IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 August 2023, accepted 2 October 2023, date of publication 12 October 2023, date of current version 30 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3324213

RESEARCH ARTICLE

Gaussian Embedding of Temporal Networks

RAPHAËL ROMERO[1], JEFREY LIJFFIJT[1], RICCARDO RASTELLI[2], MARCO CORNELI[3,4], AND TIJL DE BIE[1]

[1]Department of Engineering and Architecture, Ghent University, 9000 Ghent, Belgium
[2]School of Mathematics and Statistics, University College Dublin, Dublin 4, D04 V1W8 Ireland
[3]CNRS, Laboratoire CEPAM, Université Côte d'Azur, 06103 Nice, France
[4]CNRS, INRIA, Laboratoire LJAD, Université Côte d'Azur, 06103 Nice, France

Corresponding author: Raphaël Romero (raphael.romero@ugent.be)

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) (ERC Grant Agreement no. 615517), and under the European Union's Horizon 2020 research and innovation programme (ERC Grant Agreement no. 963924), from the Special Research Fund (BOF) of Ghent University (BOF20/IBF/117), from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, and from the FWO (Fonds Wetenschappelijk Onderzoek , project no. G0F9816N, 3G042220).

ABSTRACT Representing the nodes of continuous-time temporal graphs in a low-dimensional latent space has wide-ranging applications, from prediction to visualization. Yet, analyzing continuous-time relational data with timestamped interactions introduces unique challenges due to its sparsity. Merely embedding nodes as trajectories in the latent space overlooks this sparsity. However, a natural way to account for this sparsity is to model the uncertainty around the latent positions. In this paper, we propose TGNE (**T**emporal **G**aussian **N**etwork **E**mbedding), an innovative method that bridges two distinct strands of literature: the statistical analysis of networks via Latent Space Models (LSM) and temporal graph machine learning. TGNE embeds nodes as piece-wise linear trajectories of Gaussian distributions in the latent space, capturing both structural information and uncertainty around the trajectories. We evaluate TGNE's effectiveness in reconstructing the original graph and modelling uncertainty. The results demonstrate that TGNE generates time-varying embedding locations that can accurately reconstruct missing parts of the network based on observed ones. Furthermore, the uncertainty estimates align experimentally with the time-varying degree distribution in the network, providing valuable insights into the temporal dynamics of the graph. To facilitate reproducibility, we provide an open-source implementation of TGNE at https://github.com/aida-ugent/tgne/.

INDEX TERMS Temporal networks, latent space models, variational inference, representation learning, dimensionality reduction, networks, random graphs.

I. INTRODUCTION

Continuous-time temporal networks arise from various sources. They have successfully been used to study communication patterns [23], epidemic spreading [4], [21], and neuron firing, to name a few. Moreover, interaction data is often available with a high level of detail, making it possible to model the time dimension as continuous. In that setting, a temporal interaction network can be viewed as the realization of a collection of edge-specific point processes, enabling the use of statistical methods to study the dynamics of the interactions [22].

Latent Space Models for graphs [10] are an important class of probabilistic models, where each node in the graph is embedded into a latent space, and the probabilities of links between nodes are independently distributed according to a notion of distance between node embeddings. Such dyadic independence models allow one to reliably infer unobserved node-level information based on the observation of links between the nodes. The learned node embeddings can then be used directly for downstream tasks such as clustering or link prediction [13]. Latent Space Models have been extended to make them applicable to a variety of network types, and publicly available packages allow analyzing a broad range of relational data [20].

For continuous-time temporal graphs, however, where each interaction is allowed to occur at any time stamp,

The associate editor coordinating the review of this manuscript and approving it for publication was Tawfik Al-Hadhrami.

the translation of Latent Space Models has not been fully explored yet. Indeed, for this type of data, the point process nature of the dyad-level variables that generate the data does not allow parameterizing the nodes into simple embedding vectors, but instead would require (in theory) a full trajectory of embeddings. To cope with these limitations, the recently introduced continuous latent position model (CLPM) [27] models the temporal network as arising from a multivariate point process, while assuming the latent trajectories of embeddings to be piece-wise linear in order to derive a fully parametric model. Rastelli and Corneli [27] proposed *maximum a posteriori* inference to estimate the latent trajectories based on a history of dyadic interactions. However, the authors do not consider estimating the uncertainty of the estimated trajectories. Given the sparsity of typical interaction networks, such uncertainty may be large and vary across nodes, dyads and time. Yet, understanding this uncertainty may be crucial in many applications. To meet this need, in this work we present TGNE (Temporal Gaussian Network Embedding), which hinges on Bayesian inference to capture a time-varying notion of uncertainty of the model on the latent position. While still allowing one to visualize temporal networks in a low-dimensional space, TGNE additionally allows one to gauge the uncertainty around the latent positions in a natural and rigorous manner.

Our contributions can be summarized as follows.

- We propose TGNE: a variational approach to inference in the CLPM model that allows one to calculate trajectories of Gaussian distributions for node embeddings in a latent space, given a history of interactions.
- We develop model-based statistical analysis of simulated and real-world datasets using the obtained dynamic embeddings.
- We conduct a novel and rigorous evaluation of the uncertainty learned through the variational approximation of the posterior.
- We assess to what extent the TGNE can be used to reconstruct missing events in the temporal network.

The paper is organized as follows. In Section II we discuss related work. In Section III we provide a Point Process perspective to Continuous-Time Temporal Networks and introduce the CLPM model in light of these definitions. In Section IV, we detail TGNE, and discuss its scalability. In Section IV-E we detail and discuss the results of our experiments. Finally, in Section VI we outline potential extensions of TGNE.

II. RELATED WORK

Our work builds on previous work on Latent Space and Point Process Modelling of (Temporal) Graphs.

A. LATENT SPACE MODELS

Since Hoff's seminal paper [10], Latent Position Models (LPMs) have been extensively studied [14] and extended to various types of graphs, including weighted graphs and dynamic graphs [16], [29]. The Continuous Latent Position Model (CLPM) [27] further extends this line of research to continuous-time temporal graphs, where the latent positions of nodes are assumed to follow piece-wise linear trajectories in a latent space. Our work builds upon this model and describes a Bayesian approach for estimating the latent trajectories.

B. DYNAMIC GRAPH LAYOUT AND DIACHRONIC EMBEDDING

Dynamic Graph Layout [30] aims to find embedding configurations that not only represent the structural information of the graph but also maintain coherence over time. Similarly, Goel et al. [8] propose Diachronic Embedding, which enables embedding nodes from a knowledge graph into a coherent sequence of latent embeddings for temporal knowledge graph completion. As detailed in the method section, we also enforce temporal coherence by specifying a Gaussian Random Walk prior distribution over the latent trajectories.

C. GAUSSIAN GRAPH EMBEDDING

Recent work has explored the idea of embedding nodes in a graph as Gaussian-distributed points in a latent space, with extensions to dynamic graphs [3], [31]. However, the main focus of this line of research has primarily been on forecasting in discrete-time temporal graphs. In contrast with this, our work aims to provide a Bayesian dimensionality reduction method specifically tailored for temporal graphs in *continuous-time*.

D. POINT PROCESS MODELS FOR GRAPHS

Point Process Modeling of Temporal Graphs, particularly using Hawkes Process models, has emerged as a vibrant field of research [1], [12], [24], [32]. These models characterize the changing rates of events in a network based on latent representations. However, the interaction rates in existing models are typically modulated by static representations of the nodes. Few efforts have been dedicated to combining these Point Process decoders with continuous-time representations, which is a key aspect here.

E. TEMPORAL GRAPH NEURAL NETWORKS

Automatically learning time-varying node feature vectors from time-stamped relational data through encoder-decoder architectures is a very active field of research [11], [15], [28]. Such architectures are evaluated on two classes of tasks. *Interpolation* aims at reconstructing past events, and is mostly evaluated on knowledge graphs at a typically low time-resolution. On the other hand, Temporal GNNs are typically evaluated on their ability to *extrapolate* to the future. In contrast, the method proposed in this paper is a dimensionality-reduction method aimed at capturing both the structure of the graph at a user-specified resolution, along with uncertainty on the latent node representations.

TABLE 1. Notation summary.

Symbol	Meaning
$\mathcal{U}, \mathcal{E} \subset \mathcal{U} \times \mathcal{U}$	Set of nodes and set of potential edges
$\mathcal{T}([a,b])$	Set of events happening $t = a$ and $t = b$
$\mathcal{T}_{ij}([a,b])$	Set of timestamps of the interaction between i and j occurring in the time-interval $[a,b]$
$t \in [0,1]$	Normalized timestamps
$w_m = (i_m, j_m, t_m)$	m-th event
$\mathbf{Y}_{ij}, \boldsymbol{Y}_{ij}$	Counting process and counting function of events between i and j
$\mathcal{PP}(\lambda)$	Poisson Process with rate function λ
$\lambda_{ij}, \Lambda_{ij}$	Event Rate and Cumulative Rate
K	Number of changepoints
η_k	k-th changepoint
$\boldsymbol{z}_i^{(k)} \triangleq z_i(\eta_k)$	Latent positions of node i at time η_k
$\boldsymbol{I}_d$	$d \times d$ identity matrix
τ_0, τ	Initial and Transition prior scale on the embeddings
$p_{\boldsymbol{z}}$	Prior distribution
q_ϕ	Approximate posterior distribution
$\mathcal{N}(\mu, \sigma)$	Normal distribution with mean μ and var. σ^2
$\mu_i^{(k)}$	Variational mean of latent position of node i at time η_k
$\sigma_i^{(k)}$	Variational variance of latent position of node i at time η_k

III. PRELIMINARIES

In this section, Temporal Networks are defined from a Point Process point of view. Then the Poisson Process is defined: a particular type of point process that is used in this paper as a generative distribution of the data. Finally, we summarize the Continuous Latent Position Model (CLPM) in light of this theoretical background.

A. NOTATIONS

1) CONTINUOUS-TIME TEMPORAL NETWORKS

Let $\mathcal{U}$ denote a set of nodes, and $\mathcal{E} \subset \mathcal{U} \times \mathcal{U}$ a set of possible edges. In the current work, a **Temporal Network** is defined as a time-ordered sequence of relational events $\mathcal{T}([0,1]) = \{w_m = (i_m, j_m, t_m) | m = 1, \dots, M\}$, where M is the number of events, $0 < t_1 < \cdots < t_M < 1$ is an ordered sequence of pairwise distinct, positive time stamps, and i_m and j_m are the source and destination nodes respectively. The time-stamps are normalized to the interval $[0,1]$. For any node pair $i, j \in \mathcal{E}$, and for any $0 \leq a < b \leq 1$, we denote $\mathcal{T}_{ij}([a,b])$ the set of interaction times between i and j that occur in the interval $[a,b]$. For each $(i,j) \in \mathcal{E}$ we define the function $t \mapsto \boldsymbol{Y}_{ij}(t) \in \mathbb{N}$ that counts the number of interactions between i and j before time t. We assume that the edge-level counting functions are samples from simple point processes, and we will denote $t \mapsto \mathbf{Y}_{ij}(t)$ the *counting process* generating the time function $t \mapsto \boldsymbol{Y}_{ij}(t)$.

2) POISSON PROCESSES

A Poisson Point Process on the interval $[0,1]$ is a random variable that, when sampled from, yields a set of arrival times $t_1, \dots, t_m$. Such a random variable is governed by its rate function $\lambda : [0,1] \mapsto \mathbb{R}_+^*$, defined such that for any interval $[a,b] \subset [0,1]$, the expected number of arrival times that fall into $[a,b]$ is given by the rate measure:

$$\Lambda([a,b]) \triangleq \mathbb{E}[\mathbf{Y}(b) - \mathbf{Y}(a)] = \int_a^b \lambda(s) ds.$$

In other words, $\lambda(t)$ can be viewed as the expected number of events occurring in the interval $[t, t + dt[$. For a given rate function $\lambda : [0,1] \to \mathbb{R}_+^*$, we will write $\mathbf{Y} \sim \mathcal{PP}(\lambda)$ to express that $\mathbf{Y}$ follows a Poisson Process distribution with rate function λ. The likelihood of observing the arrival times $t_1, \dots, t_m$ under a Poisson Process of rate function λ is: $p(\{t_1, \dots, t_m\}; \lambda) = \exp(-\Lambda([0,1])) \prod_{i=1}^{m} \lambda(t_i)$.

Remark 1: This can also be written in the following exponential family form:

$$p(\{t_1, \dots, t_m\}; \lambda) = \exp\left(\int_0^1 \log \lambda(s) d\boldsymbol{Y}(s) - \Lambda([0,1]) \right),$$

where $\boldsymbol{Y}(t) = \sum_{i=1}^m \mathbb{1}_{t_i < t}$ is the counting function representing the arrival times and $\int_0^1 \log \lambda(s) d\boldsymbol{Y}(s)$ is the Stieltjes integral of the log rate with respect to $\boldsymbol{Y}$. While the natural parameter of this exponential family is the function $s \mapsto \log \lambda(s)$, $\boldsymbol{Y}$ can be interpreted as the sufficient statistics. Thus the canonical link function is the log in that case. The second term in the exponential is in turn the log-partition function of the distribution. This exponential form makes the Poisson Process a natural candidate as a generative model in a continuous-time extension of the Latent Space Distance Model [10].

B. THE CONTINUOUS-TIME LATENT SPACE MODEL

1) GENERAL SUMMARY

The Continuous-time Latent Space Model (CLPM) can be summarized as follows. Let $\mathcal{M}$ be an embedding space (typically $\mathcal{M} = \mathbb{R}^d$ with d a small latent space dimension), $\mathcal{Z} = \mathcal{C}([0,1], \mathcal{M})$ the set of continuous trajectories in that latent space and $\mathcal{C}([0,1], \mathbb{R}_+^*)$ the set of positive continuous functions on $[0,1]$. Let p_z be a prior distribution over $\mathcal{Z}$, and $f : \mathcal{Z} \times \mathcal{Z} \to \mathcal{C}([0,1], \mathbb{R}_+^*)$ be a *similarity function* mapping any pair of trajectories $\boldsymbol{z}, \boldsymbol{z}'$ to a positive rate function $f(\boldsymbol{z}, \boldsymbol{z}') = (t \mapsto f_{\boldsymbol{z},\boldsymbol{z}'}(t))$. The model supposes that the edge-level interaction times are generated independently conditioned on the latent trajectories, according to the following generative process:

$$\forall i, \mathbf{z}_i \sim p_z,$$
$$\forall i, j, \mathbf{Y}_{ij} | \boldsymbol{z}_i, \boldsymbol{z}_i \sim \mathcal{PP}(f(\boldsymbol{z}_i, \boldsymbol{z}_j)),$$

Examples of such a model include the distance model, where the similarity function is given by $f_{\boldsymbol{z},\boldsymbol{z}'}(t) = \exp(\beta - ||z_i(t) - z_j(t)||^2)$ and the dot product model corresponding to $f_{\boldsymbol{z},\boldsymbol{z}'}(t) = \exp(\beta + \langle z_i(t), z_j(t) \rangle)$.

2) A PIECEWISE LINEAR ASSUMPTION

Rastelli and Corneli [27] propose to constrain the trajectories to be piece-wise linear to make the model tractable. The

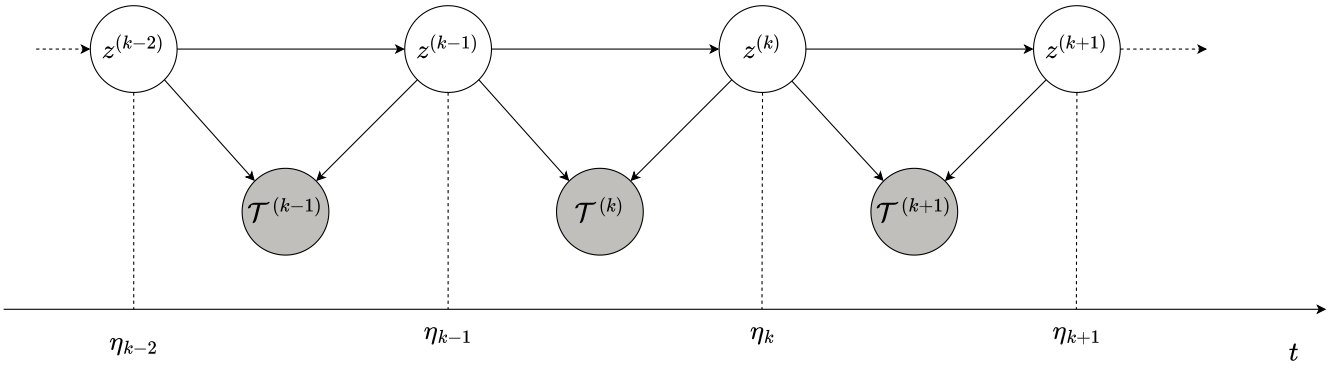


FIGURE 1. Probabilistic Graphical Model summarizing the CLPM. $\mathcal{T}^{(k)} \triangleq \mathcal{T}(I_k)$ is the history of interactions happening in the time interval $I_k = [\eta_{k-1}, \eta_k]$. $z^{(k)}$ are the snapshots of latent positions at time η_k. The chunks of history $\mathcal{T}^{(k)}$ are conditionally independent given the latent positions at the boundaries of the interval I_k ($z^{(k-1)}$ and $z^{(k)}$).

observation interval $[0, 1]$ is partitioned into a set of K intervals $I_1, \ldots, I_K$ with $I_k = [\eta_{k-1}, \eta_k]$, resulting in $K + 1$ changepoints $0 = \eta_0 < \eta_1 < \eta_2 \ldots < \eta_K = 1$. The latent trajectories are then assumed to be linear on each interval I_k. Formally, for each node i, interval k and coefficient $s \in [0, 1]$, the latent position z_i at time $t = (1 - s)\eta_{k-1} + s\eta_k \in I_k$ is:

$$z_i((1 - s)\eta_{k-1} + s\eta_k) = (1 - s)z_i(\eta_{k-1}) + sz_i(\eta_k).$$

Thus, the i-th trajectory is fully determined by its successive positions at the changepoints $\{z_i(\eta_k)|k = 0, \ldots, K\}$, which means that only $(K + 1) \times d$ variables are needed to describe it. The positions at the changepoints are referred to as **critical points** in the following, and denoted $z_i^{(k)} \triangleq z_i(\eta_k)$.

3) LOG-LIKELIHOOD OF THE CLPM

For each node pair (i, j) and each interval I_k, let $\boldsymbol{Y}_{ij}^{(k)}$ be the number of interactions between i and j that occur in the interval I_k. The associated random variables $\mathbf{Y}_{ij}^{(k)}$ are independent across node-pair and intervals, conditioned on the latent trajectories. Moreover, $\boldsymbol{Y}_{ij}^{(k)}$ only depends on the latent positions of i and j at the boundaries of the interval I_k, namely $\{z_i^{(k-1)}, z_i^{(k)}, z_j^{(k-1)}, z_j^{(k)}\}$. The independence structure of the CLPM is summarized in Figure 1. The negative log-likelihood conditioned on the latent positions thus decomposes as follows:

$$\log p(\boldsymbol{Y}|z) = -\sum_{i \neq j} \sum_{k=1}^{K} \log p(\boldsymbol{Y}_{ij}^{(k)}|z_i^{(k-1)}, z_i^{(k)}, z_j^{(k-1)}, z_j^{(k)}).$$

The terms in this decomposition are the following Poisson Process log-likelihoods:

$$-\log p(\boldsymbol{Y}_{ij}^{(k)}|z_i^{(k-1)}, z_i^{(k)}, z_j^{(k-1)}, z_j^{(k)}) = \Lambda_{ij}(I_k) - \sum_{t_{ij} \in \mathcal{T}_{ij}(I_k)} \log(\lambda_{ij}(t_{ij})). \quad (1)$$

Note that while we describe the log-likelihood in the directed case here, the undirected log-likelihood can be obtained by dividing the log-likelihood by 2, as each interaction for an undirected edge i, j would be accounted for twice in the expression above.

While the second term in 1 can be calculated directly from the parameters, the cumulative rate $\Lambda_{ij}(I_k)$ is more difficult to evaluate. We describe two options for calculating this term: the closed form already described in [27], and an approximate form based on a Riemann sum.

4) CLOSED FORM [27]

In the particular case of the Euclidean Distance model, the cumulative rate adopts the following closed form:

Theorem 1: Let

- $\Delta_{ij}(\eta_k) = z_i(\eta_k) - z_j(\eta_k)$ be the vector difference between the embeddings of node i and j,
- $\Phi : t \mapsto \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{u} e^{-\frac{u^2}{2}} du$ be the standard Normal $\mathcal{N}(0, 1)$ cumulative distribution function,
- $\sigma = \frac{1}{\sqrt{2}||\Delta_{ij}(\eta_k) - \Delta_{ij}(\eta_{k+1})||}$,
- $\mu = \frac{\langle \Delta_{ij}(\eta_k), \Delta_{ij}(\eta_k) - \Delta_{ij}(\eta_{k+1}) \rangle}{||\Delta_{ij}(\eta_k) - \Delta_{ij}(\eta_{k+1})||}$,
- $a = ||\Delta_{ij}(\eta_k)||^2 - \frac{\langle \Delta_{ij}(\eta_k), \Delta_{ij}(\eta_k) - \Delta_{ij}(\eta_{k+1}) \rangle^2}{||\Delta_{ij}(\eta_k) - \Delta_{ij}(\eta_{k+1})||^2}$.

Then the cumulative event rate between i, j writes

$$\Lambda_{ij}(I_k) = (\eta_{k-1} - \eta_k) \exp(\beta - a)\sqrt{2\pi}\left[\Phi(\frac{1 - \mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})\right]$$

While a proof of this theorem is provided in previous work [27], in the supplementary material we provide an alternative proof that can be reproduced for any case where the log of the rate can be expressed as a second-order spline in time, i.e. such that its expression on each subsequent interval is a second-order polynomial.

5) RIEMANN APPROXIMATION OF THE CUMULATIVE RATE

In the case where the rate function is not a second-order spline, we propose to approximate it simply using a Riemann sum:

$$\Lambda_{ij}(I_k) = \int_{\eta_{k-1}}^{\eta_k} \lambda_{ij}(s)ds \approx \frac{1}{R} \sum_{r=1}^{R} \lambda_{ij}(\eta_{k-1} + \frac{r - 1}{R}\eta_k)$$

where R is a pre-specified resolution parameter. This approximation allows implementing an inference procedure agnostic to the type of similarity function used. For instance, using this approximation makes it easy to consider different latent geometries such as hyperbolic or spherical embeddings.

IV. METHOD

In this section, we provide an overview of the proposed TGNE approach for performing Bayesian inference on the latent critical points, given a history of interactions.

A. PRIOR DISTRIBUTION

To reflect time continuity in the latent trajectories of the CLPM, a prior distribution is needed. The Gaussian Random Walk prior introduced in [27] biases the inference towards time-coherent and reasonably scaled configurations, promoting slowly evolving trajectories while faithfully representing the network's structure. This prior is defined for any node $i \in \{1, \dots, n\}$ and time step $k \in \{0, \dots, K\}$, as the cumulative sum of independent Gaussian increments:

$$z_i(\eta_k) = \tau_0 \epsilon_0 + \sum_{l=1}^{k} \sqrt{\eta_l - \eta_{l-1}}\, \tau \epsilon_l,$$

where $\epsilon_i \sim \mathcal{N}(0, I_d)$. The initial scale τ_0 controls the overall spread of the latent trajectories in the embedding space. The transition scale parameter τ governs the amount of allowed variation between consecutive time steps. Additionally, the variance of the Gaussian increments increases linearly with the step size $\eta_{k+1} - \eta_k$. Note that by taking infinitely small step sizes, this prior converges to a Brownian Motion in the embedding space. In our implementation, we choose a constant step size $\eta_{k+1} - \eta_k = \frac{1}{K}$, where K is the number of steps. Moreover, we select an initial scale equal to the transition scale: $\tau_0 = \tau$. This yields two hyperparameters: the scale τ and the number of changepoints (ticks) K.

B. VARIATIONAL INFERENCE ON THE CRITICAL POINTS

The objective of TGNE is to evaluate the intractable posterior distribution $p(z|\boldsymbol{Y}) \propto p(\boldsymbol{Y}|z)p(z)$ given the data $\boldsymbol{Y}$. To achieve this, we use a mean-field variational approach, where we define the following variational distribution that factorizes over nodes and changepoints as a product of independent Normal distributions:

$$q_\phi(z) = \prod_{i=1}^{n} \prod_{k=0}^{K} \mathcal{N}(z_i^{(k)}; \mu_i^{(k)}, (\sigma_i^{(k)})^2 \boldsymbol{I}_d),$$

We aim to minimize the Kullback-Leibler divergence $KL(q_\phi||p(.|\boldsymbol{Y}))$ between the variational distribution and the posterior. This is equivalent to minimizing the negative Evidence Lower Bound (ELBO):

$$\mathcal{L}(\phi) = KL(q_\phi||p(.|\boldsymbol{Y})) - \mathbb{E}_{z\sim q_\phi}[\log(p(\boldsymbol{Y}|z))],$$

The KL Divergence term can be written as shown in Theorem 2, and proved in Appendix D.

Theorem 2:

$$KL(q_\phi||p(.|\boldsymbol{Y})) = \sum_{i=1}^{n} \Bigg[\frac{||\mu_i^{(0)}||^2}{2\tau_0^2} + d \sum_{k=0}^{K} \left(\log(\frac{\sigma_i^{(k)}}{\tau}) + \frac{\tau^2}{(\sigma_i^{(k)})^2} - \frac{1}{2} \right) + \sum_{k=1}^{K} \frac{||\mu_i^{(k)} - \mu_i^{(k-1)}||^2 + (\sigma_i^{(k-1)})^2}{2\tau^2} \Bigg].$$

Following common practices in variational inference, the expected log-likelihood term is approximated using a single Monte-Carlo sample $\tilde{z} \sim q_\phi$:

$$\mathbb{E}_{z\sim q_\phi}[\log(p(\boldsymbol{Y}|z))] \approx \log(p(\boldsymbol{Y}|\tilde{z})).$$

Reparameterization [18] allows backpropagating through the latter sampling operation, by mapping standard Normal-distributed samples to the latent space through an invertible function of the variational parameters. It is used here to obtain the following differentiable loss, which can be optimized using standard gradient descent algorithms such as ADAM [17]:

$$\begin{aligned} \mathcal{L}(\phi) &\overbrace{\qquad\qquad\qquad\qquad}^{\text{Calculated using a single sample } \tilde{z} \sim q_\phi} \\ &\approx \sum_{i,j} \sum_{k=1}^{K} \Lambda_{ij}(I_k) - \sum_{t_{ij} \in \mathcal{T}_{ij}(I_k)} \log(\lambda_{ij}(t_{ij})) \\ &\quad + \sum_{i=1}^{n} \Bigg[\frac{||\mu_i^{(0)}||^2}{2\tau_0^2} + d \sum_{k=0}^{K} \left(\log\left(\frac{\sigma_i^{(k)}}{\tau}\right) + \frac{\tau^2}{(\sigma_i^{(k)})^2} - \frac{1}{2} \right) \\ &\quad + \sum_{k=1}^{K} \frac{||\mu_i^{(k)} - \mu_i^{(k-1)}||^2 + (\sigma_i^{(k-1)})^2}{2\tau^2} \Bigg]. \end{aligned} \tag{2}$$

C. EFFECT OF THE HYPERPARAMETERS

The proposed method has four hyperparameters: the dimension d, the number of changepoints K, the initial scale τ_0, and the scale τ. The number of changepoints K controls the time resolution of the latent trajectories. It should be adapted to how frequently we expect the nodes' states to change in our dataset. The initial scale τ_0 controls the scale of the initial latent positions $z_i^{(0)}$. Finally, the scale τ is a temperature parameter that controls the deviation of the latent positions between frames, namely $||z_i^{(k+1)} - z_i^{(k)}||$. To illustrate its effect, in Figure 2 it can be seen that for $\tau = 50.0$ the frames are not constrained to be close to each other, and the latent positions can change drastically between frames. On the other hand, for $\tau = 1.0$, the latent positions are constrained to be close to each other, and the frames are more similar to each other.

D. IMPLEMENTATION

We implemented our method in Pyro, a Pytorch-based probabilistic programming language [2]. This effect

handler-oriented programming language allows one to define the model as a Python function. The execution trace of the function can then be read and decorated by effect handlers, allowing one to define high-level probabilistic operations such as conditioning, or performing Stochastic Variational Inference. To optimize the variational parameters ϕ and the bias term β, we use the ADAM algorithm [17] with learning rates $\gamma = 0.01$ and $\gamma = 0.00001$ respectively.

E. SCALABILITY

We discuss two strategies to scale the method to networks with a large number of nodes: node-batching and negative sampling.

As the log likelihood term is a sum of terms over all source nodes, **node-batching** can be implemented by computing the loss and gradients on a subset of the nodes at each iteration, and then averaging the gradients over the whole dataset.

The log-likelihood decomposes as a sum of contributions from **positive** node pairs (interacting at least once) and **negative** node pairs that never interact. However, most of the node pairs in the network never interact, and thus the information conveyed by the negative pairs is redundant. This opens up the possibility of **negative sampling** which may dramatically speed up inference on networks with many nodes. We propose the following strategy, akin to the case-control approximate likelihood introduced in [26]: for each node i, we sample K nodes j such that (i, j) never interact in the network. We denote $\mathcal{P}(i)$ the set of nodes that connect with i at least once in the event history, and $\mathcal{N}(i)$ a random subset of the set of nodes that never connected with i. The log-likelihood can be approximated as:

$$\begin{aligned}\log(p(\boldsymbol{Y}|\tilde{\boldsymbol{z}})) &\approx \sum_{i\in\mathcal{U}}\left[\sum_{j\in\mathcal{P}(i)}\int_0^1 \lambda_{ij}(s)ds - \sum_{\tau\in\mathcal{T}_{ij}}\log(\lambda_{ij}(\tau))\right] \\ &+ \frac{|\mathcal{U}\setminus\mathcal{P}(i)|}{|\mathcal{N}(i)|}\left[\sum_{j\in\mathcal{N}(i)}\int_0^1 \lambda_{ij}(s)ds\right]\end{aligned}$$

V. EXPERIMENTS

We performed various experiments to answer the following research questions. First, we evaluate the uncertainty of the latent positions on simulated data, and on real-world datasets. Next, we try to understand qualitatively how the parameters of the model affect the resulting latent positions. Finally, we try to understand to what extent the TGNE method allows reconstructing the events of unobserved edges, based on the event history of the observed edges. All the experiments were run on an Intel(R) Core(TM) i7-9850H CPU @ 2.60GHz, with 1TB RAM.

Datasets: In our experiments, we used a simulated dataset, as well as four real-world datasets, for which we provide a brief description below. The **HighSchool** dataset [7] is a contact network of student in a French preparatory class High School in Marseille. Their interactions were recorded using wearable devices over 9 days. The resulting embeddings are shown in Figure 3. The **MIT Reality Mining** Dataset [6] is a dataset of face-to-face contacts between participants of an experiment ran by members of MIT media Lab. The data was collected over the course of around 9 months, between 2004 and 2005. The obtained embeddings for this dataset are shown in Figure 3b. The **Workplace** dataset is a dataset of face-to-face contacts between employees in a workplace [9]. Their interactions were recorded on 11 days (2013/06/24 to the 2013/07/05). In this work we focus on the first day of interactions. The **UCI** dataset is a Facebook-like, unattributed online communication network among students of the University of California at Irvine, along with timestamps with the temporal granularity of seconds. We used the preprocessed version from the recent DGB Benchmark [25]. A summary of the datasets is shown on Table 2, along with the associated runtimes of the TGNE method.

TABLE 2. Statistics on the Datasets, and associated runtime of TGNE for 500 epochs.

Dataset	HighSchool	RealityMining	Workplace	UCI
Nodes	180	106	92	1899
Unique edges	758	5756	755	20295
Events	9957	779868	9827	59834
Runtime (s)	45	109	40	542

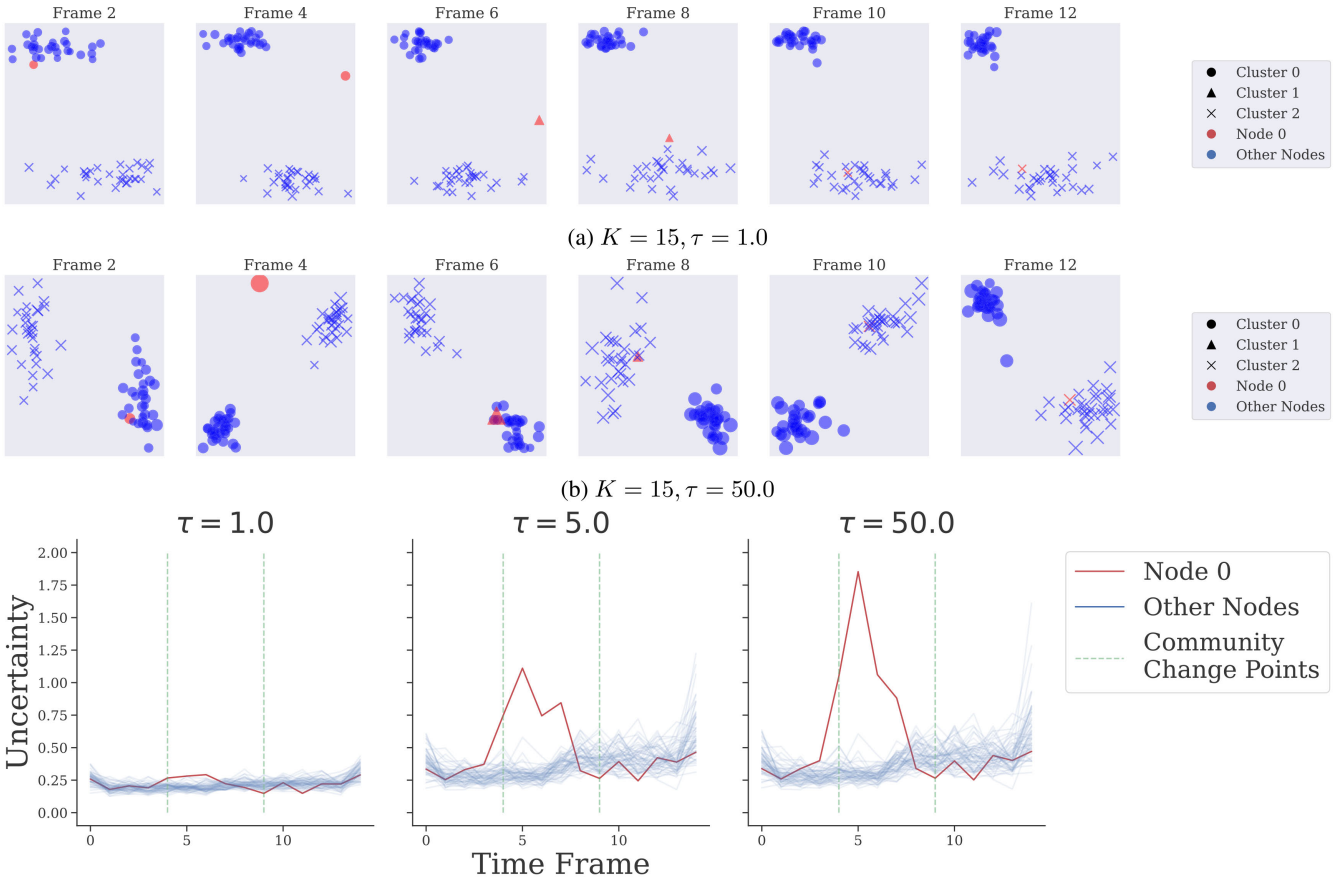
A. EXAMPLE ON DATA SIMULATED USING A STOCHASTIC BLOCK MODEL

We evaluate the estimated uncertainty of the interactions in an example simulated using a Stochastic Block model(SBM), where one node changes community over time, while all the other nodes stay in the same community. This data generation procedure is adapted from [27], but here we focus on the uncertainty aspect. A detailed explanation of the simulation procedure is provided in the Appendix. The resulting Gaussian Embeddings are shown in Figure 2a and 2b, for two sets of hyperparameters. Using a low scale parameter, the positions are located with more precision, and the trajectories evolve in a smoother way between time stamps. This is to be expected since the regularization term is stronger in that case. However, the estimated uncertainty is uniform across nodes in that case. In the high-scale regime, the trajectories evolve more freely between frames, as in that case, the between-frame regularization is weaker. However, the uncertainty (defined as in subsection V-C) of the node that changes community is higher than the uncertainty of the other nodes, as expected. On Figure 2c we show the evolution of the uncertainty of the node that changes community over time, for different value of the scale parameter.

B. UNCERTAINTY EVALUATION

1) NODE-LEVEL UNCERTAINTY

The TGNE method outputs a Gaussian distribution for each node at each individual time stamp. Thus, the uncertainty



(a) $K = 15, \tau = 1.0$

(b) $K = 15, \tau = 50.0$

(c) Uncertainty over time for $K = 15$ and different values of the prior scale τ. The estimated uncertainty increases with the scale parameter. Moreover, for a high value of τ, the node changing community will yield a higher variance around its associated embedding position

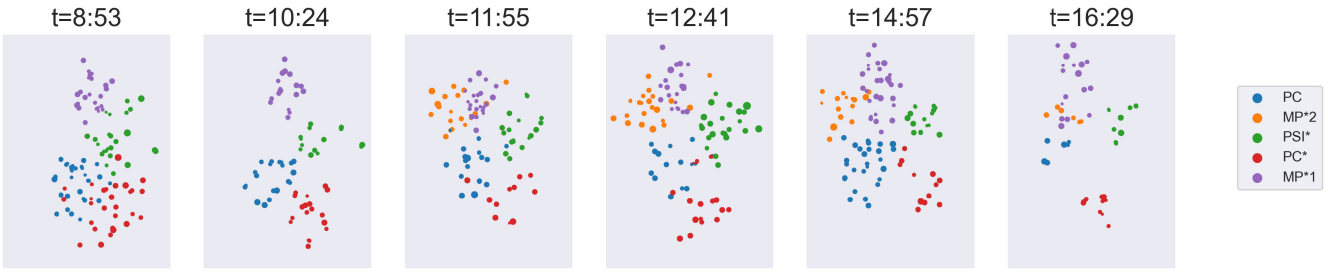
FIGURE 2. Resulting latent positions on synthetic data generated from the Stochastic Block Model. Uncertainty is represented by the size of the markers. From frames 0 to 4, the nodes are divided into two communities (Circles and crosses). Then from frames 5 to 9, node 0 becomes a triangle and forms its own community. During that transition, node 0's uncertainty increases, especially when using a less informative prior ($\tau = 50.0$). Finally, from frames 10 to 15 node 0 becomes a cross.

around the latent positions can be naturally measured through the scale of the variational Normal distribution. While there are multiple potential sources of uncertainty, we empirically assess the impact of the node degree on the uncertainty of the latent positions. Namely, for each node i and each sub-interval I_k, we measure the uncertainty of node i on interval I_k by calculating $u(i,k) = \frac{\sigma_i^{(k)} + \sigma_i^{(k+1)}}{2}$ and conversely calculate the number of interactions $\boldsymbol{N}_i(I_k)$ of the node on this interval. Moreover, we relate the uncertainty associated with a node on a given interval to the average Euclidean distance to its neighbors on the same interval. In order to display how these different values are related, in Figure 4 we represent the average uncertainty $u(i,k)$ as a function of the average distance to the neighbors within the same interval. A first observation is that higher degree node-interval pairs have less uncertainty. This property is typical of Latent Space Model, and reflects the fact that estimating the latent position is easier for nodes that have more interactions. Stated differently, for a given node pair (i,j) and an interval I_k, there are many more embedding configurations compatible with the fact that i and j do not interact in the interval I_k than with the fact that they interact many times in I_k. Thus, it is natural for the posterior distribution of a node i to be less concentrated when this node has fewer interactions. **Edge-level uncertainty**.

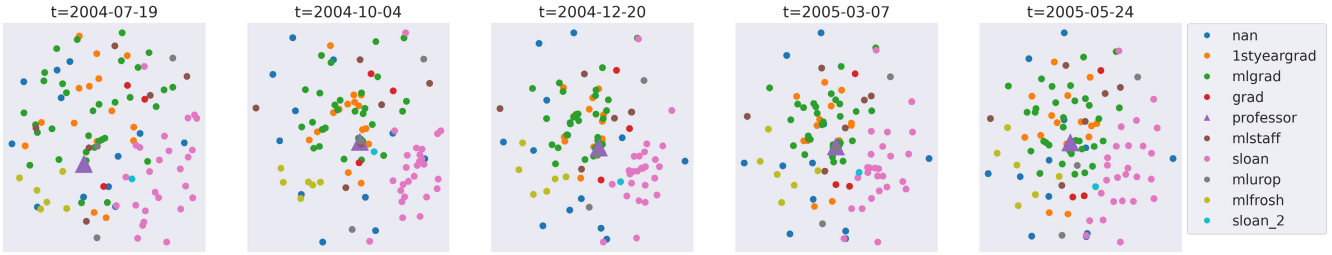
The uncertainty about the node's latent positions can be propagated into a notion of uncertainty on the distribution generating the temporal graph, materialized by the posterior predictive distribution defined by the Poisson Processes $\mathcal{PP}(\tilde{\lambda}_{ij})$ with the random variable $\tilde{\lambda}_{ij}$ being defined as:

$$\tilde{\lambda}_{ij}(t) \triangleq \mathbb{E}[\lambda_{ij}(t)|\boldsymbol{z}]$$

Note that here we get a distribution over the set of joint Poisson Process distributions. While evaluating this posterior predictive distribution is intractable, we can approximate it by sampling B i.i.d. samples from the variational distribution, i.e. $\boldsymbol{z}^{(b)} \overset{iid}{\sim} q_\phi$ for $b = 1, \ldots, B$. Then for each sample $\boldsymbol{z}^{(b)}$ we denote $\lambda_{ij}^{(b)}$ (respectively $\Lambda_{ij}^{(b)}(I_k)$) the rate function (respectively the cumulative rate) obtained by plugging $\boldsymbol{z}^{(b)}$ into the similarity function defined in III-B. We measure the uncertainty associated with edge (i,j) on interval I_k, and denote it $u(i,j,k)$ by calculating the empirical standard



(a) Resulting latent positions on the High School Dataset, with $\tau = 1.0, K = 15$. Only the nodes that appear during each time frame are shown. The color corresponds to the class of the student, while the size is proportional to the uncertainty on their latent positions. Students from similar classes (e.g. Physics/Chemistry students (PC and PC*) cluster together more than with other students (e.g. Mathematics/Physics: MP*1 and MP* 2)). Some students from PSI* (Physics and Engineering Science) still seem to interact with the PC/PC*.



(b) Resulting latent positions on the MIT Reality Mining Dataset, with $K = 15, \sigma = 0.1$. The embedding positions allow distinguishing members of different departments (the people of the Sloan Business school form a clear separate community). Moreover, the professor (represented by a purple triangle) is in the center of the embedding space and over time moves around between the communities he mostly interacts with.

FIGURE 3. Latent Positions obtained on the Highschool Dataset and the MIT Reality Mining Dataset.

deviation of the cumulative rate over the B samples. In our experiments, we found out that the model uncertainty on $\Lambda_{ij}(I_k)$ decreases with the number of interactions for i, j in I_k, that we denote here $N_{ij}(I_k)$. In Figure 4c, we observe that the linear regression slope decreases with the prior scale, suggesting that a less informative prior leads to a stronger correlation between uncertainty and the number of interactions.

There is no generic best choice of the regularization parameter, it will depend on the task. It may for example be trained using cross-validation for predictive tasks, while for unsupervised tasks it may be less straightforward to choose it well. Its effect is nonetheless evident: it introduces a bias-variance trade-off between concentrating the trajectories in the latent space over time (which would increase the **bias**) and modeling the observed interactions in time more closely (thus increasing the **variance**).

2) RELATIONSHIP BETWEEN THE UNCERTAINTY AND THE POISSON RATE

In order to visualize the relationship between the Poisson Rate and the learned notion of uncertainty, we use structured negative sampling: we select one negative event (i, j', t) for each positive event (i, j, t), by swapping the destination node j with a random node j', distinct from i and j. Then we calculate the Poisson Rate for each positive event and associated negative event, and compare it with the uncertainty propagated from the latent positions to the rate function. The results are shown in Figure 5. In general, more extreme Poisson Rates seem to be associated with less uncertainty.

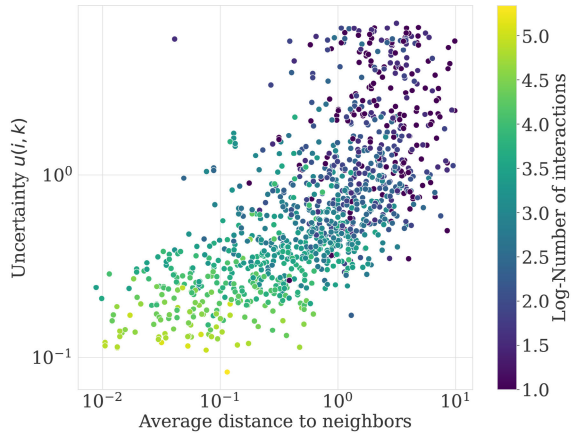
C. TEMPORAL NETWORK RECONSTRUCTION

1) SETUP

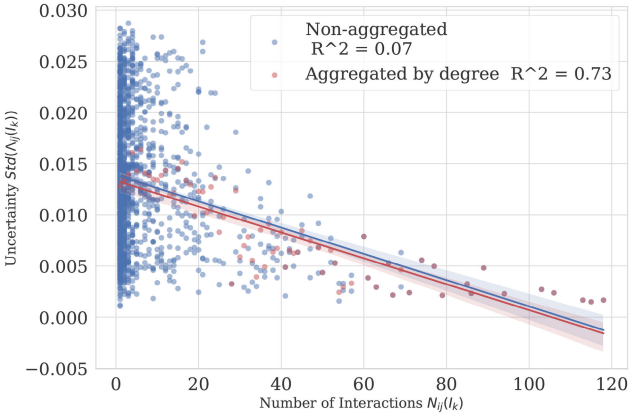
We evaluate TGNE on the task of reconstructing missing interactions from a partially observed continuous-time temporal graph. To do that, we split the edges in the network into train, validation and test sets. Then for each split, and each interval I_k, we predict whether each edge $e = (i, j)$ interacts in the interval I_k, i.e. whether there exists an *interaction* (i, j, t) in history, such that $t \in I_k$. For each interval and each positive edge we sample a single negative edge, thus casting the problem into a binary classification task of the node-pair/interval triplets (i, j, k). For the **HighSchool** dataset and the **UCI** dataset we use 10 % of the edges as test edges, and the rest as train edges. For the **Workplace** dataset we use 30 percent of the edges as test edges, and the rest as train edges.

2) BASELINES

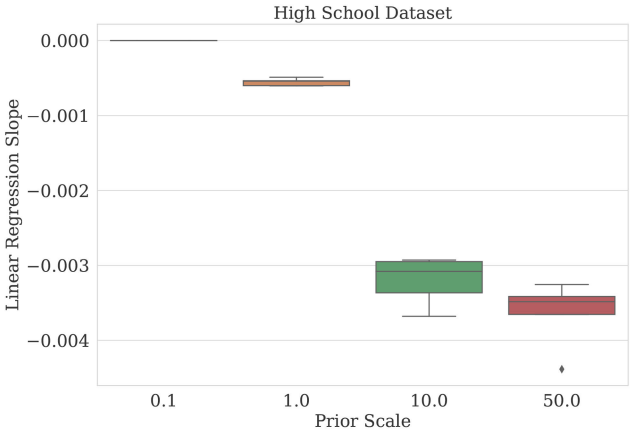
We compare four different approaches for scoring the triplets (i, j, k). For our method, a score is calculated based on a fitted **TGNE** model, as the expected amount of interaction on the interval: $score(i, j, k) = \Lambda_{ij}(I_k)$. A first baseline is derived by postulating a binary, Euclidean Latent Space Distance Model (**LSDM**) [10] on the interactions occurring on each



(a) Log-log plot of the node-level uncertainty $u(i,k)$ as a function of the average distance to the neighbors whithin the same interval, with ($\tau = 50.0, K = 15$).



(b) Edge-level uncertainty $Std(\tilde{\Lambda}_{ij}(I_k))$ as a function of $N_{ij}(I_k)$, with ($\tau = 1.0, K = 15$).



(c) Linear Regression Slope of $Std(\tilde{\Lambda}_{ij}(I_k))$ against $N_{ij}(I_k)$ for different values of the prior scale

FIGURE 4. Relationship between the Node-level uncertainty $u(i,k)$ and Edge-level uncertainty $u(i,j,k)$ and the number of interactions $N_i(I_k)$ and $N_{ij}(I_k)$ respectively, on the High School Dataset.

interval: $score(i,j,k) = \sigma(\beta - ||z_i^{(k)} - z_j^{(k)}||^2)$, where the latent positions $z_j^{(k)}$ are optimized using Maximum Likelihood

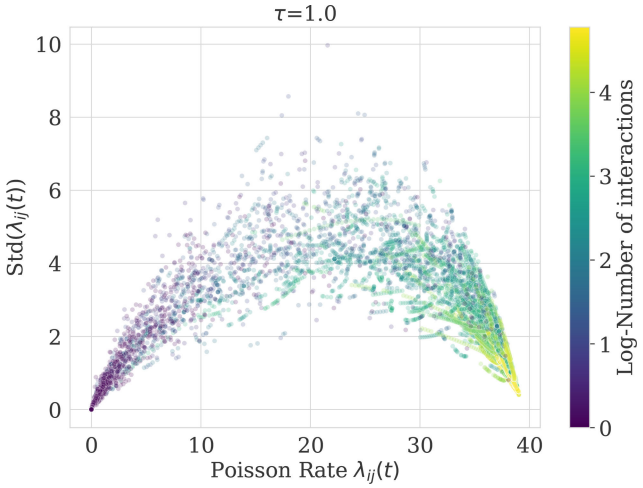


FIGURE 5. Uncertainty vs Poisson Rate on the High School Dataset with ($K = 15, \tau = 1.0$). The Poisson Rate is calculated for each positive event and associated negative event. For each event (u, v, t) we color it by the number of interactions between (u, v) in the interval I_k such that $t \in I_k$. Events with extreme Poisson Rate values are associated with a low uncertainty, while intermediate Poisson Rates are associated with a higher uncertainty.

Estimation. A second baseline is popularity-based prediction, also named Preferential Attachment (**PA**): $score(i,j,k) = deg(i,k) \cdot deg(j,k)$ where $deg(i,k) = \sum_j N_{ij}(I_k)$ is the degree of node i on interval k. Finally, we include a Random Baseline (**Random**), that calculates a random score for each triplet. In order to discuss more precisely the regularizing effect of the prior for Network reconstruction, techniques such as tensor decomposition could have been explored, however, since TGNE is derived from Latent Space Models, we decided to stick with this class of models in this work.

We leverage the probabilistic nature of the TGNE method to analyze the model uncertainty.

3) RESULTS

The results are provided in Table 3. On the High School Dataset, it can be seen that while a binary Latent Space Distance Model could be used to predict the presence/absence of links, its resulting configuration of node embeddings overfits the training data, and thus does not perform well on the test data. In contrast, the embeddings obtained using TGNE perform worse on the training set, but much better on the test set. This showcases the benefits of the regularization term on the predictive abilities of the model.

VI. DISCUSSION AND FUTURE WORK

In this paper, we discuss the performances of TGNE and its ability to capture the uncertainty on the estimated latent positions and the ability of the obtained locations to predict the occurrence of edges in successive intervals. However, some open questions remain, which we detail here for future work.

TABLE 3. AUC Results on the different datasets for $K = 15$ intervals.

Dataset	TGNE	LSDM	PA	Random
HighSchool(train)	0.959	**0.999**	0.904	0.497
HighSchool(test)	**0.885**	0.784	0.756	0.524
Workplace(train)	0.901	**1.000**	0.882	0.497
Workplace(test)	**0.702**	0.665	0.672	0.496
UCI (train)	0.972	**0.999**	0.967	0.506
UCI (test)	**0.922**	0.882	0.914	0.505

A. ADAPT THE CHANGEPOINTS TO THE DENSITY OF INTERACTIONS

The number K and the positions of the changepoints $\eta_0, \ldots, \eta_K$ are fixed in the TGNE. As all the interaction times are re-scaled to be between 0 and 1, the constant step size is fixed to $\eta_{k+1} - \eta_k = \frac{1}{K}$. However, adapting the step size to the observed rate of events would naturally produce a more fine-grained representation of the temporal network structure in sub-intervals where more events happen. This appears as a promising avenue to improve model efficiency.

B. IDENTIFIABILITY

In the high-scale regime (see for instance Figure 2b), there is significant rotation of latent configurations from one frame to the next. This is because the model fails to identify the rotations of the configurations. Although this issue is partly mitigated by the effect of the prior, it could potentially be resolved through the use of a Procrustes transform applied to the configurations of trajectories.

C. NODE INDUCTIVITY

The proposed model is transductive, meaning it is limited to the set of nodes that are provided in advance and cannot embed unobserved nodes. In contrast, an alternative approach would be to use amortization, as in seminal works such as [19], to map nodes and their context to Gaussian parameters using a parametric function. This approach would allow predicting trajectories for unobserved nodes, and allow the resulting model to scale to millions of nodes.

D. TIME INDUCTIVITY

As mentioned earlier, the TGNE model could be used to learn dynamics (or distributions thereof) in the embedding space instead of directly learning a sequence of latent distributions in the latent space. This would enable extrapolating the dynamic to future unobserved links. One-step ahead Link Prediction would be a key metric to evaluate the success of such an approach.

E. CONTINUOUS-TIME ENCODER

Finally, related to the previous point, one limitation of the proposed approach is that it relies on a *discrete time encoder* since each node is essentially mapped to a sequence of Gaussian parameters. However, one alternative approach would be to build on [5] to embed the nodes into parameters of a joint stochastic process on the node state and the network state, and using a Point Process Model as a decoder.

VII. CONCLUSION

In the present work, we introduce a principled approach to Temporal Graph embedding that leverages Variational inference to infer latent distributions on node trajectories from an observed temporal network. This is in contrast with traditional temporal graph embedding methods, where only a trajectory of points per node is usually estimated. Our results show that in the case where the prior distribution is not restrictive enough, the uncertainty coming from this greater degree of freedom in the latent space can be partially captured back in the scale parameter of the estimated normal distributions. On top of that, the reconstruction experiment showcases the need for regularization in the case of temporal graph embeddings, as it makes the obtained trajectories more easily readable visually, but also leads to better reconstruction results. Finally, we strongly believe that model-based uncertainty estimation, a critical novelty of TGNE, could not only enhance predictive performances on Temporal Networks but also extend to applications like anomaly detection and more interpretable visualizations.

ETHICAL STATEMENT

There are no ethical issues.

APPENDIX A
CODE

An implementation of TGNE is provided in the supplementary material.

The datasets can be downloaded from the following urls:

- The Reality Mining Dataset can be downloaded here. The user needs to be authorized before being allowed access to the data.
- The High School contact network dataset is publicly available here.
- The Workplace dataset can be found here.
- The UCI dataset can be downloaded from here.

APPENDIX B
GENERATION PROCEDURE FOR THE SIMULATED DATA

The simulation procedure is a temporal version of the Stochastic Block Model. In this simulation, a network of 60 nodes (indexed from 0 to 59) is observed during 3 segments of time of equal duration, denoted I_1, I_2, I_3:

- In the first segment, the nodes are split into two clusters: the nodes from 0 to 29 go into cluster C_0, while nodes 30 to 59 go into C_1.
- In the second segment, node 0 goes into its own cluster C_2, while the other nodes stay in their respective clusters.
- In the third segment, node 0 goes into the cluster C_1, thus only two clusters are present at this time.

Based on these cluster assignments, the interactions are generated using a Stochastic Block Model: we fix inter and intra-cluster interaction rate, and proceed as follows. For each

segments and each node pair (i, j), let $r(i, j, s)$ be the rate of interaction between i and j during segment s, as defined by the SBM. We first generate a number of interactions $N(i, j, s) \sim Poisson(r(i, j, s))$, and then sample $N(i, j, s)$ timestamps uniformly distributed over the time segment I_s. Finally, the interactions obtained on the three segments are concatenated and sorted by ascending timestamp.

APPENDIX C
CALCULATION OF THE CUMULATIVE RATE ON AN INTERVAL WHERE THE TRAJECTORIES ARE LINEAR

Let's calculate the integral $\int_{\eta_{k-1}}^{\eta_k} \lambda_{ij}(s)ds$. Under the piece-wise linear assumption, the trajectory of node i at time $s = ((1-t)\eta_{k-1} + t\eta_k) \in I_k$ (with $t \in [0, 1]$) can be written as:

$z_i((1-t)\eta_{k-1} + t\eta_k) = (1-t)z_i(\eta_{k-1}) + tz_i(\eta_k)$. Based on that, let's rewrite the rate function in a way that makes it easier to integrate.

The log of the rate writes:

$$\begin{aligned} log\lambda_{ij}(s) &= \beta - ||z_i(s) - z_j(s)||^2 \\ &= \beta - \gamma_{ij}(s) \end{aligned}$$

where, denoting $\Delta_{ij}(\eta_k) = z_i(\eta_k) - z_j(\eta_k)$, γ_{ij} is defined as

$$\gamma_{ij}((1-t)\eta_k + t\eta_{k+1}) = ||(1-t)\Delta_{ij}(\eta_k) + t\Delta_{ij}(\eta_{k+1})||^2. \tag{3}$$

In particular, γ_{ij} is a second-order polynomial in t. Our goal now is to express γ_{ij} as the log of the density of a normal distribution.

More precisely, let's try to write it under the form

$$\gamma_{ij}(s) = a + \frac{(t-\mu)^2}{2\sigma^2} \tag{4}$$

for some coefficients a, μ, σ, where $s = (1-t)\eta_{k-1} + t\eta_k$

On the one hand, developing the expression 3 yields:

$$\begin{aligned} &\gamma_{ij}(s) \\ &= t^2\left[||\Delta_{ij}(\eta_k)||^2 + ||\Delta_{ij}(\eta_{k+1})||^2 - 2\langle\Delta_{ij}(\eta_k), \Delta_{ij}(\eta_{k+1})\rangle\right] \\ &\quad + t\left[-2||\Delta_{ij}(\eta_k)||^2 + 2\langle\Delta_{ij}(\eta_k), \Delta_{ij}(\eta_{k+1})\rangle\right] \\ &\quad + ||\Delta_{ij}(\eta_k)||^2 \end{aligned}$$

On the other hand, developing equation 4 yields:

$$\gamma_{ij}(s) = a + t^2(\frac{1}{2\sigma^2}) + t(-\frac{\mu}{\sigma^2}) + \frac{\mu^2}{2\sigma^2}$$

Identifying the coefficients of the polynomial, we get the following system of equations:

$$\frac{1}{2\sigma^2} = ||\Delta_{ij}(\eta_k) - \Delta_{ij}(\eta_{k+1})||^2 \tag{5}$$

$$\frac{\mu}{2\sigma^2} = \langle\Delta_{ij}(\eta_k), \Delta_{ij}(\eta_k) - \Delta_{ij}(\eta_{k+1})\rangle \tag{6}$$

$$a + \frac{\mu^2}{2\sigma^2} = ||\Delta_{ij}(\eta_k)||^2 \tag{7}$$

Finally, solving the system for a, μ and σ yields:

$$\begin{aligned} \sigma &= \frac{1}{\sqrt{2}||\Delta_{ij}(\eta_k) - \Delta_{ij}(\eta_{k+1})||} \\ \mu &= \frac{\langle\Delta_{ij}(\eta_k), \Delta_{ij}(\eta_k) - \Delta_{ij}(\eta_{k+1})\rangle}{||\Delta_{ij}(\eta_k) - \Delta_{ij}(\eta_{k+1})||^2} \\ a &= ||\Delta_{ij}(\eta_k)||^2 - \frac{\langle\Delta_{ij}(\eta_k), \Delta_{ij}(\eta_k) - \Delta_{ij}(\eta_{k+1})\rangle^2}{||\Delta_{ij}(\eta_k) - \Delta_{ij}(\eta_{k+1})||^2} \end{aligned}$$

We can conclude by using two changes of variables:

$$\begin{aligned} &\int_{\eta_{k-1}}^{\eta_k} \lambda_{ij}(s)ds \\ &= \int_{\eta_{k-1}}^{\eta_k} \exp(\beta - \gamma_{ij}(s))ds \\ &= \exp(\beta)(\eta_k - \eta_{k-1}) \int_0^1 \exp(-\left(a + \frac{(t-\mu)^2}{2\sigma^2}\right))dt \\ &\quad \times (s = (1-t)\eta_{k-1} + t\eta_k \text{ such that } ds = (\eta_k - \eta_{k-1})dt) \\ &= \exp(\beta - a)(\eta_k - \eta_{k-1})\sigma \int_{-\frac{\mu}{\sigma}}^{\frac{1-\mu}{\sigma}} \exp(-\frac{u^2}{2})du \\ &\quad \times \text{ Where we set } u = \frac{t-\mu}{\sigma} \text{ such that } du = \frac{dt}{\sigma} \\ &= \exp(\beta - a)(\eta_{k-1} - \eta_k)\sigma\sqrt{2\pi}[\Phi(\frac{1-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})]. \end{aligned}$$

APPENDIX D
KL DIVERGENCE BETWEEN A GAUSSIAN MARKOV CHAIN AND A PRODUCT OF INDEPENDANT GAUSSIANS

Let $x_1, \ldots, x_T$ be some random variables and the two distributions q and p defined as:

$$\begin{aligned} q(x) &= \prod_{t=1}^{T} q_t(x_t) \\ p(x) &= p(x_1)\prod_{t=2}^{T} p_t(x_t|x_{t-1}) \end{aligned}$$

Then

$$KL(q||p) = KL(q_1, p_1) + \sum_{t=1}^{T-1} \mathbb{E}_{x_t \sim q_t}[KL(q_{t+1}||p_{t+1}(.|x_t))] \tag{8}$$

In particular, when

$$q_t(x_t) = \mathcal{N}(x_t; \mu_t, \sigma_t^2 \boldsymbol{I}_d)$$

and

$$\begin{aligned} p(x) &= \mathcal{N}(x_1; \nu_1, \tau_1^2 \boldsymbol{I}_d)\prod_{t=2}^{T} \mathcal{N}(x_t; x_{t-1}, \tau_t^2 \boldsymbol{I}_d) \\ p_1(x_1) &= \mathcal{N}(x_1; \nu_1, \tau_1^2 \boldsymbol{I}_d) \end{aligned}$$

and

$$p_t(x_t|x_{t-1}) = \mathcal{N}(x_t; x_{t-1}, \tau_t^2 \boldsymbol{I}_d)$$

Moreover, the KL Divergence between two d-dimensional Gaussian distributions is given by:

$$KL(\mathcal{N}(\mu_1, \sigma_1^2 \boldsymbol{I}_d)||\mathcal{N}(\mu_2, \sigma_2^2 \boldsymbol{I}_d)) = \frac{||\mu_2 - \mu_1||^2}{2\sigma_2^2} + d\left[\log(\frac{\sigma_2}{\sigma_1}) + \frac{\sigma_2^2}{\sigma_1^2} - \frac{1}{2}\right]$$

this yields:

$$KL(q_1||p_1) = \frac{||\mu_1 - \nu_1||^2}{2\tau_1^2} + d\left[\log(\frac{\tau_1}{\sigma_1}) + \frac{\tau_1^2}{\sigma_1^2} - \frac{1}{2}\right]$$

$$KL(q_t||p_t(.|x_{t-1})) = \frac{||x_{t-1} - \mu_t||^2}{2\tau_t^2} + d\left[\log(\frac{\tau_t}{\sigma_t}) + \frac{\tau_t^2}{\sigma_t^2} - \frac{1}{2}\right]$$

so

$$\begin{aligned}
&\mathbb{E}_{x_{t-1}\sim q_{t-1}}\left[KL(q_t||p_t(.|x_{t-1}))\right] \\
&= d\left[\log(\frac{\tau_t}{\sigma_t}) + \frac{\tau_t^2}{\sigma_t^2} - \frac{1}{2}\right] + \mathbb{E}_{x_{t-1}\sim q_{t-1}}\left[\frac{||x_{t-1} - \mu_t||^2}{2\tau_t^2}\right] \\
&= d\left[\log(\frac{\tau_t}{\sigma_t}) + \frac{\tau_t^2}{\sigma_t^2} - \frac{1}{2}\right] + \frac{1}{2\tau_t^2}\mathbb{E}_{x_{t-1}\sim q_{t-1}}\left[||x_{t-1} - \mu_{t-1}||^2 + ||\mu_t - \mu_{t-1}||^2\right] \\
&= d\left[\log(\frac{\tau_t}{\sigma_t}) + \frac{\tau_t^2}{\sigma_t^2} - \frac{1}{2}\right] + \frac{1}{2\tau_t^2}\left[||\mu_t - \mu_{t-1}||^2 + \sigma_{t-1}^2\right]
\end{aligned}$$

So finally we get

$$KL(q||p) = \frac{||\mu_1 - \nu_1||^2}{2\tau_1^2} + k\sum_{t=1}^{T}\left[\log(\frac{\tau_t}{\sigma_t}) + \frac{\tau_t^2}{\sigma_t^2} - \frac{1}{2}\right] + \sum_{t=1}^{T}\frac{||\mu_t - \mu_{t-1}||^2 + \sigma_{t-1}^2}{2\tau_t^2}$$

APPENDIX E
NEGATIVE SAMPLING STRATEGY

In this section, we provide a detailed explanation of how to calculate the loss using negative sampling. Each node i has a set of neighbors in the graph, denoted $\mathcal{P}(i)$. Each node i contributes a term to the loss function for both its positive neighbors $j \in \mathcal{P}(i)$ and negative non-neighbors $j \notin \mathcal{P}(i)$. For instance with only 1000 nodes, this leads to around 1 million terms in the loss function, which is not feasible to compute. What can be done for instance is to select r negative neighbor per positive neighbor, leading to a number of terms in the log-likelihood equal to $\sum_{i=1}^{n}(1+r)|\mathcal{P}(i)|$. Due to the power-law degree distribution, this number is much smaller than $\mathcal{U}\times(\mathcal{U}-1)$. In our implementation we use $r = 1$ negative neighbors per positive neighbor.

REFERENCES

[1] M. Arastuie, S. Paul, and K. S. Xu, "CHIP: A Hawkes process model for continuous-time networks with scalable and consistent estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–10.
[2] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: Deep universal probabilistic programming," *J. Mach. Learn. Res.*, vol. 20, pp. 973–978, Jan. 2019.
[3] A. Bojchevski and S. Günnemann, "Deep Gaussian embedding of graphs: Unsupervised inductive learning via ranking," 2018, *arXiv:1707.03815*.
[4] J. Cadena, A. P. Sales, D. Lam, H. A. Enright, E. K. Wheeler, and N. O. Fischer, "Modeling the temporal network dynamics of neuronal cultures," *PLOS Comput. Biol.*, vol. 16, no. 5, May 2020, Art. no. e1007834.
[5] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," 2019, *arXiv:1806.07366*.
[6] N. Eagle and A. S. Pentland, "Reality mining: Sensing complex social systems," *Pers. Ubiquitous Comput.*, vol. 10, no. 4, pp. 255–268, May 2006.
[7] J. Fournet and A. Barrat, "Contact patterns among high school students," *PLoS ONE*, vol. 9, no. 9, Sep. 2014, Art. no. e107878.
[8] R. Goel, S. M. Kazemi, M. Brubaker, and P. Poupart, "Diachronic embedding for temporal knowledge graph completion," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2019, pp. 3988–3995.
[9] M. Génois, C. L. Vestergaard, J. Fournet, A. Panisson, I. Bonmarin, and A. Barrat, "Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers," *Netw. Sci.*, vol. 3, no. 3, pp. 326–347, Sep. 2015.
[10] P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent space approaches to social network analysis," *J. Amer. Stat. Assoc.*, vol. 97, no. 460, pp. 1090–1098, Dec. 2002.
[11] S. Huang, F. Poursafaei, J. Danovitch, M. Fey, W. Hu, E. Rossi, and J. Leskovec, "Temporal graph benchmark for machine learning on temporal graphs," 2023, *arXiv:2307.01026*.
[12] Z. Huang, H. Soliman, S. Paul, and K. S. Xu, "A mutually exciting latent space Hawkes process model for continuous-time networks," in *Proc. 38th Conf. Uncertainty Artif. Intell.*, J. Cussens and K. Zhang, Eds. vol. 180, Aug. 2022, pp. 863–873.
[13] B. Kang, J. Lijffijt, and T. D. Bie, "Conditional network embeddings," in *Proc. 7th Int. Conf. Learn. Represent.*, May 2019, pp. 1–10.
[14] H. Kaur, R. Rastelli, N. Friel, and A. E. Raftery, "Latent position network models," 2023, *arXiv:2304.02979*.
[15] S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, and P. Poupart, "Representation learning for dynamic graphs: A survey," *J. Mach. Learn. Res.*, vol. 21, pp. 1–73, Jan. 2020.
[16] B. Kim, K. H. Lee, L. Xue, and X. Niu, "A review of dynamic network models with latent variables," *Statist. Surveys*, vol. 12, pp. 105–135, Jan. 2018.
[17] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[18] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *2nd Int. Conf. Learn. Represent.*, Y. Bengio and Y. LeCun, Eds. Banff, AB, Canada, Apr. 2014, pp. 1–9.
[19] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, *arXiv:1611.07308*.
[20] P. N. Krivitsky and M. S. Handcock, "Fitting position latent cluster models for social networks with LatentNet," *J. Stat. Softw.*, vol. 24, p. 5, Feb. 2008.
[21] A. Machens, F. Gesualdo, C. Rizzo, A. E. Tozzi, A. Barrat, and C. Cattuto, "An infectious disease model on empirical networks of human contact: Bridging the gap between dynamic network data and contact matrices," *BMC Infectious Diseases*, vol. 13, no. 1, p.185, Dec. 2013.
[22] N. Masuda and R. Lambiotte, *A Guide to Temporal Networks*. EUROPE: World Scientific, 2016.

[23] P. Panzarasa, T. Opsahl, and K. M. Carley, "Patterns and dynamics of users' behavior and interaction: Network analysis of an online community," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 60, no. 5, pp. 911–932, May 2009.
[24] F. S. Passino and N. A. Heard, "Mutually exciting point process graphs for modeling dynamic networks," *J. Comput. Graph. Statist.*, vol. 32, no. 1, pp. 116–130, Jan. 2023.
[25] F. Poursafaei, S. Huang, K. Pelrine, and R. Rabbany, "Towards better evaluation for dynamic link prediction," in *Proc. Neural Inf. Process. Syst. (NeurIPS) Datasets Benchmarks*, 2022, pp. 1–6.
[26] A. E. Raftery, X. Niu, P. D. Hoff, and K. Y. Yeung, "Fast inference for the latent space network model using a case-control approximate likelihood," *J. Comput. Graph. Statist.*, vol. 21, no. 4, pp. 901–919, Oct. 2012.
[27] R. Rastelli and M. Corneli, "Continuous latent position models for instantaneous interactions," *Netw. Sci.*, pp. 1–29, Jul. 2023. [Online]. Available: https://www.cambridge.org/core/journals/network-science/article/continuous-latent-position-models-for-instantaneous-interactions/8F4AA4E78B0593748C076F5ADF8CA280
[28] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," 2020, *arXiv:2006.10637*.
[29] P. Sarkar and A. W. Moore, "Dynamic social network analysis using latent space models," *ACM SIGKDD Explorations Newslett.*, vol. 7, no. 2, pp. 31–40, Dec. 2005.
[30] K. S. Xu, M. Kliger, and A. O. Hero, "A regularized graph layout framework for dynamic network visualization," *Data Mining Knowl. Discovery*, vol. 27, no. 1, pp. 84–116, Jul. 2013.
[31] M. Xu, A. V. Singh, and G. E. Karniadakis, "DynG2G: An efficient stochastic graph embedding method for temporal graphs," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 10, 2022, doi: 10.1109/TNNLS.2022.3178706.
[32] J. Yang, V. A. Rao, and J. Neville, "Decoupling homophily and reciprocity with latent space network models," in *Proc. 33rd Conf. Uncertainty Artif. Intell.*, G. Elidan, K. Kersting, and A. Ihler, Eds. Sydney, NSW, Australia, Aug. 2017, pp. 1–5.

RAPHAËL ROMERO received the M.Eng. degree from Télécom Paris, and the M.Sc. degree in mathematics, computer vision, and machine learning (MVA) from ENS Paris-Saclay. He is currently pursuing the Ph.D. degree in machine learning with Ghent University, under the supervision of Prof. Tijl De Bie. Previously, he was a Data Scientist with Shift Technology, Paris, for one year and interned six month in the Paris-based start-up Jalgos AI. His research interests include graph representation learning, temporal graph embedding, and graph neural networks.

JEFREY LIJFFIJT received the D.Sc. degree (Hons.) in technology from Aalto University, in 2013. He was a Research Fellow with the University of Bristol and a FWO [Pegasus]^2 Marie Skłodowska-Curie Fellow with Ghent University, Belgium, where he is currently an Assistant Professor of data science, knowledge discovery, and visual analytics with IDLab. His research interests include theory and practice of statistical modeling, knowledge discovery, data visualization, and interaction with data. He has a website at http://users.ugent.be/jlijffij/.

RICCARDO RASTELLI received the Ph.D. degree in statistics from University College Dublin, in 2017. He has been a Lecturer of statistics with the School of Mathematics and Statistics, University College Dublin, since 2018. He was a Postdoctoral Researcher with the Vienna University of Economics and Business, from 2017 to 2018. His research interests include statistical analysis of networks, with a focus on computational problems, Bayesian inference and clustering. In his research, he investigates problems and applications in various applied areas, including financial systemic risk, social and political sciences, and the analysis of biological networks, such as microbiome analysis.

MARCO CORNELI received the M.S. degree in statistics and probability from University Paris Diderot, in 2014, and the Ph.D. degree from the SAMM Laboratory, University Paris 1 Panthéon-Sorbonne, in November 2017. During his Ph.D., he dealt with machine and statistical learning approaches for clustering and change points detection in dynamic graphs. Then, he moved to Université Côte d'Azur, Nice, France, for a two years postdoctoral position (LJAD Laboratory). During this period, he involved on co-clustering techniques for homogenous data (ordinal, counting, and textual). Since 2019, he has been a part of the INRIA Research Team, MAASAI, Sophia-Antipolis, and he was a Researcher with the Center of Modeling, Simulation and Interactions, Université Côte d'Azur, where he has been a Junior Professor of AI for archaeology and history, Since September 2022. In collaboration with the researchers of the CEPAM Laboratory, his job consists of developing new AI models and algorithms to answer research questions related with the historical/archaeological data. He supervised three Ph.D. theses and published in international journals and conferences in the domains of computational statistics, machine learning, and artificial intelligence.

TIJL DE BIE received the Ph.D. degree in machine learning from KU Leuven, in 2005. He is currently a Senior Full Professor with Ghent University, Belgium. Before moving to Ghent University, he was a Reader with the University of Bristol, a Postdoctoral Researcher with KU Leuven and the University of Southampton, and a Visiting Research Scholar with U.C. Berkeley and U.C. Davis. As a former ERC Consolidator Grantee, he has extensively worked on the formalization of subjective interestingness in exploratory data mining. His current research interests include graph-based machine learning, recommender systems, data visualization, ethics and regulation of AI, and applications of AI, particularly in human resources and job market use cases.