Online Sampling of Temporal Networks

NESREEN K. AHMED, Intel Labs, USA
NICK DUFFIELD, Texas A&M University, USA
RYAN A. ROSSI, Adobe Research, USA

Temporal networks representing a stream of timestamped edges are seemingly ubiquitous in the real-world. However, the massive size and continuous nature of these networks make them fundamentally challenging to analyze and leverage for descriptive and predictive modeling tasks. In this work, we propose a general framework for temporal network sampling with unbiased estimation. We develop online, single-pass sampling algorithms and unbiased estimators for temporal network sampling. The proposed algorithms enable fast, accurate, and memory-efficient statistical estimation of temporal network patterns and properties. In addition, we propose a temporally decaying sampling algorithm with unbiased estimators for studying networks that evolve in continuous time, where the strength of links is a function of time, and the motif patterns are temporally-weighted. In contrast to the prior notion of a $\triangle t$-temporal motif, the proposed formulation and algorithms for counting temporally weighted motifs are useful for forecasting tasks in networks such as predicting future links, or a future time-series variable of nodes and links. Finally, extensive experiments on a variety of temporal networks from different domains demonstrate the effectiveness of the proposed algorithms. A detailed ablation study is provided to understand the impact of the various components of the proposed framework.

CCS Concepts: • **Theory of computation → Dynamic graph algorithms**; **Streaming, sublinear and near linear time algorithms**; **Sketching and sampling**.

Additional Key Words and Phrases: graph streams, dynamic networks, graph summarization, priority sampling, order sampling, reservoir sampling, statistical estimation, unbiased estimation, subgraph counting, triangle counting, higher-order networks, continuous-time dynamic networks

ACM Reference Format:
Nesreen K. Ahmed, Nick Duffield, and Ryan A. Rossi. 2018. Online Sampling of Temporal Networks. *ACM Trans. Knowl. Discov. Data.* 37, 4, Article 1 (August 2018), 26 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Networks provide a natural framework to model and analyze complex systems of interacting entities in various domains (*e.g.*, social, neural, communication, and technological domains) [67, 68]. Most complex networked systems of scientific interest are continuously evolving in time, while entities interact continuously, and different entities may enter or exit the system at different times. The accurate modeling and analysis of these complex systems largely depend on the network representation [41]. Therefore, it is crucial to incorporate both the heterogeneous *structural* and *temporal* information into network representations [69, 75, 81]. By incorporating the temporal

Authors' addresses: Nesreen K. Ahmed, nesreen.k.ahmed@intel.com, Intel Labs, Santa Clara, California, USA, 95054; Nick Duffield, duffieldng@tamu.edu, Texas A&M University, College Station, Texas, USA; Ryan A. Rossi, rrossi@adobe.com, Adobe Research, San Jose, California, USA.

1556-4681/2018/8-ART1 $15.00
https://doi.org/10.1145/1122445.1122456

information alongside the structural information, we obtain time-varying networks, also called *temporal networks* [43].

In temporal networks, the nodes represent the entities in the system, and the links represent the interactions among these entities across time. Unlike static networks, nodes and links in temporal networks become active at certain times, leading to changes in the network structure over time [53]. Temporal networks have been recently used to model and analyze dynamic and streaming network data, *e.g.*, to analyze and model information propagation [31, 74], epidemics [72], infections [61], user influence [21, 39], among other applications [69, 81]. However, there are fundamental challenges to the analysis of temporal networks in real-world applications. One major challenge is the massive size and streaming characteristics of temporal network data that are generated by interconnected systems, since all interactions must be stored at any given time (*e.g.*, email communications) [11]. As a result, several algorithms that were studied and designed for static networks that can fit in memory are becoming computationally intensive [8], due to their struggle to deal with the size and streaming properties of temporal networks.

One common practice is to aggregate interactions in discrete time windows (time bins) (*e.g.*, aggregate all interactions that appear in 1-day or 1-month), these are often called static *graph snapshots* [83]. Given a graph snapshot, traditional techniques can be used to study and analyze the network (*e.g.*, community detection, model learning, node ranking). Unfortunately, there are multiple challenges with employing these static aggregations. First, the choice of the size and placement of these time windows may alter the properties of the network and/or introduce a bias in the description of network dynamics [19, 42, 86, 90]. For example, a small window size will likely miss important network sub-structures that span multiple windows (*e.g.*, multi-node interactions such as motifs) [69]. On the other hand, a large window size will likely lose the temporal patterns in the data [33]. Second, modeling and analyzing bursty network traffic will likely be impacted by the placement of time windows. Finally, it is costly to consistently and reliably maintain these static aggregates for real-time applications [8, 11]. For example, it is often difficult to consistently gather these snapshots of graphs in one place, at one time, in an appropriate format for analysis. Thus, aggregates of network interactions in discrete time bins may not be an appropriate representation of temporal networks that evolve on a *continuous-time* scale [2, 35], and can often lead to errors and bias the results [69, 83, 90, 94, 97, 98].

Statistical sampling is also common in studying networks, where the goal is to select a *representative* sample (*i.e.*, subnetwork) that serves as a proxy for the full network [58]. Sampling algorithms are fundamental in studying and understanding networks [11, 47, 67]. A sampled network is called *representative*, if the characteristics of interest in the full network can be accurately estimated from the sample. Statistical sampling can provide a versatile framework to model and analyze network data. For example, when handling big data that cannot fit in memory, collecting data using limited storage/power electronic devices (*e.g.*, mobile devices, RFID), or when the measurements required to observe the entire network are costly (*e.g.*, protein interaction networks [85]).

While many network sampling techniques are studied in the context of small static networks that can fit entirely in memory [47] (*e.g.*, uniform node sampling [85], random walk sampling[52]), recently there has been a growing interest in sampling techniques for streaming network data in which temporal networks evolve continuously in time [8–10, 24, 27, 44, 45, 55, 71, 73, 82, 84] (see [11, 62] for a survey). Most existing methods for sampling streaming network data have focused on the primary objective of selecting a sample to estimate static network properties (*e.g.*, point statistics such as global triangle count or clustering coefficient). This poses an interesting and important question of how representative these samples of the characteristics of temporal networks that evolve on a *continuous-time* scale [3], such as the link strength [93], link persistence [25],

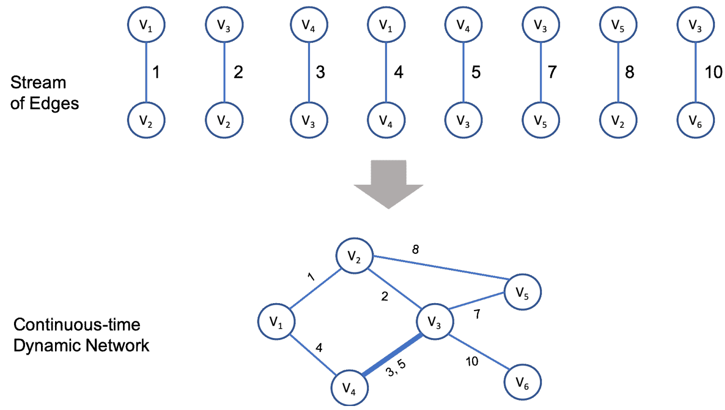

Fig. 1. An illustrative example of streaming temporal networks.

burstiness [16], temporal motifs [48], among others [4, 36, 43, 49, 50]. Although this question is important, it has thus far not been addressed in the context of streaming and online methods.

In this paper, we introduce an online importance sampling framework that extracts continuous-time dynamic network samples, in which the strength of a link (*i.e.*, edge between two nodes) can evolve continuously as a function of time. Our proposed framework samples interactions to include in the sample based on their importance weight relative to the variable of interest (*i.e.*, link strength), this enables sampling algorithms to adapt to the topological changes of temporal networks. Also, our proposed framework allows online and incremental updates, and can run efficiently in a single-pass over the data stream, where each interaction is observed and processed once upon arrival. We present an unbiased estimator of the link strength, and extend our formulation to unbiased estimators of general subgraphs in temporal networks. We also introduce the notion of *link-decay* network sampling, in which the strength of a sampled link is allowed to decay exponentially after the most recent update (*i.e.*, recent interaction). We show unbiased estimators of link strength and general subgraphs under the link-decay model.

Summary of Contributions: This work makes the following key contributions:

- We propose a general temporal network sampling framework for unbiased estimation of temporal network statistics. We develop online, single-pass, memory-efficient sampling algorithms and unbiased estimators.
- We propose a temporally decaying sampling algorithm with unbiased estimators for studying networks that evolve in continuous time, where the strength of links is a function of time, and the motif patterns and temporal statistics are temporally weighted accordingly. This temporal decay model is more useful for real-world applications such as prediction and forecasting in temporal networks.
- The proposed algorithms enable fast, accurate, and memory-efficient statistical estimation of temporal network patterns and statistics.
- Experiments on a wide variety of temporal networks demonstrate the effectiveness of the framework.

2 ONLINE SAMPLING FRAMEWORK

Here, we introduce our proposed online importance sampling framework that extracts continuous-time dynamic network samples from temporal networks. See Table 1 for a summary of notations.

Table 1. Summary of notation.

G	Temporal network
E	Set of interaction events
K	Set of Unique Edges (links)
E_t	Set of interactions $\{e_s : s \leq t\}$
K_t	Set of unique interactions in E_t arriving by time t
V_t	Set of vertices that appeared in K_t
$\widetilde{G}$	Graph induced by unique edges
N, M	number of nodes $N = \lvert V \rvert$ and edges $M = \lvert K \rvert$ in $\widetilde{G}$
$C_{e,t}$	Multiplicity (weight) of edge e at time t
$\widehat{C}_{e,t}$	Estimated multiplicity of edge e at time t
$\mathbf{C}_t$	time-dependent adjacency matrix of link strength at time t
$\widehat{K}$	Reservoir of sampled edges
m	Number of sampled edges (Sample Size), $m = \lvert \widehat{K} \rvert$
δ	Link decay rate
ϕ	Initial weight
$C_{\mathcal{M}}$	Weighted count of motif pattern $\mathcal{M}$
$\widehat{C}_{\mathcal{M}}$	Estimated weighted count of motif pattern $\mathcal{M}$
$V(e)$	Unbiased estimator of variance of edge e
$w(e)$	Sampling weight of edge e
$r(e)$	Rank of edge e in the sample

Overview. We propose an online sampling framework for temporal streaming networks which seeks to construct continuous-time, fixed-size, dynamic sampled network that can capture the evolution of the full network as it evolves in time as a stream of edges. Our framework assumes an input temporal network represented a stream of interactions links at certain times, and each interaction can be observed and/or processed only once. Sampling algorithms are allowed to store only m sampled edges, and can process the stream in a single-pass. If any two vertices interact at time $t = \tau$, their edge strength increases by 1. Figure 1 shows an illustrative example of how a continuous-time dynamic network can be formed from a stream of edges, where the edge strength is a function of the interactions among vertices over time.

2.1 Notation & Problem Definition

Edges, Interactions, and Streaming Temporal Networks. Our framework seeks to construct a continuous-time *sampled* network that can capture the characteristics and serve as a proxy of an input temporal network as it evolves continuously in time. In this paper, we draw an important distinction between *interactions* and *edges.* An interaction (contact) between two entities is an event that occurred at a certain point in time (*e.g.*, an email, text message, physical contact). On the other hand, an edge between two entities represents the link or the relationship between them, and the weight of this edge represents the strength of the relationship (*e.g.*, strength of friendship in social network [93]). We use G to denote an input temporal network, where a set of vertices V (*e.g.*, users or entities) are interacting at certain times. Let $(i, j, t) \in E$ denote the interaction event that takes place at time t, where $i, j \in V$, E is the set of interactions, E_t is the set of interactions up to time t, and K is the set of *unique* edges ($e = (i, j) \in K$) in the temporal network G. We assume these interactions are instantaneous (*i.e.*, the duration of the interaction is negligible), *e.g.*, email, tweet, text message, etc. Let C_e denote the multiplicity (weight) of an edge $e = (i, j)$, with $C_{e,t}$ being the multiplicity of the edge at time t, *i.e.*, the number of times the edge appears in interactions up to time t. Finally, we define a streaming temporal network G as a *stream of interactions* $e_1, \dots, e_t, \dots, e_T$, with $e_t = (i, j, t)$ is the interaction between $i, j \in V$ at time t.

We note that the term unique edges refers to the set of relationships that exist among the vertices. On the other hand, the term interaction refers to an event that occurred at some point in time between two vertices. As such, interactions can happen more than once between two vertices, while unique edges represent the existing relationship between two vertices. For example, in Figure 1, the edge $e = (v_3, v_4)$ has two interactions that happen at times $t = 3, 5$ respectively. Hence, the strength of $e = (v_3, v_4)$ is higher compared to other edges.

Continuous-time Dynamic Network Samples. Consider a set of $N = |V|$ interacting vertices, with their interactions represented as a streaming temporal network G, *i.e.*, $e_1, \dots, e_t, \dots, e_T$. Let $\mathbf{C}_t$ be the time-dependent adjacency matrix, whose entries $C_{ij,t} \geq 0$ represent the relationship strength between vertices $i, j \in V$ at time t. The relationship strength is a function of the edge multiplicity and time. Our framework seeks to construct, maintain, and adapt a continuous-time dynamic sampled network, represented by the matrix $\widehat{\mathbf{C}}_t$ that serves as unbiased estimator of $\mathbf{C}_t$ at any time point t, where the expected number of non-zero entries in $\widehat{\mathbf{C}}_t$ is at most m, and m is the sample size (*i.e.*, maximum number of sampled edges). Our framework makes the following assumptions:

- We assume an input temporal network represented a stream of interactions at certain times, and each interaction can be processed and observed only once.
- Any algorithm can only store m sampled edges, and is allowed a single-pass over the stream.
- If two vertices interact at time $t = \tau$, their edge strength increases by 1.

2.2 Link-Decay Network Sampling

Here, we introduce a novel online sampling framework that seeks to construct and maintain a *sampled* temporal network in which the strength of a link (*i.e.*, relationship between two friends) can evolve continuously in time. Since the sampled network serves as a proxy of the full temporal network, the sampled network is expected to capture both the structural and temporal characteristics of the full temporal network.

Temporal Link-Decay. Assume an input stream of interactions, where interactions are instantaneous (*e.g.*, email, text message, and so on). For any pair of vertices $i, j \in V$, with a set of interaction times $\tau^{(1)}, \tau^{(2)}, \dots, \tau^{(T)}$, where $0 < \tau^{(1)} < \cdots < \tau^{(T)}$, and their first interaction time is $\tau^{(1)} > 0$. Our goal is to estimate the strength of the link $e = (i, j)$ as a function of time, in which the link strength may increase or decrease based on the frequency and timings of the interactions. Consider two models of constructing an adaptive sampled network represented as a time-dependent adjacency matrix $\mathbf{C}_t$, whose entries represent the link strength $C_{ij,t}$.

The first model is the *no-decay* model, in which the link strength does not decrease over time, *i.e.*, $C_{ij,\tau^{(2)}} = C_{ij,\tau^{(1)}} + 1$. Thus, $C_{ij,t}$ is the multiplicity or a function of the frequency of an edge, and we provide an unbiased estimator for this in Theorem 1. However, the no decay model assumes the interactions are fixed once happened, taking only the frequency of interactions as the primary factor in modeling link strength, which could be particularly useful for certain applications, such as proximity interactions (*e.g.*, link strength for people attending a conference).

The second model is the link-decay model, in which the strength of the link decays exponentially after the most recent interaction, to capture the temporal evolution of the relationship between i and j at any time t. Let the initial condition of the strength of link (i, j) be $C_{ij,t_0} = 0$. Then,

$C_{ij,t} = \sum_{s=0}^{T} \theta(t - \tau^{(s)})\ \mathrm{e}^{-(t-\tau^{(s)})/\delta}$, where $\theta(t)$ is the unit step function, and the decay factor $\delta > 0$. We formulate the link strength as a stream of events (*e.g.*, signals or pulses), that can be adapted incrementally in an online fashion, so the strength of link $e = (i, j)$ at time t follows the equation,

$$C_{e,t} = C_{e,t-1} * \mathrm{e}^{-1/\delta} \tag{1}$$

And if a new interaction occurred at time t, the link strength follows,

$$C_{e,t} = C_{e,t-1} * \mathrm{e}^{-1/\delta} + 1 \tag{2}$$

Our approach discounts the contributions of interactions to the time-dependent link strength as a function of the interaction age, while adapting the sampling weight of the link to its non-discounted multiplicity. This allows us to preferentially retain the relatively small proportion of highly active links, while the capability to temporally weight motif and subgraphs resides in the estimator. This is distinct from previous approaches for temporal sampling in which the retention sampling probability for single items were, *e.g.*, exponentially discounted according to age, without regard to item frequencies as a criterion for retention; see [28].

We formulate an unbiased estimator for the link-decayed strength as a function of the link multiplicity in Section 4 (see Theorem 3). All the proposed estimators can be computed and updated efficiently in a single-pass streaming fashion using Algorithm 1. In addition to exponential decay, the unbiased estimators generalize and can be easily extended to other decay functions, such as polynomial decay. Note that in this paper, we use the term link-decay to refer to exponential link-decay.

Link decay has major advantages in network modeling that we discuss next. First, it allows us to utilize both the frequency and timings of interactions in modeling link strength. Second, it is more realistic, allowing us to avoid any potential bias that may result from partitioning interactions into time windows. Link decay is also flexible, by tuning the decay factor δ, we can determine the degree at which the strength of the link ages (*i.e.*, the half-life of a link $t_{1/2} = \delta \ln 2$). We also note that the link decay model and the unbiased estimator in Theorem 3 can generalize to allow more flexibility, by tuning the decay parameter on the network-level, the node-level, or the link-level, to allow different temporal scales at different levels of granularity.

Temporally Weighted Motifs. We showcase our formulation of estimated link strength by estimating the counts of motif frequencies in continuous time. We introduce the notion of *temporally weighted motifs* in Definition 1. Temporally weighted motifs are more meaningful and useful for practical applications especially related to prediction and forecasting where links and motifs that occur more recently as well as more frequently are more important than those occurring in the distant past.

Definition 1. (Temporally Weighted Motif) *A temporally weighted network motif $\mathcal{M}$ is a small induced subgraph pattern with n vertices, and m edges, such that $C_{\mathcal{M}}$ is the time-dependent frequency of $\mathcal{M}$ and is subject to temporal decay, and $C_{\mathcal{M},t} = \sum_{h \in H_t} \prod_{e \in h} C_{e,t}$, where H_t is the set of observed subgraphs isomorphic to $\mathcal{M}$ at time t, and $C_{e,t}$ is the link strength.*

In general, motifs represent small subgraph patterns and the motif counts were shown to reveal fundamental characteristics and design principles of complex networked systems [12, 13, 17, 63], as well as improve the accuracy of machine learning models [14, 78]. While prior work focused on aggregating interactions in time windows and analyze the aggregated graph snapshots [75, 81, 88], others have focused on aggregating motifs in $\triangle t$ time bins, and defined motif duration [56]. These approaches rely on judicious partitioning of interactions in time bins, and would certainly suffer from the limitations discussed earlier in Section 1. Time partitioning may obfuscate or dilute temporal and structural information, leading to biased results. Here, we define instead a temporal weight or strength for any observed motif, which is a function of the strength of the links participating in the motif itself. Similar to the link strength, the motif weight is subject to time decay. This formulation can also generalize to models of higher-order link decay (*i.e.*, decaying hyperedges in hypergraphs), we defer this to future work. The definition in 1 can be computed incrementally in an online fashion, and subject to approximation via sampling and unbiased estimators. We establish our sampling methodology in Algorithm 1 (see line 39) and unbiased estimators of subgraphs in Section 4 (see Theorem 1).

3 PROPOSED ALGORITHM

In this paper, we propose an online sampling framework for temporal streaming networks which seeks to construct *continuous-time*, fixed-size, dynamic sampled network that can capture the evolution of the full network as it evolves in time. Our proposed framework establishes a number of properties that we discuss next. We formally state our algorithm, called Online-TNS, in Algorithm 1.

Setup and Key Intuition. The general intuition of the proposed algorithm in Algorithm 1, is to maintain a dynamic rank-based reservoir sample $\widehat{K}$ of a fixed-size m [9, 29, 91], from a temporal network represented as stream of interactions, where edges can appear repeatedly. And, $m = |\widehat{K}|$ is the maximum possible number of sampled edges. When a new interaction $e_t = (i, j, t)$ arrives (line 3), if the edge $e = (i, j)$ has been sampled before (*i.e.*, $e = (i, j) \in \widehat{K}$), then we only need to update the edge sampling parameters (in lines 8–11) and the edge strength (line 7). However, if the edge is new (*i.e.*, $e = (i, j) \notin \widehat{K}$), then the new edge is added provisionally to the sample (line 19), and one of the $m + 1$ edges in $\widehat{K}$ gets discarded (lines 21 and 23).

Importance sampling weights and rank variables. Algorithm 1 preferentially selects edges to include in the sample based on their importance weight relative to the variable of interest (*e.g.*, relationship strength, topological features), then adapts their weights to allow edges to gain importance during stream processing. To achieve this, each arriving edge e is assigned an initial weight $w(e)$ on arrival and an iid uniform $U(0, 1]$ random variable $u(e)$. Then, Algorithm 1 computes and continuously updates a rank variable for each sampled edge $r(e) = w(e)/u(e)$ (see line 17 and line 10). This rank variable quantifies the importance/priority of the edge to remain in the sample. To keep a fixed sample size, the $m + 1$ edge with minimum rank is always discarded (lines 21 and 23). The algorithm also maintains a sample threshold z^* which is the maximum discarded rank (line 22). Thus, the inclusion probability of an edge e in the sample is: $\mathbb{P}(e \in \widehat{K}) = \mathbb{P}(r(e) > z^*) = \mathbb{P}(u(e) < w(e)/z^*) = \min\{1, w(e)/z^*\}$. Our mathematical formulation in Section 4 allows the edge sampling weight to increase when more interactions are observed (line 9). Thus, edges can gain more importance or rank that reflects the relationship strength as it evolves continuously in time. This setup will support network models that focus on capturing the relationship strength in temporal networks [43, 64].

Unbiased estimation of link strength. We use a procedure called update-edge-strength (line 25 of Algorithm 1) to dynamically maintain an unbiased estimate (see Theorem 1) of the edge strength as it evolves continuously in time. The procedure in line 25 of Algorithm 1 also maintains an unbiased estimate of the variance of the edge strength following Theorem 2. Note the strength of an edge e is a function of the edge multiplicity C_e (the number of interactions e_t where $e_t = e$). If a link-decaying model is required, the procedure called update-edge-decay can be used instead of update-edge-strength to estimate the link-decayed strength (see line 32 of Algorithm 1). We prove that our estimated link-decaying weight is unbiased in Theorem 3.

Unbiased estimation of subgraph counts. Given a motif pattern $\mathcal{M}$ of interest (*e.g.*, triangles, or small cliques), the procedure called Subgraph-Estimation in line 39 of Algorithm 1 is used to update an unbiased estimate of the count of all occurrences of the motif $\mathcal{M}$ at any time t. Theorem 1

Algorithm 1 Online Temporal Network Sampling (Online-TNS)

INPUT: Sample size m, Motif pattern $\mathcal{M}$, initial weight ϕ
OUTPUT: Estimated network $\mathbf{C}_t$, Estimated motif count $\widehat{C}_{\mathcal{M}}$

```
procedure ONLINE-TNS(m)
    K̂ = ∅; z* = 0; Ĉ_M = 0                          ▷ Initialize edge sample & threshold
    while (new interaction e_t = (i, j, t)) do
        e = (i, j)
        SUBGRAPH-ESTIMATION(e)                          ▷ Update Estimated Motif
        if (e ∈ K̂) then                                 ▷ Edge exists in K̂
            UPDATE-EDGE-STRENGTH(e)                     ▷ Update Edge Strength
            Ĉ(e) = Ĉ(e) + 1                             ▷ Increment edge multiplicity
            w(e) = w(e) + 1                             ▷ Adapt importance weight
            r(e) = w(e)/u(e)                            ▷ Adapt edge rank
            τ(e) = t                                    ▷ Last Interaction Time
        else
            //Initialize parameters for new edge
            p(e) = 1; Ĉ(e) = 1; V(e) = 0;
            u(e) = Uniform (0, 1]                       ▷ Initialize Uniform r.v.
            w(e) = ϕ                                    ▷ Initialize edge weight
            r(e) = w(e)/u(e)                            ▷ Compute edge rank
            τ(e) = t
            K̂ = K̂ ∪ {e}                                 ▷ Provisionally include e in sample
            if (|K̂| > m) then
                e* = arg min_{e'∈K̂} r(e')               ▷ Find edge with min rank
                z* = max{z*, r(e*)}                     ▷ Update threshold
                remove e* from K̂
                delete {w(e*), u(e*), p(e*), Ĉ(e*), V(e*)}

procedure UPDATE-EDGE-STRENGTH(ẽ)
    // Function to estimate edge strength (No-decay)
    if (z* > 0) then
        q = min{1, w(ẽ)/(z* p(ẽ))}
        Ĉ(ẽ) = Ĉ(ẽ)/q                                   ▷ Estimate edge strength
        V(ẽ) = V(ẽ)/q + (1 − q) ∗ Ĉ(ẽ)^2
        p(ẽ) = p(ẽ) ∗ q

procedure UPDATE-EDGE-DECAY(ẽ)
    // Function to estimate edge strength (Link-decay)
    if (z* > 0) then
        q = min{1, w(ẽ)/(z* p(ẽ))}
        Ĉ(ẽ) = e^{−δ(t−τ(ẽ))} ∗ Ĉ(ẽ)/q                   ▷ Estimate link strength
        V(ẽ) = V(ẽ)/q + (1 − q) ∗ Ĉ(ẽ)^2
        p(ẽ) = p(ẽ) ∗ q

procedure SUBGRAPH-ESTIMATION(ẽ)
    //Set of Subgraphs isomorphic to M and completed by ẽ
    H = {h ⊂ K̂ ∪ {ẽ} : h ∋ ẽ, h ≅ M}
    for h ∈ H do
        for j ∈ h \ ẽ do
            UPDATE-EDGE-STRENGTH(j)                     ▷ Update other edges
        //Increment estimated count of motif M
        Ĉ_M = Ĉ_M + ∏_{j∈h\{ẽ}} Ĉ(j)
```

is used to establish the unbiased estimator of the count of general subgraphs. The unbiased estimator of subgraph counts also applies in the case of link decay, and gives rise to temporally decayed (weighted) motifs.

Computational Efficiency and complexity. All the algorithms and estimators can run in a *single-pass* on the stream of interactions, where each interaction can be observed and processed once (see Alg 1). The main reservoir sample is implemented as a heap data structure (min-heap) with a hash table to allow efficient updates. The estimator of edge strength can be updated in constant time $O(1)$. Also, retrieving the edge with minimum rank can be done in constant time $O(1)$. Any updates to the sampling weights and rank variables can be executed in a worst-case time of $O(\log(m))$ (*i.e.*, since it will trigger a bubble-up or bubble-down heap operations). For any incoming edge $e = (i, j)$, subgraph estimators can be efficiently computed if a hash table or bloom filter is used for storing and looping over the sampled neighborhood of the sampled vertex with minimum degree and querying the hash table of the other sampled vertex. For example, if we seek to estimate triangle counts, then line 41 in Algorithm 1 can be implemented in $O(min\{\deg(i), \deg(j)\})$.

4 ADAPTIVE UNBIASED ESTIMATION

In this section, we theoretically show and discuss our formulation of unbiased estimators for temporal networks, that we use in Algorithm 1.

Edge Multiplicities. We consider a temporal network $G = (V, E)$ comprising interactions E between vertex pairs of V. Each interaction can be viewed as a representative of an edge set K comprising the unique elements of E. We will write $\widetilde{G} = (V, K)$ as the graph induced by K. Thus the stream of interactions can also be regarded as a stream $\{e_t : t \in [|E|]\}$ of non-unique edges from K. Let K_t denote the unique edges in $\{e_s : s \leq t\}$ and have arrived by time t. Also, let $\widetilde{G}_t = (V_t, K_t)$ be the induced graph, where V_t is the subset of vertices that appeared in K_t. The multiplicity $C_{e,t}$ of an edge $e \in K_t$ is the number of times it occurs in $E_t = \{e_s : s \leq t\}$, i.e., $C_{e,t} = |\{s \leq t : e_s = e\}|$. The multiplicity $C_{J,t}$ of $J \subset K_t$ is the number of distinct ordered interaction subsets $\widetilde{J} = \{e_{i_1}, \ldots, e_{i_{|J|}}\}$ with $i_j \leq t$, such that $\widetilde{J}$ is a permutation of J. Hence $C_{J,t} = \prod_{e \in J} C_{e,t}$. Given a class $\mathcal{H}$ of subgraphs of $\widetilde{G}$, we wish to estimate for each t the total multiplicity $H_t = \sum_{J \in \mathcal{H}} C_{J,t}$ of subgraphs from $\mathcal{H}$ that are present in the first t arrivals.

Sampling Edges and Estimating Edge Multiplicities. We record edge arrivals by the indicators $c_{e,t} = 1$ if $e_t = e$ and zero otherwise, and hence $C_{e,t} = \sum_{t \geq 1} c_{e,t}$. $\widehat{K}_t$ will denote the sample set of unique edges after arrival t has been processed. We maintain an estimator $\widehat{C}_{e,t}$ of $C_{e,t}$ for each $e \in \widehat{K}_t$. Implicitly $\widehat{C}_{e,t} = 0$ if $e \notin \widehat{K}_t$.

The algorithm proceeds as follows. If the arriving edge $e_t \notin \widehat{K}_{t-1}$ then e_t is provisionally included in the sample, forming $\widehat{K}'_t = \widehat{K}_t \cup \{e_t\}$, and we set $\widehat{C}_{e_t,t} = c_{e_t,t} = 1$. The new edge is assigned a random variable u_{e_t} distributed IID in $(0, 1]$. A weight $w_{i,t}$ is specified for each edge $i \in \widehat{K}'_t$ as described below, from which the edge *time-dependent priority* at time t is $r_{i,t} = w_{i,t}/u_i$. If $|\widehat{K}'_t| > m$, the edge $d_t = \arg\min_{i \in \widehat{K}'_t} r_{i,t}$ of minimum priority is discarded, and the estimates $\widehat{C}_{i,t}$ of the surviving edges $i \in \widehat{K}_t = \widehat{K}'_t \setminus \{d_t\}$ undergo inverse probability normalization through division by the conditional probability $q_{i,t}$ of retention in $\widehat{K}_t$; see (Equation 6). If the arriving edge is already in the reservoir $e_t \in \widehat{K}_{t-1}$ then we increment its multiplicity $\widehat{C}_{e_t,t} = \widehat{C}_{e_t,t-1} + 1$ and no sampling is needed, i.e., $\widehat{K}_t = \widehat{K}_{t-1}$.

Unbiased Estimation of Edge Multiplicities. Let Ω denote the (random) set of times at which the sampling step takes place, i.e., such that the arriving edge e_t is not currently in the reservoir $e_t \notin \widehat{K}_{t-1}$ and $|\widehat{K}_{t-1}| = m$. For $t \in \Omega' = \Omega \setminus \{\min \Omega\}$, let $\omega(t) = \max\{[0, t) \cap \Omega\}$ denote the next most recent time at which the sampling step took place. For $t \in \Omega'$, the sample counts present in the reservoir accrue unit increments from arrivals $e_{\omega(t)+1}, \ldots, e_{t-1}$ until the sampling step takes place at time t. For $t \in \Omega$, an edge $i \in \widehat{K}_{\omega(t)}$ is selected into $\widehat{K}_t$ if and only if $r_{i,t}$ exceeds the smallest priority of all other elements of $\widehat{K}'_t$, i.e.,

$$r_{i,t} > z_{i,t} := \min_{j \in \widehat{K}'_t \setminus \{i\}} r_{j,t} \tag{3}$$

Hence by recurrence, $i \in \widehat{K}_t$ only if $u_i < \min_s \{w_{i,s}/z_{i,s}\}$ where s takes values over

$$\{\alpha_i(t), \ldots, \omega(\omega(t)), \omega(t), t\} \tag{4}$$

where $\alpha_i(t)$ is the most recent time at which edge i was sampled into the reservoir.

This motivates the definition below where $p_{e,t}$ is the edge selection probability conditional on the thresholds z_t, and $q_{e,t}$ is the conditional probability for sampling for each increment of time. Let t_e denote the time of first arrival of edge e. For $t \in \Omega$, define $p_{e,t}$ through the iteration

$$p_{e,t} = \begin{cases} \min\{1, w_{e,t}/z_t\} & \text{if } t = \min \Omega \\ \min\{p_{e,\omega(t)}, w_{e,t}/z_t\} & \text{otherwise} \end{cases} \tag{5}$$

where $z_t = \min_{e \in \widehat{K}'_t} r_{e,t}$ for $t \in \Omega_t$ in the unrestricted minimum priority over edges in $\widehat{K}'_t$. Note that $z_{i,t} = z_t$ if $i \in \widehat{K}_t$. Then $\widehat{C}_{e,t}$ is defined by the iteration $\widehat{C}_{e,t} = 0$ for $t < t_e$ and

$$\widehat{C}_{e,t} = \left(\widehat{C}_{e,t-1} + c_{e,t}\right) \frac{I(u_i < w_{i,t}/z_{i,t})}{q_{e,t}} \tag{6}$$

where

$$q_{e,t} = \begin{cases} 1 & \text{if } t \notin \Omega \\ p_{e,t} & \text{if } t \in \Omega \text{ and } e = e_t \\ p_{e,t}/p_{e,\omega(t)} & \text{otherwise} \end{cases} \tag{7}$$

For $J \subset V$, let $t_J = \min_{j \in J} t_j$, i.e., the earliest time at which any instance of an edge in J has arrived. Let $J_t = \{j \in J : t_j \leq t\}$, i.e., the edges in J whose first instance has arrived by t. Note in our model these are deterministic. The proof of the following Theorem and others in this paper are detailed in Section A.

Theorem 1 (**Unbiased Estimation**).

(i) $\mathbb{E}[\widehat{C}_{e,t}] = C_{e,t}$ for all $t \geq 0$.

(ii) For each $J \subset V$ and $t \geq t_J$ then $\prod_{e \in J_t} (\widehat{C}_{e,t} - C_{e,t}) : t \geq t_J\}$ has expectation 0.

(iii) $\mathbb{E}[\prod_{e \in J} \widehat{C}_{e,t}] = \prod_{e \in J} C_{e,t}$ for $t \geq \max_{j \in J} t_j$.

Estimating Subgraph Multiplicities. Theorem 1 tells us for a subgraph $J \subset K_t$, that $\prod_{e \in J} \widehat{C}_{e,t}$ is an unbiased estimator of the multiplicity $\prod_{e \in J} C_{e,t}$ of subgraphs formed by distinct set of interactions isomorphic to J. Now let $h \in H_t$, the set of subgraphs of $\widetilde{G}_t$ that are isomorphic to $\mathcal{M}$ at time t. We partition the set of interactions in E_t that represent h according to the time of last arrival. Thus it is evident that $C_{\mathcal{M},t} = \sum_{s \leq t} C^{(0)}_{\mathcal{M},s}$ where $C^{(0)}_{\mathcal{M},s} = \sum_{h \in H^{(0)}_s} C_{h \setminus \{e_s\},s}$ where $H^{(0)}_s = \{h \in K_s : h \ni e_s : h \cong M\}$, meaning, for each interaction e_s, we consider subgraphs h of K_s congruent to $\mathcal{M}$ and containing e_s, and compute the multiplicity of the h with e_s removed, i.e., not counting any isomorphic sets of interactions in which the $e = e_s$ arrived previously, thus avoiding over-counting. It follows by linearity that $\widehat{C}_{\mathcal{M},t} = \sum_{s \leq t} \widehat{C}^{(0)}_{\mathcal{M},s-1}$ is an unbiased estimator of $C_{\mathcal{M},t}$ where $\widehat{C}^{(0)}_{\mathcal{M},s-1} = \sum_{h \in H^{(0)}_s} \widehat{C}_{h \setminus \{e_s\},s-1}$. Thus for each arrival e_t we estimate $C_{J,t}$ just prior to sampling of e_t by $\widehat{C}_{J,t} = \prod_{j \in J \setminus \{e_t\}} \widehat{C}_{j,t-1}$. For each $J \subset H_t$ we increment a running total of $\widehat{M}_t$ by this amount; see line 46 in Algorithm 1.

Edge Multiplicity Variance Estimation. We now discuss the unbiased estimator of the variance $\mathrm{Var}(\widehat{C}_{e,t})$.

Theorem 2 (Unbiased Variance Estimator).
Suppose $\widehat{V}_{e,t-1}$ is an unbiased estimator of $\mathrm{Var}(\widehat{C}_{e,t-1})$ that can be computed from information on the first $t-1$ arrivals. Then

$$\widehat{V}_{e,t} = \widehat{C}^2_{e,t}(1 - q_{e,t}) + I(u_e < w_{e,t}/z_t)\widehat{V}_{e,t-1}/q_{e,t} \tag{8}$$

is a unbiased estimator of $\mathrm{Var}(\widehat{C}_{e,t})$ that can be computed from information on the first t arrivals.

The computational condition expresses the property that $\widehat{V}_{e,t}$ can be computed immediately when $e \in \widehat{K}_t$. The relation (Equation 8) defines an iteration for estimating the variance $\mathrm{Var}(\widehat{C}_t)$ for any t following a time $s \in \Omega$ at which edge e was sampled into $\widehat{K}_s$, such that e remained in the reservoir at least until t. The unbiased variance estimate $\widehat{V}_{e,s}$ takes the value $1/p_{e,s} - 1$ at time s of selection into the reservoir. In practice $\widehat{V}_{e,t}$ only needs to be updated at $t \in \Omega$, i.e., when some edge is sampled into the reservoir, since $q_{e,t} = 1$ when $t \notin \Omega$.

Estimation and Variance for Link-Decay Model.
The link-delay model adapts (Sec.2.2) through

$$\widehat{C}^{\delta}_{k,t} = \left(\widehat{C}^{\delta}_{k,t-1} e^{-1/\delta} + c_{k,t}\right) \frac{I(u_k < w_{k,t}/z_t)}{q_{k,t}} \tag{9}$$

which exponentially discounts the contribution from the previous time slot.

Theorem 3 (Unbiased Estimation with Link Decay).

(i) $\widehat{C}^{\delta}_{k,t}$ is an unbiased estimator of $C^{\delta}_{k,t}$

(ii) Replacing $\widehat{C}_{k,t}$ with $\widehat{C}^{\delta}_{k,t}$ in the iteration yields an unbiased estimator $V^{\delta}_{k,t}$ of $\mathrm{Var}(\widehat{C}^{\delta}_{k,t})$

5 EXPERIMENTS

We perform extensive experiments on a wide variety of temporal networks from different domains. The temporal network data used in our experiments is shown in Table 2. We discuss baseline comparisons in Section 5.1, and perform a detailed ablation study that shows the contributions of the different components and design choices of Algorithm 1 in Section 5.2. The experiments systematically investigate the effectiveness of the framework for estimating *temporal link strength* (Section 5.2), *temporally weighted motifs* using the decay model (Section 5.2), and *temporal network statistics* (Section 5.3). We use sample fractions $p = \{0.10, 0.20, 0.30, 0.40, 0.50\}$ and all experiments are the average of five different runs, similar to the setup in prior work [54].

5.1 Comparison to Published Baselines

In Table 3, we compare Algorithm 1 to the state-of-the-art methods for multi-graph streams (in which edges can appear more than once in the stream), Triest sampling [84], reservoir sampling [91], and MultiWMascot [54] for the estimation of link strength (with no-decay). Note that both Triest and reservoir sampling methods sample edges separately, and multiple occurrences of an edge (u, v) may appear in the final sample. On the other hand, our proposed Algorithm 1 and MultiWMascot incrementally update the overall estimate of link strength of an edge (u, v), and stores an edge only once with its estimator. This leads to more space-efficient samples. However, while MultiWMascot maintains the edge estimators, it is still uses a single uniform probability p for sampling any of the edges.

Table 2. Temporal network data [76]. Note $|K|$ is the number of static edges (not including multiplicities); $|E|$= number of temporal edges; and $C_{\max}$= maximum edge weight.

Temporal Network	$\lvert V\rvert$	$\lvert K\rvert$	$\lvert E\rvert$	**days**	$C_{\max}$
sx-stackoverflow	2.6M	28.1M	47.9M	2774.3	1.04k
ia-facebook-wall-wosn	46k	183k	877k	1591.0	1.3k
wiki-talk	1.1M	2.8M	7.8M	2320.4	1.6k
bitcoin	24.5M	86.1M	129.2M	1811.7	72.6k
CollegeMsg	1.9k	14k	60k	193.7	184
ia-retweet-pol	18k	48k	61k	48.8	79
ia-prosper-loans	89k	3.3M	3.4M	2142.0	15
comm-linux-reply	26k	155k	1.0M	2921.6	1.9k
email-dnc	1.9k	4.4k	39k	982.3	634
ia-enron-email	87k	297k	1.1M	16217.5	1.4k
ia-contacts-dublin	11k	45k	416k	80.4	345
fb-forum	899	7.0k	34k	164.5	171
ia-contacts-hyper09	113	2.2k	21k	2.5	1.3k
SFHH-conf-sensor	403	9.6k	70k	1.3	1.2k
sx-superuser	192k	715k	1.4M	2773.3	139
sx-askubuntu	157k	456k	964k	2613.8	215
sx-mathoverflow	25k	188k	507k	2350.3	325

Table 3. Baseline Comparison: Relative spectral norm (*i.e.*, $\|\mathrm{C}-\widehat{\mathrm{C}}\|_2/\|\mathrm{C}\|_2$) for sampling fraction $p = 0.1$, comparison between Online-TNS (Alg 1), Triest sampling [84], Reservoir sampling [91], and MultiWMascot sampling [54].

Temporal Network	**Online-TNS**	**Triest**	**Reservoir**	**MultiWMascot**
CollegeMsg	0.0558	0.2304	0.2212	0.1990
ia-retweet-pol	0.1800	0.4103	0.4091	0.3973
ia-contacts-dublin	0.0215	0.1926	0.1937	0.1855
wiki-talk	0.1020	0.2554	0.2347	0.2343
fb-forum	0.0390	0.1900	0.1912	0.2052
sx-mathoverflow	0.0668	0.1767	0.1764	0.1584
sx-stackoverflow	0.0992	0.2114	0.2036	0.2045

We observe that both Triest and reservoir sampling were unable to produce a reasonable estimate with 59% − 83% accuracy, while MultiWMascot performed slightly better with an accuracy of 60% − 84%. Algorithm 1 produced more accurate estimates with 82% − 97% accuracy, with an average of 20% gain in accuracy compared to the baselines. Figure 2 shows the distribution of the top-k edges ($k = 10$ million) and compares the exact link strength with the estimated link strength for the four sampling algorithms, Online-TNS (Alg. 1), Triest, reservoir sampling, and MultiWMascot. Notably, Online-TNS not only accurately estimates the strength of the link but also captures the correct order of the links compared to the baselines.

5.2 Ablation Study

Our proposed framework is flexible and generic with various components and design choices. To help understand the contributions of the major components, we performed a thorough set of ablation study experiments. Our framework (in Algorithm 1) consists of two major components: Adaptive sampling and estimation (uniform vs adaptive sampling weights), and Link-decay models (no-decay

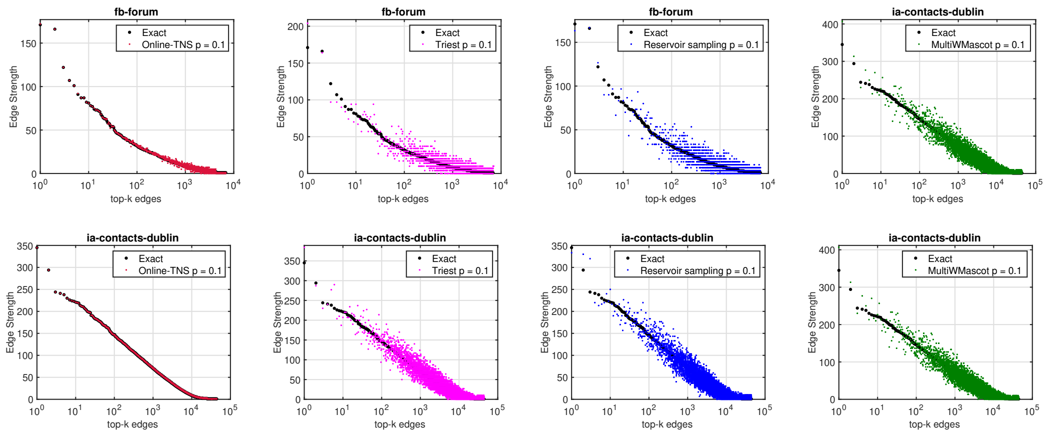

Fig. 2. Baseline comparison with Triest, reservoir sampling, and MultiWMascot sampling. Temporal link strength (No-decay) estimated distribution vs exact distribution for top-k links . Results are shown for sampling fraction $p = 0.1$.

Sampling Weights	Link Strength Estimator: No decay	Link Strength Estimator: Temporal decay
Uniform	$w(e) = \phi$ $\widehat{C}(e)$ using Line 25	$w(e) = \phi$ $\widehat{C}(e)$ using Line 32
Adaptive	**if** $e \in \widehat{K}$ $w(e) = w(e) + 1$ **else** $w(e) = \phi$ $\widehat{C}(e)$ using Line 25	**if** $e \in \widehat{K}$ $w(e) = w(e) + 1$ **else** $w(e) = \phi$ $\widehat{C}(e)$ using Line 32

Table 4. Online-TNS Framework Main Components (see Algorithm 1).

vs exponential decay). We summarize these components and design choices in Table 4. Our first ablation study experiment investigates the impact of sampling weights on the estimation accuracy. We explore two variants of Algorithm 1, (a) *Adaptive*: using Algorithm 1 with adaptive/importance sampling weights, where the sampling weights/ranks adapt to allow edges to gain importance during stream processing. (b) *Uniform*: using Algorithm 1 with fixed uniform sampling weights, where the sampling weights/ranks are uniform and assigned at the first time of sampling, and fixed during the rest of the streaming process. For variant (a), we use the exact procedure in Algorithm 1. For variant (b), we omit Lines 9 and 10 from Algorithm 1. Note that for both variants, the established estimators in Section 4 are unbiased. We compare performance of the two variants for the estimation of link strength and temporally weighted motif counts.

Estimation of Temporal Link Strength. Link strength is one of the most fundamental properties of temporal networks [93]. Therefore, estimating it in an online fashion is clearly important. Results using Alg. 1 with adaptive sampling weights and unbiased estimators (see Section 4) for temporal link strength estimation are provided in Figures 3 and 4 (top row) for the no-decay and link-decay models respectively. We show the distribution of the top-k edges ($k = 10$ million) and compare the

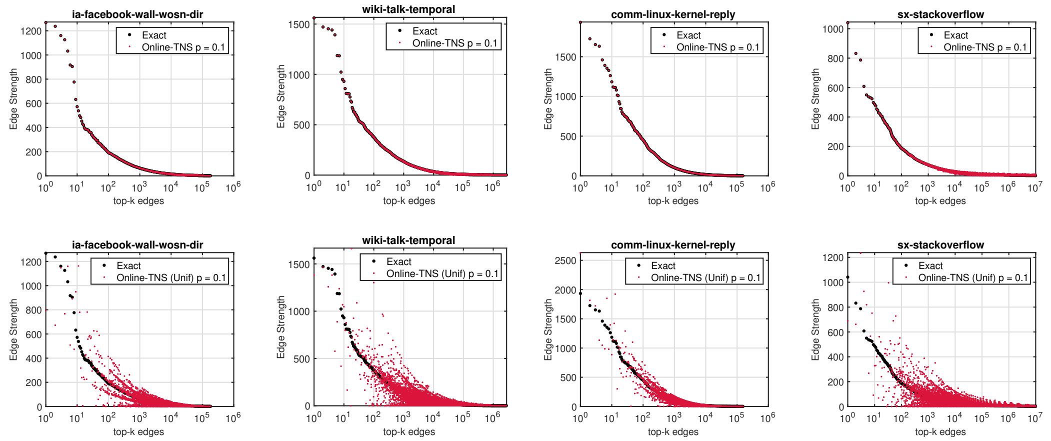

Fig. 3. Temporal link strength (No-decay) estimated distribution vs exact distribution for top-k links . Results are shown for sampling fraction $p = 0.1$. (Top) Results for Online-TNS Algorithm with adaptive sampling weights 1. (Bottom) Results for Online-TNS Algorithm with uniform sampling weights.

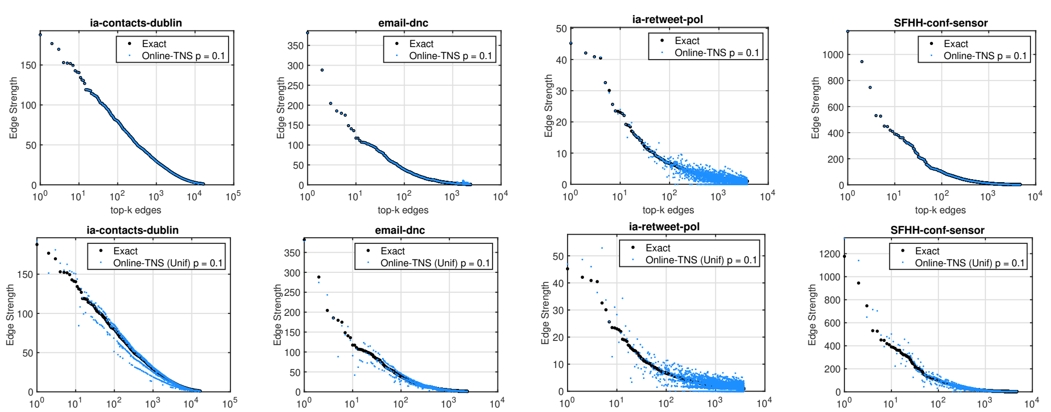

Fig. 4. Temporal decay link strength estimated distribution vs exact distribution for top-k links . Results are shown for sampling fraction $p = 0.1$. (Top) Results for Online-TNS Algorithm with adaptive sampling weights 1. (Bottom) Results for Online-TNS Algorithm with uniform sampling weights.

exact link strength with the estimated link strength. Notably our approach not only accurately estimates the strength of the link but also captures the correct order of the links (top-links ordered by their strength from high to low). From Figures 3 and 4 (top row), we observe the exact and estimated link strengths for the top-k edges to be nearly indistinguishable from one another. We also compare to Alg. 1 with uniform sampling weights, *i.e.*, Online-TNS (Unif), in which edges are assigned uniform sampling weights at the sampling time, and fixed for the rest of the streaming process, results are shown in Figures 3 and 4 (bottom row). While the estimated link strengths from Online-TNS with adaptive weights are nearly identical to the exact link strengths, estimated distributions from uniform sampling weights are significantly worse. Unlike uniform sampling weights where the weights remain constant, using the adaptive sampling weights helps the algorithm to adapt to the changing topology of the streaming network, which leads to favoring the retention of edges with higher strength.

Table 5. No-decay Results: Relative spectral norm (*i.e.*, $\|\mathbf{C} - \widehat{\mathbf{C}}\|_2 / \|\mathbf{C}\|_2$) for sampling fraction $p = 0.1$, comparison between adaptive sampling weights (Alg. 1) and uniform sampling weights.

Temporal Network	**Adaptive (Alg. 1)**	**Uniform**
ia-facebook-wall	0.0090	0.3976
sx-stackoverflow	0.0992	0.4360
wiki-talk	0.1020	0.4272
comm-linux-reply	0.0041	0.1978
fb-forum	0.0390	0.2640
ia-enron-email	0.0098	0.4080
SFHH-conf-sensor	0.0090	0.2770
ia-contacts-hyper	0.0034	0.0529

Table 6. Link-decay Results: Relative spectral norm (*i.e.*, $\|\mathbf{C} - \widehat{\mathbf{C}}\|_2 / \|\mathbf{C}\|_2$) for sampling fraction $p = 0.1$, comparison between adaptive sampling weights (Alg. 1) and uniform sampling weights.

Temporal Network	**Adaptive (Alg. 1)**	**Uniform**
CollegeMsg	0.0797	0.0819
ia-retweet-pol	0.1451	0.2723
ia-contacts-dublin	0.0058	0.1624
ia-facebook-wall	0.0335	0.0914
ia-contacts-hyper	0.0009	0.0511
SFHH-conf-sensor	0.0034	0.2283
email-dnc	0.0143	0.1506

In Tables 5 and 6, we show the relative spectral norm for online-TNS with adaptive and uniform sampling weights for no-decay and link-decay models respectively. The relative spectral norm is defined as $\|\mathbf{C} - \widehat{\mathbf{C}}\|_2 / \|\mathbf{C}\|_2$, where $\mathbf{C}$ is the exact time-dependent adjacency matrix of the input graph, whose entries represent the link strength, $\widehat{\mathbf{C}}$ is the average estimated time-dependent adjacency matrix (estimated from the sample), and $\|\mathbf{C}\|_2$ is the spectral norm of $\mathbf{C}$. The spectral norm $\|\mathbf{C} - \widehat{\mathbf{C}}\|_2$ is widely used for matrix approximations [1]. $\|\mathbf{C} - \widehat{\mathbf{C}}\|_2$ measures the strongest linear trend of $\mathbf{C}$ not captured by the estimate $\widehat{\mathbf{C}}$. The results show Online-TNS with adaptive sampling weights significantly outperforms the variant using uniform sampling weights, and captures the linear trend and structure of the data better, with an average 20% improvement over the uniform sampling weights. We also measured the error using relative Frobenius norm (*i.e.*, $\|\mathbf{C} - \widehat{\mathbf{C}}\|_F / \|\mathbf{C}\|_F$) and observed similar conclusions.

Temporally Weighted Motif Estimation. Recall that our formulation of temporal motif differs from previous work in that instead of counting motifs that occur within some time period δ, our formulation focuses on counting temporally weighted motifs where the motifs are weighted such that motifs that occur more recent and contain active links are assigned larger weight than those occurring in the distant past. This formulation is clearly more useful and important, since it can capture the evolution of the network and relationships at a continuous-time scale. Also, this formulation would be useful for many practical applications involving prediction and forecasting since it appropriately accounts for temporal statistics (in this case, motifs) that occur more recently, which are by definition more predictive of some future event. In Table 7, we show results for

Table 7. Results for temporally weighted motif count estimation. Online TNS with adaptive weights compared to online TNS with uniform weights. Relative error for triangle counts are reported using $p = 0.1$.

	Without Decay			With Decay		
Temporal Network	**Exact**	**Adaptive**	**Uniform**	**Exact**	**Adaptive**	**Uniform**
sx-stackoverflow	15B	0	0.0133	168M	0.00004	0.0006
ia-facebook-wall	435M	0.0004	0.0605	9.9M	0.0004	0.0247
wiki-talk	12B	0.0003	0.0171	394M	0.0001	0.0022
CollegeMsg	6.2M	0.0148	0.0545	2.0M	0.0003	0.0277
ia-retweet-pol	380k	0.0236	0.0341	147k	0.0001	0.0247
ia-prosper-loans	1.4M	0.0056	0.0078	232k	0.0067	0.0048
comm-linux-reply	148B	0	0.0055	242M	0.00003	0.000024
email-dnc	483M	0	0.0004	251M	0.00006	0.0097
ia-enron-email	14B	0	0.0726	329M	0.0003	0.0034
ia-contacts-dublin	382M	0.00005	0.00008	381M	0	0.0001
fb-forum	3.3M	0.0036	0.1761	763k	0.0067	0.0137
ia-contacts-hyper09	93M	0	0.00041	88M	0	0.00086
SFHH-conf-sensor	622M	0	0.0665	604M	0.0002	0.0434
sx-superuser	83M	0.0072	0.0137	2.1M	0.0013	0.0016
sx-askubuntu	71M	0.0035	0.0077	2.7M	0.0069	0.0135
sx-mathoverflow	269M	0.0008	0.0688	2.8M	0.0004	0.0003

estimating the temporally weighted motif counts. For brevity, we only show results for triangle motifs (both decay and no-decay models), but the proposed framework and unbiased estimators in Algorithm 1 and Section 4 generalize to any network motifs of larger size. For these results, we set the decay factor δ to 30 days. Notably, all of the temporally decayed motif count estimates have a relative error that is less than 0.03 as shown in Table 7 (Adaptive). Nevertheless, this demonstrates that our efficient temporal sampling framework is able to leverage accurate estimators for even the smallest sample sizes. Table 7 also shows results of Algorithm 1 with uniform sampling weights. Overall, we observe that Online-TNS with adaptive sampling weights generally outperforms Online-TNS with uniform sampling weights. We conjecture that Online-TNS with adaptive sampling is general and would be useful in various applications beyond the scope of this paper. In particular, for applications that require importance sampling with the ability to combine both topology (*e.g.*, edge multiplicity, temporal strength, subgraphs) and auxiliary information (*e.g.*, node/edge attributes and features). We will explore these applications in future work.

Sensitivity Analysis of the Decay Factor. We now study the impact of the choice of the decay factor δ on the quality of the estimates. When we choose the value for the mean lifetime δ, it is more intuitive to think about the half-life $\eta_{1/2}$ of an edge. The half-life of an edge gives the amount of time for an edge to lose half of its weight/strength in the absence of new interactions. Given $\delta > 0$, the half-life of an edge is defined as $\eta_{1/2} = \delta \ln 2$. As such the choice of δ is crucial to filter-out and down-weight the old activity in continuously evolving networks. When the half-life is short (*i.e.*, the mean lifetime δ is short), the interactions result in weak links among nodes, where the link strength dies off quickly unless the interactions occur more frequently and sustainably among the nodes. On the other hand, when the half-life is long (*i.e.*, δ is long), links are able to build a momentum and strengthen from interactions that otherwise occurred too far in time. In Figure 5, we show the estimated link strengths obtained using Online-TNS with adaptive weights for two different decay factors δ = 1-day and δ = 7-days. Clearly, the range and scale of the link strength is much higher when δ is long. In addition, we also observe that the estimated distributions of link strengths for the top-k edges are nearly indistinguishable from the exact distributions. This is due to the unbiasedness property of the proposed estimators, as the unbiasedness property holds regardless the choice of the decay parameter δ. In Table 8, we provide the temporally weighted motif count estimation obtained using Online TNS (Alg 1) with adaptive weights and different decay δ parameters. Clearly, the relative error is small for δ values, which is a result of the unbiasedness property of the proposed estimators.

Table 8. Sensitivity analysis of the decay factor δ. Results for temporally weighted motif count estimation. Relative error reported using $p = 0.1$ for triangle counts using Online TNS (Alg 1) with adaptive weights and different decay δ parameters.

	Decay Factor δ		
Temporal Network	**1-day**	**1-week**	**1-month**
CollegeMsg	0.002	0.003	0.0003
ia-retweet-pol	0.074	0.023	0.0001
ia-contacts-dublin	0.00003	0	0
ia-facebook-wall	0.0091	0.0011	0.0004
email-dnc	0.00004	0.00005	0.00006

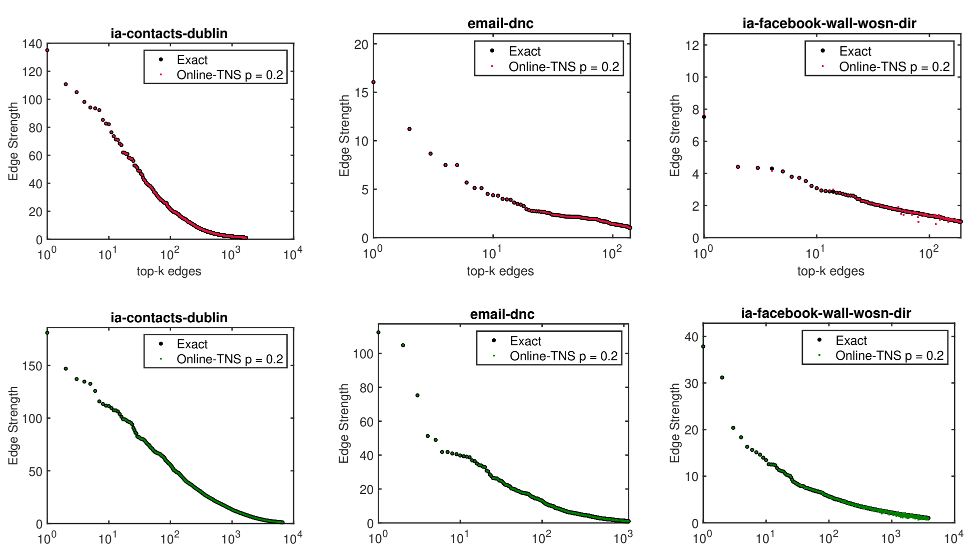

Fig. 5. Temporal link strength estimated distribution vs exact distribution for top-k links. Results are shown for sampling fraction $p = 0.2$. (Top: Red) Results for Online-TNS Algorithm 1 with adaptive sampling weights and decay factor δ = 1 day. (Bottom: Green) Results for Online-TNS Algorithm 1 with adaptive sampling weights 1 and decay factor δ = 7 days.

Table 9. Results for estimating temporal burstiness. For each temporal network, we show the estimated burstiness using different sampling probabilities (first row) compared to the exact. The relative error $|\widehat{B}-B|/B$ of the estimates is also shown.

Temporal Network	Sampling Fraction 0.1	0.2	0.3	0.4	0.5	Exact
wiki-talk	0.6196	0.6208	0.6208	0.6207	0.6206	0.6206
(error)	0.0017	0.0003	0.0003	<0.0001	<0.0001	
ia-facebook-wall-wosn	0.4482	0.4534	0.4535	0.4535	0.4535	0.4535
(error)	0.0116	0.0002	$<10^{-5}$	$<10^{-5}$	$<10^{-5}$	
bitcoin	0.7738	0.7642	0.7606	0.7586	0.7579	0.7576
(error)	0.0214	0.0087	0.0040	0.0013	0.0004	
sx-stackoverflow	0.6517	0.6712	0.6808	0.6863	0.6891	0.6898
(error)	0.0552	0.0269	0.0130	0.0050	0.0010	

5.3 Estimation of Temporal Statistics

While the proposed framework can be used to obtain unbiased estimates of arbitrary temporal network statistics, we focus in this section on two important temporal properties and their distributions including burstiness [16] and temporal link persistence [25]. For a survey of other important temporal network statistics that are applicable for estimation using the framework, see [43].

Burstiness. Burstiness B is widely used to characterize the link activity in temporal networks [43]. Burstiness is computed using the mean μ and standard deviation σ of the distribution of same-edge inter-contact times collected from all links, *i.e.*, $B = (\sigma - \mu)/(\sigma + \mu)$. The inter-contact time is the elapsed time between two subsequent same-edge interactions (*i.e.*, time between two text messages from the same pair of friends). Burstiness measures the deviation of relationship activity from a Poisson process. In Table 9, we use the proposed framework to estimate burstiness (*i.e.*, computed using the sampled network). We show the estimated burstiness for sampling fraction $p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. In addition, we also provide the relative error of the estimates across the different sampling fractions. From Table 9, we observe the relative errors are small and the estimates are shown to converge as the sampling fraction p increases. In Figure 6, we show the exact and estimated distribution of inter-contact times for sampling fractions $p = 0.1$ (top row) and $p = 0.2$ (bottom row). We observe that the estimated distribution from the sample accurately captures the exact distribution.

Temporal Link Persistence. The persistence of an edge measures the lifetime of relationships, and is computed as the elapsed time between the first interaction and the last interaction of the same edge [43]. Let L denote the average link persistence (or lifetime) computed over all edges in the full (sampled) network defined as $L = \frac{1}{|K|} \sum_{(i,j) \in K} \tau_{ij}^{(\text{last})} - \tau_{ij}^{(\text{first})}$. Relative error of estimated link persistence is shown in Table 10. In Figure 7, we show the exact and estimated distribution (*i.e.*, computed using the sampled network) of link persistence scores for sampling fractions $p = 0.1$ (top row) and $p = 0.2$ (bottom row). We observe that the estimated distributions from the sampled network across all graphs accurately captures the exact distribution (for both burstiness and persistence). We also note that the proposed algorithm Alg 1 can also handle labeled graphs (with vertex/edge categorical variables), where the estimators are computed separately for each possible label combination in a stratified fashion.

Table 10. Estimation results for temporal persistence. For each temporal network, we report relative error $|\widehat{L}-L|/L$ of the estimates using different sampling fractions.

	Sampling Fraction				
Temporal Network	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
wiki-talk	0.1380	0.0412	0.0112	0.0009	$<10^{-6}$
ia-facebook-wall-wosn	0.1232	0.0023	$<10^{-7}$	$<10^{-7}$	$<10^{-7}$
sx-stackoverflow	0.1718	0.0794	0.0357	0.0119	0.0016
bitcoin	0.1056	0.0207	0.0023	0.0020	0.0024

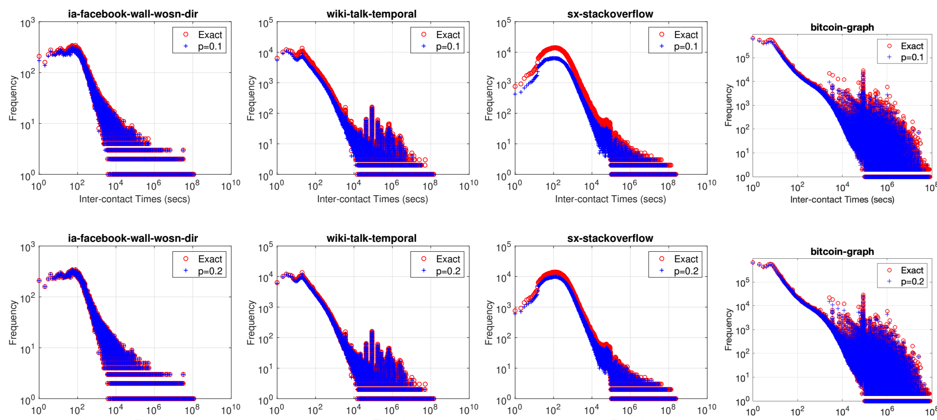

Fig. 6. Estimation results for the distribution of inter-contact times compared to the exact distribution. Results are shown for $p = 0.1$ (top) and $p = 0.2$ (bottom).

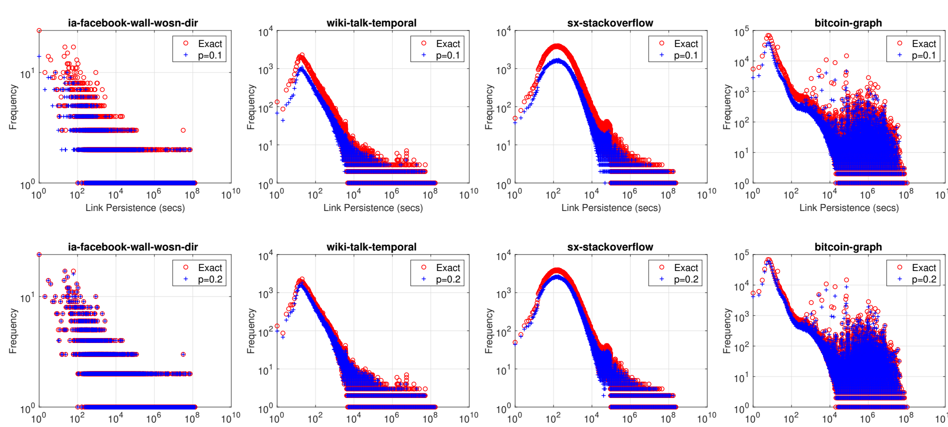

Fig. 7. Estimation results for the distribution of link persistence scores compared to the exact distribution. Results are shown for $p = 0.1$ (top) and $p = 0.2$ (bottom).

6 RELATED WORK

Sampling algorithms are fundamental in studying and understanding networks [11, 47, 67, 91], where the goal is to collect a representative sample that capture the characteristics of the full network. Network sampling has been widely studied in the context of small static networks that can fit entirely in memory [47]. For instance, there is uniform node sampling [85], random walk sampling[52], edge sampling [11], among others [15, 56]. More recently, there has been a growing interest in sampling techniques for streaming network data in which temporal networks evolve continuously in time [6–10, 27, 44, 45, 54, 55, 71, 82, 84, 95]. For seminal surveys on the topic, see [11, 62].

However, most existing methods for sampling streaming network data have focused on the primary objective of selecting a sample to estimate static network properties, *e.g.*, point statistics such as global triangle count or clustering coefficient [9]. As such, it is unclear how representative these samples are for temporal network statistics such as the link strength [93], link persistence [25], burstiness [16], temporal motifs [48], among others [43]. Despite the fundamental importance of this question, it has not been addressed in the context of streaming and online methods.

Some of the recent work on stream sampling focused on multi-graph streams [54, 84, 91], where edges may appear multiple times in the stream. However, most of these methods (e.g., Triest, reservoir sampling) mainly sample edges separately, thus, multiple occurrences of an edge (u, v) may appear in the final sample, which allocates more space. In addition, the recent work in [54] uses a uniform sampling probability to sample the edges, and stores an edge only once with its estimator. On the other hand, our proposed Algorithm 1 adaptively samples edges with probabilities proportional to their link strength, and incrementally updates the overall estimate of link strength of an edge (u, v), and stores an edge only once with its estimator. This leads to more space-efficient and accurate samples.

There has been one recent work for sampling temporal motifs [56]. However, their work focused on a different problem based on counting motifs in temporal networks that form within some time $\triangle t$ [48]. More specifically, their approach uses judicious partitioning of interactions in time bins, which can obfuscate or dilute temporal and structural information. In this paper, we formulate instead the notion of a *temporally weighted motif* based on the temporal network link decay model. We argue that this formulation is more meaningful and useful for practical applications especially related to prediction and forecasting where links and motifs that occur more recently are more important than those occurring in the distant past. In addition to the difference in problem, that work does *not* focus on streaming nor the online setting since the entire graph is loaded into memory.

The temporally decaying model of temporal networks is useful for many important predictive modeling and forecasting tasks including classification [75, 81], link prediction [20, 22, 23, 30, 34, 65, 66, 92, 96], influence modeling [39], regression [38], and anomaly detection [5, 79]. Despite the practical importance of the temporal link decaying model, our work is the first to propose network sampling and unbiased estimation algorithms for this setting. Therefore, the proposed temporal decay sampling and unbiased estimation methods bring new opportunities for many real-world applications that involve prediction and forecasting from temporal networks representing a sequence of timestamped edges. This includes recommendation [18, 30], influence modeling [39], visitor stitching [78], etc.

Moreover, there has been a lot of research on deriving new and important temporal network statistics and properties that appropriately characterize the temporal network [43]. Other recent work has focused on extending node ranking and importance measures to dynamic networks such as Katz [40] and eigenvector centrality [89]. These centrality measures use a sequence of static snapshot graphs to compute an importance or node centrality score of nodes. Since the proposed temporal sampling framework is general and can be used to estimate a time-dependent representation of the temporal network, it can be used to obtain unbiased estimates of these recent dynamic node centrality measures.

The proposed temporal network sampling framework can also be leveraged for estimation of node embeddings [80] including both community-based (proximity) and role-based structural node embeddings [14, 77]. More recently, there has been a surge in activity for developing node embedding and graph representation learning methods for temporal networks. There have been embedding methods proposed for both continuous-time dynamic networks consisting of a stream of timestamped edges [51, 57, 69] as well as discrete-time dynamic networks where the actual edge stream is approximated with a sequence of static snapshot graphs [59, 70, 79, 87, 88]. All of these works may benefit from the proposed framework as it estimates a time-dependent representation of the temporal network that can be used as input to any of these methods for learning time-dependent node embeddings.

In the context of accumulating sample based counts of repeated objects, Sticky Sampling [60] and counting Samples [37] have been proposed, together with Sample and Hold [32] in the context of network measurement, along with adaptive versions for fixed size reservoirs [26, 46]. Our approach differs from these methods in many ways and provide the following advantages. First, the cost of updating the sample is much cheaper compared to these methods. Second, the discard step is computationally cheaper being $O(1)$ to pick the minimum priority element. Third, our approach provides unbiased estimators not only for single links, but also for link-product counts for temporal motifs through Theorem 1(iii).

7 CONCLUSION

This work proposed a novel general framework for online sampling and unbiased estimation of temporal networks. The framework gives rise to online single-pass streaming sampling algorithms for estimating arbitrary temporal network statistics. We also proposed a temporal decay sampling algorithm for estimating statistics based on the temporal decay model that assumes the strength of links evolve as a function of time, and the temporal statistics and temporal motif patterns are temporally weighted accordingly. To the best of our knowledge, this work is the first to propose sampling and unbiased estimation algorithms for this setting, which is fundamentally important for practical applications involving prediction and forecasting from temporal networks. The proposed framework and temporal network sampling algorithms that arise from it, enable fast, accurate, and memory-efficient statistical estimation of temporal network patterns and properties. Finally, the experiments demonstrated the effectiveness of the proposed approach for unbiased estimation of temporal network statistics. Other graph properties such as page rank, degree distribution, and centrality would be suitable for future extensions of the proposed framework.

REFERENCES

[1] Dimitris Achlioptas, Zohar S Karnin, and Edo Liberty. 2013. Near-optimal entrywise sampling for data matrices. In *NeurIPS*. 1565–1573.
[2] Charu C Aggarwal. 2006. On biased reservoir sampling in the presence of stream evolution. In *Proceedings of the 32nd international conference on Very large data bases*. 607–618.
[3] Charu C Aggarwal. 2018. Extracting Real-Time Insights from Graphs and Social Streams. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1339–1339.
[4] Charu C. Aggarwal, Yao Li, and Philip S. Yu. [n.d.]. *On Supervised Change Detection in Graph Streams*. 289–297.
[5] Charu C Aggarwal, Yuchen Zhao, and S Yu Philip. 2011. Outlier detection in graph streams. In *ICDE*. IEEE, 399–409.
[6] Nesreen K Ahmed, Fredrick Berchmans, Jennifer Neville, and Ramana Kompella. 2010. Time-based sampling of social network activity graphs. In *MLG*. 1–9.

[7] Nesreen K Ahmed and Nick Duffield. 2020. Adaptive Shrinkage Estimation for Streaming Graphs. In *Advances in Neural Information Processing Systems*.

[8] Nesreen K Ahmed, Nick Duffield, Jennifer Neville, and Ramana Kompella. 2014. Graph sample and hold: A framework for big-graph analytics. In *KDD*. 1446–1455.

[9] Nesreen K Ahmed, Nick Duffield, Theodore L Willke, and Ryan A Rossi. 2017. On sampling from massive graph streams. *Proceedings of the VLDB Endowment* 10, 11 (2017), 1430–1441.

[10] Nesreen K. Ahmed, Nick Duffield, and Liangzhen Xia. 2018. Sampling for Approximate Bipartite Network Projection. In *IJCAI*. 3286–3292.

[11] Nesreen K Ahmed, Jennifer Neville, and Ramana Kompella. 2014. Network sampling: From static to streaming graphs. *TKDD* 8, 2 (2014), 7.

[12] Nesreen K Ahmed, Jennifer Neville, Ryan A Rossi, and Nick Duffield. 2015. Efficient graphlet counting for large networks. In *ICDM*. 1–10.

[13] Nesreen K Ahmed, Jennifer Neville, Ryan A Rossi, Nick G Duffield, and Theodore L Willke. 2017. Graphlet decomposition: Framework, algorithms, and applications. *KAIS* 50, 3 (2017), 689–722.

[14] Nesreen K Ahmed, Ryan Rossi, John Boaz Lee, Theodore L Willke, Rong Zhou, Xiangnan Kong, and Hoda Eldardiry. 2018. Learning role-based graph embeddings. *arXiv:1802.02896* (2018).

[15] Nesreen K Ahmed, Theodore L Willke, and Ryan A Rossi. 2016. Estimation of local subgraph counts. In *IEEE Big Data*. IEEE, 586–595.

[16] Albert-Laszlo Barabasi. 2005. The origin of bursts and heavy tails in human dynamics. *Nature* 435, 7039 (2005), 207.

[17] Austin R Benson, David F Gleich, and Jure Leskovec. 2016. Higher-order organization of complex networks. *Science* 353, 6295 (2016), 163–166.

[18] Homanga Bharadhwaj and Shruti Joshi. 2018. Explanations for Temporal Recommendations. *KI* 32, 4 (01 Nov 2018), 267–272.

[19] Rajmonda Sulo Caceres, Tanya Berger-Wolf, and Robert Grossman. 2011. Temporal scale of processes in dynamic networks. In *ICDM Workshops*. IEEE, 925–932.

[20] Huiyuan Chen and Jing Li. 2018. Exploiting structural and temporal evolution in dynamic link prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 427–436.

[21] Wei Chen, Laks VS Lakshmanan, and Carlos Castillo. 2013. Information and influence propagation in social networks. *Syn. Lec. on Data Man.* 5, 4 (2013).

[22] Kuo Chi, Guisheng Yin, Yuxin Dong, and Hongbin Dong. 2019. Link prediction in dynamic networks based on the attraction force between nodes. *Knowledge-Based Systems* 181 (2019), 104792.

[23] Pankaj Choudhary, Nishchol Mishra, Sanjeev Sharma, and Ravindra Patel. 2013. Link score: A novel method for time aware link prediction in social network. *ICDMW* (2013).

[24] Sutanay Choudhury, Lawrence Holder, George Chin, Abhik Ray, Sherman Beus, and John Feo. 2013. Streamworks: a system for dynamic graph search. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 1101–1104.

[25] Aaron Clauset and Nathan Eagle. 2012. Persistence and periodicity in a dynamic proximity network. *arXiv:1211.7343* (2012).

[26] Edith Cohen. 2015. Stream Sampling for Frequency Cap Statistics. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Sydney, NSW, Australia) *(KDD '15)*. Association for Computing Machinery, New York, NY, USA, 159–168. https://doi.org/10.1145/2783258.2783279

[27] Graham Cormode and Hossein Jowhari. 2014. A second look at counting triangles in graph streams. *Theoretical Computer Science* 552 (2014), 44–51.

[28] Graham Cormode, Vladislav Shkapenyuk, Divesh Srivastava, and Bojian Xu. 2009. Forward Decay: A Practical Time Decay Model for Streaming Systems. In *Proceedings of the 2009 IEEE International Conference on Data Engineering (ICDE '09)*. IEEE Computer Society, USA, 138–149. https://doi.org/10.1109/ICDE.2009.65

[29] Nick Duffield, Carsten Lund, and Mikkel Thorup. 2007. Priority sampling for estimation of arbitrary subset sums. *Journal of the ACM (JACM)* 54, 6 (2007), 32.

[30] Daniel M Dunlavy, Tamara G Kolda, and Evrim Acar. 2011. Temporal link prediction using matrix and tensor factorizations. *TKDD* 5, 2 (2011), 10.

[31] Jean-Pierre Eckmann, Elisha Moses, and Danilo Sergi. 2004. Entropy of dialogues creates coherent structures in e-mail traffic. *PNAS* 101, 40 (2004), 14333–14337.

[32] Cristian Estan and George Varghese. 2003. New Directions in Traffic Measurement and Accounting: Focusing on the Elephants, Ignoring the Mice. *ACM Trans. Comput. Syst.* 21, 3 (Aug. 2003), 270–313. https://doi.org/10.1145/859716.859719

[33] Daniel J Fenn, Mason A Porter, Peter J Mucha, Mark McDonald, Stacy Williams, Neil F Johnson, and Nick S Jones. 2012. Dynamical clustering of exchange rates. *Quantitative Finance* 12, 10 (2012), 1493–1520.

[34] Érick S Florentino, Argus AB Cavalcante, and Ronaldo R Goldschmidt. 2020. An edge creation history retrieval based method to predict links in social networks. *Knowledge-Based Systems* 205 (2020), 106268.

[35] Julio Flores and Miguel Romance. 2018. On eigenvector-like centralities for temporal networks: Discrete vs. continuous time scales. *J. Comput. Appl. Math.* 330 (2018), 1041–1051.

[36] Timothy La Fond, Jennifer Neville, and Brian Gallagher. 2018. Designing size consistent statistics for accurate anomaly detection in dynamic networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12, 4 (2018), 1–49.

[37] Phillip B. Gibbons and Yossi Matias. 1999. Synopsis Data Structures for Massive Data Sets. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms* (Baltimore, Maryland, USA) *(SODA '99)*. Society for Industrial and Applied Mathematics, USA, 909–910.

[38] David F. Gleich and Ryan A. Rossi. 2014. A Dynamical System for PageRank with Time-Dependent Teleportation. *Internet Mathematics* 10, 1-2 (2014), 188–217.

[39] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. 2010. Learning influence probabilities in social networks. In *WSDM*. ACM, 241–250.

[40] Peter Grindrod, Mark C Parsons, Desmond J Higham, and Ernesto Estrada. 2011. Communicability across evolving networks. *Phy. Rev. E* 83, 4 (2011), 046120.

[41] Petter Holme. 2015. Modern temporal network theory: a colloquium. *The European Physical Journal B* 88, 9 (2015), 234.

[42] Petter Holme and Luis EC Rocha. 2019. Impact of misinformation in temporal network epidemiology. *Network Science* 7, 1 (2019), 52–69.

[43] Petter Holme and Jari Saramäki. 2012. Temporal networks. *Physics reports* 519, 3 (2012), 97–125.

[44] Madhav Jha, Ali Pinar, and C Seshadhri. 2015. Counting triangles in real-world graph streams: Dealing with repeated edges and time windows. In *49th Asilomar Conference on Signals, Systems and Computers*. IEEE, 1507–1514.

[45] Madhav Jha, Comandur Seshadhri, and Ali Pinar. 2013. A space efficient streaming algorithm for triangle counting using the birthday paradox. In *KDD*. 589–597.

[46] Ken Keys, David Moore, and Cristian Estan. 2005. A Robust System for Accurate Real-Time Summaries of Internet Traffic. In *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (Banff, Alberta, Canada) *(SIGMETRICS '05)*. Association for Computing Machinery, New York, NY, USA, 85–96. https://doi.org/10.1145/1064212.1064223

[47] Eric D Kolaczyk and Gábor Csárdi. 2014. *Statistical analysis of network data with R*. Vol. 65. Springer.

[48] Lauri Kovanen, Márton Karsai, Kimmo Kaski, János Kertész, and Jari Saramäki. 2011. Temporal motifs in time-dependent networks. *Journal of Statistical Mechanics: Theory and Experiment* 2011, 11 (2011), P11005.

[49] Timothy La Fond, Geoffrey Sanders, Christine Klymko, et al. 2017. An ensemble framework for detecting community changes in dynamic networks. In *2017 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, 1–6.

[50] Timothy LaRock, Timothy Sakharov, Sahely Bhadra, and Tina Eliassi-Rad. 2020. Understanding the Limitations of Network Online Learning. *arXiv preprint arXiv:2001.07607* (2020).

[51] John Boaz Lee, Giang Nguyen, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim. 2019. Temporal Network Representation Learning. *arXiv:1904.06449* (2019).

[52] Jure Leskovec and Christos Faloutsos. 2006. Sampling from large graphs. In *KDD*. 631–636.

[53] Aming Li, Sean P Cornelius, Y-Y Liu, Long Wang, and A-L Barabási. 2017. The fundamental advantages of temporal networks. *Science* 358, 6366 (2017), 1042–1046.

[54] Yongsub Lim, Minsoo Jung, and U Kang. 2018. Memory-efficient and accurate sampling for counting local triangles in graph streams: from simple to multigraphs. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12, 1 (2018), 1–28.

[55] Yongsub Lim and U Kang. 2015. Mascot: Memory-efficient and accurate sampling for counting local triangles in graph streams. In *KDD*. ACM, 685–694.

[56] Paul Liu, Austin R Benson, and Moses Charikar. 2019. Sampling methods for counting temporal motifs. In *WSDM*. 294–302.

[57] Xi Liu, Ping-Chun Hsieh, Nick Duffield, Rui Chen, Muhe Xie, and Xidao Wen. 2019. Real-Time Streaming Graph Embedding Through Local Actions. In *WWW*. 285–293.

[58] Sharon L Lohr. 2019. *Sampling: Design and Analysis*. Chapman and Hall/CRC.

[59] Sedigheh Mahdavi, Shima Khoshraftar, and Aijun An. 2019. Dynamic Joint Variational Graph Autoencoders. *arXiv:1910.01963* (2019).

[60] Gurmeet Singh Manku and Rajeev Motwani. 2002. Approximate Frequency Counts over Data Streams. In *Proceedings of the 28th International Conference on Very Large Data Bases* (Hong Kong, China) *(VLDB '02)*. VLDB Endowment, 346–357.

[61] Naoki Masuda and Petter Holme. 2013. Predicting and controlling infectious disease epidemics using temporal networks. *F1000prime reports* 5 (2013).

[62] Andrew McGregor. 2014. Graph stream algorithms: a survey. *ACM SIGMOD Record* 43, 1 (2014), 9–20.

[63] 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.

[64] Giovanna Miritello, Esteban Moro, and Rubén Lara. 2011. Dynamical strength of social ties in information spreading. *Physical Review E* 83, 4 (2011), 045102.

[65] Lankeshwara Munasinghe and Ryutaro Ichise. 2012. Time Score: A New Feature for Link Prediction in Social Networks. *IEICE Transactions on Information and Systems* E95.D, 3 (2012), 821–828.

[66] Carlos Pedro Muniz, Ronaldo Goldschmidt, and Ricardo Choren. 2018. Combining contextual, temporal and topological information for unsupervised link prediction in social networks. *Knowledge-Based Systems* 156 (2018), 129–137.

[67] Mark Newman. 2018. *Networks.* Oxford university press.

[68] Mark Ed Newman, Albert-László Ed Barabási, and Duncan J Watts. 2006. *The structure and dynamics of networks.* Princeton university press.

[69] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim. 2018. Continuous-time dynamic network embeddings. In *WWW*. 969–976.

[70] Hogun Park and Jennifer Neville. 2019. Exploiting Interaction Links for Node Classification with Deep Graph Neural Networks. In *IJCAI.* 3223–3230.

[71] A Pavan, Kanat Tangwongsan, Srikanta Tirthapura, and Kun-Lung Wu. 2013. Counting and Sampling Triangles from a Graph Stream. *VLDB* 6, 14 (2013).

[72] Tiago P Peixoto and Laetitia Gauvin. 2018. Change points, memory and epidemic spreading in temporal networks. *Scientific reports* 8, 1 (2018), 15511.

[73] Abhik Ray, Lawrence B Holder, and Albert Bifet. 2019. Efficient frequent subgraph mining on large streaming graphs. *Intelligent Data Analysis* 23, 1 (2019), 103–132.

[74] Luis EC Rocha. 2017. Dynamics of air transport networks: A review from a complex systems perspective. *C. J. of Aero.* 30, 2 (2017), 469–478.

[75] Ryan Rossi and Jennifer Neville. 2012. Time-evolving relational classification and ensemble methods. In *PAKDD.* Springer, 1–13.

[76] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI.* http://networkrepository.com

[77] Ryan A. Rossi and Nesreen K. Ahmed. 2015. Role Discovery in Networks. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 27, 4 (2015), 1112–1131.

[78] Ryan A. Rossi, Nesreen K. Ahmed, Eunyee Koh, Sungchul Kim, Anup Rao, and Yasin Abbasi-Yadkori. 2020. A structural graph representation learning framework. In *WSDM.*

[79] Ryan A. Rossi, Brian Gallagher, Jennifer Neville, and Keith Henderson. 2013. Modeling Dynamic Behavior in Large Evolving Graphs. In *WSDM.* 667–676.

[80] Ryan A. Rossi, Di Jin, Sungchul Kim, Nesreen K. Ahmed, Danai Koutra, and John Boaz Lee. 2019. From Community to Role-based Graph Embeddings. In *arXiv:1908.08572.*

[81] Umang Sharan and Jennifer Neville. 2008. Temporal-relational classifiers for prediction in evolving domains. In *ICDM.* 540–549.

[82] Olivia Simpson, C Seshadhri, and Andrew McGregor. 2015. Catching the head, tail, and everything in between: A streaming algorithm for the degree distribution. In *ICDM.* 979–984.

[83] Sucheta Soundarajan, Acar Tamersoy, Elias B Khalil, Tina Eliassi-Rad, Duen Horng Chau, Brian Gallagher, and Kevin Roundy. 2016. Generating graph snapshots from streaming edge data. In *WWW.* 109–110.

[84] 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. *TKDD* 11, 4 (2017), 43.

[85] Michael PH Stumpf, Carsten Wiuf, and Robert M May. 2005. Subnets of scale-free networks are not scale-free: sampling properties of networks. *PNAS* 102, 12 (2005), 4221–4224.

[86] Rajmonda Sulo, Tanya Berger-Wolf, and Robert Grossman. 2010. Meaningful selection of temporal resolution for dynamic networks. In *MLG KDD.* 127–136.

[87] Aynaz Taheri and Tanya Berger-Wolf. 2019. Predictive Temporal Embedding of Dynamic Graphs. *ASONAM* (2019).

[88] Aynaz Taheri, Kevin Gimpel, and Tanya Berger-Wolf. 2019. Learning to Represent the Evolution of Dynamic Graphs with Recurrent Models. In *WWW.* 301–307.

[89] Dane Taylor, Sean A Myers, Aaron Clauset, Mason A Porter, and Peter J Mucha. 2017. Eigenvector-based centrality measures for temporal networks. *Multiscale Modeling & Simulation* 15, 1 (2017), 537–574.

[90] Eugenio Valdano, Michele Re Fiorentin, Chiara Poletto, and Vittoria Colizza. 2018. Epidemic threshold in continuous-time evolving networks. *Physical review letters* 120, 6 (2018), 068302.

[91] Jeffrey S Vitter. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)* 11, 1 (1985), 37–57.

[92] Xiaomin Wu, Jianshe Wu, Yafeng Li, and Qian Zhang. 2020. Link prediction of time-evolving network based on node ranking. *Knowledge-Based Systems* (2020), 105740.

[93] Rongjing Xiang, Jennifer Neville, and Monica Rogati. 2010. Modeling relationship strength in online social networks. In *WWW*. 981–990.

[94] Xin Yang and Ju Fan. 2018. Influential User Subscription on Time-Decaying Social Streams. *arXiv:1802.05305* (2018).

[95] Peixiang Zhao, Charu Aggarwal, and Gewen He. 2016. Link prediction in graph streams. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 553–564.

[96] Zhongying Zhao, Chao Li, Xuejian Zhang, Francisco Chiclana, and Enrique Herrera Viedma. 2019. An incremental method to detect communities in dynamic evolving social networks. *Knowledge-Based Systems* 163 (2019), 404–415.

[97] Lorenzo Zino, Alessandro Rizzo, and Maurizio Porfiri. 2016. Continuous-time discrete-distribution theory for activity-driven networks. *Physical review letters* 117, 22 (2016), 228302.

[98] Lorenzo Zino, Alessandro Rizzo, and Maurizio Porfiri. 2017. An analytical framework for the study of epidemic models on activity driven networks. *Journal of Complex Networks* 5, 6 (2017), 924–952.

A PROOFS OF THEOREMS

Proof of Theorem 1. Although (i) is special case of (ii), we prove (i) first then extend to (ii). We establish that

$$\mathbb{E}[\widehat{C}_{e,t}|\widehat{C}_{e,t-1},Q] - C_{e,t} = \widehat{C}_{e,t-1} - C_{e,t-1} \tag{10}$$

for all members Q of set of disjoint events whose union is identically true. Since $\widehat{C}_{e,t_e-1} = C_{e,t_e-1} = 0$ we then conclude that $\mathbb{E}[\widehat{C}_{e,t}] = C_{e,t}$. For $t_e \le s \le s'$, let $A^{(1)}_e(s) = \{e \notin \widehat{K}_{s-1}\}$ (note $A^{(1)}_e(t_e)$ is identically true), let $A^{(2)}_e(s,s')$ denote the event $\{e \in \widehat{K}_s \dots, \widehat{K}_{s'}\}$, *i.e.*, that e is in sample at all times in $[s,s']$. Then $A^{(1)}_e(s)A^{(2)}_e(s,t-1)$ is the event that e was sampled at time $s \le t-1$ and has remained in the reservoir up to and including time $t-1$. For each $t \ge t_e$ the union of the collection of events formed by $\{A^{(1)}_e(s)A^{(2)}_e(s,t-1) : s \in [t_e, t-1]\}$, and $A^{(1)}_e(t)$ is identically true.

(a) *Conditioning on* $A^{(1)}_e(t)$. On $A^{(1)}_e(t)$, $e_t \ne e$ implies $\widehat{C}_{e,t} = \widehat{C}_{e,t-1} = 0 = C_{e,t} - C_{e,t-1}$. On the other hand $e_t = e$ implies $t \in \Omega$ since the arriving edge e is not in the current sample. Further conditioning on $z_{e,t} = \min_{j \in \widehat{K}_{j,t-1}} r_{j,t-1}$ then (6) tells us

$$\mathbb{P}[e \in \widehat{K}_t | A^{(1)}_e(t), z_{e,t}] = \mathbb{P}[u_e < w_{e,t}/z_{e,t}] = p_{e,t} \tag{11}$$

and hence regardless of $z_{e,t}$ we have

$$\mathbb{E}[\widehat{C}_{e,t}|C_{e,t-1}, A^{(1)}_e(t), z_{e,t}] = \widehat{C}_{e,t-1} + C_{e,t} - C_{e,t-1} \tag{12}$$

(b) *Conditioning on* $A^{(1)}_e(s)A^{(2)}_e(s,t-1)$ *any* $s \in [t_e, t-1]$. Under this condition $e \in \widehat{K}_{t-1}$ and if furthermore $e_t \in \widehat{K}_{t-1}$ then $t \notin \Omega$ and the first line in (6) holds. Suppose instead $e_t \notin \widehat{K}_{t-1}$ so that $t \in \Omega$. Let $\mathcal{Z}_e(s,t) = \{z_{e,s'} : s' \in [s,t] \cap \Omega\}$. Observing that

$$\mathbb{P}[A^{(2)}_e(s,t)|A^{(1)}_e(s), \mathcal{Z}_e(s,t)] = \mathbb{P}[\bigcap_{s' \in [s,t] \cap \Omega} \{u_e < \frac{w_{e,s'}}{z_{e,s'}}\}] = p_{e,t}$$

then

$$\mathbb{P}[e \in \widehat{K}_t | A^{(2)}_e(t-1,s)A^{(1)}_e(s), \mathcal{Z}_e(s,t)] = \frac{\mathbb{P}[A^{(2)}_e(s,t)|A^{(1)}_e(s), \mathcal{Z}_e(s,t)]}{\mathbb{P}[A^{(2)}_e(s,t-1)|A^{(1)}_e(s), \mathcal{Z}_e(s,t-1)]} = \frac{p_{e,t}}{p_{e,\omega(t)}} = q_{e,t} \tag{13}$$

and hence

$$\mathbb{E}[\widehat{C}_{e,t}|\widehat{C}_{e,t-1}, A^{(1)}_e(s), \mathcal{Z}_e(s,t)] = \widehat{C}_{e,t-1} \tag{14}$$

independently of the conditions on the LHS of (14). As noted above, $e \in \widehat{K}_{t-1}$ when $A^{(2)}_e(s,t-1)$ is true in which case $C_{e,t} = C_{e,t-1}$ and we recover (10).

(ii) The proof employs a conditioning argument that generalizes a the property of Priority Sampling, namely, that inverse probability estimators of item samples are independent when conditioned on the priorities of other items; see [29]. Our generalization establishes that link-product form estimators of subgraph multiplicities are unbiased. Let $z_{J,t} = \min_{j\in\widehat{K}'_t\setminus J_t} r_{j,t}$. Then $J_t \subset \widehat{K}_t$ iff $u_i \le w_{i,t}/z_{J,t}$ for all $i \in J_t$, in which case $z_{J,t} = z_{j,t}$ for all $j \in J_t$. A sufficient condition for Theorem 1(ii) is that

$$\mathbb{E}[\prod_{j\in J_t}\left(\widehat{C}_{j,t} - C_{j,t}\right)|z_{J,t}, J_{t-1} \subset \widehat{K}_{t-1}] = 0 \tag{15}$$

Since $\widehat{C}_{j,t} = \left(\widehat{C}_{j,t-1} + c_{j,t}\right) I(u_j < w_{j,t}/z_{j,t})/q_{j,t}$, then conditioning on $z_{J,t}$ and $\{J_{t-1} \subset \widehat{K}_{t-1}\}$ fixes $\widehat{C}_{j,t-1}, z_{j,t}$ and $q_{j,t}$ for $j \in J_t$. If we can show furthermore that the $\{u_j : j \in J_t\}$ are independent under the same conditioning, then the conditional expectation (15) will factorize over $j \in J_t$ (the expectation and product may be interchanged) and the result follows from (i).

We establish conditional independence by an inductive argument. Denote $\mathcal{Z}_t = \{z_{J,s} : s \in [t_J, t] \cap \Omega\}$ and assume conditional on $Z_{J,t-1}, J_{t-1} \subset \widehat{K}_t$ that the $u_j : j \in J_{t-1}$ and mutually independent with each uniformly distributed on $(0, p_{j,t-1})$. Note the weights $w_{i,t} : i \in J_t$ determined by $J_{t-1} \subset \widehat{K}_t$ since arrivals are non-random. Further conditioning on $J_t \in \widehat{K}_t$ result in each i being uniform on $(0, \min\{p_{i,t-1}, w_{i,t}/z_{J,t}\}] = (0, p_{i,t}]$ so completing the induction. The property is trivial at the time t_i of first arrival of each edge. The form (iii) then follows inductively on the size of the subgraph J on expanding the product and taking expectations. ∎

Proof of Theorem 2. Here we specify $\widehat{V}_{e,t}$ being commutable from the first t arrivals to mean that it is $\mathcal{F}_t$-measurable, where $\mathcal{F}_t$ is set of random variables $\{u_{e_t} : t \in \Omega\}$ generated up to time t. By the Law of Total Variance

$$\begin{aligned}
\mathrm{Var}(\widehat{C}_{e,t}) &= \mathbb{E}[\mathrm{Var}(\widehat{C}_{e,t})|\mathcal{F}_{t-1}] + \mathrm{Var}(\mathbb{E}[\widehat{C}_{e,t}|\mathcal{F}_{t-1}]) &(16)\\
&= \mathbb{E}[\left(\frac{\widehat{C}_{e,t-1} + c_{e,t}}{q_{e,t}}\right)^2 \mathrm{Var}(I(B_e(z_t))|\mathcal{F}_{t-1}] + \mathrm{Var}(\widehat{C}_{e,t-1} + c_{e,t}) &(17)\\
&= \mathbb{E}[\left(\frac{\widehat{C}_{e,t-1} + c_{e,t}}{q_{e,t}}\right)^2 q_t(1-q_t)] + \mathrm{Var}(\widehat{C}_{e,t-1}) &(18)
\end{aligned}$$

Since $\widetilde{V}_{e,t} := \left(\frac{\widehat{C}_{e,t-1}+c_{e,t}}{q_{e,t}}\right)^2 q_t(1-q_t)$ is $\mathcal{F}_{t-1}$=measurable, then $\widetilde{V}_{e,t} I(B_e(z_t))/q_{e,t}$ is $\mathcal{F}_t$-measurable, and

$$\mathbb{E}[\frac{I(B_e(z_t))}{q_{e,t}}\widetilde{V}_{e,t}] = \mathbb{E}[\mathbb{E}[\frac{I(B_e(z_t))}{q_{e,t}}|\mathcal{F}_{e,t}]\widetilde{V}_{e,t}] = \mathbb{E}[\widetilde{V}_{e,t}] \tag{19}$$

and similarly by assumption on $\widehat{V}_{e,t-1}$,

$$\mathbb{E}[\frac{I(B_e(z_t))}{q_{e,t}}\widehat{V}_{e,t-1}] = \mathbb{E}[\widehat{V}_{e,t-1}] = \mathrm{Var}(\widehat{C}_{e,t-1}) \tag{20}$$

∎

Proof of Theorem 3. (i) follows by linearity of expectation, while (ii) follows by substitution in (8). ∎