Neighbourhood matching creates realistic surrogate temporal networks

**Antonio Longa**  
Fondazione Bruno Kessler  
https://orcid.org/0000-0003-0337-1838

**Giulia Cencetti**  
Fondazione Bruno Kessler  
https://orcid.org/0000-0002-6946-3666

**Sune Lehmann**  
Technical University of Denmark  
https://orcid.org/0000-0001-6099-2345

**Andrea Passerini**  
University of Trento  
https://orcid.org/0000-0001-6099-2345

**Bruno Lepri** (lepri@fbk.eu)  
Fondazione Bruno Kessler  
https://orcid.org/0000-0003-1275-2333

---

**Article**

**Keywords:**

**Posted Date:** May 17th, 2022

**DOI:** https://doi.org/10.21203/rs.3.rs-1638968/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.  
[Read Full License](https://creativecommons.org/licenses/by/4.0/)
Neighbourhood matching creates realistic surrogate temporal networks

A. Longa\textsuperscript{1,2}, G. Cencetti\textsuperscript{1}, S. Lehmann\textsuperscript{3,4}, A. Passerini\textsuperscript{2}, and B. Lepri\textsuperscript{1}

\textsuperscript{1}Fondazione Bruno Kessler, Trento, Italy
\textsuperscript{2}University of Trento, Trento, Italy
\textsuperscript{3}Technical University of Denmark, Kongens Lyngby, Denmark
\textsuperscript{4}Copenhagen Center for Social Data Science, Copenhagen, Denmark

\textsuperscript{⋆}Corresponding author: lepri@fbk.eu

ABSTRACT Temporal networks are essential for modeling and understanding systems whose behavior varies in time, from social interactions to biological systems. Often, however, real-world data are prohibitively expensive to collect or unshareable due to privacy concerns. A promising solution is ‘surrogate networks’, synthetic graphs with the properties of real-world networks. Until now, the generation of realistic surrogate temporal networks has remained an open problem, due to the difficulty of capturing both the temporal and topological properties of the input network, as well as their correlations, in a scalable model. Here, we propose a novel and simple method for generating surrogate temporal networks. By decomposing graphs into temporal neighborhoods surrounding each node, we can generate new networks using neighborhoods as building blocks. Our model vastly outperforms current methods across multiple examples of temporal networks in terms of both topological and dynamical similarity. We further show that beyond generating realistic interaction patterns, our method is able to capture intrinsic temporal periodicity of temporal networks, all with an execution time lower than competing methods by multiple orders of magnitude.

1 Introduction

Across the past decade, temporal networks have driven breakthroughs in real world systems across biology, communications, social interactions, and mobility. The power of temporal networks resides in their ability to capture complex dynamics such as dif-
fusion and contagion. In order to model realistic dynamics, it is often necessary to employ large temporal networks, including a large number of nodes and many temporal layers. Many state-of-the-art temporal datasets, however, are limited both in the number of agents and in the number of temporal layers. When the available data are insufficient – e.g. in long-term epidemiological simulations – datasets are extended by simply repeating the same temporal sequence multiple times, a procedure which is known to result in biases. An appealing solution to the problem of insufficient or privacy sensitive data is to use surrogate temporal networks. Surrogate temporal networks are synthetic datasets which mimic the real-world temporal patterns relevant for a desired use-case, where the actual dynamics are known through smaller studies or via available small datasets. In the case of sensible data, such as fine-grained records of social interactions, surrogate data can freely be shared. Knowledge of actual dynamics is often available because real temporal networks are known to be characterised by typical patterns of interactions, different in different domains (social, biological, infrastructural, etc.), but that we can often recognize and delineate. For these reasons, it is clear that surrogate temporal networks are highly desirable from the perspective of a number of applications. Over the past years, a large number of successful algorithms for static network generation have been proposed; however, extending these models to the dynamic regime has proven prohibitively difficult, due to greatly increased complexity introduced by the temporal dimension.

Indeed, it has become clear that temporal networks are characterized by a highly non-trivial interplay between the network topology (adjacency, degree distribution, clustering, etc.) at a given time and the temporal activation of nodes and links – how each connection changes over time (duration of interactions, patterns by which new links appear and old ones disappear, etc.). From the ‘egocentric’ perspective of an individual node, these two dimensions imply that models must take into account (i) the history of what has occurred in the preceding timesteps and (ii) the current activations of the neighboring nodes. Current network theory has not yet been able to fully understand and model the interplay between the two dimensions. So far, it is not even clear which statistics to measure. In fact, the scientific literature is full of studies focusing on the spatial dimension but unable to take into account possible temporal correlations, or – alternatively – work dedicated to model the behavior of individual nodes in time (for
example activity driven models\textsuperscript{7,31} which does not reproduce realistic network topologies\textsuperscript{32}. There exist models for link prediction that try to combine temporal and topological dimensions by using small local temporal patterns\textsuperscript{33} or building over a backbone of significant links.\textsuperscript{18} However, there is currently a dearth of models for generating surrogate networks from scratch that are able to take into account the two dimensions simultaneously. The few works, that do this, rely on temporal motifs, like Dymond\textsuperscript{34} and STM\textsuperscript{35} or on deep learning like TagGen\textsuperscript{36}. These three models described in detail in Methods, represent the state-of-the-art. We show below, however, that all these techniques generate temporal networks with massive macroscopic differences compared with the original temporal network datasets, and that – in many cases – the output-networks do not reflect the dynamic behavior of the original network.

In this work, we propose a method able to generate high temporal resolution surrogate networks that are able to match real-networks in terms of a wide range of topological and dynamic measures. Our generative algorithm is based on the idea of the egocentric temporal neighborhood\textsuperscript{37} $\mathcal{E}_n^{\{t-k,\ldots,t\}}$ for node $n$ at time $t$, including a small number $k$ of prior time steps. Here we assume that the network is represented in discrete time with each time step corresponding to a static graph, also referred to as a ‘layer’ of the network. Crucially, $\mathcal{E}_n^{\{t-k,\ldots,t\}}$ does not include interactions between the neighbors of $n$. To avoid excessive notation in the following, we simply use the term ‘neighborhood’ to describe the egocentric temporal neighborhood when there is no risk of confusion.

Conceptually our algorithm does the following. We first characterize a given real-world network in terms of neighborhoods, and then use those neighborhoods as building blocks for a new synthetic network. When we match up neighborhoods, conflicts among the egocentric perspectives of different nodes are globally solved by combining overlapping sub-networks so as to preserve as much as possible each node’s desired neighborhood. In order to extend the network into subsequent time steps, we build a local probabilistic model for suggesting new temporal interactions at time $t+1$ for each node, given the behavior during $\{t-k,\ldots,t\}$. We can further increase realism corresponding to activity modulation, such as day/night and week/weekend by building distinct probabilistic models for different times of the day or days of the week.

A major advantage of the egocentric perspective (that ignores connections among neigh-
bors of an ego node) is that it allows us to linearize the concept of node neighborhood sidestepping the subgraph isomorphism problem, making the generation process fast and scalable both in terms of the number of nodes and the number of temporal snapshots. Speed turns out to be a fundamental feature, because the other existing methods rely on algorithms of considerably higher complexity that prevent those methods from scaling to even moderately-sized networks.

We test the method, named *Egocentric Temporal Neighborhood Generator (ETN-gen)*, on a range of different temporal networks. In our testing, we mainly use social interactions datasets, because of richness and availability of these datasets, but the method is general and can be used to generate any kind of graph. The simplicity of our algorithm makes it easily interpretable, extendable and algorithmically scalable. As we show below, the surrogate networks that we generate match original networks with a high degree of accuracy, not just in terms of local features, as one might anticipate from the local generating mechanism, but with respect to global features, such as the number of interactions, the number of interacting individuals in time and density of their connections. The ability to generate surrogate temporal graphs that reproduce real behaviors allows us to obtain large as desired data, without resolution limits, while mitigating key privacy issues.

2 Results

We first briefly sketch the temporal graph generation process. Then, we use our method to generate temporal graphs which reproduce the temporal interaction patterns of a diverse set of face-to-face interaction networks, including a hospital, a workplace and a high school. See *Methods* for details on the datasets. We evaluate the quality of the generated networks in terms of interaction statistics, considering both static and temporal network properties, highlighting the advantage of our proposed method relative to the state-of-the-art. Finally, we show how the approach can be used to expand existing temporal networks, both in time and in number of nodes, something which is not possible using competing methods for temporal network generation.
2.1 The neighborhood generation process

Figure 1: ETN-gen. Top panel shows how egocentric temporal neighbourhood signatures are extracted and computed. Panel B shows how to build the probability distribution of neighborhoods, necessary to generate a provisional layer. The panel C shows how to generate a provisional layer, while panel D explains how to convert the provisional layer into a definitive one.
generation process for a small temporal network with three timesteps (see Methods for
details). Panel A shows the egocentric temporal network \( E_{n}^{\{t-k,...,t\}} \) – or simply ‘neighborhood’ – of a node \( n \). We extract this neighborhood for each node in a graph. Specifically, for a given time horizon \( k \) (\( k = 2 \) in the figure) and a given egocentric node \( n \) (\( n = E \) in the figure), \( E_{n}^{\{t-k,...,t\}} \) is defined as the network fragment which contains \( n \) and its neighbors at each of \( k + 1 \) consecutive timestamps, discarding connections between the neighbors of \( n \), and adding (temporal) connections among instances of the same node at different timestamps. Having discarded links between neighbors, \( E_{n}^{\{t-k,...,t\}} \) can be encoded as a binary string, where for each neighbor node and timestamp 1 (resp. 0) indicates the presence (resp. absence) of a link connecting to the node at that timestamp. Such neighborhoods are extracted for all nodes and all timestamps by using a sliding window over time.

Second (panel B), we build a local probability distribution designed to enable simulation of activity in future time steps. This distribution to extend the graph into future time steps is based on past neighborhood activity. Specifically the local distribution maps neighborhoods of length \( k - 1 \) (i.e. temporal neighborhoods involving \( k \) steps) to the set of all possible extensions into the future (i.e. neighborhoods of temporal depth \( k \), involving \( k + 1 \) steps), with associated probabilities estimated by Maximum Likelihood over the original temporal network. Third (panel C), for each node in the network we generate a provisional temporal extension by sampling from the probability distribution described above. We thus obtain a provisional temporal layer of the network. Last (panel D), this provisional layer is finalized by combining provisional temporal extensions of all nodes, resolving conflicts and dangling links. To connect neighborhoods, we consider a connection from node \( i \) to node \( j \) in the provisional layer a ‘request’ of \( i \) to be connected to \( j \). If this request is reciprocal, the link is validated and added to the new temporal layer (see the second step in panel D). All remaining one-directional links are validated with probability \( \alpha = 1/2 \) (third step), to preserve the overall number of connections (an \( i-j \) connection can be requested by \( i \) or by \( j \)). The procedure is repeated as many times as the desired length of the final temporal graph, always considering the last \( k \) timestamps as seeds and generating an additional one.

Above, we have described the simplest possible strategy for extending a layer into the future, but note that all random choices in the link validation process could become
preferential choices in order to optimize a specific characteristic of the final network (see Section Topological similarity evaluation). Further – which we explore below – the temporal extension can include novel nodes that are not present in the current temporal neighborhood of the ego node, and thus their identity is not known (the question mark in panel C). These nodes are connected by a ‘stub’ rather than a real connection, representing a link-to-be, and stubs are pairwise matched up at random (last step in panel D).

With the basic mechanisms in place, we take a step back and explain how to initialize the process, i.e. how to obtain the first \( k \) layers of the graph. The graph at the first timestamp is generated using a configuration model\(^4\), reproducing the degree distribution of the first layer of the original graph. The following layers up to \( k \) are generated by applying the procedure in Figure 1 to the first layer with \( k' = 1 \), to the first two layers with \( k' = 2 \) and so on until \( k' = k \).

Finally, temporal networks are often characterized by an intrinsic periodicity. In social interactions data for instance this can be due to the day-night cycle or to the difference between work days and weekends, and the organization of our societies. This is accounted for in our generation procedure by using distinct local probability distributions to extend the graph during different days of the week or times of the day. In the experiments in this paper we use distinct week/weekends or daily local probability distributions, depending on the length and variability of the input network.

### 2.2 Model evaluation

We now evaluate the quality of the generated networks based on interaction statistics by comparing the networks to empirical data as well as networks generated by a suite of state-of-the-art temporal network generation methods described below. We evaluate performance in terms of individual layer topology as well as temporal behavior. The key new feature of our network generation procedure is the ability to approximately reproduce the interaction statistics of real-world data, something the existing alternatives cannot do.

The state-of-the-art methods we consider are: Dyndom\(^3\), a model which uses the dis-
tribution of 3-nodes structures in the original graph (triads with one, two or three connections) as building blocks to generate a new temporal network; STM\textsuperscript{35} a generative model based on the distribution of small temporal motifs; and TagGen\textsuperscript{36} based on deep learning, which uses a generative adversarial network to generate temporal walks that are then combined into a temporal graph. Dymond and STM only consider local information, while TagGen is more global.

It is important to underscore that these network generation methods have not necessarily been developed with the aim of generating large temporal networks with low computational cost (see Section S2). This means that, for example, they require much more training data, need denser temporal snapshots, and therefore cannot generate high temporal resolution networks. In this regard ETN-gen, thanks to the linearization due to the egocentric perspective, is the first method that allows researchers to scale to arbitrarily sized temporal networks. In the rest of the paper, in order to evaluate our work relative to the other methods, we will report experiments only on the three smallest face-to-face interactions datasets, collected in the hospital\textsuperscript{39} in the workplace\textsuperscript{40} and in one of the high schools\textsuperscript{41} respectively. Results applying ETN-gen to larger datasets are reported in the Supplementary Information (SI).

Figure 2 reports the total number of interactions for each temporal snapshot (left) and the average number of nodes (right) in the original network, ETN-gen and the three competitors. The first clear finding from this figure is that ETN-gen (orange curves) results in time-series that are remarkably similar to those appearing in the original datasets (black curves). This is true, not just in terms of generating a number of interactions which is of the same order of magnitude as the original data (notice that different datasets have different scales on the y-axis), but also in terms of temporal patterns which are preserved with considerable accuracy, including daily and weekly periodicity.

This result, even if outstanding relative to Dymond, STM, and TagGen should not come as a surprise, as it is a direct consequence of our network generation procedure. The local probabilistic models store the probability distributions of the neighborhoods appearing in the original graph and this indirectly contains the key information about how nodes degree evolves in time. Further, our seed-network has the same degree distribution as the original graph, which allows us to statistically preserve the overall average
number of interactions of the original graph. Further, we manually input periodicity via different local probabilistic models for different times and days of the week. We highlight, however, that while using only a single local probabilistic model would remove our ability to model periodic changes in graph over time, we would still be able to model the average number of interactions, as these are automatically reproduced by the rest of the algorithm.

In contrast to ETN-gen, the results that we obtain from the current methods are significantly different from the empirical ones. To start, the curves representing TagGen, only appear in the insets, which report the same data with different $y$-axes. The number of interactions generated by TagGen is orders of magnitude larger than that of the other methods and of the original network. TagGen, however, does manage to capture day-night periodicity, which is completely lost by Dymond and STM, both of which produce a number of interactions that is stable over time. Overall, ETN-gen is the only method capable of accurately reproducing both aggregated and temporal interaction statistics.

The histograms on the right part of the figure show mean and standard deviation of the number of nodes in the networks generated by the different methods, with the horizontal dashed line representing the number of nodes in the original network. These histograms show another important result: only our method and TagGen always generate networks with the same number of nodes of the input graph, with the difference that, as shown by the insets on the left, TagGen generates orders of magnitude more interactions. Dymond and STM, on the other hand, respectively under and over represent the number of nodes, so that only ETN-gen manages to reproduce both the number of nodes and the number of interactions in the original network. The curves for the original network and ETN-gen are also reported in Figure S1 for larger datasets to which the other methods cannot be applied due to computational constraints.

### 2.3 Topological similarity evaluation

Having studied the temporal development, we now turn to structural similarity between the surrogate data and the original networks. We consider ten metrics for structural similarity: number of interactions, density,$^{34}$ interacting individuals,$^{11}$ new conversations,$^{11}$ S-metric,$^{44}$ duration of contacts,$^{11}$ edge strength in the projected weighted
Figure 2: Number of interactions in the generated network vs. competitors. Number of interactions at each timestamp, each color represents a different generation algorithm, while the original graph is depicted in black. TagGen produces a number of interactions ten times the order of the original network and it only appears in the insets for visibility.

network,[1] global clustering coefficient,[15,16] assortativity,[17] and average shortest path length,[1] (see SI for their definitions). In particular, duration of contacts and edge strength are measures that can be collected for each edge, so we obtain a distribution over all edges. All the other observables represent global characteristics of static networks, that can be measured on each singular temporal layer, such obtaining distributions over snapshots.

To compare distributions, we rely, inspired by Zeno et al.,[34] on the Kolmogorov-Smirnov
Figure 3: **Topological similarity.** Similarity of the original network with those generated by **ETN-gen, STM, TagGen and Dymond.** Each bar reports the Kolmogorov-Smirnov distance between the two distributions (original and generated) for a specific structural metric. The shorter is a bar the more similar are the distributions. Standard deviations are obtained over 10 stochastic realizations of each network. In the top inset we report the distributions of the number of interactions in real and in one instance of generated networks for the Workplace dataset.
distance\textsuperscript{[8]} to compare generated and original graphs. Distances between distributions are reported in Figure\textsuperscript{3}, where we compare graphs obtained with \textit{ETN-gen} with those from the three alternative approaches.

\textit{ETN-gen} (orange bars) clearly generates surrogate networks that are closest to the empirical networks for almost all measures, regardless of the dataset considered. Moreover, our method is substantially more stable than the competitors, as shown by the error bars which were obtained over 10 stochastic realizations of each network.

The measures for which \textit{ETN-gen} performs best are those that, together with the number of interactions (see Figure\textsuperscript{2}), are preserved by construction: the density and the number of interacting individuals in time. Here, the similarity originates from the neighborhood probability distributions, which ensure that from a statistical viewpoint, the surrogate network has the same number of interactions and the same number of individuals involved in an interaction. The same holds for the number of times that a new link appears, as these statistics are also stored in the neighborhood probability distributions. Another characteristic that is entirely captured by the egocentric temporal neighborhoods is the hub-like structure that we can find in each static layer, which is measured by the S-metric\textsuperscript{[44]}

Going beyond these ‘trivial’ consequenses of the mechanics of the generating mechanisms, the method does well at preserving interaction durations. The $k$-steps memory makes various duration possible, even long durations (because of the sliding window) unlike the case of independent layers. Moreover, the ETN distributions also encode the number of times that an interaction ends, so interactions tend not to be extremely long.

Another interesting property is the distribution of edge strengths in the projected graph. Edge strength is simply the number of times that each edge has appeared over the duration of the graph. Here, we would not necessarily expect \textit{ETN-gen} to do well as the method will tend to create networks with quite homogeneous distributions of strength. This is because it can only rely on a memory of order $k$ for edge repetitions, and does not have a long-term memory. Hence all the heterogeneous behaviors that we can find for instance in social datasets, where individuals tend to establish relationships with specific nodes and have repeated (but not necessarily consecutive) interactions with them, are not preserved by \textit{ETN-gen}. Nevertheless, we find that for the considered datasets
ETN-gen remains competitive with the other methods.

If edge strength is partially affected by the absence of long memory, the most important limitations of the egocentric perspective are highlighted by clustering, degree assortativity and average shortest path length, which are related to second-order interactions. This is the cost we pay for having a computationally efficient model applicable to arbitrary networks. Notice that while this is a problem in theory, it seems not to affect the workplace dataset, at least for clustering coefficient and average shortest path length. This is explained by the fact that the dataset is substantially sparser (see Figure S1), hence characterized by low clustering and short paths. More importantly, it must be stressed that the other approaches also are not able to reproduce these metrics, thus our method is still the most competitive on average.

At this point we note that the limitations with respect to second-order measures could be mitigated during the last step of our new temporal layer generation. In the current version we go from a prospective layer to the actual new layer by matching up nodes with a one-way suggested connection at random. At this step, we could however apply a preferential attachment devoted to maximize or minimize a specific variable. For instance, to maximize clustering we could prefer to keep edges whose nodes have one or more common neighbors, to maximize (minimize) assortativity we could connect stubs with similar (dissimilar) degree.

2.4 Dynamical similarity evaluation

Having tested our method from the structural point of view, we now test the usefulness of the surrogate networks in terms of dynamical processes unfolding upon them. We study two dynamical models: random walk and a spreading model.

2.4.1 Random walk

We simulate a temporal random walk\textsuperscript{11,24} on the original and generated networks. We compute two metrics: coverage and mean first passage time (MFPT), and compare distributions over different realizations between the input and the generated temporal net-
work using again the Kolmogorov-Smirnov distance (see SI for the definition of the metrics).

In Figure 4 (left) we report the Kolmogorov-Smirnov distance for coverage and MFPT with our method and the competitors with respect to the original network. The horizontal dashed line shows the stability of each measure on the original network. The black line is obtained comparing different performances (average over 1000 simulations of random walk for coverage, and 5 times each couple of nodes for mean first passage time) by means of the Kolmogorov-Smirnov distance. We observe that in terms of the mean first passage time ETN-gen performs better than the competitors, while in terms of coverage performance depends on the datasets: for the most dense networks (hospital and high school) the highest similarity with the original network is achieved by Dymond (but ETN-gen is ranked second), while for the workplace ETN-gen again largely outperforms all the competitors, that produce very different distributions from the original one. In general we can say that the random walk process on the ETN-gen’s surrogate networks is quite similar to the random walk on the original graph, especially if compared with surrogate data from the other methods.
2.4.2 Spreading model

We simulate a Susceptible-Infectious-Recovered (SIR) model\textsuperscript{[19]} with three possible values for the probability of disease transmission ($\lambda \in \{0.25, 0.13, 0.01\}$), and the recovery probability fixed at $\mu = 0.055$. We compute the reproduction value $R_0$ (see SI for the definition). Each experiment was repeated 100 times and the distribution of $R_0$ obtained on the original network is, again, compared with those obtained on synthetic networks by means of the Kolmogorov-Smirnov distance. Results are shown in Figure 4 (right), where again a horizontal black line shows the stability of each measure on the original network (computed averaging over 100 simulations). We observe that the results obtained with \textit{ETN-gen} are highly similar to those of the original graph and show a large degree of stability with respect to this similarity. The other methods produce networks where the dynamical behavior similarity is sometimes high and sometimes severely low, being quite sensitive to parameters and datasets. In conclusion, the method that we propose almost always produces the most similar networks to the original ones, and certainly the most stable results across datasets, across dynamical systems and across parameters.

2.5 Dataset expansion and extension

In the previous sections we have argued that \textit{ETN-gen} creates realistic surrogate temporal networks that mimic real social dynamics (both in terms of structure and in reproducing dynamical systems) – and that our method outperforms alternative solutions.

Now we ask the question: How can this tool be useful in practice? A relevant application is represented by the possibility of enlarging a given temporal dataset, both in time and in size. It is indeed common that a specific analysis, in order to yield reliable results, requires a larger population or a longer time than those characterizing collected real data. In those cases we deal with the long-standing problem of data augmentation, for which we now argue that \textit{ETN-gen} represents a promising solution. In the following we show how our method can be used for augmenting a temporal dataset, by adding temporal layers (temporal extension), but also by increasing the size of the network in terms of number of nodes (size expansion).
2.5.1 Temporal extension

The temporal extension of a dataset is straightforward: once we have calculated the neighborhood probability distributions summarizing the original graph, we can repeat the process of temporal layer addition as long as needed. At the top of Figure 5 we show an example of temporal extension of the workplace network. We have selected this dataset to show that ETN-gen is capable of capturing weekends (with no interactions) as well. To evaluate the quality of the extension, we use only the first week of the original two-week dataset (from the beginning to the vertical line) to estimate the neighborhood probability distributions. We now generate an ensemble of 10 two-week networks based on that first week. The mean and standard deviation of the number of interactions in the generated graph are reported in orange. The number of interactions of the original graph are reported in black dashed curves for the first week (the “train” dataset), and in black solid curves for the following week (the “original” dataset). Results show how the generated networks accurately recreate the original behavior beyond the timespan that was used to estimate the local probability distributions.

2.5.2 Size expansion

Here we explore the fidelity of surrogate networks with an increased number of nodes. As discussed above, it is possible to increase the size beyond that of the original network within the ETN-gen framework because the number of nodes is simply a parameter to set for the method. That said, however, the concept of size expansion requires more attention than time extension. Because, as we change the number of nodes in a network we should also consider how the density of the graph and the mean degree should change accordingly.

In the following we describe an experiment of data augmentation, assuming that we only have access to incomplete data. Incomplete data are obtained by randomly removing part of the nodes from the original network. We use the high school dataset which, with its 126 nodes, is the largest among our datasets, and we consider two reduced versions, with 30% and 70% of the nodes respectively. When removing part of the nodes from a network, we naturally remove also part of the links (all those which were be-
Figure 5: **Temporal extension and node expansion.** The mean and standard deviation of our method are shown in orange (and brown). Black dashed (and dotted) lines show the original data used to train our model, while black solid lines show the original data used to evaluate the quality of the generated network. In experiments involving temporal expansion, a vertical bar separates the temporal range used to collect training data from the one where expansion is performed. See the main text for the details.

Before connecting the eliminated nodes to the remaining ones, we hence reduce the mean degree. We should consider that an incomplete dataset has in general a reduced mean degree with respect to the real-world network, and that when we try to reconstruct the original network via data augmentation we should increase the mean degree too. See Methods for a quantification of the needed increase.

Anyway, once the desired connectivity has been chosen, ETN-*gen* allows us to generate a surrogate network with the desired number of nodes and the desired degree, while maintaining the pattern of egocentric interactions of the original dataset.
The results of the experiment on the high school dataset are shown in the middle panel of Figure 5. For each of the two reduced temporal networks we generate a temporal network with 126 nodes to try to reconstruct the original graph. We generate the initial snapshot using the configuration model based on the degree distribution of the first snapshot of the original (not reduced) graph. Then we build local probability distributions only using information from the reduced networks and use these local probability distributions to generate surrogate expanded networks from them. The expanded networks have the same number of nodes of the original one (126), enabling direct comparison. The expedient that we use to augment the mean degree from the reduced seed graph is to increase the parameter $\alpha$ of the generation process, which is the probability to confirm the uni-directional directed links in each provisional layer (set to 1/2 by default). See Methods for the details on how to compute the correct value of $\alpha$ given the original number of links and the desired density of the generated graph.

In the middle panel of Figure 5 the black solid curve represents the number of interactions in the original network, the black dashed curve those in the “train” network with 30% of the nodes and the black dotted curve those in the one with 70% of the nodes. The corresponding values for the generated networks with their standard deviations are reported in orange and brown respectively. Again, we observe the ability of our method to correctly replicate the pattern of interaction in the original network, even if fed with a small percentage of nodes from the original graph as seed.

2.5.3 Temporal extension and size expansion

We can also combine the two techniques above to simultaneously increase the number of nodes and the temporal snapshots. The results are shown in bottom panel of Figure 5 for the high school network, where the synthetic graph has been obtained by only using 50% of the nodes and the first two days of the original dataset (from the beginning to the vertical line), see the black dashed curve. Also in this case, our method is able to extend an input graph in both the temporal and the node size dimensions with remarkable accuracy.
3 Discussion

In this manuscript, we have proposed a model to generate surrogate temporal networks, i.e. synthetic networks that realistically capture the properties of real-world datasets, only making use of the information contained in egocentric temporal neighborhoods. Specifically, we generate temporal networks which accurately reproduce structural characteristics like density, number of interacting individuals, number of interactions in time, number of new conversations, and the possible presence of hubs.

The fidelity of our surrogate networks suggest that the egocentric temporal neighborhoods are fundamental building blocks; building blocks which are sufficient in terms of reconstructing temporal networks, which preserve the essential characteristics of the original graph. In this sense our work illustrates the importance of the egocentric perspective in temporal networks, opening a new direction in generating these networks.

The usefulness of surrogate networks can be evaluated by simulating dynamical systems on them, such as random walks and a SIR model. We observe that in both topological and dynamical tests, the networks generated by our model are generally closer to the original graph than those generated by different literature models. The comparison with competitors mostly highlights the fact that those models tend to neglect fundamental features that our approach is able to preserve. Indeed, even in the few cases where competing methods reproduce a single measure with slightly higher accuracy, they all have at least one measure exhibiting an extreme difference with the original graph (also including basic features like, e.g. the number of nodes).

Moreover, our approach is able to generate temporal network that have different sizes than the original one. This property can be used to increase the number of nodes and extend the network in time, providing a powerful tool for data augmentation.

The real strength of the method, however, is its simplicity. As noted above, this simplicity reveals something about the minimal fundamental building blocks of a temporal network. The same simplicity, moreover, has allowed us to formulate a fast and scalable algorithm, able to first process and then generate very large networks, with high temporal resolution, something which the existing alternatives cannot do.
The other side of the coin is that this simplicity does not capture certain topological features. This is the main limitation of the model. For instance, disregarding second-order interactions translates to a reduced ability to preserving clustering, degree correlations and average shortest path length. Similarly, the absence of long-term memory means that the model currently does not capture recurrent interactions between pairs of nodes. These features are instead well captured by more theoretical models of network generation that include aging, edge reinforcement, or in general some mechanism for memory such that contact durations and inter-event times are heterogeneous and depend on the past interactions. Memory can also be used to generate a synthetic temporal network that is organized in groups, i.e. subsets of nodes highly connected among them and less connected with the other ones. This is a characteristic often occurring in real networks, especially social networks, and it cannot be captured by small local subnetworks like egocentric temporal neighborhoods. However, long-term memory appears in literature only in theoretical models for temporal network generation, for which the goal is to obtain realistic networks by recovering some particular characteristics of the observed dynamics in real networks, but usually do not aim at reconstructing specific real networks or environments. A model which instead is built to obtain surrogate networks with an alternative approach is the one proposed by Presigny et al. This model does not generate a new network from scratch, it instead individuates a backbone of a real temporal network, defined as the global subnetwork composed of the most significant edges, and then reconstructs the missing links. This is based on a conceptually different idea, assuming that the important information concerns the global structure of the network, while the method that we are proposing focuses on how nodes behave given their interactions in last time steps. By recalling two different long-standing traditions in network science, a socio-centric versus an ego-centric perspective, we can assert that if the first one is covered, for what concerns surrogate temporal networks, by the model of Presigny et al., our model places itself in the remaining gap, filling the unexplored case of the ego-centric perspective.

Finally, the sequential nature of the method that we are proposing allows us to easily extend it in many directions. For instance using a preferential attachment in the edge validation step of the procedure. Hence many additional features could be included in future developments of the model. Possible future applications may also include the
possibility to share sensitive data while preserving privacy and also the possibility of merging data from different environments, simply building multiple local probability distributions.

4 Methods

Data description and processing. The three temporal networks studied in the main body of this work represent face-to-face human interactions collected by the SocioPatterns project\(^1\):

- **Hospital**\(^{39}\) The dataset has been collected in the geriatric ward of a university hospital\(^{60}\) in Lyon, France, over four days in December 2010. It contains interactions among medical doctors, paramedical staff, administrative staff and patients. Number of edges: 1139, number of nodes: 75.

- **Workplace**\(^{40}\) The dataset has been collected in 2013 at the *Institut National de Veille Sanitaire*, a health research institute near Paris, over two weeks. It contains interactions among individuals from five departments. Number of edges: 755, number of nodes: 92.

- **High school**\(^{41}\) The dataset has been collected in 2011 in Lycée Thiers, Marseille, France, over four days (Tuesday to Friday). It contains interactions among 118 students and 8 teachers in three different high school classes. Number of edges: 1709, number of nodes: 126.

The Supplementary Information (Section S3) contains scalability experiments on seven more temporal networks (four face-to-face human interactions, two SMS networks and two phone call networks), which were out of reach for (most of) the competitors.

Neighbourhood generation process: parameters. The gap between two consecutive temporal snapshots has been set to 5 minutes for face-to-face interaction networks and 10 minutes for SMS and phone call networks (see Supplementary Information). The

\(^1\)http://www.sociopatterns.org/
time horizon $k$ defining the egocentric temporal neighbourhood has been set to $k = 2$ in all experiments, which is the minimal horizon that preserves some temporal correlation. Local probability models have a granularity of 1 hour and a periodicity of 1 day (i.e., between 8 and 9 am in each day we use the same probability model, and the same holds for all 1 hour slots in the day), for all networks but the ones including weekends, namely Workplace and High school 2, for which the periodicity is set to 1 week.

**Size expansion: preserving interaction density.** The seed graphs for the size expansion experiment are generated by artificially reducing the original dataset (so that the original graph can be used as ground-truth). In this reduction process, whenever a node is dropped all its connections are dropped too. As a consequence, the resulting seed graph has a reduced mean degree with respect to the original one, and the expanded graph generated from it would inherit this reduced mean degree. This problem can be avoided by adjusting the $\alpha$ parameter of the generation process (the probability to confirm the unidirectional links in each provisional layer, set to 1/2 by default). In particular, we would need to set $\alpha = 1 - \frac{1}{2} \frac{\hat{L}}{L}$, where $\hat{L}$ is the average number of links in the seed graph and $L$ the desired number of links in the generated graph. However, $L$ is unknown and needs to be estimated. Something that we know, and that we want in this case to preserve, is the density, defined as $d = \frac{L}{N \cdot (N-1)/2}$ i.e. the fraction between the number of links in the seed graph and all possible links ($\hat{N}$ is the number of nodes in the seed graph). If we assume a linear growth with respect to the number of all possible edges in the network, we also have: $d = \frac{L}{\hat{N} \cdot (\hat{N}-1)/2}$, with $\hat{N}$ as the number of nodes of the generated graph (that we can choose). Combining these two equations we obtain an estimate for $L$, from which we obtain: $\alpha = 1 - \frac{\hat{N} \cdot (\hat{N}-1)}{N \cdot (N-1)/2}$. Hence, when we consider a seed with only 30% of the nodes of the high school dataset (so $N = 126$ and $\hat{N} = 38$) we should use $\alpha = 0.96$ to reproduce the same density. While if we start with 50% and 70% of the nodes (i.e. $\hat{N} = 63$ and $\hat{N} = 88$) in the seed we should use respectively $\alpha = 0.88$ and 0.76.

**Alternatives approaches for generating networks.** Dymond34 builds a temporal network considering (i) the dynamics of temporal motifs in the graph and (ii) the roles nodes play in motifs (e.g., in a wedge – two links connecting three nodes – one node plays the hub, while the remaining two act as spokes). The method has no parameters
to be set. Structural Temporal Modeling (STM) extracts counts for a predefined library of (non-egocentric) temporal motifs from the original network, and turns them into generation probabilities from which to create the temporal network. This method has no tunable parameters. TagGen is a neural-network based approach that extracts temporal random walks from the original graph and feeds them to an assembling module for generating temporal networks. TagGen has been trained with the parameters used in the original paper, namely 30 epochs with a batch size of 64 and stochastic gradient descent with a learning rate of 0.001.

5 Code and data availability

The data used to support this study are publicly available.

- The SocioPatterns data at http://www.sociopatterns.org
- The CNS data at https://doi.org/10.6084/m9.figshare.7267433
- The Friends and Family data at http://realitycommons.media.mit.edu/friendsdataset.html

The code used for the generation of temporal network is publicly available at

- https://github.com/AntonioLonga/ETNgen

References


S1 Metrics details

In the main text we use ten different topological metrics and three dynamical metrics to compare the original graphs with the synthetic ones obtained from them. The topological metrics can be divided into eight global metrics, which are computed for each temporal layer as it was a static network, and for which we report distributions over temporal layers:

- **Number of interactions.** The number of edges.

- **Density.** The ratio of edges in the graph versus the number of edges if it was a complete graph.\(^{[34]}\)

- **Interacting individuals.** The number of individuals that are interacting.\(^{[11]}\)

- **New conversations.** The number of conversations starting at this specific timestamp.\(^{[11]}\)

- **S-metric.** A measure of the extent to which a graph has a hub-like core, maximized when high-degree nodes are connected to other high-degree nodes.\(^{[44]}\)

- **Global clustering coefficient.** The ratio of the number of closed triplets to the total number of open and closed triplets.\(^{[45,46]}\)

- **Assortativity.** The degree-degree correlation of nodes that are connected.\(^{[47]}\)
• **Average shortest path length.** The average shortest path length for all possible pairs of nodes of the largest connected component for each temporal layer.

And two local metrics (distributions over edges):

• **Duration of contacts.** The mean duration (in timestamps) of interactions between each couple of nodes.

• **Edge strength in the projected weighted network.** The total number of interactions in time between each couple of nodes.

The dynamical metrics are obtained starting from two dynamical processes, a random walk and a spreading process. For random walk we use:

• **Coverage.** The number of (distinct) visited nodes starting from a random node at an initial timestamp. The simulation is repeated 1000 times using a random initial node and the initial time is set equal to the first timestamp.

• **Mean First Passage Time (MFPT).** The average time taken by the random walker to arrive for the first time at a specific node $i$, starting from a random initial position $j$ in the network. We consider each couple of nodes $(i, j)$ in the network and repeat the simulation five times for each of them.

The spreading process is a SIR model and we compute the following metric:

• **Reproduction value $R_0$.** The average number of individuals infected by the first one, with a single random node infected as seed.

### S2 Execution time comparison

The egocentric perspective, that ignores interactions among neighbors of each ego node, implies a huge simplification with respect to mining standard motifs. Traditional techniques for motifs mining indeed rely on an isomorphism test for assessing sub-network
equivalence, which is a major bottleneck for the entire procedure. For this reason, standard motifs mining techniques usually limit the search to small motifs containing a handful of nodes. The strength of ETN-gen lies in the possibility of encoding neighborhoods into a unique bit vector, boiling down sub-network equivalence to bit vector matching. This hence results in a very computationally efficient model, and the time required for network generation is drastically lower than that of the competitors. This is evident from table S1, where we report the time (in seconds) required to generate networks for the three face-to-face datasets with our algorithm and the competitors. ETN-gen is more than 15 times faster than the fastest competitor on each network, and there is a difference in time of three orders of magnitude with the slowest one.

<table>
<thead>
<tr>
<th></th>
<th>Hospital</th>
<th>Workplace</th>
<th>High School</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ETN-gen</strong></td>
<td>17s</td>
<td>52s</td>
<td>22s</td>
</tr>
<tr>
<td><strong>Dymond</strong></td>
<td>$3.6 \times 10^4$s</td>
<td>$1.4 \times 10^3$s</td>
<td>$3.2 \times 10^5$s</td>
</tr>
<tr>
<td><strong>STM</strong></td>
<td>$1.4 \times 10^3$s</td>
<td>$9.6 \times 10^2$s</td>
<td>$1.6 \times 10^3$s</td>
</tr>
<tr>
<td><strong>TagGen</strong></td>
<td>$2.7 \times 10^4$s</td>
<td>$8.7 \times 10^3$s</td>
<td>$2.4 \times 10^4$s</td>
</tr>
</tbody>
</table>

Table S1: Execution time. Time in seconds required to train and generate networks with each method on three different networks.

S3 Scalability

To show the scalability of our approach we extend the analysis to other seven networks, briefly described below.

- **High school 2** The dataset has been collected in 2012 in Lycée Thiers, Marseille, France, over seven days (Monday to Tuesday of the following week). It contains interactions among students in five different high school classes. Number of edges: 2220, number of nodes: 180.

- **High school 3** The dataset has been collected in 2013 in Lycée Thiers, Marseille, France, over five days in December. It contains interactions among students in nine different high school classes. Number of edges: 5818, number of nodes: 327.
• **Primary school**\[62\] The dataset has been collected in a primary school in France, over two days in October 2009. It contains interactions among 232 children and 10 teachers. Number of edges: 8317, number of nodes: 242.

• **SMS 1**\[17\] The dataset represents SMSs among university freshmen students in the Copenhagen University. Number of edges: 697, number of nodes: 568.

• **SMS 2**\[16\] The dataset represents SMSs among members of a young-family residential living community adjacent to a major research university in North America. Number of edges: 153, number of nodes: 85.

• **Calls 1**\[17\] The dataset represents phone calls among university freshmen students in the Copenhagen University. Number of edges: 605, number of nodes: 525.

• **Calls 2**\[16\] The dataset represents phone calls among members of a young-family residential living community adjacent to a major research university in North America. Number of edges: 432, number of nodes: 129.

Each face-to-face interaction network has been aggregated with a temporal resolution of five minutes, while SMS and phone calls networks have been aggregated within ten minutes. We opt for this different aggregations due to the natural sparsity of SMS and phone calls networks.

In Figure\[S1\] we show the original number of interactions (in black) and those generated by our method (in orange) for each network. The figure clearly shows the ability of our method in mimicking day/night and week/weekend periodicity. Moreover, our algorithm perfectly operates with different network sizes in both number of individuals and temporal length. Finally, our method is able to capture multiple picks within the same day, that could be associated to the period before and after lunch (i.e. high schools).

### S3.1 Topological similarity on additional datasets

In this section we show the effectiveness of *ETN-gen* when using alternative temporal networks as original graphs. We test some additional face-to-face interactions networks, then we consider remote communication interactions like SMS and calls networks, for
Figure S1: Number of interactions in the generated network for different datasets. Each panel shows the number of interactions of the original (black curve) and ETN-gen (orange curve) graphs. We use a temporal gap of 5 minutes for face-to-face interactions and 10 minutes for calls and SMS (intrinsically sparser networks).

a total of 10 different temporal datasets. We compare ETN-gen results with TagGen as a sole competitor. This choice is mainly due to time constraints: the other competitors would require a too long time to complete the generation process when dealing with larger datasets (see Section S2). Moreover, TagGen is the only algorithm able to always reproduce the same exact number of nodes of the input network, and the only competitor able to capture intrinsic periodicity of the network.
S3.1.1 Face-to-face interactions networks

In the main text we show how to generate surrogate networks that mimic three face-to-face interaction datasets, by making use of ETN-gen and three other alternative methods. Here, we focus on three additional face-to-face interaction datasets, namely High school 2, High school 3 and Primary school. Kolmogorov-Smirnov distances of the metrics described in the main text (see Methods) are shown in Figure S2.

Figure S2: Kolmogorov-Smirnov distances applied to 10 different topological metric distributions of face to face interaction networks.
S3.1.2 SMS and phone calls networks

Figure S3 shows the distances among original and generated distributions of the chosen topological metrics. Again, our method performs generally better than the competitor. As expected, the methodology we propose is not able to capture metrics correlated to long-term memory.

Figure S3: Kolmogorov-Smirnov distances applied to 10 different topological metric distributions of SMS and phone calls networks.
S3.2 Dynamical similarity on additional datasets

Here we report Kolmogorov-Smirnov distances between original and generated networks in terms of random walks (Coverage and Mean First Passage Time) and the $R_0$ over a SIR model.

S3.2.1 Face-to-face interaction networks

Figure S4 shows Kolmogorov-Smirnov distances between dynamic distributions of original and generated networks. Our neighbourhood approach is able to capture the coverage of random walks better than TagGen, while TagGen networks appear more similar to the original ones for what concerns the mean first passage time. The similarity of the spreading model, here represented by $R_0$, is slightly better captured by our method.

![Figure S4](image)

Figure S4: Kolmogorov-Smirnov distances applied to 10 different dynamic metric distributions of face-to-face interaction networks.

S3.2.2 SMS and phone calls networks

Figure S5 shows the ability of our model to capture the similarity of $R_0$s distributions in all networks. In terms of random walk, our neighbourhood approach again shows a greater similarity for coverage and a lower similarity for mean first passage time. This phenomenon is due to the huge sparsity of phone calls and SMS networks.
Figure S5: Kolmogorov-Smirnov distances applied to 10 different dynamic metric distributions of SMS and phone calls networks.