Using Causality-Aware Graph Neural Networks to Predict Temporal Centralities in Dynamic Graphs

Franziska Heeg
Machine Learning for Complex Networks
Center for AI and Data Science (CAIDAS)
Julius-Maximilians-Universität Würzburg
franziska.heeg@uni-wuerzburg.de

Ingo Scholtes*
Machine Learning for Complex Networks
Center for AI and Data Science (CAIDAS)
Julius-Maximilians-Universität Würzburg
ingo.scholtes@uni-wuerzburg.de

Abstract

Node centralities play a pivotal role in network science, social network analysis, and recommender systems. In temporal data, static path-based centralities like closeness or betweenness can give misleading results about the true importance of nodes in a temporal graph. To address this issue, temporal generalizations of betweenness and closeness have been defined that are based on the shortest time-respecting paths between pairs of nodes. However, a major issue of those generalizations is that the calculation of such paths is computationally expensive.

Addressing this issue, we study the application of De Bruijn Graph Neural Networks (DBGNN), a causality-aware graph neural network architecture, to predict temporal path-based centralities in time series data. We experimentally evaluate our approach in 13 temporal graphs from biological and social systems and show that it considerably improves the prediction of both betweenness and closeness centrality compared to a static Graph Convolutional Neural Network.

1 Motivation

Node centralities are important in the analysis of complex networks, with applications in network science, social network analysis, and recommender systems. An important class of centrality measures are *path-based centralities* like, e.g. betweenness or closeness centrality [1, 2], which are based on the shortest paths between all nodes. While centralities in static networks are important, we increasingly have access to time series data on temporal graphs with time-stamped edges. Due to the timing and ordering of those edges, the paths in a static time-aggregated representation of such time series data can considerably differ from *time-respecting paths* in the corresponding temporal graph. In a nutshell, two time-stamped edges $(u, v; t)$ and $(v, w; t')$ only form a time-respecting path from node u via v to w iff for the time stamps t and t' we have $t < t'$, i.e. time-respecting paths must minimally respect the arrow of time. Moreover, we often consider scenarios where we need to additionally account for a *maximum time difference* δ between time-stamped edges, i.e. we require $0 < t' - t \leq \delta$ [3]. Several works have shown that temporal correlations in the sequence of time-stamped edges can significantly change the *causal* topology of a temporal graph, i.e. which nodes can influence each other via time-respecting paths, compared to what is expected based on the static topology [4, 5, 6].

An important consequence of this is that static path-based centralities like closeness or betweenness can give misleading results about the true importance of nodes in temporal graphs. To address this issue, temporal generalizations of betweenness and closeness centrality have been defined that are based on the shortest time-respecting paths between pairs of nodes [7, 8, 9, 10]. A major issue of those generalizations is that the calculation of time-respecting paths as well as the resulting centralities is computationally expensive [11, 12, 13]. Addressing this issue, a number of recent

*also with Data Analytics Group, Department of Informatics (IfI), University of Zurich, Switzerland

Temporal Graph Learning Workshop @ NeurIPS 2023, New Orleans.

works developed methods to approximate temporal betweenness and closeness centralities in temporal graphs [13]. Additionally, few works have used deep (representation) learning techniques to predict computationally expensive path-based centralities in *static* networks [14, 15].

Research Gap and Contributions To the best of our knowledge, no prior works have considered the application of time-aware graph neural networks to predict path-based centralities in temporal graphs. Closing this gap, our work makes the following contributions:

- We introduce the problem of predicting temporal betweenness and closeness centralities of nodes in temporal graphs. We consider a situation where we have access to a training graph as well as ground truth temporal centralities and seek to predict the centralities of nodes in a future observation of the same system, which does not necessarily consist of the exact same set of nodes.
- To address this problem, we introduce a deep learning method that utilizes De Bruijn Graph Neural Networks (DBGNN), a recently proposed causality-aware graph neural network architecture [16] that is based on higher-order graph models of time-respecting paths, which capture correlations in the sequence of time-stamped edges. An overview of our approach in a toy example of a temporal graph is shown in Figure 1.
- We compare our proposed method to a Graph Convolutional Network (GCN), which only considers a static, time-aggregated weighted graph that captures the frequency and topology of edges.
- We experimentally evaluate both models in 13 time temporal graphs from biological and social systems. Our results show that the application of the time-aware DBGNN architecture considerable improves the prediction of both betweenness and closeness centrality compared to a static GCN model.

In summary, we show that predicting temporal centralities is an interesting temporal graph learning problem, which could be included in community benchmarks [17]. Moreover, our study highlights the potential of causality-aware deep learning architectures for node-level regression tasks in temporal graphs. Finally, our results are a promising step towards an approximation of temporal centralities in large data, with potential applications in social network analysis and recommender systems.

2 Background and Related Work

In the following, we provide the background of our work. We first introduce temporal graphs and define time-respecting (or causal) paths. We then cover generalizations of path-based centralities for nodes in temporal graphs. We finally discuss prior works that have studied the prediction, or approximation, of path-based centralities both in static and temporal graphs. This will motivate the research gap that is addressed by our work.

Dynamic Graphs and Causal Paths Apart from static graphs $G = (V, E)$ that capture the topology of edges $E \subseteq V \times V$ between nodes V, we increasingly have access to time-stamped interactions that can be modelled as *temporal graphs or networks* [18, 19, 3]. We define a temporal graph as $G^{\mathcal{T}} = (V, E^{\mathcal{T}})$ where V is the set of nodes and $E^{\mathcal{T}} \in V \times V \times \mathbb{R}$ is a set of (possibly directed) time-stamped edges, i.e. an edge $(v, w; t) \in E^{\mathcal{T}}$ describes an interaction fron node v to w occurring at time t. In our work, we assume that interactions are *instantaneous*, i.e. $(v, w; t) \in E^{\mathcal{T}}$ does not imply that $(v, w; t') \in E^{\mathcal{T}}$ for all $t' > t$. Hence, we do not specifically consider *growing networks*, where the time-stamp t is the creation of an edge. For a temporal network $G^{\mathcal{T}} = (V, E^{\mathcal{T}})$ it is common to consider a static, time-aggregated and weighted graph representation $G = (V, E)$, where $(v, w) \in E$ iff $(v, w; t) \in E^{\mathcal{T}}$ for some time stamp t and for the edge weights we define $w(v, w) = |\{t \in \mathbb{R} : (v, w; t) \in E^{\mathcal{T}}\}|$, i.e. the number of occurrences of time-stamped edges.

An important difference to paths in static graphs is that, in temporal networks, the temporal ordering of edges determines what we call *time-respecting or causal paths* [20, 19, 3]. For a temporal graph $G^{\mathcal{T}} = (V, E^{\mathcal{T}})$ we define a *time-respecting or causal path* of length l as sequence of nodes $v_0, \ldots, v_l$ such that the following two conditions hold:

(i) $\exists\, t_1, \ldots, t_l \,:\, (v_{i-1}, v_i; t_i) \in E^{\mathcal{T}}$ for $i = 1, \ldots, l$;

(ii) $0 < t_i - t_{i-1} \leq \delta$ for some $\delta \in \mathbb{R}$.

In contrast to definitions of time-respecting paths that only require interactions to occur in ascending temporal order, i.e. $0 < t_i - t_j$ for $j < i$ [20, 21], we additionally impose a maximum "waiting time"

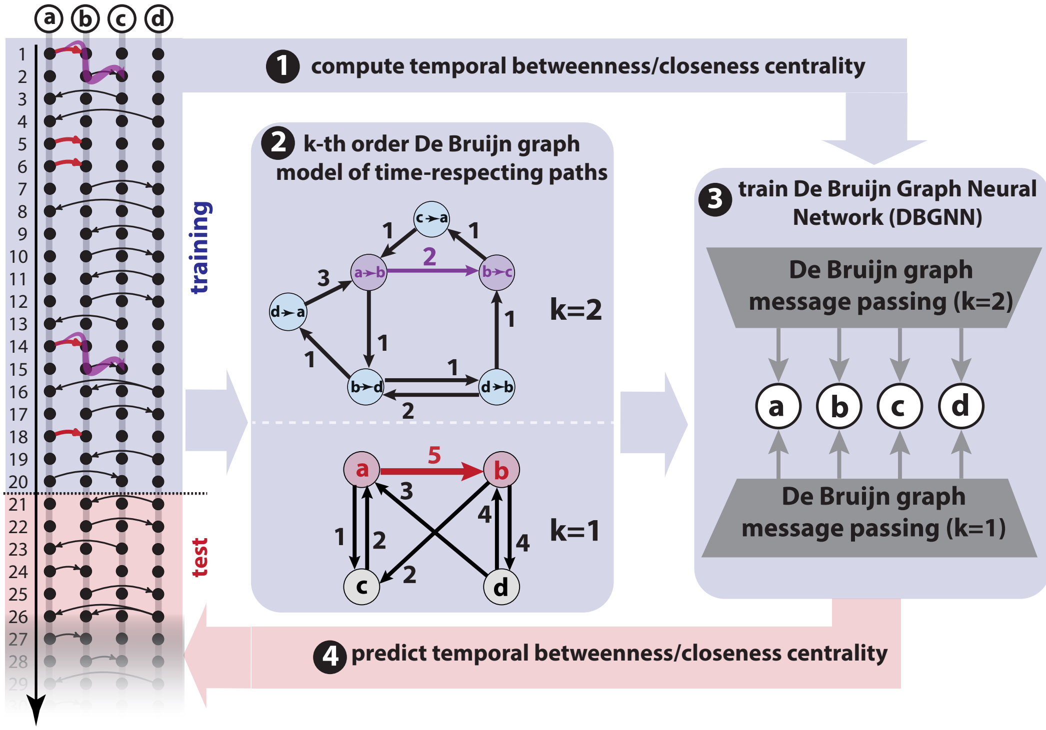


Figure 1: Overview of proposed approach to predict temporal centralities of nodes in a temporal graph: We consider a time-based split in a training and test graph (left). Calculating time-respecting paths in the training split enables us to (1) compute temporal node centralities, and (2) fit a k-th order De Bruijn graph model for time-respecting paths. The weighted edges in such a k-th order De Bruin graph capture the frequencies of time-respecting paths of length k (see time-respecting path of length one (red) and two (magenta)). (3) We then use these centralities and the k-th order models to train a De Bruijn graph neural network (DBGNN), which allows us to (4) predict temporal node centralities in the test graph.

δ [5, 19]. This implies that we only consider time-respecting paths where subsequent interactions occur within a time interval that is often defined by the processes that we study on temporal networks [12, 22]. In line with the definition for static networks, we define a *shortest time-respecting path* between two nodes v and w as a (not necessarily unique) time-respecting path of length l such that all other time-respecting paths from v to w have length $l' \geq l$. In static graphs a shortest path from v to w is necessarily a *simple* path, i.e. a path where no node occurs more than once in the sequence $v_1, \ldots, v_l$. This is not neccessarily true for shortest time-respecting path, since –due to the maximum waiting time δ– we may be forced to move between (possibly the same) nodes to continue a time-respecting path. Due to the definition of time-respecting paths with limited waiting time δ, we obtain a *temporal-topological* generalization of shortest paths to temporal graphs that accounts for the temporal ordering and timing of interactions. We note that there exist definitions of *fastest paths* that only account for the temporal rather than the topological distance [5], which we however do not consider in our work.

The definition of time-respecting paths above has the important consequence that the connectivity of nodes via time-respecting paths in a temporal network can be considerably different from paths in the corresponding time-aggregated static network. As an example, for a temporal network with two time-stamped edges $(u, v; t)$ and $(v, w; t')$ the time-aggregated network contains a path from u via v to w, while a time-respecting path from u via v to w can only exist iff $0 < t' - t \leq \delta$. In other words, while connectivity in static graphs is *transitive*, i.e. the existence of edges (or paths) connecting u to v and v to w implies that there exists a path that transitively connects u to w, the same does not hold for time-respecting paths. A large number of works have shown that this difference between paths in temporal and static graphs influences connectivity and reachability [4], the evolution of dynamical processes like diffusion or epidemic spreading [23, 6, 24, 25], cluster patterns [23, 26, 27], as well as the controllability of dynamical processes [28].

Temporal Centralities Another interesting question is how the time dimension of temporal graphs influences the importance or *centrality* of nodes [8]. To this end, several works have generalized centrality measures originally defined for static graphs to temporal networks. For our purpose we limit ourselves to generalizations of betweenness and closeness centrality, which are defined based on the shortest paths between nodes. In a static network, a node v has high *betweenness centrality* if there are many shortest paths that pass through v [2] and it has high *closeness centrality* if the overall distance to all other nodes is small [1]. We omit those standard definitions here due to space constraints but include them in appendix C.

Analogously to betweenness centrality for static graphs, for a temporal graph $G = (V, E^T)$ we define the *temporal betweenness centrality* of node $v \in V$ as

$$c_B^{temp}(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$$

where $\sigma_{s,t}$ is the number of shortest *time-respecting* paths from node s to t. Following our definition above, we consider two time-respecting paths to be the same if the sequence of traversed nodes is identical, i.e. $\sigma_{s,t}$ counts paths that traverse the same set of nodes at different times only once.

To calculate the *temporal closeness centrality* we define the temporal distance $d(u, v)$ between two nodes $u, v \in V$ as the length of a shortest time-respecting paths from u to v and thus obtain

$$c_C^{temp}(v) = \frac{1}{\sum_{u \in V} d(u, v)}.$$

Even though the definitions above closely follow those for static networks, it has been shown that the temporal centralities of nodes can differ considerably from their counterparts in static time-aggregated networks [8, 27]. These findings highlight the importance of a *time-aware* network analysis, which consider both the timing and temporal ordering of links in temporal graphs.

Approximating Path-based Centralities While path-based centralities have become an important tool in network analysis, a major issue is the computational complexity of the underlying all-pairs shortest path calculation in large graphs. For static networks, this issue can be partially alleviated by smart algorithms that speed up the calculation of betweenness centralities [29]. Even with these algorithms, calculating path-based centralities in large graphs is a challenge. Hence, a number of works considered approaches to calculate fast approximations, e.g. based on a random sampling of paths [30, 31, 32]. Another line of studies either used standard, i.e. not graph-based, machine learning techniques to leverage correlations between different centrality scores [15, 14], or used neural graph embeddings in synthetic scale-free networks to approximate the ranking of nodes [33].

Existing works on the approximation of path-based node centralities in time series data have generally focussed on a fast updating of *static* centralities in *evolving graphs* where edges are added or deleted [34, 35], rather than considering *temporal node centralities*. For the calculation of temporal closeness or betweenness centralities, the need to calculate shortest *time-respecting paths* between all pairs of nodes is a major computational challenge. In particular, the calculation of time-respecting paths with a maximum waiting time constraint, which is the definition considered in our work, has recently been shown to be an NP-hard problem [12]. Considering the approximate estimation of temporal betweenness and closeness centrality in temporal graphs, [36] generalizes static centralities to higher-order De Bruijn graphs, which capture the time-respecting path structure of a temporal graph. [13] recently proposed a sampling-based estimation of temporal betweenness centralities. To the best of our knowledge, no prior works have considered the application of deep graph learning to predict temporal node centralities in temporal graphs, which is the gap addressed by our work.

3 A Causality-Aware GNN Architecture to Predict Temporal Centralities

In the following, we first present higher-order De Bruijn graph models for time-respecting paths in temporal networks. We then describe our deep learning architecture to predict temporal node centralities.

Higher-Order De Bruijn Graph Models of Time-respecting paths Each time-respecting path gives rise to an ordered sequence $v_0, v_1, \ldots, v_l$ of traversed nodes. Let us consider a k-th order Markov chain model, where $P(v_i | v_{i-k}, \ldots, v_{i-1})$ is the probability that a time-respecting path continues to node v_i, conditional on the k previously traversed nodes. A first-order Markov chain model can be defined based on the frequencies of edges (i.e. paths of length $k = 1$) captured in a weighted time-aggregated graph, where

$$P(v_i|v_{i-1}) := \frac{w(v_{i-1}, v_i)}{\sum_j w(v_{i-1}, v_j)}.$$

While such a first-order model is justified if the temporal graph exhibits no patterns in the temporal ordering of time-stamped edges, a number of works have shown that empirical data exhibit patterns that require higher-order Markov models for time-respecting paths [23, 25, 26]. To address this issue, for $k > 1$ we can define a k-th order Markov chain model based on the frequencies of time-respecting paths of length k as

$$P(v_i|v_{i-k}, \ldots, v_{i-1}) = \frac{w(v_{i-k}, \ldots, v_i)}{\sum_j w(v_{i-k}, \ldots, v_{i-1}, v_j)},$$

where $w(v_0, \ldots, v_k)$ counts the number of time-respecting path $v_0, \ldots, v_k$ in the underlying temporal graph. For a temporal graph $G^{\mathcal{T}} = (V, E^{\mathcal{T}})$, this approach defines a static *k-th order De Bruijn graph model* $G^{(k)} = (V^{(k)}, E^{(k)})$ with

- $V^{(k)} = \{(v_0, \ldots, v_{k-1}) \mid v_0, \ldots, v_{k-1}$ is a causal walk of length $k-1$ in $G^{\mathcal{T}}\}$
- $(u, v) \in E^{(k)}$ iff
  - (i) $v = (v_1, \ldots, v_k)$ with $v_i = u_i$ for $i = 1, \ldots, k-1$
  - (ii) $u \bigoplus v = (u_0, \ldots, u_{k-1}, v_k)$ is a causal path of length k in $G^{\mathcal{T}}$.

We call this k-th order model a *De Bruijn graph model* of time-respecting paths, since it is a generalization of a k-dimensional De Bruijn graph [37], with the additional constraint that an edge only exists iff the underlying temporal network has a corresponding time-respecting path. For $k = 1$ the first-order De Bruijn graph corresponds to the commonly used static, time-aggregated graph $G = (V, E)$ of a temporal graph G^T, where edge can be considered time-respecting paths of length one and which neglects information on time dimension. For $k > 1$ we obtain *static but time-aware higher-order generalizations of time-aggregated graphs*, which are sensitive to the timing and ordering of time-stamped edges. Each node in such a k-th order De Bruin graph represents a time-respecting path of length $k - 1$, while edges represent time-respecting paths of length k. Edge weights correspond to the number of observations of time-respecting paths of length k (cf. fig. 1).

De Bruijn Graph Neural Networks for Temporal Centrality Prediction Our approach to predict temporal betweenness and closeness centrality uses the recently proposed De Bruijn Graph Neural Networks (DBGNN), a deep learning architecture that builds on k-th order De Bruijn graphs [16]. The intuition behind this approach is that, by using message passing in multiple (static) k-th order De Bruijn graph models of time-respecting paths, we obtain a *causality-aware learning algorithm* that considers both the graph topology as well as the temporal ordering and timing of interactions.

Our proposed method is summarized in fig. 1. Considering time series data on a temporal graph, we first perform a time-based split of the data into a training and test graph. We then calculate temporal closeness and betweenness centralities of nodes in the training graph and consider a supervised node-level regression problem, i.e. we use temporal centralities of nodes in the training graph to train a DBGNN model. To this end, we construct k-th order De Bruijn graph models for multiple orders k, based on the statistics of time-respecting paths of lengths k. The maximum order is determined by the temporal correlation length (i.e. the Markov order) present in a temporal graph and can be determined by statistical model selection techniques [38].

Using the update rule defined in Eq. (1) of [16], we simultaneously perform message passing in all k-th order De Bruijn graphs. For each k-th order De Bruijn graph this yields a (hidden) representation of k-th order nodes. To aggregate the resulting representation to actual (first-order) nodes in the temporal graph, we perform message passing in an additional bipartite graph, where each k-th order node $(v_0, \ldots, v_{k-1})$ is connected to first-order node v_{k-1} (cf. Eq (2) in [16] and fig. 1). Taking a node regression perspective, we use a final dense linear layer with a single output. We use the trained model to predict the temporal centralities of nodes in the test graph. Since the subset of nodes and edges that are active in the training and test graph can differ, our model must be able to generalize to temporal graphs with different nodes as well as different graph topologies. To address this, we train our models in an inductive fashion by choosing a suitably large number of dimensions for the one-hot encodings during the training phase. Our code is publicly available at `https://doi.org/10.5281/zenodo.10202792`.

4 Experimental Results

With our experimental evaluation we seek to answer the following research question:

RQ1 How does the predictive power of a causality-aware DBGNN model compare to that of a standard GCN that ignores the temporal dimension of dynamic graphs?
RQ2 How does the predictive power differ between temporal betweenness and closeness centrality and how does it vary across different temporal networks?
RQ3 How does the computational efficiency of the DBGNN-based prediction of temporal node centralities compare to that of a time-neglecting GCN architecture?
RQ4 Does the DBGNN architecture generate node embeddings that facilitate interpretability?

Experimental setup We experimentally evaluate the performance of the DBGNN architecture by predicting temporal centralities in 13 empirical temporal graphs. Since a maximum order detection in those data sets yields a maximum of two (see table 7 in appendix A), we limit the DBGNN architecture to $k = 2$. To calculate edge weights of the DBGNN model, we count time-stamped edges for weights of the first-order De Bruijn graph as well as time-respecting paths of length two for weights of the second-order De Bruijn graph (cf. fig. 1). Adopting the approach in [16] we use one message passing layer with 16 hidden dimensions for each order k and one additional bipartite message passing layer with 8 hidden dimensions. We use a sigmoid activation function for the higher-order layers and an Exponential Linear Unit (ELU) activation function for the bipartite layer.

As a baseline model, we use a Graph Convolutional Neural Network (GCN) [39], which we apply to the weighted time-aggregated representation of the temporal graphs. For the GCN model, we use two message passing layers with 16 and 8 hidden dimensions and a sigmoid activation function, respectively. As input features, we use a one-hot encoding (OHE) of nodes for both architectures. In the case of the DBGNN architecture we apply OHE to nodes in all (higher-order) layers. Addressing a node regression task, we use a final dense linear layer with a single output and an ELU activation function, and use mean squared error (MSE) as loss function for both architectures. We train both models based on the (ground-truth) temporal node centralities in the training data, using 5000 epochs with an ADAM optimizer, different learning rates, and weight decay of $5 \cdot 10^{-4}$. We additionally tested the use of dropout layers for both architectures, but found the results to be worse. In table 10 and table 11 we summarize the architecture and the hyperparameters for both models.

Data sets We use 13 data sets on temporal graphs from different contexts, including human contact patterns based on (undirected) proximity or face-to-face relations, time-stamped (directed) E-Mail communication networks, as well as antenna interactions between ants in a colony. An overview of the data sets along with a short description, key characteristics and the source is given in table 1. All data sets are publicly available from the online data repositories netzschleuder [40] and SNAP [41].

Table 1: Overview of time series data sets used in the experiment evaluation

data set	Description	Ref	Nodes	Edges	Temporal Edges	Directed	δ
ants-1-1	Ant Antenna interactions, colony 1 - filming 1	[42]	89	947	1,911	True	30 sec
ants-1-2	Ant Antenna interactions, colony 1 - filming 2	[42]	72	862	1,820	True	30 sec
ants-2-1	Ant Antenna interactions, colony 2 - filming 1	[42]	71	636	975	True	30 sec
ants-2-2	Ant Antenna interactions, colony 2 - filming 2	[42]	69	769	1,917	True	30 sec
company-emails	E-Mail exchanges in manufacturing company	[43]	167	5,784	82,927	True	60 mins
eu-email-2	E-Mail exchanges in EU institution (dept 2)	[44]	162	1,772	46,772	True	60 mins
eu-email-3	E-Mail exchanges in EU institution (dept 3)	[44]	89	1,506	12,216	True	60 mins
eu-email-4	E-Mail exchanges in EU institution (dept 4)	[44]	142	1,375	48,141	True	60 mins
sp-hospital	Face-to-face interactions in a hospital	[24]	75	1,139	32,424	False	60 mins
sp-hypertext	Face-to-face interactions at conference	[45]	113	2,498	20,818	False	60 mins
sp-workplace	Face-to-face interactions in a workspace	[46]	92	755	9,827	False	60 mins
sp-highschool	Face-to-face interactions in a highschool	[47]	327	5,818	188,508	False	60 mins
haggle	Human proximity recorded by smart devices	[48]	274	2,899	28,244	False	1 min

Evaluation procedure To evaluate our models, we first fit the pre-trained models to the test graph, i.e. we apply the trained GCN model to the weighted time-aggregated test graph and the trained DBGNN model to the De Bruijn graphs for the test data. We then use the trained models to predict temporal closeness and betweenness centralities and compare those predictions to ground truth centralities, which we obtain by exhaustively calculating all time-respecting paths in the test data. Figure 1 provides an illustration of our evaluation approach. We calculate the mean absolute error between predicted and ground truth centralities. We further use Kendall-Tau and Spearman rank correlation to compare a node ranking based on predicted centralities with a ranking obtained from ground truth centralities. We evaluate predictions in terms of mean absolute error as well as Spearman and Kendall-Tau rank correlation coefficient. Since centrality scores are often used to identify a small set of most central nodes, we further calculate the number of hits in the set of nodes with the top ten predicted centralities. Since we repeated each experiment 30 times, we report the mean and the standard deviation of all scores. We further repeated all experiments for three different learning rates between 0.1 and 0.001 and only report the best mean scores for each model individually.

Discussion of results The results of our experiments for temporal betweenness and closeness centralities are shown in table 2 and table 3, respectively. Considering **RQ1**, we find that our causality-aware DBGNN-based architecture significantly outperforms a standard GCN model for all 13 data sets and for all evaluation metrics in the case of temporal closeness centrality (with the exception of the MAE score in sp-hypertext, where we observe no significant difference). We further observe a large relative increase of the Spearman rank correlation coefficient averaging to 117 % across all data, ranging between 32 % for sp-hypertext and 436 % for eu-email-4. For temporal betweenness centrality, we find that the proposed causality-aware architecture significantly outperforms a GCN-based prediction in terms of Spearman and Kendall-Tau rank correlation for seven of the 13 data sets, while we observe no significant difference for five and better performance of the GCN model for a single data set. On average across all 13 data sets, the DBGNN architecture yields an increase in Spearman correlation by 39 %. For the seven cases where DBGNN outperforms GCN, we find relative increases in Spearman rank correlation between 18 and 97 %. For sp-highschool, where a GCN model outperforms a DBGNN-based prediction, the relative increase is 17 %.

Regarding **RQ2** we observe that the performance of a time-neglecting GCN-based prediction of temporal closeness and betweenness centrality are comparable. On the contrary, the time-aware DBGNN model performs significantly better for temporal closeness compared to temporal betweenness. Moreover, we find a large variation of predictive performance across different data sets. To further investigate the differences between data sets, in Figure 2 and Figure 3 in appendix C we plot the temporal and static closeness and betweenness centralities of nodes. The results indicate that the timing and temporal ordering of interactions translates to larger differences between temporal and static betweenness centrality, compared to closeness centrality, which could explain our observation.

A potential criticism of our approach could be that the size of a higher-order De Bruijn graph model can be considerably larger than for a standard first-order graph, thus possibly making them computationally expensive. To address this issue, and to answer **RQ3**, we investigate the scalability of our approach for all of the 13 data sets. For this, we calculate both the training and the inference time of our DBGNN-based architecture with those of a simpler GCN model. The results in table 5 and table 6 in appendix A show that the computational requirements of the DBGNN and GCN model are comparable, both during training and the inference time. We attribute this to the fact that the DBGNN architecture utilizes a compact, *static but time-aware* De Bruijn graph representation of potentially large time series, rather than requiring a representation of all time-stamped edges.

A potential advantage of our method is that it can facilitate fast approximate predictions of temporal centralities. The exact calculation of those centralities is computationally expensive as it requires to exhaustively calculate shortest time-respecting paths (with a given maximum time difference) between all pairs of nodes [49]. Highlighting this, in table 4 in the appendix we report the time required to calculate (ground truth) temporal closeness and betweenness centralities based on shortest time-respecting paths between all pairs of nodes in the test networks. Importantly, while our approach requires to fit a k-th order De Bruijn graph model in the test set, this procedure only requires to calculate time-respecting paths of exactly length k, which is a much simpler problem.

Considering **RQ4**, another aspect of our approach to use a time-aware but *static* graph neural network is that the hidden layer activations yield *static* embeddings that are based on the *causal topology* of a dynamic graph. This causal topology is influenced by (i) the topology of time-stamped links, and (ii) their timing and temporal ordering. To explain the favorable performance of our model, we hypothesize that nodes for which our model learns similar embeddings also have more similar temporal centralities, compared to the embeddings generated by a GCN model. To test this hypothesis, we apply a dimensionality reduction to the node activations generated by the last 8-dimensional bipartite layer in the DBGNN architecture, comparing it to the representation obtained from the last message passing layer of a GCN model. In appendix D we show the resulting embeddings for one representative prediction of temporal closeness using the DBGNN (left) and the GCN model (right) in the eu-email-4 data, where an additional color gradient highlights ground truth closeness centralities of nodes in the test data. The resulting plot clearly shows that the time-aware DBGNN architecture is able to capture the ranking of nodes, while the time-neglecting GCN model is not.

Table 2: Results for prediction of temporal betweenness centrality

Experiment	DBGNN MAE	DBGNN Spearmanr	DBGNN Kendalltau	DBGNN hitsIn10	GCN MAE	GCN Spearmanr	GCN Kendalltau	GCN hitsIn10
ants-1-1	**202.743** ± 0.435	**0.748** ± 0.019	**0.571** ± 0.018	**6.633** ± 0.964	207.561 ± 2.603	0.500 ± 0.145	0.357 ± 0.105	2.900 ± 1.197
ants-1-2	**8.892** ± 0.37	**0.830** ± 0.021	**0.650** ± 0.023	**6.767** ± 0.568	14.558 ± 1.373	0.421 ± 0.165	0.303 ± 0.12	4.800 ± 1.229
ants-2-1	**2.298** ± 0.117	**0.485** ± 0.05	**0.369** ± 0.038	4.5 ± 0.777	3.941 ± 0.487	0.246 ± 0.159	0.186 ± 0.121	3.2 ± 0.632
ants-2-2	25.091 ± 0.693	**0.699** ± 0.047	**0.529** ± 0.042	5.767 ± 1.223	25.541 ± 1.825	0.514 ± 0.111	0.378 ± 0.093	5.1 ± 1.101
company-emails	**41.293** ± 0.942	**0.883** ± 0.013	**0.707** ± 0.016	3.233 ± 0.971	53.054 ± 1.485	0.601 ± 0.069	0.491 ± 0.059	1.6 ± 1.075
eu-email-4	**3.915** ± 0.165	0.388 ± 0.059	0.299 ± 0.046	3.033 ± 1.377	5.671 ± 1.25	0.332 ± 0.14	0.254 ± 0.111	2.1 ± 1.595
eu-email-2	3.027 ± 0.14	0.511 ± 0.061	0.391 ± 0.048	2.867 ± 1.332	3.424 ± 0.747	0.421 ± 0.097	0.325 ± 0.077	3.6 ± 1.506
eu-email-3	**2.212** ± 0.094	**0.566** ± 0.049	**0.446** ± 0.042	5.5 ± 1.137	3.308 ± 0.47	0.298 ± 0.182	0.230 ± 0.143	4 ± 2.055
sp-hospital	48.827 ± 1.737	0.743 ± 0.036	0.546 ± 0.033	5.1 ± 1.296	**38.745** ± 3.474	0.773 ± 0.029	0.579 ± 0.032	5.6 ± 1.578
sp-hypertext	**103.697** ± 2.877	0.792 ± 0.024	0.605 ± 0.027	5.367 ± 1.299	147.505 ± 0.364	0.802 ± 0.027	0.617 ± 0.027	5.5 ± 1.08
sp-workplace	52.506 ± 1.479	0.629 ± 0.04	0.454 ± 0.032	4.233 ± 0.971	**44.164** ± 5.8	0.723 ± 0.065	0.537 ± 0.058	5.1 ± 1.101
sp-highschool	**729.497** ± 10.894	0.734 ± 0.018	0.550 ± 0.018	**4.567** ± 1.104	1345.323 ± 0.512	**0.860** ± 0.001	**0.678** ± 0.002	1.200 ± 0.422
haggle	**16.635** ± 0.399	**0.755** ± 0.005	**0.618** ± 0.006	5 ± 0.983	19.266 ± 1.043	0.637 ± 0.034	0.520 ± 0.034	6.8 ± 1.229

Table 3: Results for prediction of temporal closeness centrality

Experiment	DBGNN MAE	DBGNN Spearmanr	DBGNN Kendalltau	DBGNN hitsIn10	GCN MAE	GCN Spearmanr	GCN Kendalltau	GCN hitsIn10
ants-1-1	**4.306** ± 0.269	**0.931** ± 0.005	**0.790** ± 0.008	**8.533** ± 0.629	10.514 ± 1.082	0.422 ± 0.092	0.288 ± 0.066	3.100 ± 1.101
ants-1-2	**1.466** ± 0.083	**0.966** ± 0.005	**0.843** ± 0.012	**6.667** ± 0.479	5.829 ± 0.662	0.311 ± 0.138	0.215 ± 0.095	2.800 ± 0.919
ants-2-1	**0.552** ± 0.026	**0.981** ± 0.003	**0.895** ± 0.012	**8.400** ± 0.498	3.132 ± 0.73	0.565 ± 0.197	0.422 ± 0.156	4.800 ± 1.549
ants-2-2	**1.365** ± 0.077	**0.969** ± 0.003	**0.854** ± 0.01	**7.400** ± 0.675	4.648 ± 0.599	0.488 ± 0.164	0.359 ± 0.129	4.100 ± 0.738
company-emails	**4.006** ± 0.107	**0.976** ± 0.001	**0.876** ± 0.003	**7.000** ± 0.371	15.149 ± 0.519	0.543 ± 0.091	0.455 ± 0.062	2.600 ± 1.075
eu-email-4	**0.870** ± 0.07	**0.982** ± 0.001	**0.895** ± 0.005	**9.900** ± 0.305	5.330 ± 0.771	0.183 ± 0.21	0.134 ± 0.141	3.900 ± 1.792
eu-email-2	**0.777** ± 0.04	**0.983** ± 0.001	**0.901** ± 0.005	**8.767** ± 0.43	5.065 ± 0.812	0.453 ± 0.123	0.324 ± 0.091	1.800 ± 1.549
eu-email-3	**0.489** ± 0.04	**0.994** ± 0.001	**0.944** ± 0.005	**8.667** ± 0.661	7.309 ± 1.533	0.597 ± 0.102	0.443 ± 0.074	3.500 ± 1.509
sp-hospital	**5.008** ± 0.281	**0.892** ± 0.008	**0.726** ± 0.013	**7.233** ± 0.774	7.622 ± 0.17	0.579 ± 0.032	0.417 ± 0.022	4.700 ± 0.823
sp-hypertext	17.767 ± 1.633	**0.967** ± 0.003	**0.843** ± 0.007	**8.567** ± 0.728	17.218 ± 0.248	0.733 ± 0.013	0.552 ± 0.009	4.700 ± 0.675
sp-workplace	**2.268** ± 0.09	**0.916** ± 0.006	**0.760** ± 0.009	**7.800** ± 0.551	4.909 ± 0.468	0.570 ± 0.072	0.406 ± 0.057	3.200 ± 0.919
sp-highschool	**17.369** ± 0.917	**0.885** ± 0.005	**0.707** ± 0.006	**7.433** ± 0.774	38.054 ± 0.224	0.601 ± 0.004	0.447 ± 0.003	0.000 ± 0.0
haggle	**0.806** ± 0.122	**0.961** ± 0.002	**0.879** ± 0.004	**8.600** ± 0.498	9.987 ± 3.081	0.417 ± 0.205	0.319 ± 0.156	2.600 ± 1.955

5 Conclusion

In summary, we investigate the problem of predicting temporal betweenness and closeness centralities in temporal graphs. To this end, we use a recently proposed causality-aware graph neural network architecture, which relies on higher-order De Bruijn graph models of time-respecting paths. An empirical study in which we compare our approach with a time-neglecting graph neural network demonstrates the potential of our method. A comparative analysis in 13 empirical temporal networks highlights differences between static and temporal centralities that are likely due to the underlying temporal patterns, and shows that our model is generally better at predicting temporal closeness compared to betweenness. An evaluation of scalability reveals that our model offers training and inference times that are comparable to those of a simpler GCN model, while yielding better performance. We finally investigate the (static) embeddings produced by the last message passing layer of our architecture and show that they better capture temporal centralities compared to a GCN model.

While we hope that our work is of interest for the Temporal Graph Learning community, this workshop paper necessarily leaves open questions that we will address in future work. A first limitation is that –due to time constraints– we did not perform a comprehensive hyperparameter exploration. This affects multiple aspects of our analysis, such as the investigation of multiple values for the maximum time difference δ, the maximum order k of the De Bruijn graphs used in the DBGNN architecture, as well as the number and width of graph convolutional layers. While we do report optimal values across three learning rates for all models, a more thorough investigation of this hyperparameter must be considered future work. A second open issue is the comparison of our approach to a more comprehensive choice of baselines. This comparison should include additional time-neglecting GNN architectures like Graph Attention Networks (GAT) [50], Graph Isomorphism Networks (GINs) [51] or embedding techniques like DeepWalk [52] or node2vec [53]. Moreover, we miss a comparison against other time-aware representation learning techniques like, e.g., HONEM [54] or EVO [55]. While we did not utilize additional node features like, e.g., degrees, static centralities, or embeddings, we expect that including such features could further improve our results.

In the light of these open issues, the investigation presented for the first time in this workshop paper must necessarily appear preliminary. We nevertheless hope that our work shows that the prediction of temporal centralities is an interesting problem for temporal graph learning, which could potentially be included in community-based benchmarks like TGB [17]. Moreover our study highlights the potential of using *compact, static but time-aware graph neural network architectures* for node-level regression tasks in time series data on temporal graphs. We thus hope that our work is of interest for the Temporal Graph Learning community and we would appreciate feedback and suggestions.

Acknowledgments

Ingo Scholtes acknowledges support by the Swiss National Science Foundation (SNF), grant number 176938, as well as by the German Federal Ministry of Education and Research, grant number 100582863 (TissueNet).

References

[1] Bavelas, A. Communication patterns in task-oriented groups. *The journal of the acoustical society of America* **22**, 725–730 (1950).

[2] Freeman, L. C. A set of measures of centrality based on betweenness. *Sociometry* 35–41 (1977).

[3] Holme, P. Modern temporal network theory: a colloquium. *The European Physical Journal B* **88**, 1–30 (2015).

[4] Lentz, H. H. K., Selhorst, T. & Sokolov, I. M. Unfolding accessibility provides a macroscopic approach to temporal networks. *Phys. Rev. Lett.* **110**, 118701 (2013). URL `http://link.aps.org/doi/10.1103/PhysRevLett.110.118701`.

[5] Pan, R. K. & Saramäki, J. Path lengths, correlations, and centrality in temporal networks. *Physical Review E* **84**, 016105 (2011).

[6] Pfitzner, R., Scholtes, I., Garas, A., Tessone, C. J. & Schweitzer, F. Betweenness preference: Quantifying correlations in the topological dynamics of temporal networks. *Phys. Rev. Lett.* **110**, 198701 (2013). URL `http://link.aps.org/doi/10.1103/PhysRevLett.110.198701`. `https://doi.org/10.1103/PhysRevLett.110.198701`.

[7] Tang, J., Musolesi, M., Mascolo, C., Latora, V. & Nicosia, V. Analysing information flows and key mediators through temporal centrality metrics. In *Proceedings of the 3rd Workshop on Social Network Systems*, 1–6 (2010).

[8] Kim, H. & Anderson, R. Temporal node centrality in complex networks. *Physical Review E* **85**, 026107 (2012).

[9] Alsayed, A. & Higham, D. J. Betweenness in time dependent networks. *Chaos, Solitons & Fractals* **72**, 35–48 (2015).

[10] Tsalouchidou, I., Baeza-Yates, R., Bonchi, F., Liao, K. & Sellis, T. Temporal betweenness centrality in dynamic graphs. *International Journal of Data Science and Analytics* **9**, 257–272 (2020).

[11] Buß, S., Molter, H., Niedermeier, R. & Rymar, M. Algorithmic aspects of temporal betweenness. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, 2084–2092 (Association for Computing Machinery, New York, NY, USA, 2020). URL `https://doi.org/10.1145/3394486.3403259`.

[12] Casteigts, A., Himmel, A.-S., Molter, H. & Zschoche, P. Finding temporal paths under waiting time constraints. *Algorithmica* **83**, 2754–2802 (2021).

[13] Santoro, D. & Sarpe, I. Onbra: Rigorous estimation of the temporal betweenness centrality in temporal networks. In *Proceedings of the ACM Web Conference 2022*, 1579–1588 (2022).

[14] Grando, F., Granville, L. Z. & Lamb, L. C. Machine learning in network centrality measures: Tutorial and outlook. *ACM Comput. Surv.* **51** (2018). URL `https://doi.org/10.1145/3237192`.

[15] Grando, F. & Lamb, L. C. Estimating complex networks centrality via neural networks and machine learning. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015*, 1–8 (IEEE, 2015). URL `https://doi.org/10.1109/IJCNN.2015.7280334`.

[16] Qarkaxhija, L., Perri, V. & Scholtes, I. De bruijn goes neural: Causality-aware graph neural networks for time series data on dynamic graphs. In *Learning on Graphs Conference*, 51–1 (PMLR, 2022).

[17] Huang, S. *et al.* Temporal graph benchmark for machine learning on temporal graphs. *arXiv preprint arXiv:2307.01026* (2023).

[18] Casteigts, A., Flocchini, P., Quattrociocchi, W. & Santoro, N. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems* **27**, 387–408 (2012).

[19] Holme, P. & Saramäki, J. Temporal networks. *Phys. Rep.* **519**, 97 – 125 (2012). URL `http://www.sciencedirect.com/science/article/pii/S0370157312000841`.

[20] Kempe, D., Kleinberg, J. & Kumar, A. Connectivity and inference problems for temporal networks. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, 504–513 (2000).

[21] Badie-Modiri, A., Karsai, M. & Kivelä, M. Efficient limited-time reachability estimation in temporal networks. *Physical Review E* **101**, 052303 (2020).

[22] Badie-Modiri, A., Karsai, M. & Kivelä, M. Efficient limited-time reachability estimation in temporal networks. *Phys. Rev. E* **101**, 052303 (2020). URL https://link.aps.org/doi/10.1103/PhysRevE.101.052303.

[23] Rosvall, M., Esquivel, A. V., Lancichinetti, A., West, J. D. & Lambiotte, R. Memory in network flows and its effects on spreading dynamics and community detection. *Nature communications* **5**, 4630 (2014).

[24] Vanhems, P. *et al.* Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. *PloS one* **8**, e73970 (2013). URL http://www.sociopatterns.org/datasets/hospital-ward-dynamic-contact-network/.

[25] Scholtes, I. *et al.* Causality-driven slow-down and speed-up of diffusion in non-markovian temporal networks. *Nature Communications* **5**, 5024 (2014). URL http://www.nature.com/ncomms/2014/140924/ncomms6024/full/ncomms6024.html. https://doi.org/10.1038/ncomms6024, 1307.4030.

[26] Salnikov, V., Schaub, M. T. & Lambiotte, R. Using higher-order markov models to reveal flow-based communities in networks. *Scientific reports* **6**, 23194 (2016).

[27] Lambiotte, R., Rosvall, M. & Scholtes, I. From networks to optimal higher-order models of complex systems. *Nature physics* **15**, 313–320 (2019).

[28] Pósfai, M. & Hövel, P. Structural controllability of temporal networks. *New Journal of Physics* **16**, 123055 (2014).

[29] Brandes, U. A faster algorithm for betweenness centrality. *Journal of mathematical sociology* **25**, 163–177 (2001).

[30] Riondato, M. & Kornaropoulos, E. M. Fast approximation of betweenness centrality through sampling. In *Proceedings of the 7th ACM international conference on Web search and data mining*, 413–422 (2014).

[31] Bader, D. A., Kintali, S., Madduri, K. & Mihail, M. Approximating betweenness centrality. In *Algorithms and Models for the Web-Graph: 5th International Workshop, WAW 2007, San Diego, CA, USA, December 11-12, 2007. Proceedings 5*, 124–137 (Springer, 2007).

[32] Haghir Chehreghani, M. An efficient algorithm for approximate betweenness centrality computation. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 1489–1492 (2013).

[33] Mendonça, M. R. F., Barreto, A. & Ziviani, A. Approximating network centrality measures using node embedding and machine learning. *IEEE Trans. Netw. Sci. Eng.* **8**, 220–230 (2021). URL https://doi.org/10.1109/TNSE.2020.3035352.

[34] Bergamini, E., Meyerhenke, H. & Staudt, C. L. Approximating betweenness centrality in large evolving networks. In *2015 Proceedings of the Seventeenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, 133–146 (SIAM, 2014).

[35] Riondato, M. & Upfal, E. Abra: Approximating betweenness centrality in static and dynamic graphs with rademacher averages. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **12**, 1–38 (2018).

[36] Scholtes, I., Wider, N. & Garas, A. Higher-order aggregate networks in the analysis of temporal networks: path structures and centralities. *The European Physical Journal B* **89**, 61 (2016). URL http://dx.doi.org/10.1140/epjb/e2016-60663-0. http://dx.doi.org/10.1140/epjb/e2016-60663-0.

[37] Bruijn, N. D. A combinatorial problem. In *Nederl. Akad. Wetensch., Proc. 49*, 461–467 (1946).

[38] Scholtes, I. When is a network a network?: Multi-order graphical model selection in pathways and temporal networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, CA, August 2017*, KDD '17, 1037–1046 (ACM, New York, NY, USA, 2017). URL http://doi.acm.org/10.1145/3097983.3098145. http://doi.acm.org/10.1145/3097983.3098145.

[39] Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (OpenReview.net, 2017). URL https://openreview.net/forum?id=SJU4ayYgl.

[40] Peixoto, T. Netzschleuder. networks.skewed.de/.

[41] Leskovec, J. & Krevl, A. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data (2014).

[42] Blonder, B. & Dornhaus, A. Time-ordered networks reveal limitations to information flow in ant colonies. *PloS one* **6**, e20298 (2011).

[43] Nurek, M. & Michalski, R. Combining machine learning and social network analysis to reveal the organizational structures. *Applied Sciences* **10**, 1699 (2020).

[44] Paranjape, A., Benson, A. R. & Leskovec, J. Motifs in temporal networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, 601–610 (2017).

[45] Isella, L. *et al.* What's in a crowd? analysis of face-to-face behavioral networks. *Journal of theoretical biology* **271**, 166–180 (2011).

[46] Génois, M. & Barrat, A. Can co-location be used as a proxy for face-to-face contacts? *EPJ Data Science* **7**, 1–18 (2018).

[47] 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). URL `http://www.sociopatterns.org/datasets/high-school-dynamic-contact-networks/`.

[48] Chaintreau, A. *et al.* Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing* **6**, 606–620 (2007).

[49] Casteigts, A., Himmel, A.-S., Molter, H. & Zschoche, P. The computational complexity of finding temporal paths under waiting time constraints (2021). `1909.06437`.

[50] Veličković, P. *et al.* Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[51] Xu, K., Hu, W., Leskovec, J. & Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[52] Perozzi, B., Al-Rfou, R. & Skiena, S. Deepwalk: online learning of social representations. In Macskassy, S. A., Perlich, C., Leskovec, J., Wang, W. & Ghani, R. (eds.) *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, 701–710 (ACM, 2014). URL `https://doi.org/10.1145/2623330.2623732`.

[53] Grover, A. & Leskovec, J. node2vec: Scalable feature learning for networks. In Krishnapuram, B. *et al.* (eds.) *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 855–864 (ACM, 2016). URL `https://doi.org/10.1145/2939672.2939754`.

[54] Saebi, M., Ciampaglia, G. L., Kaplan, L. M. & Chawla, N. V. Honem: learning embedding for higher order networks. *Big Data* **8**, 255–269 (2020).

[55] Belth, C., Kamran, F., Tjandra, D. & Koutra, D. When to remember where you came from: Node representation learning in higher-order networks. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 222–225 (2019).

A Additional results

In the following, we provide additional experimental results, namely the number of time-respecting paths in the test graphs along with the required computation time (table 4), the training and inference times for the GCN and the DBGNN model (table 5 and table 6), the optimal order of a k-th order De Bruijn graph model, inferred using the statistical model selection approach from [38] (table 7), additional results for the number of hits among the top-ranked nodes for betweenness and closeness centrality (table 8 and table 9).

Table 4: Computational complexity of path calculations in the test data. All results were obtained on a workstation with AMD Ryzen 9 3900X 12-core CPU and 64 GB RAM

data set	number of time-respecting paths	computation time
ants-1-1	27,308	190.62 s
ants-1-2	1,614	1.24 s
ants-2-1	362	0.24 s
ants-2-2	3,547	6.02 s
company-emails	64,246	285.64 s
eu-email-2	11,599	22.11 s
eu-email-3	2,770	6.37 s
eu-email-4	12,951	48.63 s
sp-hospital	139,724	206.97 s
sp-hypertext	264,300	356.02 s
sp-workplace	7,350	1.04 s
sp-highschool	1,680,651	10,159.71 s
haggle	38,079	4.97 s

Table 5: Training and inference time for betweenness centrality in seconds

experiments	Train		Eval	
	DBGNN	GCN	DBGNN	GCN
ants-1-1	4.525004	3.976138	0.002476	0.002315
ants-1-2	4.415789	4.192388	0.002383	0.002264
ants-2-1	4.333990	4.093549	0.002967	0.002304
ants-2-2	4.316440	4.088560	0.002610	0.003562
company-emails	4.577491	4.918629	0.002660	0.002632
eu-email-2	6.221958	5.719684	0.002624	0.002226
eu-email-3	6.269507	5.765907	0.002470	0.002503
eu-email-4	6.419320	5.738349	0.002483	0.002109
haggle	4.474058	3.991820	0.002908	0.002142
sp-highschool	12.662649	7.989266	0.003281	0.004080
sp-hospital	8.203254	7.664485	0.003546	0.001961
sp-hypertext	13.615240	7.816432	0.002886	0.002269
sp-workplace	8.429780	7.874174	0.002305	0.002002

Table 6: Training and inference time for closeness centrality in seconds

experiments	Train		Eval	
	DBGNN	GCN	DBGNN	GCN
ants-1-1	4.321659	4.092447	0.002468	0.002743
ants-1-2	4.317691	4.237622	0.002660	0.002339
ants-2-1	4.348724	4.054908	0.002440	0.002278
ants-2-2	4.326630	4.243561	0.002483	0.002158
company-emails	4.577662	4.416060	0.003556	0.002216
eu-email-2	6.485234	5.914584	0.002528	0.002099
eu-email-3	6.450350	5.999103	0.002631	0.002527
eu-email-4	7.260510	7.324913	0.003136	0.002314
haggle	4.649249	4.042492	0.002969	0.002297
sp-highschool	12.592707	9.586796	0.003289	0.002356
sp-hospital	8.688761	8.480865	0.002494	0.002364
sp-hypertext	9.988247	8.148300	0.003354	0.002093
sp-workplace	8.595048	7.866499	0.002670	0.002505

Table 7: Result of detection of optimal order based on likelihood ratio test.

data set	K_{opt} train	K_{opt} val
ants-1-1	2	2
ants-1-2	2	2
ants-2-1	1	1
ants-2-2	2	2
company-emails	2	2
eu-email-4	1	1
eu-email-2	2	2
eu-email-3	1	1
sp-hospital	2	2
sp-hypertext	2	2
sp-workplace	2	2
sp-highschool	2	2
haggle	2	2

Table 8: Results for hitsIn5 and hitsIn10 for prediction of temporal betweenness centrality and learning rate for which each experiment performed best

Experiment	DBGNN hitsIn5	DBGNN hitsIn30	DBGNN lr	GCN hitsIn5	GCN hitsIn30	GCN lr
ants-1-1	**2.467** ± 0.973	17.533 ± 4.377	0.100	0.400 ± 0.516	17.4 ± 2.757	0.010
ants-1-2	3.2 ± 0.664	**24.133** ± 0.937	0.100	2.5 ± 0.707	17.700 ± 2.058	0.100
ants-2-1	1.633 ± 0.718	18.667 ± 1.295	0.010	1.2 ± 0.919	15.7 ± 1.947	0.010
ants-2-2	1.333 ± 0.711	21.6 ± 1.429	0.100	1.2 ± 0.632	21.5 ± 1.78	0.100
company-emails	0.2 ± 0.484	18.7 ± 1.765	0.100	0.2 ± 0.632	16.3 ± 3.164	0.001
eu-email-4	1.4 ± 0.932	14.233 ± 1.87	0.010	1.1 ± 1.101	11.7 ± 3.889	0.001
eu-email-2	1.233 ± 0.898	14.467 ± 1.717	0.001	1.4 ± 0.966	15.2 ± 2.486	0.001
eu-email-3	2.967 ± 0.615	19.467 ± 1.383	0.001	1.5 ± 1.08	15.3 ± 3.498	0.001
sp-hospital	1.467 ± 0.937	22.767 ± 1.524	0.100	2.1 ± 0.876	23.4 ± 1.43	0.010
sp-hypertext	1.3 ± 0.794	18.833 ± 1.931	0.010	2.1 ± 0.568	19.3 ± 1.337	0.001
sp-workplace	1.7 ± 0.651	19.500 ± 1.28	0.100	2.1 ± 1.101	**22.200** ± 1.135	0.100
sp-highschool	**1.933** ± 0.64	16.167 ± 1.859	0.010	1.000 ± 0.0	16 ± 1.247	0.001
haggle	1.633 ± 0.809	24.933 ± 1.015	0.010	2.8 ± 1.033	23.9 ± 1.969	0.001

Table 9: Results for hitsIn5 and hitsIn10 for prediction of temporal closeness centrality and learning rate for which each experiment performed best

	DBGNN			GCN		
Experiment	hitsIn5	hitsIn30	lr	hitsIn5	hitsIn30	lr
ants-1-1	**3.700** ± 0.535	**25.200** ± 0.925	0.100	0.600 ± 0.966	15.700 ± 1.947	0.010
ants-1-2	**3.933** ± 0.365	**27.100** ± 0.662	0.100	1.600 ± 0.516	16.900 ± 2.283	0.001
ants-2-1	**5.000** ± 0.0	**28.200** ± 0.407	0.100	2.400 ± 1.506	20.200 ± 2.741	0.100
ants-2-2	**3.767** ± 0.568	**26.367** ± 0.556	0.001	0.700 ± 0.675	20.100 ± 2.234	0.001
company-emails	**3.467** ± 0.507	**26.333** ± 0.606	0.100	1.500 ± 0.85	14.700 ± 2.406	0.001
eu-email-4	**4.167** ± 0.461	**26.833** ± 0.913	0.100	1.400 ± 1.174	10.700 ± 3.433	0.001
eu-email-2	**4.833** ± 0.379	**25.133** ± 0.73	0.100	0.900 ± 1.287	9.700 ± 3.234	0.100
eu-email-3	**4.000** ± 0.0	**27.000** ± 0.0	0.100	1.300 ± 0.949	20.600 ± 1.506	0.100
sp-hospital	**2.667** ± 0.711	**23.800** ± 0.847	0.100	1.300 ± 0.483	20.800 ± 0.422	0.001
sp-hypertext	**4.100** ± 0.481	**26.033** ± 0.765	0.001	0.800 ± 0.422	19.600 ± 0.843	0.001
sp-workplace	3.2 ± 0.551	**24.500** ± 0.572	0.010	1.7 ± 1.059	20.400 ± 1.897	0.001
sp-highschool	**2.967** ± 0.615	**21.733** ± 1.081	0.001	0.000 ± 0.0	8.300 ± 1.16	0.001
haggle	**4.333** ± 0.479	**28.433** ± 0.504	0.100	0.400 ± 0.699	17.700 ± 7.304	0.010

B Model architecture and Details on Hyperparameters

Layer	Input dimensions	Output dimensions	Activation Function
GCNConv	$\|V\|$	16	Sigmoid
GCNConv	16	8	ELU
Linear layer	8	1	ELU

Table 10: Overview of proposed model architecture for simple GCN

Layer	Input dimensions	Output dimensions	Activation Function
GCNConv first order	$\|V\|$	16	Sigmoid
GCNConv second order	$\|E\|$	16	Sigmoid
Bipartite layer	16	8	ELU
Linear layer	8	1	ELU

Table 11: Overview of proposed model architecture for DBGNN

C Static vs temporal centralities

Let $G = (V, E)$ be a (static) graph, where V is a set of vertices or nodes and $(v, w) \in E$ are potentially directed edges or links from node v to w. Let us further consider weighted graphs, where we have a function $w : E \to \mathbb{N}$ that assigns integer weights to edges. In a static network $G = (V, E)$, we define a path (or walk) of length l from v_0 to v_l as any sequence of nodes $v_0, \ldots, v_l$ iff $(v_{i-1}, v_i) \in E$ for $i = 1, \ldots, l$. If every node occurs only once in the sequence, we call the sequence a *simple* path. A shortest path between two nodes v and w is a (not necessarily unique) path of legth l such that all other paths from v to w have length $l' \geq l$.

In static networks, shortest paths between pairs of nodes allow us to define *path-based nodes centralities*, which can be used to identify influential nodes. Here, we briefly introduce two important path-based centrality measures, namely *betweenness and closeness centrality*. For static networks without temporal interactions the betweenness centrality of a node v is calculated as

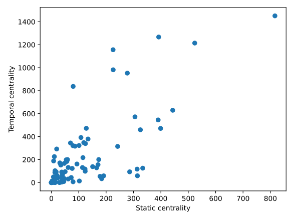
$$c_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$$

where $\sigma_{s,t}$ is the number of the shortest paths between nodes s and t and $\sigma_{s,t}(v)$ is the number of such paths that pass through node v. In other words the node is considered central if there are many shortest paths that pass through the node. The closeness centrality on the other hand is defined as

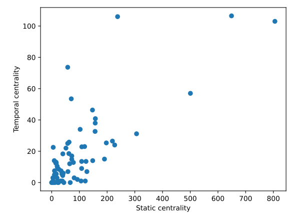
$$c_C(v) = \frac{1}{\sum_{u \in V} d(u, v)}$$

where $d(u, v)$ describes the distance (length of the shortest path) of node u to node v. Thus, in terms of closeness a node is considered more central if the overall distance to all other nodes in the graph is relatively small.

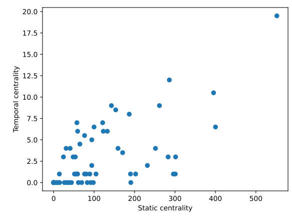
To contrast the temporal path-based centralities defined in section 2 with the corresponding static centralities defined above, in fig. 2 and fig. 3 we plot the temporal vs. static betweenness and closeness centralities of all nodes for all 13 empirical temporal graphs considered in our work (cf. table 1).



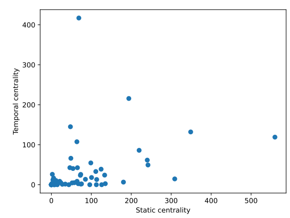
(a) ants-1-1



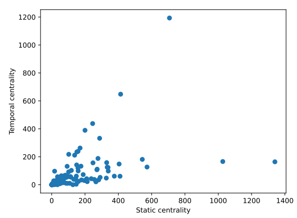
(b) ants-1-2



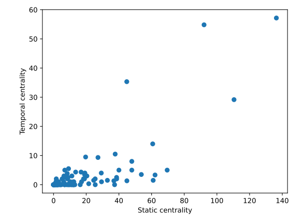
(c) ants-2-1



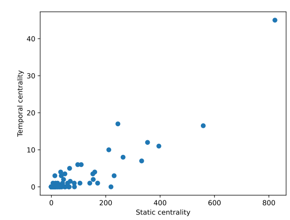
(d) ants-2-2



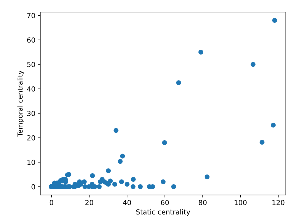
(e) company-emails



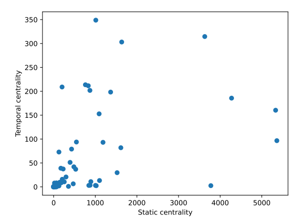
(f) eu-email-2



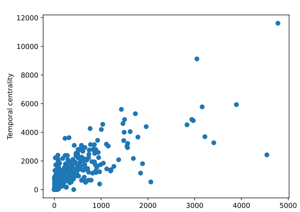
(g) eu-email-3



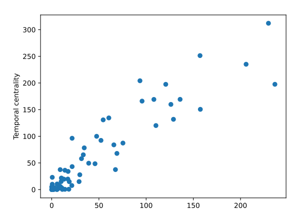
(h) eu-email-4



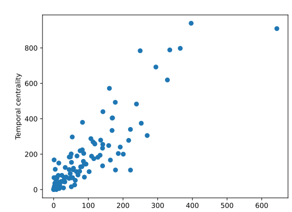
(i) haggle



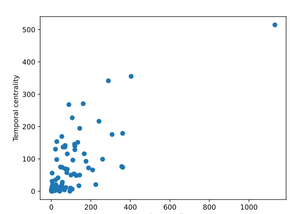
(j) sp-highschool



(k) sp-hospital

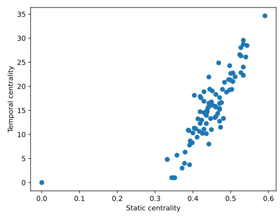


(l) sp-hypertext

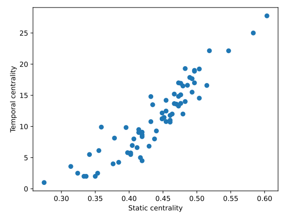


(m) sp-workplace

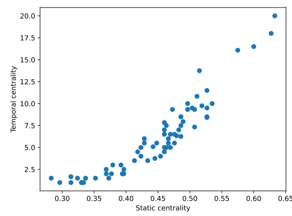
Figure 2: Static vs temporal betweenness centralities of all nodes in 13 empirical dynamic graphs



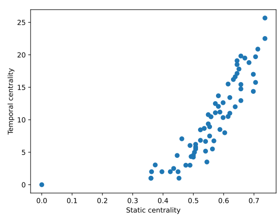
(a) ants-1-1



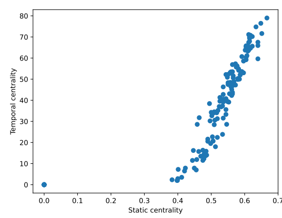
(b) ants-1-2



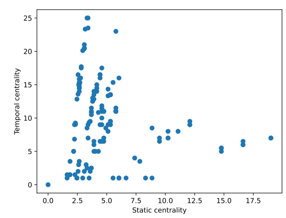
(c) ants-2-1



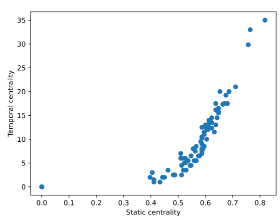
(d) ants-2-2



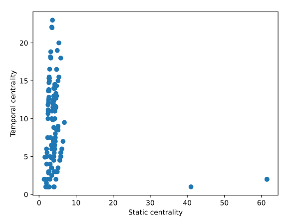
(e) company-emails



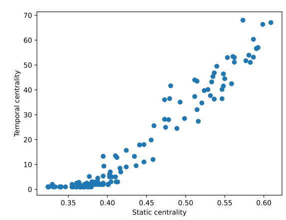
(f) eu-email-2



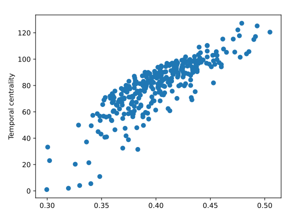
(g) eu-email-3



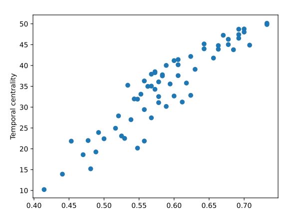
(h) eu-email-4



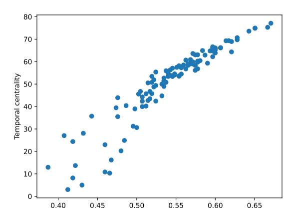
(i) haggle



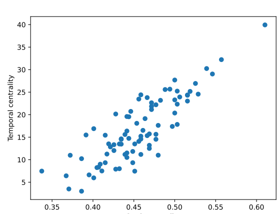
(j) sp-highschool



(k) sp-hospital



(l) sp-hypertext

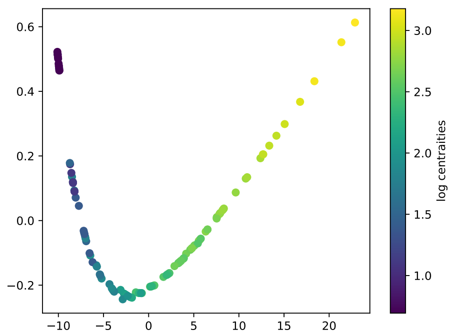


(m) sp-workplace

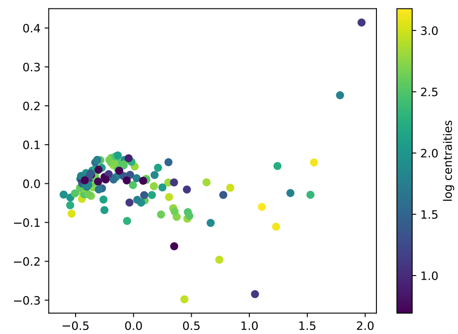
Figure 3: Static vs temporal closeness centralities of all nodes in 13 empirical dynamic graphs

D Visualization of Node embeddings

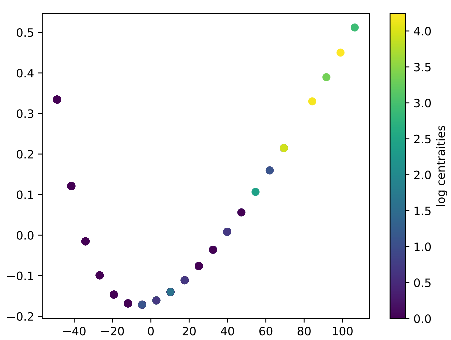
The plots in appendix D show the node emebddings obtained for a GCN and DBGNN model trained to predict temporal closeness centrality (a, b) as well as temporal betweenness centrality (c, d) for the `eu-email-4` data set.



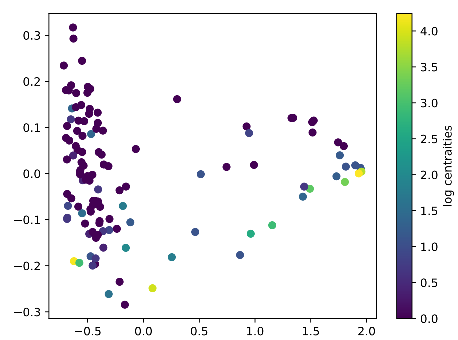
(a) Embedding of nodes based on DBGNN model trained for prediction of temporal closeness centrality in eu-email-4



(b) Embedding of nodes based on GCN architecture trained for prediction of temporal closeness centrality in eu-email-4



(c) Embedding of nodes based on DBGNN model trained for prediction of temporal betweenness centrality in eu-email-4



(d) Embedding of nodes based on GCN architecture trained for prediction of temporal betweenness centrality in eu-email-4