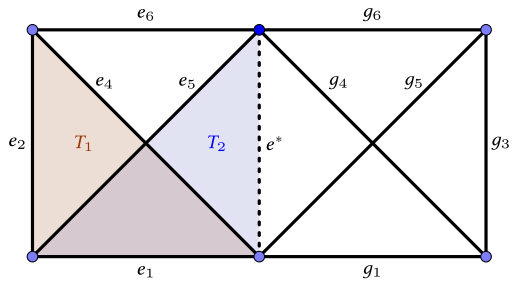


Fig. 9. Cliques sharing one edge.

$\Pr(T_1 \in \mathcal{S}_\Delta^{(t_6)}|E*) \leq \Pr(T_1 \in \mathcal{S}_\Delta^{(t_6)})$. The remaining two terms are influenced differently depending on whether $e^* \in T_1$ or $e^* \in \{e_3, e_5\}$:

—If $e^* \in T_1$: we then have $\Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)}|E^*) \leq \Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)})$. This follows from the properties of the reservoir sampling scheme as the fact that the edge e^* is in $\mathcal{S}_e$ means that one unit of the available memory space required to hold e_3 or e_5 is occupied, at least for some time, by e^*. If e^* is the last edge of T_1 observed in the stream we then have:

$$\begin{aligned}\Pr\left(S(T_1)|E^*\right) &= \Pr\left(\{e_1, e_2, e_4\} \setminus \{e^*\} \in \mathcal{S}_e^{(}t_1, 2, 4)|E^*\right)\\ &= \Pr\left(\{e_1, e_2, e_4\} \setminus \{e^*\} \in \mathcal{S}_e^{(}t_1, 2, 4)\right)\\ &\leq \frac{M_e}{t_{1,2,4}-1}\frac{M_e-1}{t_{1,2,4}-2}.\end{aligned}$$

Suppose instead that $e*$ is *not* the last edge of T_1. Assume further , without loss of generality, that $t_2 > \max\{t_1, t_4\}$. TS4C$_1$ observes T_1 iff $\{e_1, e_4\} \in \mathcal{S}_e^{(}t_2)$. As in E^* we assume that once observed e^* is *always* maintained in $\mathcal{S}_e$ until $t_{1,2,4}$ we have:

$$\begin{aligned}\Pr\left(S(T_1)|E^*\right) &= \Pr\left(\{e_1, e_4\} \setminus \{e^*\} \in \mathcal{S}_e^{(}t_1, 2, 4)|E^*\right)\\ &\leq \frac{M_e-1}{t_{1,2,4}-2}.\end{aligned}$$

—If $e^* \in \{e_3, e_5\}$: we then have $\Pr\left(S(T_1)|E^*\right) \leq \Pr\left(S(T_1)\right)$. This follows from the properties of the reservoir sampling scheme as the fact that the edge e^* is in $\mathcal{S}_e$ means that one unit of the available memory space required to hold the first two edges of T_1 until $t_{1,2,4}$ is occupied, at least for some time, by e^*. Further, using Lemma 4.2 we have:

* if $t_6 \leq M_e$, then $\Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)}|E^*) = 1$;
* if $\min\{t_3, t_5\} > t_{1,2,4}$, then $\Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)}|E^*) \geq \frac{M_e}{t_6-1}$;
* if $\max\{t_3, t_5\} > t_{1,2,4} > \min\{t_3, t_5\}$, then

$$\Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)}|E^*\right) \geq \max\left\{\frac{M_e-1}{t_6-2}, \frac{M_e-2}{t_{1,2,4}-3}\frac{t_{1,2,4}-1}{t_6-1}\right\};$$

* otherwise $\Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)}|E^*) \geq \frac{M_e-2}{t_{1,2,4}-3}\frac{t_{1,2,4}-1}{t_6-1}$.

Putting together these results we have that

$$p_\lambda^{(-1)}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1}|\delta_{\gamma_2} = p_{\gamma_2}^{-1}\right) \leq p_\lambda^{(-1)}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1}|E*\right) \leq c\frac{t_6-1}{M_e},$$

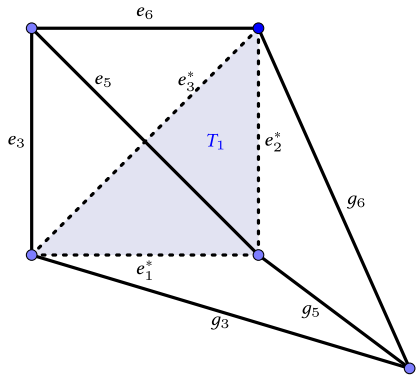


Fig. 10. Cliques sharing three edges.

with $c = s$. Hence we have $\mathrm{E}[\delta_{\lambda_1}\delta_{\gamma_2}] \leq \frac{c}{4}\frac{t_6-1}{M_e} \leq \frac{c}{4}\frac{t-1}{M_e}$. We can thus bound the contribution to the third component of Equation (7) given by the pairs of random variables corresponding to 4-cliques that share one edge as:

$$2a^{(t)}\left(c\frac{t-1}{M_e}-1\right), \tag{9}$$

where $a^{(t)}$ denotes the number of unordered pairs of 4-cliques which share one edge in $G^{(t)}$.

(3) λ and γ share three edges $\{e_1^*, e_2^*, e_3^*\}$ which form a triangle sub-structure for both λ and γ. Let us refer to Figure 10, without loss of generality let T_1 denote the triangle shared between the two cliques. We distinguish the kind of pairs for the random variables δ_{λ_i} and δ_{γ_j} cases:

— $\delta_{\lambda_i} = p_{\lambda_i}^{-1}$ if $T_1 \in \mathcal{S}_\Delta^{t_6-1} \wedge \{e_3, e_5\} \subseteq \mathcal{S}_e^{t_6-1}$ and $\delta_{\gamma_j} = p_{\gamma_j}^{-1}$ if $T_1 \in \mathcal{S}_\Delta^{t_\gamma-1} \wedge \{g_3, g_5\} \subseteq \mathcal{S}_e^{t_\gamma-1}$, where t_γ denotes the time step at which the last edge of γ is observed. This is the case for which the random variables δ_{λ_i} and δ_{γ_j} corresponds to TS4C$_1$ observing λ and γ using the *shared triangle* T_1. Let us consider the event $E* =$"$T_1 \in \mathcal{S}_\Delta^{(t_6)}$. Clearly $\Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1}) \leq \Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*)$. In this case we have $\Pr(T_1 \in \mathcal{S}^{(t_6)} | E^*) \leq 1$, while $\Pr(\{e_3, e_5\} \in \mathcal{S}_e^{(t_6)} | E^*) \leq Prob\{e_3, e_5\} \in \mathcal{S}_e^{(t_6)}$. This second fact follows from the properties of the reservoir sampling scheme as the fact that the edges e_1^*, e_2^* and e_3^* are in $\mathcal{S}_e$ *at least* for the time required for T_1 to be observed, means that at least two unit of the available memory space required to hold the edges e_3, e_5 are occupied, at least for some time. Putting together these results we have that $p_{\lambda_i}^{(-1)} Prob\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1} \leq p_{\lambda_i}^{(-1)} \Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*) \leq c(\frac{t_6-1}{M_e})^2 \frac{\tau^{(t)}}{\mathcal{S}_\Delta}$. Hence we have $\mathrm{E}[\delta_{\lambda_i}\delta_{\gamma_j}] \leq \frac{c}{4}(\frac{t_6-1}{M_e})^2 \frac{\tau^{(t)}}{\mathcal{S}_\Delta}$ and $\mathrm{Cov}[\delta_{\lambda_i}, \delta_{\gamma_j}] \leq \frac{c}{4}(\frac{t_6-1}{M_e})^2 \frac{\tau^{(t)}}{\mathcal{S}_\Delta} - \frac{1}{4}$.

— In all of the remaining cases, the random variables δ_{λ_i} and δ_{γ_j} correspond to FourEst not observing λ and γ using the shared triangle T_1 for both of them. Let T^* denote the triangle sub-structure used by TS4C$_1$ to count λ with respect to δ_{λ_i}. Let us consider the event $E^* =$"$\{e_1^*, e_2^*, e_3^*\} \cap T_1 \cap E^{(t_{1,2,4}-1)} \in \mathcal{S}_e^{t_{1,2,4}}$, $T_1 \in \mathcal{S}_\Delta^{(t_6)}$ unless one of its edges is the last edge of T^* observed on the stream, and $\{e_1^*, e_2^*, e_3^*\} \cap \{e_3, e_5\} \cap E^{(t_6-1)} \in S_e^{(t_6)}$ if $e^* \in \{e_3, e_5\}$", where $E^{(t)}$ denotes the set of edges observed up until time t included. Clearly $\Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1}) \leq \Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*)$. Note that in

this case $|\{e_1^*, e_2^*, e_3^*\} \cap T_1| + |\{e_1^*, e_2^*, e_3^*\} \cap \{e_3, e_5\}|$. By analyzing $\Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*)$ in this case using similar steps as the ones described for the other sub-cases we have $p_{\lambda_i}^{(-1)} \Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1}) \leq p_{\lambda_i}^{(-1)} \Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*) \leq c(\frac{t_6-1}{M_e})^3$. Hence we have $\mathrm{E}[\delta_{\lambda_i} \delta_{\gamma_j}] \leq \frac{c}{4}(\frac{t_6-1}{M_e})^3$ and

$$\mathrm{Cov}\left[\delta_{\lambda_i}, \delta_{\gamma_j}\right] \leq \frac{c}{4}\left(\frac{t_6-1}{M_e}\right)^3 - \frac{1}{4}.$$

We can thus bound the contribution to the third component of Equation (7) given by the pairs of random variables corresponding to 4-cliques that share three edges as:

$$2b^{(t)}\left(c\left(\frac{t-1}{M_e}\right)^2\left(\frac{1}{4}\frac{\tau^{(t)}}{M_\Delta} + \frac{3}{4}\frac{t-1}{M_e}\right) - 1\right), \tag{10}$$

where $b^{(t)}$ denotes the number of unordered pairs of 4-cliques which share three edges in $G^{(t)}$.

The Theorem follows by combining Equations (8), (9), and (10) in Equation (7). □

B PROOFS OF TECHNICAL RESULTS REGARDING TS4C$_2$

In this section, we present proofs for algorithm TS4C$_2$ discussed in Section 4.2.

Proof of Theorem 4.7. Let us define the event $E_\lambda = \lambda$ *is observed* on the stream by TS4C$_2$ using triangle $T_1 = \{e_1, e_2, e_4\}$ and triangle $T_2 = \{e_1, e_3, e_5\}$. Given the definition of TS4C$_2$ we have:

$$E_\lambda = E_{T_1} \wedge E_{T_2},$$

and hence:

$$p_\lambda = \Pr(E_\lambda) = \Pr(E_{T_1} \wedge E_{T_2}) = \Pr(E_{T_1} | E_{T_2}) \Pr(E_{T_2}).$$

In order to study $\Pr(E_{T_2})$, we shall introduce event $E_{S(T_2)}$ ="*triangle* T_2 *is observed on the stream by TS4C*$_2$." From the definition of TS4C$_2$, we know that T_2 is observed on the stream iff when the last edge of T_2 is observed on the stream at $\max\{t_1, t_3, t_5\}$ the remaining to edges are in the edge sample. Applying Bayes's rule of total probability we have:

$$\Pr(E_{T_2}) = \Pr\left(E_{T_2} | E_{S(T_2)}\right) \Pr\left(E_{S(T_2)}\right),$$

and thus:

$$p_\lambda = \Pr(E_{T_1} | E_{T_2}) \Pr\left(E_{T_2} | E_{S(T_2)}\right) \Pr\left(E_{S(T_2)}\right). \tag{11}$$

In order for T_2 to be observed by $TS4c_2$ it is required that when the last edge of T_2 is observed on the stream at $t^M_{1,3,5}$ its two remaining edges are kept in S_e. From Lemma 2.1, we have:

$$\Pr\left(E_{S(T_2)}\right) = \frac{M_e}{t_{1,3,5}-1}\frac{M_e-1}{t_{1,3,5}-2}, \tag{12}$$

and:

$$\Pr\left(E_{T_2} | E_{S(T_2)}\right) = \frac{M_e}{\tau^{t_6}}. \tag{13}$$

Let us now consider $\Pr(E_{T_1} | E_{T_2})$. In order for T_1 to be found in $\mathcal{S}_\Delta$ at t_6 it is necessary for T_1 to have been observed by TS4C$_2$. Thus, we have that $\Pr(E_{T_1} | E_{T_2}) = \Pr(E_{T_1} | E_{S(T_1)}, E_{T_2}) \Pr(E_{S(T_1)} | E_{T_2})$

$$\Pr\left(E_{T_1} | E_{S(T_1)}\right) = \frac{M_\Delta - 1}{\tau^{t_6} - 1}. \tag{14}$$

Let us now consider $\Pr\left(E_{S(T_1)} | E_{T_2}\right)$. While the content of $\mathcal{S}_\Delta$ itself does not influence the content of $\mathcal{S}_e$, the fact that T_2 is maintained in $\mathcal{S}_\Delta$ at t_6 implies that it has been observed on

the stream at a previous time. We thus have $\Pr(E_{S(T_1)}|E_{T_2}) = \Pr(E_{S(T_1)}|E_{S(T_2)})$. In order to study $p' = \Pr(E_{S(T_1)}|E_{S(T_2)})$, it is necessary to distinguish the possible (5!) different arrival orders for edges e_1,e_2, e_3, e_4 and e_5. With an efficient analysis, we however reduce the number of cases to be considered to 13.

— $t^M_{1,2,4} \leq M_e$: in this case, all edges of T_1 are observed on the stream before M_e so they are deterministically inserted in $\mathcal{S}_e$ and, thus, $p' = 1$;
— $t_1 > t^M_{2,3,4,5}$: in this case the edge e_1 that is shared by T_1 and T_2 is observed after e_2, e_3, e_4, e_5.

$$\begin{aligned} p' &= P(\{e_2, e_4\} \in {\mathcal{S}_e}^{(t_1)} | \{e_3, e_5\} \in {\mathcal{S}_e}^{(t_1)}) \\ &= P(e_2 \in \mathcal{S}_e(t_1) | \{e_3, e_4, e_5\} \in {\mathcal{S}_e}^{(t_1)}) \cdot P(e_4 \in {\mathcal{S}_e}^{(t_1)} | \{e_3, e_5\} \in {\mathcal{S}_e}^{(t_1)}) \\ &= min\left(1, \frac{M-3}{t_1-4}\right) \cdot min\left(1, \frac{M-2}{t_1-3}\right). \end{aligned}$$

— $t^M_{3,5} > t_1 > \max\{t^m_{3,5}, t_2, t_4\}$: in this case, only one of the edges e_3 and e_5 is observed after e_1, which is itself observed after all the remaining edges. Here we consider the case when e_3 is the edge to be observed last. The same considerations follows for e_5 as well.

$$\begin{aligned} p' &= P(\{e_2, e_4\} \in {\mathcal{S}_e}^{(t_1)} | e_5 \in {\mathcal{S}_e}^{(t_1)}) \\ &= P(e_2 \in \mathcal{S}_e(t_1) | \{e_4, e_5\} \in {\mathcal{S}_e}^{(t_1)}) \cdot P(e_4 \in {\mathcal{S}_e}^{(t_1)} | e_5 \in {\mathcal{S}_e}^{(t_1)}) \\ &= min\left(1, \frac{M-2}{t_1-3}\right) \cdot min\left(1, \frac{M-1}{t_1-2}\right). \end{aligned}$$

— $t^M_{3,5} > t^M_{2,4} > \max\{t^m_{3,5}, t^m_{2,4}, t_1\}$: in this case only, one of the edges e_3 and e_5 is observed after one of the edges e_2 and e_4, which is itself observed after all the remaining edges. We consider the case when e_3 and e_2 are observed last. The same procedure follows for e_4 and e_5 as well.

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in {\mathcal{S}_e}^{(t_2)} | \{e_1, e_5\} \in {\mathcal{S}_e}^{(t_3)}) \\ &= P(e_1 \in \mathcal{S}_e(t_2) | \{e_1, e_4, e_5\} \in {\mathcal{S}_e}^{(t_3)}) \cdot P(e_4 \in {\mathcal{S}_e}^{(t_2)} | \{e_1, e_5\} \in {\mathcal{S}_e}^{(t_3)}) \\ &= 1 \cdot min\left(1, \frac{M-2}{t_2-3}\right). \end{aligned}$$

— $t^m_{3,5} > t_1 > t^M_{2,4}$: in this case, both edges e_3 and e_5 are observed after e_1, which is itself observed after all the remaining edges. Here we consider the case when $t_3 > t_5 > t_1$. The same procedure follows for the case $t_5 > t_3 > t_1$.

$$\begin{aligned} p' &= P(\{e_2, e_4\} \in {\mathcal{S}_e}^{(t_1)} | \{e_1, e_5\} \in {\mathcal{S}_e}^{(t_3)}) \\ &= P(e_2 \in \mathcal{S}_e(t_1) | e_4 \in \mathcal{S}_e(t_1), \{e_1, e_5\} \in {\mathcal{S}_e}^{(t_3)}) \cdot P(e_4 \in {\mathcal{S}_e}^{(t_1)} | \{e_1, e_5\} \in {\mathcal{S}_e}^{(t_3)}) \\ &= min\left(1, \frac{M-1}{t_1-2}\right) \cdot min\left(1, \frac{M}{t_1-1}\right). \end{aligned}$$

— $t^m_{3,5} > t^M_{2,4} > t_1$: in this case, both edges e_3 and e_5 are observed after one of the edges e_2 and e_4, which is observed after e_1. We consider the case $t_2 > t_4$ and $t_3 > t_5$. The same procedure

follows for $t_4 > t_2$ and $t_5 > t_3$

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= P(e_4 \in \mathcal{S}_e(t_2) | e_1 \in \mathcal{S}_e(t_2), \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_1 \in \mathcal{S}_e{}^{(t_3)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= min\left(1, \frac{M-1}{t_2-2}\right) \cdot 1. \end{aligned}$$

– $t^M_{2,4} > t_1 > \max\{M_e, t^m_{2,4}, t^M_{3,5}\}$: in this case, both edges e_2 and e_4 are observed after e_1, which is observed after the edge reservoir is filled and after all the remaining edges. We consider the case when $t_2 > t_4$. The same procedure follows for $t_4 > t_2$.

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \\ &= P(e_4 \in \mathcal{S}_e(t_2) | e_1 \in \mathcal{S}_e(t_2), \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_1 \in \mathcal{S}_e{}^{(t_2)} | \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \\ &= \frac{M-1}{t_2-2} \cdot \frac{M-2}{t_1-3} \cdot \frac{t_1-1}{t_2-1}. \end{aligned}$$

– $t^M_{2,4} > M_e > t_1 > \max\{t^m_{2,4}, t^M_{3,5}\}$: in this case, both edges e_2 and e_4 are observed after e_1, which is observed before the edge reservoir is filled and after all the remaining edges. We consider the case when $t_2 > t_4$. The same procedure follows for $t_4 > t_2$.

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \\ &= P(e_4 \in \mathcal{S}_e(t_2) | e_1 \in \mathcal{S}_e(t_2), \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_1 \in \mathcal{S}_e{}^{(t_2)} | \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \\ &= \frac{M-1}{t_2-2} \cdot \frac{M_e}{t_2-1}. \end{aligned}$$

– $t^M_{2,4} > t^M_{3,5} > \max\{M_e, t^m_{3,5}, t^m_{2,4}, t_1\}$: in this case, only one of the edges e_2 and e_4 is observed after one of the edges e_3 and e_5, which is observed after the edge reservoir if filled and after all the remaining edges. We consider the case when $t_2 > t_4$ and $t_3 > t_5$. The same procedure follows for $t_4 > t_2$ and $t_5 > t_3$

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= P(e_1 \in \mathcal{S}_e(t_2) | e_4 \in \mathcal{S}_e(t_2), \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_4 \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= \frac{t_3-1}{t_2-1} \cdot \frac{t_3-2}{t_2-2} \cdot \frac{M_e-2}{t_3-3}. \end{aligned}$$

– $t^M_{2,4} > M_e > t^M_{3,5} > \max\{t^m_{3,5}, t^m_{2,4}, t_1\}$: in this case, only one of the edges e_2 and e_4 is observed after one of the edges e_3 and e_5, which is observed before the edge reservoir if filled and after all the remaining edges. We consider the case when $t_2 > t_4$ and $t_3 > t_5$. The same procedure follows for $t_4 > t_2$ and $t_5 > t_3$

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= P(e_1 \in \mathcal{S}_e(t_2) | e_4 \in \mathcal{S}_e(t_2), \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_4 \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= \frac{M_e}{t_2-1} \cdot \frac{M_e-1}{t_2-2}. \end{aligned}$$

– $t^m_{2,4} > t_1 > t^M_{3,5}$: in this case, both edges e_2 and e_4 are observed after e_1, which is observed after all the remaining edges. Here we consider the case when $t_2 > t_4 > t_1$. The same procedure

follows for $t_4 > t_2 > t_1$.

$$\begin{aligned} p' &= P(\{e_2, e_4\} \in \mathcal{S}_e{}^{(t_1)} | \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \\ &= P(e_2 \in \mathcal{S}_e(t_1) | e_4 \in \mathcal{S}_e(t_1), \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \cdot P(e_4 \in \mathcal{S}_e{}^{(t_1)} | \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \\ &= min\left(1, \frac{M_e - 1}{t_2 - 2}\right) \cdot min\left(1, \frac{M_e}{t_2 - 1}\right). \end{aligned}$$

— $t^m_{2,4} > t^M_{3,5} > \max\{M, t_1\}$: in this case, both edges e_2 and e_4 are observed after one of the edges e_3 and e_5, which is observed after the edge reservoir is filled and after all the remaining edges.

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= P(e_4 \in \mathcal{S}_e(t_2) | e_1 \in \mathcal{S}_e(t_2), \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_1 \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= \frac{M_e - 1}{t_2 - 2} \cdot \frac{t_3 - 1}{t_2 - 1}. \end{aligned}$$

— $t^m_{2,4} > M_e > t^M_{3,5} > t_1$: in this case, both edges e_2 and e_4 are observed after one of the edges e_3 and e_5, which is observed before the edge reservoir is filled and after all the remaining edges.

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= P(e_4 \in \mathcal{S}_e(t_2) | e_1 \in \mathcal{S}_e(t_2), \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_1 \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= \frac{M_e - 1}{t_2 - 2} \cdot \frac{M_e}{t_2 - 1}. \end{aligned}$$

The lemma follows combining the result for the values of p' with Equations (12), (13), and (14) in (B). □

Applying this lemma to an analysis similar to the one used in the proof of Theorem 4.3, we prove that the estimations obtained using TS4C$_2$ are *unbiased*.

Proof of Theorem 4.8. Let t^* denote the first step at which the number of triangles seen exceeds M_Δ. For $t \leq \min\{M_e, t^*\}$, the entire graph $G^{(t)}$ is maintained in $\mathcal{S}_e$ and all the triangles in $G^{(t)}$ are stored in $\mathcal{S}_\Delta$. Hence all the cliques in $G^{(t)}$ are deterministically observed by TS4C$_1$and we have $\varkappa^{(t)} = |C_4^{(t)}|$.

Assume now $t > \min\{M_e, t^*\}$ and assume that $|C_4^{(t)}| > 0$. Otherwise, the algorithm deterministically returns 0 as an estimation, and the statement trivially follows. For any 4-clique $\lambda \in C_4^{(t)}$ which is observed by TS4C$_2$ with probability p_λ, consider a random variable X_λ which takes value p^{-1} iff λ is actually observed by TS4C$_2$ (i.e., with probability p_λ) or zero otherwise. We thus have $\mathbb{E}[X_\lambda] = p_\lambda^{-1}\Pr(X_\lambda = p_\lambda^{-1}) = p_\lambda^{-1}p_\lambda = 1$. Recall that every time TS4C$_2$ observes a 4-clique on the stream it evaluates the probability p (according its correct value as shown in Lemma 4.7) of observing it and it correspondingly increases the running estimator by p^{-1}. We therefore can express the running estimator $\varkappa^{(t)}$ as:

$$\varkappa^{(t)} = \sum_{\lambda \in C_4^{(t)}} X_\lambda.$$

From linearity of expectation, we thus have:

$$\mathbb{E}\left[\varkappa^{(t)}\right] = \sum_{\lambda \in C_4^{(t)}} \mathbb{E}[X_\lambda] = \sum_{\lambda \in C_4^{(t)}} p_\lambda^{-1} p_\lambda = |C_4^{(t)}|.$$

□

The following proof for Theorem 4.9 follows a structure similar to the one presented for Theorem 4.4. However, the two proofs differ reflecting the different way according to which 4-cliques are detected by TS4C$_2$ compared to TS4C$_1$.

Proof of Theorem 4.9. Assume $|C_4^{(t)}| > 0$, otherwise TS4C$_1$ estimation is deterministically correct and has variance 0 and the thesis holds. For each $\lambda \in C_4^{(t)}$ let $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, without loss of generality, let us assume the edges are disposed as in Figure 1. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. Let $t_{1,2,4} = \max\{t_1, t_2, t_4, M_e + 1\}$. Let us consider the random variable δ_λ which takes value p_λ^{-1} if the 4-clique λ is observed by TS4C$_2$ using triangles $T_1 = \{e_1, e_2, e_4\}$, $T_1 = \{e_1, e_3, e_5\}$ and the final edge e_6, or zero otherwise. Let p_λ denote the probability of such event.

Since, from Lemma 4.2 we know:

$$\mathrm{Var}\left[\delta_\lambda\right] = \mathrm{E}\left[\delta_\lambda^2\right] - \mathrm{E}\left[\delta_\lambda\right]^2 = p_\lambda - 1 - 1 \leq \prod_{i=0}^{1} \frac{\tau^{(t)} - 1}{M_\Delta - 1} \prod_{i=0}^{3} \frac{t - 1 - i}{M_e - i} - 1.$$

For the estimator $\varkappa^{(t)}$ maintained by TS4C$_2$ we thus have:

$$\begin{aligned}
\mathrm{Var}\left[\varkappa^{(t)}\right] &= \mathrm{Var}\left[\sum_{\lambda \in C_4^{(t)}} \delta_\lambda\right] = \sum_{\lambda \in C_4^{(t)}} \sum_{\gamma \in C_4^{(t)}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] \\
&= \sum_{\lambda \in C_4^{(t)}} \mathrm{Var}\left[\delta_\lambda\right] + \sum_{\substack{\lambda,\, \gamma \in C_4^{(t)} \\ \lambda \neq \gamma}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] \\
&\leq |C_4^{(t)}| \left(\prod_{i=0}^{1} \frac{\tau^{(t)} - i}{M_\Delta - i} \prod_{i=0}^{3} \frac{t - 1 - i}{M_e - i} - 1 \right) + \sum_{\substack{\lambda,\, \gamma \in C_4^{(t)} \\ \lambda \neq \gamma}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right]. \qquad (15)
\end{aligned}$$

We now focus on the analysis of the sum of covariances in the right-hand side of the previous inequality. From the definition of covariance, we have:

$$\mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] = \mathrm{E}\left[\delta_\lambda \delta_\gamma\right] - \mathrm{E}\left[\delta_\lambda\right] \mathrm{E}\left[\delta_\gamma\right] = \mathrm{E}\left[\delta_\lambda \delta_\gamma\right] - 1,$$

where the last passage follows as, by construction, for all $\lambda \in C_4^{(t)}$ we have $\mathrm{E}\left[\delta_\lambda\right] = 1$. To complete the analysis of the covariance we will therefore consider the term $\mathrm{E}[\delta_\lambda \delta_\gamma]$. We have:

$$\begin{aligned}
\mathrm{E}\left[\delta_\lambda \delta_\gamma\right] &= p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} \wedge \delta_\gamma = p_\gamma^{-1}\right) \\
&= p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_{\lambda_1} = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \Pr\left(\delta_\lambda = p_\lambda^{-1}\right) \\
&= p_\lambda^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right). \qquad (16)
\end{aligned}$$

In order to conclude our analysis it will therefore be necessary to study the probability according to which TS4C$_2$ observes λ *conditioned* on the fact that γ was observed. We divide the possible pairs of 4-cliques depending on how many edges they share (if any). From Lemma A.1 we have that any pair of 4-cliques λ and γ can share either one, three or no edges, hence:

(1) λ and γ do not share any edge: As λ and γ do not share any edge, no edge of γ will be used by TS4C$_2$ to detect λ. Rather, if any of the edges (resp., triangles) of γ is included in $\S_e$ (resp., $\mathcal{S}_\Delta$), this would lessen the probability of TS4C$_2$ detecting λ as some of space in

$\mathcal{S}_e$ or $\mathcal{S}_\Delta$ may be occupied by edges or triangle sub-structures for γ. Therefore we have $\Pr(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}) \leq \Pr(\delta_\lambda = p_\lambda^{-1})$ and, thus, from Equation (16):

$$\mathrm{E}\left[\delta_\lambda \delta_\gamma\right] = p_\lambda^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \leq p_\lambda^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1}\right) = 1.$$

Therefore, we have $\mathrm{Cov}[\delta_\lambda, \delta_\gamma] \leq 0$, and we can conclude that the contribution of the covariances of the pairs of random variables corresponding to 4-cliques that do not share any edge to the summation in Equation (15) is less or equal to zero.

(2) λ and γ share exactly one edge $e^* = \lambda \cap \gamma$ as shown in Figure 9. Note first of all that if the shared edge is the last to be observed on the stream for either λ or *gamma*, then the same considerations presented for the case in which λ and γ do not share any edge apply an hence, $\mathrm{Cov}[\delta_\lambda, \delta_\gamma] \leq 0$.

In the following, we assume that e^* is not the last edge observed on the stream for neither λ nor γ. Let us consider the event $E* =$"$e^* \cap T_1 \cap E^{(t_{1,2,4}-1)} \in \mathcal{S}_e^{t_{1,2,4}}$, or $e^* \cap T_2 \cap E^{(t_{1,3,5}-1)} \in \mathcal{S}_e^{t_{1,3,5}}$". Clearly $\Pr(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}) \leq \Pr(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | E*)$. Recall from Lemma 4.2 that $\Pr(\delta_{\lambda_1} | E^*) = \Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*) \Pr(T_1 \in \mathcal{S}_\Delta^{(t_6)} | E*) \Pr(S(T_1) | E^*)$, where $S(T_1)$ denotes the event "T_1 is observed on the stream by TS4C$_1$". By applying the law of total probability e have that $\Pr(T_1 \in \mathcal{S}_\Delta^{(t_6)} | E*) \leq \Pr(T_1 \in \mathcal{S}_\Delta^{(t_6)})$. The remaining two terms are influenced differently depending on whether $e^* \in T_1$ or $e^* \in \{e_3, e_5\}$:

— If $e^* \in T_1$: we then have $\Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*) \leq \Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)})$. This follows from the properties of the reservoir sampling scheme as the fact that the edge e^* is in $\mathcal{S}_e$ means that one unit of the available memory space required to hold e_3 or e_5 is occupied, at least for some time, by e^*. If e^* is the last edge of T_1 observed in the stream we then have:

$$\begin{aligned}\Pr\left(S(T_1) | E^*\right) &= \Pr\left(\{e_1, e_2, e_4\} \setminus \{e^*\} \in \mathcal{S}_e^{(}t_1, 2, 4) | E^*\right) \\ &= \Pr\left(\{e_1, e_2, e_4\} \setminus \{e^*\} \in \mathcal{S}_e^{(}t_1, 2, 4)\right) \\ &\leq \frac{M_e}{t_{1,2,4} - 1} \frac{M_e - 1}{t_{1,2,4} - 2}.\end{aligned}$$

Suppose instead that $e*$ is *not* the last edge of T_1. Assume further , without loss of generality, that $t_2 > \max\{t_1, t_4\}$. TS4C$_1$ observes T_1 iff $\{e_1, e_4\} \in \mathcal{S}_e^{(}t_2)$. As in E^* we assume that once observed e^* is *always* maintained in $\mathcal{S}_e$ until $t_{1,2,4}$ we have:

$$\Pr\left(S(T_1) | E^*\right) = \Pr\left(\{e_1, e_4\} \setminus \{e^*\} \in \mathcal{S}_e^{(}t_1, 2, 4) | E^*\right) \leq \frac{M_e - 1}{t_{1,2,4} - 2}.$$

— If $e^* \in \{e_3, e_5\}$: we then have $\Pr\left(S(T_1) | E^*\right) \leq \Pr\left(S(T_1)\right)$. This follows from the properties of the reservoir sampling scheme as the fact that the edge e^* is in $\mathcal{S}_e$ means that one unit of the available memory space required to hold the first two edges of T_1 until $t_{1,2,4}$ is occupied, at least for some time, by e^*. Further, using Lemma 4.2 we have:
  — if $t_6 \leq M_e$, then $\Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*) = 1$;
  — if $\min\{t_3, t_5\} > t_{1,2,4}$, then $\Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*) \geq \frac{M_e}{t_6 - 1}$;
  — if $\max\{t_3, t_5\} > t_{1,2,4} > \min\{t_3, t_5\}$, then $\Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*) \geq \max\{\frac{M_e - 1}{t_6 - 2}, \frac{M_e - 2}{t_{1,2,4} - 3} \frac{t_{1,2,4} - 1}{t_6 - 1}\}$; and
  — otherwise, $\Pr(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*) \geq \frac{M_e - 2}{t_{1,2,4} - 3} \frac{t_{1,2,4} - 1}{t_6 - 1}$.

Putting together these various results we have that

$$p_\lambda^{(-1)}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}\right) \leq p_\lambda^{(-1)}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | E*\right) \leq c\frac{t_6 - 1}{M_e},$$

with $c = s$. Hence we have $\mathrm{E}[\delta_{\lambda_1}\delta_{\gamma_2}] \leq \frac{c}{4}\frac{t_6-1}{M_e} \leq \frac{c}{4}\frac{t-1}{M_e}$. We can thus bound the contribution to the third component of Equation (7) given by the pairs of random variables corresponding to 4-cliques that share one edge as:

$$2a^{(t)}\left(c\frac{t-1}{M_e} - 1\right), \tag{17}$$

where $a^{(t)}$ denotes the number of unordered pairs of 4-cliques which share one edge in $G^{(t)}$.

(3) λ and γ share three edges $\{e_1^*, e_2^*, e_3^*\}$ which form a triangle sub-structure for both λ and γ. Let us refer to Figure 10, without loss of generality let T_1 denote the triangle shared between the two cliques. We distinguish the kind of pairs for the random variables δ_{λ_i} and δ_{γ_j} cases:

— $\delta_{\lambda_i} = p_{\lambda_i}^{-1}$ if $T_1 \in \mathcal{S}_\Delta^{t_6-1} \wedge \{e_3, e_5\} \subseteq \mathcal{S}_e^{t_6-1}$ and $\delta_{\gamma_j} = p_{\gamma_j}^{-1}$ if $T_1 \in \mathcal{S}_\Delta^{t_\gamma-1} \wedge \{g_3, g_5\} \subseteq \mathcal{S}_e^{t_\gamma-1}$, where t_γ denotes the time step at which the last edge of γ is observed. This is the case for which the random variables δ_{λ_i} and δ_{γ_j} corresponds to FourEst observing λ and γ using the *shared triangle* T_1. Let us consider the event $E* =$"$T_1 \in \mathcal{S}_\Delta^{(t_6)}$. Clearly, $\Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1}) \leq \Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*)$. In this case we have $\Pr(T_1 \in \mathcal{S}^{(t_6)} | E^*) \leq 1$, while $\Pr(\{e_3, e_5\} \in \mathcal{S}_e^{(t_6)} | E^*) \leq \Pr(\{e_3, e_5\} \in \mathcal{S}_e^{(t_6)})$. This second fact follows from the properties of the reservoir sampling scheme as the fact that the edges e_1^*, e_2^*, and e_3^* are in $\mathcal{S}_e$ *at least* for the time required for T_1 to be observed, means that at least two unit of the available memory space required to hold the edges e_3 and e_5 are occupied, at least for some time. Putting together these various results we have that $p_{\lambda_i}^{(-1)}\Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1}) \leq p_{\lambda_i}^{(-1)}\Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*) \leq c(\frac{t_6-1}{M_e})^2\frac{\tau^{(t)}}{\mathcal{S}_\Delta}$. Hence we have $\mathrm{E}[\delta_{\lambda_i}\delta_{\gamma_j}] \leq \frac{c}{4}(\frac{t_6-1}{M_e})^2\frac{\tau^{(t)}}{\mathcal{S}_\Delta}$ and $\mathrm{Cov}[\delta_{\lambda_i}, \delta_{\gamma_j}] \leq \frac{c}{4}(\frac{t_6-1}{M_e})^2\frac{\tau^{(t)}}{\mathcal{S}_\Delta} - \frac{1}{4}$.

— In all the remaining cases, then the random variables δ_{λ_i} and δ_{γ_j} corresponds to FourEst not observing λ and γ using the shared triangle T_1 for both of them. Let T^* denote the triangle sub-structure used by TS4C$_1$ to count λ with respect to δ_{λ_i}. Let us consider the event $E^* =$"$\{e_1^*, e_2^*, e_3^*\} \cap T_1 \cap E^{(t_{1,2,4}-1)} \in \mathcal{S}_e^{t_{1,2,4}}$, $T_1 \in \mathcal{S}_\Delta^{(t_6)}$ unless one of its edges is the last edge of T^* observed on the stream, and $\{e_1^*, e_2^*, e_3^*\} \cap \{e_3, e_5\} \cap E^{(t_6-1)} \in S_e^{(t_6)}$ if $e^* \in \{e_3, e_5\}$", where $E^{(t)}$ denotes the set of edges observed up until time t included. Clearly $\Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1}) \leq \Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*)$. Note that in this case $|\{e_1^*, e_2^*, e_3^*\} \cap T_1| + |\{e_1^*, e_2^*, e_3^*\} \cap \{e_3, e_5\}|$. By analyzing $\Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*)$ in this case using similar steps as the ones described for the other sub-cases, we have $p_{\lambda_i}^{(-1)}\Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1}) \leq p_{\lambda_i}^{(-1)}\Pr(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*) \leq c\left(\frac{t_6-1}{M_e}\right)^3$. Hence we have $\mathrm{E}[\delta_{\lambda_i}\delta_{\gamma_j}] \leq \frac{c}{4}(\frac{t_6-1}{M_e})^3$ and $\mathrm{Cov}[\delta_{\lambda_i}, \delta_{\gamma_j}] \leq \frac{c}{4}(\frac{t_6-1}{M_e})^3 - \frac{1}{4}$.

We can thus bound the contribution to the third component of Equation (15) given by the pairs of random variables corresponding to 4-cliques that share three edges as:

$$2b^{(t)}\left(c\left(\frac{t-1}{M_e}\right)^2\left(\frac{1}{4}\frac{\tau^{(t)}}{M_\Delta} + \frac{3}{4}\frac{t-1}{M_e}\right) - 1\right), \tag{18}$$

where $b^{(t)}$ denotes the number of unordered pairs of 4-cliques which share three edges in $G^{(t)}$.

The Theorem follows by combining Equations (17) and (18) in Equation (15). □

C PROOFS OF TECHNICAL RESULTS REGARDING FOUREST

In this section, we present proofs for algorithm FOUREST discussed in Section 5.

PROOF OF THEOREM 5.2. Recall that when a new edge e_t is observed on the stream FOUREST updates the estimator $\varkappa^{(t)}$ *before* deciding whether the new edge is inserted in $\mathcal{S}$. For $t \leq M+1$, the entire graph $G^{(t)} \setminus \{e_t\}$ is maintained in $\mathcal{S}$, thus whenever an edge e_t is inserted at time $t \leq M+1$, FOUREST observes *all* the triangles which include e_t in $G^{(t)}$ with probability 1 thus increasing $\varkappa$ by one. By a simple inductive analysis we can therefore conclude that for $t \leq M+1$ we have $\varkappa^{(t)} = |C_4^{(t)}|$.

Assume now $t > M+1$ and assume that $|C_4^{(t)}| > 0$, otherwise, the algorithm deterministically returns 0 as an estimation, and the thesis follows. Recall that every time FOUREST observes a 4-clique on the stream a time t it computes the probability $p = \prod_{i=0}^{4} \frac{M-i}{t-i-1}$ of observing it and it correspondingly increases the running estimator by p^{-1}. From Lemma 2.1, the probability p computed by FOUREST does indeed correspond to the correct probability of observing a 4-clique at time t. For any 4-clique $\lambda \in C_4^{(t)}$ which is observed by TS4C$_1$ with probability p_λ, consider a random variable X_λ which takes value p^{-1} iff λ is actually observed by FOUREST (i.e., with probability p_λ) or zero otherwise. We thus have $\mathbb{E}[X_\lambda] = p_\lambda^{-1}\Pr(X_\lambda = p_\lambda^{-1}) = p_\lambda^{-1} p_\lambda = 1$.

We therefore can express the running estimator $\varkappa^{(t)}$ as: $\varkappa^{(t)} = \sum_{\lambda \in C_4^{(t)}} X_\lambda$.

From linearity of expectation, we thus have:

$$\mathbb{E}\left[\varkappa^{(t)}\right] = \sum_{\lambda \in C_4^{(t)}} \mathbb{E}[X_\lambda] = \sum_{\lambda \in C_4^{(t)}} p_\lambda^{-1} p_\lambda = |C_4^{(t)}|.$$

□

PROOF OF THEOREM 5.3. Assume $|C_4^{(t)}| > 0$ and $t > M+1$, otherwise (from Theorem 5.2) TS4C$_1$ estimation is deterministically correct and has variance 0 and the thesis holds. For each $\lambda \in C_4^{(t)}$ let $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, without loss of generality let us assume the edges are disposed as in Figure 1. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. Let us consider the random variable δ_λ (which takes value p_λ^{-1} if the 4-clique λ is observed by FOUREST, or zero otherwise. From Lemma 5.1, we have:

$$p_\lambda = \Pr\left(\delta_\lambda = p_\lambda^{-1}\right) = \prod_{i=0}^{4} \frac{M-i}{t-i-1},$$

and thus:

$$\mathrm{Var}\left[\delta_\lambda\right] = p_\lambda^{-1} - 1.$$

We can express the estimator $\varkappa^{(t)}$ as $\varkappa^{(t)} = \sum_{\lambda \in C_4^{(t)}}$. We therefore have:

$$\mathrm{Var}\left[\varkappa^{(t)}\right] = \mathrm{Var}\left[\sum_{\lambda \in C_4^{(t)}} \delta_\lambda\right] = \sum_{\lambda \in C_4^{(t)}} \sum_{\gamma \in C_4^{(t)}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] = \sum_{\lambda \in C_4^{(t)}} \mathrm{Var}\left[\delta_\lambda\right] + \sum_{\substack{\lambda, \gamma \in C_4^{(t)} \\ \lambda \neq \gamma}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right]$$

$$\leq |C_4^{(t)}| \left(\prod_{i=0}^{4} \frac{t-1-i}{M-i} - 1 \right) + \sum_{\substack{\lambda, \gamma \in C_4^{(t)} \\ \lambda \neq \gamma}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right]. \tag{19}$$

From Lemma A.1, we have that two distinct cliques λ and γ can share one, three, or no edges. In analyzing the summation of covariance terms appearing in the right-hand-side of Equation (19), we shall therefore consider separately the pairs that share, respectively, one, three, or no edges.

—λ and γ do not share any edge:

$$\begin{aligned}\mathrm{E}\left[\delta_\lambda \delta_\gamma\right] &= p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} \wedge \delta_\gamma = p_\gamma^{-1}\right) \\ &= p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \Pr\left(\delta_\lambda = p_\lambda^{-1}\right).\end{aligned}$$

The term $\Pr(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1})$ denotes the probability of FourEst observing λ conditioned of the fact that γ was observed. Note that as λ and γ do not share any edge, no edge of γ will be used by FourEst to detect λ. Rather, if any edge of γ is included in $\mathcal{S}$, this lowers the probability of FourEst detecting λ as some of space available in $\mathcal{S}$ may be occupied by edges of γ. Therefore we have $\Pr(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}) \leq \Pr(\delta_\lambda = p_\lambda^{-1})$ and thus:

$$\begin{aligned}\mathrm{E}\left[\delta_\lambda \delta_\gamma\right] &= p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \Pr\left(\delta_\gamma = p_\gamma^{-1}\right) \\ &\leq p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1}\right) \Pr\left(\delta_\gamma = p_\gamma^{-1}\right) \\ &\leq p_\lambda^{-1} p_\gamma^{-1} p_\lambda p_\gamma \\ &\leq 1.\end{aligned}$$

As $\mathrm{Cov}[\delta_\lambda, \delta_\gamma] = \mathrm{E}[\delta_\lambda \delta_\gamma] - 1$, we therefore have $\mathrm{Cov}[\delta_\lambda, \delta_\gamma] \leq 0$. Hence we can conclude that the contribution of the covariances of the pairs of random variables corresponding to 4-cliques that do not share any edge to the summation in Equation (19) is less or equal to zero.

—λ and γ share exactly one edge $e^* = \lambda \cap \gamma$ (as shown in Figure 9). Let us consider the event $E^* =$"$e^* \in \mathcal{S}_{t_6}$ unless e^* is observed at t_6". Clearly $\Pr(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}) \leq \Pr(\delta_\lambda = p_\lambda^{-1} | E*)$. Recall from Lemma 5.1 that $\Pr(\delta_\lambda | E^*) = \Pr(\{e_1, \ldots, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*)$. We can distinguish two cases: (a) $e^* = e_6$: in this case we have $\Pr(\delta_\lambda | E^*) = \Pr(\{e_1, \ldots, e_5\} \subseteq \mathcal{S}_e^{(t_6)}) = p_\lambda$; (b) $e^* \neq e_6$: in this case we have $\Pr(\delta_\lambda | E^*) = \Pr(\{e_1, \ldots, e_5\} \setminus \{e^*\} \subseteq \mathcal{S}^{(t_6)} | e^* \in \mathcal{S}_e^{(t_6)}) = \prod_{i=0}^{3} \frac{M-1-i}{t_6-2-i}$. We can therefore conclude:

$$\begin{aligned}\mathrm{E}\left[\delta_\lambda \delta_\gamma\right] &= p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \Pr\left(\delta_\gamma = p_\gamma^{-1}\right) \\ &\leq p_\lambda^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \\ &\leq \frac{t_6 - 1}{M}.\end{aligned}$$

We can thus bound the contribution to covariance summation in Equation (19) given by the pairs of random variables corresponding to 4-cliques that share one edge as:

$$a^{(t)} \left(\frac{t-1}{M} - 1 \right), \tag{20}$$

where $a^{(t)}$ denotes the number of unordered pairs of 4-cliques which share one edge in $G^{(t)}$.

— λ and γ share three edges $\{e_1^*, e_2^*, e_3^*\}$ which form a triangle sub-structure for both λ and γ. Let us refer to Figure 10. Let us consider the event $E^* =$"$\{e_1^*, e_2^*, e_3^*\} \cap E^{(t_6-1)} \subseteq \mathcal{S}^{(t_6)}$. Clearly $\Pr(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}) \leq \Pr(\delta_\lambda = p_\lambda^{-1} | E*)$. Recall that $\Pr(\delta_\lambda | E^*) = \Pr(\{e_1, \ldots, e_5\} \subseteq \mathcal{S}^{(t_6)} | E^*)$. We can distinguish two cases:
(a) $e_6 \cap \{e_1^*, e_2^*, e_3^*\} \neq \emptyset$: in this case we have $|\{e_1, \ldots, e_5\} \setminus (\{e_1^*, e_2^*, e_3^*\} \setminus \{e_6\})| = 3$ hence $\Pr(\delta_\lambda | E^*) = \Pr(\{e_1, \ldots, e_5\} \setminus (\{e_1^*, e_2^*, e_3^*\} \setminus \{e_6\}) | (\{e_1^*, e_2^*, e_3^*\} \setminus \{e_6\}) \subseteq \mathcal{S}^{(t_6)}) = \prod_{i=0}^{2} \frac{M-2-i}{t_6-3-i}$;
(b) $e_6 \cap \{e_1^*, e_2^*, e_3^*\} = \emptyset$: in this case we have $|\{e_1, \ldots, e_5\} \setminus \left(\{e_1^*, e_2^*, e_3^*\} \setminus \{e_6\}\right)| = 2$ hence $\Pr(\delta_\lambda | E^*) = \Pr\left(\{e_1, \ldots, e_5\} \setminus \{e_1^*, e_2^*, e_3^*\} | \{e_1^*, e_2^*, e_3^*\}) \subseteq \mathcal{S}^{(t_6)}\right) = \prod_{i=0}^{1} \frac{M-3-i}{t_6-4-i}$;. For the pairs of 4-cliques which share three edge we therefore have:

$$\begin{aligned} \mathrm{E}\left[\delta_\lambda \delta_\gamma\right] &= p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \Pr\left(\delta_\gamma = p_\gamma^{-1}\right) \\ &\leq p_\lambda^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \\ &\leq \prod_{i=0}^{2} \frac{t-1-i}{M-i}. \end{aligned}$$

We can thus bound the contribution to covariance summation in Equation (19) given by the pairs of random variables corresponding to 4-cliques that share one edge as:

$$b^{(t)} \left(\prod_{i=0}^{2} \frac{t-1-i}{M-i} - 1 \right) = b^{(t)} c' \left(\frac{t}{M}\right)^5 \tag{21}$$

where $b^{(t)}$ denotes the number of unordered pairs of 4-cliques which share three edges in $G^{(t)}$ and $c' = (M-1)(M-2)/M^2$.

The bound on the variance follows form the previous considerations and by combining by combining Equations (20) and (21) in Equation (19). Finally, the concentration bound is obtained by applying Chebyshev's inequality [26, Thm. 3.6]. The proof follows a reasoning analogous to that in the proof of Theorem 4.5. □

D FIVEEST: 5-CLIQUE COUNTING USING AN EDGE-ONLY SAMPLE

Lemma D.1. *Let $\lambda \in C_5^{(t)}$ with $\lambda = \{e_1, \ldots, e_{10}\}$. Assume, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_{10} > \max\{t_i, 1 \leq i \leq 9\}$. λ is observed by FiveEst at time t_{10} with probability:*

$$p_\lambda = \begin{cases} 0 & if |M| < 9, \\ 1 & if\ t_{10} \leq M+1, \\ \prod_{i=0}^{9} \frac{M-i}{t-i-1} & if\ t_{10} > M+1. \end{cases} \tag{22}$$

Theorem D.2. *Let $\varkappa^{(t)}$ the estimated number of 5-cliques in $G^{(t)}$ computed by FiveEst using memory of size $M > 9$. $\varkappa^{(t)} = |C_5^{(t)}|$ if $t \leq M+1$ and $\mathbb{E}[\varkappa^{(t)}] = |C_5^{(t)}|$ if $t > M+1$.*

The proof for Lemma D.1 (resp., Theorem D.2), closely follows the steps of the proof of Lemma 5.1 (resp., Theorem 5.2).

ALGORITHM 6: FiveEst - Single Reservoir Sampling for 5-cliques counting

Input: Edge stream Σ, integer $M \geq 6$
Output: Estimation of the number of 5-cliques $\varkappa$
$\mathcal{S}_e \leftarrow \emptyset$, $t \leftarrow 0$, $\varkappa \leftarrow 0$
for each element (u, v) from Σ **do** ▷ Process each edge (u, v) coming from the stream in discretized timesteps
  $t \leftarrow t + 1$
  Update5Cliques(u, v) ▷ Update the 5-clique estimator by considering the new 5-cliques closed by edge (u, v)
  SampleEdge$((u, v), t)$ ▷ Update the edges sample with the edge (u, v) according to RS scheme

function Update5Cliques(u, v)
  $\mathcal{N}^{\mathcal{S}}_{u,v} \leftarrow \mathcal{N}^{\mathcal{S}}_{u} \cap \mathcal{N}^{\mathcal{S}}_{v}$
  for each element (x, w, z) from $\mathcal{N}^{\mathcal{S}}_{u,v} \times \mathcal{N}^{\mathcal{S}}_{u,v} \times \mathcal{N}^{\mathcal{S}}_{u,v}$ **do**
    if $\{(x, w), (x, z), (w, z)\} \subseteq \mathcal{S}_e$ **then**
      if $t \leq M + 1$ **then**
        $p \leftarrow 1$
      else
        $p \leftarrow \prod_{i=0}^{9} \frac{M-i}{t-i-1}$
      $\varkappa \leftarrow \varkappa + p^{-1}$

REFERENCES

[1] Nesreen K. Ahmed, Nick Duffield, Jennifer Neville, and Ramana Kompella. 2014. Graph sample and hold: A framework for big-graph analytics. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1446–1455.

[2] Nesreen K. Ahmed, Nick Duffield, Theodore Willke, and Ryan A. Rossi. 2017. On sampling from massive graph streams. *Proceedings of the VLDB Endowment* 10, 11 (2017).

[3] Nesreen K. Ahmed, Nick Duffield, and Liangzhen Xia. 2018. Sampling for approximate bipartite network projection. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3286–3292.

[4] Nesreen K. Ahmed, Jennifer Neville, Ryan A. Rossi, and Nick Duffield. 2015. Efficient graphlet counting for large networks. In *Proceedings of the 2015 IEEE International Conference on Data Mining*. IEEE, 1–10.

[5] Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics* 74, 1 (2002), 47.

[6] Luca Becchetti, Paolo Boldi, Carlos Castillo, and Aristides Gionis. 2010. Efficient algorithms for large-scale local triangle counting. *ACM Transactions on Knowledge Discovery from Data* 4, 3 (2010), 1–28.

[7] Jonathan W. Berry, Bruce Hendrickson, Randall A. LaViolette, and Cynthia A. Phillips. 2011. Tolerating the community detection resolution limit with edge weighting. *Physical Review E* 83, 5 (2011), 056119.

[8] Paolo Boldi, Marco Rosa, Massimo Santini, and Sebastiano Vigna. 2011. Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks. In *Proceedings of the 20th International Conference on World Wide Web*. 587–596.

[9] Ilaria Bordino, Debora Donato, Aristides Gionis, and Stefano Leonardi. 2008. Mining large networks with subgraph counting. In *Proceedings of the 2008 8th IEEE International Conference on Data Mining*. IEEE, 737–742.

[10] Marco Bressan, Flavio Chierichetti, Ravi Kumar, Stefano Leucci, and Alessandro Panconesi. 2018. Motif counting beyond five nodes. *ACM Transactions on Knowledge Discovery from Data* 12, 4 (2018), 1–25.

[11] Marco Bressan, Stefano Leucci, and Alessandro Panconesi. 2019. Motivo: Fast motif counting via succinct color coding and adaptive sampling. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1651–1663.

[12] Lorenzo De Stefani, Erisa Terolli, and Eli Upfal. 2017. Tiered sampling: An efficient method for approximate counting sparse motifs in massive graph streams. In *Proceedings of the 2017 IEEE International Conference on Big Data (Big Data '17)*. IEEE, 776–786.

[13] Jean-Pierre Eckmann and Elisha Moses. 2002. Curvature of co-links uncovers hidden thematic layers in the world wide web. *Proceedings of the National Academy of Sciences* 99, 9 (2002), 5825–5829.
[14] Dhivya Eswaran, Christos Faloutsos, Sudipto Guha, and Nina Mishra. 2018. Spotlight: Detecting anomalies in streaming graphs. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1378–1386.
[15] Irene Finocchi, Marco Finocchi, and Emanuele G. Fusco. 2015. Clique counting in MapReduce: Algorithms and experiments. *Journal of Experimental Algorithmics* 20 (2015), 1–20.
[16] Rob J. Hyndman and Anne B. Koehler. 2006. Another look at measures of forecast accuracy. *International Journal of Forecasting* 22, 4 (2006), 679–688.
[17] Shweta Jain and C. Seshadhri. 2017. A fast and provable method for estimating clique counts using Turán's theorem. In *Proceedings of the 26th International Conference on World Wide Web*. 441–449.
[18] Madhav Jha, C. Seshadhri, and Ali Pinar. 2015. Path sampling: A fast and provable method for estimating 4-vertex subgraph counts. In *Proceedings of the 24th International Conference on World Wide Web*. 495–505.
[19] Madhav Jha, C. Seshadhri, and Ali Pinar. 2015. A space-efficient streaming algorithm for estimating transitivity and triangle counts using the birthday paradox. *ACM Transactions on Knowledge Discovery from Data* 9, 3 (2015), 1–21.
[20] Konstantin Kutzkov and Rasmus Pagh. 2014. Triangle counting in dynamic graph streams. In *Proceedings of the Scandinavian Workshop on Algorithm Theory*. Springer, 306–318.
[21] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology* 58, 7 (2007), 1019–1031.
[22] Yongsub Lim and U. Kang. 2015. Mascot: Memory-efficient and accurate sampling for counting local triangles in graph streams. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 685–694.
[23] Paul Liu, Austin R. Benson, and Moses Charikar. 2019. Sampling methods for counting temporal motifs. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 294–302.
[24] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. 2002. Network motifs: Simple building blocks of complex networks. *Science* 298, 5594 (2002), 824–827.
[25] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*. 29–42.
[26] Michael Mitzenmacher and Eli Upfal. 2017. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press.
[27] Rasmus Pagh and Charalampos E. Tsourakakis. 2012. Colorful triangle counting and a MapReduce implementation. *Information Processing Letters* 112, 7 (2012), 277–281.
[28] Kirill Paramonov, Dmitry Shemetov, and James Sharpnack. 2019. Estimating graphlet statistics via lifting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 587–595.
[29] Ha-Myung Park and Chin-Wan Chung. 2013. An efficient MapReduce algorithm for counting triangles in a very large graph. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. 539–548.
[30] A. Pavan, Kanat Tangwongsan, Srikanta Tirthapura, and Kun-Lung Wu. 2013. Counting and Sampling Triangles from a Graph Stream. In *Proceedings of the International Conference on Very Large Data Bases*. 1870–1881.
[31] Mahmudur Rahman, Mansurul Alam Bhuiyan, and Mohammad Al Hasan. 2014. Graft: An efficient graphlet counting method for large graph analysis. *IEEE Transactions on Knowledge and Data Engineering* 26, 10 (2014), 2466–2478.
[32] Seyed-Vahid Sanei-Mehri, Ahmet Erdem Sariyuce, and Srikanta Tirthapura. 2018. Butterfly counting in bipartite networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2150–2159.
[33] Lorenzo De Stefani, Alessandro Epasto, Matteo Riondato, and Eli Upfal. 2017. Triest: Counting local and global triangles in fully dynamic streams with fixed memory size. *ACM Transactions on Knowledge Discovery from Data* 11, 4 (2017), 1–50.
[34] Jeffrey S. Vitter. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software* 11, 1 (1985), 37–57.

Received March 2019; revised March 2020; accepted December 2020