Using Motif Transitions for Temporal Graph Generation

Penghang Liu
University at Buffalo
Buffalo, NY, USA
penghang@buffalo.edu

Ahmet Erdem Sarıyüce
University at Buffalo
Buffalo, NY, USA
erdem@buffalo.edu

ABSTRACT

Graph generative models are highly important for sharing surrogate data and benchmarking purposes. Real-world complex systems often exhibit dynamic nature, where the interactions among nodes change over time in the form of a temporal network. Most temporal network generation models extend the static graph generation models by incorporating temporality in the generation process. More recently, temporal motifs are used to generate temporal networks with better success. However, existing models are often restricted to a small set of predefined motif patterns due to the high computational cost of counting temporal motifs. In this work, we develop a practical temporal graph generator, Motif Transition Model (*MTM*), to generate synthetic temporal networks with realistic global and local features. Our key idea is modeling the arrival of new events as temporal motif transition processes. We first calculate the transition properties from the input graph and then simulate the motif transition processes based on the transition probabilities and transition rates. We demonstrate that our model consistently outperforms the baselines with respect to preserving various global and local temporal graph statistics and runtime performance.

CCS CONCEPTS

• **Theory of computation → Dynamic graph algorithms**; • **Computing methodologies → Network science**.

KEYWORDS

temporal networks, graph generative model, temporal motifs

ACM Reference Format:
Penghang Liu and Ahmet Erdem Sarıyüce. 2023. Using Motif Transitions for Temporal Graph Generation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3580305.3599540

1 INTRODUCTION

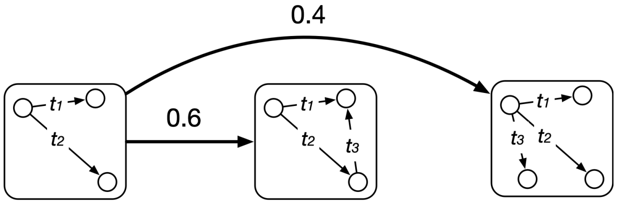


Figure 1: An example of motif transition. A wedge motif can have 0.6 probability of evolving into a triangle and 0.4 probability of becoming a 3-star.

Graphs are an effective tool to model real-world complex systems in various domains, such as communications, human interactions, financial activities, social relations, and protein interactions. One important problem is the generation of realistic synthetic networks from a given distribution of real-world networks [4–6]. Graph generators are necessary for enabling the sharing of surrogate data, such as computer network traffic or financial networks, and benchmarking studies, such as scalability and versatility tests [20].

Real-world complex systems often exhibit dynamic nature, where the interactions among nodes evolve over time. Edges are active only at certain points in time in temporal networks, such as financial transactions, communication networks, and face-to-face contacts. Temporal graph generation is relatively understudied and particularly more challenging than static graph generation due to the diverse and large nature of real-world temporal networks. Most temporal network generation models focus on reproducing the global characteristics of real-world networks. They often extend the static graph generation models by incorporating temporality in the generation process [10, 12, 13, 17]. Existing models lack a holistic approach that can consider the structural and temporal characteristics at the same time.

Temporal motifs, also known as higher-order structures, are an important building block in temporal networks [14, 21, 26, 32, 34, 38]. Temporal motif-based analysis has been used for many applications including cattle trade movements [3], editor interactions in Wikipedia [16], mobile communication networks [22, 24], and human interactions [46]. Temporal motifs are also a promising and effective tool for temporal graph generation. Purohit et al. introduced the Structural Temporal Modeling (*STM*) which computes the frequencies of a set of atomic motifs and places the motifs with the preferential attachment mechanism to generate temporal graphs [36]. Zeno et al. [44] modeled the active node behavior over time and generate three types of 3-node motifs (one edge, wedge, and triangle) with different arrival rates. However, these models suffer from the cost of counting temporal motifs which increases exponentially as the size of the motif increases. Thus, they are often limited to a custom set of motifs of limited size which are unable to capture the structural and temporal characteristics of the networks.

In this work, we develop a realistic and practical temporal graph generator. For a given input temporal network, our Motif Transition Model (*MTM*) generates a synthetic temporal network that preserves the global and local features of the input network. We introduce the motif transition to model the evolution of temporal

KDD '23, August 6–10, 2023, Long Beach, CA, USA.

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599540

motifs as a stochastic process. Figure 1 gives an example where a wedge motif can transition into two larger temporal motifs by adding new events. Our model first calculates the transition properties from the input graph and then simulates the stochastic motif transition processes based on the transition probabilities and transition rates. We compare the *MTM* against several state-of-the-art temporal graph generative models on real-world networks from various domains. We demonstrate that our model consistently outperforms the baselines with respect to preserving various global and local temporal graph statistics. Furthermore, our model is significantly more efficient than the existing baselines.

Our key contributions can be summarized as follows:

- We propose the Motif Transition Model, *MTM*, to generate a synthetic network through a stochastic process that accurately captures the global graph statistics and the temporal motif structures.
- We give algorithms to efficiently compute the motif transition properties and use those to generate the new graph. In particular, we do not count the motifs in the process which makes our model practical and scalable.
- We perform a comprehensive evaluation of our model against several baselines on various real-world networks. *MTM* outperforms the baselines in preserving the global graph characteristics and local temporal motif features of the input graph. *MTM* is also highly efficient and scalable to large networks.

2 RELATED WORK

Here we briefly summarize the previous work on the generative models for temporal networks and temporal network motifs.

2.1 Generative Models for Temporal Networks

Most of the previous studies on generative models for temporal networks are derived from the extension of static graph generation models [10, 12, 13]. Gauvin et al. [7] surveyed several random reference models for temporal networks many of which extend the Erdős-Renyi and configuration models to temporal networks. One of the most commonly used static graph generative models is the stochastic block model [2], which splits nodes into different groups and then places edges between them according to their categories. Several dynamic variants of the stochastic block model are developed [8, 11, 19, 30, 41–43, 45], in which the nodes can switch classes over time. Recently, Porter et al. [34] developed the temporal activity state block model *TASBM* to generate a temporal network with temporal motif structures. They divide the temporal graph into multiple time windows and compute the average in-event and out-event arrival rates for each node. Based on the event arrival rates between the activity groups, the model generates the events between each pair of nodes by a Poisson draw.

Inspired by the success of deep graph generative models, Zhou et al. [48] proposed the *TagGen* model to generate temporal graphs. They use a bi-level self-attention mechanism and assemble the temporal random walk sequences to form a temporal interaction network. However, their model converts the temporal graph into a few snapshots, hence the fine-grained timestamp information is lost during this process. The prohibitive time and space complexity also make their model impractical for large scale. Another line of work used higher-order and variable-order Markov chains to model the temporal sequence of edges in multiple k-th order pathways [33, 37]. Most existing temporal graph generative models do not consider the higher-order structures except temporal walks and pathways. However, many real-world events do not occur in traversal patterns, such as repetitive monthly auto-payment activities, which cannot be reproduced by the random walks or higher-order pathways. In our work, we propose an effective solution to model any potential correlation between two events that are topologically and temporally close to each other. Our model is also practical for large-scale networks.

2.2 Temporal Motifs

Higher-order temporal subgraph structures, i.e., temporal motifs, are an important property of temporal networks. Several approaches have been proposed to model temporal motifs [14, 21, 26, 32, 38]. Applications and use cases for temporal motifs are numerous. Jin et al. [15] proposed trend motifs to investigate the financial and protein networks with dynamic node weights. Zhao et al. [47] devised communication motifs to study the information propagation in human communication networks, such as call detail records (CDR) and Facebook wall post interactions. Zhang et al. [46] introduced motif-driven analysis for human interactions including phone messages, face-to-face interactions, and sexual contacts. Liu et al. studied patent oppositions and collaborations [27] and financial transaction networks [25] by using temporal motifs.

Regarding the temporal graph generative models, Purohit et al. used temporal motifs to propose Structural Temporal Modeling (*STM*) process to generate temporal networks [36]. They select a set of easy-to-compute atomic motifs, such as wedges, triangles, and squares, and for each type of atomic motif, the model calculates the independent motif frequencies (ITeM) which prohibits the overlaps between motifs [35]. Nodes and motifs in the output graph are generated based on the ITeM frequency and a preferential attachment function. Similarly, Zeno et al. [44] proposed the DYMOND model to generate dynamic networks. They consider three types of undirected static motifs (one edge, wedge, and triangle), and place motifs on active nodes with different arrival rates. However, they convert the temporal network into a sequence of snapshots and only model static motifs which do not contain any timestamp information. One limitation of these models is that they only select a limited set of temporal motifs (mostly wedges and triangles), which are not sufficient to capture more complex subgraph structures. Another drawback is that they rely on counting the number of temporal motifs and do not consider the correlations between temporal motifs of different sizes. Also, the complexity of counting temporal motifs increases exponentially as the motif size increases, thus most existing studies does not consider temporal motifs with more than three events. In our work, we use the full spectrum of temporal motifs and utilize the transitions among them to generate realistic synthetic temporal networks—we also do not count the motifs in the process which makes our model practical and scalable.

3 BACKGROUND

Here we provide the formal definition of temporal motifs, and introduce a notation method to denote different types of temporal motifs. The notations and symbols are summarized in Table 4 of Appendix A.

3.1 Temporal Motifs

We explore temporal networks which we represent as $G = (V, E)$. V is the set of nodes and E is the set of timestamped events. Each event $e_i \in E$ is a 3-tuple (u_i, v_i, t_i) that represents a directed relation from the source node u_i to the target node v_i at time t_i. The event set E is a time-ordered list of events such that $t_1 < t_2 < t_3 < \cdots < t_{|E|}$. Here we distinguish edges and events, where the edge (u, v) is the static projection of an event (u, v, t), and there may be multiple events occurring on the same edge at different times. We also define the static projection of G as $\overline{G} = \{(u_i, v_i) \,|\forall(u_i, v_i, t_i) \in G\}$.

Definition 3.1. (Temporal motif) *Given a temporal network $G = (V, E)$, an l-event temporal motif ($l \geq 2$), denoted by $M_i^l = (V', E')$, is a temporal subgraph in G such that*

- *$V' \subseteq V$, $E' \subseteq E$, $|E'| = l$,*
- *$\overline{M^l}$ is a (weakly-connected) subgraph, hence $2 \leq |V'| \leq l + 1$,*
- *Each event is connected to at least one of the previous events: $\{u'_{j+1}, v'_{j+1}\} \cap \{u'_1, v'_1, \ldots, u'_j, v'_j\} \neq \emptyset$ for any $j + 1 \leq l$ (i.e., the motif is a connected subgraph at every timestamp).*

For a given l, there are different types of temporal motifs in terms of connectivity and temporal structure, called as the motif spectrum. For instance, there are 6 types in the motif spectrum for M^2 (see Figure 2), 60 types for M^3 (see Figure 11 in Appendix C), and 888 types for M^4. We denote each type of temporal motif with a unique subscript such as M_i^2 for $1 \leq i \leq 6$, M_i^3 for $1 \leq i \leq 60$, and M_i^4 for $1 \leq i \leq 888$. We use $\{M_i^l\}$ to denote the set of instances of a motif M_i^l in G.

3.2 Motif Notation

To describe the large spectrum of temporal motifs (e.g., M_i^3 for $1 \leq i \leq 60$), we introduce a digit-based notation. We use $2l$ digits to denote a unique type of l-event temporal motif M_i^l. Each pair of digits, uv, denotes an event from the node represented by the first digit u to the node denoted by the second digit v. Digits start from zero and the digit for each node follows the chronological order of the node's appearance in the motif. The sequence of digit pairs also follows the chronological order of the events. The first two digits of any temporal motif are always `01`, to denote that the first event occurred from node `0` to node `1`. Figure 2 presents some examples. For instance, `0110` denotes a type of M^2 that consists of two opposite events occurring between two nodes (`01` and `10`). Similarly, `011202` represents a type of M^3 where the first event is from the black node (`0`) to the white node (`1`), the second one is from the white node (`1`) to the gray node (`2`), and the last event is from the black node (`0`) to the gray node (`2`). Based on the definition, each digit notation denotes one type of temporal motif, and each type of temporal motif corresponds to a unique digit notation.

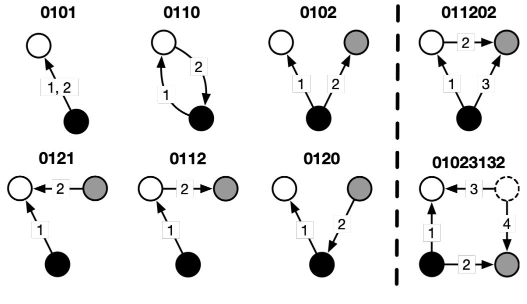


Figure 2: Six types of 2-event motifs (left) and an example of 3-event and 4-event motifs (right). We use $2l$ digits to denote a temporal motif with l events. Each event is given by a pair of digits, where the source node is the first and the target node is the second digit.

4 MOTIF TRANSITION MODEL

In this section, we introduce the motif transition model, *MTM*, to generate synthetic temporal networks with realistic global and local characteristics. Unlike the existing models that generate different types of temporal motifs independently over time [36, 44], the core idea of *MTM* is to model the evolution of temporal motifs as a stochastic process. For example, a 1-event motif can evolve into a wedge if there comes a new event that has one node in common, or a wedge motif can evolve into a triangle if there is a new event connecting the two nodes that are not directly connected. We generate temporal synthetic networks by simulating the evolution of temporal motifs.

4.1 Motif Transition Process

We use motif transition process to model the evolutions of temporal motifs. Here we givs the definition of motif transition and motif transition process.

Definition 4.1. (Motif transition) *In a given graph G, a motif $M_i^l = (V', E')$ transitions to a motif $M_j^{l+1} = (V'', E'')$ if there is a new event $e_{l+1} = (u_{l+1}, v_{l+1}, t_{l+1})$ such that*

- *$V'' = V' \cup \{u_{l+1}, v_{l+1}\}$ and $E'' = E' \cup e_{l+1}$,*
- *$\{u_{l+1}, v_{l+1}\} \cap V' \neq \emptyset$, i.e., the new event is adjacent to M_i^l,*
- *$t_{l+1} > t_l$, where t_l is the timestamp of the last event in M_i^l,*
- *M_i^l does not transition to another motif before e_{l+1} arrives, i.e, there does not exist an event $e^* = (u^*, v^*, t^*) \in G$ that satisfies all requirements above and $t^* < t_{l+1}$.*

We use $\mathcal{T}(M_i^l \rightarrow M_j^{l+1})$ to denote the transition, and $\{\mathcal{T}(M_i^l \rightarrow M_j^{l+1})\}$ to denote the set of instances of such transition in the input graph. We define the arrival time of the new event as the transition time: $\Delta_t = t_{l+1} - t_l$.

Figure 3 gives some examples of the motif transition. In our digit notation, the first motif in the transition is always a prefix of the second motif. For example, a single event (`01`) can transition into one of the six types of 2-event motifs by the arrival of a consecutive event. Similarly, a 2-event motif, say `0110`, can transition into a 3-event motif such as `011001`, `011002`.

Definition 4.2. (Motif transition process) *In a given graph G, we define the motif transition process as a sequence of motif transitions, $\mathcal{T}(M^1 \rightarrow \ldots \rightarrow M_i^l \rightarrow S)$, with respect to the* **transition size limit** *$l_{\max}$ and* **transition time limit** *δ. S denotes the stopping state. The motif transition process starts from 1-event motif and ends at M_i^l if either one of the following conditions is true:*

- *The size of the M_i^l is equal to the transition size limit, i.e., $l = l_{\max}$,*
- *Within the time window $(t_l, t_l + \delta]$, there does not exist a new event e_{l+1} to create the next transition $\mathcal{T}(M_i^l \rightarrow M_j^{l+1})$.*

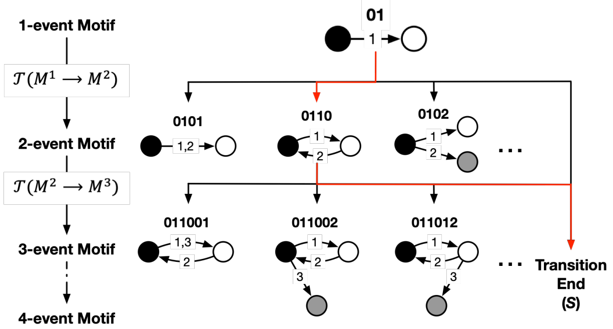


Figure 3: Examples of motif transition processes. S **indicates that the transition process ends and no more events can be added to the current motif. A transition process is sequence of motif transitions from an event** `01` **to the transition end** S**.**

Figure 4 gives examples of the motif transition process on a toy graph for $l_{\max} = 3$ and $\delta = 5s$. There is a motif transition process from 1s to 5s which stops as the size of the triangle motif is equal to the transition size limit $l_{\max}$. Another motif transition process starts at 7s and ends at 9s because there does not exist a new event in (9s, 14s] to create a new transition.

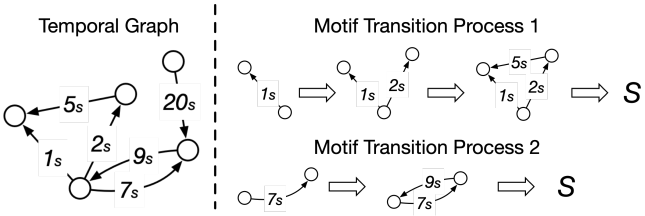


Figure 4: Motif transition processes for $l_{\max} = 3$ **and** $\delta = 5$**s.**

We categorize events in a temporal network to two classes: **cold events**, which are the first in a motif transition process, and **hot events**, which are subsequent events added in a motif transition process. We denote the set of cold events as *CE*. The first event in a temporal graph is always a cold event. Each cold event can only trigger one motif transition process, and each motif transition process only contains one cold event. Given the state of the current motif, the motif transition process captures the arrival of the new event, which can be modeled as a Markov process. Based on this idea, we develop a graph generative model to simulate temporal networks as a stochastic process of new event arrivals.

4.2 Graph Generative Model

We use motif transition processes to develop the motif transition model (*MTM*), a stochastic model to generate synthetic temporal networks. Figure 5 gives an overview of the *MTM* model. Our model takes a temporal graph as input and generates a synthetic network in two steps: we first identify the motif transition properties of the input graph, and then generate the synthetic network by simulating the motif transition processes.

4.2.1 Identifying motif transition properties. In this step, we calculate the following five properties from the original input graph, which are later used for simulating the motif transitions in the second step.

- **Degree distribution of cold events,** K_{CE}**.** We create the static projection of the cold events, denoted by $\overline{CE}$, such that $\overline{CE} = \{(u_i, v_i) \,|(u_i, v_i, t_i) \in CE\}$. We define K_{CE} as the list of in-degrees and out-degrees of the nodes in $\overline{CE}$. We use K_{CE} to generate the new cold events in the synthetic network.
- **Timestamps of cold events,** T_{CE}**.** We identify the timestamps of all the cold events in G. We use T_{CE} in generation of the cold events in the synthetic network.
- **Motif transition probabilities,** $\mathcal{P} = [P_1, P_2, \ldots, P_{|\mathbb{T}|}]$. We identify all motif transitions in the input graph and calculate the conditional probability of each type of motif transitions $\mathcal{T}(M_i^l \rightarrow M_j^{l+1})$:

$$P(M_j^{l+1}|M_i^l) = \frac{|\{\mathcal{T}(M_i^l \rightarrow M_j^{l+1})\}|}{|\{\mathcal{T}(M_i^l \rightarrow S)\}| + \sum_k |\{\mathcal{T}(M_i^l \rightarrow M_k^{l+1})\}|}$$

  where M_j^{l+1} is a type of $(l+1)$-event motif that M_i^l can transition into and $|\{\mathcal{T}(M_i^l \rightarrow M_j^{l+1})\}|$ is the number of corresponding transition instances. $|\{\mathcal{T}(M_i^l \rightarrow S)\}|$ is the number of transition processes that end at M_i^l, and $\sum_i |\{\mathcal{T}(M_i^l \rightarrow M_k^{l+1})\}|$ is the total number of all motif transition instances from M_i^l. $|\mathbb{T}|$ stands for the total number of transition types under the $l_{\max}$ limit. There are 6 types of $\mathcal{T}(M^1 \rightarrow M^2)$ transitions, 60 types of $\mathcal{T}(M^2 \rightarrow M^3)$ transitions, and 888 types of $\mathcal{T}(M^3 \rightarrow M^4)$ transitions. Threfore, $|\mathbb{T}| = 6$ when $l_{\max} = 2$, $|\mathbb{T}| = 66$ when $l_{\max} = 3$, and $|\mathbb{T}| = 954$ when $l_{\max} = 4$.
- **Motif transition rates,** $\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_{|\mathbb{T}|}]$**.** Since the motif transition is nothing but arrival of a new event, inspired by the previous works on stochastic temporal network models [29, 34], we use Poisson process to model the transition time Δ_t (i.e., the arrival time of the new event) for each type of motif transition. In particular, we set the transition rate (i.e., the arrival rate of the new event), denoted as $\lambda(M_i^l \rightarrow M_j^{l+1})$, of a motif transition to the average of the transition times of all the instances of that motif transition: $1/\text{mean}(\overrightarrow{\Delta_t})$.
- **Average number of edges in the motif transition process,** μ**.** For each motif transition process in the input graph $\mathcal{T}(M^1 \rightarrow \ldots \rightarrow M_i^l \rightarrow S)$, we construct the static projection of M_i^l and consider the number of edges in it. Then we calculate the average number of those.

4.2.2 Simulate motif transitions. In the second step, we generate the synthetic network using the motif transition properties captured in the first step. We utilize K_{CE} and T_{CE} to generate the cold events, and $\mathcal{P}$, Λ, μ to generate the hot events. We first employ the configuration model [31] to generate static edges from the degree distribution K_{CE}, and then apply weight-constrained link shuffling [7] to randomly assign the timestamps T_{CE} on the edges to create the cold events for the output graph, denoted by CE'.

After generating the cold events, CE', we process each in the chronological order to generate the hot events. For each cold event, we randomly generate a motif transition process using the transition probabilities of the input graph, $\mathcal{P}$. We store all the cold and hot events in E_{out}. We simulate the timestamps of the new events by the arrival times generated from the Poisson process

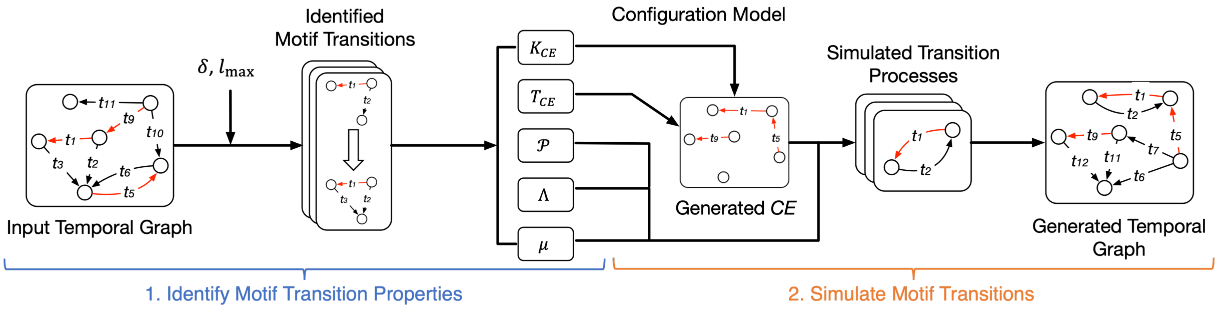


Figure 5: The proposed motif transition model (*MTM*).

Algorithm 1 MOTIF TRANSITION MODEL (MTM)

```
    Input: A temporal graph as an ordered list of events $E$, the transition
      time constraint $\delta$, the transition size constraint $l_{\max}$
    Output: A synthetic temporal graph as a list of events $E_{\text{out}}$
 1: $E_{\text{out}} \leftarrow \emptyset$
 2: $K_{CE}, T_{CE}, \mathcal{P}, \Lambda, \mu \leftarrow$ COMPUTE MOTIF TRANSITIONS($E$, $\delta$, $l_{\max}$)
 3: $CE' =$ CONFIGURATION MODEL($K_{CE}$, $T_{CE}$)
 4: for all $\{u, v, t\} \in CE'$ do
 5:     $E_{\text{out}} \leftarrow E_{\text{out}} \cup \{u, v, t\}$
 6:     $V' = [u, v]$, $M = 01$
 7:     while number of events in $M \leq l_{\max}$ do
 8:         $M' \leftarrow$ randomly choose by using $P(M'|M) \in \mathcal{P}$
 9:         if $M' = S$ then
10:             break
11:         digit$_{u'}$, digit$_{v'}$ $\leftarrow$ last two digits of $M'$        ▷ the new event
12:         if digit$_{u'} > |V'|$ then                                  ▷ request a new edge
13:             $v' = V'[\text{digit}_{v'}]$
14:             randomly choose $u'$ to form $(u', v')$ based on Equation 1
15:             $V'$.add($u'$)
16:         else if digit$_{v'} > |V'|$ then                             ▷ request a new edge
17:             $u' = V'[\text{digit}_{u'}]$
18:             randomly choose $v'$ to form $(u', v')$ based on Equation 1
19:             $V'$.add($v'$)
20:         else
21:             $u' = V'[\text{digit}_{u'}]$
22:             $v' = V'[\text{digit}_{v'}]$
23:         $t = t + \text{Pois}(\lambda(M \rightarrow M'))$
24:         $E_{\text{out}} \leftarrow E_{\text{out}} \cup \{u', v', t\}$
25:         $M = M'$
26: return $E_{\text{out}}$
```

Pois(λ). In particular, we generate the timestamp of a new event as $t_{l+1} = t_l + \Delta_t$, where $P(\Delta_t > x) = e^{-\lambda x}$. We continue adding new events to the transition process until it reaches the stopping state S. After all cold events have been processed, the model gives the event list E_{out} as the output graph.

A new event may create an edge that does not exist in the previous transition. For example, the transition from 01 to 0102 brings a new event which does not occur on an edge in the current motif. We consider two ways to select the edge (u', v') for the new event: (1) we randomly select an edge that is in the static projection of the current output event list, i.e., $(u', v') \in \overline{E_{\text{out}}}$; or (2) we randomly create a new edge that is not in $\overline{E_{\text{out}}}$, which will increase the size of $\overline{E_{\text{out}}}$ by one. We set the probability to create a new edge as

$$p((u', v') \notin \overline{E_{\text{out}}}) = \frac{|\overline{E}| - |\overline{CE'}|}{(\mu - 1)|\overline{CE'}|} \tag{1}$$

where $\overline{CE'}$ is its static projection of cold events for the output graph, $\overline{E}$ is the static projection of the input graph, and μ is the average number of edges in motif transition processes. Since each cold event will transition into a motif with μ edges on average, the transition processes will request $(\mu - 1)|\overline{CE'}|$ new edges in addition to the cold events. For each request we create a new edge based on the probability in Equation 1, which will fill the difference between the number of edges in the input graph and the cold events $|\overline{E}| - |\overline{CE'}|$.

4.2.3 MTM algorithm. Algorithm 1 provides the pseudocode of the motif transition model. We first calculate the motif transitions in Line 2 using Algorithm 2. Then we simulate the cold events in Line 3, and generate motif transitions from each cold event until the size reaches the transition size limit (Line 7) or reaches to the stopping state S (Line 9). At each step, we randomly select the next transition according to the calculated transition probabilities (Line 8), and generate the arrival time of the new event from Poisson distribution (Line 23). If the transition process requests a new edge, we randomly select it using Equation (1) (Line 12 to 19). The simulation process (Line 4 to 26) is a linear algorithm with $O(|CE| \cdot l_{\max})$ runtime complexity as the transitions of cold events are limited by $l_{\max}$.

Algorithm 2 shows the pseudocode for calculating the motif transitions. We maintain a list of active transitions, which are the motif transition processes that can be extended by a new event. For each event, we check all the active transitions to see if the event can be added to an existing transition (Line 15). If so, we record the transition count and the transition time (Lines 17 and 18). Note that if the event is added to an existing motif, it cannot be a cold event (Line 21). Before proceeding to the next event in the list, we update the set of active transitions (Line 20 and 25). After getting all the transition counts and transition times, we calculate the transition probabilities $\mathcal{P}$ and the transition rates Λ (Line 28 to 31). Note that Algorithm 2 does not explicitly count the motifs. Identifying the motif transitions (Line 6 to 25) takes $O(|E| \cdot |\mathcal{T}_{\text{active}}|)$ time. Since each cold event can only trigger one active transition, the number of active transitions, $|\mathcal{T}_{\text{active}}|$, at any step is no greater than the number of cold events, $|CE|$, in any $l_{\max} \cdot \delta$ time window. The average number of $|CE|$ in any time window is $|CE| \frac{l_{\max} \cdot \delta}{|T|}$, where $|T|$ is the timespan of the input data. Once the transitions are identified, calculating the $\mathcal{P}$ and Λ (Line 28 to 31)

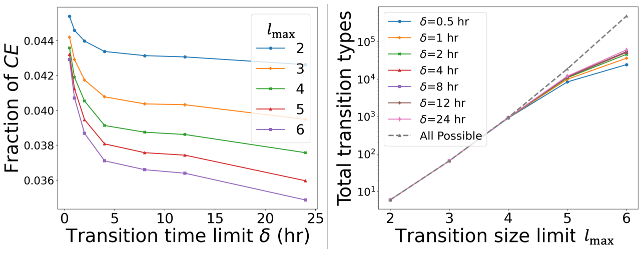
Algorithm 2 Compute Motif Transitions

```
Input: A temporal graph as an ordered list of events $E$, the transition time constraint $\delta$, the transition size constraint $l_{\max}$
Output: Degrees and timestamps of cold events $K_{CE}$ and $T_{CE}$, transition probabilities $\mathcal{P}$, transition rates $\Lambda$, average number of edges in the motif transition processes $\mu$
$CE \leftarrow \emptyset$                                    ▷ set of cold events
$L \leftarrow []$                   ▷ list of number of edges in motif transition processes
$\mathcal{T}_{\text{active}} \leftarrow []$                   ▷ list of active transition processes
$\overrightarrow{\Delta_t} \leftarrow [[]]$                        ▷ list of list of transition times
$C \leftarrow [0]$   ▷ list of # of instances for each motif transition (initially 0)
for all $(u, v, t) \in E$ do
    $ce$ = True                                     ▷ cold event flag
    for all $\mathcal{T} \in \mathcal{T}_{\text{active}}$ do
        $M \leftarrow$ the last motif of $\mathcal{T}$
        $t_{\max} \leftarrow$ the last timestamp of $M$
        $V' \leftarrow$ all the nodes in $M$
        if size of $M \geq l_{\max}$ or $t - t_{\max} > \delta$ then
            $\mathcal{T}_{\text{active}}$.remove ($\mathcal{T}$)              ▷ $\mathcal{T}$ ends
            $L$.add (number of edges in $M$)
        else if $\{u, v\} \cap V' \neq \emptyset$ then
            $M' \leftarrow M \cup (u, v, t)$                 ▷ add the new event
            $C[M \rightarrow M'] = C[M \rightarrow M'] + 1$
            $\overrightarrow{\Delta_t}[M \rightarrow M']$.add $(t - t_{\max})$
            $\mathcal{T}_{\text{active}}$.remove ($\mathcal{T}$)
            $\mathcal{T}_{\text{active}}$.add ($\mathcal{T} + M'$)  ▷ update the list of active transitions
            $ce$ = False
    if $ce$ = True then
        $CE \leftarrow CE \cup (u, v, t)$                   ▷ add to cold events
        $M' \leftarrow [(u, v, t)]$
    $\mathcal{T}_{\text{active}}$.add($M'$)              ▷ update the set of active transitions
for all $\mathcal{T} \in \mathcal{T}_{\text{active}}$ do
    $L$.add (number of edges in the last motif of $\mathcal{T}$)
for all $C[M \rightarrow M'] \in C$ do
    $P(M'|M) \leftarrow$ calculate transition probability using $C[M \rightarrow M']$
    $\lambda(M \rightarrow M') \leftarrow$ mean$(1/\Delta_t[M \rightarrow M'])$
    $P(M \rightarrow S) = 1 - \sum(P(M'|M))$
$\mathcal{P} \leftarrow$ list of $P$
$\Lambda \leftarrow$ list of $\lambda$
$K_{CE}, T_{CE} \leftarrow$ Get degrees and timestamps ($CE$)
$u \leftarrow$mean ($L$)
return $K_{CE}, T_{CE}, \mathcal{P}, \Lambda, \mu$
```

takes constant time as there are a fixed number of transition types. Therefore, the time complexity of Algorithm 2 is $O(|E| \cdot |CE| \frac{l_{\max} \cdot \delta}{|T|})$. In total, time complexity of *MTM* becomes $O(|E| \cdot |CE| \frac{l_{\max} \cdot \delta}{|T|}) + |E|)$. Note that in practice, the time window $l_{\max} \cdot \delta$ is 1,000 to 16,000 times smaller than the entire timespan of the network, $|T|$, and the fraction of cold events is less than 4.5% of all events. *MTM* requires constant space to store the transition properties, and $|\mathcal{T}_{\text{active}}|$ is a constant at each step. Therefore, the space complexity of *MTM* is $O(|CE| + |E|) = O(|E|)$.

5 EXPERIMENTS

In this section, we perform experiments to evaluate our model and compare its performance against several baseline models on various real-world networks. We first investigate the motif transition properties for different $l_{\max}$ and δ values. Then we evaluate the performance of *MTM* in three aspects: (1) the ability of preserving the global statistics of the original input graph, (2) the ability of preserving the temporal motifs structures in the original input graph, and (3) and the scalability with respect to the size of the input graph and the model parameters.



(a) Fraction of cold events **(b) Number of transition types**

Figure 6: The impact of the transition size limit ($l_{\max}$) and the transition time limit (δ). Figure 6a gives the fraction of the cold events in the input graph. Figure 6b shows the total number of transition types identified in the real-world network (straight lines) and all possible types of transitions (dashed line).

5.1 Setup

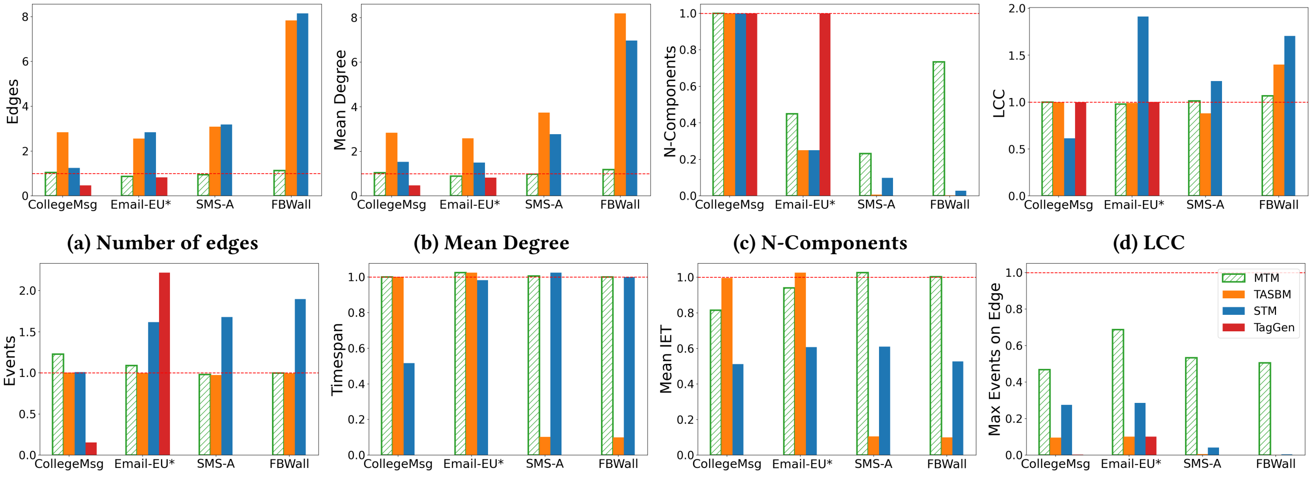
All experiments are performed on a Linux operating system running on a machine with Intel(R) Xeon(R) Gold 6130 CPU processor at 2.10 GHz with 128 GB memory. We also use NVIDIA V100 16GB GPU to run one of the baseline methods (*TagGen* model [48]). We implemented *MTM* in C++. **The code is available at https://github.com/erdemUB/KDD23-MTM.**

5.1.1 Datasets. We evaluate *MTM* on several real-world temporal networks from various domains, including `CollegeMsg`, `Email-EU`, `Email-EU*`, `FBWall`, `SuperUser`, and `StackOverflow`. The details and statistics are given in Appendix B.

5.1.2 Baselines. Different from the previous works that model the static motifs in the snapshots of a temporal network [44], our study focuses on truly temporal networks where each event has a unique timestamp. For this purpose, we select two two baseline models, *TASBM* [34] and *STM* [36], which process the temporal networks in their original forms. In addition, we consider the *TagGen* [48], a deep generative framework that uses temporal random walks.

We use the implementation of *TASBM*[1] (in C++) and *STM*[2] (in Apache Spark 2.3.0, GraphFrame 0.7.0, and Scala 2.11.8) provided by the authors. We set the number of time windows to 10 for *TASBM* as suggested by the paper. We set $\alpha = 1$ for *STM*, as we do not assume to have domain knowledge of the input graphs. Note that *TagGen* considers temporal networks as a sequence of snapshots by degrading the resolution of the original data. Based on the experiment setups described in the *TagGen* paper, we convert the `CollegeMsg` to 28 snapshots and the `Email-EU*` to 26 snapshots, while the original data has 58157 and 31750 unique timestamps respectively. We use the available implementation of *TagGen*[3] which creates the graph as a $|V| \times |V| \times |T|$ tensor. As the implementation

[1] https://github.com/aporter468/motifsanalyticalmodel
[2] https://github.com/temporal-graphs/STM
[3] https://github.com/davidchouzdw/TagGen



(a) Number of edges **(b) Mean Degree** **(c) N-Components** **(d) LCC**

(e) Number of events **(f) Timespan** **(g) Mean IET** **(h) Maximum events on edge**

Figure 7: Comparison of how the models preserve the global graph statistics. For each graph statistic, we show the ratio of the synthetic networks to the original input graphs. The horizontal dashed line indicates the value of the input graph (ratio=1). The *TagGen* does not generate results for `SMS-A` and `FBWall` because both datasets have more than 40K nodes which requires more than 384GB memory for allocation. It also gives networks with very low timespan and mean IET as it generates output graph as a few snapshots, which is why the red bars are not visible in (f), (g), and (h).

requires 384GB memory for an input graph with 40K nodes and 30 snapshots, we cannot execute the *TagGen* model on the other datasets with larger size.

5.2 Impact of Model Parameters

We first investigate the impact of the model parameters, $l_{\max}$ and δ. In particular, we compare the number of cold events and the number of transition types using different parameters. Figure 6 shows the results for `Email-EU`, and we observe similar patterns in other datasets. Using a larger $l_{\max}$ and δ allows more hot events to be added to the transitions, which leads to less cold events, as shown in Figure 6a. The decrease in the number of cold events slows down as the $l_{\max}$ and δ increase. The overall number of the cold events is small compared to the total number of events in the data, ranging from 3.5% to 4.5%. As the transition size limit ($l_{\max}$) increases, the possible types of transitions increases exponentially, as shown in Figure 6b. However, we do not identify many higher-order transitions in the real-world networks when using a large $l_{\max}$ value. For example, there are more than 466K types of possible $\mathcal{T}(M_i^5 \rightarrow M_j^6)$ transitions, but we only identify 48K in the `Email-EU` data. Changing the transition time limit, δ, does not have a strong impact on the total number of transition types. Based on our observations, we set $l_{\max} = 4$ and $\delta = 1$ hour as we achieve less benefits for larger parameters.

5.3 Preserving Global Graph Properties

Here we examine how well the *MTM* preserves the general graph statistics. For each dataset we generate 10 synthetic networks using the *TASBM*, *STM*, and *MTM*, and take the average values for each model. While each generated output is a completely different temporal graph, overall we do not observe significant variance between synthetic graphs generated by the same model. Inspired by the previous models [34, 36, 48], we evaluate four structural global graph metrics: the number of edges, the mean degree (sum of in- and out-degree), the number of connected components (N-Components), and the size of the largest connected components (LCC). Besides, we consider four temporal graph statistics: the number of events, the entire timespan of the network, the mean inter-event time, and the maximum number of events on an edge. These metrics are also examined in the previous studies on temporal networks [18, 28, 29, 39]. Figure 7 presents the ratio of synthetic graph statistics (bars) to the input graphs (dashed red line). As mentioned in the previous section, the *TagGen* model does not apply to `SMS-A` and `FBWall` which has more than 40k nodes.

Overall, *MTM* performs the best on preserving the global graph statistics of the input graphs. Except the number of components and the maximum events per edge metrics, *MTM* gives synthetic networks with less than 5% difference than the original graph. *TASBM* and *STM* yield synthetic networks with 2 to 8 times more edges than the original graphs, whereas the *TagGen* gives 50% less edges and degrees for the `CollegeMsg` data. While our model gives more accurate number of components than *TASBM* and *STM*, the *TagGen* performs the best. We also observe that *STM* tends to overamplify the size of the largest connected component.

Regarding the temporal graph statistics, *MTM* yields synthetic networks with more accurate characteristics than the baseline models. The error of the mean IET (inter-event time) is less than 20% for *MTM*. Although all the models perform poor in preserving the maximum number of events on edges, *MTM* gives synthetic networks with maximum events on edge significantly closer to the original graph. The reason is that *TASBM* generates events between two nodes purely based on the activity level and the arrival rates, without considering the correlations between the events on the same edge. *STM* only captures the repetitions of events through `0101` motifs. *TagGen* processes the original data as snapshot sequences, thus cannot capture the temporal characteristics. Our model, *MTM*, on the other hand, considers the complete motif spectrum under the given event limits, which enables to capture the repetitions in any step of the transition process.

We also examine the performance of certain distributions using the Kolmogorov-Smirnov (KS) test. In particular, we consider the in-degree, out-degree, inter-event time (IET), and timestamp distributions, and calculate the two sample KS test between the distributions in the original data and in the synthetic networks. Table 1 shows the average KS statistics (lower is better). We observe that *MTM* consistently outperforms the baseline models for all distributions over all datasets, except the IET of SMS-A.

5.4 Preserving Local Temporal Motif Statistics

In this part, we evaluate how well the *MTM* preserves the temporal motif statistics in the real-world networks compare to the baselines. For each dataset, we generate 10 synthetic networks using *MTM* (with $l_{\max} = 4$ and $\delta = 1$ hour) and the baseline models. We exclude *TagGen* model here because it generates output graphs as a sequence of static snapshots which does not contain temporal motif structures. We consider the inter-event time constraint Δ_C for counting temporal motifs, which requires that the time difference between each pair of consecutive events in the motif is less than Δ_C. Note that using a larger Δ_C threshold allows to discover more temporal motifs. However, a large Δ_C has less power to control the relevance between consecutive events in the motif, and increases the computation cost. Given that the mean inter-event time in all datasets is less than 600 seconds (see Table 5), we set $\Delta_C = 1$ hour and compute the counts of all 2-event, 3-event, and 4-event temporal motifs. We compare the number of motifs in the synthetic networks to the original input. We measure the difference by the mean squared relative error (MSRE),

$$MSRE = \frac{1}{r}\sum_{i}^{r}\left(\frac{|\{M_{G_i}\}| - |\{M_{\text{original}}\}|}{|\{M_{G_i}\}|}\right)^2,$$

where $|\{M_{G_i}\}|$ is the number of motif instances identified in the i-th generated synthetic network, $|\{M_{\text{original}}\}|$ is the number of motif instances in the original network, and $r = 10$ is the number of synthetic networks generated.

Table 1: Kolmogorov-Smirnov (KS) statistics to compare the distributions in original and synthetic networks (lower is better).

Data	Model	In-degree	Out-degree	IET	Timestamp
CollegeMsg	*TASBM*	0.311	0.269	0.435	1.000
	STM	0.261	0.398	0.525	1.000
	TagGen	0.224	0.283	0.984	1.000
	MTM	**0.075**	**0.195**	**0.096**	**0.078**
Email-Eu*	*TASBM*	0.918	0.925	0.408	0.019
	STM	0.722	0.732	0.242	0.480
	TagGen	0.463	0.463	0.736	1.000
	MTM	**0.113**	**0.080**	**0.121**	**0.011**
SMS-A	*TASBM*	0.687	0.631	**0.123**	0.929
	STM	0.668	0.653	0.362	0.067
	MTM	**0.096**	**0.225**	0.168	**0.003**
FBWall	*TASBM*	0.231	0.202	0.445	1.000
	STM	0.431	0.429	0.513	1.000
	MTM	**0.030**	**0.054**	**0.028**	**0.008**

Table 2: The MSRE results for temporal motif counts.

Data	Model	2-event	3-event	4-event
CollegeMsg	*TASBM*	41.916	1214.121	17086.425
	STM	296.274	3268.675	7704.935
	MTM	**0.004**	**0.001**	**0.035**
Email-Eu*	*TASBM*	0.634	2.885	5.852
	STM	2.457	6.333	6.046
	MTM	**0.070**	**0.152**	**0.230**
SMS-A	*TASBM*	8.946	58.271	152.080
	STM	515.802	3655.160	4420.164
	MTM	**0.0001**	**0.010**	**0.131**
FBWall	*TASBM*	3.605	34.467	171.514
	STM	1569.383	10691.863	11694.559
	MTM	**0.0001**	**0.021**	**0.467**

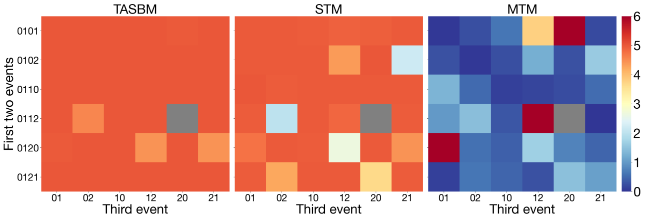
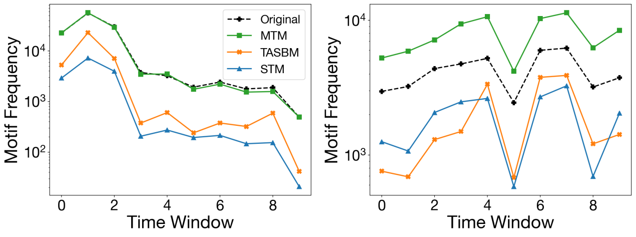


Figure 8: The MSRE results (color-coded) for all 3-node 3-event motifs in `CollegeMsg` network. Each block represents a type of motif; the row shows the notation of the first two events and the column shows the third event. Note that there are no `011220` motifs in the original network thus it does not have a valid MSRE value.

5.4.1 *Total temporal motif counts.* Table 2 shows the MSRE results based on the total number of 2-event, 3-event, and 4-event motifs. *MTM* significantly outperforms the baseline models across all the datasets. *TASBM* and *STM* cannot preserve the 4-event motifs in `CollegeMsg` and `FBWall`. *MTM* generates synthetic networks with very similar motif counts where the MSRE is less than 0.5. Note that *TASBM* splits the input data into different time windows and generate synthetic network for each slice independently. Therefore, it achieves better MSRE for large datasets such as `FBWall` when compared to *STM*. The MSRE of 3-event motifs is usually 10 times larger than the 2-event motifs and the error is even larger for the 4-event motifs. Larger size motifs are more complex and difficult to reproduce through generative models.

5.4.2 *Motif spectrum counts.* Next, we dive into the motif spectrums and measure the MSRE for each type of motifs. Figure 8 gives the MSRE results for all 3-node 3-event motifs in the synthetic `CollegeMsg` networks. Overall, our model gives 5 to 10 times smaller MSRE results than the baseline models. *TASBM* assigns nodes to different activity groups and then generates the events between groups with different arrival rates. Since it does not utilize temporal motifs for graph generation, the MSRE are high for all types of motifs. *STM* selects the triangle motifs (`011202`, `012012`, `010221`, etc.) as a part of the atomic motif patterns, hence it yields smaller MSRE for those. However, it cannot reproduce the distribution of other types of temporal motifs, thus yields high MSRE for 2-event, 3-event, and 4-event motifs in total.



(a) `CollegeMsg`

(b) `Email-EU*`

Figure 9: The trend of the total number of temporal motifs over time in the original input and the synthetic networks.

5.4.3 *Distribution of motif counts over time.* Lastly, we investigate the distributions of the temporal motifs in the synthetic and real-world networks over time. We split each dataset into 10 equal size time windows, and generate synthetic networks using the events in each interval. We calculate the total number of 2-event, 3-event, and 4-event motifs in each time window and show the trends in Figure 9 and Figure 12 of Appendix D. Our model accurately simulates the trends in temporal motif counts over the time. The baseline models capture the trends to some extent but do not fit well to the actual numbers of temporal motifs, especially for the time windows with less counts of temporal motifs. For example, the last time window of the `CollegeMsg` network has only 548 events (1% of the entire data), but still contains a significant number of temporal motifs. The generative power of the baseline models degrades significantly in this window, especially for the *STM* which does not generate any 3-event and 4-event motifs. Our model is robust to the changes in event density over time.

5.5 Runtime Analysis

Here we examine the runtime performance of the *MTM*. We first compare our model with baselines on all datasets. Note that we use a GPU to run *TagGen* model. Table 3 shows the average runtime of the 10 experiments. We also give the runtime for motif counting to show how a hypothetical model that counts motifs would compare. Our model takes significantly less time when compared to the baselines. The runtime of both baseline models increases drastically as the size of the input data increases. For larger datasets such as the `FBWall`, our model is 391 and 84 times faster than the *TASBM* and *STM*, respectively. In addition, our model is up to 231 times faster than motif counting computation. This shows one of the key benefits of our model: we do not explicitly count the temporal motifs, instead we only compute the motif transition probabilities.

We also evaluate the runtime of *MTM* with respect to the two model parameters, $l_{\max}$ and δ. We select two large datasets (`Super User` and `StackOverflow`) with more than one million events, and measure the average runtime for identifying motif transition properties (step 1) and simulating motif transitions (step 2), respectively. To explore the impact of $l_{\max}$, we set $\delta = 1$ hour and change $l_{\max}$ from 2 to 6. Similarly, we set $l_{\max} = 4$ and calculate the average runtime with $\delta = [0.5, 1, 2, 4, 8, 12, 24]$ hours. Figure 10 shows the results. We observe that the runtimes increase gradually as the two parameters increase. However, the increase is less steep when $l_{\max} > 4$ and $\delta > 8$. One possible reason is that the higher-order temporal motifs are less common in real world networks, hence the additional cost of identifying motif transitions becomes less as the transition limits increase. Another reason is that larger $l_{\max}$ and δ values allow more events to be added to the transition processes. As a result, the number of cold events decreases, so as the number of active transition processes at each time step. This reduces the cost of identifying temporal motif transitions.

Table 3: The average runtime (seconds) for the generative models. We also show the number of events ($|E|$) for each dataset.

Data	`Email-EU*`	`CollegeMsg`	`SMS-A`	`FBWall`
$\lvert E \rvert$	43035	59835	548182	876933
TASBM	24.7	36.8	11454.7	22100.9
STM	146.6	870.2	2565.1	4745.9
TagGen	2147.4	15919.8	N/A	N/A
Motif Counting	163.2	294.7	37957.7	1565.7
MTM	**2.5**	**2.8**	**163.9**	**56.5**

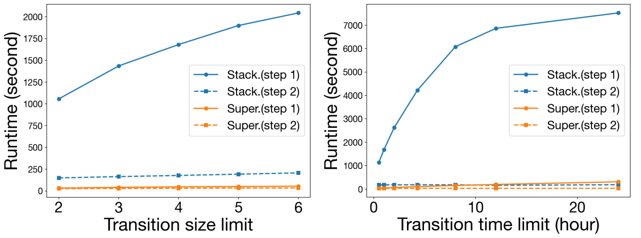


Figure 10: The average runtime of *MTM* with respect to the transition size limit $l_{\max}$ (left) and the transition time limit δ (right).

6 CONCLUSION

In this paper, we propose the *MTM* model to generate a temporal network that preserves the global and local features of an input graph. Our model first calculates five motif transition properties from the input graph, and then generates the synthetic networks by simulating the motif transition processes. We evaluate the performance of our model on seven datasets from various domains. The experimental results show that our model is able to preserve the structural and temporal characteristics of the input graph, such as the size of the network, the average in- and out-degree, and the mean inter-event time. Our model shows superior performance on reproducing the temporal motif spectrums, even for the higher-order temporal motifs in large networks. Last, but not least, our model 117 times faster (on average) than the baselines.

One potential opportunity for future research is developing the real-world applications of the motif transition models, such as generating temporal network with specified motif spectrums or detecting structural and temporal anomalies. Another possible direction is to consider other distributions to model the motif transition times, such as the temporal hawk process [9] and self-exciting cascading Poisson process [28]. It would also be interesting to explore other graph characteristics, such as the time-respecting paths and community structures, in the generated synthetic graphs and compare against the input network.

ACKNOWLEDGMENTS

This research was supported by NSF awards OAC-2107089 and IIS-2236789, and used resources from the Center for Computational Research at the University at Buffalo [1].

REFERENCES

[1] Center for Computational Research, University at Buffalo, 2021. http://hdl.handle.net/10477/79221.

[2] Airoldi, E. M., Blei, D., Fienberg, S., and Xing, E. Mixed membership stochastic blockmodels. *Advances in neural information processing systems 21* (2008).

[3] Bajardi, P., Barrat, A., Natale, F., Savini, L., and Colizza, V. Dynamical patterns of cattle trade movements. *PLoS ONE 6*, 5 (2011), e19869.

[4] Chakrabarti, D., and Faloutsos, C. Graph mining: Laws, generators, and algorithms. *ACM computing surveys (CSUR) 38*, 1 (2006), 2–es.

[5] Chung, F., and Lu, L. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences 99*, 25 (2002), 15879–15882.

[6] Erdős, P., and Rényi, A. On random graphs i. *Publicationes mathematicae 6*, 1 (1959), 290–297.

[7] Gauvin, L., Génois, M., Karsai, M., Kivelä, M., Takaguchi, T., Valdano, E., and Vestergaard, C. L. Randomized reference models for temporal networks. *SIAM Review 64*, 4 (2022), 763–830.

[8] Ghasemian, A., Zhang, P., Clauset, A., Moore, C., and Peel, L. Detectability thresholds and optimal algorithms for community structure in dynamic networks. *Physical Review X 6*, 3 (2016), 031005.

[9] Hawkes, A. G., and Oakes, D. A cluster process representation of a self-exciting process. *Journal of applied probability 11*, 3 (1974), 493–503.

[10] Hill, S. A., and Braha, D. Dynamic model of time-dependent complex networks. *Physical Review E 82*, 4 (2010), 046105.

[11] Ho, Q., Song, L., and Xing, E. Evolving cluster mixed-membership blockmodel for time-evolving networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (2011), JMLR Workshop and Conference Proceedings, pp. 342–350.

[12] Holme, P. Epidemiologically optimal static networks from temporal network data. *PLoS computational biology 9*, 7 (2013), e1003142.

[13] Holme, P., and Saramäki, J. Temporal networks. *Physics Rep. 519*, 3 (2012), 97–125.

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

[15] Jin, R., McCallen, S., and Almaas, E. Trend motif: A graph mining approach for analysis of dynamic complex networks. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)* (2007), IEEE, pp. 541–546.

[16] Jurgens, D., and Lu, T.-C. Temporal motifs reveal the dynamics of editor interactions in wikipedia. In *Proceedings of the International AAAI Conference on Web and Social Media* (2012), vol. 6, pp. 162–169.

[17] Karrer, B., and Newman, M. E. Random graph models for directed acyclic networks. *Physical Review E 80*, 4 (2009), 046110.

[18] Karsai, M., Kaski, K., Barabási, A.-L., and Kertész, J. Universal features of correlated bursty behaviour. *Scientific reports 2*, 1 (2012), 397.

[19] Kim, M., and Leskovec, J. Nonparametric multi-group membership model for dynamic networks. *Advances in neural information processing systems 26* (2013).

[20] Kolda, T. G., Pinar, A., Plantenga, T., and Seshadhri, C. A scalable generative graph model with community structure. *SIAM Journal on Scientific Computing 36*, 5 (2014), C424–C452.

[21] Kovanen, L., Karsai, M., Kaski, K., Kertész, J., and Saramäki, J. Temporal motifs in time-dependent networks. *Journal of Statistical Mechanics 2011*, 11 (2011), P11005.

[22] Kovanen, L., Kaski, K., Kertész, J., and Saramäki, J. Temporal motifs reveal homophily, gender-specific patterns, and group talk in call sequences. *Proceedings of the National Academy of Sciences 110*, 45 (2013), 18070–18075.

[23] Leskovec, J., and Krevl, A. SNAP Datasets, June 2014.

[24] Li, M.-X., Palchykov, V., Jiang, Z.-Q., Kaski, K., Kertész, J., Miccichè, S., Tumminello, M., Zhou, W.-X., and Mantegna, R. N. Statistically validated mobile communication networks: The evolution of motifs in european and chinese data. *New Journal of Physics 16*, 8 (2014), 083038.

[25] Liu, P., Acharyya, R., Tillman, R. E., Kimura, S., Masuda, N., and Sariyuce, A. E. Temporal motifs for financial networks: A study on mercari, jpmc, and venmo platforms. https://arxiv.org/abs/2301.07791.

[26] Liu, P., Guarrasi, V., and Sariyuce, A. E. Temporal network motifs: Models, limitations, evaluation. *IEEE Transactions on Knowledge and Data Engineering 35*, 1 (2023), 945–957.

[27] Liu, P., Masuda, N., Kito, T., and Sarıyüce, A. E. Temporal motifs in patent opposition and collaboration networks. *Scientific Reports 12*, 1917 (2022) (2022).

[28] Malmgren, R. D., Stouffer, D. B., Motter, A. E., and Amaral, L. A. A poissonian explanation for heavy tails in e-mail communication. *Proceedings of the National Academy of Sciences 105*, 47 (2008), 18153–18158.

[29] Masuda, N., and Lambiotte, R. *A Guide to Temporal Networks, Second Edition.* World Scientific, Singapore, 2020.

[30] Matias, C., and Miele, V. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 79*, 4 (2017), 1119–1141.

[31] Newman, M. E., Strogatz, S. H., and Watts, D. J. Random graphs with arbitrary degree distributions and their applications. *Physical review E 64*, 2 (2001), 026118.

[32] Paranjape, A., Benson, A. R., and Leskovec, J. Motifs in temporal networks. In *ACM Conf. on Web Search and Data Mining* (2017), ACM, pp. 601–610.

[33] Peixoto, T. P., and Rosvall, M. Modelling sequences and temporal networks with dynamic community structures. *Nature communications 8*, 1 (2017), 582.

[34] Porter, A., Mirzasoleiman, B., and Leskovec, J. Analytical models for motifs in temporal networks. In *Companion Proceedings of the Web Conference 2022* (2022), pp. 903–909.

[35] Purohit, S., Chin, G., and Holder, L. B. Item: Independent temporal motifs to summarize and compare temporal networks. *Intelligent Data Analysis 26*, 4 (2022), 1071–1096.

[36] Purohit, S., Holder, L. B., and Chin, G. Temporal graph generation based on a distribution of temporal motifs. In *Proceedings of the 14th International Workshop on Mining and Learning with Graphs* (2018), vol. 7.

[37] 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* (2017), pp. 1037–1046.

[38] Song, C., Ge, T., Chen, C., and Wang, J. Event pattern matching over graph streams. *Proc. of the VLDB Endowment 8*, 4 (2014), 413–424.

[39] Vázquez, A., Oliveira, J. G., Dezsö, Z., Goh, K.-I., Kondor, I., and Barabási, A.-L. Modeling bursts and heavy tails in human dynamics. *Physical Review E 73*, 3 (2006), 036127.

[40] Viswanath, B., Mislove, A., Cha, M., and Gummadi, K. P. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)* (August 2009).

[41] Xing, E. P., Fu, W., and Song, L. A state-space mixed membership blockmodel for dynamic network tomography. *The Annals of Applied Statistics 4*, 2 (2010), 535–566.

[42] Xu, K. S., and Hero, A. O. Dynamic stochastic blockmodels: Statistical models for time-evolving networks. In *International conference on social computing, behavioral-cultural modeling, and prediction* (2013), Springer, pp. 201–210.

[43] Yang, T., Chi, Y., Zhu, S., Gong, Y., and Jin, R. Detecting communities and their evolutions in dynamic social networks—a bayesian approach. *Machine learning 82*, 2 (2011), 157–189.

[44] Zeno, G., La Fond, T., and Neville, J. Dymond: Dynamic motif-nodes network generative model. In *Proceedings of the Web Conference 2021* (2021), pp. 718–729.

[45] Zhang, X., Moore, C., and Newman, M. E. Random graph models for dynamic networks. *The European Physical Journal B 90*, 10 (2017), 1–14.

[46] Zhang, Y.-Q., Li, X., Xu, J., and Vasilakos, A. V. Human interactive patterns in temporal networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems 45*, 2 (2015), 214–222.

[47] Zhao, Q., Tian, Y., He, Q., Oliver, N., Jin, R., and Lee, W.-C. Communication motifs: a tool to characterize social communications. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (2010), pp. 1645–1648.

[48] Zhou, D., Zheng, L., Han, J., and He, J. A data-driven graph generative model for temporal interaction networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2020), pp. 401–411.

A NOTATIONS

Table 4: Notation of symbols

Symbol	Description
$G = (V, E)$	temporal graph
$\overline{G} = (V, \overline{E})$	the static projection of G
M_i^l	temporal motif with l events
$\mathcal{T}(M_i^l \to M_j^{l+1})$	motif transition from M_i^l to M_j^{l+1}
$\mathcal{T}(M^1 \to \ldots \to S)$	motif transition process
$P(M_j^{l+1} \mid M_i^l)$	transition probability
$\lambda(M_i^l \to M_i^{l+1})$	transition rate
Δ_t	transition time
CE	set of cold events
K_{CE}	degrees of cold events
T_{CE}	timestamps of cold events
μ	average number of edges in motif transition processes
$l_{\max}$	transition size limit
δ	transition time limit
$\lvert\mathbb{T}\rvert$	total number of transition types
$\lvert T\rvert$	the timespan of the input graph

B DATA STATISTICS

Table 5: Statistics of the temporal network datasets.

Name	Nodes	Edges	Events	Timespan (days)	mean IET (sec.)
CollegeMsg	1.90K	20.3K	59.8K	193	273.1
Email-EU	986	24.9K	332K	803	209.0
Email-EU*	80	1184	43.0K	500	514.2
SMS-A	44.4K	69.0K	548K	338	53.3
FBWall	47.0K	274K	877K	1560	152.9
SuperUser	194K	925K	1.44M	2773	166.0
StackOverflow	260K	4.15M	6.35M	886	12.0

We use several real-world temporal networks from various domains. Table 5 give the statistics. In addition to the number of nodes, edges, and events, we give timespan and the mean inter-event time (i.e., average of the time intervals between all pairs of consecutive events) for each dataset. CollegeMsg [23] and SMS-A are phone message networks, in which an event (u, v, t) represents a message sent from u to v at time t. Email-EU is the emails between members of a European research institution [23], in which an event (u, v, t) indicates an email sent from person u to person v at time t (we also include the Email-EU* network used by [34], which contains 80 densely connected nodes in largest connected component of the original Email-EU data). FBWall contains the posts between users on Facebook in the New Orleans region [40], where an event (u, v, t) denotes user u posted on the user v's wall at time t. SuperUser and StackOverflow are the interaction networks from stack exchange websites [23], where an event (u, v, t) stands for user u posted an answer/comment on user v's question/answer at time t. The time resolution of all the networks is one second.

C ALL 3-EVENT MOTIFS

Here we list all 60 types of 3-event temporal motifs in Figure 11.

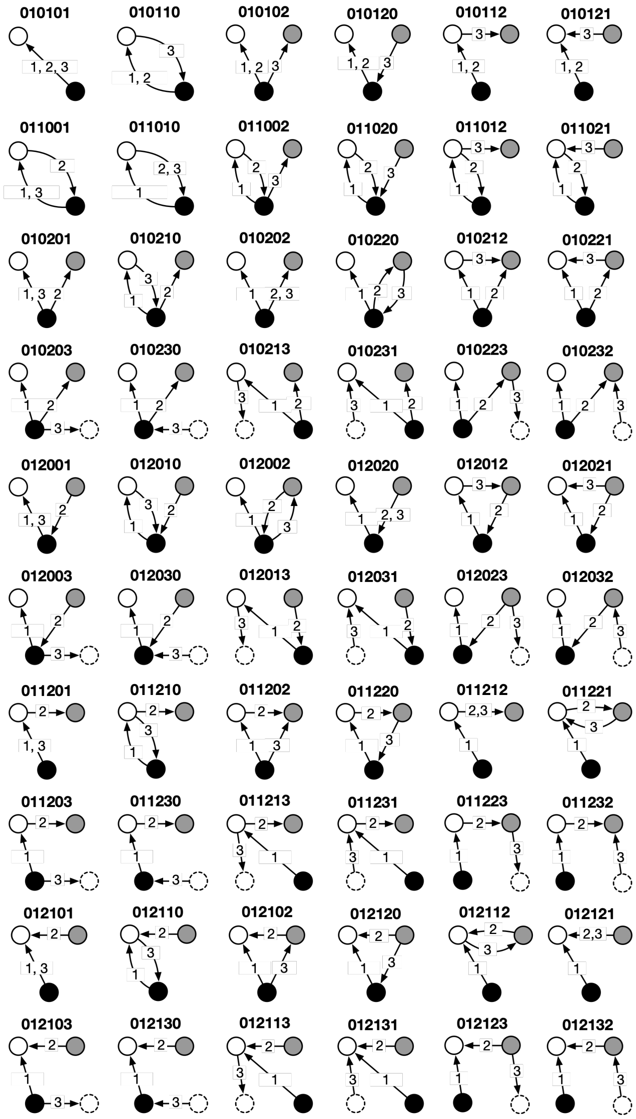
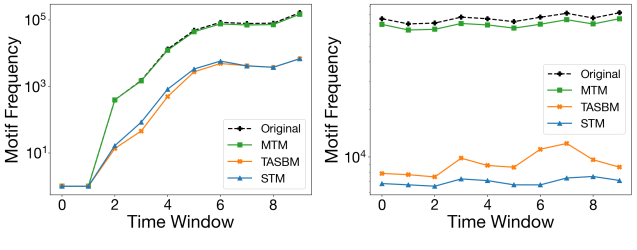


Figure 11: All 60 types of 3-event motifs.

D ADDITIONAL RESULTS



(a) FBWall

(b) SMS-A

Figure 12: The trend of the total number of temporal motifs over time in the original input and the synthetic networks.