

Temporal Network Sampling

Nesreen K. Ahmed
Intel Research Labs
nesreen.k.ahmed@intel.com

Nick Duffield
Texas A&M University
duffieldng@tamu.edu

Ryan A. Rossi
Adobe Research
rossi@adobe.com

ABSTRACT

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 Δt -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.

KEYWORDS

Temporal network sampling, online sampling, unbiased estimation, temporal patterns, temporal link decay, temporal weighted motifs

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) [46, 47]. 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 [27]. Therefore, it is crucial to incorporate both the heterogeneous *structural* and *temporal* information into network representations [48, 53, 59]. By incorporating the temporal information alongside the structural information, we obtain time-varying networks, also called *temporal networks* [29].

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 [36]. Temporal networks have been recently used to model and analyze dynamic and streaming network data, e.g., to analyze and model information propagation [21, 52], epidemics [51], infections [42], user influence [16, 25], among other

applications [48, 59]. 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) [7]. As a result, several algorithms that were studied and designed for static networks that can fit in memory are becoming computationally intractable [4], this is 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* [61]. 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 [15, 28, 64, 68]. 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) [48]. On the other hand, a large window size will likely lose the temporal patterns in the data [22]. 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 [4, 7]. 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 [23], and can often lead to errors and bias the results [48, 68, 71–73].

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 [40]. Sampling algorithms are fundamental in studying and understanding networks [7, 32, 46]. 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 [63]).

While many network sampling techniques are studied in the context of small static networks that can fit entirely in memory [32] (e.g., uniform node sampling [63], random walk sampling [35]), recently there has been a growing interest in sampling techniques for streaming network data in which temporal networks evolve

continuously in time [4–6, 18, 30, 31, 37, 50, 60, 62] (see [7, 43] 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, such as the link strength [70], link persistence [17], burstiness [12], temporal motifs [33], among others [29]. 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 once. 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 describe a general temporal network sampling framework for unbiased estimation of temporal network statistics. We develop online, single-pass sampling algorithms and unbiased estimators for temporal network sampling.
- 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 properties.
- 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.

2.1 Notation & Problem Definition

Edges, Interactions, and Streaming Temporal Networks. Our framework seeks to construct a continuous-time *sampled* network that

Table 1: Summary of notation.

G	temporal network
E	Set of interaction events
K	Set of Unique Edges (links)
$\tilde{G}$	graph induced by unique edges
N, M	number of nodes $N = V $ and edges $M = K $ in the graph $\tilde{G}$
$C_{e,t}$	Multiplicity (weight) of edge e at time t
$\hat{C}_{e,t}$	Estimated multiplicity of edge e at time t
C_t	time-dependent adjacency matrix of link strength at time t
$\hat{K}$	Reservoir of sampled edges
m	Number of sampled edges (Sample Size), $m = \hat{K} $
δ	Link decay rate
C_M	Weighted count of motif pattern M
$\hat{C}_M$	Estimated weighted count of motif pattern 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

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 [70]). 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 ($e = (i, j) \in K$). 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 occurs 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 .

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 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 the edge multiplicity and time. Our framework seeks to construct, maintain, and adapt a continuous-time dynamic sampled network, represented by the matrix $\hat{C}_t$ that serves as unbiased estimator of C_t at any time point t , where the expected number of non-zero entries in $\hat{C}_t$ 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. And, the sampled network serves as a proxy of the full temporal network, thus, 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 $t_0 < \tau^{(1)} < \dots < \tau^{(T)}$, and their first interaction time is $\tau^{(1)} > t_0$. 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 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, $C_{ij,t} = \sum_{s=0}^T \theta(t - \tau^{(s)}) e^{-(t - \tau^{(s)})/\delta}$, 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,

$$C_{e,t} = C_{e,t-1} * e^{-1/\delta} \quad (1)$$

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

$$C_{e,t} = C_{e,t-1} * e^{-1/\delta} + 1 \quad (2)$$

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.

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*, see 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 frequently are more important than those occurring in the distant past.

DEFINITION 1 (TEMPORALLY WEIGHTED NETWORK 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 [8, 9, 13, 44], as well as improve the accuracy of machine learning models [10, 56]. While prior work focused on aggregating interactions in time windows and analyze the aggregated graph snapshots [53, 59, 66], others have focused on aggregating motifs in Δt time bins, and defined motif duration [38]. 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 [5, 19, 69], 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

Algorithm 1 Online Temporal Network Sampling (Online-TNS)**INPUT:** Sample size m , Motif pattern $\mathcal{M}$, initial weight ϕ **OUTPUT:** Estimated network C_t , Estimated motif count $\widehat{C}_M$

```

1 procedure ONLINE-TNS( $m$ )
2    $\widehat{K} = \emptyset$ ;  $z^* = 0$ ;  $\widehat{C}_M = 0$       ▶ Initialize edge sample & threshold
3   while (new interaction  $e_t = (i, j, t)$ ) do
4      $e = (i, j)$ 
5     SUBGRAPH-ESTIMATION( $e$ )      ▶ Update Estimated Motif Count
6     if ( $e \in \widehat{K}$ ) then      ▶ Edge exists in  $\widehat{K}$ 
7       UPDATE-EDGE-STRENGTH( $e$ )      ▶ Update Edge Strength
8        $\widehat{C}(e) = \widehat{C}(e) + 1$       ▶ Increment edge multiplicity
9        $w(e) = w(e) + 1$       ▶ Adapt importance weight
10       $r(e) = w(e)/u(e)$       ▶ Adapt edge rank
11       $\tau(e) = t$       ▶ Last Interaction Time
12    else
13      //Initialize parameters for new edge
14       $p(e) = 1$ ;  $\widehat{C}(e) = 1$ ;  $V(e) = 0$ ;
15       $u(e) = \text{Uniform}(0, 1]$       ▶ Initialize Uniform r.v.
16       $w(e) = \phi$       ▶ Initialize edge weight
17       $r(e) = w(e)/u(e)$       ▶ Compute edge rank
18       $\tau(e) = t$ 
19       $\widehat{K} = \widehat{K} \cup \{e\}$       ▶ Provisionally include  $e$  in sample
20      if ( $|\widehat{K}| > m$ ) then
21         $e^* = \arg \min_{e' \in \widehat{K}} r(e')$       ▶ Find edge with min rank
22         $z^* = \max\{z^*, r(e^*)\}$       ▶ Update threshold
23        remove  $e^*$  from  $\widehat{K}$ 
24        delete  $\{w(e^*), u(e^*), p(e^*), \widehat{C}(e^*), V(e^*)\}$ 

```

```

25 procedure UPDATE-EDGE-STRENGTH( $\bar{e}$ )
26   // Function to estimate edge strength (No-decay)
27   if ( $z^* > 0$ ) then
28      $q = \min\{1, w(\bar{e})/(z^* p(\bar{e}))\}$ 
29      $\widehat{C}(\bar{e}) = \widehat{C}(\bar{e})/q$       ▶ Estimate edge strength
30      $V(\bar{e}) = V(\bar{e})/q + (1 - q) * \widehat{C}(\bar{e})^2$ 
31      $p(\bar{e}) = p(\bar{e}) * q$ 

```

```

32 procedure UPDATE-EDGE-DECAY( $\bar{e}$ )
33   // Function to estimate edge strength (Link-decay)
34   if ( $z^* > 0$ ) then
35      $q = \min\{1, w(\bar{e})/(z^* p(\bar{e}))\}$ 
36      $\widehat{C}(\bar{e}) = e^{-\delta(t-\tau(\bar{e}))} * \widehat{C}(\bar{e})/q$  ▶ Estimate link-decayed strength
37      $V(\bar{e}) = V(\bar{e})/q + (1 - q) * \widehat{C}(\bar{e})^2$ 
38      $p(\bar{e}) = p(\bar{e}) * q$ 

```

```

39 procedure SUBGRAPH-ESTIMATION( $\bar{e}$ )
40   //Set of Subgraphs isomorphic to  $\mathcal{M}$  and completed by  $\bar{e}$ 
41    $H = \{h \subset \widehat{K} \cup \{\bar{e}\} : h \ni \bar{e}, h \cong \mathcal{M}\}$ 
42   for  $h \in H$  do
43     for  $j \in h \setminus \bar{e}$  do
44       UPDATE-EDGE-STRENGTH( $j$ ) ▶ Update other edge estimates
45   //Increment estimated count of motif  $\mathcal{M}$ 
46    $\widehat{C}_M = \widehat{C}_M + \prod_{j \in h \setminus \{\bar{e}\}} \widehat{C}(j)$ 

```

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 $u(e) = w(e)/u(e)$ (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). 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 [29, 45].

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 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

Here, we 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 $\tilde{G} = (V, K)$ as the graph induced by K . Thus the stream of interaction can also be regarded as a stream $\{e_t : t \in [|E|]\}$ of non-unique edges from K . Let K_t denote the unique interactions in $\{e_s : s \leq t\}$ and $\tilde{G}_t = (V_t, K_t)$ the induced graph. 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 sets $\tilde{J} = \{e_{i_1}, \dots, e_{i_{|J|}}\}$ with $i_j \leq t$, such that $\tilde{J}$ is a permutation of J . Hence $C_{J,t} = \prod_{e \in J} C_{e,t}$. Given a class $\mathcal{H}$ of subgraphs of $\tilde{G}$, 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 $C_{e,t} = \sum_{s \leq t} c_{e,s}$. $\tilde{K}_t$ will denote the sample set of unique edges after arrival t has been processed. We maintain an estimator $\hat{C}_{e,t}$ of $C_{e,t}$ for each $e \in \tilde{K}_t$. Implicitly $\hat{C}_{e,t} = 0$ if $e \notin \tilde{K}_t$.

Sampling proceeds as follows. If the arriving edge $e_t \neq \tilde{K}_{t-1}$ then e_t is provisionally included in the sample, forming $\tilde{K}'_t = \tilde{K}_{t-1} \cup \{e_t\}$, and we set $\hat{C}_{e_t,t} = c_{e_t,t} = 1$. 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 $i \in \tilde{K}'_t$ as described below, from which the edge *time-dependent* priority at time t is $r_{i,t} = w_{i,t}/u_i$. If $|\tilde{K}'_t| > m$, the edge $d_t = \arg \min_{i \in \tilde{K}'_t} r_{i,t}$ of minimum priority is discarded, and the estimates $\hat{C}_{i,t}$ of the surviving edges $i \in \tilde{K}_t = \tilde{K}'_t \setminus \{d_t\}$ undergo inverse probability normalization through division by the conditional probability $q_{i,t}$ of retention in $\tilde{K}_t$; see (5). If the arriving edge is already in the reservoir $e_t \in \tilde{K}_{t-1}$ then we increment $\hat{C}_{e_t,t} = \hat{C}_{e_t,t-1} + 1$ and no sampling is needed, i.e., $\tilde{K}_t = \tilde{K}_{t-1}$.

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

$$r_{i,t} > z_{i,t} := \min_{j \in \tilde{K}'_t \setminus \{i\}} r_{j,t} \quad (3)$$

Hence by recurrence, $i \in \tilde{K}_t$ requires only if $u_i < \min_s \{w_{i,s}/z_{i,s}\}$ where s takes values over $\{\alpha_i(t), \dots, \omega(\omega(t)), \omega(t), t\}$ where $\alpha_i(t)$ is the most recent time at which edge i was sampled into the reservoir.

This motivates the definition below the $p_{e,t}$, the edge selection probability conditional on the thresholds z_t , and $q_{e,t}$ 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

$$p_{e,t} = \begin{cases} \min\{1, w_{e,t}/z_t\} & \text{if } t = \min \Omega \\ \min\{p_{e,\omega(t)}, w_{e,t}/z_t\} & \text{otherwise} \end{cases} \quad (4)$$

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

$$\hat{C}_{e,t} = \left(\hat{C}_{e,t-1} + c_{e,t} \right) \frac{I(u_i < w_{i,t}/z_{i,t})}{q_{e,t}} \quad (5)$$

where

$$q_{e,t} = \begin{cases} 1 & \text{if } t \notin \Omega \\ p_{e,t} & \text{if } t \in \Omega \text{ and } e = e_t \\ p_{e,t}/p_{e,\omega(t)} & \text{otherwise} \end{cases} \quad (6)$$

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 8.

THEOREM 1.

- (i) $\mathbb{E}[\hat{C}_{e,t}] = C_{e,t}$ for all $t \geq 0$.
- (ii) For each $J \subset V$ and $t \geq t_J$ then $\prod_{e \in J_t} (\hat{C}_{e,t} - C_{e,t}) : t \geq t_J$ has expectation 0.
- (iii) $\mathbb{E}[\prod_{e \in J} \hat{C}_{e,t}] = \prod_{e \in J} C_{e,t}$ for $t \geq \max_{j \in J} t_j$.

Estimating Subgraph Multiplicities Theorem 1 tell us for a subgraph $J \subset K_t$, that $\prod_{e \in J} \hat{C}_{e,t}$ 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 $\tilde{G}_t$ 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 $C_{\mathcal{M},t} = \sum_{s \leq t} C_{\mathcal{M},s}^{(0)}$ where $C_{\mathcal{M},s}^{(0)} = \sum_{h \in H_s^{(0)}} C_{h \setminus \{e_s\},s}$ where $H_s^{(0)} = \{h \in K_s : h \ni e_s : h \cong \mathcal{M}\}$, 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 $\hat{C}_{\mathcal{M},t} = \sum_{s \leq t} \hat{C}_{\mathcal{M},s-1}^{(0)}$ is an unbiased estimator of $C_{\mathcal{M},t}$ where $\hat{C}_{\mathcal{M},s-1}^{(0)} = \sum_{h \in H_s^{(0)}} \hat{C}_{h \setminus \{e_s\},s-1}$. Thus for each arrival e_t we estimate $C_{J,t}$ just prior to sampling of e_t by $\hat{C}_{J,t} = \prod_{j \in J \setminus \{e_t\}} \hat{C}_{j,t-1}$. For each $J \subset H_t$ we increment a running total of $\hat{M}_t$ by this amount; see line 46 in Algorithm 1.

Edge Multiplicity Estimation Variance

THEOREM 2. Suppose $\hat{V}_{e,t-1}$ is an unbiased estimator of $\text{Var}(\hat{C}_{e,t-1})$ that can be computed from information on the first $t-1$ arrivals. Then

$$\hat{V}_{e,t} = \hat{C}_{e,t}^2(1 - q_{e,t}) + I(B_e(z_t))\hat{V}_{e,t-1}/q_{e,t} \quad (7)$$

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

The computability condition expresses the property that $\widehat{V}_{e,t}$ can be computed immediately when $e \in \widehat{K}_t$. The relation (7) defines an iteration for estimating the variance $\text{Var}(\widehat{C}_t)$ 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 the 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

$$\widehat{C}_{k,t}^\delta = \left(\widehat{C}_{k,t-1}^\delta e^{-1/\delta} + c_{k,t} \right) \frac{I(u_k < w_{k,t}/z_{k,t})}{q_{k,t}} \quad (8)$$

which exponentially discounts the contribution from the previous time slot.

- THEOREM 3.** (i) $\widehat{C}_{k,t}^\delta$ is an unbiased estimator of $C_{k,t}^\delta$
(ii) Replacing $\widehat{C}_{k,t}$ with $\widehat{C}_{k,t}^\delta$ in the iteration yields an unbiased estimator $V_{k,t}^\delta$ of $\text{Var}(\widehat{C}_{k,t}^\delta)$

5 EXPERIMENTS

In this section, we perform extensive experiments on a wide variety of temporal networks. We systematically investigate the effectiveness of the framework for estimating *temporal network statistics* (Section 5.1), *temporal link strengths* (Section 5.2), and *temporally weighted motifs* (Section 5.3) using the decay model. We repeat the experiment five different times with sample fractions $p = \{0.10, 0.20, 0.40, 0.50\}$. The temporal network data used in our experiments is shown in Table 2.

5.1 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 [12] and temporal link persistence [17]. For a survey of other important temporal network statistics that are applicable for estimation using the framework, see [29].

Burstiness. Burstiness B is widely used to characterize the link activity in temporal networks [29]. 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 3, 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 3, we observe the relative errors are small and the estimates are shown to converge as the sampling fraction p increases. In Figure 1, we show the exact and estimated distribution of inter-contact times for sampling fractions $p = 0.1$ (top row) and

Table 2: Temporal network data [54]. Note $|K|$ is the number of static edges (not including multiplicities); $|E|$ = number of temporal edges; and $C_{\max}$ = maximum edge weight.

TEMPORAL NETWORK	$ V $	$ K $	$ E $	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-EU-core	986	16k	332k	803.9	5.0k
email-dnc	1.9k	4.4k	39k	982.3	634
ia-radoslaw-email	167	3.2k	83k	271.2	2.9k
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
SMS-A	44k	52k	548k	338.3	10k
ia-contacts-hyper09	113	2.2k	21k	2.5	1.3k
ia-contact	274	2.1k	28k	4.0	168
ia-primary-school	242	8.3k	126k	1.4	764
ia-hospital-ward	75	1.1k	32k	4.0	1.1k
ia-workplace-cont	92	755	9.8k	11.4	737
ia-highschool-cont	327	5.8k	189k	4.0	2.9k
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
copres-LyonSchool	242	27k	6.6M	1.4	2.5k
copres-Thiers13	328	43k	19M	4.4	5.0k
copres-LH10	73	1.4k	150k	3.0	4.4k

$p = 0.2$ (bottom row). We observe that the estimated distribution from the sample accurately captures the exact distribution.

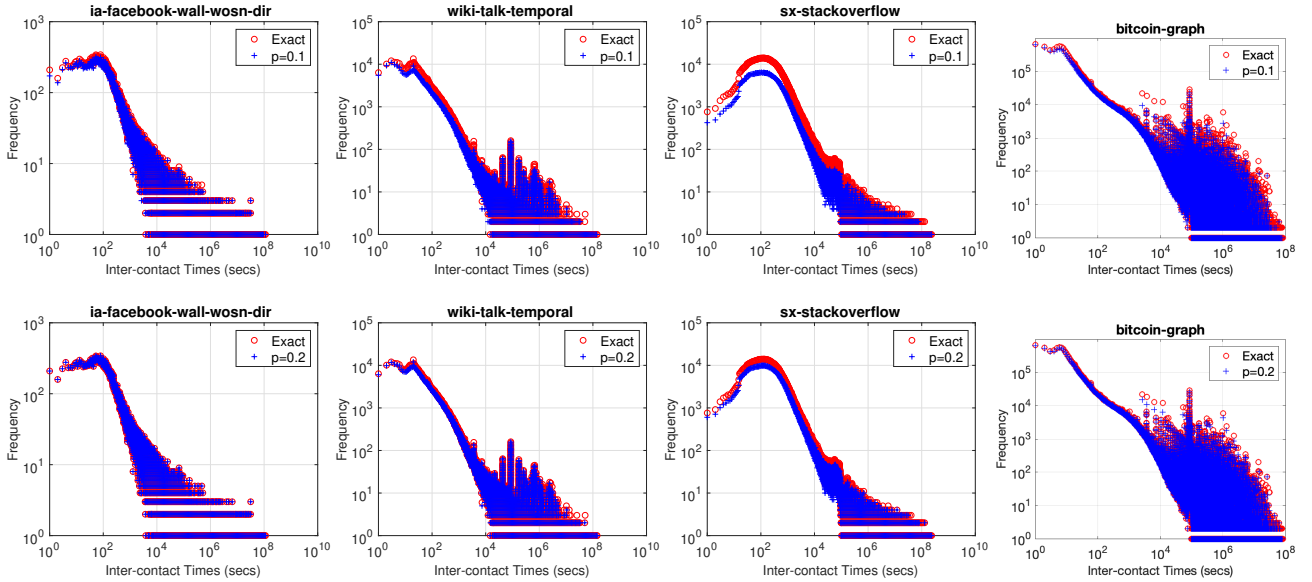
Temporal Link Persistence. The persistence of an edge measures the lifetime of relationships, and is computed as the elapsed time between the first interaction and the last interaction of the same edge [29]. Let L denote the average link persistence (or lifetime) computed over all edges in the full (sampled) 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 4. In Figure 2, 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).

5.2 Estimation of Temporal Link Strength

Link strength is one of the most fundamental properties of temporal networks [70]. Therefore, estimating it in an online fashion is clearly important. Results using Alg. 1 and unbiased estimators (see Section 4) for temporal link strength estimation are provided in Figure 3 (top row). We show the distribution of the top- k edges ($k = 10$ million) and compare the exact link strength vs 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 Figure 3 (top row), we observe the exact and estimated link

Table 3: 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 $|\hat{B}-B|/B$ of the estimates is also shown.

TEMPORAL NETWORK	SAMPLING FRACTION					Exact
	0.1	0.2	0.3	0.4	0.5	
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	

**Figure 1: 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).**Table 4: Estimation results for temporal persistence.** For each temporal network, we report relative error $|\hat{L}-L|/L$ of the estimates using different sampling fractions.

TEMPORAL NETWORK	SAMPLING FRACTION				
	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

strengths for the top- k edges to be nearly indistinguishable from one another. We also compare to uniform edge sampling (Unif-ES), in which links are sampled from the stream uniformly with the same probability, results are shown in Figure 3 (bottom row). We use our proposed unbiased estimators in Section 4 to estimate the

link strength even in the case of uniform edge sampling. While the estimated link strengths from Online-TNS are nearly identical to the exact link strengths, estimated distributions from uniform edge sampling are significantly worse. Due to space constraints, results

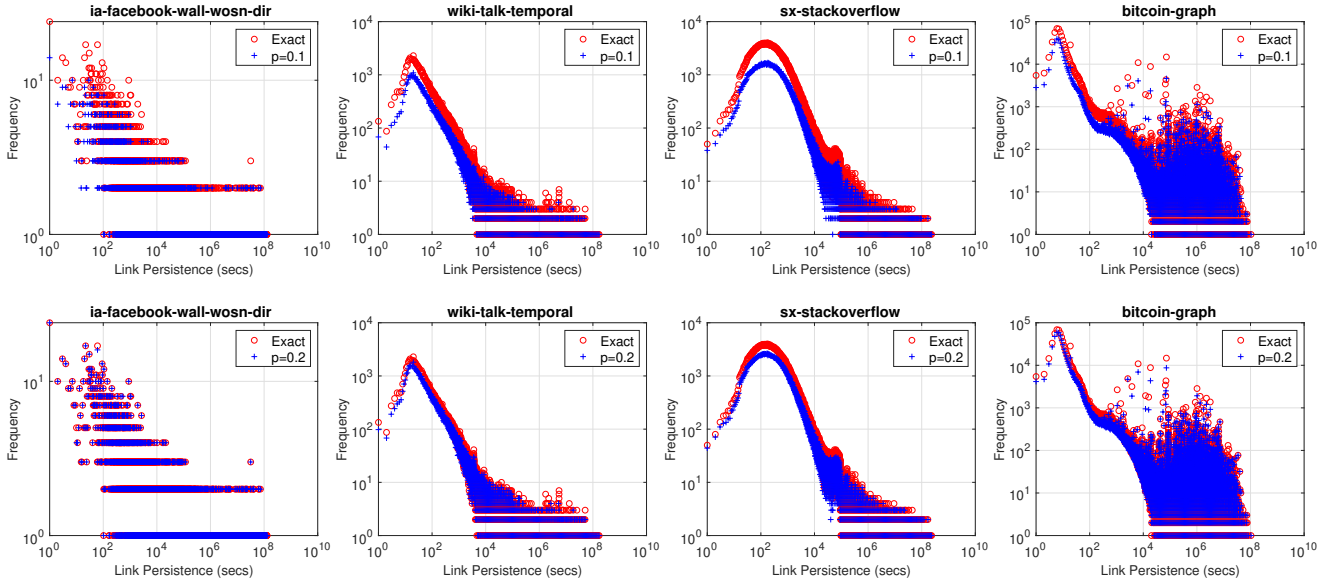


Figure 2: Estimation results for the distribution of link persistence scores compared to the exact distribution. Results are shown for $p = 0.1$ (top) and $p = 0.2$ (bottom).

for other graphs have been removed. However, similar results were observed.

In Table 5, we show the relative spectral norm for online-TNS and uniform edge sampling (Unif-ES). The relative spectral norm is defined as $\|C - \hat{C}\|_2 / \|C\|_2$, where C is the exact time-dependent adjacency matrix of the input graph, whose entries represent the link strength, $\hat{C}$ is the average estimated time-dependent adjacency matrix (estimated from the sample), and $\|C\|_2$ is the spectral norm of C . The spectral norm $\|C - \hat{C}\|_2$ is widely used for matrix approximations [1]. $\|C - \hat{C}\|_2$ measures the strongest linear trend of C not captured by the estimate $\hat{C}$. The results show Online-TNS significantly outperforms uniform edge sampling, and captures the linear trend and structure of the data better than uniform sampling.

Table 5: Relative spectral norm results for $p = 0.1$.

TEMPORAL NETWORK	Online-TNS (Alg. 1)	Unif-ES
ia-facebook-wall	0.0090	0.3976
sx-stackoverflow	0.0992	0.4360
comm-linux-reply	0.0041	0.1978
ia-enron-email	0.0098	0.4080
SFHH-conf-sensor	0.0090	0.2769
ia-contacts-hyper	0.0034	0.0529

5.3 Estimation of Temporally Weighted Motifs

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 temporal motifs are weighted such that motifs that occur more recent 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 6, we show results for 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. 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 0.01 as shown in Table 6 (last column). Furthermore, many of the estimates of the motif counts (with or without decay) have a relative error of 0. This is due to the number of unique links being significantly less than the number of temporal interactions. Nevertheless, this demonstrates that our efficient temporal sampling framework is able to leverage accurate estimators for even the smallest sample sizes.

6 RELATED WORK

Sampling algorithms are fundamental in studying and understanding networks [7, 32, 46]. A sampled network is called *representative*, if the characteristics of interest in the full network can be accurately estimated from the sample [7]. Network sampling has been widely studied in the context of small static networks that can fit entirely in memory [32]. For instance, there is uniform node sampling [63], random walk sampling [35], edge sampling [7], among others [11, 38]. More recently, there has been a growing interest in sampling techniques for streaming network data in which temporal networks evolve continuously in time [3–6, 18, 30, 31, 37, 50, 60, 62]. For seminal surveys on the topic, see [7, 43].

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 [5]. However, it is

Temporal Network Sampling

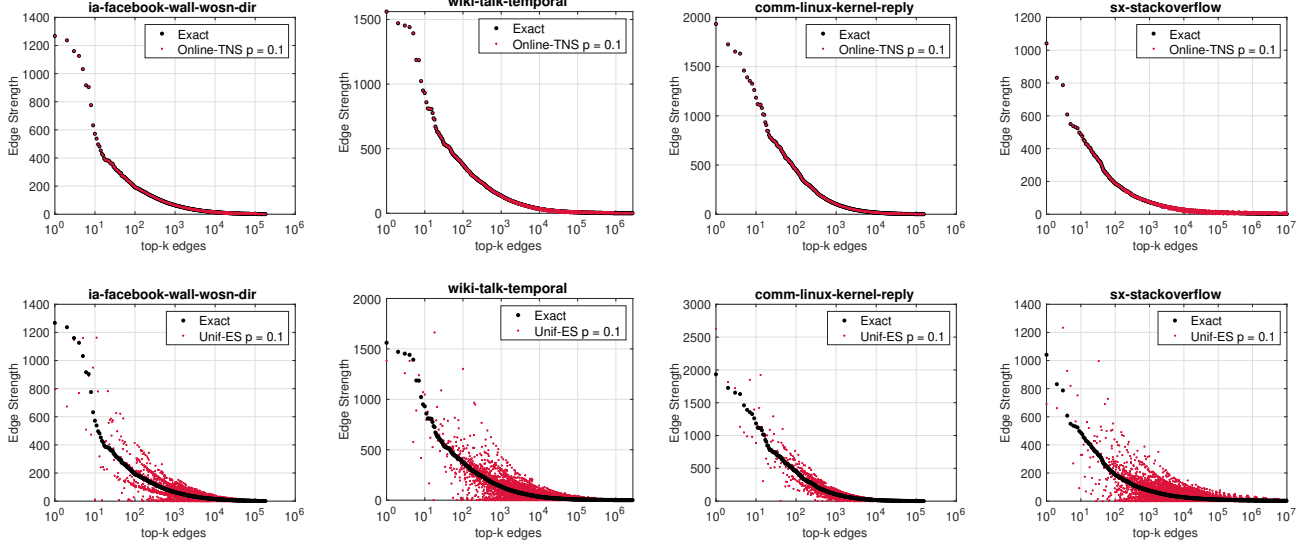


Figure 3: Temporal link strength estimated distribution vs exact distribution for top-k links . Results are shown for sampling fraction $p = 0.1$. (Top) Results for Online-TNS Algorithm 1. (Bottom) Results for uniform edge sampling (Unif-ES).

Table 6: Results for temporally weighted motif count estimation. Results for triangle counts are reported using $p = 0.1$.

TEMPORAL NETWORK	WITHOUT DECAY			WITH DECAY		
	exact	est.	rel. err.	exact	est.	rel. err.
sx-stackoverflow	15B	15B	0	168M	168M	0.0008
ia-facebook-wall	435M	435M	0.0004	9.9M	9.9M	0.0004
wiki-talk	12B	12B	0.0003	394M	394M	0.0001
CollegeMsg	6.2M	6.1M	0.0148	2.0M	2.0M	0.0003
ia-retweet-pol	380k	371k	0.0236	147k	147k	0.0001
ia-prosper-loans	1.4M	1.4M	0.0056	232k	230k	0.0067
comm-linux-reply	148B	148B	0	242M	242M	0.0002
email-EU-core	21B	21B	0	285M	285M	0
email-dnc	483M	483M	0	251M	251M	0
ia-radoslaw-email	4.1B	4.1B	0	134M	134M	0
ia-enron-email	14B	14B	0	329M	329M	0.0003
ia-contacts-dublin	382M	382M	0.0001	381M	381M	0
fb-forum	3.3M	3.3M	0.0036	763k	758k	0.0067
SMS-A	217M	217M	0	4.0M	4.0M	0
ia-contacts-hyper09	93M	93M	0	88M	88M	0
ia-contact	306M	306M	0	286M	286M	0
ia-primary-school	1.7B	1.7B	0	1.6B	1.6B	0
ia-hospital-ward	1.7B	1.7B	0	1.6B	1.6B	0
ia-workplace-cont	23M	23M	0	17M	17M	0
ia-highschool-cont	10B	10B	0	9.0B	9.0B	0
SFHH-conf-sensor	622M	622M	0	604M	604M	0.0002
sx-superuser	83M	82M	0.0072	2.1M	2.1M	0.0017
sx-askubuntu	71M	70M	0.0035	2.7M	2.7M	0.0069
sx-mathoverflow	269M	269M	0.0008	2.8M	2.8M	0.0005
copres-LyonSchool	65T	65T	0	63T	63T	0
copres-Thiers13	1.5P	1.5P	0	1.3P	1.3P	0
copres-LH10	200B	200B	0	190B	190B	0

unclear how representative these samples are for temporal network statistics such as the link strength [70], link persistence [17], burstiness [12], temporal motifs [33], among others [29]. Despite the fundamental importance of this question, it has not been addressed in the context of streaming and online methods.

The temporally decaying model of temporal networks is useful for many important predictive modeling and forecasting tasks including classification [53, 59], link prediction [20], influence modeling [25], regression [24], and anomaly detection [2, 57]. 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 [14, 20], influence modeling [25], visitor stitching [56], among many others.

There has also been a lot of research on deriving new and important temporal network statistics and properties that appropriately characterize the temporal network [29]. Other recent work has focused on extending node ranking and importance measures to dynamic networks such as Katz [26] and eigenvector centrality [67]. 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 [58] including both community-based (proximity) and role-based structural node embeddings [10, 55]. 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 [34, 39, 48] as well as discrete-time dynamic networks where the actual edge stream is approximated with a sequence of static snapshot graphs [41, 49, 57, 65, 66]. 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.

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 experimental results demonstrated the effectiveness of the proposed approach for unbiased estimation of temporal network statistics.

8 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

$$\mathbb{E}[\widehat{C}_{e,t} | \widehat{C}_{e,t-1}, Q] - C_{e,t} = \widehat{C}_{e,t-1} - C_{e,t-1} \quad (9)$$

for all members Q of a covering partition (i.e., a set of disjoint events whose union is identically true). Since $\widehat{C}_{e,t-1} = C_{e,t-1} = 0$ we then conclude that $\mathbb{E}[\widehat{C}_{e,t}] = C_{e,t}$. For $t_e \leq e \leq e'$ let $A_e^{(1)}(s) = \{e \notin \widehat{K}_{t-1}\}$ (note $A_e^{(1)}(t_e)$ is identically true), let $A_e^{(2)}(s, s')$ 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 for each $t \geq t_e$ the collection of events formed by $\{A_e^{(1)}(s)A_e^{(2)}(s, t-1) : s \in [t_e, t-1]\}$, and $A_e^{(1)}(t)$ is a covering partition.

(a) *Conditioning on $A_e^{(1)}(t)$.* On $A_e^{(1)}(t)$, $e_t \neq e$ 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 $t \in \Omega$. Further conditioning on $z_{e,t} = \min_{j \in \widehat{K}_{t-1}} r_{j,t-1}$ then (5) tells us

$$\mathbb{P}[e \in \widehat{K}_t | A_e^{(1)}(t), z_{e,t}] = \mathbb{P}[u_e < w_{e,t}/z_{e,t}] = p_{e,t} \quad (10)$$

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

$$\mathbb{E}[\widehat{C}_{e,t} | C_{e,t-1}, A_e^{(1)}(t), z_{e,t}] = \widehat{C}_{e,t-1} + C_{e,t} - C_{e,t-1} \quad (11)$$

(b) *Conditioning on $A_e^{(1)}(s)A_e^{(2)}(s, t-1)$ 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 (5) holds. Suppose instead $e_t \notin \widehat{K}_{t-1}$ so that $t \in \Omega$. Let $\mathcal{Z}_e(t, s) = \{z_{e,s'} : s' \in [s, t] \cap \Omega\}$. Observing that

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

then

$$\begin{aligned} \mathbb{P}[e \in \widehat{K}_t | A_e^{(2)}(t-1, s)A_e^{(1)}(s), \mathcal{Z}_e(t, s)] \\ = \frac{\mathbb{P}[B_e(t, s) | A_e^{(1)}(s), \mathcal{Z}_e(t, s)]}{\mathbb{P}[B_e(t-1, s) | A_e^{(1)}(s), \mathcal{Z}_e(t-1, s)]} = \frac{p_{e,t}}{p_{e,\omega(t)}} = q_{e,t} \end{aligned} \quad (12)$$

and hence

$$\mathbb{E}[\widehat{C}_{e,t} | \widehat{C}_{e,t-1}, J_t A_e^{(1)}(s), \mathcal{Z}_e(t, s)] = \widehat{C}_{e,t-1} \quad (13)$$

independently of the conditions on the LHS of (13). As noted above, $e \in \widehat{K}_{t-1}$ on $B_e(t-1, s)$ hence $C_{e,t} = C_{e,t-1}$ and we recover (9). Since we now established (9) over all members Q of a covering partition, the proof of (i) is complete.

(ii) For $i \in \widehat{K}_t'$ let $z_{J,t} = \min_{j \in \widehat{K}_t' \setminus J_t} r_{j,t}$. Then $J_t \in \widehat{K}_t$ iff $u_i \leq w_{i,t}/z_{J,t}$ for all $i \in J_t$. Then a sufficient condition for (ii) is that $\mathbb{E}[\prod_{j \in J_t} (\widehat{C}_{j,t} - C_{j,t}) | Z_{J,t}, J_{t-1} \subset \widehat{K}_{t-1}] = 0$. A further sufficient condition for the latter relation is that distributions of the $\{u_j : j \in J_t\}$ and independent under the conditioning $Z_{J,t}, J_{t-1} \subset \widehat{K}_{t-1}$, for then the conditional expectation factorizes and the result follows from (i).

A specific form of conditional independence follows by induction. Let $\mathcal{Z}_{J,t} = \{z_{J,s} : s \in [t_J, t]\}$. 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_1 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

$$\text{Var}(\widehat{C}_{e,t}) = \mathbb{E}[\text{Var}(\widehat{C}_{e,t}) | \mathcal{F}_{t-1}] + \text{Var}(\mathbb{E}[\widehat{C}_{e,t} | \mathcal{F}_{t-1}]) \quad (14)$$

$$\begin{aligned} &= \mathbb{E}\left[\left(\frac{\widehat{C}_{e,t-1} + c_{e,t}}{q_{e,t}}\right)^2 \text{Var}(I(B_e(z_t)) | \mathcal{F}_{t-1})\right. \\ &\quad \left. + \text{Var}(\widehat{C}_{e,t-1} + c_{e,t})\right] \end{aligned} \quad (15)$$

$$= \mathbb{E}\left[\left(\frac{\widehat{C}_{e,t-1} + c_{e,t}}{q_{e,t}}\right)^2 q_t(1 - q_t)\right] + \text{Var}(\widehat{C}_{e,t-1}) \quad (16)$$

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

$$\mathbb{E}\left[\frac{I(B_e(z_t))}{q_{e,t}} \widehat{V}_{e,t}\right] = \mathbb{E}\left[\mathbb{E}\left[\frac{I(B_e(z_t))}{q_{e,t}} | \mathcal{F}_{e,t}\right] \widehat{V}_{e,t}\right] = \mathbb{E}[\widehat{V}_{e,t}] \quad (17)$$

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

$$\mathbb{E}\left[\frac{I(B_e(z_t))}{q_{e,t}} \widehat{V}_{e,t-1}\right] = \mathbb{E}[\widehat{V}_{e,t-1}] = \text{Var}(\widehat{C}_{e,t-1}) \quad (18)$$

PROOF OF THEOREM 3. (i) follows by linearity of expectation, while (ii) follows by substitution in (7). ■

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under Grant No. IIS-1848596.

REFERENCES

- [1] Dimitris Achlioptas, Zohar S Karnin, and Edo Liberty. 2013. Near-optimal entry-wise sampling for data matrices. In *NeurIPS*. 1565–1573.
- [2] Charu C Aggarwal, Yuchen Zhao, and S Yu Philip. 2011. Outlier detection in graph streams. In *ICDE*. IEEE, 399–409.
- [3] Nesreen K Ahmed, Fredrick Berchmans, Jennifer Neville, and Ramana Kompella. 2010. Time-based sampling of social network activity graphs. In *MLG*. 1–9.
- [4] 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.
- [5] 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.
- [6] Nesreen K. Ahmed, Nick Duffield, and Liangzhen Xia. 2018. Sampling for Approximate Bipartite Network Projection. In *IJCAI*. 3286–3292.
- [7] Nesreen K Ahmed, Jennifer Neville, and Ramana Kompella. 2014. Network sampling: From static to streaming graphs. *TKDD* 8, 2 (2014), 7.
- [8] Nesreen K Ahmed, Jennifer Neville, Ryan A Rossi, and Nick Duffield. 2015. Efficient graphlet counting for large networks. In *ICDM*. 1–10.
- [9] 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.
- [10] Nesreen K Ahmed, Ryan Rossi, John Boaz Lee, Theodore L Willke, Rong Zhou, Xiangan Kong, and Hoda Eldardiry. 2018. Learning role-based graph embeddings. *arXiv:1802.02896* (2018).
- [11] Nesreen K Ahmed, Theodore L Willke, and Ryan A Rossi. 2016. Estimation of local subgraph counts. In *IEEE Big Data*. IEEE, 586–595.
- [12] Albert-László Barabási. 2005. The origin of bursts and heavy tails in human dynamics. *Nature* 435, 7039 (2005), 207.
- [13] Austin R Benson, David F Gleich, and Jure Leskovec. 2016. Higher-order organization of complex networks. *Science* 353, 6295 (2016), 163–166.
- [14] Homanga Bharadhwaj and Shruti Joshi. 2018. Explanations for Temporal Recommendations. *KI* 32, 4 (01 Nov 2018), 267–272.
- [15] Rajmonda Sulo Caceres, Tanya Berger-Wolf, and Robert Grossman. 2011. Temporal scale of processes in dynamic networks. In *ICDM Workshops*. IEEE, 925–932.
- [16] Wei Chen, Laks VS Lakshmanan, and Carlos Castillo. 2013. Information and influence propagation in social networks. *Syn. Lec. on Data Man.* 5, 4 (2013).
- [17] Aaron Clauset and Nathan Eagle. 2012. Persistence and periodicity in a dynamic proximity network. *arXiv:1211.7343* (2012).
- [18] Graham Cormode and Hossein Jowhari. 2014. A second look at counting triangles in graph streams. *Theoretical Computer Science* 552 (2014), 44–51.
- [19] 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.
- [20] Daniel M Dunlavy, Tamara G Kolda, and Evrim Acar. 2011. Temporal link prediction using matrix and tensor factorizations. *TKDD* 5, 2 (2011), 10.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. 2010. Learning influence probabilities in social networks. In *WSDM*. ACM, 241–250.
- [26] Peter Grindrod, Mark C Parsons, Desmond J Higham, and Ernesto Estrada. 2011. Communicability across evolving networks. *Phys. Rev. E* 83, 4 (2011), 046120.
- [27] Petter Holme. 2015. Modern temporal network theory: a colloquium. *The European Physical Journal B* 88, 9 (2015), 234.
- [28] Petter Holme and Luis EC Rocha. 2019. Impact of misinformation in temporal network epidemiology. *Network Science* 7, 1 (2019), 52–69.
- [29] Petter Holme and Jari Saramäki. 2012. Temporal networks. *Physics reports* 519, 3 (2012), 97–125.
- [30] 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.
- [31] Madhav Jha, Comandur Seshadhri, and Ali Pinar. 2013. A space efficient streaming algorithm for triangle counting using the birthday paradox. In *KDD*. 589–597.
- [32] Eric D Kolaczyk and Gábor Csárdi. 2014. *Statistical analysis of network data with R*. Vol. 65. Springer.
- [33] 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.
- [34] John Boaz Lee, Giang Nguyen, Ryan A Rossi, Nesreen K Ahmed, Eunye Koh, and Sungchul Kim. 2019. Temporal Network Representation Learning. *arXiv:1904.06449* (2019).
- [35] Jure Leskovec and Christos Faloutsos. 2006. Sampling from large graphs. In *KDD*. 631–636.
- [36] 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.
- [37] Yongsub Lim and U Kang. 2015. Mascot: Memory-efficient and accurate sampling for counting local triangles in graph streams. In *KDD*. ACM, 685–694.
- [38] Paul Liu, Austin R Benson, and Moses Charikar. 2019. Sampling methods for counting temporal motifs. In *WSDM*. 294–302.
- [39] 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.
- [40] Sharon L Lohr. 2019. *Sampling: Design and Analysis*. Chapman and Hall/CRC.
- [41] Sedigheh Mahdavi, Shima Khoshraftar, and Aijun An. 2019. Dynamic Joint Variational Graph Autoencoders. *arXiv:1910.01963* (2019).
- [42] Naoki Masuda and Petter Holme. 2013. Predicting and controlling infectious disease epidemics using temporal networks. *F1000prime reports* 5 (2013).
- [43] Andrew McGregor. 2014. Graph stream algorithms: a survey. *ACM SIGMOD Record* 43, 1 (2014), 9–20.
- [44] 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.
- [45] Giovanna Miritello, Esteban Moro, and Rubén Lara. 2011. Dynamical strength of social ties in information spreading. *Physical Review E* 83, 4 (2011), 045102.
- [46] Mark Newman. 2018. *Networks*. Oxford university press.
- [47] Mark Ed Newman, Albert-László Ed Barabási, and Duncan J Watts. 2006. *The structure and dynamics of networks*. Princeton university press.
- [48] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunye Koh, and Sungchul Kim. 2018. Continuous-time dynamic network embeddings. In *WWW*. 969–976.
- [49] Hogun Park and Jennifer Neville. 2019. Exploiting Interaction Links for Node Classification with Deep Graph Neural Networks. In *IJCAI*. 3223–3230.
- [50] A Pavan, Kanat Tangwongsan, Srikanta Tirathapura, and Kun-Lung Wu. 2013. Counting and Sampling Triangles from a Graph Stream. *VLDB* 6, 14 (2013).
- [51] Tiago P Peixoto and Laetitia Gauvin. 2018. Change points, memory and epidemic spreading in temporal networks. *Scientific reports* 8, 1 (2018), 15511.
- [52] 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.
- [53] Ryan Rossi and Jennifer Neville. 2012. Time-evolving relational classification and ensemble methods. In *PAKDD*. Springer, 1–13.
- [54] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*. <http://networkrepository.com>
- [55] 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.
- [56] Ryan A. Rossi, Nesreen K. Ahmed, Eunye Koh, Sungchul Kim, Anup Rao, and Yasin Abbasi-Yadkori. 2020. A structural graph representation learning framework. In *WSDM*.
- [57] Ryan A. Rossi, Brian Gallagher, Jennifer Neville, and Keith Henderson. 2013. Modeling Dynamic Behavior in Large Evolving Graphs. In *WSDM*. 667–676.
- [58] 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*.
- [59] Umang Sharan and Jennifer Neville. 2008. Temporal-relational classifiers for prediction in evolving domains. In *ICDM*. 540–549.
- [60] 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.
- [61] Sucheta Soundarajan, Acar Tamer Soy, 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.
- [62] 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.
- [63] 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.
- [64] Rajmonda Sulo, Tanya Berger-Wolf, and Robert Grossman. 2010. Meaningful selection of temporal resolution for dynamic networks. In *MLG KDD*. 127–136.
- [65] Aynaz Taheri and Tanya Berger-Wolf. 2019. Predictive Temporal Embedding of Dynamic Graphs. *ASONAM* (2019).
- [66] Aynaz Taheri, Kevin Gimpel, and Tanya Berger-Wolf. 2019. Learning to Represent the Evolution of Dynamic Graphs with Recurrent Models. In *WWW*. 301–307.
- [67] 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.
- [68] 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.

- [69] Jeffrey S Vitter. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)* 11, 1 (1985), 37–57.
- [70] Rongjing Xiang, Jennifer Neville, and Monica Rogati. 2010. Modeling relationship strength in online social networks. In *WWW*. 981–990.
- [71] Xin Yang and Ju Fan. 2018. Influential User Subscription on Time-Decaying Social Streams. *arXiv:1802.05305* (2018).
- [72] Lorenzo Zino, Alessandro Rizzo, and Maurizio Porfiri. 2016. Continuous-time discrete-distribution theory for activity-driven networks. *Physical review letters* 117, 22 (2016), 228302.
- [73] 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.