nature communications

Article

https://doi.org/10.1038/s41467-024-54280-4

An embedding-based distance for temporal graphs

Received: 15 April 2024

Accepted: 6 November 2024

Published online: 17 November 2024

Check for updates

Lorenzo Dall'Amico [1], Alain Barrat [2,3] & Ciro Cattuto [1,3]

Temporal graphs are commonly used to represent time-resolved relations between entities in many natural and artificial systems. Many techniques were devised to investigate the evolution of temporal graphs by comparing their state at different time points. However, quantifying the similarity between temporal graphs as a whole is an open problem. Here, we use embeddings based on time-respecting random walks to introduce a new notion of distance between temporal graphs. This distance is well-defined for pairs of temporal graphs with different numbers of nodes and different time spans. We study the case of a matched pair of graphs, when a known relation exists between their nodes, and the case of unmatched graphs, when such a relation is unavailable and the graphs may be of different sizes. We use empirical and synthetic temporal network data to show that the distance we introduce discriminates graphs with different topological and temporal properties. We provide an efficient implementation of the distance computation suitable for large-scale temporal graphs.

Graphs are ubiquitous mathematical entities formed by a set of *nodes* and one of the edges connecting pairs of nodes[1–3]. They can model a wide range of interacting systems, such as social[4,5], technological[6], spatial[7], and biological networks[8], and owe their popularity to their ability to encode the complex relational structure of these systems. Real-world systems, moreover, often have temporal properties that cannot be encoded in static graphs and call for modeling based on time-resolved network representations, known as *temporal graphs*[9]. Examples of these include transportation[10] and ecological networks[11], human close-range interactions[12–16], collaboration networks[17–19], etc.

Given their ubiquitous use in representing such diverse kinds of systems, it is crucial to be able to compare them. In fact, defining and computing a similarity measure between pairs of graphs[20] underpins many important applications and tasks, including machine learning tasks such as graph classification. However, given the high dimensionality of graphs and their structural complexity, many different notions of distance can be devised, capturing different properties of interest. Hence, many definitions of distance for static graphs were introduced, as reviewed in refs. 21–25. One of the most straightforward approaches is the *edit distance*[26], which counts the number of elementary changes (edge or node removal or addition) needed to transform one graph into another and only accounts for local information. Approaches that probe the global graph structure are usually based on matrix inversion[27,28] but generally have a high computational cost and approximations are needed to achieve satisfactory computational performances. The above methods are all designed for comparing pairs of graphs with a known correspondence between nodes, a case we will refer to as *matched* graphs. The more general case of *unmatched* graphs entails defining distances for pairs of graphs for which a mapping between nodes is unavailable, and the graphs may have different numbers of nodes. In this case, one can either extract and compare a suitable set of chosen graph features, as done in Refs. 29,30, or leverage the spectral properties of appropriate matrix representations[31–34]. In this case, the computational complexity is also a bottleneck, and approximation heuristics are often required[21].

In the case of temporal graphs, the temporal dimension adds to the complexity of graph comparison. Many of the distance measures mentioned above for static graphs were used to study temporal graphs[35], typically to compare the state of the graph at different points in time and to identify anomalies and change points in the temporal evolution of the graph[36–40]. Here instead, we study the problem of comparing temporal graphs, each taken as a whole. Unlike the

[1]ISI Foundation, Turin 10126, Italy. [2]Aix-Marseille Univ, Université de Toulon, CNRS, CPT, Marseille 13009, France. [3]These authors contributed equally: Alain Barrat, Ciro Cattuto. e-mail: lorenzo.dallamico@isi.it

snapshot-based methods mentioned above[36–40], we rely on a temporal embedding with a dimensionality that is independent of the temporal span of the graphs, so that we can compare temporal graphs with arbitrary, and different, temporal spans. Few notions of distance were proposed to compare temporal graphs with arbitrary time spans. In particular, the method of ref. 41 is based on a set of features that can be tailored to a specific research question (similarly to ref. 29 for static graphs). Ref. 42, on the other hand, addresses the same problem of temporal graph comparison we tackle here, but it assumes the graphs to be matched, and it is computationally inefficient (the graph dissimilarity is obtained by solving an NP-hard optimization problem and the authors propose an approximation running in $\mathcal{O}(n^3)$ operations, where n is the number of graph nodes). Finally, ref. 43 only considers the *unmatched* case and focuses on a specific property of temporal graphs, namely the comparison of temporal paths' statistics, to provide a definition of *dissimilarity*.

Here, we propose a computationally efficient method to compute an actual distance metric between pairs of temporal graphs, considering both the matched and unmatched cases. We build our comparison on top of graph embeddings, leveraging ideas from the local and global approaches mentioned above and encoding differences at the topological and temporal levels. Commented `Python` code that implements the approach described here is publicly available at the link provided in ref. 44.

Results

Before delving into the details of our main contribution, we first state some basic definitions.

Definition 1. (Temporal graph) A temporal graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$ is a tuple $(\mathcal{V},\mathcal{E},\mathcal{W})$, where $\mathcal{V}$ is a set of n nodes, $\mathcal{E}$ a set of temporal edges between pairs of nodes, and $\mathcal{W}$ a set of edge weights. For a discrete variable t denoting time, all temporal edges $e \in \mathcal{E}$ have the form $e=(i,\ j,\ t)$, representing an interaction between nodes i and j at time t and w_e indicates the positive weight of edge e. We say an edge (ij) is *active* at time t if $(i,j,t) \in \mathcal{E}$. A node i is active at time t if it belongs to an active edge at t.

The graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$ can be represented with a sequence of *weighted adjacency matrices*, $\{W_t\}_{t=1,\ldots,T}$, where T is the number of time points. Each of these matrices has size $n \times n$ and entries $(W_t)_{ij}=w_{(i,j,t)}>0$ if $(i,j,t) \in \mathcal{E}$ and $(W_t)_{ij}=0$ otherwise. In the following, we will refer to these matrices as "snapshots" of the temporal graph at given times and to T as the number of snapshots[45]. In a more general description of temporal networks, each interaction may be represented with a tuple (i,j,t,τ) where $\tau \in \mathbb{R}^+$ is an interaction duration. We adopt the former notation because it is simpler to handle in the following. The two notations are completely interchangeable whenever the time is discrete and a finite temporal resolution t_{res} is set. In fact, all contacts with a duration within t_{res} are considered to be instantaneous, and (i,j,t,τ) can be represented by a set of interactions of unitary duration: $\{(i,j,t+\delta)\}_{\delta\in\{0,\ldots,\tau-1\}}$. For this reason, Definition 1 is sufficiently general, as it can also describe interactions lasting more than one temporal unit. For increasing values of t_{res}, the temporal graph gets progressively more aggregated and loses information on the temporal ordering of contacts within t_{res}, but partial information can be retained with edge weights expressing the fraction of t_{res} in which the edge was active.

Let us now introduce the concept of matched and unmatched graphs.

Definition 2. (Matched graphs) Let $\mathcal{G}_1(\mathcal{V}_1,\mathcal{E}_1,\mathcal{W}_1)$ and $\mathcal{G}_2(\mathcal{V}_2,\mathcal{E}_2,\mathcal{W}_2)$ be two temporal graphs with n_1 and n_2 nodes, respectively. We say that $\mathcal{G}_1$ and $\mathcal{G}_2$ are matched if $n_1=n_2$ and there exists a known bijective function $\pi:\mathcal{V}_1\rightarrow\mathcal{V}_2$ that maps each node of $\mathcal{G}_1$ to a node of $\mathcal{G}_2$. The two graphs are otherwise said to be unmatched.

When two graphs have the same size, i.e., the same number of nodes, there are always many possible mappings π between their nodes, but for the matched approach to be valuable, a known mapping associated with an external notion of node identity across the two graphs is necessary (e.g., nodes corresponding to the same persons in two temporal social networks). An implicit consequence of Definition 2 is that all graph pairs with a different number of nodes are considered to be unmatched. We now state the requirements for our distance measure and illustrate our main result.

In the following, we define two types of distance functions, d_{m}, d_{u} between matched and unmatched graphs, respectively. The inputs to these functions are two temporal graphs, $\mathcal{G}_1$ and $\mathcal{G}_2$, with potentially different numbers of time points (snapshots) T_1 and T_2. In both cases, the distances should fulfill the following standard requirements[46]: (i) non-negativity; (ii) separation axiom; (iii) symmetry; (iv) triangle inequality. In addition, both distances should be able by design to detect differences induced by the permutation of time indices. The matched distance should also differentiate between temporal networks differing only by a permutation of node indices, while the unmatched one should be invariant under such transformation.

Our proposed distances are computed in two steps: (i) we generate a temporal graph embedding, given by a matrix of size $(n\times d)$ (with d independent of T), that encodes relevant topological and temporal properties by leveraging time-respecting random walks on the temporal network; (ii) we define the distance in the embedded space, treating separately the matched and the unmatched cases. The distance computation pipeline is illustrated in Fig. 1. The steps outlined above are described in detail in the following two sections titled

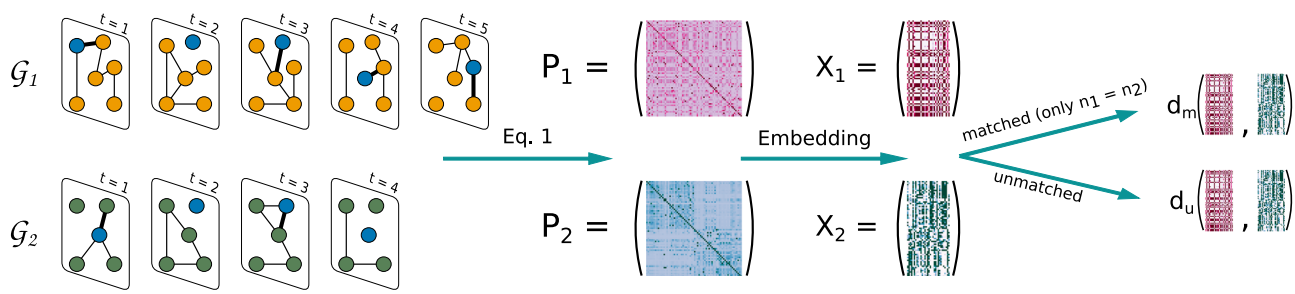

Fig. 1 | Distance computation between two temporal graphs. The inputs to the distance function are two temporal graphs $\mathcal{G}_1$ (top, orange nodes) and $\mathcal{G}_2$ (bottom, green nodes), generally with a different number of snapshots ($T_1=5$, $T_2=4$, in the example) and a different number of nodes n (here $n_1=6$, $n_2=5$). Each graph is represented as a matrix $P \in \mathbb{R}^{n\times n}$, according to Eq. (1), with entry (ij) encoding the probability that a random walker goes from i to j following a time-respecting path (the walker's position at each time point is indicated in blue). The matrices P_1, P_2 are then embedded using the `EDRep` algorithm, mapping them to $X_1 \in \mathbb{R}^{n_1\times d}$ and $X_2=\mathbb{R}^{n_2\times d}$. Finally, the matched—Eq. (5)—and unmatched—Eq. (6) —distances are calculated. Notice that a necessary condition to compute d_{m} is that $n_1=n_2$.

Temporal graph embedding and Defining an embedding-based distance.

Temporal graph embedding

Graphs are rich data representations and it is challenging to define simple mathematical operations to manipulate them. Graph embeddings provide an interesting avenue for doing so, as they yield a representation in vector space that preserves some of the graphs' relevant properties[47–50]. In the case of temporal graphs, the additional challenge of dealing with their temporal dimension has been tackled in many different ways: ref. 51 represents time as an additional dimension of an adjacency tensor; in evolutionary spectral clustering[52–56], a different embedding is obtained for each snapshot, and smoothness conditions are implemented on their temporal evolution; more recent work[57–59] introduced approaches based on time-respecting random walks, which encode temporal properties relevant for dynamical processes occurring on temporal graphs; refs. 39,40 also build one embedding per snapshot for studying the graph evolution and comparing different time points; the authors argue that a graph comparison metric should depend on the whole graph structure and thus define graph instead of node embeddings.

Following an approach similar to ref. 59, we start from time-respecting random walks to generate temporal graph embeddings, which we use to define a distance based on global temporal graph properties. The embeddings are obtained as the solution to an optimization problem. We define a loss function whose argument is a transition matrix P that is the limiting distribution of time-respecting random walks over the temporal graph $\mathcal{G}$ to be embedded. T being the number of snapshots of $\mathcal{G}$, we consider an ensemble of random walks of length ℓ drawn uniformly at random between 1 and T, and starting from a randomly chosen node at time $t = T - \ell + 1$; at each time step, the walker, situated, e.g., on node i, can either move to a random neighbor of i in the snapshot at time t or stay in place on i; if the walker is situated on a node that is not active at time t, i.e., which has no neighbors in this snapshot, the walker stays in place with probability one. The graph snapshots on the left of Fig. 1 show examples of such time-respecting random walks on the input graphs.

Specifically, given a temporal graph $\mathcal{G}$, with W_t the snapshot adjacency matrices at times $t = 1, \cdots, T$, we define for each snapshot $\hat{L}_t = (D_t + I_n)^{-1}(W_t + I_n)$, where D_t is the degree matrix $D_t = \mathrm{diag}(W_t \mathbf{1}_n)$ of the snapshot, I_n is the identity matrix of size $n \times n$, and $\mathbf{1}_n$ is the n-sized unity vector. We then define the matrix P as

$$P = \frac{1}{T}\sum_{\tau=1}^{T} \hat{L}_\tau \hat{L}_{\tau+1} \cdots \hat{L}_T \ , \tag{1}$$

The matrices $\hat{L}_t$ and P are sometimes referred to as *temporal transition matrix* and *global transition matrix*, respectively. Time-respecting random walks depend on several properties of the temporal graph, that may include, for instance, the presence of a non-Markovian behavior, and of a broad distribution of the edge activity and inter-event durations[60]. The limiting probability matrix P of time-respecting random walks encodes the aforementioned properties, and it is sensitive to the permutation of time indices. We remark that the random walks we use can be of arbitrary length. Hence, the matrix P is, in principle, sensitive to all time scales of the temporal graph at hand.

Given the above probability matrix P and a chosen embedding dimensionality d, we now generate, for each node i of $\mathcal{G}$, a unitary node embedding vector $\mathbf{x}_i \in \mathbb{R}^d$. To this end, we define $X \in \mathbb{R}^{n \times d}$ as a matrix where each row represents an embedding vector. We then express the loss function as:

$$\mathcal{L}_P(X) = -\sum_{i,j \in \mathcal{V}} \left[P_{ij} \log Q_{ij}(X) - \frac{\mathbf{x}_i^T \mathbf{x}_j}{n} \right] \ , \tag{2}$$

$$Q_{ij}(X) = \frac{e^{\mathbf{x}_i^T \mathbf{x}_j}}{\sum_{k \in \mathcal{V}} e^{\mathbf{x}_i^T \mathbf{x}_k}} := \frac{e^{\mathbf{x}_i^T \mathbf{x}_j}}{Z_i} \ . \tag{3}$$

This loss function is the sum of the cross entropies between the distributions in the rows of P and the variational distributions in the rows of $Q(X)$, with the addition of a regularization term. Since Z_i is computed in $\mathcal{O}(n)$ operations for each i, the function $\mathcal{L}_P$ requires $\mathcal{O}(n^2)$ operations to be computed. We adopt the recently proposed `EDRep` algorithm[61] that introduces an efficient approximation of Z_i, optimizing $\mathcal{L}_P$ in $O(n)$ operations. For further details, we refer the reader to the *Methods* section.

The matrix X obtained by optimizing $\mathcal{L}_P$ defines the embedding of the temporal graph $\mathcal{G}$. In the following, we show how to leverage this embedding to define a distance between temporal graphs.

Defining an embedding-based distance

We want to define a distance between temporal graph embeddings satisfying the requirements stated above. Notice that the embedding vectors $\mathbf{x}$ are defined up to an orthogonal transformation applied to the rows of X. In fact, suppose $R \in \mathbb{R}^{d \times d}$ is an orthogonal matrix, and $\tilde{\mathbf{x}}_i = R\mathbf{x}_i$, then $\mathbf{x}_i^T \mathbf{x}_j = \tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j$ and the loss function of Eq. (2) is $\mathcal{L}_P(\tilde{X}) = \mathcal{L}_P(X)$. Hence, our distance must be invariant under orthogonal transformations. Moreover, as discussed above, the distance for unmatched graphs must also be invariant with respect to node permutations.

Given these requirements, we now discuss separately our choices for the the matched and unmatched cases. In the *Evaluation with empirical data* section, we show the ability of our distances to discriminate between temporal graph classes that differ for topological or temporal properties.

Distance definition for matched graphs. Let X_1 and $X_2 \in \mathbb{R}^{n \times d}$ be the embedding matrices of two matched temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$. To satisfy the invariance requirements discussed above, rather than defining our distance directly in terms of the matrices X_1 and X_2, we use the auxiliary matrices $M_{X_1} = X_1 X_1^T$ and $M_{X_2} = X_2 X_2^T$ of size $n \times n$. These matrices are by construction invariant with respect to orthogonal transformations of the embedding space and encode the global structure of the corresponding temporal graphs, as the embeddings are generated using the complete temporal network information. We define the distance d_{m} between matched temporal graphs as:

$$d_{\mathrm{m}}(\mathcal{G}_1, \mathcal{G}_2) := \| M_{X_1} - M_{X_2} \|_{\mathrm{F}} \ , \tag{4}$$

where $\| \cdot \|_{\mathrm{F}}$ is the Frobenius norm. Each matrix M encodes the similarity in the embedding space between all pairs of nodes of the corresponding temporal graph, and d_{m} quantifies thus the difference between these similarity patterns. In the *Methods* section, we show that we do not need to actually compute the large $(n \times n)$ M_{X_1} and M_{X_2} matrices, but that the distance can be computed using the generally smaller $d \times d$ matrices as follows:

$$d_{\mathrm{m}}(\mathcal{G}_1, \mathcal{G}_2) = \sqrt{\left\| X_1^T X_1 \right\|_{\mathrm{F}}^2 + \left\| X_2^T X_2 \right\|_{\mathrm{F}}^2 - 2\left\| X_1^T X_2 \right\|_{\mathrm{F}}^2} \ , \tag{5}$$

We can easily verify that d_{m} is a distance, as it satisfies all the requirements mentioned above and is generally not invariant under the permutation of node indices. Notice that, even if d_{m} can be computed for any pair of graphs with the same size, this distance is only meaningful for matched graphs (i.e., with known matching). In Eq. (5) indeed, the term $X_1^T X_2$ implies that X_1 and X_2 have the same size and that their rows are ordered according to the same node indices.

Table 1 | Summary properties of the SocioPatterns time-resolved proximity networks used here

Graph name	Description	n	Duration	T
Primary school[76,84]	Children of 10 classes of a primary school	242	2 days	194
High school 1[85]	Students of 3 classes of a high school in 2011	126	4 days	453
High school 2[85]	Students of 5 classes of a high school in 2012	180	7 days	1215
High school 3[86]	Students of 9 classes of a high school in 2013	327	5 days	605
Baboons[87]	A group of baboons	13	27 days	3986
Households[88]	People of a village in Malawi	86	26 days	1926
Hospital[89]	Patients and health-care workers of a hospital	75	5 days	579
Conference[13]	People at a medical conference	405	2 days	190
Office[90]	Office workers in an office building	92	19 days	1646

Graph name is used to identify the graphs; *Description* provides concise information on the context where data were collected; n is the number of graph nodes; *Duration* is the temporal span of the dataset, and T is the number of graph snapshots. The temporal resolution is $t_{res} = 10$ min for all datasets.

Distance definition for unmatched graphs. In the case of *unmatched* graphs, there is no correspondence between nodes, and the distance, as we discussed, should also be invariant with respect to node permutations. We, therefore, define the distance between unmatched graphs using auxiliary vectors. Specifically, we denote with $\boldsymbol{\lambda}(A)$ the vector of ordered eigenvalues of a matrix A and define our distance between two *unmatched* temporal graphs as

$$d_{\mathrm{u}}(\mathcal{G}_1, \mathcal{G}_2) = \left\| \boldsymbol{\lambda}\left(\frac{X_1^T X_1}{n_1}\right) - \boldsymbol{\lambda}\left(\frac{X_2^T X_2}{n_2}\right) \right\|_2, \tag{6}$$

where X_1 and X_2 are, as before, the embeddings matrices of the two graphs, with sizes $n_1 = |\mathcal{V}_1|$ and $n_2 = |\mathcal{V}_2|$. The normalization of the covariance matrices $X_1^T X_1$ and $X_2^T X_2$ is chosen so that their eigenvalues are independent of the graph size.

In the Methods section, we show that the covariance matrices $X_1^T X_1$ and $X_2^T X_2$ (of size $d \times d$) are indeed an appropriate argument of the distance function d_{u}, since they are independent of the ordering of the nodes and any other orthogonal transformation applied to the columns of X_1 and X_2. However, a direct comparison of the entries of $X_1^T X_1$ and $X_2^T X_2$ would require a one-to-one correspondence between the embedding dimensions, i.e., the columns of X_1 and X_2. This mapping is unavailable because the EDRep algorithm, like most vector-space embedding techniques, makes no guarantees on the correspondence between the dimensions of the embedding vectors related to different graphs. To tackle this issue, we thus compare the spectra, which are invariant with respect to orthogonal transformations of the rows of the embedding matrices. This choice takes inspiration from spectral distance definitions for static graphs, such as ref. 33, in which graphs are compared by computing the distance between the ordered sets of the eigenvalues of their matrix representations (adjacency, Laplacian, non-backtracking, etc.)

Computational complexity

As discussed in the Methods, the computational complexity of the embedding step is $\mathcal{O}\left(nd^2 + d \cdot \min(n^2, E)\right)$, where $E = \sum_t |\mathcal{E}_t|$ is the total number of temporal edges, n is the number of graph nodes and d is the embedding dimension. On top of this, the distance d_{m} is then obtained with $\mathcal{O}(nd^2 + d^2)$ additional operations, while d_{u} requires $\mathcal{O}(nd^2 + d^3)$ additional operations.

Evaluation with empirical data

We now evaluate the distance definitions we introduced, particularly with respect to their sensitivity to important topological and temporal properties of empirical temporal graphs. We carry out three kinds of evaluation tests using both synthetic and empirical temporal graph data. We use, in particular, empirical data shared by the SocioPatterns collaboration (sociopatterns.org) describing time-resolved, close-range proximity interactions of humans and animals in a variety of real-world environments (see Table 1 of the Methods section). These data are known to exhibit rich multi-scale network structures, complex temporal activity patterns, correlations between these two, and other complex network features[62]. Hence, they provide a natural benchmark to assess the sensitivity of our distance definitions to several specific properties of real-world temporal graphs. It is important to note that these datasets are by no means an exhaustive sample of temporal networks. However, our goal here is to show that the distances we introduce are sensitive to complex properties of temporal networks that we know are present in these particular datasets. Therefore, the evaluations we outline below do not depend on testing our distances on a comprehensive set of temporal networks but rather rely on making tests using data that are well understood, have been and are widely used by the scientific community, and are known to exhibit many important properties of temporal graph data.

1. Discriminating between classes of temporal graphs. We consider topologically different synthetic temporal graphs of different sizes and we cluster them using our temporal graph distance d_{u}, checking how well the clusters we find match the known classes of the generative method we used, irrespective of graph size. We then carry out the same test using empirical graphs on human proximity in school settings.
2. Detecting partial node relabeling. We consider a synthetic temporal graph, select a fraction of its nodes, shuffle their indices and compute the distance d_{m} between the original and the relabeled graphs. We investigate the behavior of the distance d_{m} as a function of the fraction of relabeled nodes.
3. Discriminating between temporal graph randomizations. We use the empirical data to elucidate which of the many properties of real temporal graphs are discriminated by the distances we introduced. We proceed as follows: given an empirical temporal graph, we apply to it a set of randomization operations, where each operation is designed to preserve specific properties of the original data (e.g., the activity time series of each node). We then attempt to discriminate the randomized temporal graphs from one another using our distance definitions and quantify the performance of such discrimination tasks. We test both d_{m} and d_{u}; however, by design, this evaluation strategy compares temporal graphs with the same number of nodes.

Discriminating between classes of temporal graphs. To test the ability of our model to distinguish topological network features, we proceed as follows. We consider four random generative models for static graphs and generate 250 instances from each model with size uniformly distributed between $n = 200$ and $n = 1800$ and constant average degree. The models we use are (i) the Erdős-Renyi (ER) model[63]; (ii) the stochastic block model (SBM), capable of generating

graphs with a known community structure[64]; (iii) the configuration model (CM), yielding graphs with an arbitrary degree distribution[65]; (iv) the geometric model (GM), which generates edges based on proximity in a latent space[66]. Unlike the first three, the latter model yields graphs with a high clustering coefficient. Details on these generative models are provided in the Methods section. We then turn these graphs into temporal graphs by assigning an activity time series to each edge of the static graph, i.e., a set of timestamps at which that edge is active. To do this, we copy the activity time series of a randomly chosen edge of an empirical temporal graph, namely the *Conference* dataset, which has the largest number of nodes among the empirical datasets we use. Details on the properties of this dataset are given in Table 1. We then deploy our distance definition d_u to compute a pairwise distance matrix between all these temporal graphs and use this matrix as an input for an unsupervised clustering algorithm to label each graph. The unsupervised clustering algorithm only takes the distance matrix and the number of clusters k (here $k = 4$) as inputs. Here, we first perform a spectral embedding of the distance matrix between graph pairs using non-negative matrix factorization (NMF)[67] extracting k non-negative components and then applying the k-means clustering algorithm with k classes[68]. We note that successively embedding and clustering is a standard approach, and using NMF is most suited here since the distance matrix is non-negative. Finally, we compare the original temporal graph labels (i.e., the classes of our generative model) with the partition labels yielded by unsupervised clustering based on our distances: we quantify the match between the true partition and the inferred one using the normalized mutual information (NMI), which yields a performance metric ranging between 0 (random guess) and 1 (perfect reconstruction).

Figure 2A shows the normalized mutual information (NMI) between the known temporal graph classes (associated with the four generative models) and the inferred cluster labels, as a function of the embedding dimension d. The discrimination performance is low for small values of d, but for $d \gtrsim 8$, the accuracy is high and becomes insensitive to the specific value of d. The inset in Fig. 2A shows the UMAP[69] two-dimensional embedding of the vectors $\boldsymbol{\lambda} \in \mathbb{R}^{32}$ used in the distance d_u. Each symbol corresponds to one temporal graph, the color and marker shape indicate its generative model, and the marker size is proportional to the number of graph nodes. We observe that the graph size does not appear to affect distances systematically and that the $\boldsymbol{\lambda}$ vectors used by the distance d_u allow us to discriminate between all four generative models. Indeed, we verified that the distances d_u between graph pairs and the Euclidean distances between their respective UMAP embeddings are strongly correlated, with a highly significant Spearman correlation coefficient of -0.92.

We now carry out a test using the distance d_u on empirical temporal graphs. We look at the challenging setting in which the graph pairs considered have comparable sizes and time spans and are collected in similar contexts. To do so, we use the SocioPatterns datasets collected in schools *Primary school, High school 1, High school 2, High school 3*, described in Table 1, representing time-resolved, close-range proximity interactions of individuals grouped in multiple school classes, over several days. We extract sub-temporal graphs from these datasets and compare them with one another using the distance d_u. We associate the school classes with a label: *Primary* for all primary school classes; *Bio* for the high school classes on biology; *Other* for the remaining classes.

We consider a pair of classes (c_1, c_2) from the same dataset, and we build a sub-graph restricted to all the interactions on a given day "Day" between the students of the classes c_1 and c_2. We finally compute the two-dimensional UMAP embedding from the vector $\boldsymbol{\lambda}$ appearing in Eq. (6). Figure 2B shows the result of this procedure for all possible triplets (c_1, c_2, Day). Class labels are indicated by the markers' color and style. We observe a very high correlation between the distances d_u of graph pairs and the Euclidean distances between their UMAP embedding vectors, with a Spearman correlation of 0.88. That is, our distance definition can tell apart the networks extracted from the primary school dataset from those coming from the high school dataset. Moreover, a weaker but still visible separation exists in the embeddings of (Bio-Other) with respect to the other triplets. This is likely because biology classes were often held in school laboratories rather than in classrooms, leading to different interaction patterns. Our distance can discriminate between temporal networks that are, in principle, similar (school classes over one day) with no major structural or temporal differences.

Detecting partial node relabeling. One required property for matched distance is to distinguish between graphs that differ only upon a node relabeling. To evaluate the ability of our distance to accomplish this task, we work with the synthetic temporal graphs defined in the section Discriminating between classes of temporal graphs. For each random graph model, we (i) generate an instance of the random graph with $n = 1000$ nodes; (ii) randomly select a fraction α of nodes and shuffle their labels; (iii) compute the distance between the original graph and the partially relabeled one. This is repeated for 25 different realizations of the partial relabeling. Figure 3 shows the distance d_m (rescaled by the graph size) as a function of α. The plots confirm that the distance depends on partial node relabeling for all the considered generative models, with a positive distance as soon as $\alpha > 0$, and increases with the fraction α of relabeled nodes. Notably, the plot

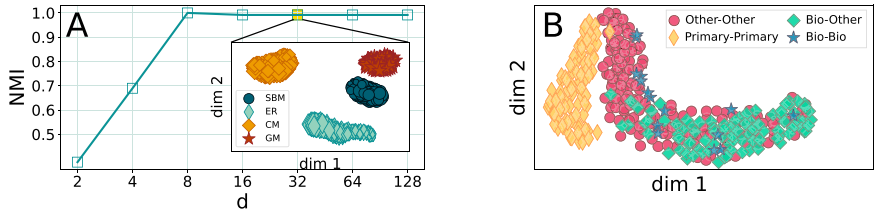

Fig. 2 | Validation of the distances on graphs of varying size. A Accuracy of the distance-based clustering against the ground truth classes in terms of normalized mutual information (NMI), as a function of the embedding dimension d used in the embedding step shown in Fig. 1. The clustering task consists of recognizing four classes of synthetic temporal graphs with an unsupervised algorithm based on the distance d_u. The classes are obtained by generating a graph from either of four models (stochastic block model (SBM), configuration model (CM), Erdős-Renyi (ER), and geometric model (GM)) with constant degree equal to 4.8, and the temporal component is obtained by sampling the edge activity of an empirical graph, as detailed in the main text. Inset of panel **A** Scatter plot of UMAP dimensionality reduction in two dimensions of the vector $\boldsymbol{\lambda}$ appearing in the definition of d_u given in Eq. (6), with $d = 32$. Each point refers to a temporal graph; the color and marker style refer to the generative model of its static component, while the marker size is proportional to the number of nodes. **B** Two-dimensional UMAP embedding of $\boldsymbol{\lambda}$ for the temporal graph obtained by selecting two classes and a day of interaction for all possible (c_1, c_2, Day) triplets in the SocioPatterns datasets describing temporal graphs of human proximity in schools. Each point refers to a triplet, and the color and marker style are assigned according to the class labels: the primary school classes form a group on their own, and the other three groups (Other-Other, Bio-Other, Bio-Bio) belong to the high school datasets, where Bio are the biology classes, and Other the remaining ones. In all cases, the temporal networks are aggregated over a scale of $t_{res} = 10$ min.

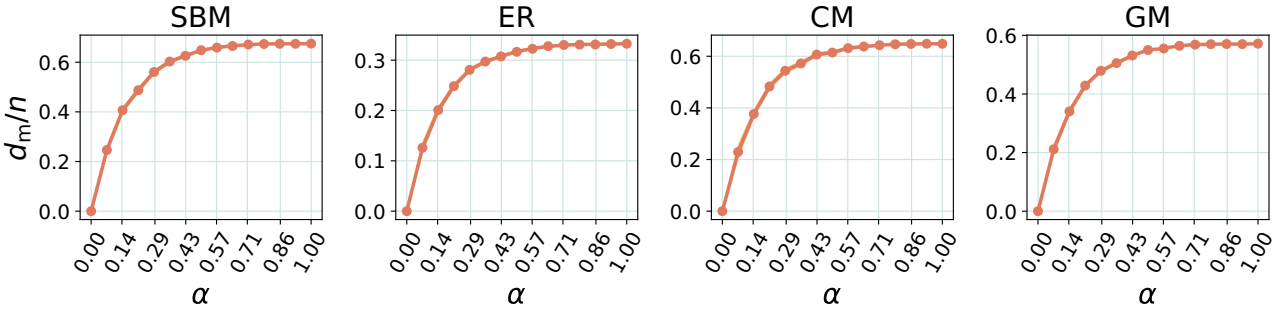

Fig. 3 | Detection of partial node relabeling. Normalized matched distance d_m/n between a temporal graph and itself, upon partial node relabeling, as a function of the fraction α of relabeled nodes. The plots refer to the four generative models used in Fig. 2A: stochastic block model (SBM), Erdős-Renyi (ER), configuration model (CM), and geometric model (GM). with an average degree equal to 4.8. For each graph, the temporal component is obtained by sampling the edge activity of an empirical graph, as detailed in the main text. The barely visible shadow line is the standard deviation of the distance across 25 randomizations of the partial relabeling.

suggests a high discriminative power even for small α values, as the distance increases rapidly with α.

Discriminating between temporal graph randomizations. Let us now consider 9 empirical `SocioPatterns` datasets, whose basic properties are summarized in Table 1 of the *Methods* section. Following ref. 70, we select six types of randomization operations R_i ($i = 1, \ldots, 6$), described in detail in the *Methods* section. In all cases, we preserve the number of nodes and the number of snapshots of the original temporal graph. These six randomizations are chosen because they allow us to inspect relevant temporal graph properties, including the interaction duration distribution, the node activation time series, the strength distribution in the aggregated graph, or the presence of a community structure. We can thus investigate graph properties both at the topological and temporal levels.

To quantify the discriminative performance of our temporal graph distance, we proceed as follows, separately for each empirical temporal graph: given the empirical temporal graph and a pair of randomization operations (R_i, R_j), we generate a set of 250 realizations using R_i, labeled as 0, and a set of 250 realizations using R_j, labeled as 1, for a total of 500 temporal graphs. We then compute the matrix of distances between these 500 graphs and use it to cluster them in 2 clusters, following the same procedure used for Fig. 2A. Figure 4 reports the NMI values between the real labels and the unsupervised algorithm result for each pair of randomizations (R_i, R_j), and for both the matched distance d_m (left panel) and the unmatched distance d_u (right panel). Within each panel, each matrix refers to one empirical temporal graph, and its entries are the NMI values described above. A high NMI entry indicates that our distance discriminates well between the temporal graphs generated according to the two randomization procedures in the corresponding row and column of the matrix entry, i.e., that the distance is sensitive to the properties preserved by either R_i or R_j. Low NMI values may either correspond to an inability of the distance to capture those properties or simply reflect that the empirical temporal graph is statistically random as far as those properties are concerned, as observed in particular for several pairs of randomizations in the *Baboons* dataset. Thus, it is enough to find one empirical dataset for which the distance can be distinguished between a pair of randomizations to conclude that the distance is indeed sensitive to the corresponding graph properties.

Discussion

We introduced a novel definition of distance between pairs of temporal graphs. This definition entails two steps. First, the temporal graphs are embedded in Euclidean space, and then a distance is defined in the embedding space. For the first step, we use an embedding based on time-respecting random walks over the temporal graph. Such walks are known to depend on and to encode important structural and temporal properties of time-varying graphs[9,43,59,60,71]. For the second step, we proposed two possibilities for the distance definition: if a mapping is known between the nodes of the graphs to be compared, we consider a distance definition that leverages such mapping; in the more general case, when such a mapping is unavailable, we put forward a definition that makes it possible to compare graphs with arbitrarily different sizes (numbers of nodes). In both cases, since the size of the embedding matrix we use does not depend on the graph's temporal span, it is possible to embed temporal graphs with different durations in the same embedding space and thus compute a distance between them. The evaluation of our approach on both synthetic and empirical data shows that the proposed distance is sensitive to structural differences (e.g., degree distribution, clustering coefficient, or presence of communities) as well as to temporal differences (e.g., burstiness or node activity patterns) of the temporal graphs being compared.

The methodology we developed is very general and customizable, as several alternative choices are possible for both steps of our approach, namely the chosen embedding method and distance. The specific choices we study here are motivated by two considerations: (i) a theoretical one, given the importance of time-respecting random walks in encoding information on temporal graphs, and thus in generating distributed representations of temporal graphs[9,59,60,71]; (ii) a numerical one, which led us to choose the `EDRep` algorithm of ref. 61 because of its efficiency and scalability. Our evaluation results support these choices a posteriori.

Some limitations of our approach are worth mentioning. First, the part of our evaluation based on empirical data focused on a limited set of temporal graphs, namely temporal networks describing the close-range proximity of humans. However, these temporal graphs are known to feature a broad variety of representative structural and temporal features, such as time-varying community structures, burstiness of edge activity, fat-tailed distributions of interaction durations, and more[13,15]. Second, we only considered the cases of fully matched graphs (a bijective relation between nodes) or no known matching. An interesting intermediate situation would be partially matched graphs, in which only a subset of nodes are matched across the two graphs. Tackling this challenging case would yield an interesting extension of our work. Finally, our unmatched distance allows us to compare graphs with sizes (number of nodes) that potentially differ even by orders of magnitude. Even if our definition can deal with such extreme cases, we believe that this kind of comparison calls for a more profound question on what it means to compare entities that differ so much. This case might require tailored definitions of distance that leverage domain-specific knowledge.

https://doi.org/10.1038/s41467-024-54280-4

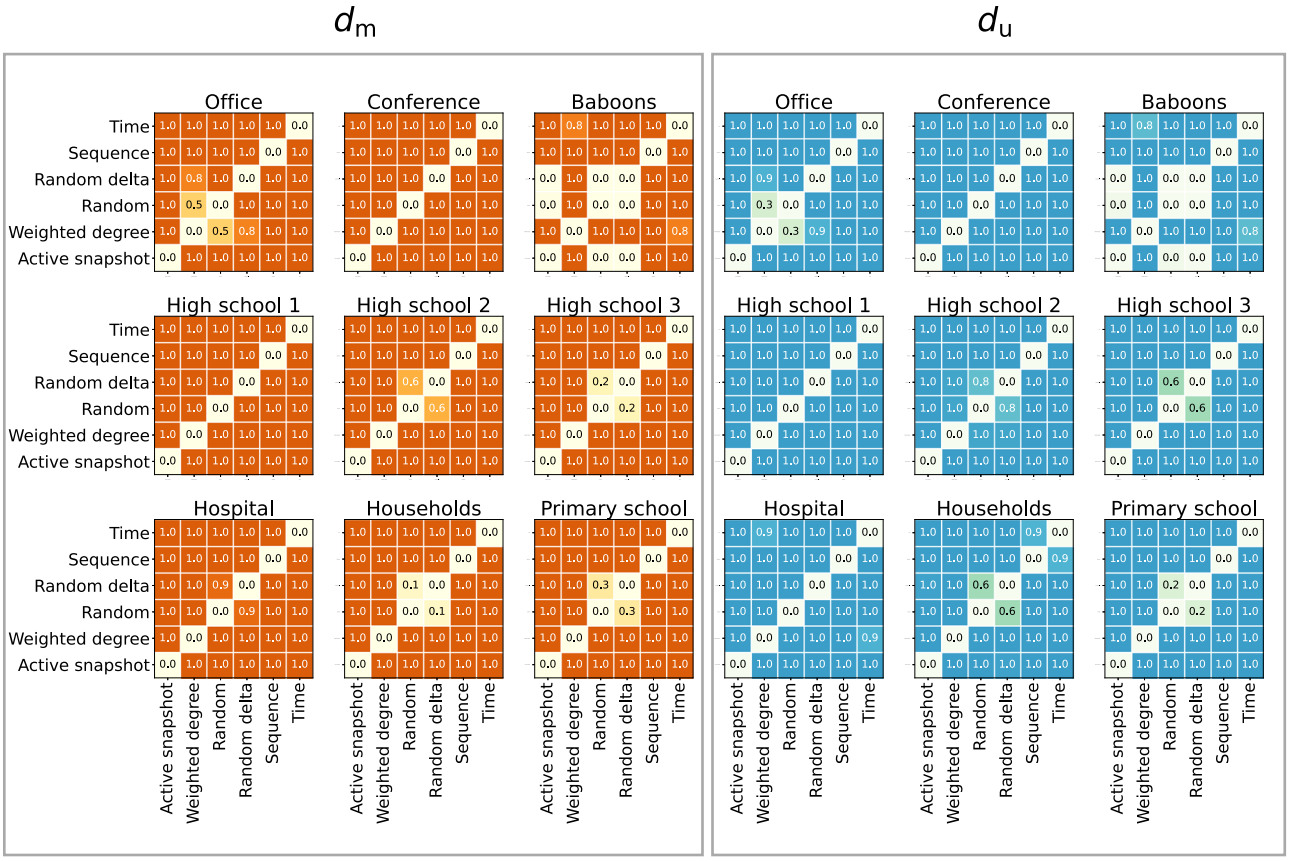

Fig. 4 | Distance-based graph clustering for ensembles of temporal graphs generated according to different randomization techniques. Left panel: results for the matched distance d_m of Eq. (5). *Right panel*: results for the unmatched distance d_u of Eq. (6). Within each panel, each matrix corresponds to one of the 9 `SocioPatterns` temporal graphs described in Table 1. The rows and columns of each matrix correspond to the same set of six randomization techniques we used, described in the Methods sections. Each input graph is represented as a sequence of temporal edges (i, j, t) as per Definition 1. The randomizations act on the temporal edges as follows. *Random*: preserves the number of temporal edges and randomizes the node and time indices; Random delta: preserves the number of temporal edges and interaction duration distribution; *Active snapshot*: preserves the number of edges at each time step and the times at which each node is *active*, i.e., at which it has at least one neighbor. *Time*: preserves the aggregated weighted graph structure, i.e., the number of times each edge is active in the temporal graph; Sequence; preserves each snapshot's adjacency matrix and randomizes the order in which they appear; *Weighted degree*: preserves the total number of temporal edges involving each node. For each pair of randomizations, we infer the randomization method of each temporal graph via an unsupervised distance-based clustering algorithm, and we compare the inferred randomization method with the known true one. Each matrix entry reports (value and color coding) the accuracy of the inferred labels (randomization methods), quantified as the normalized mutual information (NMI) between the inferred and true labels.

Defining a distance between temporal graphs opens the door to a wealth of potential applications, of which our evaluation only offers possible examples. On the one hand, such a distance can support data analysis of temporal graph data by extending the procedure of Fig. 4 to a richer set of randomizations[51]. Indeed, as discussed, the inability to distinguish pairs of randomizations of a given temporal graph can be regarded as an invariance property of the graph itself. On the other hand, computing a distance between graphs can provide a crucial tool for the validation of generative models that is often limited to comparing a set of statistical properties[41], while our distance definition enables us to carry out a comparison at the global level. In addition, our distance could underpin the very process of generating synthetic temporal graphs by enabling approaches such as *GANS*[72] that depend on a global distance and have proved extremely powerful in generating realistic data[73,74]. In particular, generating synthetic human proximity data with realistic topological and temporal properties could be used to better simulate infectious disease spread and other phenomena of interest for public health research[75–80]. Finally, we know that empirical (temporal) graphs are hard to anonymize[79,81,82]. To minimize the risk of node re-identification, performing some perturbation operations on the graph might be necessary before making it publicly available. However, such operations risk destroying essential patterns and information of empirical data. The distance we introduced could help tackle the trade-offs between re-identification risk and information loss by better quantifying the latter.

Methods

The `EDRep` algorithm

The `EDRep` algorithm[61] was recently proposed to efficiently generate embeddings given a probability matrix encoding the affinity between the embedded items. This is done by optimizing the cost function of Eq. (2) under the constraint $\|\mathbf{x}_i\| = 1$ and obtaining a low-dimensional representation of matrix P. A known problem of this type of cost function is the computational complexity because the normalization constants $Z_i = \sum_{k \in \mathcal{V}} e^{\mathbf{x}_i^T \mathbf{x}_k}$ require $\mathcal{O}(n^2)$ to be calculated. However, ref. 61 describes an efficient way to estimate all the Z_i values in $\mathcal{O}(n)$. This is accomplished by first subdividing the nodes into q groups based on the embedding matrix X. Here, q is a parameter of the algorithm, and larger q values generally lead to a higher accuracy, but very good results are already obtained for $q = 1$. Second, for each $a = 1, \ldots, q$,

one computes the mean $\boldsymbol{\mu}_a$ and the covariance matrix Ω_a of the set of embedding vectors of the nodes in group a. Denoting the number of nodes in group a by π_a, one obtains the estimation of Z_i as

$$Z_i \approx \sum_{a=1}^{q} \pi_a \exp\left\{ \mathbf{x}_i^T \boldsymbol{\mu}_a + \frac{1}{2} \mathbf{x}_i^T \Omega_a \mathbf{x}_i \right\} . \tag{7}$$

Coming to the algorithm's computational complexity, given that our choice of P is the sum of matrix products, the algorithm can be implemented in two forms: one in which P is never formally computed and the sequence $\{\hat{L}_t\}_{t=1,\ldots,T}$ is fed to the algorithm; and one in which P is explicitly computed. Letting $X \in \mathbb{R}^{n\times d}$ be the matrix storing the embedding vectors in its rows, the computational complexity is determined by the matrix product PX needed to obtain the gradient of $\mathcal{L}_P$ defined in Eq. (2). In the former case, this is computed in $\mathcal{O}(nd^2 + dE)$ operations, where $E = \sum_{t=1}^{T} |\mathcal{E}_t|$ is the number of temporal edges. In the second case, instead, the complexity is given by the number of non-zero entries of P, which cannot exceed n^2. The former implementation is particularly convenient when n is large and the $\hat{L}_t$ matrices are very sparse. This happens because P may be dense, even if the snapshots are sparse, and for large values of n, representing the matrix P in computer memory might be challenging. Conversely, for small graphs, n^2 may be smaller than the number of temporal edges, making the latter implementation more convenient.

Matched distance definition

From Eq. (5), we have

$$\begin{aligned}
d_{\mathrm{m}}^2(X,Y) &= \sum_{i,j\in\mathcal{V}} \left[(M_X)_{ij} - (M_Y)_{ij} \right]^2 \\
&= \sum_{i,j\in\mathcal{V}} \left[(M_X)_{ij}(M_X)_{ij} + (M_Y)_{ij}(M_Y)_{ij} - 2(M_X)_{ij}(M_Y)_{ij} \right] \\
&\overset{(a)}{=} \sum_{i,j\in\mathcal{V}} \left[(M_X)_{ij}(M_X)_{ji} + (M_Y)_{ij}(M_Y)_{ji} - 2(M_X)_{ij}(M_Y)_{ji} \right] \\
&= \sum_{i\in\mathcal{V}} \left[(M_X M_X^T)_{ii} + (M_Y M_Y^T)_{ii} - 2(M_X M_Y^T)_{ii} \right] \\
&= \mathrm{tr}(M_X M_X^T) + \mathrm{tr}(M_Y M_Y^T) - 2\mathrm{tr}(M_X M_Y^T) \\
&\overset{(b)}{=} \mathrm{tr}(XX^TXX^T) + \mathrm{tr}(YY^TYY^T) - 2\mathrm{tr}(XX^TYY^T) \\
&\overset{(c)}{=} \mathrm{tr}(X^TXX^TX) + \mathrm{tr}(Y^TYY^TY) - 2\mathrm{tr}(Y^TXX^TY) \\
&\overset{(d)}{=} \| X^TX \|_{\mathrm{F}}^2 + \| Y^TY \|_{\mathrm{F}}^2 - 2 \| X^TY \|_{\mathrm{F}}^2 ,
\end{aligned} \tag{8}$$

where in (a) we exploited that M_X and M_Y are symmetric, in (b) we used the definition of M_X, M_Y, in (c) we used that property of the trace stating that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$, and finally in ($d$) we applied the definition of the Frobenius norm.

Properties of the unmatched distance

In this section, we show that the unmatched distance is invariant with respect to node permutations, and, more generally, to any orthogonal transformation applied to the columns of the embedding matrix. We then show that the unmatched distance is also invariant with respect to any orthogonal transformation applied to the rows of the embedding matrix. This is necessary because the `EDRep` loss function is invariant under this type of transformation.

We consider a bijective node mapping $\pi : \mathcal{V} \to \mathcal{V}$. We let $Q \in \mathbb{R}^{n\times n}$ be the permutation matrix defined as $Q_{ij} = \delta_{j,\pi(i)}$. In the matrix multiplication, Q swaps the entry i with $\pi(i)$:

$$\bar{X}_{ia} := (QX)_{ia} = \sum_{j\in\mathcal{V}} Q_{ij} X_{ja} = \sum_{j\in\mathcal{V}} \delta_{j,\pi(i)} X_{ja} = X_{\pi(i),a} . \tag{9}$$

The matrix Q is orthogonal, in fact

$$(QQ^T)_{ij} = \sum_{k\in\mathcal{V}} Q_{ik} Q_{jk} = \sum_{k\in\mathcal{V}} \delta_{k,\pi(i)} \delta_{k,\pi(j)} = \delta_{\pi(i),\pi(j)} = \delta_{ij} , \tag{10}$$

where the last equality follows from π being a bijective mapping. As a consequence, $QQ^T = Q^TQ = I_n$. The distance d_{u} depends on the embedding matrix X only through X^TX. We now show that this expression is invariant under node permutation of the embedding matrix, or any orthogonal matrix Q. Indeed, let $\bar{X} = QX$, then

$$\bar{X}^T\bar{X} = X^T \underbrace{Q^TQ}_{I_n} X = X^TX . \tag{11}$$

We now show that the distance d_{u} is also invariant under the orthogonal transformation applied to the embedding matrix rows. Let $R \in \mathbb{R}^{d\times d}$ be an orthogonal matrix and let $\tilde{X} = XR$. Letting $\lambda_i(M)$ be the i-th smallest eigenvalue of M, then, for all i, $\lambda_i(AB) = \lambda_i(BA)$ [Theorem 1.3.22[83]]. It follows

$$\lambda_i(\tilde{X}^T\tilde{X}) = \lambda_i(R^TX^TXR) = \lambda_i(\underbrace{RR^T}_{I_d} X^TX) = \lambda_i(X^TX) , \tag{12}$$

thus proving the claim.

Temporal graph randomizations

In Table 1 we summarize the properties of the temporal graphs we used to conduct our tests. Here we give a more detailed description of the randomization techniques we adopted, defined in terms of the quantities they preserve. According to the method, it may be more convenient to represent the temporal graph as a sequence of instantaneous interactions (i, j, t), as a sequence of interactions with a duration (i, j, t, τ), or as a sequence of weighted adjacency matrices $W^{(t)}$[70]. Before randomization, time is discretized at the scale of 10 min in each dataset, and the cumulative interaction time in each 10-min window is used as weight.

1. *Random*. Temporal edges are represented as (i, j, t) and all three indices are randomized, avoiding self-edges (i and j randomly sampled with replacement between 1 and n, and t between 1 and T). **Preserved quantities**: number of temporal edges.
2. *Random delta*. Temporal edges are represented as (i, j, t, τ). Once again, i, j, t are randomized (t is randomly sampled with replacement between 1 and $T - \tau$), while τ is preserved. **Preserved quantities**: number of temporal edges and interaction duration distribution.
3. *Active snapshot*. At each time step t, the edges are randomly replaced between active nodes at t, i.e., that had at least one neighbor in the original snapshot. **Preserved quantities**: number of edges at each time step and activity pattern of each node.
4. *Time*. Temporal edges are represented as (i, j, t) and only the index t is randomly sampled with replacement. **Preserved quantities**: aggregated graph structure.
5. *Sequence*. The graph is represented as a sequence of weighted adjacency matrices $W^{(t)}$ and the indices t are shuffled. **Preserved quantities**: the structure of each snapshot.
6. *Weighted degree*. Temporal edges are represented as (i, j, t) and all three indices are randomized as in *Random* but with the constraint that each node appears in the same number of temporal edges as in the original network. **Preserved quantities**: nodes weighted degree.

Synthetic models

We here provide a formal definition of the models used to generate the synthetic graphs under analysis. Even though we considered four

models, three of these can be generated from the *degree-corrected stochastic block model*[64] by changing its parameters.

Definition 3. (Degree-corrected stochastic block model) Let $\mathcal{V}$ be a set of n nodes and $\boldsymbol{\ell} \in [k]^n$ be a vector mapping each node to a class. Further, let $C \in \mathbb{R}^{k\times k}$ be a positive symmetric matrix and $\boldsymbol{\theta} \in \mathbb{R}^n$ be a vector satisfying $\boldsymbol{\theta}^T\mathbf{1}_n = n$. The entries of the graph adjacency matrix $A \in [0,1]^{n\times n}$ are generated independently (up to symmetry) at random with probability

$$\mathbb{P}(A_{ij}=1) = \min\left(1, \theta_i\theta_j \frac{C_{\ell_i,\ell_j}}{n}\right) . \tag{13}$$

The vector $\boldsymbol{\ell}$ contains the labels and gives a community structure in the case in which $C_{a,a} > C_{a,b}$ for $b \neq a$, meaning that there is a higher probability that two nodes in the same community will get connected. The value θ_i is proportional to the expected degree of node i. For this reason, if one chooses $\boldsymbol{\theta} = \mathbf{1}_n$ and $\boldsymbol{\ell} = \mathbf{1}_n$, one gets the Erdős-Renyi model, in which every node has the same expected degree, and there are no communities. The configuration model, instead, is obtained by letting $\boldsymbol{\ell} = \mathbf{1}_n$, but changing the value of $\boldsymbol{\theta}$ to create an arbitrary degree distribution that we choose to be a (properly rescaled) uniform distribution between 3 and 10 raised to the power 4. Finally, the stochastic block model is obtained from a labeling vector different from $\mathbf{1}_n$ and letting $\boldsymbol{\theta} = \mathbf{1}_n$. We consider $k = 5$ communities of equal size with $C_{a,b} = 20\delta_{ab} + (1-\delta_{ab})$, with δ the Kroeneker symbol.

Let us finally introduce the random geometric model.

Definition 4. (Random geometric model) Let $\mathcal{V}$ be a set of n nodes. For each $i \in \mathcal{V}$ let $\mathbf{x}_i \in \mathbb{R}^2$ be a random vector with norm $\|\mathbf{x}_i\| \leq 1$. The entries of the graph adjacency matrix $A \in [0,1]^{n\times n}$ are generated independently (up to symmetry) at random with probability

$$\mathbb{P}(A_{ij}=1) = e^{-\beta\|\mathbf{x}_i-\mathbf{x}_j\|} , \tag{14}$$

for some positive β.

Note that even though the entries of A are drawn at random, this model can generate graphs with a high clustering coefficient because the probability depends on the relative distance between fixed embedding vectors. In our simulations, we set $\beta = 20$.

Data availability

All data used in this article are publicly available at http://www.sociopatterns.org/datasets/ and at ref. 44.

Code availability

Commented `Python` code to reproduce our results is available at ref. 44.

References

1. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509–512 (1999).
2. Newman, M. E. The structure and function of complex networks. *SIAM Rev.* **45**, 167–256 (2003).
3. Barrat, A., Barthélemy, M. & Vespignani, A. *Dynamical Processes on Complex Networks* (Cambridge Univ. Press, 2008).
4. Scott, J. Trend report social network analysis. *Sociology* **22**, 109–127 (1988).
5. Wasserman, S. & Faust, K. *Social Network Analysis: Methods and Applications* (Cambridge Univ. Press, 1994).
6. Amaral, L. A. N., Scala, A., Barthelemy, M. & Stanley, H. E. Classes of small-world networks. *Proc. Natl Acad. Sci. USA* **97**, 11149–11152 (2000).
7. Barthélemy, M. Spatial networks. *Phys. Rep.* **499**, 1–101 (2011).
8. Alm, E. & Arkin, A. P. Biological networks. *Curr. Opin. Struct. Biol.* **13**, 193–202 (2003).
9. Holme, P. & Saramäki, J. Temporal networks. *Phys. Rep.* **519**, 97–125 (2012).
10. Zhao, L. et al. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transport. Syst.* **21**, 3848–3858 (2019).
11. Blonder, B., Wey, T. W., Dornhaus, A., James, R. & Sih, A. Temporal dynamics and network analysis. *Methods Ecol. Evol.* **3**, 958–972 (2012).
12. Eagle, N., Pentland, A. & Lazer, D. Inferring friendship network structure by using mobile phone data. *Proc. Natl Acad. Sci. USA* **106**, 15274–15278 (2009).
13. Cattuto, C. et al. Dynamics of person-to-person interactions from distributed rfid sensor networks. *PLoS ONE* **5**, e11596 (2010).
14. Barrat, A. et al. Empirical temporal networks of face-to-face human interactions. *Eur. Phys. J. Spec. Top.* **222**, 1295–1309 (2013).
15. Barrat, A. & Cattuto, C. in *Temporal Networks* (eds Saramäki, J. & Holme, P.) (Springer, 2013).
16. Stopczynski, A. et al. Measuring large-scale social networks with high resolution. *PLoS ONE* **9**, e95978 (2014).
17. Bravo-Hermsdorff, G. et al. Gender and collaboration patterns in a temporal scientific authorship network. *Appl. Netw. Sci.* **4**, 1–17 (2019).
18. Cazabet, R., Jensen, P. & Borgnat, P. Tracking the evolution of temporal patterns of usage in bicycle-sharing systems using non-negative matrix factorization on multiple sliding windows. *Int. J. Urban Sci.* **22**, 147–161 (2018).
19. Carstensen, T. A., Olafsson, A. S., Bech, N. M., Poulsen, T. S. & Zhao, C. The spatio-temporal development of copenhagen's bicycle infrastructure 1912–2013. *Geografisk Tidsskr. Dan. J. Geogr.* **115**, 142–156 (2015).
20. Bellet, A., Habrard, A. & Sebban, M. Metric learning. *Synth. Lectures Artif. Intell. Mach. Learn.* **9**, 1–151 (2015).
21. Wills, P. & Meyer, F. G. Metrics for graph comparison: a practitioner's guide. *PLos ONE* **15**, e0228728 (2020).
22. Tantardini, M., Ieva, F., Tajoli, L. & Piccardi, C. Comparing methods for comparing networks. *Sci. Rep.* **9**, 1–19 (2019).
23. Harrison, H. et al. Network comparison and the within-ensemble graph distance. *Proc. R. Soc. A* **476**, 20190744 (2020).
24. Barros, C. D., Mendonça, M. R., Vieira, A. B. & Ziviani, A. A survey on embedding dynamic graphs. *ACM Comput. Surv.* **55**, 1–37 (2021).
25. Ma, G., Ahmed, N. K., Willke, T. L. & Yu, P. S. Deep graph similarity learning: a survey. *Data Min. Knowl. Discov.* **35**, 688–725 (2021).
26. Sanfeliu, A. & Fu, K.-S. A distance measure between attributed relational graphs for pattern recognition. In *IEEE Transactions on Systems, Man, and Cybernetics* 353–362 (IEEE, 1983).
27. Koutra, D., Vogelstein, J. T. & Faloutsos, C. Deltacon: a principled massive-graph similarity function. In *Proc. the 2013 SIAM International Conference on Data Mining* 162–170 (SIAM, 2013).
28. Monnig, N. D. & Meyer, F. G. The resistance perturbation distance: a metric for the analysis of dynamic networks. *Discret. Appl. Math.* **236**, 347–386 (2018).
29. Berlingerio, M., Koutra, D., Eliassi-Rad, T. & Faloutsos, C. Netsimile: a scalable approach to size-independent network similarity. Preprint at arXiv:1209.2684 (2012).
30. Bagrow, J. P. & Bollt, E. M. An information-theoretic, all-scales approach to comparing networks. *Appl. Netw. Sci.* **4**, 1–15 (2019).
31. Apolloni, N. N. W. B. et al. An introduction to spectral distances in networks. In *Neural Nets WIRN10: Proc. 20th Italian Workshop on Neural Nets* (IOS Press, 2011).
32. Shimada, Y., Hirata, Y., Ikeguchi, T. & Aihara, K. Graph distance for complex networks. *Sci. Rep.* **6**, 1–6 (2016).

33. Torres, L., Suárez-Serrato, P. & Eliassi-Rad, T. Non-backtracking cycles: length spectrum theory and graph mining applications. *Appl. Netw. Sci.*, **4**, 1–35 (2019).
34. Tsitsulin, A., Mottin, D., Karras, P., Bronstein, A. & Müller, E. Netlsd: hearing the shape of a graph. In *Proc. 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 2347–2356 (Association for Computing Machinery, 2018).
35. Donnat, C. & Holmes, S. Tracking network dynamics: a survey using graph distances. *Ann. Appl. Stat.* **12**, 971–1012 (2018).
36. Masuda, N. & Holme, P. Detecting sequences of system states in temporal networks. *Sci. Rep.* **9**, 1–11 (2019).
37. Gelardi, V., Fagot, J., Barrat, A. & Claidière, N. Detecting social (in) stability in primates from their temporal co-presence network. *Anim. Behav.* **157**, 239–254 (2019).
38. Pedreschi, N. et al. Dynamic core-periphery structure of information sharing networks in entorhinal cortex and hippocampus. *Netw. Neurosci.* **4**, 946–975 (2020).
39. Beladev, M., Rokach, L., Katz, G., Guy, I. & Radinsky, K. tdgraphembed: temporal dynamic graph-level embedding. In *Proc. 29th ACM International Conference on Information & Knowledge Management* 55–64 (Association for Computing Machinery, 2020).
40. Huang, C., Wang, L., Cao, X., Ma, W. & Vosoughi, S. Learning dynamic graph embeddings using random walk with temporal backtracking. In *NeurIPS 2022 Temporal Graph Learning Workshop* (2022).
41. Le Bail, D., Génois, M. & Barrat, A. Flow of temporal network properties under local aggregation and time shuffling:a tool for characterizing, comparing and classifying temporal networks. *J. Phys. A Math. Theor.* http://iopscience.iop.org/article/10.1088/1751-8121/ad7b8e (2024).
42. Froese, V., Jain, B., Niedermeier, R. & Renken, M. Comparing temporal graphs using dynamic time warping. *Soc. Netw. Anal. Min.* **10**, 1–16 (2020).
43. Zhan, X. -X. et al. Measuring and utilizing temporal network dissimilarity. Preprint at arXiv:2111.01334 (2021).
44. Dall'Amico, L., Barrat, A. & Cattuto, C. An embedding-based distance for temporal graphs. Preprint at https://doi.org/10.5281/zenodo.13880695 (2024).
45. Rossetti, G. & Cazabet, R. Community discovery in dynamic networks: a survey. *ACM Comput. Surv.* **51**, 1–37 (2018).
46. Gromov, M., Katz, M., Pansu, P. & Semmes, S. *Metric structures for Riemannian and Non-Riemannian Spaces* (Springer, 1999).
47. Cai, H., Zheng, V. W. & Chang, K. C.-C. A comprehensive survey of graph embedding: problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.* **30**, 1616–1637 (2018).
48. Goyal, P. & Ferrara, E. Graph embedding techniques, applications, and performance: a survey. *Knowl. Based Syst.* **151**, 78–94 (2018).
49. Xu, M. Understanding graph embedding methods and their applications. *SIAM Rev.* **63**, 825–853 (2021).
50. Makarov, I., Kiselev, D., Nikitinsky, N. & Subelj, L. Survey on graph embeddings and their applications to machine learning problems on graphs. *PeerJ Comput. Sci.* **7**, e357 (2021).
51. Gauvin, L., Panisson, A. & Cattuto, C. Detecting the community structure and activity patterns of temporal networks: a non-negative tensor factorization approach. *PLoS ONE* **9**, e86028 (2014).
52. Chi, Y., Song, X., Zhou, D., Hino, K. & Tseng, B. L. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 153–162 (Association for Computing Machinery, 2007).
53. Qin, X., Dai, W., Jiao, P., Wang, W. & Yuan, N. A multi-similarity spectral clustering method for community detection in dynamic networks. *Sci. Rep.* **6**, 1–11 (2016).
54. Liu, F., Choi, D., Xie, L. & Roeder, K. Global spectral clustering in dynamic networks. *Proc. Natl Acad. Sci. USA* **115**, 927–932 (2018).
55. Xu, K. S., Kliger, M. & Hero III, A. O. Adaptive evolutionary clustering. *Data Min. Knowl. Discov.* **28**, 304–336 (2014).
56. Dall'Amico, L., Couillet, R. & Tremblay, N. Community detection in sparse time-evolving graphs with a dynamical bethe-hessian. *Adv. Neural Inf. Process. Syst.* **33**, 7486–7497 (2020).
57. Zuo, Y. et al. Embedding temporal network via neighborhood formation. In *Proc. 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 2857–2866 (Association for Computing Machinery, 2018).
58. Nguyen, G. H. et al. Continuous-time dynamic network embeddings. In *Companion Proceedings of the Web Conference 2018* 969–976 (Association for Computing Machinery, 2018).
59. Sato, K., Oka, M., Barrat, A. & Cattuto, C. Predicting partially observed processes on temporal networks by dynamics-aware node embeddings (dyane). *EPJ Data Sci.* **10**, 22 (2021).
60. Saramäki, J. & Holme, P. Exploring temporal networks with greedy walks. *Eur. Phys. J. B* **88**, 1–8 (2015).
61. Dall'Amico, L. & Belliardo, E. M. Efficient distributed representations with linear-time attention scores normalization. Preprint at arXiv:2303.17475 (2023).
62. Isella, L. et al. What's in a crowd? analysis of face-to-face behavioral networks. *J. Theor. Biol.* **271**, 166–180 (2011).
63. Erdős, P. & Rényi, A. et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* **5**, 17–60 (1960).
64. Karrer, B. & Newman, M. E. Stochastic blockmodels and community structure in networks. *Phys. Rev. E* **83**, 016107 (2011).
65. Bollobás, B. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *Eur. J. Comb.* **1**, 311–316 (1980).
66. Dall, J. & Christensen, M. Random geometric graphs. *Phys. Rev. E* **66**, 016121 (2002).
67. Févotte, C. & Idier, J. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Comput.* **23**, 2421–2456 (2011).
68. MacQueen, J. et al. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability* 281–297 (University of California Press, 1967).
69. Sainburg, T., McInnes, L. & Gentner, T. Q. Parametric UMAP embeddings for representation and semisupervised learning. *Neural Comput.* **33**, 2881–2907 (2021).
70. Gauvin, L. et al. Randomized reference models for temporal networks. *SIAM Rev.* **64**, 763–830 (2022).
71. Pan, R. K. & Saramäki, J. Path lengths, correlations, and centrality in temporal networks. *Phys. Rev. E* **84**, 016105 (2011).
72. Goodfellow, I. et al. Generative adversarial nets. *Adv. Neural Inform. Process. Syst.* **27** (2014).
73. Faez, F., Ommi, Y., Baghshah, M. S. & Rabiee, H. R. Deep graph generators: a survey. *IEEE Access* **9**, 106675–106702 (2021).
74. Alqahtani, H., Kavakli-Thorne, M. & Kumar, G. Applications of generative adversarial networks (gans): an updated review. *Arch. Comput. Methods Eng.* **28**, 525–552 (2021).
75. Machens, A. et al. An infectious disease model on empirical networks of human contact: bridging the gap between dynamic network data and contact matrices. *BMC Infect. Dis.* **13**, 1–18 (2013).
76. Gemmetto, V., Barrat, A. & Cattuto, C. Mitigation of infectious disease at school: targeted class closure vs school closure. *BMC Infect. Dis.* **14**, 1–10 (2014).
77. Mistry, D. et al. Inferring high-resolution human mixing patterns for disease modeling. *Nat. Commun.* **12**, 323 (2021).
78. Cencetti, G. et al. Digital proximity tracing on empirical contact networks for pandemic control. *Nat. Commun.* **12**, 1655 (2021).
79. Crețu, A.-M. et al. Interaction data are identifiable even across long periods of time. *Nat. Commun.* **13**, 313 (2022).

80. Colosi, E. et al. Screening and vaccination against covid-19 to minimise school closure: a modelling study. *Lancet Infect. Dis.* **22**, 977–989 (2022).
81. Backstrom, L., Dwork, C. & Kleinberg, J. Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In *Proc. 16th International Conference on World Wide Web* 181–190 (Association for Computing Machinery, 2007).
82. Romanini, D., Lehmann, S. & Kivelä, M. Privacy and uniqueness of neighborhoods in social networks. *Sci. Rep.* **11**, 20104 (2021).
83. Horn, R. A. & Johnson, C. R. *Matrix Analysis* (Cambridge Univ. Press, 2012).
84. Stehlé, J. et al. High-resolution measurements of face-to-face contact patterns in a primary school. *PLoS ONE* **6**, e23176 (2011).
85. Fournet, J. & Barrat, A. Contact patterns among high school students. *PLoS ONE* **9**, e107878 (2014).
86. Mastrandrea, R., Fournet, J. & Barrat, A. Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PLoS ONE* **10**, e0136497 (2015).
87. Gelardi, V., Godard, J., Paleressompoulle, D., Claidière, N. & Barrat, A. Measuring social networks in primates: wearable sensors versus direct observations. *Proc. R. Soc. A* **476**, 20190737 (2020).
88. Ozella, L. et al. Using wearable proximity sensors to characterize social contact patterns in a village of rural malawi. *EPJ Data Sci.* **10**, 46 (2021).
89. Vanhems, P. et al. Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. *PLoS ONE* **8**, e73970 (2013).
90. Génois, M. & Barrat, A. Can co-location be used as a proxy for face-to-face contacts? *EPJ Data Sci.* **7**, 1–18 (2018).

Acknowledgements

L.D. and C.C. acknowledge support from the Lagrange Project of the ISI Foundation, funded by CRT Foundation, and from Fondation Botnar (EPFL COVID-19 Real-Time Epidemiology I-DAIR Pathfinder). A.B. acknowledges support from the Agence Nationale de la Recherche (ANR) project DATAREDUX (ANR-19-CE46-0008).

Author contributions

L.D. formalized the research problem, developed computer code, carried out the analysis, and wrote the first version of the manuscript. L.D., A.B., and C.C. contributed to framing the research questions, discussing the results, and writing the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-024-54280-4.

Correspondence and requests for materials should be addressed to Lorenzo Dall'Amico.

Peer review information *Nature Communications* thanks Amir Ghasemian, and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2024