Continuous-Time Graph Representation with Sequential Survival Process

Abdulkadir Çelikkanat, Nikolaos Nakis, Morten Mørup

Department of Applied Mathematics and Computer Science, Technical University of Denmark
abce@dtu.dk, nnak@dtu.dk, mmor@dtu.dk

Abstract

Over the past two decades, there has been a tremendous increase in the growth of representation learning methods for graphs, with numerous applications across various fields, including bioinformatics, chemistry, and the social sciences. However, current dynamic network approaches focus on discrete-time networks or treat links in continuous-time networks as instantaneous events. Therefore, these approaches have limitations in capturing the persistence or absence of links that continuously emerge and disappear over time for particular durations. To address this, we propose a novel stochastic process relying on survival functions to model the durations of links and their absences over time. This forms a generic new likelihood specification explicitly accounting for intermittent edge-persistent networks, namely GRAS^{2}P: Graph Representation with Sequential Survival Process. We apply the developed framework to a recent continuous time dynamic latent distance model characterizing network dynamics in terms of a sequence of piecewise linear movements of nodes in latent space. We quantitatively assess the developed framework in various downstream tasks, such as link prediction and network completion, demonstrating that the developed modeling framework accounting for link persistence and absence well tracks the intrinsic trajectories of nodes in a latent space and captures the underlying characteristics of evolving network structure.

1 Introduction

In diverse fields spanning physical and social sciences, entities ranging from minuscule scales like microorganisms and proteins to larger scales such as humans to scales of celestial objects like planets and galaxies always exert influence upon and interact with one another. These evolving and intricate interconnections inherently translate into networks, providing a versatile framework to encapsulate the subtle interplay of relationships. In this regard, networks (or graphs) have become essential for investigating and comprehending the intricate dynamics of these complex systems evolving over time (Newman 2003).

Representation learning models on graphs have gained popularity due to their ability to effectively extract knowledge from networks and achieve various objectives like predicting linkage and node properties (Hamilton 2020; Zhang et al. 2020). However, their primary emphasis has been on static networks. The early works relied either on random walks (Perozzi, Al-Rfou, and Skiena 2014; Grover and Leskovec 2016; Tang et al. 2015) or matrix factorization techniques (Qiu et al. 2018, 2019). In recent years, Graph Neural Network (GNN) architectures have become a prominent way to address network embedding problems (Wu et al. 2021), and a plethora of methods have also been developed to address a variety of network types, such as signed networks (Li et al. 2020; Nakis et al. 2023) and knowledge graphs (Dai et al. 2020), or to serve diverse purposes like encoding the hierarchical structure of networks in learning node embeddings (Bhowmick et al. 2020; Nakis et al. 2022).

Lately, there has been a growing interest in modeling and learning latent representations of temporal networks, encompassing the transient nature of node interactions (Xue et al. 2022). The evolving focus seeks to unveil a richer understanding of node interactions throughout time, accounting for relationships' complex and evolving dynamics. Importantly, dynamic network modeling can thereby reveal intricate patterns within network structures that static approaches cannot adequately address. Initially, these dynamic modeling approaches focused on discrete time networks (Sarkar and Moore 2006; Ishiguro et al. 2010; Heaukulani and Ghahramani 2013; Herlau, Mørup, and Schmidt 2013; Yang et al. 2021). However, in recent years, substantial attention has also been devoted to the modeling of continuous-time networks. Prominent works have utilized Poisson (Fan et al. 2021; Trivedi et al. 2019; Celikkanat, Nakis, and Mørup 2022) and Hawkes processes (Hawkes 1971b,a; Blundell, Beck, and Heller 2012; Arastuie, Paul, and Xu 2020; Delattre, Fournier, and Hoffmann 2016; Zuo et al. 2018; Lu et al. 2019; Huang et al. 2022; Yang, Rao, and Neville 2017) in order to define principled learning procedures under continuous-time network likelihoods of event-based data. Contrary to the previous studies, which work on a network block level, the HTNE (Zuo et al. 2018) extends the Hawkes process modeling to account for node-level embeddings. Furthermore, the GNN extensions for continuous-time dynamic networks, TGN (Rossi et al. 2020), and the temporal-point process of M^{2}DNE (Lu et al. 2019) use a case-control approach optimizing a binary cross-entropy loss. In particular, M^{2}DNE takes into account both pairwise interactions at the micro level and broader network-wide dynamics at the macro level. Finally, non-likelihood-based procedures utilizing dynamic random walks such as (CTDNE) (Nguyen et al. 2018) perform temporal random walks based on the observed continuous-time interactions.

However, the currently existing approaches designed for modeling continuous-time dynamic networks exhibit significant limitations. In particular, when utilizing the event-based Poisson Process or extended Hawkes Process, they treat network links as instantaneous events, whereas the case-control approach using binary cross-entropy does not explicitly account for edge persistence in the likelihood. Nevertheless, numerous continuous-time dynamic networks in real-world scenarios surpass these perspectives. Links within these real-world networks often showcase intermittent patterns as interactions persist and dissipate consecutively over time. This nuanced nature of network dynamics necessitates a more comprehensive approach to accurately account for the persistent presence and absence of edges between nodes.

There are many prominent examples in which we can observe intermittent persistent linkage structures in real-world scenarios. For instance, consider a social media platform where users can follow or unfollow each other and thereby form a connection with each other over different time periods or contact and collaboration networks in which people can respectively be together or collaborate for extended periods of time. These intermittent persistent pairwise dynamics challenge traditional continuous-time dynamic network models that only account for the event of a tie but not its persistence and static models that assume constant and steady relationships. Although there are recent efforts modeling networks of intermittent link characteristics, they rely on certain assumptions, such as the stochastic block model (Rastelli and Fop 2020; Xu 2015), and are unable to produce continuous-time latent representations. There is, therefore, a need for new continuous-time dynamic network modeling approaches that can explicitly account for network connections that persist and dissipate consecutively over time.

In this paper, we introduce the continuous-time Graph Sequential Survival Process (GRAS^{2}P). Specifically, we extend the traditional usage of Survival analysis to the realm of network science by developing a *Sequential Survival* process that can capture the dynamic persistence of links and their absence in networks. To the best of our knowledge, this is the first approach capable of explicitly characterizing networks featuring intermittent time-persistent linkage structures. The main contributions can be outlined as:

- **A Novel Counting Process**. We introduce a novel stochastic process by leveraging the survival analysis to model the intermittent time-persistent linkage structure of the networks forming the GRAS^{2}P model. We further highlight the utility of the GRAS^{2}P model considering the recently proposed continuous-time node embedding procedure (Celikkanat, Nakis, and Mørup 2022).
- **Experimental Validation**. We conduct extensive experiments on diverse real-world datasets to evaluate GRAS^{2}P. The results showcase its effectiveness in capturing intricate characteristics of networks by explicitly accounting for intermittent edge persistence, and outperforming baseline methods in downstream tasks.
- **Visualization Tool**. We show that the proposed approach can embed continuous-time edge persistent dynamic complex networks in low dimensional spaces accurately, thereby also serving as a visualization tool to get insights into the intricate temporal dynamics of link-persistent networks.

Implementation. The source codes and other details can be found at https://abdcelikkanat.github.io/projects/grassp

2 Proposed Model

In this section, we present our proposed approach, but before delving into the details, we will first establish the notations used throughout the paper. Without loss of generality, we can suppose that the timeline begins at time 0 and ends at T, and we will use $[T]$ to denote the time interval $[0, T)$. We employ the conventional notation, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to indicate a graph where $\mathcal{V} = \{1, \ldots, N\}$ is the vertex set and $\mathcal{E} := \cup_{i,j \in \mathcal{V}} \mathcal{E}_{ij}$ refers to the edge set of the network, comprising of pairwise temporal links, $\mathcal{E}_{ij}$, for each pair $(i, j) \in \mathcal{V}^2$.

Again, it is worth emphasizing that we assume a pair of nodes consists of sequential links indicating intermittent interactions over time (Holme and Saramäki 2012). An existing link (i.e., interaction) can disappear and then emerge again. In this regard, we will utilize tuple (i, j, t_k, t_{k+1}) to denote a link between nodes i and j for the interval from $t_k \in [T]$ up to $t_{k+1} \in [T]$. We provide the formal description of the networks considered in this work in Definition 2.1 below:

Definition 2.1 (Continuous-time Interval Graphs)**.** A *continuous-time interval network* over a timeline $[T] := [0, T]$ is an ordered pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \ldots, N\}$ is the set of nodes and $\mathcal{E} := \{(i, j, t_k, t_{k+1}) \in \mathcal{V}^2 \times [T]^2 \mid t_k < t_{k+1}\}$ denotes the set of non-overlapping temporal links, i.e. if (i, j, t_k, t_{k+1}) and $(i, j, \bar{t}_l, \bar{t}_{l+1})$ are distinct links of (i, j) pair, then it satisfies $[t_k, t_{k+1}) \cap [\bar{t}_l, \bar{t}_{l+1}) = \emptyset$.

We call the initial and the last time of each link period as an *event time*, and for practical purposes, we always suppose 0 is also an event time for each node pair. In addition, we introduce the *state* function, $s : \mathcal{V}^2 \times [T] \to \mathcal{S}$ as an indicator of the presence or absence of a link for a given time $t \in [T]$ where $\mathcal{S} := \{1, -1\}$ is the state space ($+1$ symbolizes the existence of the link and -1 its absence). Note that the state of each pair is constant until the next event time; thereby, we omit the input variable from the state function, $s(t)$, for convenience, and we write s to denote the state of the interaction for the corresponding interval. In this regard, we can partition the timeline with respect to the values of the state function for each node pair (i.e., depending on whether a link exists or not), so if a pair consists of M events $e_0 = 0 < e_1 < \cdots < e_m < \cdots < e_{M-1} < T$ then there must exist M consecutive intervals, $\{[e_m, e_{m+1}) \subseteq [T] : \forall m \in \{0, \ldots, M-1\}\}$, having different states.

Even though networks with sporadic interactions over time are prevalent in real-world contexts such as contact or social networks, to the best of our knowledge, they have not been considered previously to learn the continuous-time latent node representations.

Sequential Survival Process

Many research fields strongly emphasize modeling the time duration required for an event to unfold, such as investigating the lifespan of living organisms or analyzing the reliability of mechanical systems. The term ”survival” is mostly employed in those works to describe the duration leading up to the occurrence of death or failure, which is a measure that encapsulates the essence of lifetime estimation and plays a fundamental role in understanding the dynamics of complex systems. More formally, for a given continuous random variable, T, representing the lifetime of an object or a system, the survival function is given by

$$S(t) := \mathbb{P}\{T > t\} = \int_{t}^{\infty} f(u)\mathrm{d}u = 1 - F(t)$$

where $F(t)$ and $f(t)$ indicate the cumulative distribution and probability density functions. It can also be reparameterized using an associated hazard function $\lambda : [0, \infty) \rightarrow \mathbb{R}^{+}$ as:

$$S(t) = \exp\left(-\int_{0}^{t} \lambda(t')dt'\right).$$

For further details on survival analysis, we recommend unfamiliar readers refer to the work of (Jenkins 2005).

Our approach characterizes node interactions by utilizing the power of survival functions. As mentioned before, we always assume that the state of the model alters after each event time point. Therefore, we design our *Sequential Survival* process as consecutive survival functions denoting ”surviving” and ”dying” events. In other words, for a given initial state, $s_0 \in \mathcal{S}$, we define the process, $\{M(t) : t \geq 0\}$ as a counting process showing the total number of occurrences or events that have happened up to time t. Hence, we write the probability of the random variable $M(t)$ being equal to m as follows:

$$p_{M(t)}(m) = \int\limits_{\boldsymbol{\xi}\in\mathcal{R}} \prod_{k=1}^{m} \frac{\int_{e_{k-1}}^{e_k} \lambda(s_k, t')\mathrm{d}t'}{\exp\left(\int_{e_{k-1}}^{e_k} \lambda(s_k, t')\mathrm{d}t'\right)} \mathrm{d}\boldsymbol{\xi} \quad (1)$$

where $\lambda(s_k, t)$ is the hazard rate for given time $t \in [T]$ and state $s_k \in \mathcal{S}$, and $\mathcal{R} := \{(t_1, \ldots, t_m) \in [T]^m : 0 \leq t_1 < t_2 < \cdots t_m < T)\}$ is the domain of the integration.

We can also write the likelihood function of the process from the probability given in Eq. (1). Let $\Xi = (\Phi, M(t))$ be a random variable where $M(t)$ denotes the number of events up to time t, and Φ is the corresponding ordered event sequence. Then, we can write the marginal distribution of $M(t)$ by integrating over all possible ordered sequences in the set $\mathcal{R}$. In other words,

$$p_{M(t)}(m) = \int_{\boldsymbol{\xi}\in\mathcal{R}} p_{(\Phi, M(t))}(\boldsymbol{\xi}, m) d\boldsymbol{\xi},$$

and by using the fundamental theorem of calculus, we can obtain the probability density function of the random variable, $\Xi = (\Phi, M(t))$, evaluated at $((e_1, \ldots, e_m), m)$ as:

$$p_{(\Phi, M(t))}((e_1, \ldots, e_m), m) = \prod_{k=1}^{m} \frac{\lambda(s_k, e_k)}{\exp\left(\int\limits_{e_{k-1}}^{e_k} \lambda(s_k, t')\mathrm{d}t'\right)}. \quad (2)$$

Problem Formulation

Our objective is to learn continuous-time node representations in a metric space $(\mathsf{X}, d_{\mathsf{X}})$ to uncover underlying temporal patterns of a network so the pairwise distances among nodes in a latent space should acquire the temporal changes within the network. We will use, $\mathbf{r}(i, t)$ or simply $\mathbf{r}_i(t)$, to denote the embedding of node $i \in \mathcal{V}$ at time $t \in [T]$ in a D-dimensional space ($D \ll |\mathcal{V}|$). More specifically, we desire to obtain a map $\mathbf{r} : \mathcal{V} \times [T] \rightarrow \mathsf{X}$ satisfying

$$\int_{e_l}^{e_u} \psi^{s}\left(d_{\mathsf{X}}\big(\mathbf{r}(i,t), \mathbf{r}(j,t)\big)\right)\mathrm{d}t \approx \int_{e_l}^{e_u} \boldsymbol{\lambda}_{ij}(s, t')\mathrm{d}t' \quad (3)$$

for a continuous function $\psi^{s} : \mathbb{R} \rightarrow \mathbb{R}^{+}$ and all $(i, j, s) \in \mathcal{V}^2 \times \mathcal{S}$ where $\boldsymbol{\lambda}(s, t)$ indicates the *true hazard rate* between i and j at time $t \in [T]$ and state $s \in \mathcal{S}$.

Since we assume that a node pair has connections of alternating states (i.e., link or non-link periods) over time, we utilize the Sequential Survival process introduced in the previous part to characterize these intermittent persistent edges. In this regard, by using Eq (2), we can write the log-likelihood function for the whole network as follows:

$$\begin{aligned}\mathcal{L}(\Omega|\mathcal{G}) &:= \log p(\mathcal{G}|\Omega)\\ &= \sum_{i,j\in\mathcal{V}} \sum_{m=1}^{|\mathcal{E}_{ij}|} \left(\log \lambda_{ij}(s_m, e_m) - \int\limits_{e_m}^{e_{m+1}} \lambda_{ij}(s_m, t)\mathrm{d}t\right)\end{aligned} \quad (4)$$

where Ω refers to the set of model hyper-parameters.

To learn continuous-time node dynamics, we consider the latent distance modeling framework (Hoff, Raftery, and Handcock 2002). We leverage the hazard functions to define the latent representations uncovering the evolving relationships between nodes in the network. Based on our assumption, when a pair of nodes has a link or interaction at a particular time, it is expected to dissolve eventually. As a result, their latent positions should also naturally drift apart from each other over time to reflect their temporal connection strength. Conversely, when they do not have any connection, they might interact in the future, so their latent positions should also approach each other to reflect the potential for a coming link. In this regard, we define the hazard function, $\lambda_{ij}(s, t)$ as follows:

$$\lambda_{ij}(s, t) := \exp\left(\beta(s) + s\|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2\right). \quad (5)$$

for node pair $(i, j) \in \mathcal{V}^2$ and state $s \in \mathcal{S} := \{-1, 1\}$. We incorporate bias terms ($\beta(s)$) for each state value in the definition of the hazard function given in Eq. (5), and they are responsible for capturing the global information in the network (Krivitsky et al. 2009; Celikkanat, Nakis, and Mørup 2022). We further use the squared Euclidean metric (Smith, Asta, and Calder 2019; Nickel and Kiela 2017). Using this formulation, Lemma 2.1 ensures the latent representations of nodes will be positioned close enough or significantly distant from each other depending on the state (i.e., link or non-link periods) of the node pairs.

Lemma 2.1. *Let e_k and e_{k+1} be a consecutive event times following a Sequential Survival process for node pair*

$(i,j) \in \mathcal{V}^2$. *Then, the average squared distance between nodes during interval* $[e_k, e_{k+1})$ *associated with survival function* $S(\cdot)$ *and state* $s \in \{-1,1\}$ *can be bounded by*

$$b(-1) \leq \frac{1}{(e_{k+1}-e_k)} \int_{e_k}^{e_{k+1}} \|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2 dt \leq b(+1)$$

where $b(s) := -2s\log(e_{k+1} - e_k) - s\log S(e_{k+1}) - s\beta(s)$.

Proof. Please refer to the appendix on the project page. □

Continuous-Time Node Representations Using Piecewise Linear Approximations

For the embedding vectors $\{\mathbf{r}_i(t) : i \in \mathcal{V}, t \in [T]\}$ we consider the continuous-time extension of the latent distance model proposed in (Celikkanat, Nakis, and Mørup 2022) in the context of event-based (Poisson Process likelihood) graphs using analytically tractable piecewise linear approximations of latent dynamics. Specifically, we define each node embedding as a linear function depending on time:

$$\mathbf{r}_i(t) := \mathbf{x}_i + \mathbf{v}_i t \tag{6}$$

The definition can be understood as assigning the initial position ($\mathbf{x}_i$) and velocity ($\mathbf{v}_i$) to each node, enabling us to locate the node's position in the latent space at any given time. However, it also constrains the motion capacity of nodes in the embedding space, as they are limited to moving in a single direction. To overcome this limitation, the model is extended by dividing the timeline into B equal-sized bins, introducing bin-specific velocity vectors. More specifically, the latent position of node $i \in \mathcal{V}$ at time $t \in [T]$ is given by

$$\mathbf{r}_i(t) := \mathbf{x}_i^{(0)} + \Delta_B \mathbf{v}_i^{(1)} + \Delta_B \mathbf{v}_i^{(2)} + \cdots + \\ + \Delta_B \mathbf{v}_i^{(b)} + \cdots + \mathrm{mod}(t, \Delta_B)\mathbf{v}_i^{(\lfloor t/\Delta_B \rfloor + 1)} \tag{7}$$

where Δ_B is the bin width (i.e. T/B), and $\mathrm{mod}(\cdot,\cdot)$ is the modulo operation giving the remaining time. Importantly, employing such a piecewise interpretation of the timeline enables tracking the paths of nodes in the embedding space effectively, and by augmenting the number of bins, more accurate trajectories can be obtained. In particular, the use of finer-grained divisions in the timeline allows for a more detailed and precise representation of node movements, leading to improved accuracy in capturing their dynamics within the embedding space (Celikkanat, Nakis, and Mørup 2022).

Regularization. In order to control the nodes' mobility in the latent space, we incorporate a prior distribution for the velocity vectors. Imagine a situation for a pair of nodes only interacting with each other during a period; the model situates them closely in the embedding space when they have a link. Nevertheless, their distance in the latent space tends towards infinity as the link is inactive. Therefore, we assume the velocity vectors, $\mathbf{v} \in \mathbb{R}^{B \times N \times D}$, follow a multivariate normal distribution with zero mean:

$$\mathrm{vect}(\mathbf{v}) \sim \mathcal{N}(\mathbf{0}, \lambda^2 \Sigma)$$

where λ is the scaling coefficient, and $\Sigma \in \mathbb{R}^{BND \times BND}$ is a diagonal matrix defined as a Kronecker product of three other vectors. In other words, $\Sigma := \mathrm{diag}(\boldsymbol{\sigma}_B \otimes \boldsymbol{\sigma}_N \otimes \boldsymbol{\sigma}_D)$ where the vectors, $\boldsymbol{\sigma}_B$, $\boldsymbol{\sigma}_N$ and $\boldsymbol{\sigma}_D$ are responsible for the influence of the model's bins, nodes, and dimensions, respectively. Here, the notation, $\otimes$, symbolizes the Kronecker product, and $\mathrm{vect}(\mathbf{z})$ represents the vectorization operator converting the given tensor into a vector form. We constrain $\boldsymbol{\sigma}_B$ and $\boldsymbol{\sigma}_N$ within the standard $(B-1)$ and $(N-1)$-simplex sets, and we define $\boldsymbol{\sigma}_D$ as $\mathbf{1}_D = (1, 1, \ldots, 1) \in \mathbb{R}^D$ to have uncorrelated dimensions. To sum up, we can restrain the embedding space by utilizing the prior distribution since it allows us to control the motions of nodes. For higher values of the scaling factor λ, the model can have more flexibility, enabling more dynamic node movements in the latent space, whereas lower values restrict node mobility, resulting in more static node representations. Notably, with this regularization, the model can be considered a continuous-time extension of the discrete-time latent distance model based on diffusion considered in (Sarkar and Moore 2006) in which the diffusion between time-bins propagates continuously.

Inference

Our objective function defined by the log-likelihood given in Eq. (4) together with the log-prior regularization term is a non-convex function, so the learning strategy applied for inferring the model's hyper-parameters is crucial to avoid poor-quality local minima in the resulting representations.

The position vectors ($\mathbf{x}$) are initialized uniformly within the $[-1, 1]$ range at random. The bias terms ($\boldsymbol{\beta}$) and velocities ($\mathbf{v}$) are sampled from the standard normal distribution. The prior parameters, ($\boldsymbol{\sigma}_B$, $\boldsymbol{\sigma}_N$) are set to $\mathbf{1}_B/B$ and $\mathbf{1}_N/N$ at the beginning. We follow a sequential learning strategy for training the model, i.e., we optimize different sets of parameters in stages. Firstly, we optimize the velocities ($\mathbf{v}$) for 100 epochs. Then, we include the initial positions ($\mathbf{x}$) into the optimization procedure, and we continue to train the model by optimizing these two parameters ($\mathbf{x}$, $\mathbf{v}$) together for another 100 epochs. Finally, we incorporate the bias and prior parameters and optimize all model hyper-parameters together. In total, we use 300 epochs for the whole learning procedure, and the *Adam optimizer* (Kingma and Ba 2017) is employed with an initial learning rate of 0.1. In the experiments, we set the number of bins (B) to 100 to ensure sufficient capacity for tracking nodes in the latent space ($D = 2$).

3 Experimental Evaluation

In this section, we will examine the performance of the proposed model over a diverse range of networks varying in size and characteristics. But before delving into the experimental evaluations, we will first present details regarding the experimental setup, considered datasets, and baseline approaches.

Experimental Setup. We first split the networks into two sets, such that the events taking place within the last 10% of the timeline are considered for the future prediction task. Furthermore, we randomly choose 20% node pairs among all possible dyads in the initial first part, and they are divided into two equal-sized groups to design the validation and testing sets. The residual network does not contain any link from these dyads, and it forms the training set. If there

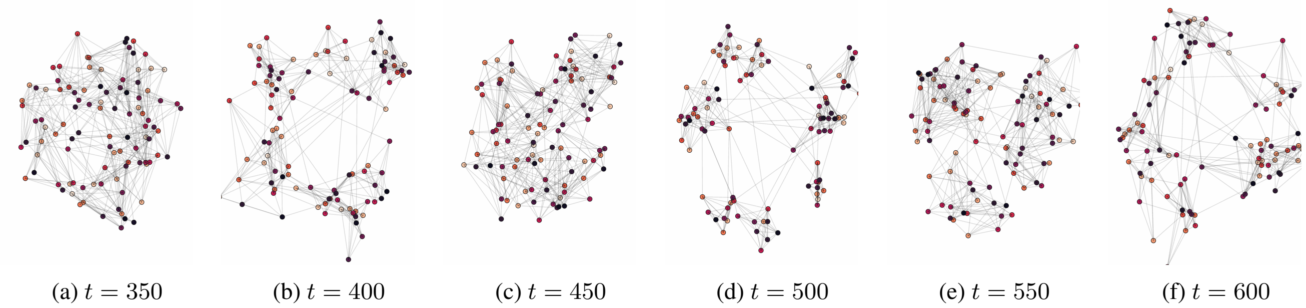


Figure 1: Snapshots of the continuous-time embeddings learned by GRAS^{2}P for various time points over *Synthetic-β*.

is any node pair without any link period during the training time but included in the prediction set, it is also excluded from the network.

We need to generate the labeled data to perform link prediction tasks. For this purpose, we divide the timeline of each sampled dyad into segments based on the state values. Within these segments, we randomly select time t to define a sample interval $[t-\epsilon, t+\epsilon]$, and ϵ is set to $10^{-2} \times T$ where T is the dataset's timeline length. We deliberately exclude samples containing the event times where the state of the corresponding dyad changes since it is impractical to predict whether a link exists for the periods with multiple states.

We organized these generated samples into two categories as "*simple*" and "*hard*" sets. The "*hard*" set consists of samples for the node pairs having at least one linked and non-linked period over time (Huang et al. 2023). Contrarily, dyads having stable states throughout the timeline produce "*simple*" sets of samples since predicting the labels (i.e., state) of these instances is relatively straightforward. Additionally, the samples generated for the future link prediction task are categorized based on the dyads' linkage during the training time by following the study (Poursafaei et al. 2022).

We consider an equal number of k link and non-link samples, and $k := \min\{10^3, \text{link set size}, \text{non-link set size}\}$. Each link or non-link category contains $h/2$ elements picked up from the *hard* set of size h, and we randomly select the remaining $k - h/2$ samples from the residual *hard* and the *simple* instances. For the experiments, we report both AUC-ROC and AUC-PR scores to comprehensively evaluate the models' performances across different aspects of true and false positives and precision-recall characteristics.

Datasets

We treat the networks as undirected and employ the finest available temporal granularity for the input timestamps, including measurements at the level of seconds and milliseconds. We tailor the datasets according to the chosen baselines to enable a meaningful comparison. For instance, we transform dynamic networks into static weighted and unweighted networks by aggregating links over time for static baselines. Additionally, we exclude the non-link events for the baselines since they cannot process these data points.

In the experiments, we have used several real datasets of diverse types, including a social network (*Facebook*) (Viswanath et al. 2009), collaboration graph (*NeurIPS*), and three contact networks (Génois et al. 2015; Isella et al. 2011). Due to the constraints on the page number, we provide the details regarding the real datasets on the project page. We also generated two synthetic networks to examine the model's predictive behavior in controlled settings. The link and non-link event times of the *Synthetic-α* graph are generated from the Sequential Survival process introduced in Section 2, and the initial embedding locations and velocities are sampled from a multivariate normal distribution as described in (Celikkanat, Nakis, and Mørup 2022). For the *Synthetic-β* network, we divide the timeline into equal-sized 8 bins and randomly group the nodes into 10 clusters for each bin. Then, we establish connections between nodes within the same cluster with a probability of 0.8, while nodes belonging to different clusters are linked with a 10^{-2} probability. These links stay persistent until the next bin.

Baselines

Due to the lack of an analogous approach to our proposed model explicitly accounting for dynamic networks with intermittent persistent linkage structures, we establish several static and dynamic methods as baselines to compare with the performance of our model. (i) LDM (Krivitsky et al. 2009; Hoff 2005) is a static node embedding approach for unweighted graphs in which we used the Poisson formulation in (Nakis et al. 2022) when modeling the links with node-specific random effect terms. (ii) PIVEM (Celikkanat, Nakis, and Mørup 2022) can be viewed as an expansion of the LDM model, which models event-based continuous-time dynamic networks using the Nonhomogeneous Poisson process as opposed to the presently considered GRAS^{2}P. (iii) NODE2VEC (Grover and Leskovec 2016) is very well-known random-walk based node embedding and we here report best obtained result considering both unweighted and weighted graphs. (iv) CTDNE (Nguyen et al. 2018) also perform random walks but it concentrates on the temporal networks. (v) HTNE (Zuo et al. 2018) relies on the Hawkes process to learn node embeddings and to model the neighbor arrival events. Finally, (vi) M^{2}DNE (Lu et al. 2019) embeds the dynamic graphs by capturing the micro and macro network properties using case-control inference with cross-entropy loss. For all embedding methods we set $D = 2$.

		LDM	NODE2VEC	CTDNE	HTNE	M^{2}DNE	PIVEM	GRAS^{2}P
Synthetic-α	ROC	.711 ± .004	.743 ± .002	.692 ± .007	.698 ± .021	.558 ± .008	$\underline{.744 \pm .002}$	**.810 ± .009**
	PR	.630 ± .006	$\underline{.667 \pm .009}$	.650 ± .007	.645 ± .019	.582 ± .004	.653 ± .004	**.751 ± .011**
Synthetic-β	ROC	.491 ± .020	.534 ± .008	.502 ± .008	.525 ± .004	.517 ± .013	$\underline{.593 \pm .006}$	**.677 ± .018**
	PR	.486 ± .016	.498 ± .007	.502 ± .010	.517 ± .006	.522 ± .015	$\underline{.587 \pm .011}$	**.646 ± .022**
Contacts	ROC	.508 ± .008	$\underline{.584 \pm .004}$	.564 ± .034	.472 ± .024	.486 ± .013	.493 ± .006	**.680 ± .013**
	PR	.490 ± .004	$\underline{.555 \pm .023}$	.543 ± .036	.477 ± .023	.500 ± .008	.492 ± .016	**.641 ± .023**
HyperText	ROC	$\underline{.541 \pm .015}$	.533 ± .012	.462 ± .016	.441 ± .017	.461 ± .021	.426 ± .013	**.692 ± .010**
	PR	$\underline{.503 \pm .010}$	.490 ± .013	.477 ± .016	.449 ± .009	.479 ± .023	.437 ± .007	**.656 ± .024**
Infectious	ROC	$\underline{.689 \pm .007}$	.671 ± .003	.639 ± .006	.653 ± .013	.554 ± .005	.669 ± .004	**.742 ± .026**
	PR	$\underline{.615 \pm .007}$	.601 ± .005	.593 ± .005	.596 ± .010	.560 ± .009	.598 ± .004	**.673 ± .024**
Facebook	ROC	.717 ± .004	.675 ± .001	.539 ± .005	.608 ± .001	.570 ± .010	$\underline{.710 \pm .002}$	**.723 ± .010**
	PR	.659 ± .006	.603 ± .005	.538 ± .013	.575 ± .001	.562 ± .009	$\underline{.662 \pm .002}$	**.671 ± .012**
NeurIPS	ROC	.679 ± .010	.697 ± .005	.558 ± .020	.654 ± .025	.531 ± .005	**.748 ± .010**	$\underline{.735 \pm .029}$
	PR	.618 ± .016	.606 ± .020	.552 ± .025	.613 ± .026	.553 ± .011	**.761 ± .020**	$\underline{.749 \pm .021}$

Table 1: The performance comparison of the models for the network completion experiment across diverse datasets.

		LDM	NODE2VEC	CTDNE	HTNE	M^{2}DNE	PIVEM	GRAS^{2}P
Synthetic-α	ROC	.702 ± .002	.693 ± .003	.638 ± .006	.675 ± .011	.507 ± .002	$\underline{.749 \pm .002}$	**.845 ± .006**
	PR	.654 ± .006	.627 ± .011	.596 ± .009	.639 ± .007	.566 ± .003	$\underline{.665 \pm .002}$	**.782 ± .009**
Synthetic-β	ROC	.564 ± .009	.507 ± .006	.512 ± .008	.544 ± .007	.511 ± .002	$\underline{.680 \pm .006}$	**.744 ± .019**
	PR	.553 ± .006	.494 ± .005	.511 ± .007	.528 ± .007	.513 ± .002	$\underline{.652 \pm .008}$	**.701 ± .013**
Contacts	ROC	$\underline{.593 \pm .004}$	.556 ± .004	.534 ± .018	.528 ± .004	.534 ± .002	.496 ± .006	**.825 ± .008**
	PR	$\underline{.541 \pm .003}$	.523 ± .015	.528 ± .017	.510 ± .008	.537 ± .004	.465 ± .002	**.754 ± .014**
HyperText	ROC	$\underline{.550 \pm .002}$	.535 ± .004	.477 ± .012	.473 ± .011	.489 ± .003	.430 ± .002	**.760 ± .004**
	PR	$\underline{.513 \pm .003}$	.507 ± .007	.488 ± .010	.470 ± .008	.479 ± .004	.431 ± .001	**.689 ± .007**
Infectious	ROC	$\underline{.701 \pm .006}$	.688 ± .003	.667 ± .005	.676 ± .009	.579 ± .002	.666 ± .005	**.788 ± .015**
	PR	$\underline{.626 \pm .008}$	.602 ± .007	.606 ± .008	.613 ± .008	.584 ± .005	.577 ± .006	**.697 ± .013**
Facebook	ROC	$\underline{.682 \pm .005}$	.645 ± .003	.544 ± .007	.624 ± .002	.582 ± .009	.673 ± .003	**.731 ± .010**
	PR	.615 ± .009	.589 ± .008	.535 ± .008	.590 ± .009	.573 ± .009	$\underline{.617 \pm .004}$	**.667 ± .013**
NeurIPS	ROC	$\underline{.760 \pm .007}$	.720 ± .003	.598 ± .004	.731 ± .008	.594 ± .001	.698 ± .002	**.889 ± .013**
	PR	.687 ± .010	.631 ± .007	.590 ± .010	.659 ± .006	.599 ± .003	$\underline{.711 \pm .002}$	**.819 ± .020**

Table 2: The performance comparison of the models for the network reconstruction experiment across diverse datasets.

Network Reconstruction. Our goal is to see how effectively the models can capture the temporal structural changes within the network over time. In pursuit of this objective, we seek to reconstruct both link and non-link periods. As depicted in Table 2, our method, GRAS^{2}P, exhibits notably superior performance compared to the baseline approaches. This marked improvement is attributed to the incapability of the other models to represent intermittent persistent linkage structures accurately.

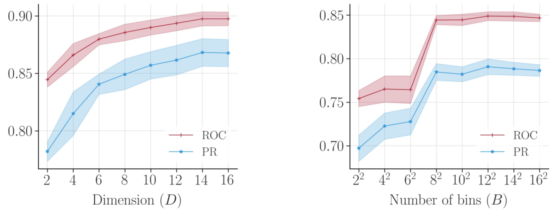
Network Completion. Many real networks often contain noisy or missing links for various reasons, such as problems in the data collection processes or privacy concerns preventing the full disclosure of ties. In this regard, our aim is to evaluate the models' capacity to generalize the linkage structure in the training set. Table 1 illustrates that our model once more demonstrates a substantial performance advantage over the baselines, except for the *NeurIPS* dataset. Given the network's yearly time resolution, the event-based approach, PIVEM, effectively captures its temporal structure. Importantly, our approach also displays a comparable level of performance in this scenario.

Future Link Prediction. The absence of a predictable periodic linkage pattern in the networks poses a significant challenge in forecasting future connections, particularly those at more distant time points. This complexity is evident in Table 3, where our model consistently surpasses all baselines on the *Synthetic*-α dataset, but its performance for *Synthetic*-β is not optimal. It can be explained by the fact that the links in the *Synthetic*-α network are sampled from the *Sequential Survival* process, but the temporal clusters in the *Synthetic*-β network are randomly formed (please see the dataset section for details). For specific network structures, static embedding models also showcase satisfactory performance since they are able to capture the global net-

		LDM	NODE2VEC	CTDNE	HTNE	M^{2}DNE	PIVEM	GRAS^{2}P
Synthetic-α	ROC	.748 ± .007	.756 ± .005	.652 ± .012	.784 ± .013	.654 ± .011	.740 ± .007	**.902 ± .011**
	PR	.719 ± .012	.700 ± .020	.636 ± .019	.800 ± .016	.745 ± .008	.741 ± .005	**.918 ± .008**
Synthetic-β	ROC	.515 ± .018	.538 ± .004	.503 ± .020	.560 ± .006	.519 ± .012	**.894 ± .005**	.880 ± .012
	PR	.525 ± .021	.501 ± .007	.494 ± .016	.548 ± .004	.554 ± .014	**.845 ± .007**	.843 ± .014
Contacts	ROC	**.821 ± .004**	.703 ± .002	.635 ± .013	.727 ± .002	.590 ± .002	.692 ± .005	.793 ± .013
	PR	**.773 ± .005**	.648 ± .006	.599 ± .014	.689 ± .004	.610 ± .006	.675 ± .004	.752 ± .019
HyperText	ROC	**.663 ± .004**	.553 ± .003	.503 ± .010	.530 ± .018	.548 ± .004	.559 ± .003	.654 ± .005
	PR	.609 ± .003	.516 ± .008	.503 ± .006	.518 ± .014	.529 ± .008	.534 ± .002	**.612 ± .010**
Infectious	ROC	**.958 ± .004**	.869 ± .002	.847 ± .008	.893 ± .013	.655 ± .008	.945 ± .006	.943 ± .017
	PR	**.943 ± .008**	.818 ± .007	.820 ± .014	.853 ± .007	.698 ± .009	.932 ± .006	.923 ± .025
Facebook	ROC	**.781 ± .007**	.694 ± .003	.564 ± .005	.626 ± .003	.609 ± .015	.775 ± .002	.705 ± .009
	PR	.765 ± .009	.653 ± .004	.557 ± .004	.599 ± .011	.603 ± .011	**.766 ± .003**	.648 ± .009
NeurIPS	ROC	.682 ± .019	.695 ± .012	.637 ± .007	.676 ± .014	.661 ± .006	.623 ± .010	**.820 ± .008**
	PR	.634 ± .024	.621 ± .018	.615 ± .015	.635 ± .025	.674 ± .014	.628 ± .006	**.788 ± .018**

Table 3: The performance comparison of the models for the future link prediction experiment across diverse datasets.

work information due to the aggregation of the links over time. Similarly, by choosing small values for the covariance factor, λ, we can restrict the dynamics of our approach.



(a) Impact of Dimension Size

(b) Impact of Bin Count

Figure 2: Impact of the hyper-parameters over *Synthetic-α*.

Impact of Dimension Size. Figure 2a shows a clear correlation between the increase in dimension size and performance improvement over the *Synthetic-α* network. With the introduction of each new dimension, the model's capacity augments, and the model becomes more adept at capturing intricate patterns within the network. For smaller dimensions, the model demonstrates performance on par with more high dimensions. It is important to highlight that the two-dimensional representations sustain competitive performance enabling easy visualization and extraction of insights into the complex and dynamic nature of networks.

Impact of Bin Count. The impact of the number of bins on performance improvements is evident in Figure 2b, as the model's capacity to capture subtle temporal changes increases at finer granularity levels. Notably, the model's performance nearly reaches near-optimal performance around $B = 64$, after which it demonstrates saturation.

Continuous-Time Dynamic Visualization. Network visualization offers valuable insights for practitioners into the intricate architectures of complex networks. Nonetheless, numerous methods necessitate high-dimensional spaces to achieve satisfactory results for downstream tasks. Therefore, practitioners must utilize dimension-reduction techniques to generate visualizations conducive to human comprehension. Furthermore, many temporal models yield static embeddings, lacking the capacity to produce continuous-time node representations. In this regard, our model proves to be a versatile tool, effectively balancing the tradeoff between performance and dimension sizes.

Figure 1 showcases the acquired embeddings across various selected time instances over the *Synthetic-β* dataset. The network takes an entirely new structural form for each time point $t = 100k$ ($k \in \{4, \ldots, 6\}$), including 10 new clusters appearing randomly. Our model learns these temporal structures, particularly when interconnections between clusters remain relatively sparse. As time progresses, nodes within the latent space gradually adjust their positions to align with the evolving new random connections, making the clusters indistinguishable ($t = 100k - 50$ where $k \in \{4, \ldots, 6\}$). Additional visualizations are included on the project page.

4 Conclusion

In this study, we introduced a novel representation learning model, GRAS^{2}P, designed specifically for continuous-time networks exhibiting intermittent persistent linkage patterns over time. Our proposed approach characterizes node connections using the proposed Sequential Survival process. Our experimental results clearly demonstrate the superiority of GRAS^{2}P over established baseline methods across multiple networks of varying properties. Notably, our model effectively balances the trade-off between dimensionality and performance, making it a valuable tool for visualizations.

We aim to extend the methodology for diverse forms of temporal graphs, including weighted and signed networks. Moreover, we plan to tailor our approach to address large-scale networks while also capturing potentially recurring periodic structures within the networks considering different model specifications beyond piecewise linear dynamics.

Acknowledgments

We appreciate the reviewers for their constructive feedback and their insightful comments. We gratefully acknowledge the Independent Research Fund Denmark for supporting this work [grant number: 0136-00315B].

References

Arastuie, M.; Paul, S.; and Xu, K. 2020. CHIP: A Hawkes Process Model for Continuous-time Networks with Scalable and Consistent Estimation. In *NeurIPS*, volume 33, 16983–16996.

Bhowmick, A. K.; Meneni, K.; Danisch, M.; Guillaume, J.-L.; and Mitra, B. 2020. LouvainNE: Hierarchical Louvain Method for High Quality and Scalable Network Embedding. In *WSDM*, 43–51.

Blundell, C.; Beck, J.; and Heller, K. A. 2012. Modelling Reciprocating Relationships with Hawkes Processes. In *NeurIPS*, volume 25.

Celikkanat, A.; Nakis, N.; and Mørup, M. 2022. Piecewise-Velocity Model for Learning Continuous-time Dynamic Node Representations. In *The First Learning on Graphs Conference*.

Dai, Y.; Wang, S.; Xiong, N. N.; and Guo, W. 2020. A survey on knowledge graph embedding: Approaches, applications and benchmarks. *Electronics*, 9(5): 750.

Delattre, S.; Fournier, N.; and Hoffmann, M. 2016. Hawkes processes on large networks. *The Annals of Applied Probability*, 26(1): 216 – 261.

Fan, X.; Li, B.; Zhou, F.; and SIsson, S. 2021. Continuous-time edge modelling using non-parametric point processes. *NeurIPS*, 34: 2319–2330.

Génois, M.; Vestergaard, C. L.; Fournet, J.; Panisson, A.; Bonmarin, I.; and Barrat, A. 2015. Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. *Network Science*, 3(3): 326–347.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864.

Hamilton, W. L. 2020. *Graph representation learning*. Morgan & Claypool Publishers.

Hawkes, A. G. 1971a. Point Spectra of Some Mutually Exciting Point Processes. *J. R. Stat. Soc*, 33(3): 438–443.

Hawkes, A. G. 1971b. Spectra of Some Self-Exciting and Mutually Exciting Point Processes. *Biometrika*, 58(1): 83–90.

Heaukulani, C.; and Ghahramani, Z. 2013. Dynamic probabilistic models for latent feature propagation in social networks. In *ICML*, 275–283.

Herlau, T.; Mørup, M.; and Schmidt, M. 2013. Modeling temporal evolution and multiscale structure in networks. In *International Conference on Machine Learning*, 960–968. PMLR.

Hoff, P. D. 2005. Bilinear mixed-effects models for dyadic data. *Journal of the American Statistical association*, 100(469): 286–295.

Hoff, P. D.; Raftery, A. E.; and Handcock, M. S. 2002. Latent Space Approaches to Social Network Analysis. *JASA*, 97(460): 1090–1098.

Holme, P.; and Saramäki, J. 2012. Temporal networks. *Physics Reports*, 519(3): 97–125.

Huang, S.; Poursafaei, F.; Danovitch, J.; Fey, M.; Hu, W.; Rossi, E.; Leskovec, J.; Bronstein, M.; Rabusseau, G.; and Rabbany, R. 2023. Temporal Graph Benchmark for Machine Learning on Temporal Graphs. arXiv:2307.01026.

Huang, Z.; Soliman, H.; Paul, S.; and Xu, K. S. 2022. A mutually exciting latent space Hawkes process model for continuous-time networks. In *Uncertainty in Artificial Intelligence*, 863–873. PMLR.

Isella, L.; Stehlé, J.; Barrat, A.; Cattuto, C.; Pinton, J.-F.; and Van den Broeck, W. 2011. What's in a crowd? Analysis of face-to-face behavioral networks. *Journal of theoretical biology*, 271(1): 166–180.

Ishiguro, K.; Iwata, T.; Ueda, N.; and Tenenbaum, J. 2010. Dynamic infinite relational model for time-varying relational data analysis. *NeurIPS*, 23.

Jenkins, S. P. 2005. Survival analysis. *Unpublished manuscript, Institute for Social and Economic Research, University of Essex, Colchester, UK*, 42: 54–56.

Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Krivitsky, P. N.; Handcock, M. S.; Raftery, A. E.; and Hoff, P. D. 2009. Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social networks*, 31(3): 204–213.

Li, Y.; Tian, Y.; Zhang, J.; and Chang, Y. 2020. Learning signed network embedding via graph attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 4772–4779.

Lu, Y.; Wang, X.; Shi, C.; Yu, P. S.; and Ye, Y. 2019. Temporal network embedding with micro-and macro-dynamics. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 469–478.

Nakis, N.; Celikkanat, A.; Boucherie, L.; Djurhuus, C.; Burmester, F.; Holmelund, D. M.; Frolcová, M.; and Mørup, M. 2023. Characterizing Polarization in Social Networks using the Signed Relational Latent Distance Model. In *AISTATS*, volume 206, 11489–11505. PMLR.

Nakis, N.; Çelikkanat, A.; Jørgensen, S. L.; and Mørup, M. 2022. A Hierarchical Block Distance Model for Ultra Low-Dimensional Graph Representations. arXiv:2204.05885.

Newman, M. E. 2003. The structure and function of complex networks. *SIAM review*, 45(2): 167–256.

Nguyen, G. H.; Lee, J. B.; Rossi, R. A.; Ahmed, N. K.; Koh, E.; and Kim, S. 2018. Continuous-time dynamic network embeddings. In *Companion proceedings of the the web conference 2018*, 969–976.

Nickel, M.; and Kiela, D. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *NIPS*, volume 30.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. DeepWalk: Online Learning of Social Representations. In *KDD*, 701–710.

Poursafaei, F.; Huang, S.; Pelrine, K.; and Rabbany, R. 2022. Towards better evaluation for dynamic link prediction. *Advances in Neural Information Processing Systems*, 35: 32928–32941.

Qiu, J.; Dong, Y.; Ma, H.; Li, J.; Wang, C.; Wang, K.; and Tang, J. 2019. NetSMF: Large-Scale Network Embedding As Sparse Matrix Factorization. In *WWW*.

Qiu, J.; Dong, Y.; Ma, H.; Li, J.; Wang, K.; and Tang, J. 2018. Network Embedding As Matrix Factorization: Unifying DeepWalk, LINE, PTE, and Node2Vec. In *WSDM*, 459–467.

Rastelli, R.; and Fop, M. 2020. A stochastic block model for interaction lengths. *Advances in Data Analysis and Classification*, 14: 485–512.

Rossi, E.; Chamberlain, B.; Frasca, F.; Eynard, D.; Monti, F.; and Bronstein, M. 2020. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*.

Sarkar, P.; and Moore, A. W. 2006. Dynamic Social Network Analysis using Latent Space Models. In Weiss, Y.; Schölkopf, B.; and Platt, J. C., eds., *Advances in Neural Information Processing Systems 18*, 1145–1152. MIT Press.

Smith, A. L.; Asta, D. M.; and Calder, C. A. 2019. The Geometry of Continuous Latent Space Models for Network Data. *Statistical Science*, 34(3): 428 – 453.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. LINE: Large-scale Information Network Embedding. In *WWW*, 1067–1077.

Trivedi, R.; Farajtabar, M.; Biswal, P.; and Zha, H. 2019. DyRep: Learning Representations over Dynamic Graphs. In *ICLR*.

Viswanath, B.; Mislove, A.; Cha, M.; and Gummadi, K. P. 2009. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM workshop on Online social networks*, 37–42.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.*, 32(1): 4–24.

Xu, K. 2015. Stochastic block transition models for dynamic networks. In *Artificial Intelligence and Statistics*, 1079–1087. PMLR.

Xue, G.; Zhong, M.; Li, J.; Chen, J.; Zhai, C.; and Kong, R. 2022. Dynamic network embedding survey. *Neurocomputing*, 472: 212–223.

Yang, J.; Rao, V. A.; and Neville, J. 2017. Decoupling Homophily and Reciprocity with Latent Space Network Models. In *UAI*.

Yang, M.; Zhou, M.; Kalander, M.; Huang, Z.; and King, I. 2021. Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1975–1985.

Zhang, D.; Yin, J.; Zhu, X.; and Zhang, C. 2020. Network Representation Learning: A Survey. *IEEE Trans. Big Data*, 6(1).

Zuo, Y.; Liu, G.; Lin, H.; Guo, J.; Hu, X.; and Wu, J. 2018. Embedding temporal network via neighborhood formation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2857–2866.