Supplementary Material: Random Walk based Conditional Generative Model for Temporal Networks with Attributes

Stratis Limnios
The Alan Turing Institute
London, UK
slimnios@turing.ac.uk

Andrew Elliott
Department of Mathematics and Statistics
University of Glasgow &
The Alan Turing Institute, London, UK
Andrew.Elliott@glasgow.ac.uk

Mihai Cucuringu
Department of Statistics & Mathematical Institute
University of Oxford &
The Alan Turing Institute, London, UK
mihai.cucuringu@stats.ox.ac.uk

Gesine Reinert
Department of Statistics
University of Oxford &
The Alan Turing Institute, London, UK
reinert@stats.ox.ac.uk

The supplementary materials is organized in two sections. The first section gives the detailed low rank approximation we use for the generation in section 1. Then we present the remaining experiments conducted on the benchmark datasets in section 2.

1 Dimensionality Reduction and Scaling

As mentioned in the paper one of the drawbacks of the CTWALK pipeline is the size of the resulting tensor fed as input to the bi-level multi-head attention network. Coupling that with the CTGAN discriminator calls for scaling down this tensor. To be more specific, since each node at each time point is assigned a different embedding of size e_s, being the concatenation of the node Deepwalk embedding and its w_k attributes, the overall tensor is of size $n \times T \times e_s$ where n is the number of nodes and T. If a node does not exist at a given time point, then a random embedding is generated using a normal distribution with mean 0 and variance $\frac{1}{|T|}$.

An additional complication arises from a mismatch between CTGAN generator output type and TAGGEN input type. While the TAGGEN discriminator requires a node/timepoint embedding for each particular position in the random walk, the CTGAN generator gives a probability distribution over each of the constituent parts of the embedding. Thus, to square this circle, we consider the expected embedding when choosing a time point $\tau \in \{1, \ldots, T\}$ and, independently, a node $n \in \mathcal{V}$,

$$E[X] = \sum_t \sum_n P(\tau = t) P(N = n) T(t, n), \tag{1}$$

where $T(t, n) \in R^{e_s}$ is the embedding at each time t and node n.

Therefore, we propose the following solution using parallel factor analysis, a form of PCA/SVD that generalizes to arbitrary tensors. It uses a rank-r approximation for the tensor

$$T(t, n, i) \approx \sum_r a_{tr} b_{nr} d_{ir}, \tag{2}$$

where $T(t, n, i)$ is the i^{th} entry of the embedding $T(t, n, i)$, and r indexes the low rank approximation. Putting everything together, we arrive at

$$E[X_i] \quad = \quad \sum_t \sum_n P(\tau = t) P(N = n) T(t, n, i)$$

NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research.

$$
\begin{aligned}
&\approx \sum_t \sum_n P(\tau = t) P(N = n) \sum_r a_{tr} b_{nr} d_{ir} \\
&\approx \sum_r \left(\sum_t P(\tau = t) a_{tr} \right) \left(\sum_n P(N = n) b_{nr} \right) d_{ir}.
\end{aligned}
$$

This representation is very efficient as:

- The two inner summations can be re-written as matrix multiplications,
- they can be combined using element-wise products,
- the final summation can be written as a final matrix multiplication.

Hence, using this low-rank decomposition we obtain a method that can be applied to largerdata sets.

2 Additional Results

In this section we provide the remaining results of the experiments that were mentioned in the Experiments section of the main paper. Indeed in figure 1 we have the attribute comparison for the cycling dataset and figure 2 presents the results of the time series related statistics over the three benchmark datasets. The experiments were conducted using a machine with 32Gb of RAM, an Intel Core i7-10700K processor and a Titan XP Graphics card with 12Gb of graphic memory.

We observe that CTWALK performs as well if not better than CTGAN for attribute generation on the cycling dataset as emphasized by figure 1. Indeed we see that it manages to get the end station rankings closer to the original dataset ones, as well as having closer start hour distribution and ride duration.

In figure 2 we can see that CTWALK outperforms in most cases the temporal synthetic graph attributes produced by CTGAN. This is expected indeed as CTGAN takes the time parameter as another discrete variable and not as an independent time parameter, as CTWALK does.

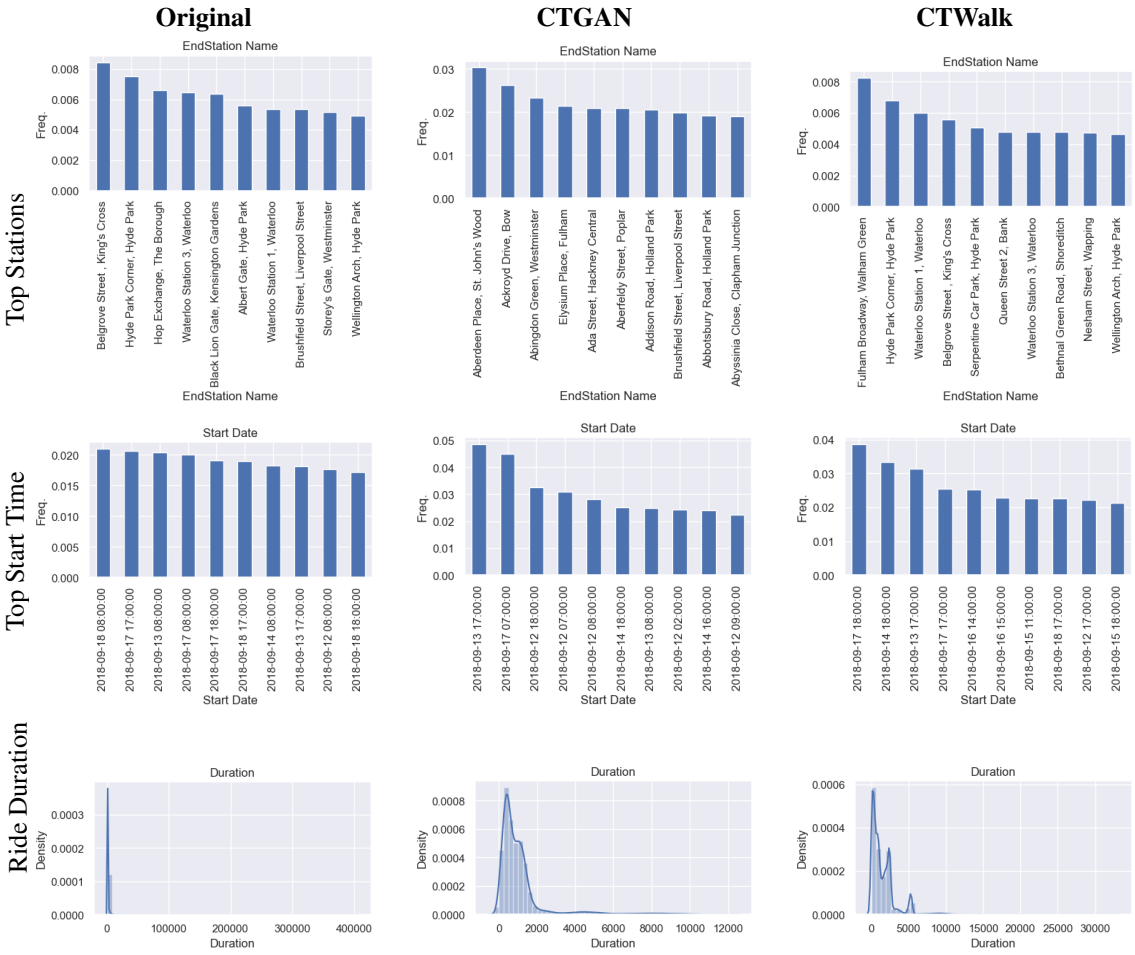

Figure 1: Discrete and continuous attributes results comparison for the synthetic Cycling Data.

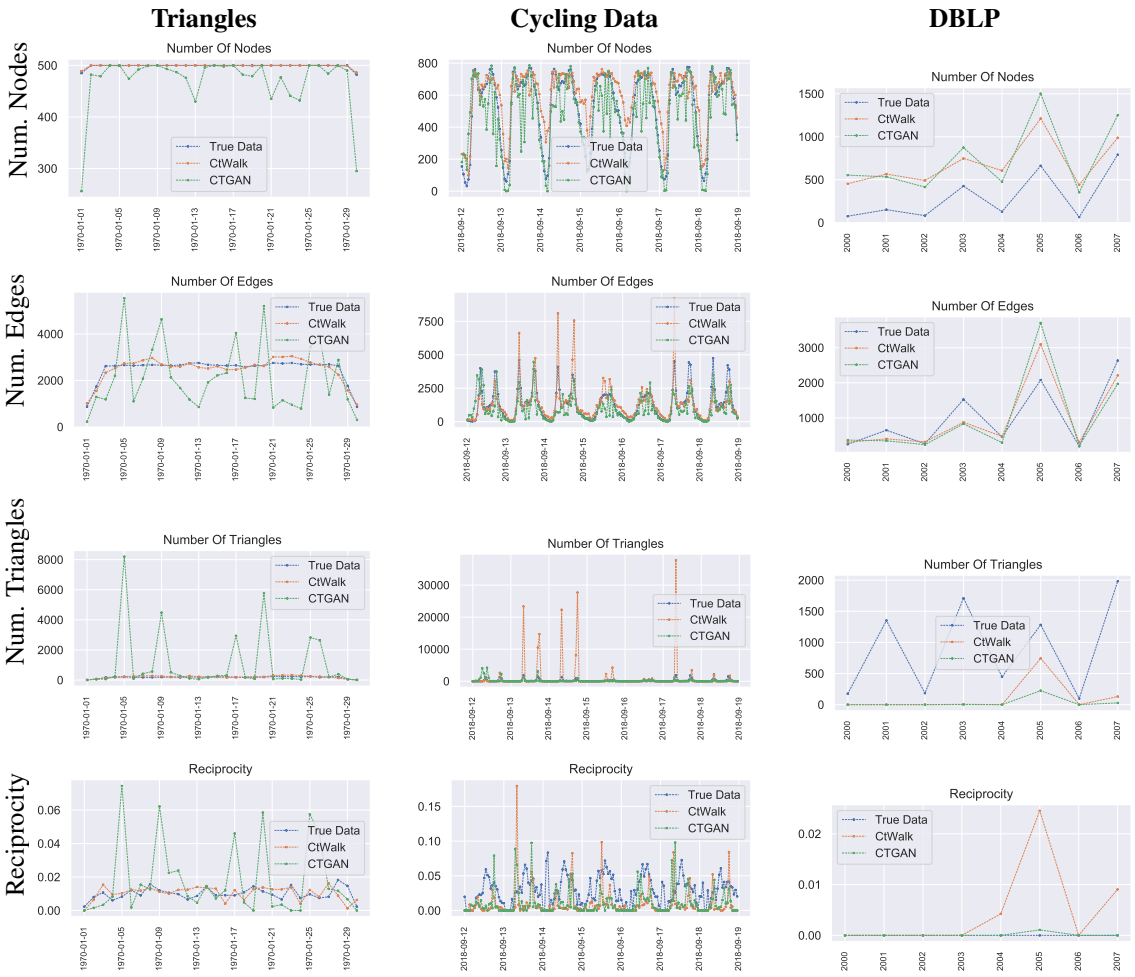

Figure 2: Time series results for several statistics (Number of Nodes, Number of Edges, Number of Triangles and Reciprocity) on the synthetic and real world networks (Triangles, Cycling Data and DBLP).