ODEN: Simultaneous Approximation of Multiple Motif Counts in Large Temporal Networks

Ilie Sarpe
Department of Information Engineering
University of Padova
Padova, Italy
sarpeilie@dei.unipd.it

Fabio Vandin
Department of Information Engineering
University of Padova
Padova, Italy
fabio.vandin@unipd.it

ABSTRACT

Counting the number of occurrences of small connected subgraphs, called *temporal motifs*, has become a fundamental primitive for the analysis of *temporal networks*, whose edges are annotated with the time of the event they represent. One of the main complications in studying temporal motifs is the large number of motifs that can be built even with a limited number of vertices or edges. As a consequence, since in many applications motifs are employed for exploratory analyses, the user needs to iteratively select and analyze several motifs that represent different aspects of the network, resulting in an inefficient, time-consuming process. This problem is exacerbated in large networks, where the analysis of even a single motif is computationally demanding. As a solution, in this work we propose and study the problem of *simultaneously* counting the number of occurrences of *multiple* temporal motifs, all corresponding to the same (static) topology (e.g., a triangle). Given that for large temporal networks computing the exact counts is unfeasible, we propose ODEN, a sampling-based algorithm that provides an accurate approximation of *all* the counts of the motifs. We provide analytical bounds on the number of samples required by ODEN to compute rigorous, probabilistic, relative approximations. Our extensive experimental evaluation shows that ODEN enables the approximation of the counts of motifs in temporal networks in a fraction of the time needed by state-of-the-art methods, and that it also reports more accurate approximations than such methods.

CCS CONCEPTS

• **Mathematics of computing → Probabilistic algorithms**; • **Theory of computation → Graph algorithms analysis**.

KEYWORDS

temporal motifs, sampling algorithm, temporal networks, randomized algorithm

ACM Reference Format:
Ilie Sarpe and Fabio Vandin. 2021. ODEN: Simultaneous Approximation of Multiple Motif Counts in Large Temporal Networks. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482459

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482459

1 INTRODUCTION

Networks are ubiquitous representations that model a wide range of real-world systems, such as social networks [9], citation networks [10], biological systems [12], and many others [29]. One of the most fundamental primitives in network analysis is the mining of *motifs* [27, 28, 38] (or *graphlets* [7, 32]), which requires to count the occurrences of small connected subgraphs of k nodes. Motifs represent key building blocks of networks, and they provide useful insights in wide range of applications such as network classification [26, 39], network clustering [3], and community detection [2].

Modern networks contain rich information about their edges or nodes [8, 18, 34, 45] in addition to graph structure. One of the most important information is the time at which the interactions, represented by edges, occur. Networks for which such information is available are called *temporal* [14, 15]; novel insights about the underlying *dynamics* of the systems can be uncovered by the analysis of such networks [20–22]. In recent years, many primitives [16, 19, 31, 37] have been proposed as counterpart, in temporal networks, to the study of subgraph patterns for nontemporal networks, with each primitive capturing different temporal aspects of a network. One of the most important such primitives is the study of *temporal motifs* [31]. Temporal motifs are small connected subgraphs with k nodes and ℓ edges occurring with a prescribed order within a time interval of duration δ. Temporal motifs describe the patterns shaping interactions over the network, e.g., networks from similar domains tend to have similar temporal motif counts [31], and their analysis is useful in many applications, e.g., anomalies detection [4], network classification [40], and social networks [6].

The temporal dimension poses several challenges in the analyses of motifs. A major challenge is represented by the large number of temporal motifs that can be build even with a limited number of vertices and edges. For example, even considering directed (and connected) temporal motifs with only 3 vertices and 3 edges, there are 32 such motifs. In several domains when motifs are studied in the exploratory analysis of a temporal network it is almost impossible for the data analyst to known *a priori* which motif is the most interesting and useful. In social networks, a set of 3 vertices represents the smallest non trivial community, and different temporal motifs with 3 vertices describe different patterns of interactions in such community. Hence, studying all such motifs can provide novel insights on the interactions within such communities. In network classification, considering the counts of all the 32 motifs with 3 vertices and 3 edges lead to models with improved accuracy [40].

However, since state-of-the-art approaches for general temporal motifs only allow the analysis of one motif at the time, the user needs to *iteratively* select and analyze the various motifs, resulting in an inefficient and time consuming process, in particular for large networks.

In this paper, we define and study the problem of simultaneously counting the occurrences of various temporal motifs. In particular, we consider all motifs corresponding to the same *static* target template (e.g., all triangles - see Fig. 1a). This problem is extremely challenging, since computing the count of even a single temporal motif is NP-Hard in general [23], with existing state-of-the-art approaches having complexity exponential in the number of edges of the motif to obtain even a single motif's count [23, 36, 42].

The task of counting temporal motifs is hindered by the sheer size of modern datasets and, therefore, scalable techniques are needed to deal with such amount of data. Since exact approaches [13, 24, 31] are impractical, rigorous and efficient approximation algorithms providing tight guarantees are needed. In this work we develop ODEN, a sampling algorithm that provides a high quality approximation for the problem of counting multiple temporal motifs with the same static topology. Our main contributions are as follows:

- We propose the *motif template counting problem*, where, given a temporal network, a k-node target template graph H, the number ℓ of edges of each temporal motif, and a bound δ on the duration of the temporal motifs, the problem requires to output all the counts of the temporal motifs whose static topology corresponds to H and having exactly ℓ temporal edges, occurring within δ-time.
- We propose ODEN, a randomized sampling algorithm providing a high quality approximation for the motif template counting problem. ODEN's approach is to sample a set of motif occurrences, ensuring that they all share the same static topology H. Thus, ODEN takes advantage of the constraint that all motifs must share a common target template H, aggregating the computation of all motif counts in a sample. ODEN's approximation, as in other data mining applications, is controlled by two parameters ε, η, which control respectively the quality and the confidence of the approximations.
- We show a tight and efficiently computable bound on the number of samples required by ODEN for the approximation to be within ε error with confidence $> 1 - \eta$ for *all* temporal motif's counts.
- We perform large scale experiments using datasets with up to billions of temporal edges, showing that ODEN requires a fraction of the time required by state-of-the-art approximation algorithms for single motif counts, and that it reports sharper estimates. We then provide a parallel implementation of ODEN displaying almost linear speedup in many configurations. We also show how ODEN provides novel insights on the dynamics of a real-world temporal network.

2 PRELIMINARIES

In this section we introduce the basic notions that we will use throughout the work, and we define the computational problem of counting multiple temporal motifs sharing a common target template graph. We start by defining temporal networks.

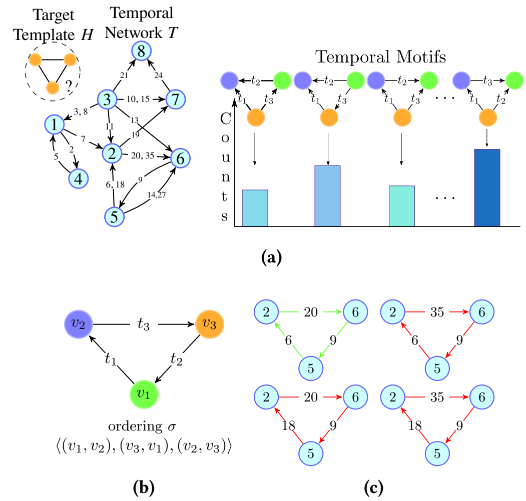


Figure 1: (1a): Motif template counting problem overview: given a temporal network and a (static) target template, compute the counts of all temporal motifs that map on the template. (1b): Temporal motif, with $k = 3, \ell = 3$, and its ordering σ. (1c): Sequences of edges of the network in (1a) among nodes $\{2, 5, 6\}$ that map topologically on the motif in (1b). For $\delta = 15$ only the green sequence is a δ-instance of the motif, since the timestamps respect σ and $t'_\ell - t'_1 = 20 - 6 \leq \delta$. The red sequences are not δ-instances, since they do not respect such constraint or do not respect the ordering σ.

Definition 2.1. A *temporal network* is a pair $T = (V, E)$ where, $V = \{v_1, \ldots, v_n\}$ and $E = \{(x, y, t) : x, y \in V, x \neq y, t \in \mathbb{R}^+\}$ with $|V| = n$ and $|E| = m$.

Given $(x, y, t) \in E$, we say that t is the *timestamp* of the directed edge (x, y). Given a temporal network T, by ignoring the timestamps of its edges we obtain the associated *undirected projected static network*, defined as follows.

Definition 2.2. The *undirected projected static network* of a temporal network $T = (V, E)$ is the pair $G_T = (V, E_T)$ that is an undirected network, such that $E_T = \{\{x, y\} : (x, y, t) \in E\}$.

We will often use the term *static network* to denote a network whose edges are without timestamps. Next we introduce the definition of *temporal motifs* as defined by Paranjape et al. [31], which are small, connected subgraphs representing patterns of interest.

Definition 2.3. A *k-node ℓ-edge temporal motif* M is a pair $M = (\mathcal{K}, \sigma)$ where $\mathcal{K} = (V_\mathcal{K}, E_\mathcal{K})$ is a directed and weakly connected *multigraph* where $V_\mathcal{K} = \{v_1, \ldots, v_k\}$, $E_\mathcal{K} = \{(x, y) : x, y \in V_\mathcal{K}, x \neq y\}$ s.t. $|V_\mathcal{K}| = k$ and $|E_\mathcal{K}| = \ell$, and σ is an ordering of $E_\mathcal{K}$.

Note that a k-node ℓ-edge temporal motif $M = (\mathcal{K}, \sigma)$ is also identified by the sequence $\langle(x_1, y_1), \ldots, (x_\ell, y_\ell)\rangle$ of edges ordered according to σ; we will often use such representation for a motif M (see Fig. (1b) for an example). Given a k-node ℓ-edge temporal motif

M, the values of k and ℓ are determined by $V_{\mathcal{K}}$ and $E_{\mathcal{K}}$. We will therefore use the term *temporal motif*, or simply *motif*, when k and ℓ are clear from context. Given a temporal motif $M = ((V_{\mathcal{K}}, E_{\mathcal{K}}), \sigma)$, we denote with $G_u[M]$ the undirected graph corresponding to the underlying undirected graph structure of the multigraph $\mathcal{K}$ of M, that is $G_u[M] = (V_{\mathcal{K}}, E^u_M)$ where $E^u_M = \{\{x,y\} : (x,y) \vee (y,x) \in E_{\mathcal{K}}\}$ (i.e., E^u_M is the *set* of undirected edges associated to the multiset $E_{\mathcal{K}}$). Notice that directed edges of the form $(x,y), (y,x)$ as well as multiple directed edges $(x,y), (x,y), \ldots$ from $E_{\mathcal{K}}$ are represented by the same undirected edge $\{x,y\}$ in E^u_M.

For a fixed temporal motif M, we are interested in identifying its realizations in T appearing within at most δ-time *duration*, as captured by the following definition.

Definition 2.4. Given a temporal network $T = (V, E)$ and $\delta \in \mathbb{R}^+$, a time ordered sequence $S = \langle (x'_1, y'_1, t'_1), \ldots, (x'_\ell, y'_\ell, t'_\ell) \rangle$ of ℓ unique temporal edges from T is a *δ-instance* of the temporal motif $M = \langle (x_1, y_1), \ldots, (x_\ell, y_\ell) \rangle$ if:

(1) there exists a bijection f on the vertices such that $f(x'_i) = x_i$ and $f(y'_i) = y_i$, $i = 1, \ldots, \ell$; and
(2) the edges of S occur within δ time, i.e., $t'_\ell - t'_1 \leq \delta$.

Exploring different values of δ in the above definition often leads to different insights on the temporal network that may be discovered through the analysis of the motifs [1, 14, 19, 30]. Note that in a δ-instance of the temporal motif $M = (\mathcal{K}, \sigma)$ the edge timestamps must be sorted according to the ordering σ (see Fig. (1c) for an example). In fact, σ plays a key role in defining a temporal motif, with different orderings of the same multigraph $\mathcal{K}$ reflecting diverse dynamic properties captured by the motif.

For a given directed multigraph $\mathcal{K}$ with $|E_{\mathcal{K}}| = \ell$ edges, in general not all the $\ell!$ orderings of its edges define *distinct* temporal motifs. We therefore introduce the following equivalence relation.

Definition 2.5. Let M_1 and M_2 be two temporal motifs. Let $M_1 = \langle (x^1_1, y^1_1), \ldots, (x^1_\ell, y^1_\ell) \rangle$, and $M_2 = \langle (x^2_1, y^2_1), \ldots, (x^2_\ell, y^2_\ell) \rangle$ be the sequences of edges of M_1 and M_2, respectively. We say that M_1 and M_2 are *not distinct* (denoted with $M_1 \cong_\tau M_2$) if there exists a bijection g on the vertices such that $g(x^1_i) = x^2_i$ and $g(y^1_i) = y^2_i$, $i = 1, \ldots, \ell$.

We provide an example of the definition above in Figure 2.

Given two networks (undirected or temporal) G, G' we say that $G' = (V', E')$ is a *subgraph* of $G = (V, E)$ (denoted with $G' \subseteq G$) if $V' \subseteq V$ and $E' \subseteq E$. Note that we require a subgraph to be *edge* induced. To conclude the preliminary notions, we recall the definition of *static graph isomorphism*.

Definition 2.6. Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ we say that the two graphs are *isomorphic*, denoted with $G \simeq H$ if and only if there exists a bijection $f : V_G \mapsto V_H$ on the vertices such that $e = (u, v) \in E_G \Leftrightarrow e' = (f(u), f(v)) \in E_H$.

Let $\mathcal{U}(M, \delta) = \{I : I \text{ is a } \delta\text{-instance of } M\}$ be the *set of (all) δ-instances* of the motif M in T. The *count* of M is $C_M(\delta) = |\mathcal{U}(M, \delta)|$, denoted with C_M when δ is clear from the context.

Given a static undirected graph H, which we call the *target template*, we are interested in solving the problem of computing the number of δ-instances of all temporal motifs with ℓ edges and all corresponding to the same static graph H. More formally, given the *target template* $H = (V_H, E_H)$, which is a simple and connected graph, and $\ell \geq |E_H| \in \mathbb{Z}_+$, let $\mathcal{M}(H, \ell)$ be the set of *distinct* temporal motifs with ℓ edges whose underlying undirected graph structure corresponds to H, that is $\mathcal{M}(H, \ell)$ contains motifs $M_i = ((V^i_{\mathcal{K}}, E^i_{\mathcal{K}}), \sigma_i)$, $i = 1, 2, \ldots$, such that i) $G_u[M_i] \simeq H$; ii) $|E^i_{\mathcal{K}}| = \ell$; and iii) $M_i \not\cong_\tau M_j, \forall j \neq i$.

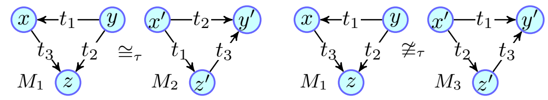


Figure 2: (Left): The two motifs are *not distinct*: let $\sigma_1 = \langle (y,x), (y,z), (x,z) \rangle$ and $\sigma_2 = \langle (x',z'), (x',y'), (z',y') \rangle$ corresponding to M_1 and M_2, then the function $f : V^1_{\mathcal{K}} \mapsto V^2_{\mathcal{K}}$ defined by $f(x) = z', f(y) = x', f(z) = y'$ preserves both the topology and the ordering as from Definition 2.5. (Right): The two motifs are *distinct* since there is no map $f : V^1_{\mathcal{K}} \mapsto V^3_{\mathcal{K}}$ preserving both the topology and ordering.

Let us explain intuitively the constrains above. First, H imposes a constraint on the *undirected* static topology the temporal motifs of interest (that are directed subgraphs) should have. That is, it requires all the motifs to have the same underlying graph structure ($G_u[M]$), which must be isomorphic to H. This is a useful way to represent multiple related temporal motifs. For example, in social network analysis by fixing H as an undirected triangle we consider in $\mathcal{M}(H, \ell)$ all temporal motifs that characterize the communication between groups of three friends (i.e., each motif will represent a *different* form of communication among all such groups [31]). The second constraint requires each motif $M_i \in \mathcal{M}(H, \ell)$ to have *exactly* $\ell \geq |E_H|$ edges, with ℓ provided in input by the user. Fixing the parameter ℓ is motivated by the fact that motifs with different values of ℓ (even with the same target template structure H) reflect different patterns of interaction (e.g, a group of friends that exchanges $\ell = 3$ or $\ell = 4$ messages). As we will show empirically in Section 5.4, such counts vary significantly with ℓ for fixed H and δ. Finally, the third constraint ensures that we only count distinct motifs, i.e., motifs representing different patterns.

We now define the motif template counting problem.

Problem 1. ***Motif template counting problem.*** *Given a temporal network T, a static* undirected *target graph $H = (V_H, E_H)$, $\ell \in \mathbb{Z}_+, \ell \geq |E_H|$, and a parameter $\delta \in \mathbb{R}_+$, find the counts $C_{M_i}(\delta)$ of motifs $M_i \in \mathcal{M}(H, \ell)$, $i = 1, \ldots, |\mathcal{M}(H, \ell)|$ in T.*

We now provide an example of the different motifs to be counted for different values of ℓ with a fixed target template H.

Example 2.7. Let $H = (\{v_1, v_2\}, \{\{v_1, v_2\}\})$, that is, the target template is an edge. Let $e_1 = (v_1, v_2)$ and $e_2 = (v_2, v_1)$. By varying $\ell \in \{2, 3\}$ the motifs in $\mathcal{M}(H, \ell)$, for which we want to compute the counts, are: $M_1 = \langle e_1, e_1 \rangle$ and $M_2 = \langle e_1, e_2 \rangle$ for $\ell = 2$ (i.e., $|\mathcal{M}(H, 2)| = 2$) while $M_1 = \langle e_1, e_1, e_1 \rangle, M_2 = \langle e_1, e_2, e_1 \rangle, M_3 = \langle e_1, e_2, e_2 \rangle, M_4 = \langle e_1, e_1, e_2 \rangle$ for $\ell = 3$ (i.e., $|\mathcal{M}(H, 3)| = 4$).

Since solving the counting problem *exactly* is NP-Hard in general[1] even for one single temporal motif, we aim at providing high-quality approximations to the motif counts as follows.

[1]The hardness depends on the topology of the motif. For example for triangles and single edges there exist polynomial time-algorithms, even if they are impracticable

Problem 2. ***Motif template approximation problem.*** *Given the input parameters of Problem 1 and additional parameters $\varepsilon \in \mathbb{R}_+, \eta \in (0,1)$, compute approximations $C'_{M_i}(\delta)$ of counts $C_{M_i}(\delta)$ of motifs $M_i \in \mathcal{M}(H,\ell), i = 1,\ldots,|\mathcal{M}(H,\ell)|$, such that $\mathbb{P}[\exists i \in \{1,\ldots,|\mathcal{M}(H,\ell)|\} : |C'_{M_i}(\delta) - C_{M_i}(\delta)| \geq \varepsilon C_{M_i}(\delta)] \leq \eta$, that is $C'_{M_i}(\delta)$ is a relative ε-approximation to the count $C_{M_i}(\delta)$ with probability $\geq 1-\eta$ for all $i = 1,\ldots,|\mathcal{M}(H,\ell)|$ simultaneously.*

3 RELATED WORKS

Much work has been done on enumerating and approximating k-node motifs in (nontemporal) networks. We refer the interested reader to the surveys [33, 43]. However, such works cannot be easily adapted to temporal motifs since they do not properly account for the temporal information [14, 31]. Many different definitions of temporal networks and temporal patterns have been proposed: here we will focus only on those works that are relevant for our work, the interested reader may refer to [14, 15, 17, 25] for a more general overview.

Our work builds on the work of Paranjape et al. [31] which first introduced the definition of temporal motif used here, and the problem of counting single temporal motifs. The authors provided a general algorithm for counting a single temporal motif by enumerating all the subsequences of edges that map on a single static subgraph. Their approach is not feasible on large datasets since it requires exhaustive enumeration of *all* subgraphs of the undirected projected static network G_T that are isomorphic to the target template H. The authors also proposed efficient algorithms and data-structures for counting 3-node 3-edge motifs, which may be used for the exact counting subroutines within oᴅᴇN sampling framework. In addition to the algorithmic contributions, the authors also showed that networks from similar domains tend to exhibit similar temporal motif counts. They also showed how motif counts can provide significant insights on the communication patterns in many networks, highlighting the importance of studying temporal motifs in temporal networks.

Other *exact* algorithms have been proposed for the problem of counting a single motif, or for slightly different problems. Mackey et al. [24] presented a backtracking algorithm for counting a single temporal motif that can be use for any motif. Boekhout et al. [6] developed exact algorithms for counting temporal motifs in multi-layer temporal networks (i.e., each edge is a tuple (x, y, t, a) with a denoting the layer of each edge), they also discuss efficient data-structures for counting 4-node 4-edge motifs, which may also be adapted for the exact counting subroutines in our sampling framework oᴅᴇN. Being exact, both such algorithms do not scale on massive datasets due to large time and memory requirements.

Several approximation algorithms have been proposed in recent years for estimating the count of a *single* motif. Liu et al. [23] proposed a temporal-partition based sampling approach. Wang et al. [42] introduced a sampling-based algorithm that selects temporal edges with a fixed probability specified by the user. Lastly, Sarpe and Vandin [36] proposed PRESTO, an algorithm based on uniform sampling of small windows of the temporal network T. All such sampling algorithms can be used to analyze a single temporal motif but become inefficient as the number of motifs to be counted grows, such as in Problem 2. In fact, they cannot leverage the additional information that all motifs $M_1, \ldots, M_{|\mathcal{M}(H,\ell)|}$ must share a common static topology isomorphic to H. As stated in Section 1, when analysing a temporal network it is hard to know *a-priori* which motif is representing important functions for the network, therefore one often relies on testing all possible orderings σ over one fixed target template H for fixed ℓ, δ [31, 40] (as in Prob. 1) resulting in a time consuming and inefficient procedure. Our approach instead supports the direct analysis of *multiple* temporal motifs, enabling the study of hundreds of temporal motifs on massive networks in a very limited time.

on very large networks. Interestingly, counting temporal star-shaped motifs is NP-Hard [23], while on static networks such motifs can be counted in polynomial time.

4 ODEN

In this section we present oᴅᴇN, our algorithm to address the motif template approximation problem (Prob. 2). We start in Section 4.1 with an overview of oᴅᴇN. We then describe the algorithm in Section 4.2, analyze its time complexity in Section 4.3 and its theoretical guarantees, including an efficiently computable bound on the number of samples required to obtain the desired probabilistic guarantees, in Section 4.4.

4.1 Overview of oᴅᴇN

Our algorithm oᴅᴇN estimates of the counts of motifs in $\mathcal{M}(H,\ell)$. The main idea is to avoid the explicit generation all the motifs $M_i \in \mathcal{M}(H,\ell), i = 1,\ldots,|\mathcal{M}(H,\ell)|$ to count them one at the time as it is required by existing algorithms that approximate a single motif count. oᴅᴇN instead leverages the fact that the topology of all motifs must to be isomorphic to the target template H, by reusing the computation while estimating the motif counts.

An overview of the main strategy adopted by our algorithm is presented in Figure 3. Given the input parameters of Problem 2, where H is the target template, the idea behind our procedure is to consider the undirected static projected graph G_T of the input temporal network T and proceed as follows: i) find a set of subgraphs in the static graph G_T that are isomorphic to H by first *sampling* an edge e_R of G_T with some probability p_{e_R}, where p_{e_R} depends, potentially, on e_R and the temporal network T, and then enumerating all subgraphs of G_T isomorphic to H and containing e_R; ii) for each such subgraph, consider the corresponding temporal subgraph and compute all the counts of the subsequences of ℓ edges occurring within δ-time in such temporal subgraph; iii) for each such subsequence identified, find the corresponding motif in $\mathcal{M}(H,\ell)$, for which the subsequence is a δ-instance of, and update a count for each motif identified; iv) weight each motif count opportunely in order to maintain an unbiased estimate of global motif counts; v) repeat steps i)-iv) a sufficient number of iterations to guarantee the desired (ε, η)-approximation (see Problem 2).

4.2 Algorithm Description

oᴅᴇN is described in Algorithm 1. It first computes $G_T = (V, E_T)$, the undirected projected static graph of T (line 1), and initializes $C_{estimates}$ (line 2) used to store the estimates of motif counts, which are used to compute the estimators $C'_{M_i}, i = 1,\ldots,|\mathcal{M}(H,\ell)|$. Then it repeats s times (line 3) the following procedure: i) pick a random edge e_R from G_T (line 4) according to some probability distribution

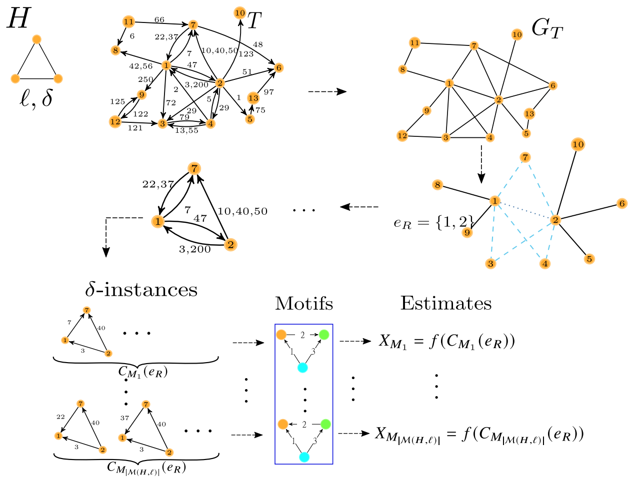


Figure 3: Overview of oᴅᴇN's approximation strategy. Let H be a triangle, and $\ell = 3, \delta = 40$. oᴅᴇN first collects the static projected network G_T, then samples an edge $e_R \in G_T$ randomly ($e_R = \{1, 2\}$ in the figure) and enumerates all the subgraphs of G_T isomorphic to H containing e_R. For each subgraph it collects the corresponding temporal network, counts the δ-instances of the motifs, and combines the different counts to obtain unbiased estimates of motif counts. This procedure is repeated to obtain concentrated estimates.

over the edges of E_T; ii) enumerate all the subgraphs h of G_T such that $h \simeq H$ and $e_R \in h$ (line 5); note that this enumeration step is local to e_R; iii) for each such h (line 6), collect the corresponding temporal graph, i.e., all edges in T for which their static projected edge is an edge of h (line 7), sort the sequence of edges of such graph by increasing timestamps and apply some pruning criteria (lines 8-9); iv) if the sequence is not pruned, then update the estimates of the number of δ-instances of each temporal motif by calling the routine `FastUpdate` (line 10). `FastUpdate` features an efficient implementation of the general algorithm by Paranjape et al. [31], for which we devised efficient encodings of the motifs within integers through bitwise operations. Such function updates $C_{estimates}$ in order to maintain for each motif the count that will be used to output its unbiased estimate (see Appendix B of [35] for details). Let $C_{M_i}(e)$ be the number of δ-instances in T of $M_i, i = 1, \ldots, |\mathcal{M}(H, \ell)|$ whose undirected projected static network contains edge $e \in G_T$. `FastUpdate` updates the estimate of the counts for each motif M_i by summing its unbiased estimate obtained at the j-th iteration (i.e., $X^j_{M_i} = C_{M_i}(e_R)/(|E_H| p_{e_R})$). Once the procedure is repeated s times, for each motif $M_i \in \mathcal{M}(H, \ell), i = 1, \ldots, |\mathcal{M}(H, \ell)|$, oᴅᴇN computes the final estimate $C'_{M_i} = \frac{1}{s}\sum_{j=1}^{s} X^j_{M_i}$ where $X^j_{M_i} = \frac{1}{|E_H|}\sum_{e \in G_T} C_{M_i}(e) X_e / p_e$ is the estimate obtained at the j-th iteration (with X_e being a bernoulli random variable denoting if edge $e \in G_T$ is sampled at the j-th iteration, s.t. $\mathbb{P}[X_e = 1] = p_e$) and outputs it together with the motif (we output σ_i over the node-set V_H) (lines 12-13). We show in Lemma 4.1 that oᴅᴇN outputs unbiased estimates for all the motif counts.

Algorithm 1: oᴅᴇN

Input: $T = (V, E), H = (V_H, E_H), \delta, s, \ell$

Output: $(M_i, C'_{M_i}), i = 1, \ldots, |\mathcal{M}(H, \ell)|$ where C'_{M_i} is an estimate of C_{M_i} for the motifs in $\mathcal{M}(H, \ell)$.

1 $G_T = (V, E_T) \leftarrow$ `UndirectedStaticProjection`(T)
2 $C_{estimates} \leftarrow \{\}$
3 **for** $j \leftarrow 1$ *to* s **do**
4     $e_R = \{x_R, y_R\} \leftarrow$ `RandomEdge`$(p(e) : e \in E_T)$
5     $\mathcal{H} \leftarrow \{h \subseteq G_T : h \simeq H, \{x_R, y_R\} \in h\}$
6   **foreach** $h \in \mathcal{H}$ **do**
7         $S \leftarrow \{(x, y, t), (y, x, t) \in E : \{x, y\} \in h\}$
8   `SortInPlace`(S) ▷ `By increasing timestamps`
9   **if** `*Pruning criteria are not met*` **then**
10   `FastUpdate`$(\delta, S, C_{estimates}, p(e_R), H)$
11 **foreach** $(M, X_M) \in C_{estimates}$ **do**
12     $C'_M \leftarrow \frac{X_M}{s}$
13   **output** (M, C'_M)

We briefly discuss the pruning criteria used in line 9. Given a candidate temporal graph S for which $G_S \simeq H$ holds, we check in linear time if S can contain a δ-instance of a motif or not: since S is already sorted by increasing timestamps (see line 8), we efficiently check if there are at least ℓ edges within δ-time. If not, then we prune the sequence (since by definition a δ-instance of a motif with k-nodes, and ℓ-edges must have ℓ edges occurring within δ-time). We thus avoid calling the subroutine `FastUpdate`, which has an exponential complexity in general (see Section 4.3), on S.

We now discuss the probability distribution used to sample a random edge e_R from G_T (line 4), while we describe the subroutine `FastUpdate` that updates the motif estimates at each iteration (line 10) and the algorithms employed for the static enumeration in Appendix B of [35] for space constraints (Sections B.1 and B.2).

Since our final estimate is an average over s samples of the variables $X^j_{M_i}, i = 1, \ldots, |\mathcal{M}(H, \ell)|, j = 1, \ldots, s$, and given that $X^j_{M_i}$ is an unbiased estimate (see Lemma 4.1) the final estimate is also a consistent estimator (i.e., it converges to C_{M_i} as $s \to \infty$) if each edge has a positive probability of being sampled[2]. Thus *any* probability mass assigning positive probabilities on edges can be adopted. We considered different distributions over the edges of E_T:

(1) *Uniform*: $p_e = 1/|E_T|, e \in E_T$;
(2) *Static degree based*: $p_e = d(e)/(\sum_{e' \in E_T} d(e')), e \in E_T$ where $d(e = \{x, y\}) = d(x) + d(y)$ is the degree of the edge as sum of the degree of its nodes $x, y \in V$ in G_T;
(3) *Temporal degree based*: $p_e = \phi(e)/(\sum_{e' \in E_T} \phi(e'))$ with $\phi(e = \{x, y\}) = |\{t : \exists (x, z, t) \vee (z, x, t) \in E\}| + |\{t : \exists (z, y, t) \vee (y, z, t) \in E, z \neq x\}|, e \in E_T$;
(4) *Temporal edge weight based*: $p_{e=\{x,y\}} = |\{(x, y, t), (y, x, t) \in E\}|/m, e \in E_T$;

We empirically found the distribution (4) to be the fastest to converge for small number s of iterations, thus we use it in our

[2]More formally it is only necessary to assign to each δ-instance a known positive sampling probability.

analysis. We observe that many other candidate distributions can be designed (e.g., combining two of those already listed with weights $\xi, 1-\xi, \xi \in (0,1)$) making our framework extremely versatile.

We conclude by summarizing some nice properties of our algorithm: 1) it computes the estimates only for the temporal motifs occurring in the input temporal network T (except for the very unpractical case where the motifs in $\mathcal{M}(H,\ell)$ have all zero counts) without generating all the possible candidates, while existing sampling techniques require to first generate all the candidates and then to execute the algorithms on such candidates, even for motifs with zero counts; 2) it takes advantage of the constraint that all motifs share the same underlying topology (H), saving computation when estimating the different counts; 3) it is trivially parallelizable: all the s iterations can be executed in parallel; 4) it can easily use most of the fast state-of-the-art subgraph enumeration algorithms developed for the exact subgraph isomorphism problem (see Appendix B.2 of [35]).

4.3 Time Complexity

In this section we briefly describe the time complexity of ODEN. ODEN needs to compute the probabilities $p(e)$ of edges in advance, which requires a $O(|E_T|)$ preprocessing step. Interestingly, this step does not depend on the target template H, so it can be reused for different target templates H. One of the most expensive steps in Algorithm 1 is the local enumeration to identify the set $\mathcal{H}$ which in general requires exponential time (line 5). For specific topologies this step can be implemented very efficiently with symmetry breaking conditions and min-degree expansion. For example, if H is a triangle this “local” enumeration to $e_R = \{x_R, y_R\}$ can be done in $O(\min(d_{x_R}, d_{y_R}))$ time. Let $|\mathcal{H}^*|$ be the maximum cardinality of a set of subgraphs isomorphic to H and adjacent to an edge in G_T. Let $|S^*|$ denote the maximum cardinality of a set S collected (in line 7) by our algorithm ODEN. Sorting S^* requires $O(|S^*|\log|S^*|)$ time. The subroutine `FastCount` has a complexity dominated by $O((|S^*|+\ell)|E_H|^{\ell})$ (see [31] and App. B.1 of [35] for more details). So overall the complexity of our procedure is $O(|E_T| + s(\zeta_{enum} + |\mathcal{H}^*|(|S^*|\log(|S^*|) + |E_H|^{\ell}(|S^*|+\ell))))$, where ζ_{enum} is the time required by the static enumerator used as subroutine to compute the set $\mathcal{H}^*$. Such step in general is exponential in the number of edges of $|E_T|$ and depends on the exact technique used as subroutine. The final complexity accounts for the cycle (in line 3) that is repeated s times. The parallel version of our algorithm, which executes the cycle of line 3 in parallel on ω processing units available, leads to a time complexity of $O(|E_T| + s/\omega(\zeta_{enum} + |\mathcal{H}^*|(|S^*|\log(|S^*|) + |E_H|^{\ell}(|S^*|+\ell))))$.

4.4 Theoretical Guarantees

In this section we present the theoretical guarantees provided by ODEN. All proofs are provided in Appendix D of [35].

Recall that our algorithm outputs, for each motif $M_i \in \mathcal{M}(H,\ell), i = 1,\ldots,|\mathcal{M}(H,\ell)|$, the following estimate: $C'_{M_i} = \frac{1}{s}\sum_{j=1}^{s} X^j_{M_i} = \frac{1}{s|E_H|}\sum_{j=1}^{s}\sum_{e\in G_T} C_M(e)X_e/p_e$. The following shows that such estimates are unbiased estimates of $C_{M_i}, i = 1,\ldots,|\mathcal{M}(H,\ell)|$.

Lemma 4.1. *For each motif-count pair (M_i, C'_{M_i}) reported in output by ODEN, C'_{M_i} is an unbiased estimate to C_{M_i}, that is $\mathbb{E}[C'_{M_i}] = C_{M_i}$*

Let $\alpha = \min_{\{x,y\}\in E_T}\{|\{(x,y,t),(y,x,t) \in E\}|\}$, i.e., the minimum number of temporal edges of T that map on an edge in G_T. We now give an upper bound to the variance of the estimates provided by Algorithm 1 for each motif reported in output.

Lemma 4.2. *For each motif-count pair (M_i, C'_{M_i}) reported in output by ODEN, it holds $Var[C'_{M_i}] \le \frac{C^2_{M_i}}{s}\left(\frac{m}{\alpha|E_H|} - 1\right)$*

To give a bound on the number s of samples required by ODEN to output a ε-approximation that holds on all motifs in output with probability $> 1-\eta$, we combine Bennett’s inequality [5], an advanced result on the concentration of sums for independent random variables as reported in [36], with a union bound, obtaining the following main result.

Theorem 4.3. *Let s be the number of iterations of ODEN, let $\varepsilon \in \mathbb{R}^+$, and $\eta \in (0,1)$. If $s \ge \left(\frac{m}{\alpha|E_H|} - 1\right)\frac{1}{(1+\varepsilon)\ln(1+\varepsilon)-\varepsilon}\ln\left(\frac{2|\mathcal{M}(H,\ell)|}{\eta}\right)$ then*

$$\mathbb{P}[\exists i \in \{1,\ldots,|\mathcal{M}(H,\ell)|\} : |C'_{M_i} - C_{M_i}| \ge \varepsilon C_{M_i}] \le \eta.$$

5 EXPERIMENTAL EVALUATION

We implemented ODEN and tested it on several large datasets (see Section 5.1 for details on setup, and data). Our experimental evaluation has the following goals: compare ODEN with state-of-the-art algorithms for approximating motif counts (Section 5.2); evaluate the scalability of a simple parallel implementation of ODEN (Section 5.3); provide a case study highlighting the usefulness of using ODEN (Section 5.4) to analyze real-world temporal networks.

5.1 Setup, and Datasets

We briefly describe the setup and the large-scale datasets used in our experimental evaluation.

We implemented our algorithm ODEN in C++20 and compiled it under `gcc 9.3` with optimization flag enabled (implementation available at https://github.com/VandinLab/odeN), additional details on the implementation are in Appendix C of [35]. We compared ODEN with four different baselines, denoted as `PRESTO-A` (`PR-A`), `PRESTO-E` (`PR-E`) [36], `LS` [23], and `ES` [42]. We used the original implementations available from the authors. We performed all experiments under Ubuntu 20.04 on a machine with 64 cores, Intel Xeon E5-2698 2.3GHz, running each algorithm single threaded and with 300GB of maximum RAM allowed.

The datasets used in our experimental evaluation are reported in Table 1, which shows the number of nodes and edges of T, the precision of the timestamps, the timespan of the network, the number $|E_T|$ of undirected edges in the corresponding undirected projected static network G_T, the maximum degree $d_{\max}$ of a node in G_T and the maximum number $w_{\max}$ of temporal edges that are mapped on the same static edge in G_T. The datasets are from different domains: SO is a network that models interactions from the Stack-Overflow platform [31], BI is a network of Bitcoin transactions [23], RE a network built from comments on the platform Reddit [23], and EC is a *bipartite* temporal network build from IPv4 packets exchanged

Table 1: Datasets used and their statistics. See Section 5.1 for details on the statistics reported.

Name	n	m	$\lvert E_T \rvert$	$d_{\max}$	$w_{\max}$	Precision	Timespan
SO	2.58M	47.9M	28.1M	44K	594	sec	2774 (days)
BI	48.1M	113M	84.3M	2.4M	24.2K	sec	2585 (days)
RE	8.40M	636M	435.3M	0.3M	165K	sec	3687 (days)
EC	11.16M	2.32B	66.8M	0.3M	3.8M	μ-sec	62.0 (mins)

between Chicago and Seattle [36]. See the original papers for more details on the networks and the processes they model.

When measuring the running times for the various algorithms we exclude the time to read the dataset. Since ES's implementation supports only values of ℓ up to 4, we do not report results for ES and $\ell > 4$. Unless otherwise stated we used $\delta = 86400$ for SO and RE, $\delta = 43200$ on BI, and $\delta = 50000$ on EC, as done in previous works [23, 31, 42]. Since all algorithms used in our comparison have different parameters and only ODEN counts multiple motifs simultaneously, we used the following procedure to choose the parameters. For a given target template H and ℓ, we run PRESTO-A, PRESTO-E, LS, and ES for each motif in $\mathcal{M}(H,\ell)$ with fixed parameters, and computed their running time as the sum of the running times required by the single motifs in $\mathcal{M}(H,\ell)$. We then fixed the parameters of ODEN so that its running time would be at most the same as the other methods, or be close to it. All the parameters used in the experiments (including sample sizes) are reported with the source code. To extract the *exact* counts of motifs we used a modified version of the algorithm by Mackey et al. [24]. We do not report the running times of such algorithm since, even though it employs parallelism, it still runs several orders of magnitude slower than approximate approaches.

5.2 Approximation Quality and Running Time

In this section we compared the quality of the estimates and the running times of ODEN and the baseline sampling approaches.

To evaluate the approximations qualities we used the MAPE (Mean Average Percentage Error) metric over ten executions of each algorithm and parameter configuration. The MAPE is computed as follows: let C'_{M_i} be the estimate of $C_{M_i}, i = 1,\ldots,|\mathcal{M}(H,\ell)|$, returned by an algorithm, then the relative error of such estimate is $|C'_{M_i} - C_{M_i}|/C_{M_i}$. The MAPE is the average over the ten runs of the relative errors, in percentage. On each of the ten runs we also measured the running time of each algorithm, for which we will report the arithmetic mean.

We first discuss the quality of the estimates for different datasets when H is a triangle and $\ell \in \{4,5\}$. For $\ell = 4$ there are $|\mathcal{M}(H,\ell)| = 96$ triangles, while for $\ell = 5, |\mathcal{M}(H,\ell)|$ is 800. So as long as ℓ increases the approximation task becomes more challenging, due to the exponential growth of the number of motifs. We also observe that, to the best of our knowledge, such a huge number of temporal motifs was never tested before on large datasets due to the limitations of existing algorithms, while, as we will show, ODEN renders the approximation task practical even on hundreds of motifs.

The results on the SO dataset are shown in Figure 4a. ODEN provides much sharper estimates than state-of-the-art sampling techniques for single motif estimations on motifs $M_1,\ldots,M_{|\mathcal{M}(H,\ell)|}$: the relative error on $\ell = 4$-edge triangles is bounded by 5%, and

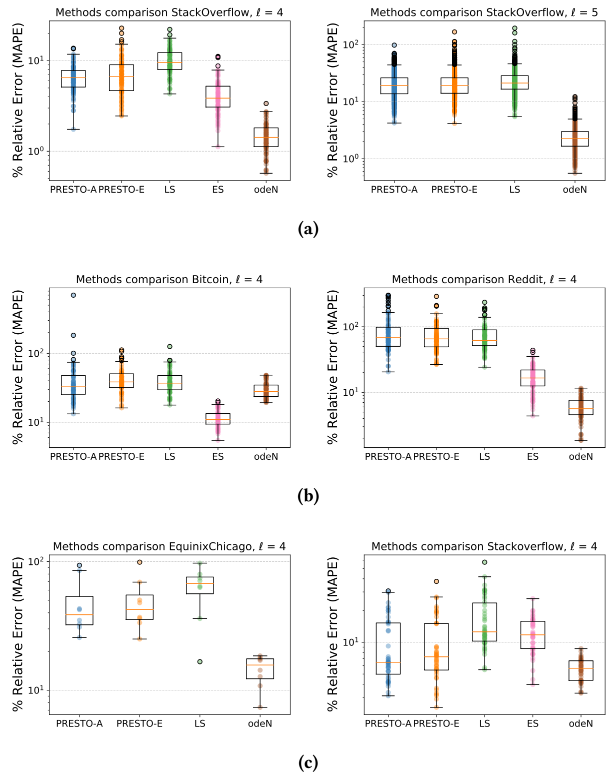


Figure 4: Approximation error on different datasets. (4a): SO dataset, H is a triangle, for $\ell = 4$ (left) and $\ell = 5$ (right). (4b): H is a triangle, $\ell = 4$, BI dataset (left) and RE dataset (right). (4c): EC dataset, H is an edge, $\ell = 4$ (left); SO dataset, H is a square, $\ell = 4$.

for $\ell = 5$-edge triangles (where $|\mathcal{M}(H,\ell)| = 800$) the relative error is bounded by 12% while state-of-the-art algorithms report much less accurate estimates, with twice the relative error of ODEN, on each configuration. We report the running times to obtain such estimates in Table 2. Interestingly, ODEN is more than 3× faster with $\ell = 4$ than any sampling algorithm and 1.7× faster with $\ell = 5$. For the other datasets, since extracting all the exact counts for $\ell > 4$ is extremely time consuming, requiring up to months of computation, we will not discuss the approximation qualities for $\ell = 5$ (since we do not have the exact counts to evaluate them).

On dataset BI (Figure 4b left) ODEN provides more concentrated estimates for the $|\mathcal{M}(H,\ell)| = 96$ triangles than other algorithms but ES, which also has a smaller running time than ODEN. This may be related to the static graph structure of BI, which has some very high-degree nodes (see Table 1). Therefore ODEN may sample edges with very high degree nodes, introducing an over counting in its estimates. Nonetheless, for higher values of ℓ this issue is amortized over the growing number of motifs $|\mathcal{M}(H,\ell)|$.

On dataset RE (Figure 4b right) the estimates by ODEN are all within 13% of relative error and improve significantly over state-of-the-art sampling algorithms, up to one order of magnitude of precision. Such estimates were notably obtained with significantly

Table 2: Running times (in seconds) to obtain the results in Figure 4 (results are showed following the order in Figure 4). Under H we report the topology of H used: T for triangles, E for edges, and S for squares. "-" denotes not applicable, while "✗" denotes out of RAM.

Dataset	ℓ	H	PR-A	PR-E	LS	ES	ODEN
SO	4	T	533.4	537.7	555.5	567.2	**174.4**
SO	5	T	4405	4408	4390	-	**2515**
BI	4	T	2048.6	2065.2	2754.6	**1602.9**	1948.9
RE	4	T	9787.1	10165.8	14289.7	13172.3	**6814.9**
EC	4	E	2581.5	3014.9	2981.9	✗	**1234.3**
SO	4	S	15613.7	16718.7	14344.6	26118.3	**4517.9**

smaller running time than state-of-the-art sampling algorithms, improving up to 2× the running time of ES and 1.4× over PRESTO (as reported in Table 2).

Finally, on the EC datasets, which is a bipartite temporal network with more than 2 billion edges we evaluated the approximation qualities with H being an edge and $\ell = 4$ (for which $|\mathcal{M}(H,\ell)| = 8$), such motifs have fundamental importance in the analysis of temporal networks since they can be seen as building blocks [15, 44]. We report the results on such motifs in Figure 4c (left) (ES is not shown since it did not terminate with the allowed memory budget). The estimates of ODEN are well concentrated and within 20% of relative error, while other sampling approaches provide approximations with a relative error up to 90% or more. Moreover, ODEN's results were obtained with a speedup of at least 2× over all the other sampling algorithms, rendering the approximations task feasible in a small amount of time on very large temporal networks.

To illustrate the enormous advantage of ODEN over existing state of the art exact and approximation algorithms, we compared the various algorithms on dataset SO when H is set to be a square and $\ell = 4$, for which $|\mathcal{M}(H,\ell)| = 48$. As [42] observed, among the 4-edge square motifs there are 16 motifs that do not grow as a single component (i.e., their orderings start with $\langle(1,2)(3,4)\cdots\rangle$). Estimating the counts of such motifs is particularly hard for most of the current state-of-the-art sampling algorithms since they generate a large number of partial matchings, while such aspect does not impact ODEN. The results are shown in Figure 4c (right). ODEN provides tight approximations under 9% of relative error for *all* four-edge square motifs, while other sampling algorithms fail to provide sharp estimates for some of the motifs. Surprisingly, as shown in Table 2, to obtain such estimates ODEN required less than 1.3 hours of computation while the exact computation of the counts required more than two weeks, and ODEN it is at least 3× times faster than all algorithms, and it is 5.4× times faster than ES.

Overall, these results show that our algorithm ODEN achieves much more precise estimates within a significant smaller running time than state of the art sampling algorithms when estimating the counts $C_{M_1},\dots,C_{M_{|\mathcal{M}(H,\ell)|}}$ for different values of ℓ and different topologies of the target template H (see Problem 1 in Section 2).

5.3 Parallel Implementation

In this section we briefly describe the advantages of a simple parallel implementation of Algorithm 1. As discussed in Section 4.2 the for cycle (from line 3) can be trivially parallelized, therefore we implemented such strategy through a *thread pooling* design pattern.

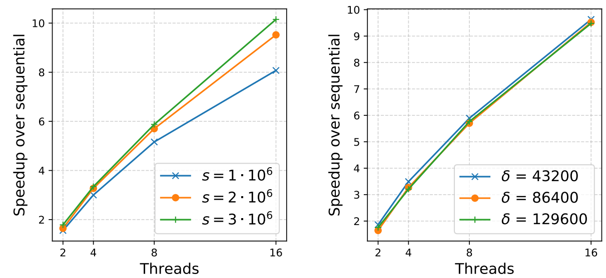


Figure 5: Speed-up of ODEN's parallel implementation. (Left): Varying s and fixed δ; (Right) Varying δ and fixed s.

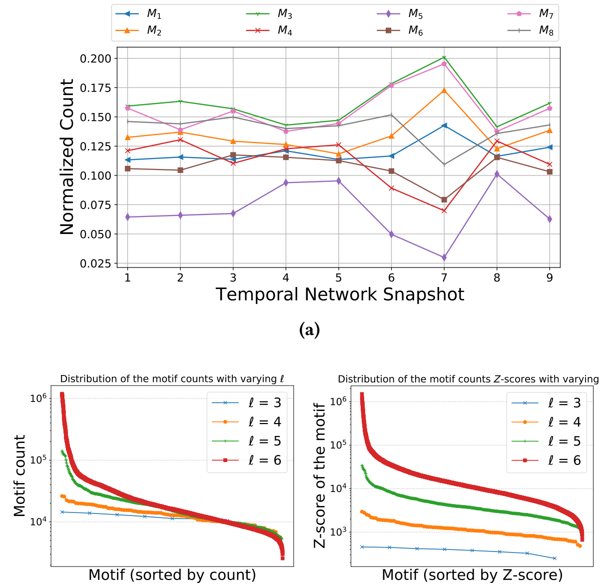
We describe the results obtained with H set to be a triangle, $\ell = 4$, and on the dataset SO; similar results are observed for other datasets. We tested the speedup achieved with $\omega \in \{2,4,8,16\}$ threads over the sequential implementation. Let T_ω the average running time with ω threads over ten execution of ODEN with fixed parameters, with T_1 being the average time for running the algorithm sequentially. We report the value of $T_1/T_\omega, \omega \in \{2,4,8,16\}$, i.e., the speedup over the sequential implementation. Fig. 5 (Left) shows the speedup across different values of the sample size s, with $\delta = 86400$. We observe an almost linear speedup up to 4 threads and then a slightly worse performance, especially for small sample sizes, that may be related to the time needed to process each sample. Fig. 5 (Right) shows how the speedup changes for $s = 2 \cdot 10^6$ and different values of δ. We note that our algorithm ODEN seems not to be impacted by the value of δ, and always attaining similar performances. Interestingly, as captured by our analysis in Section 4.3, the algorithm does not reach a fully linear speedup since we did not parallelized the computation of the sampling probabilities $p(e), e \in E_T$. As a remark, our parallel implementation is not optimized, and more advanced parallel strategies may substantially increase its speedup.

5.4 A Case Study

In this section we illustrate how counting multiple motifs, corresponding to the same target template H, with ODEN can be used to extract useful insights from a temporal network. We consider a real-world activity network from Facebook [41]. In such network, each node represents a user and a temporal edge (u,v,t) indicates that user u posted on v's wall at time t (see the original publication [41] for more details). The network contains information collected from September 2006 to January 2009. After removing self-loops, the network has n=45.7K nodes, m=826K temporal edges, and $|E_T|$=179K static (undirected) edges. We will fist show how analyzing the motif counts obtained with ODEN provides complementary insights to those in [41], that relied on mostly static analyses. We then conclude by discussing how the counts of the network evolve by varying only the parameter ℓ (i.e., fixing H,δ), showing that such counts surprisingly differ with different values of such parameter.

In the original paper [41], the authors partitioned the Facebook network in nine different snapshots (obtaining nine projected static networks), with each snapshot spanning 90 days of interactions in the network. The authors observed that consecutive snapshots have small resemblance, i.e., on average only 45% of the edges are preserved through consecutive snapshots. The authors also observed that despite this difference all the snapshots have similar, almost invariant, structural properties in terms of their clustering coefficient, average degree distribution, and others. We used ODEN (with $\varepsilon = 1, \eta = 0.1$) to compare the temporal networks associated to the snapshots by computing the counts of the 8 temporal motifs in $\mathcal{M}(H, \ell = 3)$ with H being a triangle and $\delta = 86400 = 1$ day. On each snapshot, after extracting the motif counts, we computed for each motif M its normalized count on the snapshot as $C_M / \sum_{i=1}^{8} C_{M_i}$. The results are reported in Fig. (6a) (see Appendix E of [35] for a visual representation of the motifs). Interestingly, even if in [41] the authors highlight small resemblance through different snapshots, the counts of the motifs are stable across the different snapshots, especially by looking at the first three and the last two snapshots. Surprisingly on snapshots 6 and 7, which correspond to the period of observation of mid-2008, we observe that there is a significant variation in the motif counts w.r.t. the previous months. This is the period where the authors of [41] observed a change in Facebook's interface (that led to a drop in the growth of the network) that seems to be correlated to the variation on the motif counts. Even more surprisingly, this aspect is not captured by a static analysis of the snapshots as performed in [41]. Thus, our temporal motifs analysis through ODEN is able to capture a variation in the growth of the network that the static analysis cannot highlight. (We discuss how the motifs and their counts can be used to characterize the activity on the network in Appendix E of [35]).

We then analyzed how the different motif counts of the whole network change by varying the parameter ℓ. We fixed H a triangle and run ODEN with $\varepsilon = 1, \eta = 0.1, \delta = 86400$. The results are shown in Figure (6b). We observe that the counts of $M_1, \ldots, M_{|\mathcal{M}(H,\ell)|}$ vary significantly by increasing ℓ. For $\ell = 3$ almost all the motifs have the same counts, while for larger ℓ there are some motifs with very high counts (i.e., overrepresented) and some other motifs that are underrepresented. Overall the highest counts range from 10^4 to 10^6 from $\ell = 3$ up to $\ell = 6$. To understand if these counts increase only by chance, we performed a widely used statistical test (e..g, [11, 20]) by computing the Z-scores of the different motif counts under the following null model [28]. We generated 500 random networks by the timeline shuffling random model [11], which redistributes all the timestamps by fixing the *directed* projected static network. For each motif $M_i, i = 1, \ldots, |\mathcal{M}(H, \ell)|$ we computed a Z-score that is defined as follows: let C_{M_i} be the count of the motif in the original network and let $C^1_{M_i}, \ldots, C^{500}_{M_i}$ be its counts on the j-th random network $j \in \{1, \ldots, 500\}$. The Z-score is computed as, $Z_{M_i} = (C_{M_i} - \sum_{j=1}^{500} C^j_{M_i}/500)/\mathrm{std}(C^1_{M_i}, \ldots, C^{500}_{M_i})$ where $\mathrm{std}(\cdot)$ denotes the standard deviation. The results are in Fig. (6c), and they show that the counts in Fig. (6b) are very significant and not due to random fluctuations (higher Z-scores indicate that such motif counts are significantly more frequent in T than in the networks permutated randomly). Interestingly, the Z-scores in Figure (6c) follow a similar law to the counts in Figure (6b), with the highest Z-scores increasing significantly every time ℓ increases. Notably the highest Z-scores of motifs with $\ell = 6$ are more than 3 orders of magnitude larger than the Z-scores of motifs with $\ell = 3$. (We discuss some of the significant motifs in Appendix E of [35]).



(a) (b) (c)

Figure 6: (6a): Counts of the motifs in $\mathcal{M}(H, 3)$ with H a triangle on each temporal network corresponding to one snapshot in [41]. (6b): Counts on the full Facebook network with varying ℓ. (6c): Z-scores of the motif counts with varying ℓ.

6 CONCLUSIONS

In this work we introduced ODEN, our algorithm to obtain rigorous, high-quality, probabilistic approximations of the counts of multiple motifs with the same static topology in large temporal networks. Our experimental evaluation shows that ODEN allows to analyze several motifs in large networks in a fraction of the time required by state-of-the-art approaches. We believe that our algorithm ODEN will be of practical interest in the analysis of temporal networks, complementing many of the existing tools and helping in understanding complex networked systems and their patterns.

There are several interesting directions for future research, including devising better edge probability distributions for ODEN and choosing such distribution based on the characteristics of the dataset, since different datasets can have very different temporal edges distributions (e.g., with skewed behaviours [36]) and, thus, there may not exist a unique distribution that is effective for all temporal networks. Another direction of future research is the derivation of improved bounds for the number of samples required by ODEN, using for example statistical learning theory concepts, such as pseudodimensions or Rademacher averages.

ACKNOWLEDGMENTS

This work was supported, in part, by MIUR of Italy, under PRIN Project n. 20174LF3T8 AHeAD, and grant L. 232 (Dipartimenti di Eccellenza), and by the U. of Padova project "SID 2020: RATED-X".

REFERENCES

[1] Paolo Bajardi, Alain Barrat, Fabrizio Natale, Lara Savini, and Vittoria Colizza. 2011. Dynamical Patterns of Cattle Trade Movements. *PLoS ONE* 6, 5 (may 2011), e19869. https://doi.org/10.1371/journal.pone.0019869

[2] V. Batagelj and M. Zaversnik. 2003. An O(m) Algorithm for Cores Decomposition of Networks. *Advances in Data Analysis and Classification, 2011. Volume 5, Number 2, 129-145* (Oct. 2003). arXiv:cs.DS/cs/0310049

[3] Jeffrey Baumes, Mark K. Goldberg, Mukkai S. Krishnamoorthy, Malik Magdon-Ismail, and Nathan Preston. 2005. Finding communities by clustering a graph into overlapping subgraphs. In *AC 2005, Proceedings of the IADIS International Conference on Applied Computing, Algarve, Portugal, February 22-25, 2005, Volume 1*, Nuno Guimarães and Pedro T. Isaías (Eds.). IADIS, 97–104.

[4] Caleb Belth, Xinyi Zheng, and Danai Koutra. 2020. Mining Persistent Activity in Continually Evolving Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. https://doi.org/10.1145/3394486.3403136

[5] George Bennett. 1962. Probability Inequalities for the Sum of Independent Random Variables. *J. Amer. Statist. Assoc.* 57, 297 (mar 1962), 33–45. https://doi.org/10.1080/01621459.1962.10482149

[6] Hanjo D Boekhout, Walter A Kosters, and Frank W Takes. 2019. Efficiently counting complex multilayer temporal motifs in large-scale networks. *Computational Social Networks* 6, 1 (2019), 1–34.

[7] Marco Bressan, Stefano Leucci, and Alessandro Panconesi. 2019. Motivo: Fast Motif Counting via Succinct Color Coding and Adaptive Sampling. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1651–1663. https://doi.org/10.14778/3342263.3342640

[8] Matteo Ceccarello, Carlo Fantozzi, Andrea Pietracaprina, Geppino Pucci, and Fabio Vandin. 2017. Clustering uncertain graphs. *Proceedings of the VLDB Endowment* 11, 4 (dec 2017), 472–484. https://doi.org/10.14778/3186728.3164143

[9] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*. ACM Press. https://doi.org/10.1145/2020408.2020579

[10] Ying Ding. 2011. Scientific collaboration and endorsement: Network analysis of coauthorship and citation networks. *Journal of Informetrics* 5, 1 (jan 2011), 187–203. https://doi.org/10.1016/j.joi.2010.10.008

[11] Laetitia Gauvin, Mathieu Génois, Márton Karsai, Mikko Kivelä, Taro Takaguchi, Eugenio Valdano, and Christian L. Vestergaard. 2018. Randomized reference models for temporal networks. (June 2018). arXiv:physics.soc-ph/1806.04032

[12] M. Girvan and M. E. J. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99, 12 (jun 2002), 7821–7826. https://doi.org/10.1073/pnas.122653799

[13] Saket Gurukar, Sayan Ranu, and Balaraman Ravindran. 2015. COMMIT. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM. https://doi.org/10.1145/2723372.2737791

[14] Petter Holme and Jari Saramäki. 2012. Temporal networks. *Physics Reports* 519, 3 (oct 2012), 97–125. https://doi.org/10.1016/j.physrep.2012.03.001

[15] Petter Holme and Jari Saramäki (Eds.). 2019. *Temporal Network Theory*. Springer International Publishing. https://doi.org/10.1007/978-3-030-23495-9

[16] Y. Hulovatyy, H. Chen, and T. Milenković. 2015. Exploring the structure and function of temporal networks with dynamic graphlets. *Bioinformatics* 31, 12 (jun 2015), i171–i180. https://doi.org/10.1093/bioinformatics/btv227

[17] Ali Jazayeri and Christopher C Yang. 2020. Motif discovery algorithms in static and temporal networks: A survey. *Journal of Complex Networks* 8, 4 (aug 2020). https://doi.org/10.1093/comnet/cnaa031

[18] Chrysanthi Kosyfaki, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. 2018. Flow Motifs in Interaction Networks. (Oct. 2018). arXiv:cs.SI/1810.08408

[19] 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 (nov 2011), P11005. https://doi.org/10.1088/1742-5468/2011/11/p11005

[20] L. Kovanen, K. Kaski, J. Kertesz, and J. Saramaki. 2013. Temporal motifs reveal homophily, gender-specific patterns, and group talk in call sequences. *Proceedings of the National Academy of Sciences* 110, 45 (oct 2013), 18070–18075. https://doi.org/10.1073/pnas.1307941110

[21] Rohit Kumar and Toon Calders. 2018. 2SCENT. *Proceedings of the VLDB Endowment* 11, 11 (jul 2018), 1441–1453. https://doi.org/10.14778/3236187.3269460

[22] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. 2006. Structure and evolution of online social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*. ACM Press. https://doi.org/10.1145/1150402.1150476

[23] Paul Liu, Austin R. Benson, and Moses Charikar. 2019. Sampling Methods for Counting Temporal Motifs. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (Melbourne VIC, Australia) *(WSDM '19)*. Association for Computing Machinery, New York, NY, USA, 294–302. https://doi.org/10.1145/3289600.3290988

[24] Patrick Mackey, Katherine Porterfield, Erin Fitzhenry, Sutanay Choudhury, and George Chin Jr. 2018. A Chronological Edge-Driven Approach to Temporal Subgraph Isomorphism. (Jan. 2018). arXiv:cs.DS/1801.08098

[25] Naoki Masuda and Renaud Lambiotte. 2016. *A Guide to Temporal Networks*. WORLD SCIENTIFIC (EUROPE). https://doi.org/10.1142/q0033

[26] Tijana Milenković and Nataša Pržulj. 2008. Uncovering Biological Network Function via Graphlet Degree Signatures. *Cancer Informatics* 6 (jan 2008), CIN.S680. https://doi.org/10.4137/cin.s680

[27] R. Milo. 2002. Network Motifs: Simple Building Blocks of Complex Networks. *Science* 298, 5594 (oct 2002), 824–827. https://doi.org/10.1126/science.298.5594.824

[28] R. Milo. 2004. Superfamilies of Evolved and Designed Networks. *Science* 303, 5663 (mar 2004), 1538–1542. https://doi.org/10.1126/science.1089167

[29] Mark Newman. 2010. *Networks*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199206650.001.0001

[30] Pietro Panzarasa, Tore Opsahl, and Kathleen M. Carley. 2009. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *Journal of the American Society for Information Science and Technology* 60, 5 (may 2009), 911–932. https://doi.org/10.1002/asi.21015

[31] Ashwin Paranjape, Austin R Benson, and Jure Leskovec. 2017. Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 601–610.

[32] N. Przulj. 2007. Biological network comparison using graphlet degree distribution. *Bioinformatics* 23, 2 (jan 2007), e177–e183. https://doi.org/10.1093/bioinformatics/btl301

[33] Pedro Ribeiro, Pedro Paredes, Miguel EP Silva, David Aparicio, and Fernando Silva. 2019. A survey on subgraph counting: concepts, algorithms and applications to network motifs and graphlets. *arXiv preprint arXiv:1910.13011* (2019).

[34] Ryan A. Rossi, Nesreen K. Ahmed, Aldo Carranza, David Arbour, Anup Rao, Sungchul Kim, and Eunyee Koh. 2021. Heterogeneous Graphlets. *ACM Transactions on Knowledge Discovery from Data* 15, 1 (jan 2021), 1–43. https://doi.org/10.1145/3418773

[35] Ilie Sarpe and Fabio Vandin. 2021. odeN: Simultaneous Approximation of Multiple Motif Counts in Large Temporal Networks (Extended Version). *arXiv* (2021). https://arxiv.org/abs/2108.08734

[36] Ilie Sarpe and Fabio Vandin. 2021. PRESTO: Simple and Scalable Sampling Techniques for the Rigorous Approximation of Temporal Motif Counts. *SIAM International Conference on Data Mining* (2021). https://doi.org/10.1137/1.9781611976700.17

[37] Alice C. Schwarze and Mason A. Porter. 2020. Motifs for processes on networks. (July 2020). arXiv:physics.soc-ph/2007.07447

[38] Shai S. Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. 2002. Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics* 31, 1 (apr 2002), 64–68. https://doi.org/10.1038/ng881

[39] Nino Shervashidze, S. V. N. Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten M. Borgwardt. 2009. Efficient graphlet kernels for large graph comparison. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009 (JMLR Proceedings)*, David A. Van Dyk and Max Welling (Eds.), Vol. 5. JMLR.org, 488–495. http://proceedings.mlr.press/v5/shervashidze09a.html

[40] Kun Tu, Jian Li, Don Towsley, Dave Braines, and Liam D. Turner. 2019. gl2vec. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM. https://doi.org/10.1145/3341161.3342908

[41] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. 2009. On the evolution of user interaction in Facebook. In *Proceedings of the 2nd ACM workshop on Online social networks - WOSN '09*. ACM Press. https://doi.org/10.1145/1592665.1592675

[42] Jingjing Wang, Yanhao Wang, Wenjun Jiang, Yuchen Li, and Kian-Lee Tan. 2020. Efficient Sampling Algorithms for Approximate Temporal Motif Counting. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. ACM. https://doi.org/10.1145/3340531.3411862

[43] Shuo Yu, Yufan Feng, Da Zhang, Hayat Dino Bedru, Bo Xu, and Feng Xia. 2020. Motif discovery in networks: A survey. *Computer Science Review* 37 (2020), 100267.

[44] Qiankun Zhao, Yuan Tian, Qi He, Nuria Oliver, Ruoming Jin, and Wang-Chien Lee. 2010. Communication motifs. In *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10*. ACM Press. https://doi.org/10.1145/1871437.1871694

[45] Bo Zong, Xusheng Xiao, Zhichun Li, Zhenyu Wu, Zhiyun Qian, Xifeng Yan, Ambuj K. Singh, and Guofei Jiang. 2015. Behavior query discovery in system-generated temporal graphs. *Proceedings of the VLDB Endowment* 9, 4 (dec 2015), 240–251. https://doi.org/10.14778/2856318.2856320