

Advances in Complex Systems
 © World Scientific Publishing Company

Event Graphs: Advances and Applications of Second-Order Time-Unfolded Temporal Network Models

ANDREW MELLOR

Andrew Mellor

Mathematical Institute, University of Oxford
mellor@maths.ox.ac.uk

Received (received date)

Revised (revised date)

Recent advances in data collection and storage have allowed both researchers and industry alike to collect data in real time. Much of this data comes in the form of ‘events’, or timestamped interactions, such as email and social media posts, website clickstreams, or protein-protein interactions. This type of data poses new challenges for modelling, especially if we wish to preserve all temporal features and structure. We highlight several recent approaches in modelling higher-order temporal interaction and bring them together under the umbrella of *event graphs*. Through examples, we demonstrate how event graphs can be used to understand the higher-order topological-temporal structure of temporal networks and capture properties of the network that are unobservable when considering either a static (or time-aggregated) model. We introduce new algorithms for temporal motif enumeration and provide a novel analysis of the communicability centrality for temporal networks. Furthermore, we show that by modelling a temporal network as an event graph our analysis extends easily to non-dyadic interactions, known as hyper-events.

Keywords: temporal networks, higher-order models, hypergraphs, percolation, motifs

1. Introduction

As an abstraction of complex systems, networks have been a fundamental tool for research across many disciplines. Typically network models assume that the relationships between objects (nodes) are static or unchanging. The static network assumption is usually made for either ease of analysis, or simply because of the lack of temporal data. This is particularly the case in the biological sciences where data collection can be both difficult and costly.

In recent years our capacity to collect data has improved drastically, especially in the digital domain where many systems can be monitored autonomously. This means that for many applications individual timestamped interactions between objects (also called *events*) are recorded. In social networks, events may take the form of messages between users, or in biology, an event may be the interaction between two proteins at a certain time. In some cases, the events are not timestamped but are temporally ordered. This type of data falls under the wider category of sequential

data. The modelling and analysis of these timestamped events (or sequential data) has become a vital task with applications in data mining, social network analysis, and computational biology.

Despite static networks providing a useful abstraction of complex systems many of these systems themselves are not stationary and the interactions between objects change over time or appear only in a particular order [21]. Systems that evolve over time can instead be modelled as a *temporal network*. While most systems exhibit at least some temporal dependency there are relatively few tools to capture their underlying topological-temporal behaviour [34, 21, 20]. There are many ways to represent temporal networks however we focus on the event-based representation [23]. In this representation, a temporal network is described by a time-ordered sequence of temporal events $(e_i)_{i=1}^m$ where each event is of the form $e_i = (u_i, v_i, t_i, \delta_i)$. Here u_i and v_i are the interacting node pair of the i th event, which occurs at time t_i , and lasts for a duration δ_i . If the network is directed this is interpreted as a directed event from u_i to v_i . The duration of events can be omitted or set to zero for instantaneous interaction.

Studies of temporal networks typically revolve around the timings between events occurring, or the *inter-event time* [40]. How these times are distributed have been shown to have an effect on dynamics evolving across the system [32, 33]. Other studies have investigated the prevalence of temporal motifs in these systems [26, 27, 43]. Network motifs are small repeated patterns of interaction in the network and their application in the temporal setting has perhaps shown the most promise in characterising behaviour in temporal networks.

In this article we provide a perspective on previous approaches to event modelling, showing how they can be formulated and studied as event graphs and illustrating where event-based methods can be applied. The contribution of this paper is to show that under the formulation of event graphs, higher-order models of events are easily extended and applied. We show this by a novel extension to include hyper-events (events with non-dyadic interactions), a new (and efficient) algorithm for motif enumeration, and a new centrality measure on events which in turn offers more detailed insight to node-based centrality.

Time-unfolded Graphical Models

Where temporal networks differ most from their static counterparts is the concept of a time-respecting path. Figure 1(left) shows an example temporal network where edges are labelled with the time at which they occur. If we consider only the induced aggregated static network we would observe a path from node E to node B, however no such path exists in the real network. By simply ignoring the ordering of events we potentially exaggerate the number of possible paths through the network. Naturally, this can lead to overestimation when considering the proliferation of a spreading dynamic on the network.

One way to preserve the temporal ordering of paths is by considering time-

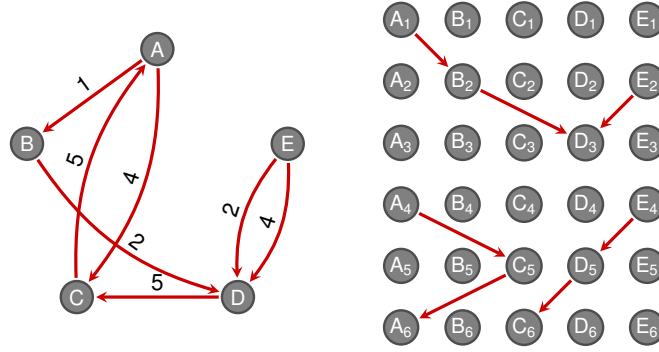


Fig. 1: An example temporal network (left) and the corresponding time-unfolded model of the network (right). Edges in the temporal network are labelled with the time at which they appear (in this case, instantaneously). In the time-unfolded model, nodes are replaced by time-indexed copies of themselves. Edges are then created between a time-indexed node and another time-indexed node that is strictly further ahead in time. Nodes can be connected to nodes in the next time interval for a discrete time model of instantaneous contact (as presented here), or to nodes at some later point in time for contacts with durations.

unfolded (or time-node) graph [24, 41, 38, 54]. Here nodes are replaced by time-indexed copies of themselves and edges can only travel from older to newer time-indexed nodes. The utility of these graphs are that they are amenable to traditional static network methods and can be efficient to work with as they are both directed and acyclic. Figure 1(right) shows the corresponding time-unfolded model for the temporal network. Using this model it is easy to confirm that there is no temporal path from E to B.

Higher-order Graphical Models

The concept of a second-order model has been around for a long time in the form of line graphs [19], De Bruijn graphs [10], or Hashimoto graphs [18]. In this class of model, the edges of the original static network become the nodes of the new model, and edges are connected to other edges if they have a node in common. In the temporal network literature, these models have been referred to as *memory networks* [30, 46], owing to their use of modelling second-order Markov random walks, or random walks with *memory*. Models which incorporate memory are particularly useful for considering walks on temporal networks. As an example, consider the temporal network of individual journeys across the London tube network. These journeys are the result of many commuters travelling to and from work and can not be well modelled by a random walk on the underlying static network [29]. Upon arrival at a particular station, the likelihood of your next destination depends on,

at the very minimum, of where you were last, e.g. you would be very unlikely to back-track unless you were particularly lost! This type of process can however be captured with a higher-order model [49].

Figure 2 shows an example of a first- and second-order model for a random walk process on a network. Edge weights represent the probability of traversing from state to state. In the first-order model the likelihood of the traversal of an edge is dependent only on the current node. In contrast, the probability of traversing the $D \rightarrow B$ edge in the second-order model depends whether we have previously arrived at D from C or A (probabilities 0.4 and 0.5 respectively).

Beyond second-order Markov models there have been recent works which consider more general higher- or variable-order models [51, 50, 49]. Unfortunately, the number of possible states increases exponentially with the chosen model order, which can lead to computational intractability. This issue can be remedied to some extent by considering higher-order states as and when a lower-order representation is insufficient to describe the data [56].

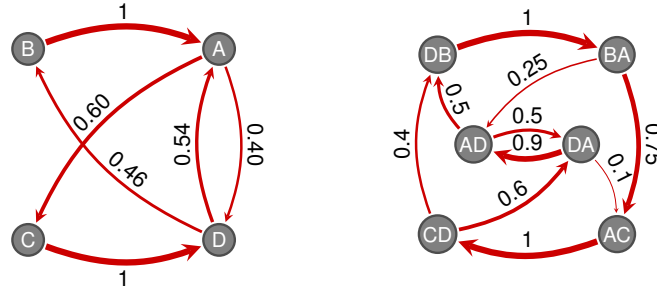


Fig. 2: An example of a first-order (left), and second-order (right) graphical model. In each case, a random walker can transition between states with some probability (indicated by the edge weights). In the first-order model, the probability of transition between nodes is conditional only on the current node whereas in the second-order model the transition is dependent on both the current and previous node. For example, $P(X_{t+1} = D|X_t = A, X_{t-1} = Z) = 0.4$ for any node Z , whereas $P(X_{t+1} = D|X_t = A, X_{t-1} = B) = 0.25$ and $P(X_{t+1} = D|X_t = A, X_{t-1} = D) = 0.9$.

Finally, it is worth remarking that ‘higher-order’ interaction has a secondary meaning beyond higher-order Markov models. The models outlined above have all been based on the assumption of pairwise (or dyadic) interaction. However much like the static network assumption, the idea of dyadic interaction is often more convenient than it is valid. The concept of hypergraphs [5], where interactions (edges) can include multiple nodes, and the corresponding directed hypergraphs [11] have been around for multiple decades and have been well studied. More recently how-

ever there has been work studying non-dyadic interaction over time through both topological perspectives, in simplicies and simplicial complexes [3], or by considering the structure of a growing and evolving hypergraph [55].

Contributions

There are many ways to represent temporal networks [21], depending on modelling assumptions, the flavour of analysis undertaken, and the availability of data. Stream graphs and link streams model the persistence of the nodes and edges of a graph [31]. Similarly, time-varying graphs [8] model the availability of edges for transport using a presence indicator function. Algebraic approaches to edge persistence are given in [2]. In contrast, event graphs model the sequence of events themselves. This representation is particularly useful in the context of processing streams of interaction data given that data often is in the form of events (x messaged y , or x bought item y at time t , for example). A statistical and generative model for events is presented in the relational event framework [7], developed for the social sciences. This model fits parameters controlling several network mechanisms (e.g. preferential attachment, triadic closure, two-event motifs) and derives the likelihood of observing a particular event sequence. We distinguish ourselves from this work by focusing on the structure of event graphs and their applications without specifying an underlying model for their synthesis.

In this article we generalise the concept of an event graph and show how the representation offers a flexible approach to addressing temporal networks analysis. This representation allows for events to be connected using different connection rules, depending on the questions asked of the network. The resulting event graph generated is always a directed acyclic graph. As a result, the event graph model offers efficient algorithms to calculate network properties and avoids issues such as choosing suitable time windows to aggregate temporal network data, as is often done in temporal network analysis. To illustrate how event graphs can be used we provide a number of example applications to a selection of temporal networks. These examples include:

- (1) The unification of two motif definitions as subgraphs of an appropriate event graph, and novel (and efficient) methods for the enumeration of temporal motifs from data.
- (2) A second-order breakdown of node centrality (in the form of communicability [15]) through consideration of event centralities. This new perspective can distinguish *why* nodes are central, rather than simply *if* they are central.
- (3) Extensions of percolation analysis in temporal networks to include non-dyadic interactions.

We show that the event graph also provides a natural way to model non-dyadic temporal interactions between nodes allowing us to derive non-dyadic extensions to several network properties. To the author's knowledge, this is the first model for

sequences of non-dyadic events at a network level (sequences of events have been modelled for an individual however [4]).

In Section 2 we introduce event graphs and in Section 3 we show how they can be generalised to include non-dyadic interaction in the form of hyper-events. In Section 4 we give example applications of event graphs (both novel examples and examples taken from previous works) before concluding in Section 5.

2. Event Graphs

An event graph combines both the notion of a higher-order model (in particular a *second-order* model) with that of a time-unfolded model. We consider a temporal network defined by a sequence of temporal events $(e_i)_{i=1}^m$ where each event is a triplet of the form $e_i = (u_i, v_i, t_i)$ (as described in Section 1). For the purpose of generality we remain ambiguous as to whether this represents a directed event $u_i \rightarrow v_i$ or an undirected event $u_i \leftrightarrow v_i$, although we clarify when needed.

We assume that a node may participate in only one event at a time (although multiple events can occur at the same time if independent). We can remove this restriction by merging overlapping events into a *hyper-event*, as we show in Section 3. However, if the time resolution of the data is low and a node appears in a large number of events at the same time then an event-based representation may be unsuitable and a sequence of static graphs may be appropriate.

For any two events we can compare the node sets $\{u_i, v_i\}$ and $\{u_j, v_j\}$, and the time between the two events occurring, i.e. the *inter-event time*.

Definition 1. (Inter-event Time (IET)) *The inter-event time τ between two events $e_i = (u_i, v_i, t_i)$ and $e_j = (u_j, v_j, t_j)$ is given by*

$$\tau(e_i, e_j) = \begin{cases} t_j - t_i & \text{if } t_j > t_i \\ 0 & \text{otherwise.} \end{cases}$$

If the events are not instantaneous and have durations δ_i and δ_j respectively then

$$\tau(e_i, e_j) = \begin{cases} t_j - (t_i + \delta_i) & \text{if } t_j > t_i + \delta_i \\ 0 & \text{otherwise.} \end{cases}$$

For simplicity of notation we write $\tau_{ij} = \tau(e_i, e_j)$.

We now define an event graph in full generality before describing particular examples.

Definition 2. (Event Graph) *An event graph G is a directed static graph given by the $G = (V, T, E, L)$ where E is a set of temporal events and $L \subseteq E \times E$ is a set of directed edges between events.*

This defines a family of possible event graphs for a set of events E dependent on the edge set L . Here we consider $E \subseteq V^2 \times T$, with V being the set of nodes and T being the set of event times. Furthermore, it is often useful to consider a weighted event

graph G^τ where the edges of the event graph are weighted by the inter-event times between events. The weighted event graph provides a means to threshold edges and investigate the percolation properties (Section 4).

Let $X \in \mathbb{B}^{|V| \times |E|}$ be the node-event incidence matrix, where $\mathbb{B} = \{0, 1\}$, and $X_{ij} = 1$ if node i is in event j , and is zero otherwise. If events are directed we can define two incidence matrices $X_s, X_t \in \mathbb{B}$ where $(X_s)_{ij} = 1$ if node i is a *source* of event j and zero otherwise. Similarly $(X_t)_{ij} = 1$ if node i is a *target* of event j and zero otherwise. An unweighted event graph can be represented by an event adjacency matrix $M \in \mathbb{B}^{|E| \times |E|}$, with $M_{ij} = 1$ if $(i, j) \in L$ and zero otherwise. We can recover the adjacency matrix (aggregated over time) by considering $A = XX^\top - D$, where D is the diagonal matrix of node degrees. Furthermore, $(XM^\top X)_{ij}$ counts the number of walks of length two from i to j and hence $(XM^k X^\top)_{ij}$ counts the number of walks of length $k + 1$. Note that these quantities can be calculated without first discretising time and creating a sequence of static adjacency matrices A_1, A_2, \dots .

2.1. Event Connection Rules

An event graph is defined purely through the adjacency matrix M which in turn can be generated through event connection rules. An event graph should be seen as a particular representation of a temporal network that can be used to examine higher-order properties of the network, and choice of event connection function should be dependent on either the type of temporal network or the particular dynamic under study (e.g. contagion or message-passing). Furthermore, the computational cost of constructing an event graph varies strongly with the choice of event connection rule so this should also be considered. The remainder of this section is devoted to exploring particular event connection rules which have recently been proposed. In Figure 3 we give an example of a temporal network and the resulting event graphs under different connection rules.

The simplest non-trivial connection rule is Δt -adjacency, first defined in [26]. Following their definition, two events are Δt -adjacent if they share at least one node, and the time between the two events (IET) is no greater than Δt , for some prescribed $\Delta t \in \mathbb{R}^+$. The adjacency matrix for the Δt -adjacent event graph is given by

$$M_{ij} = \begin{cases} 1 & \text{if } (0 < \tau_{ij} \leq \Delta t) \text{ and } (\{u_i, v_i\} \cap \{u_j, v_j\} \neq \emptyset) \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Adjacency makes intuitive sense assuming that there are no external (or unobserved) interactions between agents; information can only be transmitted between two events if there is one or more common node where that information can persist (although for directed networks two adjacent events do not necessarily form a temporal path). The parameter Δt is an upper bound for how long this information could persist, or how close two events need to be for us to consider a causal relationship. In the limit $\Delta t \rightarrow \infty$ there are no restrictions on temporal proximity,

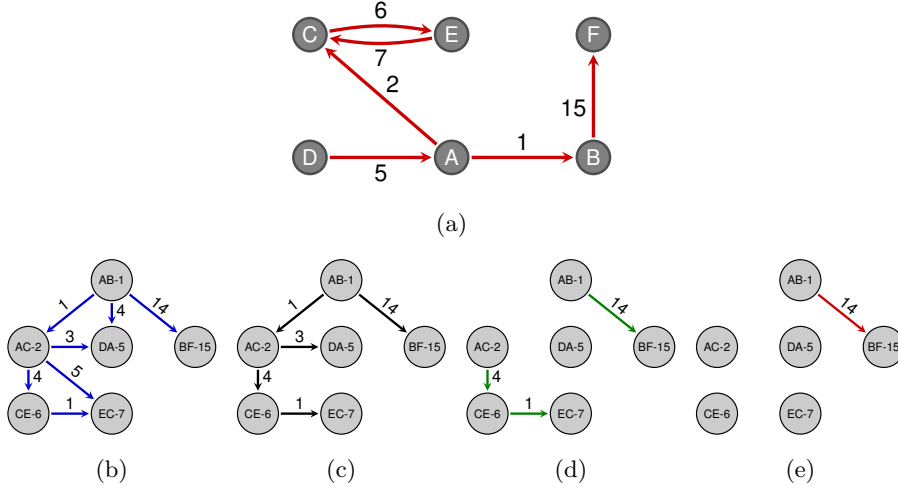


Fig. 3: Event graph examples using multiple event connection rules. (a) An example temporal network where edges are labelled with the time at which they occur. (b) A Δt -adjacent event graph with $\Delta t = 20$. (c) A node-subsequent Δt -adjacent event graph with $\Delta t = 20$. (d) A walk-forming event graph (with no Δt restriction). The prescription of walk-forming events is much stricter than Δt -adjacency hence there are far fewer edges in the event graph. (e) A minimum-gap non-backtracking event graph with $(\Delta t_1, \Delta t_2) = (5, 15)$. Events now have a minimum inter-event time required for them to be connected which results in fewer edges. For the event graphs, edges are weighted by the inter-event time.

and the requirement is only that events are adjacent.

An event can be connected to any number of subsequent events provided the adjacency criteria is met. However, in certain cases, we are interested in only the subsequent event for each node in a given event (as in [26, 37]). Let

$$A^t(S) = \{(u, v, t') \in E \text{ s.t. } S \cap \{u, v\} \neq \emptyset \text{ and } t' > t\}$$

be the set of all events that a set of nodes S participates in after a time t . The *node-subsequent* Δt -adjacent event graph is given by adjacency matrix

$$M_{ij} = \begin{cases} 1 & \text{if } (0 < \tau_{ij} \leq \Delta t) \text{ and } [(j = \min\{k | e_k \in A^{t_i}(\{u_i\})\}) \\ & \text{or } (j = \min\{k | e_k \in A^{t_i}(\{v_i\})\})] \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Note that in this example the construction of each of the adjacency matrix requires knowledge of the set of all events (to pick only the subsequent event). However, if events are added to the graph sequentially then edge additions can be made independently as they are guaranteed to be subsequent events.

We will primarily consider the Δt -adjacent event graph and the node-subsequent variant, however other event graphs have been proposed. Temporal (or time-respecting) paths and walks^a are an important feature of temporal networks as, analogously to paths and walks in static networks, they allow for the dissemination of information through the network. For directed events where an event (u_i, v_i, t_i) represents a directed interaction $u_i \rightarrow v_i$ the event graph which captures all temporal walks is given by

$$M_{ij} = \begin{cases} 1 & \text{if } (0 < \tau_{ij} \leq \Delta t) \text{ and } (u_j = v_i) \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

This states that two events are connected if the target of the first event is the source of the second event.

For certain temporal network processes, it may be suitable to introduce a minimum time between events occurring for them to be considered to be connected. One such example is for aviation networks, i.e. the network of scheduled flights, where it is an unrealistic assumption that a connection can be made at an airport without first taking time to traverse across the airport from one plane to another [12]. This motivates us to introduce a non-zero lower bound on τ_{ij} . Keeping with the air travel example, it would also be pointless to immediately return along the same route that you have just travelled so it would be sensible to remove these backtracking possibilities. Together these rules combine to give an event graph defined by

$$M_{ij} = \begin{cases} 1 & \text{if } (\Delta t_1 < \tau_{ij} \leq \Delta t_2) \text{ and } (u_j = v_i) \text{ and } (v_j \neq u_i), \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

where $\Delta t_1, \Delta t_2 \in \mathbb{R}^+$ and $\Delta t_1 < \Delta t_2$. The non-backtracking event graph draws some similarity to the non-backtracking matrix representation of a static graph [22], or the Hashimoto matrix [18]. Non-backtracking operators have previously been used in the static case for community detection [28] and spectral clustering however their applications in a temporal setting have yet to be fully explored [48, 16].

3. Non-dyadic Interactions

Although many studies of networks (and temporal networks) revolve around the assumption of pairwise interaction between nodes, this assumption often does not hold or is an oversimplification. In networks of academic collaboration, cliques are formed between authors when they publish together. These interactions are non-dyadic, although when abstracting to a network an individual edge between two authors is considered independently of the clique it came from (for publications

^a Temporal paths are distinguished from temporal walks by the number of times the walk/path can visit each node. Temporal paths may only visit each node at most once, whereas temporal walks can visit each node multiple times.

with three or more authors). Email correspondence and other digital communication is also another area where non-dyadic interaction occurs. Emails can be sent to any number of people in a single interaction however this type of behaviour is indistinguishable from multiple person-to-person emails if interactions are aggregated over time.

Higher-order interactions can be captured within an event graph by introducing the notion of *hyper-events* and the *temporal hypergraph*.

Definition 3. (Temporal Hypergraph) Let $V \subseteq \mathbb{N}$ be a set of nodes, $T \subseteq \mathbb{R}_0^+$ a set of times, $D \subseteq \mathbb{R}_0^+$ a set of durations, and E a set of temporal hyper-events. A temporal hypergraph is defined by the quadruple $G = (V, T, D, E)$ where temporal hyper-events take the form

$$e_i = (U_i, t_i, \delta_i)$$

where $U_i \subseteq V$. For directed temporal hypergraphs hyper-events take the form

$$e_i = (U_i, V_i, t_i, \delta_i)$$

where $U_i, V_i \subseteq V$ and $U_i \cap V_i = \emptyset$ analogously to non-temporal hypergraphs.

If we fix $|U_i| = k$ for all i we call the resulting undirected hypergraph *k-uniform*. Similarly, for directed events if we restrict $|U_i| = k$ and $|V_i| = m$ for all i we have a (k, m) -uniform hypergraph.

This generalises the concept of an event sequence and hence extends temporal networks to include non-dyadic interactions. If we consider a 2-uniform (or (1, 1)-uniform) hypergraph we see that we arrive back at the familiar undirected (and directed) event sequences of the previous section. As before we assume that a node may be involved with at most one event at any given time, although now a node may interact with multiple other nodes during the event. The definition of an event graph for temporal hypergraphs is identical to the definition of dyadic interaction (Definition 2).

3.1. Event Connection Rules

While the definition of an event graph is unchanged for non-dyadic interaction the event connection rules for hyper-events require further generalisation.

Δt -Adjacency. For two hyper-events to be Δt -adjacent requires that one or more of the nodes in the first event (source or target) appear in the second event. For directed events, the Δt -adjacent event graph is given by

$$M_{ij} = \begin{cases} 1 & \text{if } (0 < \tau_{ij} \leq \Delta t) \text{ and } ((U_1 \cup V_1) \cap (U_2 \cup V_2) \neq \emptyset) \\ 0 & \text{otherwise.} \end{cases}$$

For undirected events the topological requirement is $U_1 \cap U_2 \neq \emptyset$. Node-subsequent adjacency can also be trivially defined for hyper-events.

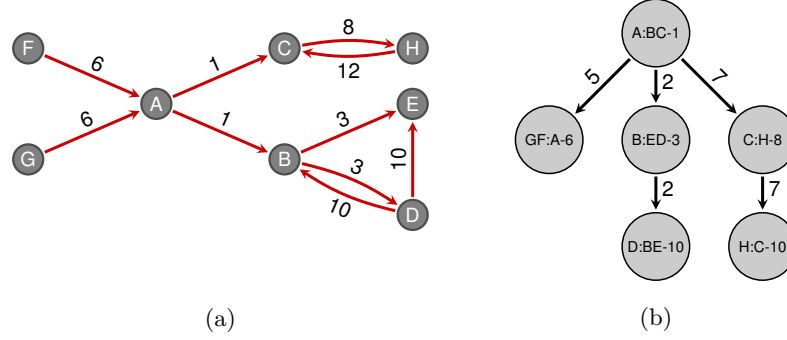


Fig. 4: Event graph construction for temporal hypergraphs. In (a) we have an example temporal hypergraph where an event can include more than two nodes, represented here by the identical timestamp on ingoing or outgoing edges. In (b) we have the corresponding node-subsequent Δt -adjacent event graph. Events are labelled as [Source Nodes]:[Target Nodes]-[Time] and edges are weighted by the IET as before.

Walk-forming. Walk-forming carries the stricter topological restriction of $(V_1 \cap U_2 \neq \emptyset)$ in comparison to Δt -adjacency. For undirected hyper-events all Δt -adjacent events are walk-forming.

4. Applications

In this section we explore some of the analysis of temporal networks that has been conducted with event graphs and highlight new avenues for further research through examples.

Throughout this section we will use several datasets (both real and synthetic) which are detailed in Table 1. These cover a range of durations (from 1 day to 29 years) and varying numbers of nodes and events. We have also included a mixture of directed and undirected networks and graphs with non-dyadic interactions. This data is readily available online at the respective references, aside from [47] which is available upon reasonable request. The **sociopatterns-primary** network in its original form is dyadic. We generate a sequence of hyper-events by first aggregating all nodes which are connected (not necessarily pairwise connected) at the same time, and then check for the persistence of this set across time to create an event duration. A new event is therefore created each time a node enters or leaves the node set. The **random-complete** network is generated synthetically in an iterative fashion. At each timestep, two nodes are picked at random with one chosen randomly to be the source, and the other the target. Time is advanced by a random increment drawn from an exponential distribution with mean one. This is equivalent to randomly sampling edges from a complete graph.

Table 1: Data descriptions.

Name	Nodes	Events	Duration	Hyp.	Dir.	Description
twitter-emirates[35]	53,251	167,664	1 day	Yes	Yes	Posts taken from Twitter using the keyword 'Emirates.'
random-complete	500	50,000	50,163	No	Yes	A random network where at each iteration a random node is connected to another random node.
academic-coauthors[47]	30,927	54,177	29 years	Yes	No	A network of article coauthorship taken from ArXiv (<code>math</code> and <code>math-ph</code>).
social-ucirvine[42]	1,899	59,835	193 days	No	Yes	An online social network from UC Irvine.
sociopatterns-primary[53, 52]	242	38,923	2 days	Yes	No	A proximity network of school children.

We use this array of data to show that methods on event graphs are largely agnostic to the type of event and so these temporal networks can be compared, regardless of the event type.

4.1. Motif Counting

Motifs are crucial in helping to understand the structure and function of networks and have seen applications across a number of fields, in particular in the biological sciences [1, 39, 45]. Motifs in temporal networks are however less well understood, and there is not a unified definition of a temporal motif. Certain definitions consider the underlying induced static graph (ignoring temporal ordering) [57], or use heuristics for motif counting [17]. We give the definitions of two types of temporal motif which we name the *windowed temporal motif* [43] and *sequential temporal motif* [26] for clarity, and show that they both can be described by subgraphs of the event graph.

Definition 4. (Windowed temporal motif [43]) *A k -node, l -event, δ -temporal motif is a sequence of l time-ordered events (e_1, \dots, e_l) such that $t_l - t_1 \leq \delta$ and the induced aggregate network has k nodes.*

Definition 5. (Sequential temporal motif [26]) *An l -event Δt -temporal motif is a graph of l events where all pairs (e_i, e_j) of events are Δt -connected, that is there exists a sequence of events $(e_n)_{n=1}^k$ such that $e_i = e_1, \dots, e_k = e_j$ and all pairs of consecutive events are Δt -adjacent. Furthermore, it is required that all events for a node are consecutive (i.e. no events omitted).*

These definitions differ in two ways. Firstly as the name suggests, the windowed motif has to occur strictly within a time window of length δ , whereas the sequential motif requires only that there is a chain of events whose inter-event times are no less than Δt . This can result in motifs with a duration much greater than Δt . Secondly, sequential motifs require that all events of each node in the motif are consecutive, that is, if a node is involved in events e_i, e_j, e_k (with $t_i < t_j < t_k$) then events e_i and e_k cannot form a motif since e_j is omitted. This reduces the number of possible

motifs drastically, however the output is more susceptible to noise in the data as a ‘noisy’ intermediate event may obscure a ‘true’ motif.

As an aside, there also exist statistical models of motif generation, such as those in the relational event model [7] and other two-event motif models [6]. These generative models have rules for the appearance of two-event motifs (such as the probability of an event causing a motif-forming event decaying geometrically in time), however, the output of these models can still be represented by windowed temporal motifs with $\delta \rightarrow \infty$. By contrast, the definitions of motifs here are only to enumerate motifs from empirical data, regardless of how they were generated.

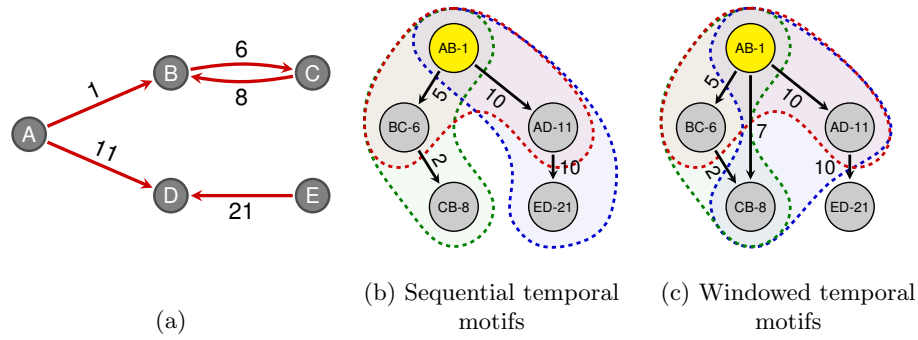


Fig. 5: Schematics for the calculation of temporal motifs using an event graph. In (a) we have an example temporal network, where edges are labelled at the time they occur. For (b) we construct the node-sequential Δt -adjacent event graph with $\Delta t = 10$. Sequential motifs (as described in [26]) can be found by either breadth-first or depth-first search from a node ensuring that any edge has a weight of 10 or less. In this case, there are three possible 3-event motifs which begin with the event $A \rightarrow B$ at time 1. For (c) we construct the Δt -adjacent event graph with $\Delta t = 10$. We can then find windowed temporal motifs (as described in [43]) in the same fashion as (a), except this time the maximum path length cannot exceed 10. This also results in three 3-event motifs, however only two motifs are shared between definitions.

Figure 5 shows how the motifs of a temporal network (a) can be represented as subgraphs of the node-subsequent Δt -adjacent event graph (b) and Δt -adjacent event graph (c). Starting from a given initial event (here highlighted in yellow), the associated 3-event motifs can be found by traversing the edges of the event graph. Sequential Δt -temporal motifs can be found by traversing edges whose weight is less than Δt (or the event graph can be pre-pruned for efficiency). Motifs generated in this fashion require a further check to confirm that all events for participant nodes are included if they fall within the timeframe of the motif. This is referred to as a *valid* temporal motif [26]. Windowed δ -temporal motifs require edges to be traversed ensuring that the maximum path length does not exceed δ . This is why

$(A,B,1),(A,D,11),(E,D,21)$ is not a valid windowed δ -temporal motif with $\delta = 10$.

The reasoning above suggests that temporal motifs can be universally defined as subgraphs of an appropriate event graph. This formalism means that we can naturally extend temporal motifs to non-dyadic interactions since event graphs are defined for hyper-events. This does, however, add to the issue of the explosion in the number of possible motifs as the number of events increases. If the size of hyper-events are unrestricted then motif counting becomes impractical and meaningless. One solution to this issue is instead to consider only whether new nodes are added to a previous event, old nodes are removed, or a mixture of both.

The event graph framework provides a unifying view on temporal motifs and it is also a computationally efficient means to count them. A Δt -adjacent event graph can be constructed from an ordered sequence of M events in $\mathcal{O}(M)$ time as each event can be added to the graph by making comparisons to the previous k events where $k = \lambda \Delta t \ll M$ and λ is the rate of events generated per unit time. If we consider a node-subsequent Δt -adjacent event graph (as in [26]) this reduces to an $\mathcal{O}(1)$ lookup.

Finding all motifs of all sizes in a temporal network is equivalent to enumerating all subgraphs of the event graph and is therefore intractable. However, progress can be made by exploiting that the event graph is a directed acyclic graph and using a dynamic programming approach to find motifs up to a fixed size [44, 26]. More efficient methods exist for particular motifs, however. As each motif can be represented by a (non-unique) subgraph of the event graph, this reduces the task to finding specific subgraphs of the event graph. The subgraph representation of a motif becomes unique in the combination with event relation data, i.e. the motif formed by two events.

To demonstrate we give an example of finding 3-node, 3-event stars and triangles. All of these motifs share the same subgraph given in Figure 6 (a proof is given in Appendix B). To find such subgraphs we can:

- (1) Pick a root event e_r .
 - (2) Find a child event $e_c \in \text{Children}(e_r)$ such that $S := \text{Children}(e_r) \cap \text{Children}(e_c) \neq \emptyset$.
- A motif is then formed from (e_r, e_c, e_s) for each $e_s \in S$.

For each root this is an $\mathcal{O}(k^2)$ operation and so the procedure of locating all these subgraphs is $\mathcal{O}(mk^2)$. Finally, each subgraph can be converted into a motif by considering the relationship between pairs of events in the subgraph. Two such motifs are shown in Figure 6 (there are 36 in total). This is an $\mathcal{O}(1)$ operation. Including the construction of the event graph, these motifs can be found in $\mathcal{O}(m(k + k^2))$ time, where k is constant. This matches the optimal result of [43] with $\mathcal{O}(m)$ complexity up to constant factors.

Using the event graph method has two major benefits over the counting methods of [43]. Firstly, by keeping track of all root nodes we can enumerate the motifs,

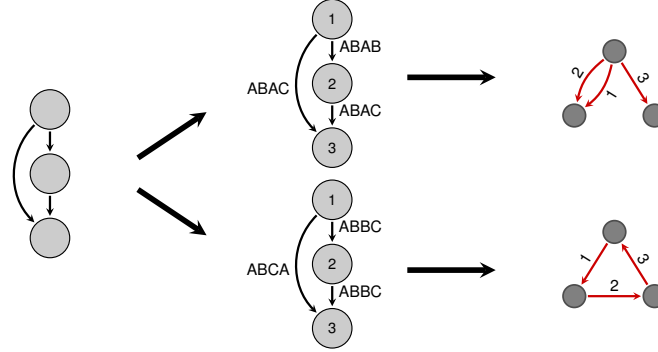


Fig. 6: (Left) The subgraph representing all 3-node, 3-event triangle and star motifs. (Middle) The subgraph annotated with two example pairwise relationships between events. As an example, a pair of repeated events is represented by ABAB, and two events originating from the same node is represented by ABAC. Note that only two relationships need to be found and the third is implied by the subgraph structure. (Right) The corresponding temporal motifs

meaning that we know exactly where and when within the temporal network they occur. This means that motif prevalence can be tracked over time, or the motif participation of individual nodes can be considered. Secondly, the event graph need only be constructed once, allowing further motifs to be calculated or further analysis to be conducted.

4.2. Event Centrality

Much like edge centralities for static networks (e.g. edge betweenness [13]) we can define new centralities for temporal events. Event centralities can be used to understand the importance of an event in terms of its ability reach to nodes and events further forward in time, or how well it creates a bridge between otherwise disjoint sets of nodes. While there are many possibilities for defining centrality, here we consider an application of the communicability centrality [15, 14] to temporal events. We show that not only does event centrality provide a means to rank events based on their reach to other events, but also that it uncovers patterns in node communicability which were previously hidden.

The dynamic communicability matrix [15] is defined at a time t by

$$Q(t) = \prod_{s=0}^t [I - \alpha A(s)]^{-1},$$

where $A(s)$ is the (node) adjacency matrix at time interval s , and I is the identity. This definition is written in terms of a sequence of adjacency matrices, however, this can be reconciled with an event-based representation by considering a time interval

such that each event belongs to its own interval. The communicability matrix Q counts all weighted temporal walks from one node to another, where the traversal of each edge is given a multiplicative weight $0 < \alpha < 1$. Temporal walks can also be weighted by their age through the *running* dynamic communicability matrix $S(t)$, defined iteratively by

$$S(t) = [I + e^{-\beta\delta t} S(t-1)] [I - \alpha A(t)]^{-1} - I,$$

where β is the temporal downweighting factor, and δt is the time between intervals.

Now consider a directed temporal network of events $(e_i)_{i=1}^m$ with $e_i = (u_i, v_i, t_i)$, consisting of n nodes, and modelled by the walk-forming event graph

$$M_{ij} = \begin{cases} 1 & \text{if } (\tau_{ij} > 0) \text{ and } (v_i = u_j) \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Here we have taken $\Delta t \rightarrow \infty$ to avoid the temporal restriction on connected events. The event communicability matrix is given by $Q_E = (I - \alpha M)^{-1}$. Furthermore, we can incorporate the temporal decay of walks by considering a weighted event graph of the form

$$M_{ij}^* = \begin{cases} \exp(-\beta\tau_{ij}) & \text{if } (\tau_{ij} > 0) \text{ and } (v_i = u_j) \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The event communicability matrix is constructed as before using M^* instead.

The sum of weighted walks start at each event (referred to as the broadcast centrality) is given by $\mathbf{b} = Q_E \mathbf{1}_m$ where $\mathbf{1}_s = (1, \dots, 1)$ is a vector of dimension s . For temporal decay we can consider two centralities, either by the right multiplication of M^* by $\mathbf{1}_m$ or by $d(T)$ where

$$d_i(T) = e^{-\beta(T-e_i)}.$$

The former downweights walks in time only for the duration of the walk, while the latter downweights walks in time even after they have terminated (which aligns with the running dynamic communicability). We may interpret this difference as “how much information has been transferred from event to event, assuming loss over time and at each transmission?”, in contrast to the running dynamic communicability which is “how much information has been transferred from event to event recently?”. Finally, we can recover the dynamic communicability matrix as $Q = X_s(\alpha Q_E)X_t^\top$ where X_s and X_t are the node-event-source and node-event-target incidence matrices defined in Section 2 (a sketch proof is given in Appendix A). Writing the broadcast centrality for nodes as $\mathbf{b} = X_s Q_E \mathbf{1}_n$ shows that a node is only as central as the events it participates in. There is therefore extra information in this representation as we can consider the distribution of event centralities for which a node participates as well as its overall centrality.

The need to consider event centrality can be seen with a simple example. Consider a random temporal network of n nodes that is generated in discrete time. At each iteration t , a source node u is picked uniformly at random from $\{1, \dots, n\}$,

and a target v picked uniformly from $\{1, \dots, n\} \setminus u$ to create a temporal event (u, v, t) . We consider a special node u^* , such that if an event (u^*, v, t) occurs for some v, t , the subsequent event will be $(v, w, t + 1)$ for some w . This way we create a node with a walk-forming advantage as there is guaranteed to be a walk of length two for every event that node u^* initiates, and this walk will occur quickly (at the next time step). We anticipate that we should be able to easily identify this node from the broadcast centrality.

We consider the above model, run for $n = 20$ nodes and $m = 1000$ events. Using the time-decay centrality (but not the running dynamic centrality)^b we calculate the broadcast scores of all nodes in the network.

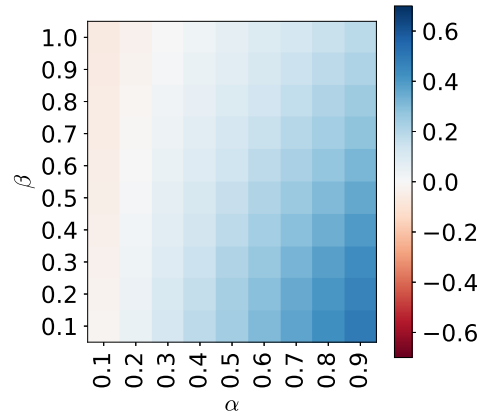


Fig. 7: Percentage difference in broadcast score between the node u^* and the competitor node u^c . For large values of α , i.e. $\alpha > 0.5$, the node u^* has a significant advantage over the nearest competition as one would expect. However for smaller values of α (and in particular for high values of β), this advantage is drastically reduced, and the two nodes are either indistinguishable or the competitor prevails.

In Figure 7 we plot the percentage difference in broadcast centrality between node u^* and the next nearest competitor node u^c . To elaborate, if node u^* is the most central, then we compare with the second most central node. If node u^* is not the most central node, we compare it with the most central node. Here, we see that there is a range of α and β for which node u^* is the most prominent node (blue), but there are regions of parameter space where either it is indistinguishable from another node (white), or it is not the most central node (red). This is potentially problematic for non-synthetic datasets where we do not know exactly how the choice of α and β affects the centrality ranking.

^b This way we eliminate the issue of where we ‘stop’ our analysis as in the running dynamic centrality $b(t)$ is a continuous function of time.

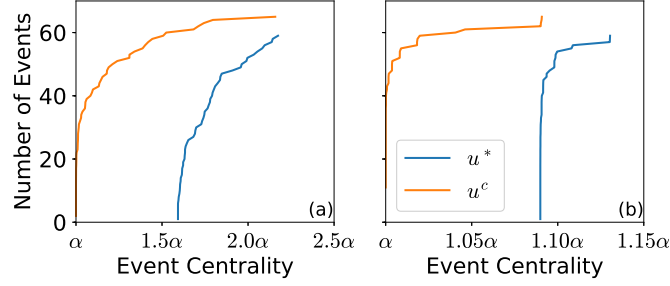


Fig. 8: The cumulative distribution of event centralities for the special node u^* and the challenger node u^c for two sets of parameters. In (a), where $(\alpha, \beta) = (0.8, 0.3)$, we see that u^* has participated in fewer events than the closest challenger however the events that u^* participates in have at least a centrality of $\alpha + \alpha^2$. Despite participating in more events than u^* , the majority of events for node u^c have a centrality less than $\alpha + \alpha^2$, resulting in a higher broadcast score for node u^* . In contrast for (b), where $(\alpha, \beta) = (0.2, 0.8)$, despite all events for u^* being more central than any of those for u^c , the difference between the event centralities is outweighed by the number of events, and therefore node u^c is deemed to be the most central.

To remedy this situation we can examine the distribution of event centralities that each node participates in. In Figure 8 we examine the distribution of event centralities for node u^* and the closest competitor for two different parameter regimes; one where node u^* is the most central (a), and one where it is not (b). In both cases, the competitor node participates in more events than u^* . However, in both cases, we can identify node u^* as the minimum event centrality is greater than the majority of event centralities for the other node. This means that node u^* is participating in fewer but important events than u^c . So how does u^* lose the top ranking? In Figure 8(b) node u^* is still participating in more important events, however *the relative extent of importance* of those events has diminished. Comparing the scale of the x-axis in both subplots, in (a) the most important event is over twice as important as the least. In (b) this is reduced drastically and the most important event is less than 15% more important than the least. In this scenario, when all events are approximately equal, the number of events a node participates in (degree) becomes most important.

As a more realistic example, we examine the broadcast centralities of nodes in the **social-ucirvine** dataset. Typically nodes are ranked by their broadcast centrality as a proxy to measure how well that node can send information around the network. In Figure 9(a) we show the distribution of event centralities for the top five ranking nodes in the temporal network using parameters $(\alpha, \beta) = (0.4, \ln(2)/1800)$. Taking $\beta = \ln(2)/1800$ means that the weight of temporal walks is discounted by one half every 30 minutes. In this parameter regime, the top broadcasting nodes are not

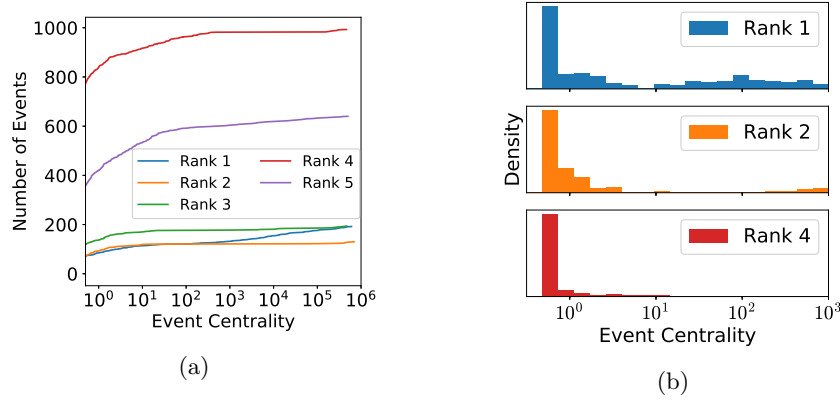


Fig. 9: (a) The cumulative distribution of event centralities for the top ranked nodes in the `social-ucirvine` dataset with $\alpha = 0.4$ and $\beta = \ln(2)/1800 \approx 4 \times 10^{-4}$. The top three ranked nodes are not those that participate in the largest number of events but in fact participate in a smaller number of more central events. (b) Event centrality distributions for three of the top five nodes. The top-ranked node participates in events which have a broad spread of centralities themselves. The second-ranked node participates primarily in low centrality events however also participates in many highly central events. The fourth ranked node (with the highest post rate of the top five) primarily is involved with only low centrality events.

those which participate in the most events but instead participate in more central events. Figure 9(b) shows the event centrality distribution for three of the top five accounts, and as in the synthetic example, each display a different profile from where their centrality is derived. The top-ranking node participates events which have a range of centralities (including high centrality events). The second-ranked node (and similarly the third, not shown) is central due to its participation in a small number of highly central events. The fourth (and fifth) ranked nodes derive their centrality from participating in a high number of less central events. These types of insights help understand the roles of different nodes in the temporal network. For example, in marketing on social media, advertisers are looking for influential individuals who generate engagement beyond simple mass-messaging. In this case the top three ranking nodes do this to varying extent, however, the nodes ranked fourth and fifth do not, and would perhaps be seen as ‘spammers’ rather than influencers.

It is also of note that after constructing the event graph we can acquire the broadcast centralities efficiently by solving the linear system $(I - \alpha M)\mathbf{x} = \mathbf{1}$. As M represents a directed acyclic graph it is upper triangular, and so the system can be solved efficiently by reverse substitution. Furthermore, there is no restriction of the value of α as in the original centrality description [15]. This makes the event graph approach both more insightful and less restrictive than the standard approach.

4.3. Percolation

This aspect of event graphs has recently received the most focus in [26, 25] and [37]. These examples consider the Δt -adjacent event graph. The introduction of the parameter Δt naturally poses the question of how the structure of the event graph varies with Δt . The answer to this question is non-trivial as the event graph incorporates not only the distribution of the inter-event times but is also a function of the topological interactions between nodes (i.e. the underlying graph structure). While this analysis has been conducted before, in this example we show how it can be extended to non-dyadic data through the use of the event graph. Furthermore, we show how non-dyadic and dyadic data can be compared side-by-side.

Definition 6. (Temporal component) *A temporal component of an event graph is a weakly connected component of the event graph, that is, a connected component of the graph if the directionality of edges is ignored^c.*

Note that as the event graph is a directed acyclic graph, there are no strongly connected components. Let $C_{\Delta t}$ be the set of temporal components in the Δt -adjacent event graph, and let each component $c \in C_{\Delta t}$ be defined by the event set $E^c \subseteq E$. To illustrate the characteristics of the event graph we consider three properties of temporal components (first given in [25] but generalised to hyperevents here): the number of events, the number of nodes, and the component duration. These are defined for directed events by

$$\begin{aligned} N_{\text{events}}(E) &= |E|, \\ N_{\text{nodes}}(E) &= \left| \bigcup_{(U,V,t) \in E} (U \cup V) \right|, \\ D(E) &= \max_{(U,V,t) \in E} t - \min_{(U,V,t) \in E} t, \end{aligned}$$

respectively. Equivalent expressions for undirected events follow easily.

The evolution of the largest component sizes (for these three properties) for a range of values of Δt is given in Figure 10(a)-(c). To be able to compare the different networks we rescale Δt by the 90th percentile of the IET distribution of the event graph. The synthetic network **random-complete** displays a single transition across all three properties for a single value of Δt . The **twitter-emirates** network makes up 60% of events and 80% of nodes as $\Delta t \rightarrow (\Delta t)_{90\%}$, however for a small value of Δt there is a temporal component which spans the entire time period. This is explainable by the presence of automated accounts (also known as *bots*) which can be programmed to post messages periodically. Once Δt exceeds the periodicity of the bot the temporal component which contains the bot will have a duration spanning the entire time period and would extend ad infinitum with more data

^c Temporal components have also been referred to as ‘maximal Δt -connected subgraphs’ specifically when the event connection rule is Δt -adjacency [26].

collection. The **sociopatterns-primary** network sees that most nodes feature in the largest component for small Δt . As we discuss later, this is problematic when trying to prevent a spreading process from proliferating across the network. Finally, the largest component in the **academic-coauthors** network has the smallest relative size for both nodes and events. This is somewhat expected given that the effort required to jointly publish a paper together far exceeds that of sending a single message in an online conversation. Furthermore, while cross-group and cross-field collaborations may occur, they are more likely to be infrequent so these connections may well fall into the last decile of the IET distribution.

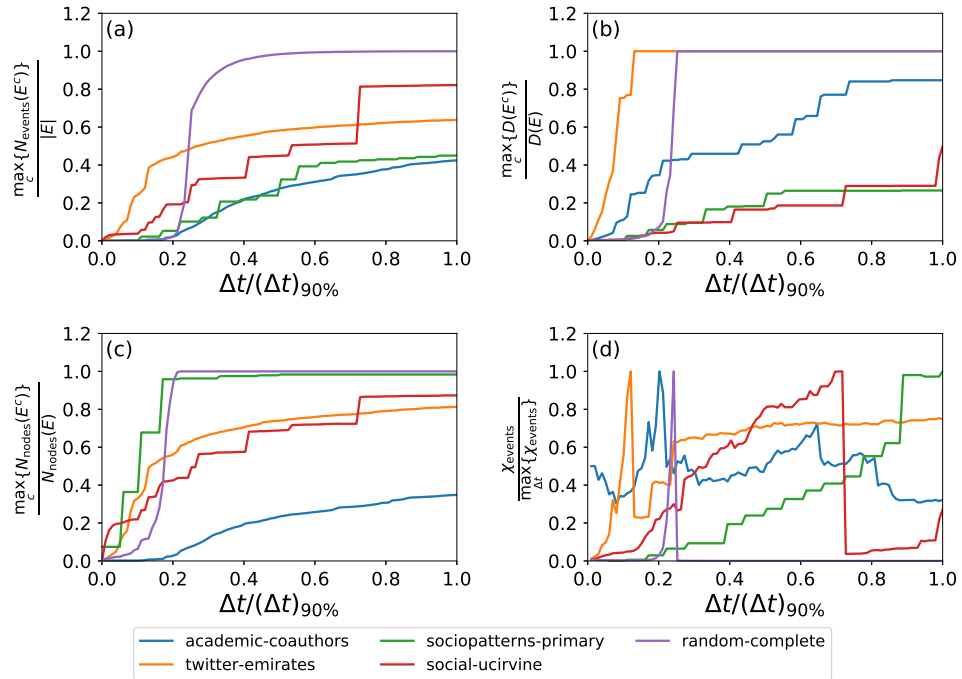


Fig. 10: The size of the largest component in the event graph in terms of (a) the number of events, (b) the duration of the component, and (c) the number of nodes in the component. Each is reported as a fraction of the total number of events, total duration, and the total number of nodes of the data respectively. The event susceptibility is given in (d), defined as the average squared-size of all components barring the largest. To compare across datasets the susceptibility has been rescaled by the maximum value attained. The parameter Δt is rescaled by the 90th percentile of the IET distribution for each dataset.

Recent work considers a toy model which samples events from an Erdős-Rényi network and investigates the connectivity of the event graph as a function of the

average degree of the underlying network, as well as Δt [25]. The particular focus of this work is to find the critical point Δt_c which separates a disconnected event graph to that which contains a significant largest component. In addition to the largest temporal component, they also investigate the average squared-size of all other components, excluding the largest. This is referred to as the *susceptibility* (denoted by χ and originating from magnetic susceptibility in statistical physics). The peaks in susceptibility are associated with critical points or timescales in the network. For their toy model (and similar to the **random-complete** network) there is one peak associated with the rapid increase in component size. The picture for real-world networks, however, is not so clear. Investigating over a network of sexual-interaction, a mobile communications network, and a network of air transportation (flights), shows that although there may be a large dominant peak (or critical point), there are often other smaller peaks. This hints at there being multiple timescales involved in the formation of the temporal network. We can see examples of this in Figure 10(d). The **random-complete** network has a clear peak, whereas for **twitter-emirates** and **academic-coauthors** there are multiple peaks. For the **sociopatterns-primary** dataset such a peak does not exist, but instead exhibits a step-wise increase with Δt .

What does percolation analysis tell us about dynamics on these networks? The Δt -adjacent event graph captures all temporal walks through the temporal network and hence captures how well a process can spread through the temporal network. The parameter Δt represents the minimum time that a spreading process must persist on a node in order for it to spread effectively. Consider an epidemic model on the **sociopatterns-primary** dataset. If the recovery time from infection is less than 0.2 (working in units relative to the 90th IET percentile) then the spreading process can only reach a fraction of all nodes under perfect conditions (Figure 10(c)), whereas if the recovery time is longer then it can potentially reach all nodes. This type of information is incredibly useful and allows us to ascertain the possible reach of a spreading process starting at a particular point in time, or understand how to reduce spread by removing central events (as calculated in Section 4.2).

Using the event graph we can compare the characteristic timescales and spreading properties of both dyadic and non-dyadic interactions in a meaningful way. This allows us to perform percolation analysis on a wider range of data, illustrated here through the study of non-dyadic collaboration networks, proximity networks, and social networks.

5. Conclusion

In this article, we have described event graphs, second-order time-unfolded graphical models of a temporal network. These generalise several previous approaches [37, 25] and provide an efficient means to study higher-order properties of temporal networks, as well as offering further insight into previously studied aspects of temporal networks. We showed that the event graph provides a natural means to extend

temporal network analysis to non-dyadic interactions (known as hyper-events) so that both non-dyadic and dyadic interaction networks can be studied in unison. Our examples show that event graphs can be a useful tool in a number of applications. In motif analysis, they provide an efficient means to enumerate motifs in a temporal network. Centrality analysis using event graphs also showed how we can use higher-order objects (events) to distinguish between nodes with similar centralities, and again providing an efficient means to do so. Although not discussed here they also can be used to decompose temporal networks and classify collective behaviour [35].

Event graphs are not without their limitations, however. There are considerations on the computational complexity of the event graph construction. As a second-order model there are m events in the event graph, with $m \gg n$ typically (unless each node interacts only once). The increase in the size of the graph considered can be counteracted by employing efficient algorithms that exploit the fact that the event graph is a directed acyclic graph [9]. One such example is that the shortest and longest paths can be found on directed acyclic graphs in linear time. In contrast, on arbitrary graphs, the best shortest path algorithms have greater than linear time complexity, and the longest path problem is NP-hard.

There can also difficulty choosing the parameter Δt (if we choose Δt -adjacency as our connection rule), however, this can be addressed by further research into the critical timescales of temporal networks. Furthermore, there are unaddressed issues with temporal networks which have multiple timescales and how these should be modelled in general.

As the data we are collecting is becoming more complex, and *relational* data is becoming more prevalent, we are seeing a shift away from simplistic first-order models towards second- and higher-order models. In this article we make some progress in both senses of the term ‘higher-order’, modelling both second-order dependences and non-dyadic interactions. There are however still many more avenues to explore to aid our understanding of temporal networks and to create meaningful and computationally feasible higher-order models.

Code Availability

Code is available freely online as part of the `eventgraphs`^d Python package [36]. Notebooks which cover the analysis of this article are available in the `examples/advances_and_applications_paper` folder of the repository.

References

- [1] Alon, U., Network motifs: Theory and experimental approaches, *Nature Reviews Genetics* **8** (2007) 450–461.

^d <https://github.com/empiricalstateofmind/eventgraphs>

- [2] Batagelj, V. and Praprotnik, S., An algebraic approach to temporal network analysis based on temporal quantities, *Social Network Analysis and Mining* **6** (2016) 28.
- [3] Benson, A. R., Abebe, R., Schaub, M. T., Jadbabaie, A., and Kleinberg, J., Simplicial closure and higher-order link prediction, *Proceedings of the National Academy of Sciences* **115** (2018).
- [4] Benson, A. R., Kumar, R., and Tomkins, A., Sequences of Sets, in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (ACM, 2018), pp. 1148–1157.
- [5] Berge, C., *Hypergraphs: Combinatorics of Finite Sets*, Vol. 45 (Elsevier, 1984).
- [6] Brandes, U., Lerner, J., and Snijders, T. A. B., Networks Evolving Step by Step: Statistical Analysis of Dyadic Event Data, in *2009 International Conference on Advances in Social Network Analysis and Mining* (2009), pp. 200–205, doi:10.1109/ASONAM.2009.28.
- [7] Butts, C. T., 4. A Relational Event Framework for Social Action, *Sociological Methodology* **38** (2008) 155–200.
- [8] Casteigts, A., Flocchini, P., Quattrociocchi, W., and Santoro, N., Time-Varying Graphs and Dynamic Networks, *arXiv:1012.0009 [physics]* (2010).
- [9] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms* (MIT press, 2009).
- [10] De Bruijn, N. G., A combinatorial problem, *Koninklijke Nederlandse Akademie v. Wetenschappen* **49** (1946) 758–764.
- [11] Gallo, G., Longo, G., Pallottino, S., and Nguyen, S., Directed hypergraphs and applications, *Discrete applied mathematics* **42** (1993) 177–201.
- [12] Gao, Z., *Spatiotemporal Analysis of Air-Travel Networks*, MSc in Mathematical Modelling and Scientific Computing, University of Oxford (2017).
- [13] Girvan, M. and Newman, M. E., Community structure in social and biological networks, *Proceedings of the National Academy of Sciences* **99** (2002) 7821–7826.
- [14] Grindrod, P. and Higham, D. J., A dynamical systems view of network centrality, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science* **470** (2014) 20130835.
- [15] Grindrod, P., Parsons, M. C., Higham, D. J., and Estrada, E., Communicability across evolving networks, *Physical Review E* **83** (2011) 046120.
- [16] Gueuning, M., Lambiotte, R., and Delvenne, J.-C., Backtracking and Mixing Rate of Diffusion on Uncorrelated Temporal Networks, *Entropy* **19** (2017) 542.
- [17] Gurukar, S., Ranu, S., and Ravindran, B., Commit: A scalable approach to mining communication motifs from dynamic networks, in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (ACM, 2015), pp. 475–489.
- [18] Hashimoto, K.-i., Zeta functions of finite graphs and representations of p-adic groups, in *Automorphic Forms and Geometry of Arithmetic Varieties* (Elsevier, 1989), pp. 211–280.
- [19] Hemminger, R. L. and Beineke, L. W., Line graphs and line digraphs, *Selected topics in graph theory* **1** (1978) 291–305.
- [20] Holme, P., Modern temporal network theory: A colloquium, *The European Physical Journal B* **88** (2015) 234.
- [21] Holme, P. and Saramäki, J., *Temporal Networks* (Springer, 2013).
- [22] Horton, M. D., Stark, H., and Terras, A. A., What are zeta functions of graphs and what are they good for?, *Contemporary Mathematics* **415** (2006) 173–190.
- [23] Kempe, D., Kleinberg, J., and Kumar, A., Connectivity and inference problems for temporal networks, in *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing* (ACM, 2000), pp. 504–513.

- [24] Kempe, D., Kleinberg, J., and Kumar, A., Connectivity and Inference Problems for Temporal Networks, *Journal of Computer and System Sciences* **64** (2002) 820–842.
- [25] Kivelä, M., Cambe, J., Saramäki, J., and Karsai, M., Mapping temporal-network percolation to weighted, static event graphs, *Scientific reports* **8** (2018) 12357.
- [26] Kovanen, L., Karsai, M., Kaski, K., Kertész, J., and Saramäki, J., Temporal motifs in time-dependent networks, *Journal of Statistical Mechanics: Theory and Experiment* **2011** (2011) P11005.
- [27] Kovanen, L., Kaski, K., Kertész, J., and Saramäki, J., Temporal motifs reveal homophily, gender-specific patterns, and group talk in call sequences, *Proceedings of the National Academy of Sciences* **110** (2013) 18070–18075.
- [28] Krzakala, F., Moore, C., Mossel, E., Neeman, J., Sly, A., Zdeborová, L., and Zhang, P., Spectral redemption in clustering sparse networks, *Proceedings of the National Academy of Sciences* **110** (2013) 20935–20940.
- [29] Lambiotte, R., Rosvall, M., and Scholtes, I., Understanding Complex Systems: From Networks to Optimal Higher-Order Models, *arXiv preprint arXiv:1806.05977* (2018).
- [30] Lambiotte, R., Salnikov, V., and Rosvall, M., Effect of memory on the dynamics of random walks on networks, *Journal of Complex Networks* **3** (2014) 177–188.
- [31] Latapy, M., Viard, T., and Magnien, C., Stream graphs and link streams for the modeling of interactions over time, *Social Network Analysis and Mining* **8** (2018) 61.
- [32] Masuda, N. and Holme, P., Predicting and controlling infectious disease epidemics using temporal networks, *F1000prime reports* **5** (2013).
- [33] Masuda, N., Klemm, K., and Eguíluz, V. M., Temporal networks: Slowing down diffusion by long lasting interactions, *Physical Review Letters* **111** (2013) 188701.
- [34] Masuda, N. and Lambiotte, R., *A Guide to Temporal Networks*, Vol. 4 (World Scientific, 2016).
- [35] Mellor, A., Analysing Collective Behaviour in Temporal Networks Using Event Graphs and Temporal Motifs, *arXiv preprint arXiv:1801.10527* (2018).
- [36] Mellor, A., Eventgraphs Python package (v0.2) (2018).
- [37] Mellor, A., The temporal event graph, *Journal of Complex Networks* **6** (2018) 639–659.
- [38] Michail, O., An introduction to temporal graphs: An algorithmic perspective, *Internet Mathematics* **12** (2016) 239–280.
- [39] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U., Network motifs: Simple building blocks of complex networks, *Science* **298** (2002) 824–827.
- [40] Navaroli, N. M. and Smyth, P., Modeling Response Time in Digital Human Communication., in *ICWSM* (2015), pp. 278–287.
- [41] Nicosia, V., Tang, J., Musolesi, M., Russo, G., Mascolo, C., and Latora, V., Components in time-varying graphs, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **22** (2012) 023101.
- [42] Panzarasa, P., Opsahl, T., and Carley, K. M., 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** (2009) 911–932.
- [43] Paranjape, A., Benson, A. R., and Leskovec, J., Motifs in temporal networks, in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (ACM, 2017), pp. 601–610.
- [44] Peng, Y., Jiang, Y., and Radivojac, P., Enumerating consistent sub-graphs of directed acyclic graphs: An insight into biomedical ontologies, *Bioinformatics* **34** (2018) i313–i322.
- [45] Pržulj, N., Biological network comparison using graphlet degree distribution, *Bioin-*

- formatics* **23** (2007) e177–e183.
- [46] Rosvall, M., Esquivel, A. V., Lancichinetti, A., West, J. D., and Lambiotte, R., Memory in network flows and its effects on spreading dynamics and community detection, *Nature communications* **5** (2014).
 - [47] Salnikov, V., Cassese, D., Lambiotte, R., and Jones, N. S., Co-occurrence simplicial complexes in mathematics: Identifying the holes of knowledge, *Applied Network Science* **3** (2018) 37.
 - [48] Saramäki, J. and Holme, P., Exploring temporal networks with greedy walks, *The European Physical Journal B* **88** (2015).
 - [49] Scholtes, I., When is a Network a Network?: Multi-Order Graphical Model Selection in Pathways and Temporal Networks, in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, 2017), pp. 1037–1046.
 - [50] Scholtes, I., Wider, N., and Garas, A., Higher-order aggregate networks in the analysis of temporal networks: Path structures and centralities, *The European Physical Journal B* **89** (2016) 61.
 - [51] Scholtes, I., Wider, N., Pfitzner, R., Garas, A., Tessone, C. J., and Schweitzer, F., Causality-driven slow-down and speed-up of diffusion in non-Markovian temporal networks, *Nature Communications* **5** (2014).
 - [52] sociopatterns.org, SocioPatterns (2018).
 - [53] Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J.-F., Quaggiotto, M., Van den Broeck, W., Régis, C., Lina, B., *et al.*, High-resolution measurements of face-to-face contact patterns in a primary school, *PloS one* **6** (2011) e23176.
 - [54] Takaguchi, T., Yano, Y., and Yoshida, Y., Coverage centralities for temporal networks, *The European Physical Journal B* **89** (2016) 35.
 - [55] Taramasco, C., Cointet, J.-P., and Roth, C., Academic team formation as evolving hypergraphs, *Scientometrics* **85** (2010) 721–740.
 - [56] Xu, J., Wickramaratne, T. L., and Chawla, N. V., Representing higher-order dependencies in networks, *Science Advances* **2** (2016) e1600028.
 - [57] Zhao, Q., Tian, Y., He, Q., Oliver, N., Jin, R., and Lee, W.-C., Communication motifs: A tool to characterize social communications, in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (ACM, 2010), pp. 1645–1648.

Appendix A. Event and Node Communicability Centrality

Claim: We claim that the event and node dynamic communicability matrices (Q_E and Q respectively) are related by $Q = X_s(\alpha Q_E)X_t^\top$, where X_s and X_t are the node-source and node-target incidence matrices respectively (defined in the main text).

Proof: To prove this we show that the matrix Q_E captures all temporally weighted walks in the temporal network. Suppose there is a temporal path of length k from node u^* to node v^* . Then by definition, there exists a sequence of timestamped edges (or events)

$$(n_1, n_2, t_1), (n_2, n_3, t_2), \dots, (n_k, n_{k+1}, t_k)$$

such that $n_1 = u^*$ and $n_{k+1} = v^*$ and $t_i < t_{i+1}$ for all i . For ease, label these events e_1, e_2, \dots, e_k .

We now construct the event graph matrix M of these events. For all i , $M_{i,i+1} = 1$ as $v_i = n_{i+1} = u_{i+1}$ and $t_{i+1} - t_i > 0$. The graph consists of a single chain of length $k - 1$, $e_1 \rightarrow e_2 \rightarrow \dots \rightarrow e_k$ assuming that all nodes $(n_i)_{i=1}^{k+1}$ are distinct^e. This implies $(M^{k-1})_{1k} = 1$ with $u_1 = u^*$ and $v_k = v^*$. Since the chosen walk is arbitrary $\alpha^k M^{k-1}$ will capture all weighted walks of length k on the event graph.

Therefore, to capture all walks we consider

$$\alpha Q_E := \alpha(I - \alpha M)^{-1} = \alpha I + \alpha^2 M + \alpha^3 M^2 + \dots$$

noting that $M^0 = I$. We consider (αQ_E) as walks of length k on the event graph correspond to walks of length $k + 1$ on the original network. We can guarantee convergence as the event graph is a DAG, so there exists an integer R such that $M^R = 0$. We can therefore chose α freely in $[0, \infty)$ (although we choose $\alpha \in (0, 1)$ typically). The matrix Q_E acts similarly to the matrix Q in that it contains the information of all walks in the network. To find all walks which start at a particular node, we premultiply by X_s . The non-zero entries in each row of $X_s Q_E$ give the contribution of each event to the node. By a similar argument, we can show that $Q_E X_t^\top$ counts the weighted number of walks from each event to each node. Combining these we can calculate the weight of walks that start at one node and end at another, recovering the node communicability matrix as

$$Q = X_s (\alpha Q_E) X_t^\top.$$

Appendix B. Motifs as Subgraphs

Claim: All 3-node, 3-event motifs are represented by a subgraph $e_i \rightarrow e_j \rightarrow e_k, e_i \rightarrow e_k$ of the Δt -adjacent event graph.

Proof: Consider three events e_1, e_2, e_3 which involve three nodes. To be a motif they must form a connected component, so trivially let $e_1 \rightarrow e_2 \rightarrow e_3$. Let $e_1 = (a, b, t_1)$. There are two possibilities, either e_1 only shares one node with e_2 , or shares both nodes. If they share both nodes then either $e_3 = (a, c, t_3)$ or $e_3 = (b, c, t_3)$ and so $e_1 \rightarrow e_3$ also. If they share only one node then $e_3 \in \{(a, c, t_3), (b, c, t_3), (a, b, t_3)\}$ and so $e_1 \rightarrow e_3$ also.

^e If they are not distinct then there potentially will be other edges and other possible walks, although none will have a length greater than $k - 1$.