Djurdjevac Conrad *et al.*
Applied Network Science (2025) 10:1
https://doi.org/10.1007/s41109-024-00687-3

Applied Network Science

RESEARCH Open Access

Detection of dynamic communities in temporal networks with sparse data

Nataša Djurdjevac Conrad[1], Elisa Tonello[2], Johannes Zonker[1] and Heike Siebert[3*]

*Correspondence:
hsiebert@math.uni-kiel.de

[1] Modeling and Simulation of Complex Processes, Zuse Institute Berlin, Takustrasse 7, 14195 Berlin, Germany
[2] Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 7, 14195 Berlin, Germany
[3] Department of Mathematics, Kiel University, Heinrich-Hecht-Platz 6, 24118 Kiel, Germany

Abstract

Temporal networks are a powerful tool for studying the dynamic nature of a wide range of real-world complex systems, including social, biological and physical systems. In particular, detection of dynamic communities within these networks can help identify important cohesive structures and fundamental mechanisms driving systems behaviour. However, when working with real-world systems, available data is often limited and sparse, due to missing data on systems entities, their evolution and interactions, as well as uncertainty regarding temporal resolution. This can hinder accurate representation of the system over time and result in incomplete or biased community dynamics. In this paper, we consider established methods for community detection and, using synthetic data experiments and real-world case studies, we evaluate the impact of data sparsity on the quality of identified dynamic communities. Our results give valuable insights on the evolution of systems with sparse data, which are less studied in existing literature, but are frequently encountered in real-world applications.

Keywords: Temporal networks, Sparse data, Dynamic communities, Temporal resolution, Temporal clustering

Introduction

Understanding complex systems through network models is a topic of great interest in a variety of domains, spanning from social sciences, archaeology and humanities, to physics, mathematics and computer science. The analysis of such networks has been shown to be crucial for uncovering properties of the underlying systems. Most prominently, the identification of *communities* within complex networks provides insights into how their structure shapes their function. Communities (often referred to as clusters or modules) represent subsets of nodes within a network that exhibit stronger intra-group connections compared to inter-group connections. These communities encapsulate meaningful organizational structures of the underlying complex system, indicating relevant structural and functional units. They help delineate distinct entities and subsystems, revealing fundamental patterns of organization (Newman and Girvan 2004; Cherifi et al. 2019; Fortunato and Newman 2022; Newman etal. 2006).

In networks that are subject to changes in time, identification of communities and their evolution is pursued as a means to describe and understand important shifts and

network events (Rossetti and Cazabet 2018; Rojas et al. 2021). When the available data have a temporal dimension, multiple instances of a network representation are defined, one for each time point of interest, obtaining a so-called *temporal network* (often referred to as time-evolving or dynamic network) (Holme and Saramäki 2012), where nodes and edges can appear or disappear, or edge weights can change. The addition of a temporal dimension poses significant additional challenges for the community detection problem. First, the objective becomes the identification of groups of nodes that not only constitute significant clusters in separate instances, but also possibly reveal a coherent structure that traverses layers. The resulting communities have their own dynamics that includes their appearance or disappearance in time, as well as splitting into several or merging with other communities. There exists a plethora of approaches available for temporal clustering (Masuda and Lambiotte 2016; Cherifi et al. 2019), including approaches for clustering multilayer networks[1] Mucha etal. (2010); Boccaletti et al. (2014); Rossetti and Cazabet (2018); Huang et al. (2021). Despite some recent reviews (Cazabet et al. 2020), there is still a great need for comparison and more systematic clarification of precise strength and weaknesses of different techniques. This is especially desirable as network definitions grow in abstraction and detach from the meaning of the underlying data, making the interpretation of results in the specific application area more and more challenging. A second challenge of dealing with temporal data is that capturing dynamic evolution of a system requires tracking the system's state across multiple temporal instances and possibly across different temporal scales. Obtaining such information for all entities within the system and their interactions can be exceedingly costly and in many cases infeasible, as temporal data for complex systems are often sparse and incomplete, especially when studying historical or natural systems. For example, archaeological networks, that are constructed from data, represent only parts of complete networks characterizing the system, but these can not be fully recovered (Kim and Leskovec 2011; Daems et al. 2024). Incomplete records, gaps in observations and uncertainties in temporal information make it challenging to reconstruct the system's temporal evolution accurately. This, in turn, can lead to incomplete or biased community identification in associated networks. Recently, several studies have focused on how missing information in temporal networks impacts the identification of time-respecting paths and flow-based communities (Smiljanić etal. 2020). However, a comparative study on how sparse data affects the dynamic community detection is largely missing.

In this work, we want to give a contribution in this direction, by evaluating the efficacy of different community detection methods with respect to availability of data on network structure and temporal resolution. More precisely, we study how sparsity of data on nodes, edges and information about their "evolution", influences the quality of dynamic communities. We start by following the ideas from Cazabet et al. (2020), where different algorithms for clustering temporal networks are compared on a benchmark set. To capture different levels of stability in the network structure over time, we use two distinct approaches to generating temporal networks. While the first allows only for gradual structural modifications, the second enables us to more closely investigate

[1] Temporal networks are special cases of multilayer networks.

networks that can exhibit significant structural changes from one time point to the next. The two scenarios are also reflected in the case studies we consider, where we juxtapose the results on synthetic data with findings for two real-world case studies.

The paper is organized as follows: In Sect. 2 we start by introducing the background and notation on temporal networks and dynamic communities. Then, in Sect. 3 we conduct a quantitative comparison between established methods for community detection using synthetic data experiments. We study the impact of missing data on two real-world case studies in Sect. 4. Finally, in Sect. 5, we discuss open questions and point out possible directions for future research.

Methods

In this section, we introduce the basic terminology of temporal networks and dynamic communities. We have in mind application areas where the varying edges represent connections or relationships between the entities that can change in time, and that are potentially only the result of partial observations at given time snapshots. As our focus is on temporal networks that are constructed from sparse data, we first present approaches that construct network structures incorporating different types of missing information. Next, we study the community structures underlying these temporal layers and their changes over time. We summarize essential ideas behind the main paradigms and the measures for evaluating their performance when applied to networks with sparse data.

Setting the scene

Temporal networks (also often called dynamic and time-evolving networks) have been researched in a variety of contexts, see for example (Mucha etal. 2010; Boccaletti et al. 2014; Rossetti and Cazabet 2018; Huang et al. 2021) and references therein. Here, we view a temporal network as an ordered set of temporal layers, each represented by undirected graphs sharing the same set of nodes. We define a temporal network $G = \{G_1, \ldots, G_L\}$ by a set of L graphs $G_l = (V, E_l)$ defined at each temporal layer $l = 1, \ldots, L$. In particular, we assume that all layers consist of the same set V of n nodes and that the edges E_l in each layer l can be either unweighted or weighted. The (weighted) adjacency matrix of a layer l is denoted by $W^l = (W^l_{ij})_{i,j=1,\ldots,n}$ and it encodes information about the connections between the nodes.

Various approaches for anylyzing static networks have been translated to the context of temporal networks. The problem of finding communities in temporal networks has certainly gained a lot of attention and new approaches are continuously being explored and proposed, see (Rossetti and Cazabet 2018) for a recent review. Many of these approaches reduce the problem to finding communities in static networks, by either looking at clusters in separate layers or by combining the information from several layers into one big network. From a plethora of approaches for clustering static networks, modularity-based approaches are among the most popular ones because of their intuitive definition, good performance and simple implementation. They are based on optimizing a modularity score of a partition $C = \{c_1, \ldots, c_r\}$, which is a measure of the significance of the density of edges inside communities versus the density expected in a graph where edges are chosen at random:

$$Q(C) = \frac{1}{\sum_k s_k} \sum_{i,j} \left(W_{ij} - \frac{s_i s_j}{\sum_k s_k} \right) \delta(c(i), c(j)),$$

where $s_i = \sum_{j=1}^{n} W_{ij}$ is the weighted degree of a node i and $c(i)$ is the community of i. Many variants of the definition of modularity and additional algorithmic approaches have been suggested, see (Fortunato 2010) for an overview. Here, we will use the Louvain method (Blondel et al. 2008), a greedy optimization algorithm that can find satisfactory solutions in reasonable time in real-world networks. It is important to note that despite its popularity, modularity optimization suffers from several critical limitations (Fortunato and Barthelemy 2007; Ghasemian et al. 2020; Good et al. 2010; Guimera et al. 2004; McDiarmid and Skerman 2013; Peixoto 2023), which concern for example the resolution limit, meaning that the method struggles to detect small communities within large networks, even when these communities are well-defined (Fortunato and Barthelemy 2007) and the problem of overfitting, which can lead to the identification of high-scoring communities even in fully random networks (Guimera et al. 2004; Ghasemian et al. 2020). Despite these limitations, modularity optimization is still widely used and many studies focus on investigating its performance in different scenarios. Here we will also use Louvain-based approaches for finding network communities, building on the method comparison as introduced in Cazabet et al. (2020), putting the focus on the effects of data sparsity.

Constructing networks with missing data

When modeling real-world systems, we often face the problem of missing data, that occurs due to partial observations, limitations in data collection and other factors. In many cases, the extent of data absence remains unclear, posing challenges in quantifying uncertainties in such systems and using approaches for network structure analysis, dynamic modeling and clustering. Here, we focus on the following types of missing data:

(a) *missing edges*: occurring in the absence of observed interactions between entities over time;
(b) *missing nodes*: occurring when not all entities in the system can be observed over the whole time period or at some points in time;
(c) *missing temporal information*: happening due to interruptions in data collection or when it is not possible to accurately estimate temporal information from the data.

In order to test how (a-c) can affect different temporal clustering approaches, we will introduce techniques to account for each of them.

Missing edges: To understand the influence of partial knowledge on edges in a temporal network, we consider all edges from all layers as distinct entities and thus allow for repeated entries of edges that appear in more than one layer. Then, we sample a particular fraction of edges r from this set and subsequently reassign each edge to its original layer. This means that the fraction r of sampled edges is not necessarily uniformly reflected in each layer. The number r represents the fraction of the full information that is available for analysis. The procedure is illustrated in Fig. 1b. In the benchmarks we discuss in Sect. 3, we will test the effect of $r = 0.7$ and $r = 0.3$, i.e. when 70% and 30% of

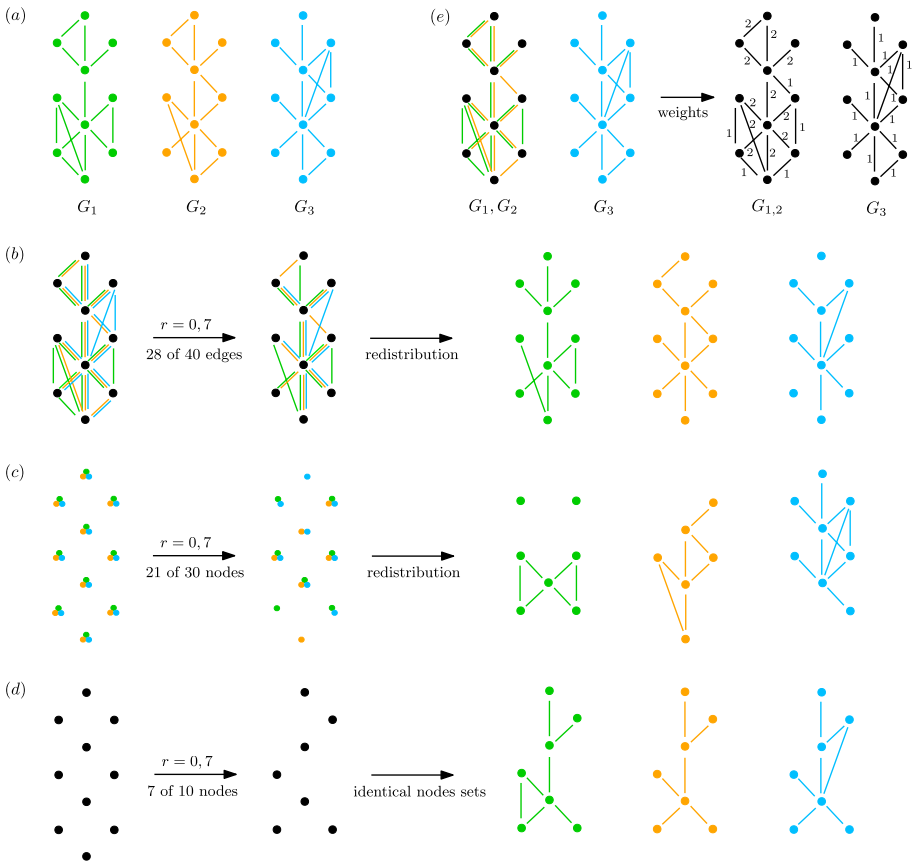

Fig. 1 In **a** a temporal network with three layers, ten nodes and forty edges combined. In **b** edge sampling by sampling uniformly from the combined edge set (left) and then assigning the sampled edges to their original networks (right). In **c** node sampling by sampling uniformly from a set of nodes generated by assigning nodes layer identifiers (variable node sampling). Nodes are then reassigned to their original layers, edges incident to non-sampled nodes are deleted, as is the case in (**d**), but here nodes are sampled from the node set without layer identifiers (fixed node sampling). In **e**, the first two layers are aggregated. In the resulting temporal network (right), edge weights reflecting edge abundance in the merging are added and in networks that do not undergo merging all edges are labeled with 1

the original edge information is available, or in other words when 30% and 70% edges, respectively, are missing. Clearly, there are other possible approaches to edge sampling, but we observed no substantial difference in the general trends for the relative performance of clustering methods in different scenarios as discussed in Sect. 3.

Missing nodes: We discuss two different ways of sampling nodes to account for possible scenarios of missing nodes. In the first case, we consider instances of the same node from different layers as distinct nodes, and sample a fraction r of nodes over the resulting set of $n \cdot L$ nodes with repeated entries. The layers of the resulting dynamic network have only edges that are incident to the sampled nodes. This scenario accounts for situations where information on different nodes is missing in various points in time and we will thus refer to it as *variable node sampling*. In the second case, we simulate complete absence of information about some nodes during the whole period of time. To achieve this, we sample a fraction r of nodes from the set of n nodes of the dynamic network, keeping the same sample of nodes in all layers. We will refer to this scenario as *fixed*

node sampling. Both scenarios are illustrated in Fig. 1c and d. Note that in these scenarios both nodes and edges are missing, since with the deletion of a node its incident edges are also deleted. In this respect, node sampling yields more severe scenarios of missing information than edge sampling. While mixed node and edge sampling is certainly a further possibility, the differences are rather gradual and so we restrict ourselves to the above presented scenarios.

Missing temporal information: In this case, we want to mimic absence of detailed knowledge about the correct time resolution. We achieve this by merging a number of consecutive layers into one, obtaining in the end a temporal network with a smaller number of layers than the original network. Merging of two layers is done by including all nodes and all edges that are present in any of the two layers. Edge weights in the resulting network are obtained as the number of instances of the edge in the temporal layers that got merged. This procedure leads to loosing a precise temporal component of merged edges. In the limit of all layers being merged together, we obtain a static network. The procedure is illustrated in Fig. 1e. Note that merging of layers can also be used as a tool in the network analysis to consolidate information from several layers to potentially counteract the effects of incomplete structural data. We call the method of merging all layers temporal aggregation and we will discuss it in more detail below.

Clustering of temporal networks

Similar to the problem of clustering static networks, many approaches exist for clustering dynamic networks. Comparison of these approaches is difficult, because their efficacy varies across different contexts due to the rich nature of networks and their characteristics. Following the ideas in Cazabet et al. (2020), where such comparison has been done, we focus on the three methods that performed slightly better in many contexts: label-smoothing, no-smoothing and smoothed-graph. We include the method of temporal aggregation and we give a brief overview of these methods to test their performance in the context of networks that are sparse due to the missing data.

Here, all of these approaches adopt the Louvain method for finding the communities in static networks (Blondel et al. 2008). While testing these approaches, we found that varying parameters can yield slightly better performances in specific cases, but we did not observe general rules to guide parameter selection based on network characteristics. Therefore, following the approach in Cazabet et al. (2020), we use the same default parameter values without further tuning for each of the clustering methods as implemented in the library `tnetwork` ((https://github.com/Yquetzal/tnetwork)). In the following, we present the main ideas of these methods.

Temporal aggregation Temporal aggregation is performed by merging all layers as described above, i.e. such that the resulting network is weighted and the edge weights correspond to the number of instances of these edge during the whole time period. This means that temporal information gets completely lost, as edges from different layers get aggregated to form a static network. There are clear disadvantages of this approach when applied on temporal networks with many edge and community changes over time. However, in examples where the communities are not changing drastically, this approach gives good results and is therefore often used in practice (Krings et al. 2012), as it is simple to implement and computationally efficient.

No-smoothing In this approach, clusters are found in each layer independently using the static Louvain community detection algorithm (Newman and Girvan 2004). Then, communities in different layers are matched using a similarity measure. Here, we adopt a one-step matching approach (Cazabet et al. 2020), where clusters at time layer l are compared against clusters at layer $l-1$. Given two sets of communities $\{A_1,\dots,A_r\}$ and $\{B_1,\dots,B_s\}$, the input for the problem is a non-negative matrix of size $r \times s$ containing in cell ij a measure of similarity between clusters A_i and B_j. A natural choice for a similarity measure is the Jaccard index

$$J(A_i,B_j) = \frac{|A_i \cap B_j|}{|A_i \cup B_j|}.$$

To keep the approach as simple and general as possible, as in Cazabet et al. (2020) the assignment of each label is set to a maximum of one cluster. The no-smoothing method is simple, computationally efficient, and well-suited for networks with abrupt community changes. However, it can miss gradual changes in community structure over time, leading to inconsistencies across layers and reduced accuracy in capturing long-term trends (Sattar et al. 2023). For more sophisticated approaches, e.g. multistep matching, see (Lorenz-Spreen et al. 2018) and references in Rossetti and Cazabet (2018).

Smoothed-graph The smoothed-graph method is a modification of the no-smoothing approach, where information on communities from previous layers is used to create a more gradual and consistent evolution of communities over time. We consider here a variant of the method (Guo et al. 2014) as described in Cazabet et al. (2020), where the communities are found at each layer l using the community partition at the preceding layer $l-1$. This is done by using a memory parameter α, that determines the importance of the previous partition, and a modified adjacency matrix

$$\widetilde{W}^l_{ij} = \alpha W^l_{ij} + (1-\alpha) C^{l-1}_{ij},$$

where $C^{l-1}_{ij} = 1$ if nodes i and j belong to the same community in layer $l-1$ and otherwise $C^{l-1}_{ij} = 0$. Then, matching of communities is performed as in the no-smoothing approach. In the benchmarks, the memory parameter α is set to 0.9 as in Cazabet et al. (2020). Smoothing does increase computational costs compared to the no-smoothing approach, but this method has much better performance for networks where communities change gradually (Sattar et al. 2023).

Label-smoothing This method, introduced in Falkowski and Spiliopoulou (2007), performs the same clustering at the layer level as no-smoothing. Next, it uses a different matching procedure to label the communities, by clustering a graph having the communities as nodes that are connected by weighted edges reflecting the similarity between any pairs of communities. Weights of these edges are obtained as Jaccard values between the corresponding communities. With this approach, temporal clusters are determined based on a more global view than in the cluster matching utilized in the methods above. The latter only takes adjacent layers into account when matching clusters while the cluster graph considered here captures similarity of clusters even if their temporal distance is large.

Evaluation of network communities

In order to evaluate the validity of a clustering method, the communities that this method finds need to be compared to the ground-truth communities. Unfortunately, in most real-world examples the ground-truth communities are generally not available, which poses a big challenge for validation and applicability of existing methods to real-world data. Benchmark generators, on the other hand, often can be designed to provide information on the ground-truth communities along with synthetically generated networks.

There exist many different approaches for comparison of partitions, see (Fortunato 2010; Warrens and Hoef 2022) for recent reviews. Here, we focus on Adjusted Rand Index (ARI) and Adjusted Mutual Information (AMI), that are refined versions of two popular measures: the Rand Index (RI) and Mutual Information (MI), respectively (Rand 1971; Hubert and Arabie 1985; Xuan etal. 2010). RI measures the similarity between the two partitions by calculating the proportion of node pairs that are either in the same cluster or in different clusters in both the given and reference partitions. RI yields a value between zero and one, where higher RI value indicates a more similar clustering, thus closer to the ground-truth. MI quantifies the shared information between two partitions, reflecting the degree to which the assignment in one partition can inform on the assignment in the other partition. Similar to RI, higher MI values suggest a closer resemblance between the partitions. ARI and AMI additionally implement adjustments that normalize the RI and MI scores for the possibility of random agreement.

Computing the average ARI or AMI over the network layers gives an idea of the accuracy of community detection at the static level, but does not account for the dynamical aspect of the structure. We therefore use, as in Cazabet et al. (2020), the *longitudinal* versions of these scores, called LARI and LAMI. These are calculated by considering all layers at the same time, viewing instances of the same nodes in different layers as distinct nodes. One single ARI or AMI is thus calculated for the dynamic network, accounting for the possible persistence or change in the community assignments of nodes. To evaluate the accuracy of communities obtained after layer merging, for each bin of size i, the resulting partitions are repeated i times, once for each layer in the bin, to allow the comparison with the ground-truth partitions. Since we did not find any significant difference in the impact of sparsity between LAMI and LARI scores in the examples we considered, in the following we will only show results for mutual information. For illustration, a plot reporting results based on LARI in the first scenario considered below can be found in Appendix A.

Dynamic communities in synthetic networks

We start by conducting experiments on synthetic networks to compare how the different clustering approaches, namely aggregation, no-smoothing, smoothed-graph and label-smoothing as described above, perform depending on the level of data sparsity. In this setting, the evaluation of results is facilitated since the ground truth is known, which is often not the case for real-life data sets. In a first scenario, we utilize the temporal network generating software included in `tnetwork` described in Cazabet et al. (2020), which enables the creation of network structures that progressively

Table 1 Parameters considered for benchmarks generation

tnetwork (Cazabet et al. 2020)	**multilayerGM (Jeub 2019)**
$\mu \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$	$\mu \in \{0.0, 0.1, 0.2, 0.4, 0.6, 0.8, 0.9, 1.0\}$
n. communities $m = 10$	n. communities $m = 10$
community size $5 \leq s \leq 15$	n. nodes $n = 100$
n. operations $o \in \{20, 50\}$	n. layers $l = 100$
delay $d \in \{20, 0\}$	stability $p \in \{0.99, 0.9\}$
sample size $r \in \{0.7, 0.3\}$	sample size $r \in \{0.7, 0.3\}$
bin size $i \in \{10, 100\}$	bin size $i \in \{2, 10\}$

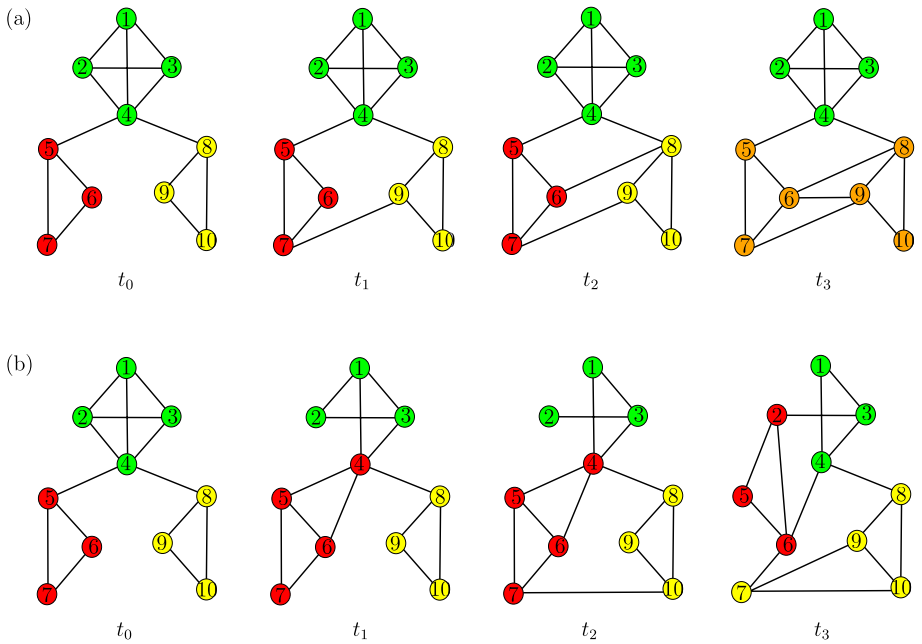

Fig. 2 In **a** an idealized illustration of a temporal network resulting from the `tnetwork` approach with "merge" as set event. Changes in network structure are gradual, slowly edges are added until in time step t_3 a new cluster appears. In **b** an illustration of the `multilayerGM` creating an instable scenario with edges getting redistributed less gradually. Nodes can more easily change cluster affiliation (from t_0 to t_1) and overall considerable changes are possible (from t_2 to t_3)

evolve according to scenarios consisting of a sequence of events. Several clustering methods for dynamics networks are tested in Cazabet et al. (2020) on generated networks and compared to identify relative strengths and weaknesses in relation to different measures (static and longitudinal scores, smoothness and scalability). We can thus start our investigation in a well-tested setting yielding sensible choices for the involved parameters. To broaden our considerations, we then include a second scenario focusing on less stable communities employing the generating software `multilayerGM` introduced in Bazzi et al. (2020). The potential differences in the resulting networks are illustrated in Fig. 2 and explained in more detail in the following sections. The parameter choices for both benchmarks are summarized in Table 1 and will be further discussed in the following sections. The detection of dynamic clusters

and computation of scores is in all cases done via the implementation provided in the `tnetwork` library.

Benchmark generation with `tnetwork`

The first benchmark generator we use is implemented in the `tnetwork` library (https://github.com/Yquetzal/tnetwork). This generator was introduced for the purpose of comparing the accuracy and performance of algorithms designed to detect evolving communities in dynamic networks. Here we give a brief outline of the approach, for the purpose of illustrating the parameter choices. We refer the reader to the original paper for a detailed description. The graphs are generated to have a defined community structure that changes according to a given scenario description, defined by a series of *events* that can modify the community assignment of nodes. The authors define a set of possible events (including merge, split, birth and death) each with some associated parameters. For instance, the *delay* parameter allows to customize the waiting time before an event starts after it is triggered. The sequences of events define a scenario for the evolution of the community structure.

Networks with communities defined by the given scenarios are then generated using a variation of the stochastic block model (Cazabet et al. 2020; Granell et al. 2015). The aim is to generate communities that are subject to events, but are slowly evolving, meaning that only a small number of edge changes happen at each step to implement the evolution from the structure before the event to the structure after the event, as illustrated in Fig. 2a. The number of layers in the network is therefore not imposed, but rather grows with the chosen number of operations, delay parameter and size of the communities. The internal and external densities of the communities are modulated by a community strength parameter μ, where lower values of μ indicate stronger communities, i.e. higher internal and lower external densities. The networks are generated as undirected and unweighted.

For our experiments, we create test networks with the scenarios and setting of the benchmark generator used in Cazabet et al. (2020). Starting with a given number of clusters, the evolution of the community structure is described by a series of merge or split operations. For generating the first type of synthetic networks, we use the same parameters adopted in Cazabet et al. (2020), creating networks with 10 communities with minimum and maximum size 5 and 15 respectively, with 20 merge or split operations, and with a delay parameter set to 20. To test a scenario with higher variability in the community structure, we create a second set of networks with 50 operations. It should be noted that the speed of change in community structure does not vary, and a higher number of operations results instead in an increased number of layers. To limit the growth in the number of steps, in this second case we set the delay parameter to zero. The resulting dynamic networks have about 1000 and 2500 layers respectively. The parameter μ defining the community strength varies between 0.05 and 0.5. We run each experiment 20 times and show the average scores.

When we perform aggregation in our analyses, we use the method `aggregate_sliding_window` of the `DynGraph` class. The network layers are aggregated as weighted graphs, meaning that the graphs obtained have weighted edges, where the

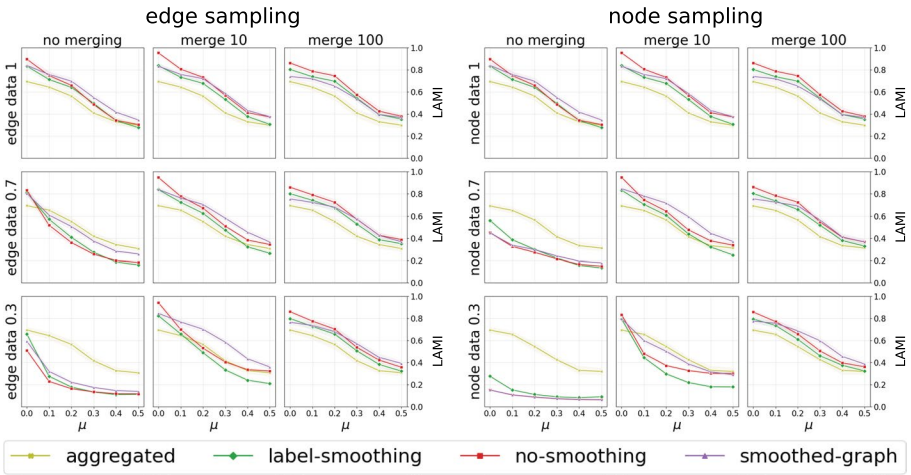

Fig. 3 LAMI obtained on benchmarks generated with `tnetwork`, setting the number of operations to 20 and the delay to 20, under different sampling and merging scenarios

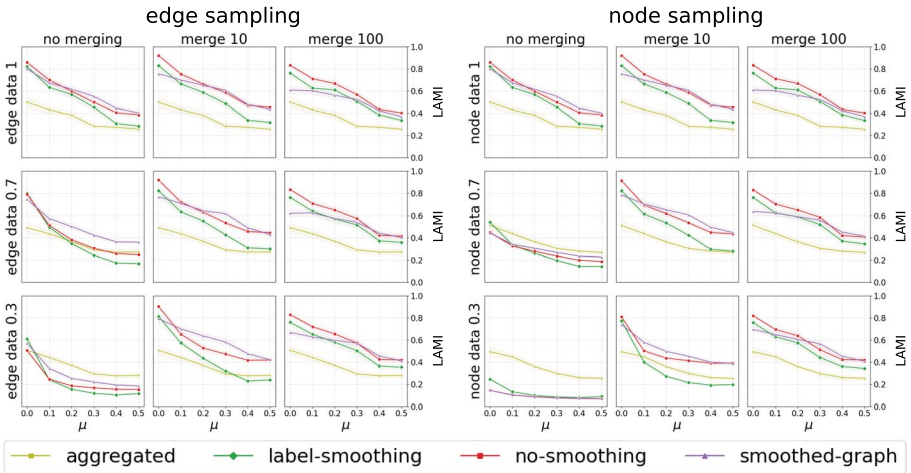

Fig. 4 LAMI obtained on benchmarks generated with `tnetwork`, setting the number of operations to 50 and the delay to 0, under different sampling and merging scenarios

weight of an edge counts the number of instances of the edge in the time bin. We use bin sizes equal to 10 and 100 for benchmarks generated with `tnetwork`.

Results Figs. 3, 4 and 5 summarize the results obtained on the benchmark networks generated with `tnetwork`, in the sampling and merging scenarios described in Sect. 2.2. Figure 3 shows scores for networks generated with 20 operations, while for Fig. 4 we asked for a higher number of operations, while setting the delay to the minimum, as described. Thus, the latter can be seen as a scenario with less stable communities over time than the first. In addition to the accuracy for the methods we mentioned, the plots display the score obtained by aggregating the full dynamic network as described in Sect. 2.3. This can be seen as the extreme case with respect to the bin sizes for aggregation, since the full temporal data set is aggregated. It gives a reference point to compare the results to, to get an idea of whether, as the data available on the underlying network decreases, the methods are still able to capture the dynamic aspect of the community

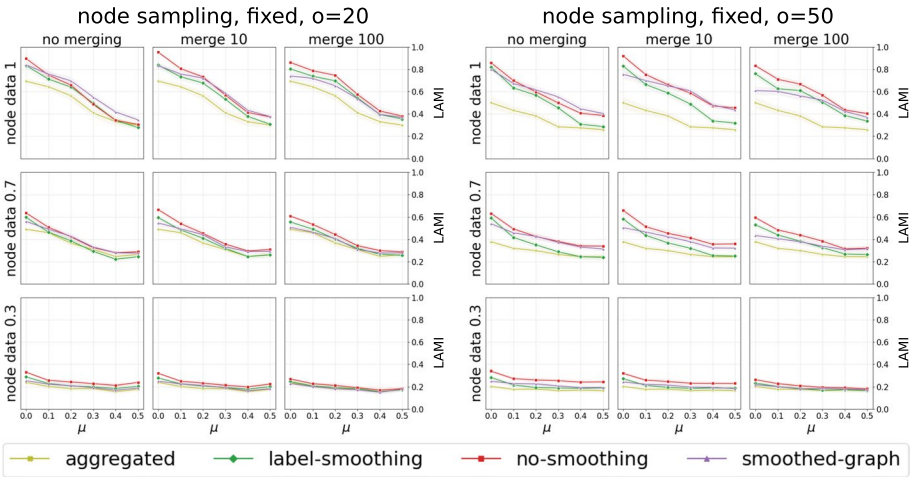

Fig. 5 LAMI obtained on benchmarks generated with `tnetwork`, setting the number of operations to 20 and 50 and fixed node sampling

structure, or whether it would be more beneficial to simply collect and use all the available information, disregarding the temporal dimension.

First we can confirm some general trends observed in Cazabet et al. (2020) on the comparative performance of the methods in terms of longitudinal scores. The accuracy of the methods is quite similar, with no-smoothing tending to outperform the other methods for low values of μ, while the other methods can sometimes do better on the other side of the spectrum.

Effects of sampling and merging Let us look at the effects of sampling and merging, starting with variable edge and node sampling, i.e., different edges and nodes are sampled in each layer. By looking at how the accuracies change in the first column of the plots for edge resp. node sampling in Fig. 3, we can observe that the scores quickly drop below the accuracy obtained by aggregating all layers, with no-smoothing being the most impacted method. The effect is more prominent for node sampling where nodes are removed, as is to be expected, since the removal of nodes, carrying with it the removal of incident edges, has a more drastic effect on the graphs.

Looking, on the other hand, at the plots for both scenarios from left to right, we see the accuracies under increasing levels of layer aggregation, corresponding to the scenario simulating missing temporal information, i.e., not having the full temporal resolution. Here, the curve for full aggregation naturally does not change. We can observe from the first row that the accuracies for all other methods remain steadily high, or even improve for the method no-smoothing, so there is no significant effect of loosing the higher resolution. In addition, when sampling data at increasing levels, we can see how the merging of layers can strongly compensate for the absence of information. So lowering the temporal resolution can consolidate information of adjacent layers yielding better clustering results. Interestingly, full aggregation often performs better than the other methods employed without any aggregation (resp. first columns), but is out-performed if at least some layer aggregation is conducted. This shows that the more restricted aggregation is capable of preserving some more information on the temporal communities that might get lost when aggregation is employed too intensively. We will discuss the

importance of choosing suitable bin sizes in more detail in the later sections on empirical instead of synthetic data.

Lastly, as a sanity check for our results, we consider the scenarios where nodes are eliminated from all layers (Fig. 5, left). In other words, the information inherent in the deleted nodes is completely lost and cannot be recovered from earlier or later layers. As expected, the effect of layer aggregation is absent, independent of the sampling rate. There is no benefit in merging data from different layers - no method can recover data that is simply not there.

Increased network variability One important observation of our first study is that merging, even at moderate levels of layer aggregation, has the potential to improve accuracy and thus cushion the effects of incomplete data. To clarify this result, we need to see it in the context of the specifications of the generated data. In this first data set, we produced rather stable networks, utilizing not too many events and a delay in event execution that further contributes to the overall stability. This fact is also reflected in the scores obtained by clustering the dynamic network as an aggregate single static network (yellow line), which are quite high especially when the communities are strong (low levels of μ).

To look at networks with higher levels of variability, we generate benchmarks with 50 operations, and set the delay for the triggering of operations to zero (Figs. 4 and 5, right). We can see that the accuracy obtained with full network aggregation (yellow lines) significantly drops. All the trends observed in Fig. 3 however still apply here in Fig. 4. Moderate levels of network aggregation keep the scores to high levels.

Delving a bit deeper into this aspect, we see that even in the second setting the overall community structure is quite stable. The benchmark generator of Cazabet et al. (2020) was simply designed to generate slowly evolving community structures, and increasing the number of operations has a limited effect on the network stability. To circumvent this restriction and obtain a clearer picture of the effect of community stability over time on the results, we thus employ a second network generator allowing for high levels of instability.

Benchmark generation with `multilayerGM`

We now consider a second benchmark generator, introduced in Bazzi et al. (2020) and available in `multilayerGM` (Jeub 2019). This generator does not implement community events, and allows instead to define dependencies between layers, which are specified via so-called inter-layer dependency tensors. In the temporal case, the dependency of a layer's partition is based on the layer directly preceding it. The stability of the dynamic community structure can be varied by setting parameters that define, for each layer, the probability of copying the community assignment of a node from the assignment of the same node in the preceding layer. Here we only generate temporal networks with uniform dependencies, so that the degree of similarity between layers is captured by a single parameter $p \in [0, 1]$. This allows us to explore the effect on accuracy scores of different speeds of change for the community structure, although the change does not concern possible events such as merge or split, but rather only the assignment of nodes to communities. Contrary to the other benchmark generator, the number of layers is one of the parameters set by the user. One can therefore experiment with different

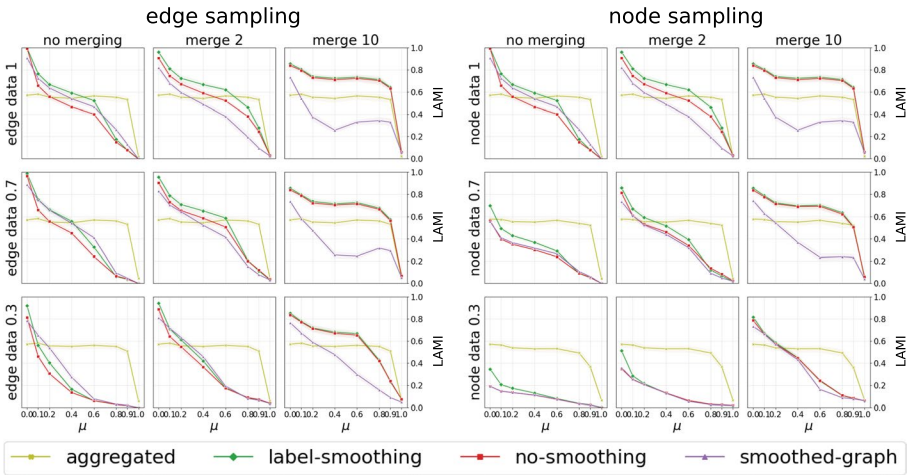

Fig. 6 LAMI obtained on benchmarks generated with `multilayerGM`, setting p to 0.99, under different sampling and merging scenarios

combinations of speeds of change and number of steps, observing, for example, how a relatively small change can compound over many layers, resulting in a structure that is more difficult to capture via approaches that rely strongly on layer aggregation. With probability $1-p$ the community is sampled from a *null distribution* which can be specified by the user. Thus we can generate temporal networks that change much more quickly then those generated by `tnetwork`, as illustrated in Fig. 2b. Community assignments for the first layer are always sampled from the null distribution. For the latter we use the implementation of a symmetric Dirichlet distribution offered by the framework. This is parametrized on the number of communities (or *mesosets*) m and the concentration parameter θ. For $\theta = 1$, the null distribution is sampled uniformly at random among the categorical probability distributions with m categories, so that the sizes of the clusters are heterogeneously distributed. Large values of θ favor uniform distributions, which lead to communities that are similar in size, and on the opposite side values of θ that are close to zero produce unbalanced distributions and cluster sizes. To generate a multilayer network for a given partition, this framework also uses a variant of a previously formulated stochastic block model generator. Similarly to the case of the benchmark generator previously discussed, one can specify a value for the community-mixing parameter $\mu \in [0, 1]$.

For our experiments, we set the parameter θ to 1. We fix the number of communities to 10 and generate networks with 100 nodes, to create scenarios comparable to those generated with `tnetwork`. Here, we have to choose the number of layers, which we set to 100. This number is significantly smaller than the average number of layers resulting from the other benchmark generator, but are more than enough for us to observe a large variability in the community structure. We create dynamic networks that are very stable, taking $p = 0.99$, and a second family of networks using $p = 0.9$. For this generator, we consider values of μ between 0 and 1, since for this benchmark communities should be detectable for any value below 1 (Bazzi etal. 2016). Adapted to the smaller number of layers, we then choose bins sizes 2 and 10 for the merging of layers. We run each experiment 20 times. The plots presented in Figs. 6, 7 and 8 show the average scores.

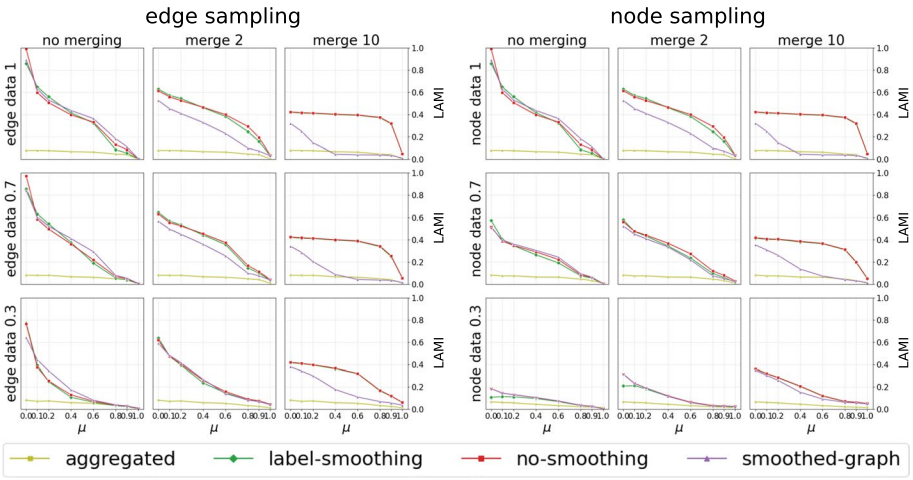

Fig. 7 LAMI obtained on benchmarks generated with `multilayerGM`, setting p to 0.9, under different sampling and merging scenarios

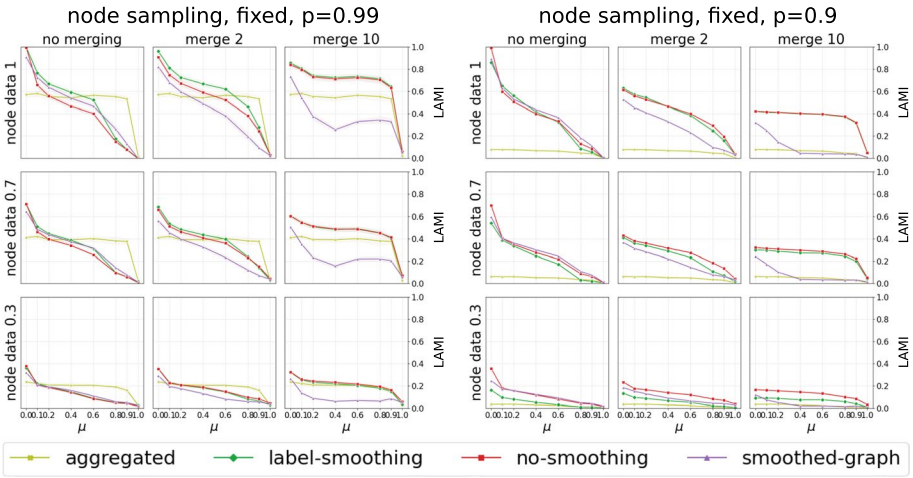

Fig. 8 LAMI obtained on benchmarks generated with `multilayerGM`, setting p to 0.99 or 0.9, and with fixed node sampling

Results We first look at networks generated with a high stability level ($p = 0.99$), so that we can compare more easily to the scenarios generated with `tnetwork`. The plots in Figs. 6 and 8, left, display results in similar fashion to the ones presented previously. As in the case of the other benchmark generator, for variable edge and node sampling, layer merging has a positive impact on accuracy in particular for no-smoothing, limiting the drop in accuracy that otherwise occurs as the amount of available data is decreased. With respect to the previous benchmark generator, we can observe, in general, a stronger decrease in accuracy for smoothed-graph and more stable scores for the other methods, as the strength of the communities decreases and as the size of merged bins increases.

With a modest drop in stability, taking $p = 0.9$, we find a steep decline in accuracy when clustering the aggregation of all layers (Figs. 7 and 8, right, yellow lines). Lack of temporal resolution has a much stronger effect than before. For all clustering

methods, there is still a positive effect of layer aggregation to counteract missing structural information, but it is more modest.

The two scenarios generated with `multilayerGM` thus allow us to sharpen our results from the first benchmark study. Again we can observe the beneficial effect of merging layers and thus aggregating information to possibly counteract layer-specific loss of information. However, we see that the result quality depends on the stability of the communities over time. In a stable setting, the information present in earlier and later layers is well-suited to approximate the information in the current layer and thus can compensate for incomplete data there. This effect is severely restricted if adjacent layers differ significantly as is the case in the last setting we investigated exhibiting strongly unstable communities. Here, merging might very well blur the distinct structures appearing in one bin and lead to poor results. This effect can be counteracted if additional information on the stability of communities through time is available, such as information on time points where strong outside influences act on the network and may give rise to more pronounced changes in the network structure. Such knowledge can then be utilized to postulate and test bin sizes that, on the one hand, give rise to the beneficial effects of aggregation to counteract data sparseness and, on the other hand, do not destroy distinct temporal substructures reflecting points of high instability. We will illustrate this point in a more concrete scenario in Sect. 4.2.

Dynamic communities in networks generated from empirical data

In this section we explore the results of considered clustering approaches on temporal networks arising from real-world systems. We study the effect of varying level of data sparsity on both stable and dynamic clusters. We start with a primary school dataset (Stehlé et al. 2011; Gemmetto et al. 2014), that is characterized by everyday, long interactions between pupils of the same class that form clusters and by short, random interactions between pupils from different classes. In the second example we examine dynamic cluster behaviour, where the groups of hunter-gatherers form clusters that are changing over a long period of time, governed by their movement in the geographical landscape and other environmental factors of Central Africa (Zonker et al. 2023). In both of these examples, we face the typical problem of real-world datasets, where the ground truth is not known, but has to be inferred for validation purposes. To overcome this issue, available metadata is very often used as a ground truth. However, real-world networks can correspond to different sets of metadata, both observed and unobserved, which makes the choice of only one (or some) of these arbitrary and unjustified (Peel et al. 2017). When ground truth on the system is unavailable, metadata certainly represents a valuable resource that can be used to learn more about the network's underlying organizing principles. In the absence of a useful reference partition, we have to devise alternative ways of evaluating the goodness of detected partitions (see Sect. 4.2).

Primary school data

We consider the primary school contact dataset (Stehlé et al. 2011; Gemmetto et al. 2014), that was made available by the SocioPatterns collaboration (http://www.sociopatterns.org/). The dataset contains records of contacts between individuals (pupils and teachers), detected during approximately 17 h spread over two days. The dataset

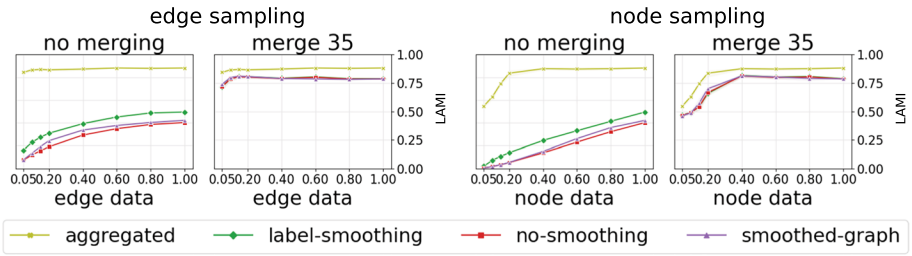

Fig. 9 LAMI obtained with four clustering methods applied on primary school contact data interpreted as a dynamic network

has been extensively studied in different contexts (see e.g. Gauvin et al. 2014; Génois and Barrat 2018; Arruda et al. 2021; Bovet et al. 2022; Contisciani et al. 2022). It comes equipped with metadata which identifies each individual as either a teacher or a pupil, and provides the class attended by each pupil. Specifically, there are 10 classes, with two classes for each grade from grade 1 to grade 5. As discussed above, this classification may not represent the ground truth (Peel et al. 2017). Nevertheless, we will use it as the reference partition in communities and run the dataset through the same setting that we applied to benchmark networks in previous sections, calculating standard goodness scores. The reference partition consists therefore of 11 clusters, two for each grade, plus one for the teachers, as considered in other studies (Gauvin et al. 2014; Génois and Barrat 2018; Arruda et al. 2021; Bovet et al. 2022; Contisciani et al. 2022; Sattar et al. 2023). Children in the same class spend a significant amount of time together, however there are breaks where children from some different classes can mix. It should also be noted that we do not expect teachers to be identified as a separate group, given that the majority of their time is spent in class with children (Stehlé et al. 2011). In absence of alternative strong choices, we use the separation in 10 classes plus teachers as reference partition, like it is used in other references (Gauvin et al. 2014; Génois and Barrat 2018; Arruda et al. 2021; Bovet et al. 2022; Contisciani et al. 2022; Sattar et al. 2023).

The community partition we use as a reference is not dynamical. More sophisticated analysis of the dynamics of communities in this dataset are of course possible; see for instance (Contisciani et al. 2022), where a flow stability approach providing "backward" and "forward" partitions at different time resolutions highlights for example some information about the time spent together by different classes. In contrast, the analysis here is limited by the choice of comparing the results to a reference partition. With this limitation, we can primarily observe how strongly the interaction data reflects the partition into classes and teachers, and how loss of information impacts the ability to retrieve clusters that are as close as possible to the available metadata.

By considering intervals of 15 min, we can create a dynamic network with 70 layers, where an edge connects two individuals if these are in contact at least once in the interval. In this network we apply the same clustering methods as the ones from the previous sections, and we analyze and contrast the results based on varying levels of data sparsity. This includes scenarios of edge and node sampling, with sampling size between 0.05 and 1, as well as comparisons made between unmerged data and merged data with a time interval length of 35.

The results for the LAMI score are shown in Fig. 9. Layer merging achieves higher accuracies as the number of merged layer increases, approaching the accuracy obtained by interpreting the full dataset as a static network. Similarly, aggregation achieves results with accuracies of about 80%, by producing 8 clusters: the two classes in grade 3 are grouped together, the two classes in grade 5 are grouped together, and the remaining classes fall into separate clusters. Teachers are not detected as a separate community. The grouping of the two grade 3 classes and the two 5 grade classes is not too surprising, given the amount of contact between the two subclasses (Stehlé et al. 2011). The partitions obtained in Contisciani et al. (2022) also cluster some pairs of classes in the same grade together. We can observe that moderate variable node and edge sampling have little effect due to the redundancy in the data. As observed for the benchmarch networks, the impact on LAMI is more significant in the scenarios of fixed node sampling. This example demonstrates that if the clusters do not change in time, then, for the purpose of identification, there is no advantage in studying the available data as a dynamic network. In such cases aggregation of time-layers gives the highest LAMI scores and accuracies improve as more layers are being merged. Incorporating additional information on the data, for example accounting for breaks when children from different classes can mix, might, of course, lead to better results. Here we do not use any additional information.

Data on mobility of African hunter-gatherers

For the next example, we consider the real-world setting of hunter-gatherer communities in Central Africa. We will analyze a dataset on the residential mobility of hunter-gatherers generated by an agent-based model (Zonker et al. 2023). For the mobility dynamics of this model, environmental influences were taken into account, most importantly the suitability of the land. To assess the suitability of an area for hunter-gatherers, an Environmental Niche Model (ENM) was constructed that explains the likelihood of the presence of contemporary hunter-gatherers by environmental and climatic factors such as temperature and vegetation (Padilla-Iglesias et al. 2023). Through paleoclimatic reconstructions (Padilla-Iglesias et al. 2022) for up to 120000 years before present (BP), it is possible to use the ENM to estimate the likelihood for the presence of hunter-gatherers and thus the suitability of the landscape in ancient times. The spatial resolution of our data is ca. 10km $\times$ 10km per grid cell. The temporal resolution of the estimated suitability landscapes follows the temporal resolution of the paleoclimatic reconstructions and changes every 2000 (simulated) years in the considered time period of 116000-110000 BP, i.e. there are three different suitability landscapes. While also the friction of the landscape and distances between camps influence the mobility of the model and thus the spatial distribution of the agents, the suitability is the most important factor for the emergence of agent clusters. As the suitability changes with time, so does the clustering of the agents.

In addition to mobility dynamics, the ABM of Zonker et al. (2023) also includes interactions between agents to model the exchange of information and culture. To determine which agents can communicate with each other, a (temporal) interaction network is constructed based on the current positions in each time step. For the network construction we can consider different radius sizes, to account for short- and long-range interactions between agents. The choice of the interaction radius determines the structure of

the interaction network, in general a smaller radius leads to more sparse networks. For the example, we consider a total of 45 equidistant ABM snapshots within the selected time period and distinguish between two cases for network construction. In both cases, we use a radius R for the network construction that is much larger than the interaction radius used in the original model, so that the networks of the snapshots will be dense enough.

Case A: For the choice of a small radius R, only short-range interactions between agents are possible, i.e. interaction between agents that are close in space. The resulting interaction network is consisting of many small clusters of agents that change in time driven by agents' movements in space. The positions of these clusters are determined by the minima of the suitability landscape, which indicate the most suitable areas. The choice of small R imposes a decomposition of regions with many minima into smaller clusters, see Fig. 10.

Case B: When considering large values of R, we take into account long-range interactions, i.e. interactions between agents that are further apart in space. The resulting interaction network consists of larger clusters that are more stable in time than clusters in Case A, as they are resistant to the small perturbations coming from movements of individual agents. Here, cluster changes are driven by the global changes of the landscape in time, see Fig. 11.

In the following, we will apply temporal clustering to networks in Case A and Case B, for different levels of data sparsity that will be coming from: (1) Reduced number of available agents (nodes and their edges); (2) Reduced temporal resolution. Since information about ground-truth communities is missing in this example, we will not use the index scores from section 3. Instead, we will discuss qualitative results. From the approaches presented in Section 2.3, here we will use the no-smoothing method to detect dynamic clusters. This is necessary because we use only a few static snapshots of the system, rather than having continuously changing positions of the agents. However, since we are focusing on the methods, we will not discuss further how this affects the anthropological interpretability of the clustering results.

Results Case A: First we analyse the networks generated with radius $R = 10$ grid cells, which corresponds to 100 km, and consider cases with different temporal sparsity, see Fig. 10. For single time slices ("no merging"), we are able to identify the agent clusters around the minima of the suitability landscape. When merging every 15 time-slices during which the landscape is fixed, we find a reduced number of clusters, corresponding to regions with more frequent spatial transitions between suitable areas. If the merging of time slices is not done uniformly, i.e. the first 30 time slices are merged and then the last 15 time slices, we observe a loss of information on clusters in the first and the second suitability landscape, as these get effectively aggregated and only a separation in 2 big clusters is captured. Finally, when all slices are merged together, which is equivalent to time-aggregation, we observe a big loss of information on granularity, as the smaller clusters get merged into 2 main clusters, that are separated well in space and time and robust even to major changes in the landscape. Between these main clusters, agent transitions in the underlying ABM are rare events.

Case B: Next, we analyze networks generated with $R = 50$ grid cells, which corresponds to 500 km, and compare different choices of temporal sparsity, see Fig. 11. For

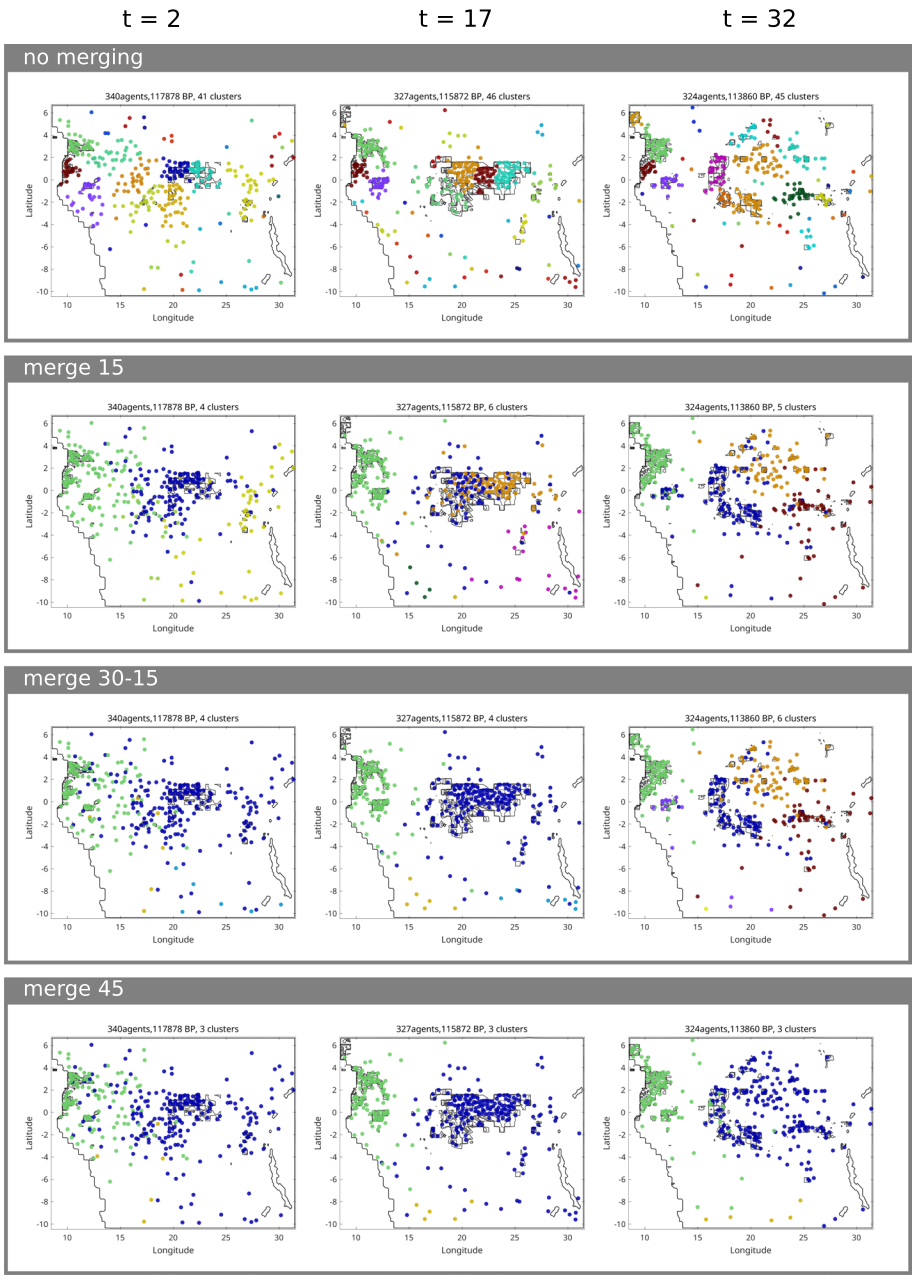

Fig. 10 Simulation snapshots of the hunter-gatherer model for $R = 10$ at different times $t = 2, t = 17$ and $t = 32$. Nodes have spatial position in the time-dependent suitability landscape that is visualized by contour lines. Color of nodes corresponds to the cluster they belong to. Network edges are not shown for better visualization

single time slices, we get a clustering on the spatial scale similar to Case A with the merging of 15 time slices corresponding to one landscape. These clusters correspond to regions with more frequent agent interactions within. With the larger radius, the spatial resolution is coarser and the interaction network already connects multiple suitable areas within one time slice. However, we can observe some differences in the clusterings comparing the spatial aggregation of this case and the temporal aggregation of the

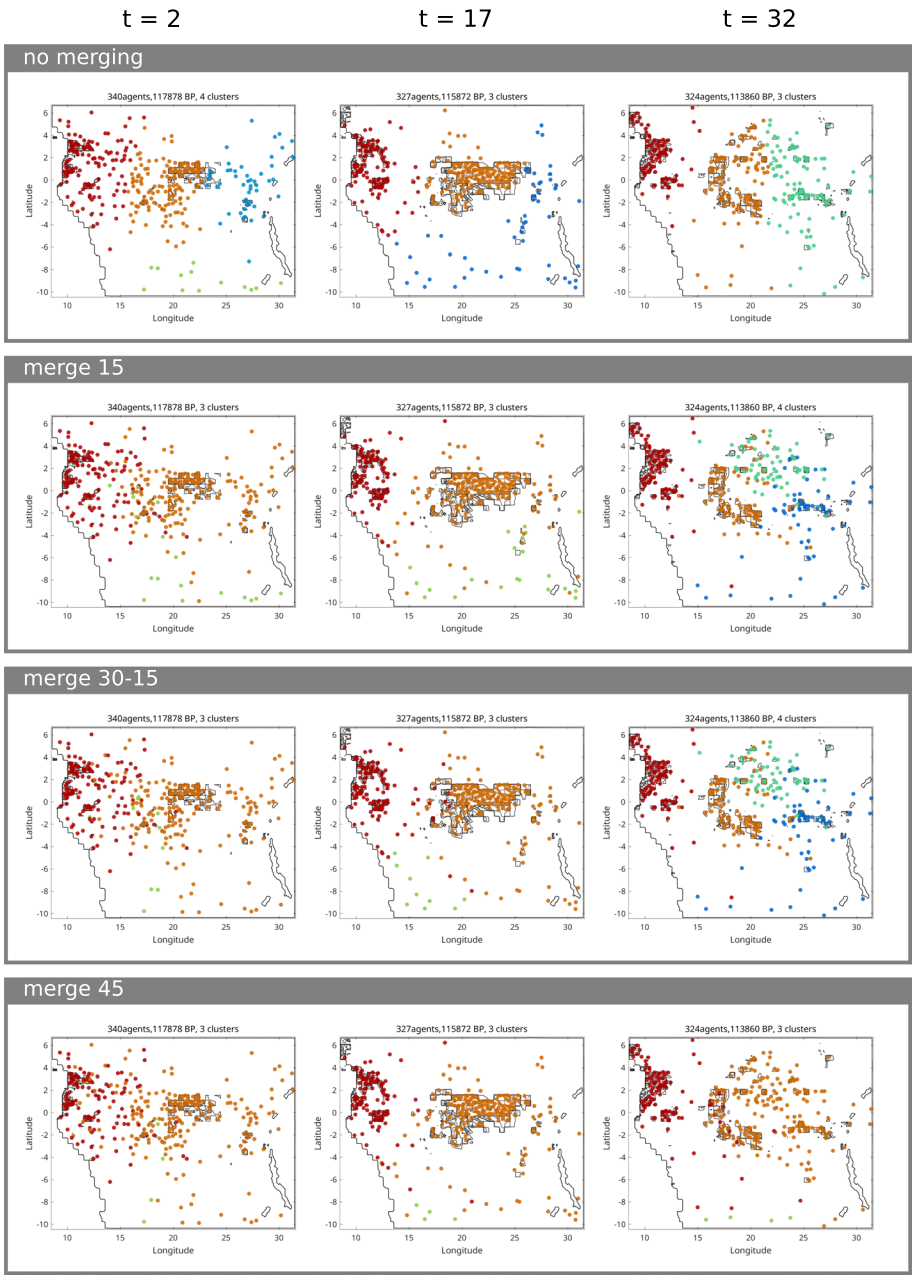

Fig. 11 Simulation snapshots of the hunter-gatherer model for $R = 50$ at different times $t = 2, t = 17$ and $t = 32$. Nodes have spatial position in the time-dependent suitability landscape that is visualized by contour lines. Color of nodes corresponds to the cluster they belong to. Network edges are not shown for better visualization

last, e.g., for the slice $t = 17$ the orange cluster in the middle is not separated in Case B. Merging blocks of 15 time slices with $R = 50$ that correspond to the same fixed landscape leads to the following changes when compared to the single time slices: (1) At t=2, where the orange (middle) cluster merges with the blue one, due to agents equilibrating in the middle well and grouping together. (2) At t=32, we obtain now 4 clusters as agents get distributed to the new wells. It is interesting that in one case (namely (1)),

information about granularity is lost, while in the other (2), it is recovered. This is due to the additional position data from later time slices of each time period being closer to the equilibrium distribution associated with the landscape. Unlike in Case A, here we do not see big differences when merging blocks of 15 time slices versus when the first 30 and then the last 15 time slices are merged. This is due to the choice of a large radius, spatial proximity and similarity of the two main wells in this suitability landscape. Merging all slices (i.e. merge 45), which corresponds to time-aggregation, results in a loss of temporal resolution of clusters. However, like in Case A, it still enables the identification of the two main clusters consisting of agents that are well separated in space and time across all time-frames.

Finally, we study the effects of sampling nodes. We discuss both the case of varied node sampling, where the nodes in each time slice are sampled (see Fig. 12) and the case of sampling a set of fixed nodes that is removed for all time slices (see Fig. 13). As before, sampling of nodes has the effect that only edges that are incident to the sampled nodes are kept in the network. Note that in this example we do not explicitly consider sampling of edges, as the available data for this system is about nodes, i.e. position of the hunter-gatherer camps, and these positions determine the possible (interaction) edges.

For the varied node sampling, we randomly select 30% of the agents in each snapshot. In both Case A and Case B, we are able to identify the most prominent clusters in the most suitable regions. However, in Case A, we identify fewer clusters overall compared to no sampling, and the larger clusters that are more widely distributed and not centered around a local minimum are split into smaller clusters. Since we have many isolated agents that are identified as their own cluster in the case of a small radius, it makes sense that we have fewer clusters when we consider only a fraction of the agents due to sampling. For the more widespread structures in the data, the small radius increases the likelihood that some connecting agents will be missing, splitting the already sparse interaction network into more smaller connected components. In the setting of Case B, this effect is limited, as the larger interaction radius leads to more stable interaction networks. If we combine node sampling with frame merging in both case A and case B, we can identify the clusters of the corresponding merging scenario without sampling.

For the fixed node sampling, we randomly select 30% of the agents and consider their positions for all time slices. For individual snapshots. the results differ not that much from the varied node sampling (see Fig. 13). This is to be expected, as consistently selecting the same agents to be missing in each snapshot has no effect in this case. On the other hand, the detection of cluster evolution is better in the fixed sampling case, where nodes from one layer can be consistently matched to nodes in the next layer, allowing, as a consequence, the reconstruction of cluster dynamics. This effect is more prominent when the radius is larger, especially at low sampling values, since larger clusters have increased chances of matching. In the merging case we are able to identify the major clusters but not all of the smaller clusters that are identified when we merge the snapshots without sampling. This can be explained by the mobility dynamics of the model, as agents move considerably within the suitable areas. Thus, while there may be some missing connections in one snapshot, this is not necessarily true for the next snapshots, even if the same agents are selected. However, in the case that all agents of a smaller cluster c are missing it is unlikely

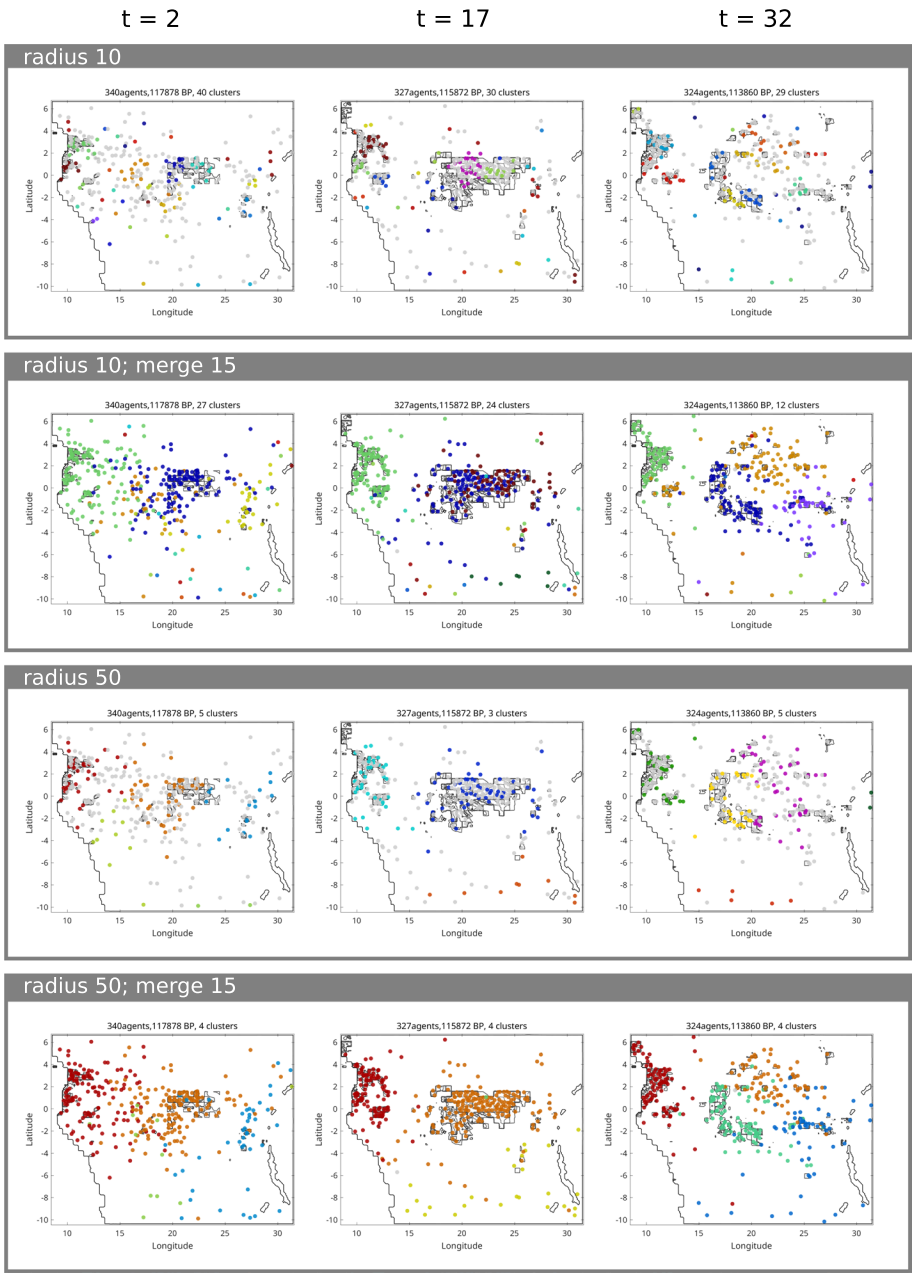

Fig. 12 Simulation snapshots for varied node sampling where 30% of nodes are randomly selected in each snapshot. Top two rows: $R = 10$, Bottom two rows: $R = 50$. Nodes have spatial position in the time-dependent suitability landscape that is visualized by contour lines. Color of nodes corresponds to the cluster they belong to. Network edges are not shown for better visualization

that multiple agents that have been chosen through the sampling will transition through unsuitable terrain to the suitable area corresponding to c. In this case we would not be able to recover the information about the cluster from the data.

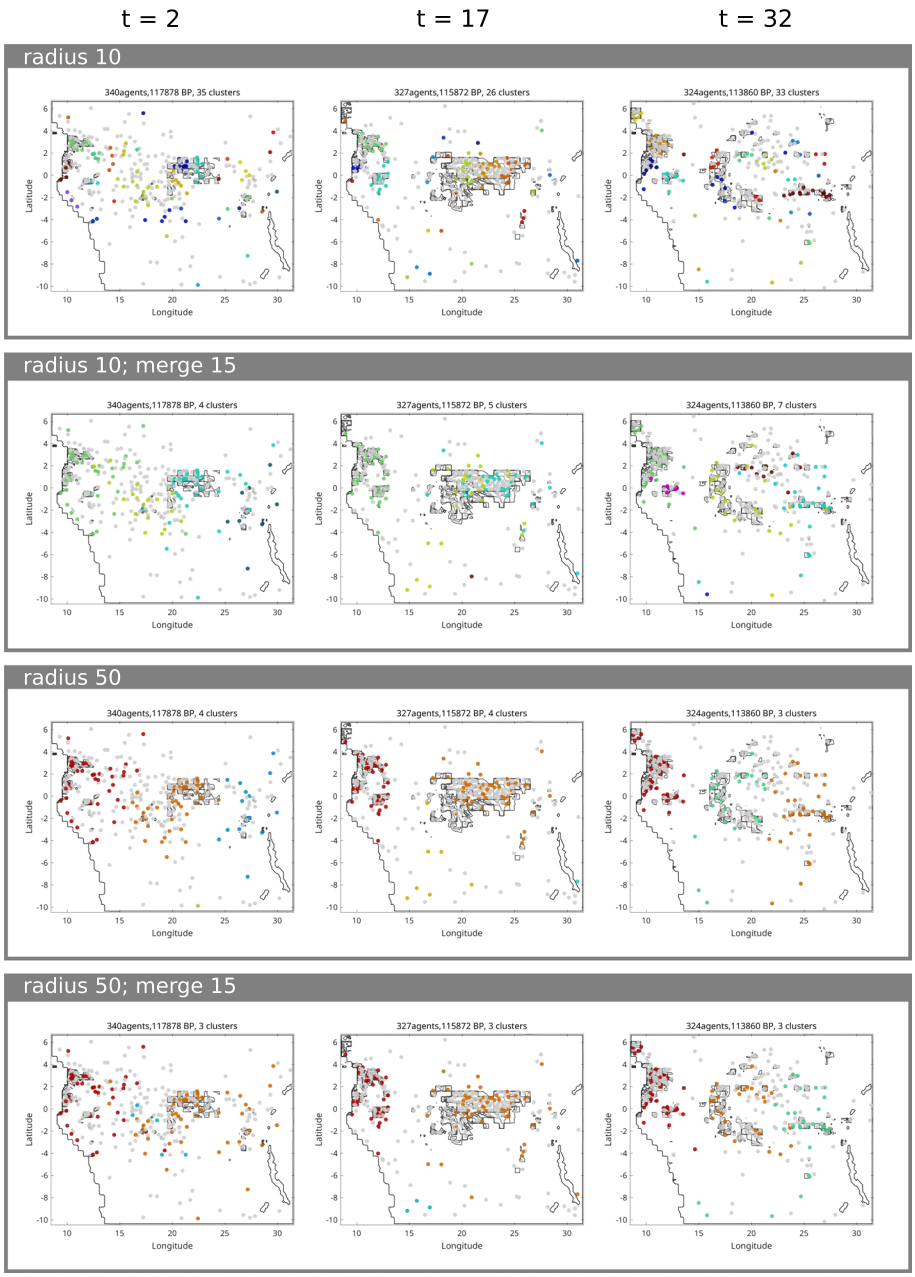

Fig. 13 Simulation snapshots for fixed node sampling where 30% of nodes are randomly selected for all time slices. Top two rows: $R = 10$, Bottom two rows: $R = 50$. Nodes have spatial position in the time-dependent suitability landscape that is visualized by contour lines. Color of nodes corresponds to the cluster they belong to. Network edges are not shown for better visualization

Conclusions

Temporal network modeling has shown a lot of potential to capture essential features of complex time-evolving systems. Detecting dynamic communities is certainly a central

analysis tool in this context. However, the temporal component often exacerbates the difficulties modelers are facing in terms of the restricted availability of data on the considered systems. This includes lack of data on network structure through time as well as uncertainty regarding temporal resolution. Here, we evaluated the impact of structural and temporal data sparseness, first on synthetic data and then on application-based case studies. We considered several different well-established methods for dynamic community detection with a focus on discovering trends in the impact of data loss on performance quality and approaches to mitigating that impact.

For investigating the effect of incomplete structural information we considered both scenarios with missing edges as well as with missing nodes via the variable sampling. The observed trends are consistent - the effects of missing nodes being expectedly stronger, since a node deletion can induce several edge deletions. We see that low levels of this type of data loss do not impact result quality significantly. To help evaluate the significance of levels of deficit in the data, we included scores obtained by aggregating all available data, thus creating a reference to establish whether aiming at the reconstruction of a dynamic structure has become unrealistic. As expected, substantial data loss has severe effects on the dynamic community detection methods, to the point where full aggregation of all layers, i.e., completely disregarding temporal information, outperforms the temporal methods, at least for networks changing gradually over time. The levels of data loss at which a dynamic reconstruction appears to be unfeasible seem to depend on the community strengths and to a smaller degree on the method.

We investigated the effect of aggregation in more detail, taking a dual view on it. First, we use merging of a certain number of subsequent temporal layers to model the loss of information on the temporal resolution of the system. This type of information loss is decidedly detrimental for highly variable networks that exhibit drastic structural changes. Full temporal aggregation performs particularly badly in this setting. Conversely, effects are rather weak for slowly evolving networks with fairly stable temporal communities. Here, we can observe that results of the dynamic community detection methods even benefit from aggregating layer information. This leads us to consider merging of layers from a second view point, as method to compensate for structural data loss. As mentioned above, full aggregation performs better than the dynamical methods when facing significantly incomplete data in rather stable systems. Since the community structure in such systems does not change much from layer to layer, the loss of data from one layer can be compensated by information on other layers in the network. Our results show that this effect can be even better exploited by coupling a more judicious merging still allowing to capture temporal changes with dynamic community detection methods. We can then counteract lack of information of one layer with information captured in adjacent layers while still preserving a degree of temporal resolution that allows to track changes in the community landscape. Good results are to be expected if the merging bin size can be chosen according to the rate of change in the dynamic system. Clearly, for highly unstable systems the effects of missing information will still be much more problematic. For such systems only small bin sizes can be chosen, meaning there is

only restricted possibility to compensate data loss, since otherwise the drastic structural changes over time will blur the temporal communities.

Choosing appropriate bin sizes for analyzing longitudinal data poses a significant challenge across disciplines, including archaeology, where it can impact interpretations of historical events. Additional to identifying appropriate bin sizes that reflect the time-resolution of the system, this issue also occurs on the methodological level within existing temporal network clustering approaches. For instance, in smoothed graph based approaches, the challenge is to determine the so-called memory parameter, which defines the connection weight between temporal layers. In the methods using matching procedures, it is not clear how to choose the memory parameter corresponding to the importance weights of clusterings from different layers. The choice of this parameter can significantly affect the granularity and precision of obtained results. One potential solution could be to develop a change point detection approach that would help to identify points in time where significant shifts occur within the data, and use these to adaptively adjust bin sizes. Furthermore, integrating additional knowledge about the system could help in inferring time ranges where the system is rather stable and time points where more drastic changes are expected to occur. We have illustrated such a scenario in our last case study, where information about changes in the landscape are not explicitly included in the network model, but they do impact the evolution of the clusters. However, these ideas are out of the scope of this paper and will be explored in the future.

An additional topic that we did not consider here concerns the question of detectability, that is, of whether it is theoretically possible to design an algorithm that can recover the community structure. The theory of detectability for community identification has gained significant attention in recent years, in particular with results and conjectures predicting precise thresholds for the feasibility of the community detection problem for some stochastic block models (Abbe 2018). For the static network case, the work of Decelle et al. (2011a, 2011b) described a phase transition for the community detection problem, identifying, with methods of statistical physics, the threshold level as a function of the number of communities and their strength. In Ghasemian et al. (2016) these types of predictions were extended to a dynamic network case, also based on a stochastic block model, considering the strength of the community structure as well as its rate of change. The work identified a limit that in the static case coincides with that of Decelle et al. (2011a, 2011b), while taking into account the probability of changing labels between layers. The analysis is limited to a specific dynamic stochastic block model, and, to understand if a similar one applies to the benchmarks we considered, the process for the estimation of the threshold would need to be adapted and verified, to account for the difference in the definitions. Further work could then investigate whether such analyses can be extended to the identification of thresholds as functions not only of the strength and rate of change of the community structure, but also of the level of data availability.

Appendix A : Additional plots

See Fig. 14

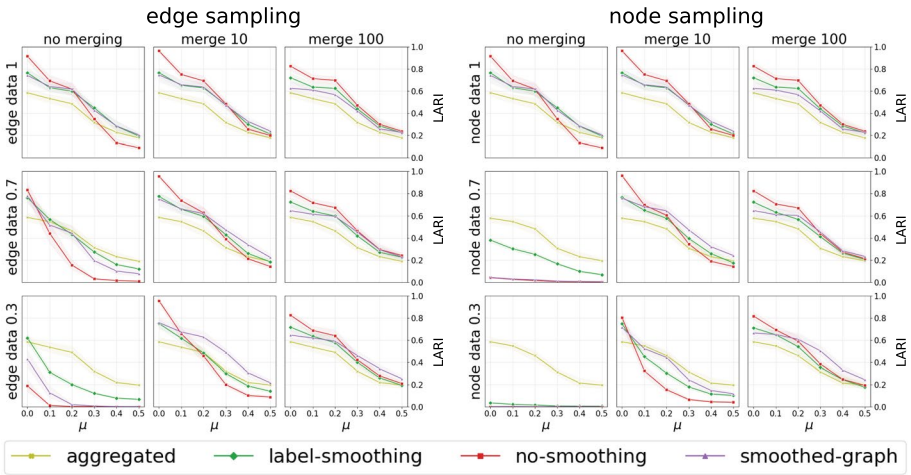

Fig. 14 LARI obtained on benchmarks generated with `tnetwork`, setting the number of operations to 20 and the delay to 20, under different sampling and merging scenarios

Acknowledgements
We would like to thank Filip Blašković for valuable insights about the manuscript.

Author contributions
N.D.C., E.T. and H.S. designed research; N.D.C., E.T., J.Z. and H.S. performed research; N.D.C., E.T., J.Z. and H.S. designed simulation experiments; E.T. and J.Z. implemented software; N.D.C., E.T., J.Z. and H.S. wrote the paper. All authors read and approved the final manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL. This work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy through grant EXC-2046 ***The Berlin Mathematics Research Center MATH+*** (project no. 390685689).

Availability of data and materials
The primary school contact dataset (Stehlé et al. 2011; Gemmetto et al. 2014) was made available by the SocioPatterns collaboration under the following link: http://www.sociopatterns.org/. The hunter-gatherer data was generated by an agent-based model Zonker et al. (2023) and is available under Zonker et al. (2023).

Declarations

Competing interests
The authors declare no Conflict of interest.

Received: 14 August 2024 Accepted: 19 December 2024
Published online: 07 January 2025

References

Abbe E (2018) Community detection and stochastic block models: recent developments. J Mach Learn Res 18(177):1–86
Arruda G, Tizzani M, Moreno Y (2021) Phase transitions and stability of dynamical processes on hypergraphs. Commun Phys 4(1):24
Bazzi M, Jeub LG, Arenas A, Howison SD, Porter MA (2020) A framework for the construction of generative models for mesoscale structure in multilayer networks. Phys Rev Res 2(2):023100
Bazzi M, Jeub LG, Arenas A, Howison SD, Porter MA (2020) Generative benchmark models for mesoscale structure in multilayer networks. Phys. Rev. Research 2, 023100
Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Statist Mech Theory Exp 2008(10):10008
Boccaletti S, Bianconi G, Criado R, Del Genio CI, Gómez-Gardenes J, Romance M, Sendina-Nadal I, Wang Z, Zanin M (2014) The structure and dynamics of multilayer networks. Phys Rep 544(1):1–122
Bovet A, Delvenne J-C, Lambiotte R (2022) Flow stability for dynamic community detection. Sci Adv 8(19):3063
Cazabet R, Boudebza S, Rossetti G (2020) Evaluating community detection algorithms for progressively evolving graphs. J Complex Netw 8(6):027
Cherifi H, Palla G, Szymanski BK, Lu X (2019) On community structure in complex networks: challenges and opportunities. Appl Netw Sci 4(1):1–35

Contisciani M, Battiston F, De Bacco C (2022) Inference of hyperedges and overlapping communities in hypergraphs. Nat Commun 13(1):7229
Daems D, Coco E, Gillreath-Brown A, Kafetzaki D (2024) The effects of time-averaging on archaeological networks. J Archaeol Method Theory 31(2):473–506
Decelle A, Krzakala F, Moore C, Zdeborová L (2011) Inference and phase transitions in the detection of modules in sparse networks. Phys Rev Lett 107(6):065701
Decelle A, Krzakala F, Moore C, Zdeborová L (2011) Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. Phys Rev E-Statist Nonlinear Soft Matter Phys 84(6):066106
Falkowski T, Spiliopoulou M (2007) Data mining for community dynamics. Künstliche Intell 21(3):23–29
Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174
Fortunato S, Barthelemy M (2007) Resolution limit in community detection. Proc Natl Acad Sci 104(1):36–41
Fortunato S, Newman ME (2022) 20 years of network community detection. Nat Phys 18:1–3
Gauvin L, Panisson A, Cattuto C (2014) Detecting the community structure and activity patterns of temporal networks: a non-negative tensor factorization approach. PloS One 9(1):86028
Gemmetto V, Barrat A, Cattuto C (2014) Mitigation of infectious disease at school: targeted class closure vs school closure. BMC Infect Dis 14(1):1–10
Génois M, Barrat A (2018) Can co-location be used as a proxy for face-to-face contacts? EPJ Data Sci 7(1):1–18
Ghasemian A, Zhang P, Clauset A, Moore C, Peel L (2016) Detectability thresholds and optimal algorithms for community structure in dynamic networks. Phys Rev X 6(3):031005
Ghasemian A, Hosseinmardi H, Clauset A (2020) Evaluating overfit and underfit in models of network community structure. IEEE Trans Knowl Data Eng 32(9):1722–1735. https://doi.org/10.1109/TKDE.2019.2911585
Good BH, De Montjoye Y-A, Clauset A (2010) Performance of modularity maximization in practical contexts. Phys Rev E-Statist Nonlinear Soft Matter Phys 81(4):046106
Granell C, Darst RK, Arenas A, Fortunato S, Gómez S (2015) Benchmark model to assess community structure in evolving networks. Phys Rev E 92(1):012805
Guimera R, Sales-Pardo M, Amaral LAN (2004) Modularity from fluctuations in random graphs and complex networks. Phys Rev E-Statist Nonlinear Soft Matter Phys 70(2):025101
Guo C, Wang J, Zhang Z (2014) Evolutionary community structure discovery in dynamic weighted networks. Phys A Statist Mech Appl 413:565–576. https://doi.org/10.1016/j.physa.2014.07.004
Holme P, Saramäki J (2012) Temporal networks. Phys Rep 519(3):97–125
Huang X, Chen D, Ren T, Wang D (2021) A survey of community detection methods in multilayer networks. Data Min Knowl Discov 35(1):1–45
Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218. https://doi.org/10.1007/BF01908075
Jeub LGS (2019) A Python framework for generating multilayer networks with planted mesoscale structure . https://github.com/MultilayerGM/MultilayerGM-py
Kim M, Leskovec J (2011) The network completion problem: Inferring missing nodes and edges in networks. In: proceedings of the 2011 SIAM international conference on data mining, pp. 47–58. SIAM
Krings G, Karsai M, Bernhardsson S, Blondel VD, Saramäki J (2012) Effects of time window size and placement on the structure of an aggregated communication network. EPJ Data Sci 1(1):4. https://doi.org/10.1140/epjds4
Lorenz-Spreen P, Wolf F, Braun J, Ghoshal G, Djurdjevac Conrad N, Hövel P (2018) Tracking online topics over time: understanding dynamic hashtag communities. Comput Soc Netw 5(1):1–18
Masuda N, Lambiotte R (2016) A guide to temporal networks. World Scientific, Singapore
McDiarmid C, Skerman F (2013) Modularity in random regular graphs and lattices. Electron Notes Discret Math 43:431–437
Mucha PJ, Richardson T, Macon K, Porter MA, Onnela J-P (2010) Community structure in time-dependent, multiscale, and multiplex networks. Science 328(5980):876–878
Newman ME, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69(2):026113
Newman M, Barabási A-L, Watts DJ (2006) The structure and dynamics of networks. Princeton University Press, Princeton. https://doi.org/10.1515/9781400841356
Padilla-Iglesias C, Atmore LM, Olivero J, Lupo K, Manica A, Arango Isaza E, Vinicius L, Migliano AB (2022) Population interconnectivity over the past 120,000 years explains distribution and diversity of central african hunter-gatherers. Proc Natl Acad Sci 119(21):2113936119
Padilla-Iglesias C, Grove M, Blinkhorn J (2023) Ecological drivers of hunter-gatherer lithic technology from the middle and later stone age in central africa. Quater Sci Rev 322:108390
Peel L, Larremore DB, Clauset A (2017) The ground truth about metadata and community detection in networks. Sci Adv 3(5):1602548. https://doi.org/10.1126/sciadv.1602548 (https://www.science.org/doi/pdf/10.1126/sciadv.1602548)
Peixoto TP (2023) Descriptive Vs. inferential community detection in networks: pitfalls, myths and half-truths. elements in the structure and dynamics of complex networks. Cambridge University Press, Cambridge
Rand WM (1971) Objective criteria for the evaluation of clustering methods. J Am Statist Assoc 66(336):846–850
Rojas A, Calatayud J, Kowalewski M, Neuman M, Rosvall M (2021) A multiscale view of the phanerozoic fossil record reveals the three major biotic transitions. Commun Biol 4(1):1–8
Rossetti G, Cazabet R (2018) Community discovery in dynamic networks: a survey. ACM Comput Surv (CSUR) 51(2):1–37
Sattar NS, Buluc A, Ibrahim KZ, Arifuzzaman S (2023) Exploring temporal community evolution: algorithmic approaches and parallel optimization for dynamic community detection. Appl Netw Sci 8(1):64. https://doi.org/10.1007/s41109-023-00592-1
Smiljanić J, Edler D, Rosvall M (2020) Mapping flows on sparse networks with missing links. Phys Rev E 102:012302. https://doi.org/10.1103/PhysRevE.102.012302
Stehlé J, Voirin N, Barrat A, Cattuto C, Isella L, Pinton J-F, Quaggiotto M, Broeck W, Régis C, Lina B (2011) High-resolution measurements of face-to-face contact patterns in a primary school. PloS One 6(8):23176
Warrens MJ, Hoef H (2022) Understanding the adjusted rand index and other partition comparison indices based on counting object pairs. J Classif 39(3):487–509

Vinh XN, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. J Mach Learn Res 11(95):2837–2854
Zonker J, Padilla-Iglesias C, Djurdjevac Conrad N (2023) Insights into drivers of mobility and cultural dynamics of African hunter-gatherers over the past 120000 years. R Soc Open Sci 10(11):230495

Publisher's Note