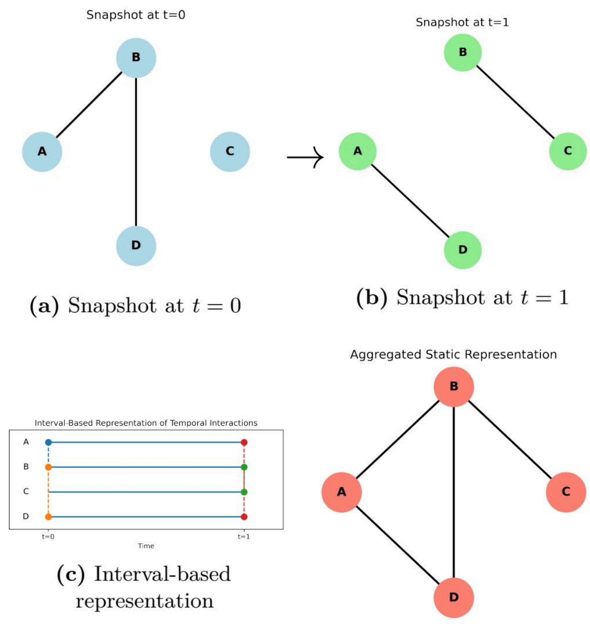
Course Handout on Temporal Networks

William Alexander Bailkoski[a,b], Philipp Heinrich Liehe[a,b], and Matic Pristavnik[a]

[a]University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, SI-1000 Ljubljana, Slovenia; [b]Universitat Pompeu Fabra, Department of Engineering, Carrer de Roc Boronat 138, E-08018 Barcelona, Spain

The manuscript was compiled on May 23, 2025

This handout provides a comprehensive overview of temporal networks, an essential framework for understanding systems where interactions and connectivity evolve over time. We explore the fundamental concepts of representing time-varying network data, discuss key modeling frameworks and the most common temporal network analysis metrics. In the final section of the handout, we present a practical approach to basic community detection on a high-quality temporal network dataset and link our materials for students to experiment with. This handout aims to equip students with the foundational knowledge to reason about, analyze and interpret complex dynamic systems through the lens of network science, highlighting the limitations of static approaches and the applications offered by a temporal perspective.



(a) Snapshot at $t = 0$ **(b)** Snapshot at $t = 1$

(c) Interval-based representation **(d)** Aggregated static representation

Fig. 1. Conceptual illustration of different temporal network representations: **(a)** and **(b)** illustrate snapshot-based temporal representations at sequential time steps. **(c)** shows the interval-based representation highlighting node activity over time. **(d)** demonstrates the aggregated static network, illustrating the loss of temporal information.

Beyond Static Snapshots In the study of complex systems, network science has provided a powerful framework for understanding the relationships and interactions between constituent entities (1). Traditional network analysis usually deals with static representations, where connections are treated as if they exist persistently over time, essentially giving us a single snapshot representation of the underlying system. While this approach has yielded significant insights into network topology, many real-world systems are inherently dynamic, with interactions appearing, disappearing, and changing in nature over time (2). Ignoring this temporal dimension can lead to an incomplete, and sometimes misleading, understanding of system behavior, information propagation, and structural evolution (3). This handout delves into the realm of *temporal networks*, providing a foundational understanding of their structure, analysis, and significance.

Defining Temporal Networks and Their Components

A temporal network explicitly incorporates the timing of interactions. Formally, it can be defined as a pair $G_T = (V, E_T)$, where V is a set of nodes (or vertices) and E_T is a set of time-stamped edges (or contacts) that represent interactions between pairs of nodes at specific points or intervals in time (2). Unlike static graphs where an edge (u, v) either exists or does not, in a temporal network, an interaction between nodes u and v is associated with one or more temporal markers, such as a timestamp t, an interval $[t_{start}, t_{end}]$, or a sequence of such markers. The fundamental difference from static networks hence lies in this explicit encoding of time. This means that concepts like paths, reachability, and centrality must be re-evaluated to respect the temporal ordering and availability of connections (3). For instance, a path from node A to node C via node B is only valid if the interaction (A, B) occurs before the interaction (B, C) in a way that allows for traversal (see figure 2).

The Significance of Temporal Dynamics. The explicit inclusion of time is not merely a technical detail; it fundamentally alters our understanding of network structure and function. Many processes occurring on networks, such as the spread of information or diseases, are constrained by the temporal ordering of contacts (2). For instance, in business contexts, understanding the temporal patterns of communication within an organization can reveal dynamic workflows, identify bottlenecks, or trace the propagation of influence much more accurately than a static "who-knows-whom" map aggregated for the entire lifetime of a network. Similarly, financial networks exhibit strong temporal dependencies where the timing of transactions is critical (4). As illustrated in Figure 1, representing networks with explicit temporal information (through snapshots or interval-based visualizations) captures dynamic interaction patterns and clearly highlights the evolution of connections.

All authors contributed equally to this work.

[1]To whom correspondence should be addressed. E-mail: liehe.p.h@gmail.com.

In contrast, aggregating interactions into a static network, as shown in Figure 1(d), obscures these critical temporal dynamics, resulting in the loss of important temporal structure and information.

The study of temporal networks has thus become a vibrant subfield of network science, addressing questions like:

- How do we define and measure "centrality" or "importance" when connections are fleeting? (3)
- How do communities of nodes form, evolve, and dissolve over time? (4, 5)
- How do temporal patterns like burstiness (short periods of high activity followed by long quiescence) or correlations in contact times affect dynamic processes? (2)

Our choice of how to model these temporal aspects (e.g., the granularity of time or whether to aggregate data) directly impacts the network properties we observe and the conclusions we can draw. For example, aggregating interactions over too long a period can create spurious paths or overestimate connectivity, masking the true causal pathways of influence or contagion (2). In short, domain knowledge and a deep understanding of possible underlying phenomena that one wants to investigate are crucial for an appropriate utilization of temporal networks as a representation of dynamic systems.

Real-World Temporal Networks: Characteristics and Examples. Real-world temporal networks span a vast array of domains and exhibit diverse characteristics. Examples include:

- **Communication Networks:** Email exchanges, phone calls, social media interactions (e.g., mentions, messages) are often characterized by bursty activity and strong temporal correlations (2).
- **Proximity Networks:** Face-to-face interactions, often captured by sensors, reveal patterns of human mobility and social group formation. These are crucial for understanding disease spread.
- **Infrastructure Networks:** Transportation networks (flights, public transit) or information networks (internet routing changes) have scheduled or demand-driven temporal dynamics.
- **Biological Networks:** Protein-protein interactions can be transient, and neural networks exhibit dynamic activity patterns over various timescales.

Understanding these specific temporal characteristics is vital for building accurate models and making reliable predictions (6). For instance, the efficiency of information diffusion might be significantly different in a network with regular, predictable contacts versus one with highly unpredictable, bursty contacts (3).

Representing and Modeling Temporal Networks

Understanding and analyzing temporal networks begins with how we choose to represent their evolving structure and interactions. This choice is not merely a technicality; it fundamentally shapes the types of questions we can ask and the analytical tools we can employ (2). This section outlines fundamental temporal concepts, common data representations, and key modeling frameworks.

Fundamental Temporal Concepts. At its core, a temporal network extends the static graph by incorporating the dimension of time. This introduces several key concepts:

- **Nodes and Edges with Temporal Attributes:** While nodes (V) are often considered persistent entities, their states or attributes might change over time. Edges (E_T), representing interactions or relationships, are explicitly time-dependent. Their existence is not continuous but rather tied to specific temporal markers.
- **Time Stamps:** A specific point in time, t, often used to mark the occurrence of an instantaneous event or contact (e.g., an email sent at time t) (2).
- **Durations (Δt):** The length of time for which an interaction persists or a state is active. For an interaction starting at t_{start} and ending at t_{end}, the duration is $\Delta t = t_{end} - t_{start}$.
- **Intervals $[t_{start}, t_{end}]$:** A continuous period during which an edge is active or a node possesses a certain property. The definition and handling of time (discrete vs. continuous, ordered events vs. concurrent intervals) are crucial and can significantly impact analysis.
- **Contact/Event:** A specific instance of an interaction or co-presence between two nodes, (u, v), occurring at a particular time t or over an interval $[t_{start}, t_{end}]$.

The nature of time itself in these networks can be discrete (e.g., time steps, days) or continuous. Data can be time-stamped (marking instants) or interval-based (marking durations) (6).

Common Representations of Temporal Networks. The raw data capturing temporal interactions can be structured in several ways for analysis. The choice of representation often involves a trade-off between temporal resolution, data volume, and analytical complexity and is highly domain- and task-dependent.

In a **Snapshot-Based Network**, the dynamic system is discretized into an ordered sequence of static graphs, $G_1, G_2, \ldots, G_M$, where each $G_i = (V, E_i)$ represents the network topology within a specific time window (**Definition 1**).

Def. 1: Snapshot-Based Graph ($\mathcal{G}_S$)

Network $\mathcal{G}_S$ consists of an ordered sequence of M static graphs (snapshots):

$$\mathcal{G}_S = (G_1, G_2, \ldots, G_M). \quad [1]$$

Each snapshot $G_t = (V, E_t)$ for $t \in \{1, \ldots, M\}$ consists of the set of nodes V and the set of edges E_t active within a specific time window $[(t-1)\Delta t, t\Delta t]$ (2).

The choice of the temporal resolution Δt is arbitrary (though limited by the resolution of the data) and should be adapted to the domain and purpose of the analysis performed. In contrast, **Event-Based Networks** list all observed interactions (contacts or events) chronologically and hence have a predefined temporal resolution that captures the true underlying temporal granularity of the dynamic system (**Definition 2**).

Def. 2: Event-Based Graph ($\mathcal{G}_E$)

Interactions in $\mathcal{G}_E$ are recorded as a chronological list of discrete events:

$$\mathcal{G}_E = \{(u_k, v_k, t_k, [\delta t_k])\}_{k=1}^{K}. \qquad [2]$$

Each tuple typically denotes an interaction between nodes u_k and v_k at time t_k, optionally including a duration δt_k.

While this representation ensures that all dynamic information is captured, it can lead to very large datasets, especially for systems with many interactions or long durations. Global structural properties might be harder to visualize or compute directly without some form of aggregation or specialized algorithms. Expanding on this concept, **Link Streams (Stream Graphs)** are a more recent formalism that views a temporal network as a continuous stream of links (t, u, v) along with start and end times for nodes and the stream itself. It provides a rigorous mathematical framework to generalize classical graph concepts to the temporal domain (7).

Def. 3: Link Stream (Stream Graph) ($\mathcal{G}_{LS}$)

A Link Stream $\mathcal{G}_{LS}$ is formally defined as a tuple:

$$\mathcal{G}_{LS} = (T_S, V_S, E_S, \psi_S), \qquad [3]$$

where T_S is the time domain (e.g., an interval representing the lifetime of the stream), V_S is the set of nodes, $E_S \subseteq V_S \times V_S$ is a set of possible links (node pairs), and $\psi_S : E_S \to \mathcal{P}(T_S)$ is a function mapping each link $(u, v) \in E_S$ to the set of time points or intervals within T_S during which it is active (7).

The choice among these representations makes a huge difference for the quality and appropriateness of downstream analysis and depends on the nature of the available data, the specific research questions, and computational considerations. Snapshot sequences simplify analysis by allowing the application of static graph tools (like NetworkX) at discrete time points, but the aggregation over Δt inherently leads to a loss of fine-grained temporal information and the choice of this resolution can significantly alter observed network properties (3). Contact sequences, conversely, offer maximum temporal resolution, which is crucial for studying causality and exact timings, but may lead to very large datasets and require specialized algorithms for analysis (2). Link streams provide a more abstract and comprehensive mathematical formalism, aiming to unify these perspectives and enable rigorous definitions of temporal graph concepts (7).

In general, analysis might often involve transitioning between representations or using hybrid approaches. For example, one might start with a contact sequence and then generate snapshots for specific types of analysis or visualization.

Widely Used Modeling Frameworks. Beyond representing empirical data, various modeling frameworks aim to generate synthetic temporal networks with specific properties or to describe the underlying mechanisms driving their evolution.

Time-Varying Graphs (TVGs) This framework, often an umbrella term for discrete-time representations like snapshot sequences (8), considers a graph $G = (V, E)$ where the active edge set $E(t) \subseteq E$ evolves over discrete time steps $t \in \{1, \ldots, T\}$. TVGs are particularly useful for **analyzing reachability, temporal paths, and designing time-dependent routing algorithms**.

Activity-Driven Networks (ADNs) ADNs are generative models where link formation is driven by node activity levels (9). Each node i has an activity rate a_i; at each time step, active nodes connect to a number of randomly chosen nodes for that instant. This approach can reproduce stylized facts of temporal networks, such as bursty inter-event times and heterogeneous aggregated degree distributions, using simple rules. ADNs are well-suited for modeling systems **driven by individual initiatives** (e.g., human communication, proximity networks) and for studying how activity potentials influence spreading processes (3, 9).

Memory Models (Temporal Networks with Memory) These models posit that interaction history shapes future network structure (3, 10). The likelihood of link formation or reactivation can depend on past events, such as time since last contact, interaction frequency, or specific temporal motifs. Incorporating memory is vital for capturing long-range temporal correlations, non-Markovian dynamics, and realistic (especially human) behavior. Such models aid in understanding phenomena like **social tie reinforcement, path dependency in diffusion, and the emergence of stable interaction patterns**, offering deeper insights into network evolution despite their analytical complexity (6).

Other modeling approaches include temporal extensions of stochastic block models, agent-based models where node movement or state changes drive link formation, and models based on preferential attachment adapted for temporal dynamics. The choice of model depends on the specific temporal characteristics one aims to reproduce or study.

Temporal Network Analysis

Temporal networks require distinct analytical approaches due to the explicit ordering and timing of interactions. Concepts from static networks, such as connectivity and centrality, must be reconsidered to accurately reflect temporal dynamics.

Time-Respecting Paths and Temporal Reachability. Static networks implicitly assume edges are permanently active, allowing for paths that may not exist when the temporal order of interactions is considered. In temporal networks, paths must explicitly respect the **chronological order of interactions**, fundamentally altering notions of connectivity and reachability.

Def. 4: Time-Respecting Path

A time-respecting path from node u to node v in a temporal network is defined as a sequence of contacts $(u, u_1, t_1), (u_1, u_2, t_2), \ldots, (u_{k-1}, v, t_k)$, such that each timestamp t_i represents the occurrence of interaction i and satisfies the chronological constraint $t_1 < t_2 < \cdots < t_k$. If contacts have durations, the constraint is that the ending time of each contact precedes the starting time of the next.

Time-respecting paths lead to specific path concepts unique to temporal networks, including the *fastest path* (shortest total duration from start to end), *foremost path* (earliest arrival at destination), and *shortest-hop path* (fewest temporal steps).

The temporal ordering constraint introduces significant challenges related to reachability, as illustrated in Figure 2. The figure demonstrates a key difference from static network intuition: connectivity is *non-transitive* and heavily depends on the timing of interactions. For instance, although there appears to be a static path from node A to node C via node B, this path is only feasible if the timing of interactions is appropriate. In the example shown, if information or a disease spreads from node A, it may fail to reach node C due to temporal ordering constraints. Conversely, initiating the spread from another node (e.g., node D) could result in successful propagation through the entire network. This temporal dependency and asymmetry in reachability highlight essential dynamics that cannot be captured when aggregating temporal interactions into static graphs (2).

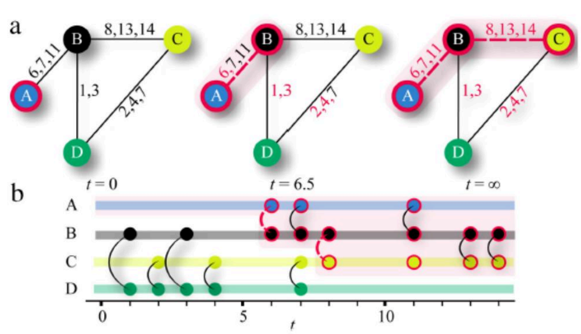


Fig. 2. Temporal reachability issue illustrating that node A can only reach node C through node B when temporal interactions are appropriately timed. Red highlights indicate reachable nodes at given times. Adapted from (2).

Temporal Centrality. In dynamic contexts, node importance cannot rely solely on static structural positions. Instead, centrality measures must incorporate temporal activity and participation in dynamic processes to adequately identify key nodes in evolving systems(3). With that, the notion of **centrality becomes more process- and time-dependent**.

Several classical centrality measures have temporal analogs, modified to incorporate the time dimension:

Def. 5: Temporal Degree Centrality

Temporal degree centrality for node v can be defined as:

$$C_{degree}(v) = \frac{|\{(v, u, t) \in E_T\}|}{T}, \quad [4]$$

where $|\{(v, u, t)\}|$ is the number of interactions involving v and T is the total observation period.

Temporal degree centrality measures node activity or interaction frequency over time.

To define other temporal centralities clearly, we first introduce the concept of *shortest temporal paths*. In temporal networks, a **shortest temporal path** between two nodes is a sequence of contacts that respect the chronological ordering of interactions, ensuring each subsequent contact occurs after the previous one. The *shortest* criterion can be based either on the minimal number of hops (contacts) or the minimal total duration.

Def. 6: Temporal Betweenness Centrality (11)

Temporal betweenness centrality for a node i at time t is defined as:

$$C_i^B(t) = \frac{1}{(N-1)(N-2)} \sum_{\substack{j \in V \\ j \neq i}} \sum_{\substack{k \in V \\ k \neq i \wedge k \neq j}} \frac{U(i, t, j, k)}{|S_{jk}^h|}, \quad [5]$$

where S_{jk}^h is the set of shortest temporal paths from node j to node k at time t, and $U(i, t, j, k)$ is an indicator function that equals 1 if node i is part of any path in S_{jk}^h at time t, and 0 otherwise. If no such paths exist ($S_{jk}^h = \emptyset$), $C_i^B(t)$ is set to 0.

Temporal betweenness centrality captures a node's ability to facilitate communication over time-respecting paths, accounting not only for whether it lies on a shortest path between pairs of nodes but also when it does so. This reflects the timing and frequency of a node's strategic importance in information flows (11).

Def. 7: Temporal Closeness Centrality

Temporal closeness centrality for node v is defined as:

$$C_{close}(v) = \frac{N-1}{\sum_{u \neq v} d_{temporal}(v, u)}, \quad [6]$$

where $d_{temporal}(v, u)$ is the length of the shortest time-respecting path (fastest or foremost) from node v to node u, and N is the number of nodes reachable from v.

Temporal closeness measures the efficiency with which a node can reach others, crucial for understanding rapid response capabilities or information dissemination speed.

Metrics Emphasizing Temporal Characteristics. Additional metrics focus specifically on temporal characteristics:

- **Burstiness**: High intermittent activity versus consistent interaction patterns.

- **Inter-event times**: Distributions of times between node interactions, influencing process dynamics such as information spread.

These metrics highlight node roles beyond traditional centrality, providing deeper insights into temporal behaviors.

Network-Wide Temporal Metrics. Network-wide temporal metrics provide overarching insights into the dynamics and structural characteristics of temporal networks. These metrics extend beyond node-specific centrality measures to capture global properties of temporal interactions:

Temporal Density

Temporal density quantifies the intensity of interactions observed over time, defined as:

$$D_T = \frac{|E_T|}{|V| \cdot (|V| - 1) \cdot T}, \qquad [7]$$

where $|E_T|$ is the number of observed interactions, $|V|$ is the number of nodes, and T is the total observation period.

Temporal Clustering Coefficient

Temporal clustering coefficient measures how interactions cluster over time, adapted from traditional clustering coefficients to incorporate temporal dynamics:

$$C_T = \frac{3 \times \text{number of temporal triangles}}{\text{number of connected triples}}, \qquad [8]$$

where a temporal triangle is a set of three nodes that interact within a short time window, and a connected triple is a set of three nodes where at least two are connected at some point in time.

Temporal Modularity

Temporal modularity evaluates the presence of distinct temporal communities within networks, defined as:

$$Q_T = \sum_{s=1}^{S} \left[\frac{L_s - (d_s^{\text{temporal}})^2}{2m} \right], \qquad [9]$$

where S is the number of communities, L_s is the number of edges within community s, d_s^{temporal} is the sum of the degrees of nodes in community s over time, and m is the total number of edges in the network over time.

Understanding these metrics offers a comprehensive view of how temporal networks evolve and function across various domains. For instance, real-world social networks often exhibit high temporal density and assortativity, indicating frequent interactions among nodes with similar temporal behaviors. Temporal modularity reflects how interactions form distinct temporal communities, highlighting patterns of cohesive interaction over time.

Integrating network-wide temporal metrics enhances our ability to analyze temporal networks, revealing nuanced patterns of interaction dynamics and structural resilience.

Python tools for Temporal Network Analysis

While truly state-of-the art solutions are sparse, several Python libraries facilitate some aspects of temporal network analysis, each offering distinct functionalities tailored to specific types of analyses and network representations. Here, we briefly introduce the most widely used libraries, highlighting their strengths and limitations to guide their selection based on analytical needs.

DyNetX is specifically designed for analyzing temporal networks and directly supports the creation, visualization, and analysis of both **snapshot-based and event-based temporal networks**. Its strengths include built-in methods for community detection, temporal centrality measures, and convenient data structures for handling temporal dynamics explicitly. However, DyNetX has fewer available resources and a smaller community compared to more established libraries.

NetworkX being widely-used, for classical (static) networks, does not inherently support temporal networks; however, temporal analysis can be approximated by applying **traditional static analysis methods to snapshot-based graphs** and subsequently aggregating the results. While NetworkX boasts extensive documentation, a large community, and versatile functionalities, its major limitation for temporal analysis is the manual implementation required to handle temporal dynamics and inability to handle continuous timescale networks without prior aggregation.

Teneto is tailored explicitly for temporal network analysis, especially in neuroscience and similar scientific contexts. It excels at analyzing time-series data and provides methods for temporal centrality, motifs, and community detection specifically optimized for **neuroscientific** applications. Despite its strong capabilities in specific domains, Teneto has a narrower focus and limited general applicability outside these areas.

PathpyG is designed for analyzing higher-order interaction sequences and temporal paths. Its core strength lies in its rigorous treatment of time-respecting paths and higher-order network representations, making it ideal for detailed **pathway and process-oriented analyses**. However, PathpyG can be somewhat complex for simpler temporal analyses and has a steeper learning curve compared to other libraries.

Choosing the appropriate library depends largely on the analysis requirements: use DyNetX for straightforward temporal network tasks and community detection, NetworkX for snapshot-based approximations and its large ecosystem, Teneto for neuroscientific applications, and Pathpy for detailed higher-order temporal path analysis.

Applied: Community Detection in Temporal Networks

The following section will detail challenges of community detection in temporal networks compared to static networks and provide students with a github repository that can be used as a basis for experimentation with a high-quality temporal network - the SocioPatterns Primary School co-presence dataset (12).

Community detection in temporal networks differs fundamentally from static networks. Since communities in dynamic environments evolve, it is insufficient to analyze isolated snapshots independently; rather, we must track how each community changes over time. To adequately capture these dynamics, an algorithm must explicitly account for events affecting communities:

1. **Dissolution**: A community disappears because it splits into multiple communities or merges into another.
2. **Continuation**: A community persists from one snapshot to the next.
3. **Splitting**: Nodes within one community separate to form distinct communities.
4. **Merging**: Multiple communities combine to form a new, larger community.

Currently, common libraries do not include easily adaptable, ready-to-use temporal community detection algorithms for snapshot-based networks. Therefore, we developed our own implementation using Python and DyNetX. Below, we outline how the algorithm operates, enabling hands-on experimentation with provided code.

Community Tracking Algorithm. Our algorithm employs the snapshot-based representation of temporal networks, allowing us to apply traditional static community detection algorithms to each snapshot sequentially. Initially, we detect communities in the first snapshot. Subsequently, for every new snapshot, communities are identified again, and temporal continuity is resolved by calculating similarities between newly detected communities and previously existing ones.

Pseudocode for Temporal Community Tracking

Input: Temporal network as snapshots $G_1, G_2, \ldots, G_T$, similarity threshold τ
Detect initial communities in snapshot G_1
for each snapshot G_t, $t = 2, \ldots, T$ **do**
    Detect communities in current snapshot G_t
    Compute Jaccard similarity between current and previous communities
    for each current community **do**
        if no similar previous community ($J < \tau$) **then**
            Mark as **new community**
        else if exactly one similar previous community ($J \geq \tau$) **then**
            Mark as **continuation**
        else if multiple similar previous communities ($J \geq \tau$) **then**
            Mark as **merger**, replace previous communities
    for each previous community **do**
        if multiple similar current communities ($J \geq \tau$) **then**
            Mark as **split**, track new communities separately
        else if no similar current community **then**
            Mark as **dissolved**

Similarity between communities is quantified using the Jaccard index, defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|},$$

where A and B represent two sets of community members (nodes). The Jaccard index ranges from 0 (no similarity) to 1 (identical sets).

The algorithm resolves community evolution using the following logic:

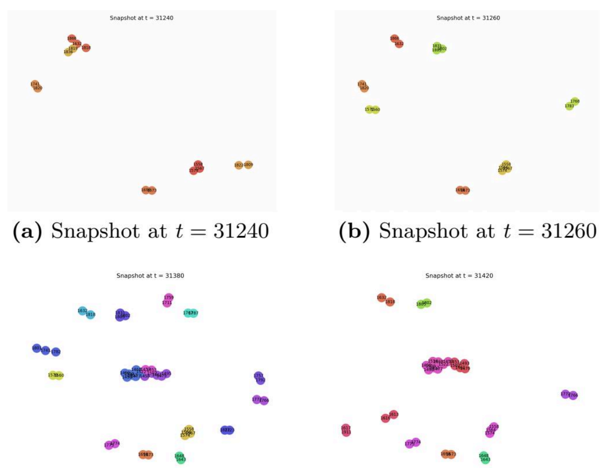
1. If a newly detected community has no previous similar community, it is treated as a newly emerged community.
2. If exactly one previous community closely matches, it represents the continuation of that existing community.
3. Multiple previous similar communities indicate a merger; the new community replaces these older communities.
4. One previous community matching several new communities indicates splitting; tracking focuses on new communities.

This process iterates for each snapshot until the full network evolution is tracked. The detailed algorithmic steps are concisely represented in pseudocode below:

Visualization of Temporal Communities. Our algorithm provides multiple visualizations to interpret community dynamics, including:

- Community sizes over time.
- Node membership timelines.
- Community evolution Sankey diagrams.
- Snapshots with nodes colored by community.

Due to readability constraints with large temporal networks, the snapshot visualization (Figure 3) is the most practical for intuitive understanding.



(a) Snapshot at $t = 31240$ **(b)** Snapshot at $t = 31260$

(b) Snapshot at $t = 31380$ **(b)** Snapshot at $t = 31420$

Fig. 3. Community detection snapshots illustrating tracking of evolving communities. Colors indicate community membership across snapshots, demonstrating continuity, splitting, merging, and re-emergence of communities.

In figure 3 we can see that our algorithm is able to track communities between different snapshots. We can see that it can even track communities that disappeared for some time period and later reappeared after a few seconds. This type of snapshot-based community visualization can be refined and potentially be used to create animations that show the development of dynamic systems over time, as demonstrated for the same dataset by Peter Holme at https://petterhol.me/2021/06/19/some-temporal-network-visualizations/.

Limitations and Practical Considerations. A critical aspect of our approach is the sensitivity to the chosen similarity threshold τ. Higher thresholds may result in overly fragmented communities, while lower thresholds risk generating overly broad "super-communities." Users are encouraged to experiment with this parameter based on the network's characteristics and analytical goals. All code and detailed implementation are accessible via our GitHub repository: https://github.com/MaticVP/INAProject.

1. Mark Newman. *Networks*. Oxford University Press, 07 2018. ISBN 9780198805090. . URL https://doi.org/10.1093/oso/9780198805090.001.0001.
2. Petter Holme and Jari Saramäki. Temporal networks. *Physics reports*, 519(3):97–125, 2012. . URL https://arxiv.org/abs/1108.1780.
3. Petter Holme. Modern temporal network theory: a colloquium. *The European Physical Journal B*, 88(9):1–30, 2015. . URL https://arxiv.org/abs/1508.01303.
4. Marya Bazzi, Mason A Porter, Stacy Williams, Mark McDonald, Daniel J Fenn, and Sam D Howison. Community detection in temporal multilayer networks, with an application to correlation networks. *Multiscale Modeling & Simulation*, 14(1):1–41, 2016. URL https://people.maths.ox.ac.uk/howison/papers/CommunityDetection2016.pdf.
5. Giulio Virginio Clemente and Diego Garlaschelli. Linking through time: Memory-enhanced community discovery in temporal networks. *Phys. Rev. Res.*, 6:043204, Nov 2024. . URL https://link.aps.org/doi/10.1103/PhysRevResearch.6.043204.
6. Petter Holme and Jari Saramäki, editors. *Temporal Network Theory*. Springer, 2023. ISBN 978-3-031-30399-9.
7. Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8(1):1–29, 2018.
8. Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.
9. Nicola Perra, Bruno Gonçalves, Romualdo Pastor-Satorras, and Alessandro Vespignani. Activity driven modeling of time varying networks. *Scientific reports*, 2(1):469, 2012.
10. Alexei Vázquez, João G Oliveira, Zoltán Dezsõ, Kwang-Il Goh, Imre Kondor, and Albert-László Barabási. Impact of memory on human dynamics. *Physical Review E*, 73.
11. John Tang, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Vincenzo Nicosia. Analysing information flows and key mediators through temporal centrality metrics. In *Proceedings of the 3rd Workshop on Social Network Systems*, SNS '10, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450300803. . URL https://doi.org/10.1145/1852658.1852661.
12. Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, and Philippe Vanhems. High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE*, 6(8):e23176, 08 2011. . URL http://dx.doi.org/10.1371/journal.pone.0023176.