Dynamic Network Modeling from Motif-Activity

Giselle Zeno
Purdue University
gzenotor@purdue.edu

Timothy La Fond
Lawrence Livermore National Laboratory
lafond1@llnl.gov

Jennifer Neville
Purdue University
neville@purdue.edu

ABSTRACT

Graph structure in dynamic networks changes rapidly. Using temporal information about their connections, models for dynamic networks can be developed and used to understand the process of how their structure changes over time. Additionally, higher-order motifs have been established as building blocks for the structure of networks. In this paper, we first demonstrate empirically in three dynamic network datasets, that motifs with edges: (1) do not transition from one motif type to another (e.g, wedges becoming triangles and vice-versa); (2) motifs re-appear in other time periods and the rate depends on their configuration. We propose the Dynamic Motif-Activity Model (DMA) for sampling synthetic dynamic graphs with parameters learned from an observed network. We evaluate our DMA model, with two dynamic graph generative model baselines, by measuring different graph structure metrics in the generated synthetic graphs and comparing with the graph used as input. Our results show that employing motifs captures the underlying graph structure and modeling their activity recreates the fast changes seen in dynamic networks.

CCS CONCEPTS

• **Computing methodologies** → **Network science**.

KEYWORDS

temporal graphs, dynamic networks, motifs, motif evolution, network evolution

ACM Reference Format:
Giselle Zeno, Timothy La Fond, and Jennifer Neville. 2020. Dynamic Network Modeling from Motif-Activity. In *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3366424.3383301

1 INTRODUCTION

Networks are a way to study complex systems from a wide range of domains, such as social, biological, computing and communication networks. Until recently, these networks have been studied as static graphs, by modeling them as growing networks or aggregating temporal data into one graph. In reality, most of these networks are dynamic in nature and evolve over time, with nodes and edges being added and removed. Many generative models for static graphs have aimed to generate synthetic graphs that can simulate real-world networks. There have been extensions to the Erdős-Rényi random graph model [3], such as Exponential Random Graph Models (ERGMs) [27] and Chung-Lu models [2], to produce the degree distributions observed in social networks. Other models, such as Stochastic Block Models [19] and the Kronecker Product Graph Model (KPGM) [12], can generate clusters of connected nodes (communities). Statistical models where model parameters can be estimated from observed data, such as ERGMs and KPGMs, allow generation of graphs by sampling from the estimated distribution. However, these methods focus on capturing either global or local graph properties, but not both. Moreover, the graphs generated by both models do not reflect the natural variability of real-world graphs [17]. An extension to KPGMs, the mixed-KPGM (mKPGM) [18] uses parameter tying to capture the variance in the underlying distribution.

Network motifs or graphlets are small induced subgraphs occurring in a larger network structure, which have been shown to be building blocks for complex networks [15]. Graph motif structures, such as triangles, wedges, squares, etc. and the number of instances in graphs are a way to characterize and compare those graphs [28]. Comparing the triangle and wedge counts of two graphs is effectively contrasting them based on clustering coefficient and this is a means of doing anomaly detection or classification of graphs. The definition of motifs and graphlets has been extended to temporal networks by having all the edges in a given motif occur inside a time period [6, 20, 30]. Most of the work on temporal motifs has focused on modeling the evolution of these structures over time, such as wedges becoming triangles, in aggregated temporal data [1, 20, 29]. More recently, generative models using temporal motifs have been proposed for synthetic temporal networks where links are aggregated over time [23]. However, this approach assumes that edges will not be removed once they are placed.

Modeling this behavior of dynamic networks where links appear and disappear, such as communication patterns in social networks, is of interest. In this paper, we propose a dynamic graph generative model using temporal motifs as building blocks. In Section 2, we go over related work, with the same problem formulation, in more detail. In Section 3, we study the temporal motif behavior in subsequent time windows for various temporal network datasets. We found that, unlike in the aggregated temporal graphs, motifs did not evolve. We also observed that once a motif showed up, it would reappear in that same configuration (i.e., wedges remained wedges). In Section 4, we lay out our motif-activity dynamic generative model. In Section 5, we evaluate the structure of the temporal graphs generated by our model against two baselines. These baselines are also generative models for dynamic graphs but do not use motifs. We used various metrics for global and local graph structure to see if the generated synthetic graphs are similar to the dataset used for parameter estimation. We found that by using a motif-based approach, our model generates both local and global

WWW '20 Companion, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7024-0/20/04.
https://doi.org/10.1145/3366424.3383301

clustering close to what is observed in the real data. Each of the metrics we used measure different graph structure properties and should be considered together. To compare the graphs utilizing all the metrics, we performed a principal component analysis (PCA) to plot how far or close the graphs are to each other. We then show that our model has the most overlap with the real data. Finally, in Section 6, we discuss the results and go over future work.

2 RELATED WORK

Most models for temporal or dynamic networks have focused on modeling the edges over time, such as [4, 7, 24]. A straightforward approach to generating temporal networks is to generate first a static graph from some model, and for each link generate a sequence of contacts [5]. Holme [4] uses an approach where they draw degrees from a probability distribution and matches the nodes in random pairs for placing links. Then, for every link, they generate an active interval duration from a truncated power-law distribution, and uniform random starting time within that time frame. Rocha and Blondel [24] use a similar method where the active interval of a node starts when another node's interval ends. Another approach is to start with an empty graph. Then, for every node, they make it active according to a probability and connect it to m random nodes. Perra et al. [21] use a truncated power-law distribution for each node's probability of being active. Laurent et al. [11] extend this model to include memory driven interactions and cyclic closure. Other extensions include aging effects [16] and lifetimes of links [25]. Vestergard et al. [26] model nodes and links as being active or inactive using temporal memory effects. Zhang et al. [29] study the evolution of motifs in temporal networks by looking at changes in bipartite motifs in subsequent timesteps. Benson et al. [1] study higher-order networks and model how 3-node motifs evolve from being empty to becoming a triangle in aggregated temporal graphs. Purohit et al. [23] propose a generative model for synthetic temporal networks where links are aggregated over time (i.e., no link deletions).

3 EMPIRICAL STUDY

3.1 Datasets

3.1.1 Enron emails. The Enron email network consists of emails sent between employees of Enron [8, 9]. Nodes in the network are individual employees and edges are individual emails.

3.1.2 EU emails. This is the email communication network of a large, undisclosed European institution [9, 13]. Nodes represent individual persons and edges that at least one email has been sent from one person to the other. All edges are simple and spam emails have been removed from the dataset.

3.1.3 Wikipedia links (English). This dynamic network shows the evolution of hyperlinks between Wikipedia articles [9, 22]. The nodes represent articles. Edges include time-stamps and indicate that a hyperlink was added or removed depending on the edge weight (-1 for removal or +1 for addition).

3.2 Motif Evolution

Motifs have been demonstrated to evolve in aggregated temporal graphs (i.e. triadic closure) [1, 20, 29]. In this initial study, we investigated if similar dynamics occurred when considering subsequent time windows (e.g., if they appear, merge, split and/or disappear over time). We assume an underlying time-homogeneous discrete Markov process, where the graph structure at the next timestep $t+1$ depends on the current timestep t. Each timestep corresponds to a time window of the temporal graph. Then, we consider all 3-node motifs at each timestep to either transform from one motif type to another or remain the same, and isomorphisms are combined into the same configuration.

The transition probability matrices for both email datasets (Enron and EU) show that the motifs with edges (i.e., 1-edge, wedge, and triangle) will either keep their current motif type, or become empty with almost the equal probability (Figures 1a, 1b). This is also reflected when looking at the total transition counts across time periods (Figures 4, 5). For each motif type with edges, the counts of staying is very close to the counts of becoming empty at the next time period. In contrast, for the Wikipedia dataset, graph keeps growing with more links between articles being added with very few removed (Figure 6). This makes it unlikely to see any motif with edges becoming empty (Figure 1c).

We did not observe motifs with edges changing from one motif type to another (e.g, wedges becoming triangles and vice-versa), even when picking different time windows to create the timesteps. The matrix of total transition counts for each dataset (Figures 4, 5, 6) show that the only non-zero entries are: (1) the first column, where motifs become empty; (2) the first row, where motifs with edges appear after some time; (3) the diagonal, where the motifs keep the same motif type.

4 DYNAMIC MOTIF-ACTIVITY MODEL

Our generative model assumes there are two underlying processes: (1) nodes in the graph become active and remain that way, (2) motifs are "born" in some configuration, which we call motif type, and they reappear according to some rate. We define a dynamic graph as a sequence of graphs over time $G = \{G_1, \ldots, G_T\}$, where T is the number of timesteps. Our task it to sample a synthetic dynamic graph with similar structure and behavior as the observed data used to learn the model parameters.

We model the time until nodes become active as Exponential random variables with the same rate λ_V. Since we consider all possible 3-node motifs, there will be edges shared amongst them. Therefore, to estimate the inter-arrival rate for each motif, the counts of how many times it appeared is weighted by the number of shared edges with other motifs. For each motif type with edges (i.e., 1-edge, wedge, and triangle), we fit an Exponential distribution with the motif inter-arrival rates of that type. Motivated by our findings in Section 3, when a motif is first sampled it will keep the same configuration in the future. We also keep the motif type proportions constant when sampling new motifs.

4.1 Sampling

Our general sampling procedure is described in Alg. 1: In Line 2, we first sample the nodes that are active at each timestep (Alg. 3).

(a) Enron emails **(b) EU emails** **(c) Wikipedia links**

Figure 1: Motif Transition Probabilities

Whenever new nodes become active, we calculate the new triplets of active nodes that are now eligible to be sampled as a motif in Line 5. In Line 6, we proceed to sample motifs and their types (Alg. 4). We store the motifs we just sampled in Line 7, and sample their edges afterwards (Alg. 6).

For now, we assume we have as input initial motif types and edge placements for the triplets. Then, if there aren't enough triplets assigned to one of the motif types "buckets" in the initial assignments to maintain the proportions, we reassign triplets to meet the expected counts for each "bucket" and add or remove any edges necessary (Alg. 5). In line 10, we store the edges assigned to each motif. Next, we sample inter-arrival times for the motifs (Alg. 7). Finally, we construct the dynamic graph $G = \{G_1, \ldots, G_T\}$ using the motif edge placements M^E and timestamps M^S sampled.

4.2 Learning

Given an observed dynamic graph $G = \{G_1, \ldots, G_T\}$, consisting of a sequence of graphs G_t that correspond to time windows, we estimate the input parameters for our generative process. We begin by estimating the node arrival rate $\widehat{\lambda}_V$ using the time from when the dataset begins until each node has it's first edge.

Then, in Alg. 2 we find the motifs in each graph snapshot G_t, $\forall t \in [1, \ldots, T]$. For each 3-node motif $\{u, v, w\}$, we find its motif type i at timestep t (Line 8). If we have previously seen $\{u, v, w\}$ and the motif type i is of higher order at the current timestep t, then we update the type stored to be i (Line 13). For example, if we observe $\{u, v, w\}$ is a triangle at timestep t and previously saw it as a wedge, we update $M^T(\{u, v, w\})$ as a triangle.

We calculate the proportions $\widehat{p}_M^{(i)}$ of each motif type in the graph, where i is the number of links in the motif (i.e., $i = 1$ for 1-edge, $i = 2$ for wedge, and $i = 3$ for triangle).

$$\widehat{p}_M^{(i)} = \frac{\left\{\{u, v, w\} \in M \,\middle|\, M^T(\{u, v, w\}) = i\right\}}{\binom{N}{3}}, \quad \text{for } i \in [1, 2, 3]$$

$$\widehat{p}_M^{(0)} = 1 - \sum_{i=1}^{n} \widehat{p}_M^{(i)} \tag{1}$$

where M is the set of motifs, and $\{u, v, w\}$ is a motif consisting of the nodes u, v, w.

Algorithm 1: `SampleDynamicGraph`

input : T num. of timesteps to generate,
N num. of nodes,
λ_V rate at which nodes become active,
$p_M = (p_M^{(0)}, \ldots, p_M^{(3)})$ proportions motif types,
$\lambda_M = (\lambda_M^{(1)}, \ldots, \lambda_M^{(3)})$ rates of inter-arrival rates,
I^T initial motif type assignments,
I^E initial motif edge assignments

output: $G = \{G_1, \ldots, G_T\}$, where $G_t = (V_t, E_t, S_t)$

1 **begin**
2 $V \leftarrow$ `GetActiveNodes(`T, N, λ_V`)`
3 $M \leftarrow \emptyset, M^S \leftarrow \emptyset, M^E \leftarrow \emptyset$
4 **for** $t \in [1, \ldots, T]$ **do**
`// New active triplets at timestep` t
5 $U_t \leftarrow \{m = \{u, v, w\} \subseteq V_t \mid u < v < w,\ m \notin U_{t-1}\}$
6 $M_t, M_t^T \leftarrow$ `SampleMotifTypes(`U_t, p_M, I^T`)`
7 $M \leftarrow M \cup M_t$ `// save new motifs`
8 $M^T \leftarrow M^T \cup M_t^T$ `// store their types`
9 $M_t^E \leftarrow$ `SampleMotifEdges(`M_t, M^T, I^T, I^E`)`
10 $M^E \leftarrow M^E \cup M_t^E$ `// store their edges`
11 $M_t^S \leftarrow$ `SampleMotifTimestamps(`t, M_t, M^T, λ_M`)`
12 $M^S \leftarrow M^S \cup M_t^S$ `// store their timestamps`
13 $G \leftarrow$ `ConstructGraph(`M, M^E, M^S`)`

Finally, for each motif type with edges ($i \in [1, 2, 3]$), we learn a rate $\widehat{\lambda}_M^{(i)}$ of inter-arrival rates from the motifs of that type (Eq. 2a, Eq. 2b). We do not need to estimate rates for the empty motif type ($i = 0$).

$$\widehat{\lambda}_M^{(i)} = \frac{\sum_{\{u,v,w\} \in M^{(i)}} \left(\widehat{\lambda}_M(\{u, v, w\})\right)}{|M^{(i)}|} \tag{2a}$$

$$\widehat{\lambda}_M(\{u, v, w\}) = \frac{\sum_{t=1}^{T} C_t^M(\{u, v, w\})}{T} \tag{2b}$$

Since edges might be shared by more than one motif, we use edge-weighted Poisson counts C_t^M. per timestep t, to estimate the inter-arrival rate for each motif $\{u, v, w\}$ (Eq. 3). The weights $W_t^{(i)}$ will

depend on the motif type i of $\{u, v, w\}$ and are calculated for each edge $(s, d) \in E_t(\{u, v, w\})$ (Eq. 4).

$$C_t^M(\{u, v, w\}) = \frac{\sum_{(s,d)\in E_t(\{u,v,w\})} W_t^{(i)}((s, d))}{\left|E_t(\{u, v, w\}\right|} \tag{3}$$

We calculate the edge weights depending on motif type i (i.e., Eq. 4a for triangles, Eq. 4b for wedges, and Eq. 4c for 1-edge motifs). We give larger edge-weight to motifs types with more edges, since they are more likely to produce the observed edges. This also ensures that motifs types with smaller proportion $p_M^{(i)}$, such as triangles, have a high enough inter-arrival rate to show up.

$$W_t^{(3)}((s, d)) = \frac{1}{\#\text{ triangles } (s, d) \text{ is in}} \tag{4a}$$

$$W_t^{(2)}((s, d)) = \begin{cases} 0 & \text{if (s,d) in a } \textit{triangle} \\ \frac{1}{\#\text{ wedges } (s, d) \text{ is in}} & \text{otherwise} \end{cases} \tag{4b}$$

$$W_t^{(1)}((s, d)) = \begin{cases} 0 & \text{if (s,d) in a } \textit{triangle} \text{ or } \textit{wedge} \\ \frac{1}{\#\text{ 1-edges } (s, d) \text{ is in}} & \text{otherwise} \end{cases} \tag{4c}$$

5 METHODOLOGY

5.1 Evaluation

In our empirical study we found motifs show up and disappear at different times in the email datasets. We also observed that the Enron email dataset mid-timeline (before the scandal became public) has similar behavior to the EU email dataset. We evaluate on the Enron email dataset during February 2000 and we created daily timesteps (excluding weekends and holidays), for a total of 20 timesteps.

The related work using motif-based models for temporal graphs focus on the aggregate graph and not the dynamic changes over time. With that in mind, we picked two baselines that aim to model the changes in dynamic graphs. Using the Enron graph as input, we estimate initial motif configurations and all parameters of our DMA model as described in subsection 4.2. We also estimate all parameters of the baselines from the Enron graph as described below in subsubsection 5.2.1 and subsubsection 5.2.2.

5.2 Baselines

5.2.1 Static Networks with Link Dynamics Model (SNLD). We used an approach based on [4], where they begin by generating a static graph and then generate a series of events. Their procedure begins by sampling degrees from a probability distribution. They refer to these degrees as "stubs" and they create links by connecting these "stubs" randomly. Finally, for each link, they assign a time-series from an inter-event distribution

We start by sampling the degrees from a Truncated Power-law distribution. Since our starting point is a static graph, we assume all the nodes to be active already. Then, we sample inter-event times for every edge. We found that we could best model the edge inter-event times in the real data using an Exponential distribution. To learn the Truncated Power-law parameters, we aggregated and simplified the real graph.

5.2.2 Activity-Driven Network Model (ADN). We use the approach in [11], which extends the model in [21] by adding memory effects and triadic closure. The triadic closure takes place when node i connects to node k forming a triangle with it's current neighbor j and the memory effect is added counting the number of times that the nodes have connected up to the current time t. The procedure starts by creating an empty graph G_t at each timestep. Then, for each node i: delete it with probability p_d or mark it as active with probability a_i. If the node is "deleted", then the edges in the current timestep are removed, the counts of connections set to zero, and another degree is sampled to estimate a new a_i. If a node i is sampled as active, we connect it to either: (1) a neighbor j, (2) a neighbor of j, or (3) a random node. We base the probability to create new edges a_i on the degree of node i, which we sample from a Truncated Power-law distribution. We estimate the parameters using the average degree across timesteps for the nodes in the real graph. There is a fixed probability p_d for any node being "deleted" (losing its previous connections memory and sampling a new a_i). We estimate this probability using the average ratio of nodes becoming disconnected in the next timestep. To estimate the probability for triadic closure (forming a triangle), we use the average global clustering coefficient across timesteps.

5.3 Results

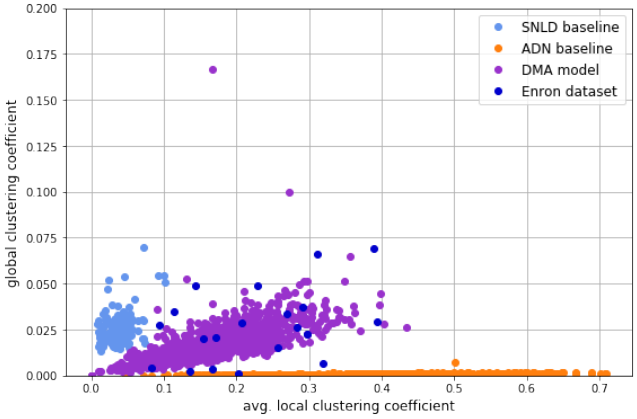
We use the following graph structure metrics to evaluate the models against the real data: density, average local clustering coefficient, global clustering coefficient, diameter and average path length of largest connected component (LCC), average and median node degree, average betweenness centrality, and s-metric.

Density measures ratio of edges in the graph versus the number of edges if it were a complete graph. The local clustering coefficient of a node measures how close are its neighbors to becoming a clique. While the global clustering coefficient measures the ratio of closed triplets (triangles) to open and closed triplets (wedges and triangles). The betweenness centrality of a node measures the number of shortest paths that pass through that node. S-metric measures the extent to which a graph has hub-like structure [14].

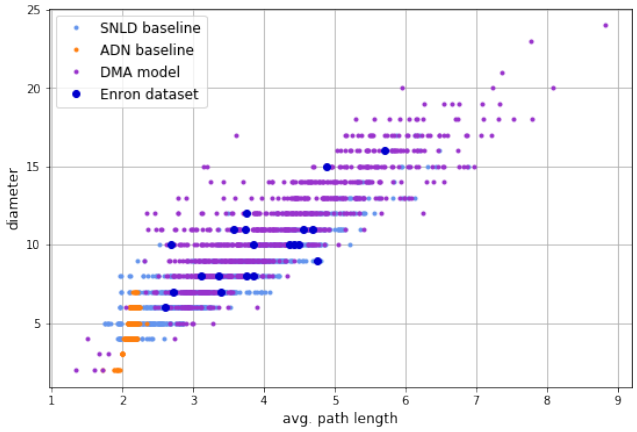
Table 1: Average Metrics for Graph Snapshots

	SNLD	ADN	DMA	Enron
density	0.0004	0.0007	0.0005	0.0008
avg. local clustering	0.005	0.378	0.172	0.226
global clustering	0.003	0.001	0.019	0.027
path length LCC	3.437	2.123	4.142	3.911
diameter LCC	7.908	4.562	10.888	9.900
avg. node degree	0.352	0.587	0.421	0.729
median node degree	0.000	0.000	0.000	0.050
avg. betweenness	1.200	37.26	16.16	61.75
s-metric	462.2	51,438.3	2,832.3	14,507.6

5.3.1 Generated Graph Snapshots. We generated 1,000 timesteps with each of the models to account for any variability during the beginning of the graph generation. The averages across timesteps for each model are reported in Table 1. The SNLD baseline generates almost no local or global clustering in the graph snapshots. The

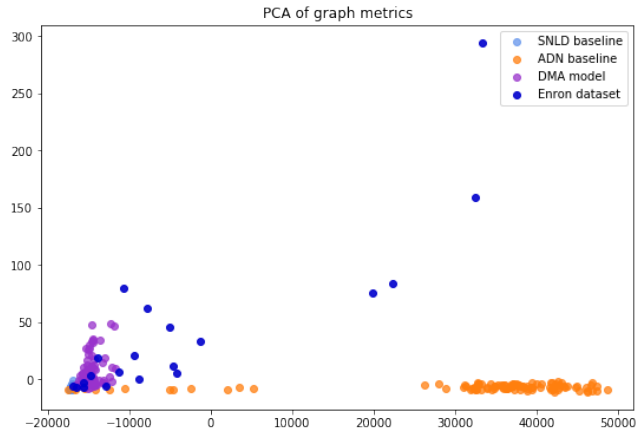


(a) Local and Global Clustering Coefficients

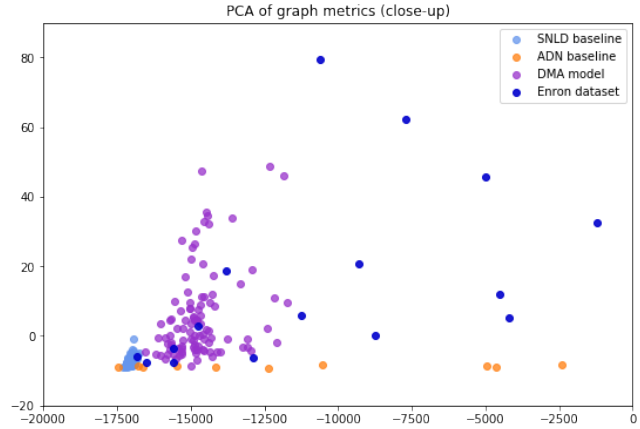


(b) Diameter and Average Path Length

Figure 2: Graph Snapshot Metrics



(a) Overview of all models and real data



(b) Close up of SNLD baseline and our model

Figure 3: PCA of Metrics for the Graph Snapshots

snapshots it generates have many isolated edges and wedges, and long chains, which explain why there is almost no clustering but large path lengths. The DMA model creates graph snapshots with multiple hubs connected to each other by longer chains, similar to what we have seen in the Enron dataset. Some of the Enron graph snapshots have large hubs which result in a higher s-metric on average. The ADN baseline generates more local clustering but very little global clustering. The graph snapshots have very large hubs, giving a large s-metric value, and a few isolated edges. This explains the high local clustering but small diameter and global clustering.

To give context to these observations and reported averages, we plot the local and global clustering coefficients in Figure 2a, as well as the average path length and diameter of the graphs in in Figure 2b, for all timesteps. Both the ADN baseline and the DMA model are able to closely match the diameter and average path lengths. The DMA model is also able to generate local and global clustering that matches the Enron dataset.

We also visualize how close the generated snapshots are to the dataset, by doing a principal component analysis with all the metrics (Figure 3a). We also created a "zoomed in" version to look at any overlaps between the models in the lower left corner (Figure 3b). We can observe, that when considering all the metrics, the DMA model generates snapshots with the most similar structure to the real data.

Table 2: Metrics for Aggregated Graphs

	SNLD	ADN	DMA	Enron
density	0.0045	0.0047	0.0071	0.0080
avg. local clustering	0.027	0.716	0.308	0.537
global clustering	0.031	0.007	0.048	0.091
path length LCC	4.74	2.00	3.35	3.86
diameter LCC	13.00	2.00	8.00	9.00
avg. node degree	3.54	3.76	5.66	6.33
median node degree	2.00	3.00	3.00	3.00
avg. betweenness	1,308	395	832	1,100
s-metric	105,814	1,754,359	797,402	922,121

5.3.2 Generated Aggregated Graphs. We also evaluate the overall graph structure without the temporal aspect by looking at the

metrics for the aggregated graphs (Table 2). Since we generated graphs for many more timesteps, we sampled 20 timesteps from the synthetic graphs in order to compare with the Enron dataset. The aggregated temporal graph generated with the DMA model is the closest to the Enron email network in most metrics. The ADN baseline achieves more local clustering and the highest s-metric due to the very large hubs it creates.

6 DISCUSSION AND FUTURE WORK

We demonstrated in our empirical study of dynamic networks that motifs with edges: (1) do not change configurations (e.g., wedges becoming triangles and vice-versa); (2) once they appear, they will stay during the next time window or disappear. Whereas, the motif evolution process mentioned in [1, 20, 29] occurs in aggregated temporal graphs. In our evaluation, we observe that edge-based models for dynamic graphs are not able to capture the higher-order interactions; as is also the case with aggregate temporal graphs [1]. Moreover, even though the second baseline (ADN) [11] adds memory effects and a triadic closure mechanism, it does not create the amount of global clustering seen in the real data. Our Dynamic Motif-Activity Model (DMA), which maintains the motif type proportions, generates synthetic graphs with overall structure close to the dynamic graph used for parameter estimation. Furthermore, we tested our approach of using edge-weights, for the motif type inter-arrival rates estimation, by comparing the rates learned on a synthetic dynamic network with the ones sampled during its generation. We found that using this technique when unsure of which are the motifs present in the data, we can learn the rate parameters of the "true" distribution.

Our current model assumes as input initial motif type assignments for some of the triplets. Then, depending on how many motifs are needed to meet the expected counts of each motif type, it reassigns motif types for the triplets to maintain the motif type proportions. We are working on a method to sample the motif types from a probability distribution based on node roles [10]. For example, we would like to model the probability of a triplet becoming a wedge based on how likely the nodes in the triplet are to participate in a wedge.

ACKNOWLEDGMENTS

This research is supported by NSF under contract numbers: CCF-0939370 and IIS-1618690.

REFERENCES

[1] Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, and Jon Kleinberg. 2018. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences of the United States of America* 115, 48 (2018), E11221–E11230. https://doi.org/10.1073/pnas.1800683115 arXiv:1802.06916

[2] Fan Chung and Linyuan Lu. 2002. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences of the United States of America* 99, 25 (2002), 15879–15882. https://doi.org/10.1073/pnas.252631999

[3] P Erdös and a Rényi. 1959. On random graphs. *Publicationes Mathematicae* 6 (1959), 290–297. https://doi.org/10.2307/1999405 arXiv:1205.2923

[4] Petter Holme. 2013. Epidemiologically Optimal Static Networks from Temporal Network Data. *PLoS Computational Biology* 9, 7 (2013). https://doi.org/10.1371/journal.pcbi.1003142

[5] Petter Holme. 2015. Modern temporal network theory: a colloquium. *European Physical Journal B* 88, 9 (2015). https://doi.org/10.1140/epjb/e2015-60657-4 arXiv:1508.01303

[6] Yuriy Hulovatyy, Huili Chen, and T Milenković. 2015. Exploring the structure and function of temporal networks with dynamic graphlets. *Bioinformatics* 31, 12 (2015), i171–i180.

[7] Brian Karrer and M. E. J. Newman. 2009. Random graph models for directed acyclic networks. *Physical Review E* (2009). https://doi.org/10.1103/PhysRevE.80.046110 arXiv:0907.4346

[8] Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*. Springer, 217–226.

[9] Jérôme Kunegis. 2013. KONECT - The koblenz network collection. *Proceedings of the 22nd International Conference on World Wide Web* (2013), 1343–1350.

[10] Timothy La Fond, Dan Roberts, Jennifer Neville, James Tyler, and Stacey Connaughton. 2012. The impact of communication structure and interpersonal dependencies on distributed teams. (2012), 558–565.

[11] Guillaume Laurent, Jari Saramäki, and Márton Karsai. 2015. From calls to communities: a model for time-varying social networks. *European Physical Journal B* 88, 11 (2015), 1–10. https://doi.org/10.1140/epjb/e2015-60481-x arXiv:1506.00393

[12] Jure Leskovec, D Chakrabarti, J Kleinberg, C Faloutsos, and Zoubin Ghahramani. 2010. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research* 11 (2010), 985–1042. https://doi.org/10.1145/1756006.1756039 arXiv:0812.4905

[13] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2007. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007), 2.

[14] Lun Li, David Alderson, John C Doyle, and Walter Willinger. 2006. Towards a Theory of Scale-Free Graphs : Definition , Properties , and Implications. 2 (2006), 431–523.

[15] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. 2002. Network motifs: Simple building blocks of complex networks. *Science* 298, 5594 (2002), 824–827. https://doi.org/10.1126/science.298.5594.824

[16] Antoine Moinet, Michele Starnini, and Romualdo Pastor-Satorras. 2015. Burstiness and Aging in Social Temporal Networks. *Physical Review Letters* 114, 10 (2015). https://doi.org/10.1103/PhysRevLett.114.108701 arXiv:1412.0587

[17] Sebastian Moreno and Jennifer Neville. 2009. An Investigation of the Distributional Characteristics of Generative Graph Models. *Win* (2009).

[18] Sebastian Moreno, Jennifer Neville, and Sergey Kirshner. 2018. Tied kronecker product graph models to capture variance in network populations. *ACM Transactions on Knowledge Discovery from Data* 12, 3 (2018). https://doi.org/10.1145/3161885

[19] Krzysztof Nowicki and Tom a. B Snijders. 2001. Estimation and Prediction for Stochastic Blockstructures. *J. Amer. Statist. Assoc.* 96, 455 (2001), 1077–1087. https://doi.org/10.1198/016214501753208735

[20] Ashwin Paranjape, Austin R Benson, and Jure Leskovec. 2017. Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 601–610.

[21] N. Perra, B. Gonçalves, R. Pastor-Satorras, and A. Vespignani. 2012. Activity driven modeling of time varying networks. *Scientific Reports* 2 (2012), 1–7. https://doi.org/10.1038/srep00469 arXiv:1203.5351

[22] Julia Preusse, Jérôme Kunegis, Matthias Thimm, Steffen Staab, and Thomas Gottron. 2013. Structural Dynamics of Knowledge Networks. *ICWSM* 17 (2013), 18.

[23] Sumit Purohit, Lawrence B Holder, and George Chin. [n.d.]. Temporal Graph Generation Based on a Distribution of Temporal Motifs. *Proceedings of the 14th International Workshop on Mining and Learning with Graphs* ([n. d.]), 7.

[24] Luis E.C. Rocha and Vincent D. Blondel. 2013. Bursts of Vertex Activation and Epidemics in Evolving Networks. *PLoS Computational Biology* 9, 3 (2013). https://doi.org/10.1371/journal.pcbi.1002974

[25] Albert Sunny, Bhushan Kotnis, and Joy Kuri. 2015. Dynamics of history-dependent epidemics in temporal networks. *Physical Review E* 92, 2 (2015), 022811.

[26] Christian L. Vestergaard, Mathieu Génois, and Alain Barrat. 2014. How memory generates heterogeneous dynamics in temporal networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 90, 4 (2014), 1–19. https://doi.org/10.1103/PhysRevE.90.042805 arXiv:1409.1805

[27] Stanley Wasserman and Philippa Pattison. 1996. Logit models and logistic regressions for social networks: I. An introduction to markov graphs and p. *Psychometrika* 61, 3 (1996), 401–425. https://doi.org/10.1007/BF02294547

[28] Ömer Nebil Yaveroäa Lu, Noël Malod-Dognin, Darren Davis, Zoran Levnajic, Vuk Janjic, Rasa Karapandza, Aleksandar Stojmirovic, and Nataša Pržulj. 2014. Revealing the hidden Language of complex networks. *Scientific Reports* 4 (2014), 1–9. https://doi.org/10.1038/srep04547

[29] Xin Zhang, Shuai Shao, H Eugene Stanley, and Shlomo Havlin. 2014. Dynamic motifs in socio-economic networks. *EPL (Europhysics Letters)* 108, 5 (2014), 58001.

[30] Qiankun Zhao, Yuan Tian, Qi He, Nuria Oliver, Ruoming Jin, and Wang Chien Lee. 2010. Communication motifs: A tool to characterize social communications. *International Conference on Information and Knowledge Management, Proceedings* (2010), 1645–1648. https://doi.org/10.1145/1871437.1871694

A ADDITIONAL FIGURES

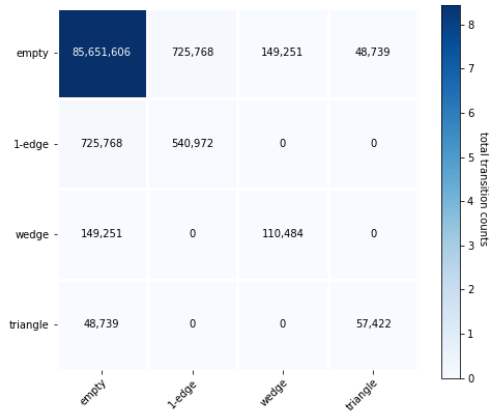


Figure 4: Total Transition Counts: Enron emails

Figure 5: Total Transition Counts: EU emails

Figure 6: Total Transition Counts: Wikipedia links

B ALGORITHMS

Algorithm 2: `GetMotifsGraph`

input : $G = \{G_1, \ldots, G_T\}$ graph to learn from
T num. of timesteps
output: M motifs in G
M^T motif types

begin
  $M \leftarrow \emptyset$; $M^T \leftarrow \emptyset$
  for $t \in [1, \ldots, T]$ **do**
    for $(u, v) \in E_t$ **do**
      if $u \neq v$ **then**
        for $w \in V_t$ **do**
          if $w \neq u$ **and** $w \neq v$ **then**
            `// number unique edges is motif type` i
            $i = |M^E(\{u, v, w\})|$
            if $\{u, v, w\} \notin M$ **then**
              $M \leftarrow M \cup \{\{u, v, w\}\}$
              $M^T(\{u, v, w\}) \leftarrow i$
            else if $i > M^T(\{u, v, w\})$ **then**
              `// prefer higher-order motif types`
              $M^T(\{u, v, w\}) \leftarrow i$

Algorithm 3: `GetActiveNodes`

input : T num. of timesteps
N num. of nodes
λ_V node arrival rate
output: V active nodes per timestep

begin
  $V_t \leftarrow \emptyset$, $\forall j \in [1, \ldots, T]$
  for $v \in [1, \ldots, N]$ **do**
    $a \sim Exp(\lambda_V)$ `// sample arrival time`
    for $t \in [a, \ldots, T]$ **do**
      `// add to active nodes each timestep` t
      $V_t \leftarrow V_t \cup \{v\}$
  $V = V_1 \cup \ldots \cup V_T$

Algorithm 4: `SampleMotifTypes`

input : U_t active triplets
M^T motif types
$p_M = (p_M^{(0)}, \ldots, p_M^{(3)})$ proportions motif types,
I^T initial type assignments
output: M_t sampled motifs,
M^T updated motif types

begin
  Shuffle(U_t)
  // Categorize triplets using type from data
  $B^{(3)} \leftarrow \{\{u,v,w\} \in U_t \mid I^T(m) = 3\}$ // triangles
  $B^{(2)} \leftarrow \{\{u,v,w\} \in U_t \mid I^T(m) = 2\}$ // wedges
  $B^{(1)} \leftarrow \{\{u,v,w\} \in U_t \mid I^T(m) = 1\}$ // edges
  $B^{(0)} \leftarrow U_t - B^{(1)} - B^{(2)} - B^{(3)}$ // empty
  // Ensure expected proportions
  $n^{(i)} \leftarrow \lfloor p_M^{(i)} \cdot |U_t| \rfloor, \ \forall i \in [1,2,3]$
  // triangles
  $B^{(3)}, B^{(2)} \leftarrow$ AddRemoveBuckets($B^{(3)}, B^{(2)}, n^{(3)}$)
  // wedges
  $B^{(2)}, B^{(1)} \leftarrow$ AddRemoveBuckets($B^{(2)}, B^{(1)}, n^{(2)}$)
  // 1-edge
  $B^{(1)}, B^{(0)} \leftarrow$ AddRemoveBuckets($B^{(1)}, B^{(0)}, n^{(1)}$)
  // Sampled motifs and motif types
  $M_t \leftarrow B^{(1)} \cup B^{(2)} \cup B^{(3)}$ // motifs with edges
  $M^T(\{u,v,w\}) \leftarrow i, \ \forall i \in [1,2,3], \ \forall \{u,v,w\} \in B^{(i)}$

Algorithm 5: `AddRemoveBuckets`

input : B^L, B^R triplet sets (left, right)
n_L desired proportion for B^L
output: B^L, B^R updated sets

begin
  if $|B^L| > n_L$ **then**
    // Send extra triplets to B^R
    $B^R \leftarrow B^R \cup \{B^L_{(j)}, \ \forall j \in [n_L, \ldots, |B^L|]\}$
    $B^L \leftarrow B^L - B^R$
  else if $|B^L| < n_L$ **then**
    // Add triplets from B^R
    $B^L \leftarrow B^L \cup \{B^R_{(j)}, \ \forall j \in [1, \cdots, n_L - |B^L|]\}$
    $B^R \leftarrow B^R - B^L$

Algorithm 6: `SampleMotifEdges`

input : M_t active triplets
M^T motif types
I^T initial type assignments
I^E initial edge placements
output: M_t^E motif edges

begin
  for $m = \{u,v,w\} \in M_t$ **do**
    $num \leftarrow \text{abs}(I^T(m) - M^T(m))$
    if $M^T(m) < I^T(m)$ **then** // add edges
      $choices \leftarrow \binom{\{u,v,w\}}{2} - I^E(m)$
      $M_t^E(m)$.append(RandomChoice($choices, num$))
    else if $M^T(m) > I^T(m)$ **then** // remove edges
      $choices \leftarrow \binom{\{u,v,w\}}{2} - I^E(m)$
      $M_t^E(m)$.remove(RandomChoice($choices, num$))

Algorithm 7: `SampleMotifTimestamps`

input : t timestep triplets become active
M_t sampled motifs
M^T motif types
$\lambda_M = (\lambda_M^{(1)}, \ldots, \lambda_M^{(3)})$ rates distribution
output: M_t^S motif timestamps

begin
  for $\{u,v,w\} \in M_t$ **do**
    $i \leftarrow M^T(\{u,v,w\})$ // motif type
    // sample inter-arrival rate
    $\beta_M(\{u,v,w\}) \sim Exp(\lambda_M^{(i)})$
    $\lambda_M(\{u,v,w\}) \leftarrow \frac{1}{\beta_M(\{u,v,w\})}$
    $prev \leftarrow t$ // first time motif can appear
    // sample inter-arrival time
    $next \sim Exp(\lambda_M(\{u,v,w\}))$
    while $prev + next < T$ **do**
      // save timestamp to list
      $M_t^S(\{u,v,w\})$.append($prev + next$)
      $prev \leftarrow prev + next$
      // sample next inter-arrival time
      $next \sim Exp(\lambda_M(\{u,v,w\}))$