



HAL
open science

Continuous Latent Position Models for Instantaneous Interactions

Riccardo Rastelli, Marco Corneli

► **To cite this version:**

Riccardo Rastelli, Marco Corneli. Continuous Latent Position Models for Instantaneous Interactions. Network Science, 2023, 11 (4), pp.560-588. 10.1017/nws.2023.14 . hal-03517392

HAL Id: hal-03517392

<https://hal.science/hal-03517392>

Submitted on 7 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Continuous Latent Position Models for Instantaneous Interactions

* Riccardo Rastelli¹ and Marco Corneli²

¹School of Mathematics and Statistics, University College Dublin, Dublin, Ireland,
riccardo.rastelli@ucd.ie

²Université Côte d’Azur, INRIA, Center of Modeling, Simulation and Interactions, MAASAI Team,
Nice, France, marco.corneli@univ-cotedazur.fr

Abstract

We create a framework to analyze the timing and frequency of instantaneous interactions between pairs of entities. This type of interaction data is especially common nowadays, and easily available. Examples of instantaneous interactions include email networks, phone call networks and some common types of technological and transportation networks. Our framework relies on a novel extension of the latent position network model: we assume that the entities are embedded in a latent Euclidean space, and that they move along individual trajectories which are continuous over time. These trajectories are used to characterize the timing and frequency of the pairwise interactions. We discuss an inferential framework where we estimate the individual trajectories from the observed interaction data, and propose applications on artificial and real data.

Keywords: Latent Position Models; Dynamic Networks; Non-Homogeneous Poisson Process; Spatial Embeddings; Statistical Network Analysis.

1 Introduction

The Latent Position Model (LPM, Hoff et al. 2002) is a widely used statistical model that can be used to characterize a network through a latent space representation. The model embeds the nodes of the network as points in the real plane, and then uses these

*The authors contributed equally to this work.

latent features to explain the observed interactions between the entities. This provides a neat and easy-to-interpret graphical representation of the observed interaction data, which is able to capture some extremely common empirical features such as transitivity and homophily.

In this paper, we propose a new LPM that can be used to model repeated instantaneous interactions between entities, over an arbitrary time interval. The time dimension is continuous, and an interaction between any two nodes may happen at any point in time. We propose a data generative mechanism which is inspired by the extensive literature on LPMs, and we define an efficient estimation framework to fit our model.

Since the foundational work of Hoff et al. (2002), the literature on LPMs has been developed in many directions, both from the methodological and from the applied point of views. Recent review papers on the topic include Salter-Townshend et al. (2012), Rastelli et al. (2016), Raftery (2017), and Sosa and Buitrago (2020).

As regards statistical methodology, the original paper of Hoff et al. (2002) defined a framework to infer and interpret a LPM for binary interactions. The authors introduced two types of LPMs: the **projection model** and the **distance model**.

The **projection model** postulates that the probability that an edge appears between any two nodes is determined by the dot product of the latent coordinates of the two respective nodes. As a consequence, a crucial contribution for the edge probability is given by the direction the nodes point towards. By contrast, the distance model defines the connection probability as a function of the Euclidean distance between the two nodes. Nodes that are located close to each other are more likely to connect than nodes that are located far apart. Both models provide a clear representation of the interaction data which can be used to study the network’s topology, or to construct model-based summaries and visualizations, or predictions.

More recently, the projection model and its variations have been extensively studied and used in a variety of applications (see Hoff 2005; Hoff 2018 and references therein). This model has also clear connections to a rich machine learning literature on spatial embeddings, which include Rahimi and Recht (2007), Lee and Seung (1999), and Halko et al. (2011). Variations of the projection model have been extended to dynamic settings (Durante and Dunson 2014; Durante and Dunson 2016), and other types of networks frameworks (Durante et al. 2017).

As regards the **distance model**, this has been extended by Handcock et al. (2007)

and Krivitsky et al. (2009) to represent clustering of the nodes and more flexible degree distributions. In the context of networks evolving over time, dynamic extensions of the model have been considered in Sarkar and Moore (2006), and more recently in several works including Sewell and Chen (2015b) and Friel et al. (2016) for binary interactions. The recent review paper of Kim et al. (2018) provides additional references on dynamic network modeling. Other relevant and interesting works that revolve around the distance models in either static or dynamic settings include Gollini and Murphy (2014) and Salter-Townshend and McCormick (2017) for multi-view networks, Sewell and Chen (2016) for dynamic weighted networks, and Gormley and Murphy (2007) and Sewell and Chen (2015a) for networks of rankings. We also mention Raftery et al. (2012), Fosdick et al. (2019), Rastelli et al. (2018), and Tafakori et al. (2019) which introduce original and closely related modeling or computational ideas.

Crucially, we note that all existing dynamic LPMs consider a discrete time dimension, whereby the interactions are observed at a number of different points in time¹. By contrast, a fundamental original aspect of this paper is that it considers a fully continuous LPM whereby the interactions are instantaneous, and they can happen at any point in time. Continuous networks of this type are especially common and widely available, as they include email networks (Klimt and Yang 2004), functional brain networks (Park and Friston 2013), and other networks of human interactions (see Cattuto et al. 2010; Barrat and Cattuto 2013 and references therein).

Some of the approaches that have been proposed in the statistics literature to model instantaneous interactions include Corneli et al. (2017) and Matias et al. (2018), however we note that these approaches rely on extensions of the stochastic blockmodel (Nowicki and Snijders 2001), and not on the LPM. Another relevant strand of literature focuses instead on modeling this type of data using Hawkes processes (see Junuthula et al. 2019 and references therein).

We propose our new Continuous Latent Position Model (CLPM) both for the projection model framework and for the distance model framework. In our approach, each of the nodes is characterized by a latent trajectory on the real plane, which is assumed to be a piece-wise linear curve. The interactions between any two nodes are modeled as events of a inhomogeneous Poisson point process, whose rate is determined by the

¹One exception is Durante and Dunson (2016) but we find their approach and goals fundamentally different from ours.

instantaneous positions of the nodes, at each point in time. The piece-wise linear curve assumption gives sufficient flexibility regarding the possible trajectories, while not affecting the purely continuous nature of the framework, in that the rate of the Poisson process is not piece-wise constant. This is a major difference with respect to other approaches that have been considered (Corneli et al. 2017 and one of the approaches of Matias et al. 2018).

We propose a penalized likelihood approach to perform inference, and we use optimization via gradient descent to obtain optimal estimates of the model parameters. We have created a software that implements our estimation method, which is publicly available from *CLPM GitHub repository* (2021).

The paper is structured as follows: in Section 2 we introduce our new model and its two variants (i.e. the projection and distance model), and we derive the main equations that are used in the paper; in Section 3 we describe our approach to estimate the model parameters; in Section 5 we illustrate our procedure on three synthetic datasets, whereas in Section 6 we propose real data applications. We give final comments and conclusions in Section 7.

2 The model

2.1 Modeling the interaction times

The data that we observe is stored as a list of interactions (or *edge list*) in the format $\mathcal{E} := \{(\tau_e, i_e, j_e)\}_{e \in \mathbb{N}}$, where $\tau_e \in [0, T]$ for all e is the interaction time between the nodes i_e and j_e , with $i_e, j_e \in \{1, \dots, N\}$. We consider undirected interactions without self loops, although extensions to the directed case are straightforward. We emphasize that all interactions are instantaneous, i.e. their length is not relevant or not recorded. An interaction between two nodes may occur at any point in time $\tau_e \in [0, T]$. Let us now formally introduce the list of the interaction times between two arbitrary nodes i and j :

$$\mathcal{E}_{ij} = \{\tau_1^{(i,j)}, \dots, \tau_{E_{ij}}^{(i,j)}\}, \quad (1)$$

where E_{ij} is the total number of times i interacts with j before T . We assume that the interaction times in the above equation are the realization of a inhomogeneous Poisson point process with instantaneous rate function denoted with $\lambda_{ij}(t) \geq 0, \forall t \in [0, T]$ and nodes i and j . Using a more convenient (but equivalent) characterization, we state that the waiting time for a new interaction event between i and j is exponential with a variable

rate that changes over time. Then, if we assume that the inhomogeneous point processes are independent for all pairs i and j , the likelihood function for the rates can be written as:

$$\mathcal{L}(\boldsymbol{\lambda}) = \prod_{i,j: i < j} \left[\left(\prod_{e \in \mathcal{E}_{ij}} \lambda_{ij}(\tau_e) \right) \exp \left\{ - \int_0^T \lambda_{ij}(t) dt \right\} \right], \quad (2)$$

where, for simplicity, we have removed the superscript (i, j) from $\tau_e^{(i,j)}$. In the sections below we will specify the conditions that make the processes independent.

2.2 Latent positions

Our goal is to embed the nodes of the network into a latent space, such that the latent positions are the primary driving factor behind the frequency and timing of the interactions between the nodes. Crucially, since the time dimension is continuous and interactions can happen at any point in time, we aim at creating a modeling framework which also evolves continuously over time. Thus, the fundamental assumption of our model is that, at any point in time, the Poisson rate function $\lambda_{ij}(t)$ is determined by the latent positions of the corresponding nodes, which we denote $\mathbf{z}_i(t) \in \mathbb{R}^2$ and $\mathbf{z}_j(t) \in \mathbb{R}^2$.

Remark. We assume that the number of dimensions of the latent space is equal to 2, because the main interest of the proposed approach is in latent space visualization of the network. However, we note that the generative model presented in this section can be easily extended to the case $\mathbf{z}_i(t) \in \mathbb{R}^d$, with $d > 2$.

To facilitate the inference task, the trajectories are assumed to be piece-wise linear curves, characterized by a number of user-defined change points in the time dimension. These change points are in common across the trajectories of all nodes and they determine the points in time when the linear motions of the nodes change direction and speed. This means that we must define a grid of the time dimension through K change points

$$0 = \eta_1 < \eta_2 < \dots < \eta_{K-1} < \eta_K = T$$

that are common across all trajectories. We stress that this modeling choice is only meant to restrict the variety of continuous trajectories that we may consider, as it allows us to use a tractable parametric structure while keeping a high flexibility regarding the trajectories that can be obtained. Also, we make the assumption that, within any two consecutive critical points, the speed at which any given node moves remains constant.

As a consequence, we only need to store the coordinates of the nodes at the change points, since all the intermediate positions can then be obtained with:

$$\mathbf{z}_i((1-t)\eta_k + t\eta_{k+1}) = (1-t)\mathbf{z}_i(\eta_k) + t\mathbf{z}_i(\eta_{k+1}) \quad \forall t \in [0, 1] \quad (3)$$

for any change points η_k and η_{k+1} and node i .

2.2.1 Projection Model

Similarly to the foundational paper of Hoff et al. (2002), we introduce two possible characterizations of the rates through the latent positions: one is inspired by the projection model, the other is inspired by the distance model. In our projection model, we assume that:

- i) the latent positions are constrained within the first quadrant of \mathbb{R}^2 , i.e. $\mathbf{z}_i(t) \in \mathbb{R}^+ \times \mathbb{R}^+$, for all $t \in [0, T]$ and node i ;
- ii) the rate is exactly equal to the dot product between the positions, i.e.

$$\lambda_{ij}(t) = \langle \mathbf{z}_i(t), \mathbf{z}_j(t) \rangle, \text{ for all } t \in [0, T] \text{ and nodes } i \text{ and } j.$$

As a consequence, the further the nodes are positioned from the origin, the more frequent their interactions will be, especially towards other nodes that are aligned in the same direction². Viceversa, we are not expecting frequent interactions for nodes that are located too close to the origin, or between pairs of nodes forming an angle which is close to 90 degrees. The restriction of the latent space to the first quadrant guarantees that the rate remains always non-negative, and we argue that this assumption does not diminish the flexibility nor interpretability of the model.

By taking the logarithm of Eq. (2) and replacing $\lambda_{ij}(t)$, the log-likelihood for the projection model is

$$\log \mathcal{L}(\mathbf{Z}) = \sum_{i,j:i < j} \left\{ \left(\sum_{e \in \mathcal{E}_{ij}} \log (\langle \mathbf{z}_i(\tau_e), \mathbf{z}_j(\tau_e) \rangle) \right) - \int_0^T \langle \mathbf{z}_i(t), \mathbf{z}_j(t) \rangle dt \right\} \quad (4)$$

Remark. When expressing the Poisson rate $\lambda_{ij}(\cdot)$ as a function of the latent trajectories, we move from the unconditional independence assumption leading to Eq. (2) to a conditional one. The timelines of events for all pairs of nodes are independent *given* the latent trajectories.

²Indeed, we could say equivalently

$$\lambda_{ij}(t) = \cos(\alpha_{ij}) \|\mathbf{z}_i(t)\| \|\mathbf{z}_j(t)\|$$

where $\alpha_{ij} \in [0, \frac{\pi}{2}]$ is the angle between $\mathbf{z}_i(t)$ and $\mathbf{z}_j(t)$ and $\|\cdot\|$ is the Euclidean norm.

As proven in Appendix A, the integral term appearing in Eq. (4) can be calculated analytically, thus leading to a closed form expression for the log-likelihood of the projection model.

2.2.2 Distance Model

Here, we introduce a version of the latent position model that uses the latent Euclidean distances between the nodes, rather than the dot products. We argue that, in this context, the distance model provides both more flexibility and easier interpretability. The simulation studies that we perform in Section 5 will highlight these advantages.

In the distance model, we assume that:

$$\log \lambda_{ij}(t) = \beta - \|\mathbf{z}_i(t) - \mathbf{z}_j(t)\|^2 \quad (5)$$

where the last term corresponds to the squared Euclidean distance between nodes i and j at time t . We also introduced an intercept term β as per the original LPM by Hoff et al. (2002). The intercept term affects all nodes but extensions of the model where it become specific to each node can also be considered for both the projection and the distance models. By taking the logarithm of Eq. (2) and using Eq. (5) the log-likelihood of the distance model becomes:

$$\log \mathcal{L}(\beta, \mathbf{Z}) = \sum_{i,j} \left\{ \left(\sum_{e \in \mathcal{E}_{ij}} (\beta - \|\mathbf{z}_i(\tau_e) - \mathbf{z}_j(\tau_e)\|^2) \right) - \int_0^T e^{\beta - \|\mathbf{z}_i(s) - \mathbf{z}_j(s)\|^2} ds \right\} \quad (6)$$

Similarly to the projection model, also the above log-likelihood has a closed form, since the integral inside the brackets can be calculated analytically (proof in Appendix B).

2.3 Penalized likelihood

Due to the piece-wise linearity assumption in Eq. (5), for each node we only need to estimate its positions at times $\{\eta_k\}_{k \in [K]}$. In order to avoid over fitting and obtain more interpretable and meaningful results, we use prior distributions over the latent positions at $\{\eta_k\}_{k \in [K]}$, as a means to penalize large velocities of the nodes in the latent space.

Projection Model: as a penalization, in this case we require that:

$$\left\langle \frac{\mathbf{z}_i(\eta_{k+1})}{\|\mathbf{z}_i(\eta_{k+1})\|}, \frac{\mathbf{z}_i(\eta_k)}{\|\mathbf{z}_i(\eta_k)\|} \right\rangle \stackrel{\perp}{\sim} \mathcal{TN}(\mu, (\eta_{k+1} - \eta_k)\sigma^2, 0, 1) \quad \forall k = 1, \dots, K - 1 \quad (7)$$

for all nodes, where μ and σ^2 are hyper-parameters to be fixed by the user. The above assumption states that the cosine of the angle between the position of a node at a change point, and the position of the same node at the following change point, follows a truncated Gaussian distribution (in $[0, 1]$). In the applications we choose $\mu \approx 1$ and a small value of σ^2 so that we require that any node rotates around the origin as little as necessary.

Distance Model. We define Gaussian random walk priors on the critical points of the latent trajectories:

$$\begin{aligned} \mathbf{z}_i(0) &\stackrel{iid}{\sim} \mathcal{N}(0, \sigma_0^2 I_2) \\ \mathbf{z}_i(\eta_{k+1}) \mid \mathbf{z}_i(\eta_k) &\stackrel{\perp}{\sim} \mathcal{N}(\mathbf{z}_i(\eta_k), (\eta_{k+1} - \eta_k)\sigma^2 I_2) \quad \forall k = 1, \dots, K - 1 \end{aligned} \tag{8}$$

for every node i where I_2 is the identity matrix of order two. The equation above (with $\sigma^2 = 1$) would correspond to a Brownian motion process for the i -th latent trajectory, except that we would only observe it at the change points, where the latent positions are estimated. However, as the number of change points increases, the prior that we specify tends to a scaled Brownian motion on the plane. The parameters σ_0^2 and σ^2 are user-defined, hence, similarly to the projection model, the Gaussian priors can be used as penalizations. In order to obtain sensible penalizations, we choose small values of the variance parameters, as to ensure that the nodes are scattered around the origin of the space, and that the speed of the nodes along the trajectories is not too large. In this way, the nodes are forced to move as little as necessary, making the latent visualization of the network easier to read and interpret.

Remark. The likelihood function of the original latent distance model of Hoff et al. (2002) is not identifiable with respect to translations, rotations, and reflections of the latent positions. This is a challenging issue in a Bayesian setting that relies on sampling from the posterior distribution. In fact, the posterior samples become non-interpretable, since the affine transformations may have occurred during the collection of the sample (Shortreed et al. 2006). These non-identifiabilities are not especially relevant in an optimization setting, since usually the equivalent configurations of model parameters lead to the same qualitative results and interpretations. However, a case for non-identifiability can be made for dynamic networks, since translations, rotations, and reflections can occur across time, thus affecting results and interpretation. The penalizations that we introduce in this paper ensure that the nodes move as little as necessary, thus disfavouring any rotations, translations and reflections of the space. As a consequence, the penalizations directly

address the identifiability issues and the latent point process remains comparable across time.

3 Inference

In this section, we discuss the inference for the distance model described in Section 2.2.2, but an analogous procedure is considered for the projection model.

Recalling that we work with undirected graphs and in force of Eq. (8), the log-likelihood is

$$\log \mathcal{L}(\beta, \mathbf{Z}) = \sum_{i=1}^N \left\{ \frac{1}{2} \sum_{\substack{j=1 \\ j \neq i}}^N \left[\left(\sum_{\substack{e \in \mathcal{E}_{ij} \\ j \neq i}} (\beta - \|\mathbf{z}_i(\tau_e) - \mathbf{z}_j(\tau_e)\|^2) \right) - \int_0^T e^{\beta - \|\mathbf{z}_i(s) - \mathbf{z}_j(s)\|^2} ds \right] - \frac{1}{2\sigma^2} \sum_{k=1}^K \|\mathbf{z}_i(\eta_k) - \mathbf{z}_i(\eta_{k-1})\|^2 - \frac{1}{2\sigma_0^2} \|\mathbf{z}_i(\eta_0)\|^2 \right\} + C, \quad (9)$$

where C is a constant term that does not depend on (\mathbf{Z}, β) and the integral can be explicitly computed as shown in Appendix B. Since this log-likelihood has a closed form, we implement it and rely on automatic differentiation (Griewank 1989; Baydin et al. 2018) to maximize it numerically, with respect to (β, \mathbf{Z}) , via gradient descent (GD)³. Moreover, as it can be seen in Eq. (9), the log-likelihood is additive in the number of nodes. Potentially, this remark allows one to speed up the inference of the model parameters by means of stochastic gradient descent (Bottou 2010). Indeed, let us introduce ψ_1, \dots, ψ_n such that

$$\psi_i(\beta, \mathbf{Z}) := \frac{1}{2} \sum_{\substack{j=1 \\ j \neq i}}^N \left[\left(\sum_{\substack{e \in \mathcal{E}_{ij} \\ j \neq i}} (\beta - \|\mathbf{z}_i(\tau_e) - \mathbf{z}_j(\tau_e)\|^2) \right) - \int_0^T e^{\beta - \|\mathbf{z}_i(s) - \mathbf{z}_j(s)\|^2} ds \right] - \frac{1}{2\sigma^2} \sum_{k=1}^K \|\mathbf{z}_i(\eta_k) - \mathbf{z}_i(\eta_{k-1})\|^2 - \frac{1}{2\sigma_0^2} \|\mathbf{z}_i(\eta_0)\|^2 \quad (10)$$

and a discrete random variable $\Psi(\beta, \mathbf{Z})$ such that

$$\pi := \pi_i := \mathbb{P}\{\Psi(\beta, \mathbf{Z}) = \psi_i(\beta, \mathbf{Z}) | \mathbf{Z}\} = \frac{1}{N}, \quad \forall i \in \{1, \dots, N\}$$

³Since we maximize a log-likelihood, gradient *ascent* is performed instead of gradient *descent*. However, since maximizing the log-likelihood is equivalent to minimize its opposite and in order to adopt the same convention as in machine/statistical learning literature we keep using the expression “gradient descent”

where we stress that the above probability is conditional to \mathbf{Z} and given the model parameter β . Then, let us denote ∇ the gradient operator with respect to (β, \mathbf{Z}) and \mathbb{E}_π the expectation taken with respect to the probability measure π introduced above (and hence with \mathbf{Z} given). Then, we have the following

Proposition 1. $N\nabla\Psi(\beta, \mathbf{Z})$ is an unbiased estimator of $\nabla \log \mathcal{L}(\beta, \mathbf{Z})$.

Proof.

$$\mathbb{E}_\pi [N\nabla\Psi(\beta, \mathbf{Z})] = \sum_{i=1}^N \nabla\psi_i(\beta, \mathbf{Z}) = \nabla \left(\sum_{i=1}^N \psi_i(\beta, \mathbf{Z}) \right) = \nabla \log \mathcal{L}(\beta, \mathbf{Z}),$$

where the last equality follows from the additivity of the gradient operator. \square

The above proposition allows us to sample (subsets of) nodes uniformly at random, with re-injection, and use each sample (a.k.a. mini-batch) to update the model parameters via stochastic gradient descent (SGD), as shown in Bottou (2010). In more details, if $\theta := \{\beta, \mathbf{Z}\}$ denotes the set of the model parameters, at the k -th iteration of the SGD algorithm, θ is updated as follows

$$\theta := \theta + \rho \frac{N}{|B_k|} \nabla \psi_{B_k}(\theta),$$

where the hyper-parameter ρ is the learning rate, B_k is a set of $|B_k|$ nodes extracted uniformly at random and $\psi_{B_k}(\theta)$ refers to the estimator of the full-batch log-likelihood, based on the data batch B_k , namely:

$$\begin{aligned} \psi_{B_k}(\beta, \mathbf{Z}) := & \frac{1}{2} \sum_{i \in B_k} \left[\sum_{\substack{j=1 \\ j \neq i}}^N \left(\sum_{e \in \mathcal{E}_{ij}} (\beta - \|\mathbf{z}_i(\tau_e) - \mathbf{z}_j(\tau_e)\|^2) \right) - \int_0^T e^{\beta - \|\mathbf{z}_i(s) - \mathbf{z}_j(s)\|^2} ds \right] \\ & - \frac{1}{2\sigma^2} \sum_{k=1}^K \|\mathbf{z}_i(\eta_k) - \mathbf{z}_i(\eta_{k-1})\|^2 - \frac{1}{2\sigma_0^2} \|\mathbf{z}_i(\eta_0)\|^2. \end{aligned}$$

We stress that, if $|B_k| = 1$, the above equation reduces to Eq. (10); conversely if $|B_k| = N$ SGD reduces to full-batch gradient descent. We finally note that the above two equations state that the model parameters can be updated, at each iteration, based on a sub-graph with N nodes whose links are uniquely those connecting the nodes in B_k with their neighbours of order one (a.k.a. *friends*).

The estimation algorithm and visualization tools are implemented in a software repository, called CLPM, which is publicly available (*CLPM GitHub repository* 2021).

4 Interpretation and output

Both the distance and the projection model provide a visual representation of the latent space as output. Due to the continuous time dimension, the results are most easily shown as a video. For this paper, the code and results are publicly available from *CLPM GitHub repository* (2021).

The evolution of the latent space provides a visualization which can be used to qualitatively assess the connectivity, both at the global level (e.g. contractions and expansions of the latent space) as well as at a local level (e.g. which nodes have more connections and when).

For the projection model, we expect highly connected nodes to be located far from the origin. They would have a higher chance to interact with any other node. On the other hand, nodes that are close to the origin will have lower connectivity, overall. For both types of nodes, the angle in-between them will also play a role, favouring interactions between nodes that point in the same direction.

For the distance model, we are expecting nodes to take more central positions as they become more active, and, clearly, communities arise when clusters of points are observed. In order to capture this particular behaviour, we introduce a quantitative measure of clustering, or "clusteredness" of the latent space. To construct this measure, we choose an arbitrary threshold value τ , and consider circles of radius τ around each of the nodes, in the latent space. If we consider an arbitrary node, we want to count how many other nodes fall within its circle, at each point in time. By averaging this measure across all nodes, we obtain our clusteredness index, defined as the average number of nodes that fall within a random node's circle.

This measure evolves continuously over time, and we can easily calculate it from the algorithm's output. The relative increases and decreases of the measure over time can permit an appreciation of how the latent space can locally contract, to create communities within the network.

5 Experiments: synthetic data

In this section we illustrate applications of our methodology on artificial data. We propose two types of frameworks: in the first one, we consider dynamic block structures (which involve the presence of communities, hubs, and isolated points). In this case, our aim is

to inspect how the network dynamics are captured by CLPM. In the second framework, we generate data using the distance CLPM and we aim at recovering the simulated trajectories for each node.

5.1 Dynamic block structures

Simulation study 1. In this first experiment, we use a data generative mechanism that relies on a dynamic blockmodel structure for instantaneous interactions (Corneli et al. 2017). We specifically focus on a special case of a dynamic stochastic blockmodel where we can have community structure, but we cannot have disassortative mixing, i.e. the rate of interactions within a community cannot be smaller than the rate of interactions between communities. In this framework, the dynamic stochastic blockmodel approximately corresponds to a special case of our distance CLPM, whereby the nodes clustered together essentially are located nearby.

In the generative framework that we consider the only node-specific information is the cluster label, hence, this structure is not as flexible as the CLPM as regards modeling node’s individual behaviours. So, our goal here is to obtain a latent space visualization for these data, and to ensure that CLPM can accurately capture and highlight the presence of communities. An aspect of particular importance is how CLPM reacts to the creation and dissolution of communities over time: for this purpose, our generated data includes changes in the community structure over time.

For this setup, we consider the time interval $[0, 40]$ (for simplicity we use seconds as a unit measure of time), and divide this into 4 consecutive time segments of 10 seconds each. In each of the 4 time segments, 60 nodes are arranged into different community structures. Thus, any changes in community structure are synchronous for all nodes and they happen at the endpoints of a time segment. The rate of interactions between any two nodes is determined by their group allocations in that specific time segment. The rate remains constant in each time segment, so that we effectively have a piece-wise homogeneous Poisson process over time, for each dyad.

We denote with $X^{(s)} \in \mathbb{N}^{N \times N}$ a simulated weighted interaction matrix which counts how many interactions occur in the s -th time segment for each dyad:

$$X_{ij}^{(s)} | \mathbf{C} \sim \mathcal{P} \left(\theta_{\mathbf{c}_i \mathbf{c}_j}^{(s)} \right),$$

where $\mathcal{P}(\cdot)$ indicates the Poisson probability mass function, and \mathbf{C} is a latent vector of length N indicating the cluster labels of each of the nodes. Once we know the number

of interactions for each dyad and each segment, the timing of these interactions can be sampled from a uniform distribution in the respective time segment. More in detail, the rate parameters are characterized as follows:

- i) in the time segment $[0, 10[$, the expected number of interactions is the same for every pair of nodes: $\theta_{c_i c_j}^{(1)} = 1$, for all i and j ;
- ii) in the time segment $[10, 20[$, three communities emerge, in particular $\theta_{11}^{(2)} = 10$, $\theta_{22}^{(2)} = 5$ and $\theta_{33}^{(2)} = 1$, whereas the rate for any two nodes in different communities is 1;
- iii) in the time segment $[20, 30[$, the first community splits and each half joins a different existing community. The two remaining communities are characterized by $\theta_{11}^{(3)} = \theta_{22}^{(3)} = 5$. Again, any two nodes in different communities interact with rate 1;
- iv) in the time segment $[30, 40]$, we are back to the same structure as in i).

Throughout the simulation, node 1 always behaves as a hub, and node 60 is always isolated. This means that node 1 interacts with rate 10 at all times with any other node, whereas node 60 interacts with rate 0.01 at all times with any other node, regardless of any cluster label.

In Figure 1 we show a collection of snapshots at some critical time points, for the projection model. The full videos of the results are provided in the code repository. The main observation is that the communities are clearly captured at all times, and they are clearly visually separated. While the two clusters with a stronger community structure are almost aligned to the axes, the non-community in the second time segment, which has low interaction rate, is instead positioned more centrally and it is more dispersed, but still separated from the others. The hub is always located very far from the origin and from other points, since this guarantees a large dot product value with respect to all other nodes. By contrast, the isolated node is always located at the origin of the space.

Figure 2 shows instead the snapshots for the distance model. In this case, the clusters are clearly separated at all times. The cluster with a strong community structure is less dispersed than the clusters with a weaker community structure. The hub is constantly positioned in the centre of the space, as to minimize the distance from all of the nodes at the same time. The isolated node is instead wandering in the outskirts of the latent social space. The creation and dissolution of communities only happens right at the proximity of start/end of each time segment.

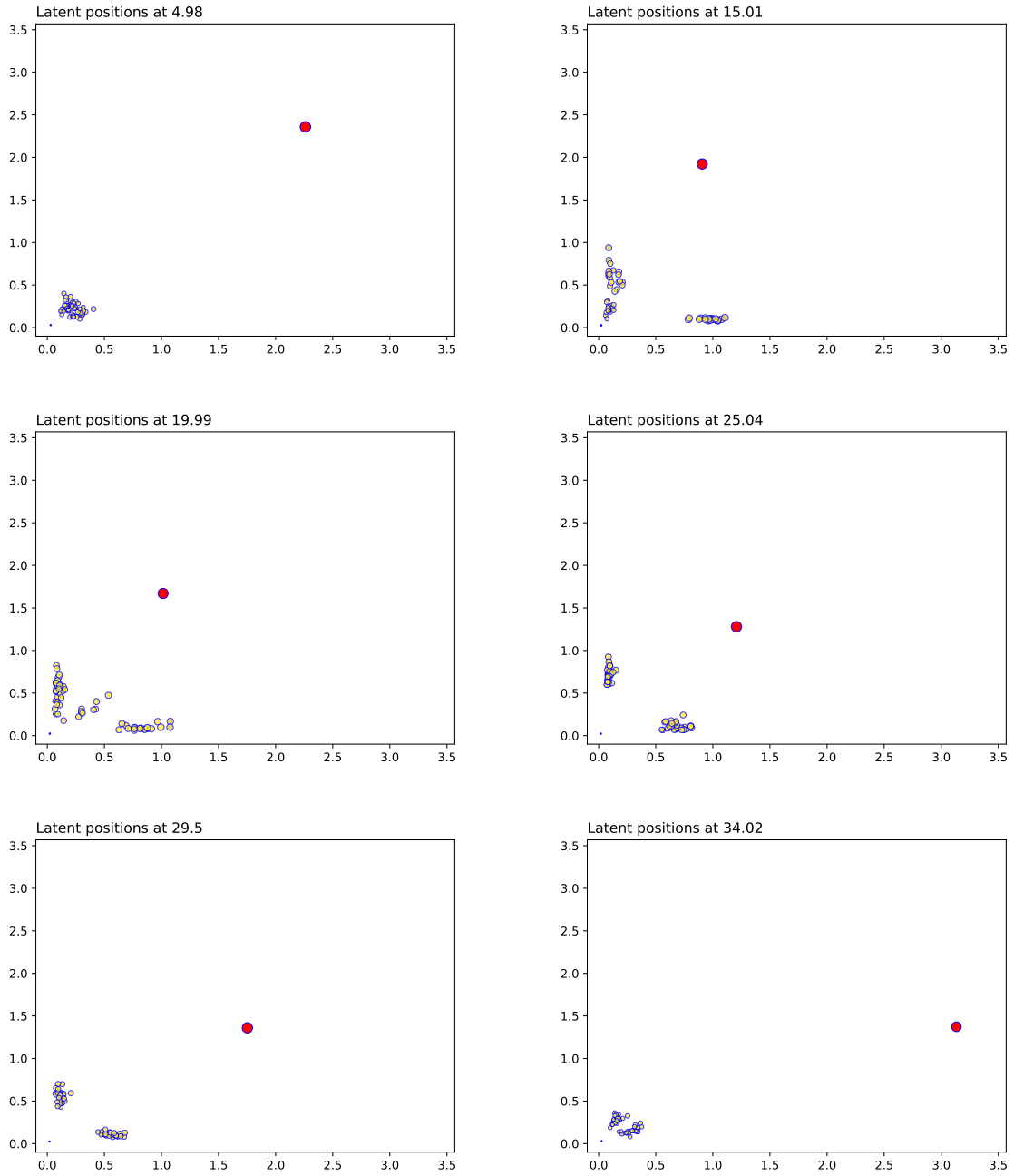


Figure 1: *Simulation study 1: snapshots for the projection model.*

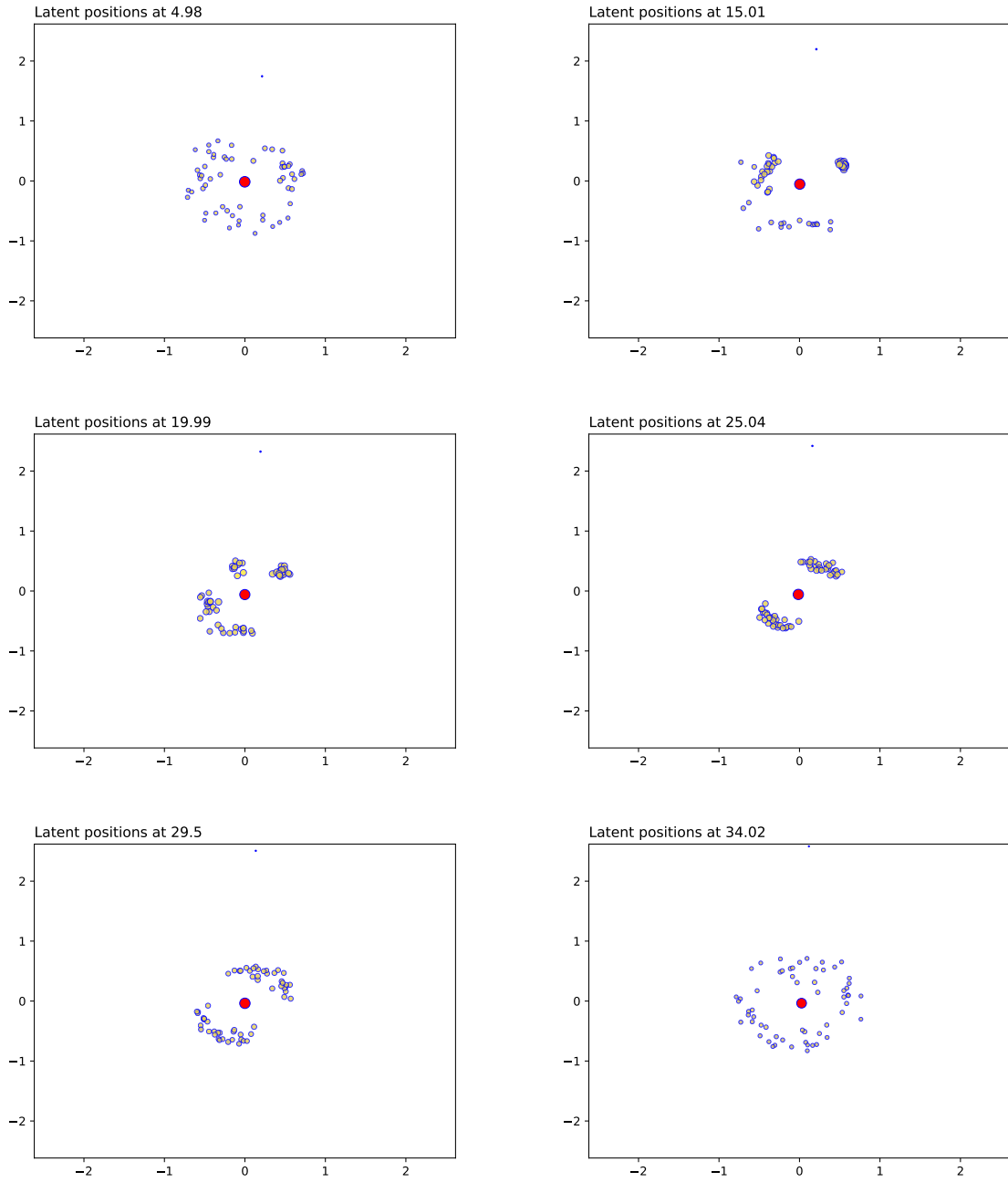


Figure 2: *Simulation study 1: snapshots for the distance model.*

Remark. One main difference between the projection model and the distance model is that, in two dimensions, the distance model seems more suited for representing a diverse community structure. The main reason is that the projection model requires nodes of different communities (which have few interactions with each other) to be distributed along the axes, respectively. Indeed, this guarantees that they are close to perpendicular, hence facilitating a strong separation between communities. Now, in dimension two, this clearly can only happen with no more than two communities for the projection model. By contrast, the distance model does not have this limitation, and it can more easily accommodate a large number of completely separated communities. We use this argument to favour the use of the distance CLPM in two dimensions, in our applications. However, we note that the projection model may be of interest in higher dimension ($d \gg 2$) for purposes other than visualization (e.g. clustering or sub-space tracking).

Technical details regarding the simulation’s parameters, including penalization terms and number of change points, can be consulted on the CLPM code repository.

Simulation study 2. In the second simulation study, we use again a blockmodel structure, however, in this case we approximate a continuous time framework by defining very short time segments, and letting the communities change from one time segment to the next. Since creations and dissolutions of communities would be unlikely in such a short period of time, we keep the community memberships unchanged, and we progressively increase the cohesiveness of the communities. This means that we progressively increase the rates of interactions between any pairs of nodes that belong to the same community, while keeping any other rate constant. The rate of interactions within each community starts at value 1 and increases in a step-wise fashion over 40 segments, up to the value 5. The time interval is $[0, 40]$, and we consider two communities. Half way through the simulation, a special node moves from one community to the other.

For the projection CLPM, we show the results in Figure 3, whereas Figure 4 shows the results for the distance model. Both approaches clearly capture the reinforcement of the communities over time by aggregating the nodes of each group. We observe this behaviour both for the projection model and for the distance model. The projection model also exhibits nodes getting farther from the centre of the space, since this would give them higher interaction rates, overall. As concerns the special node moving from one community to the other, this is well captured in that the node transitions smoothly after approximately 20 seconds, in both models.

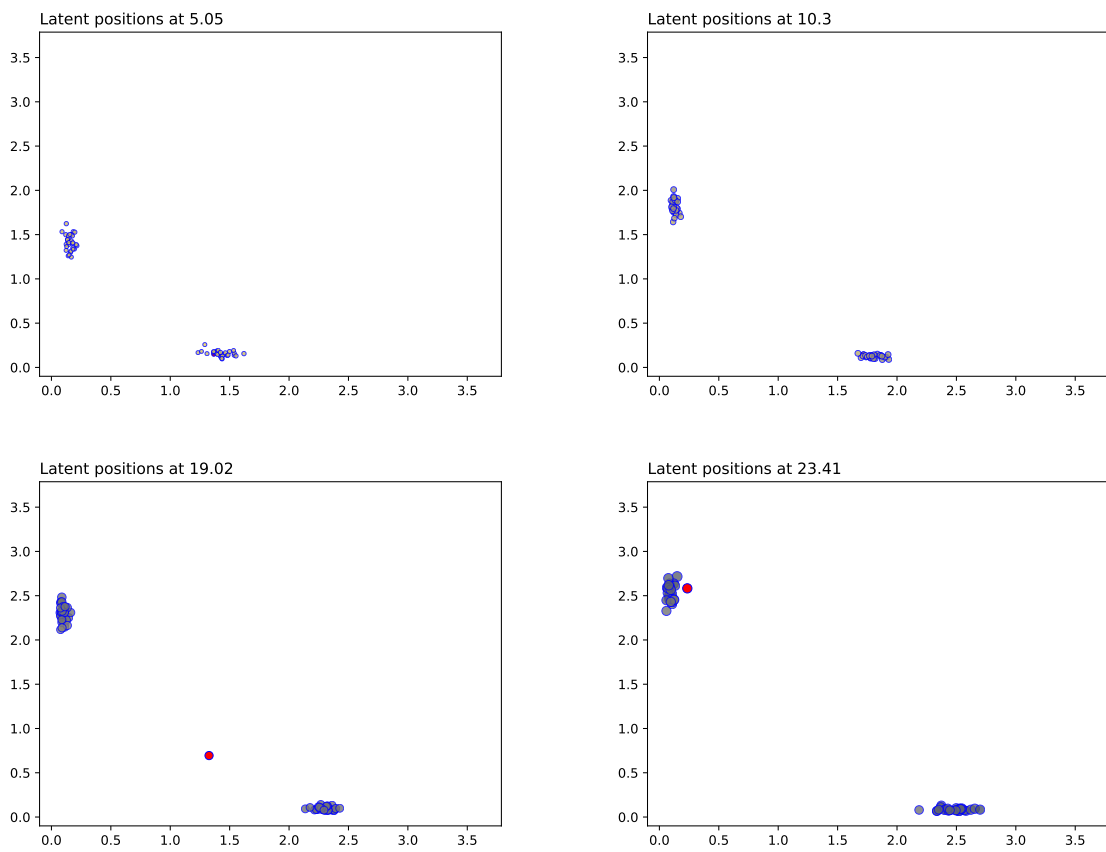


Figure 3: *Simulation study 2: snapshots for the projection model.*

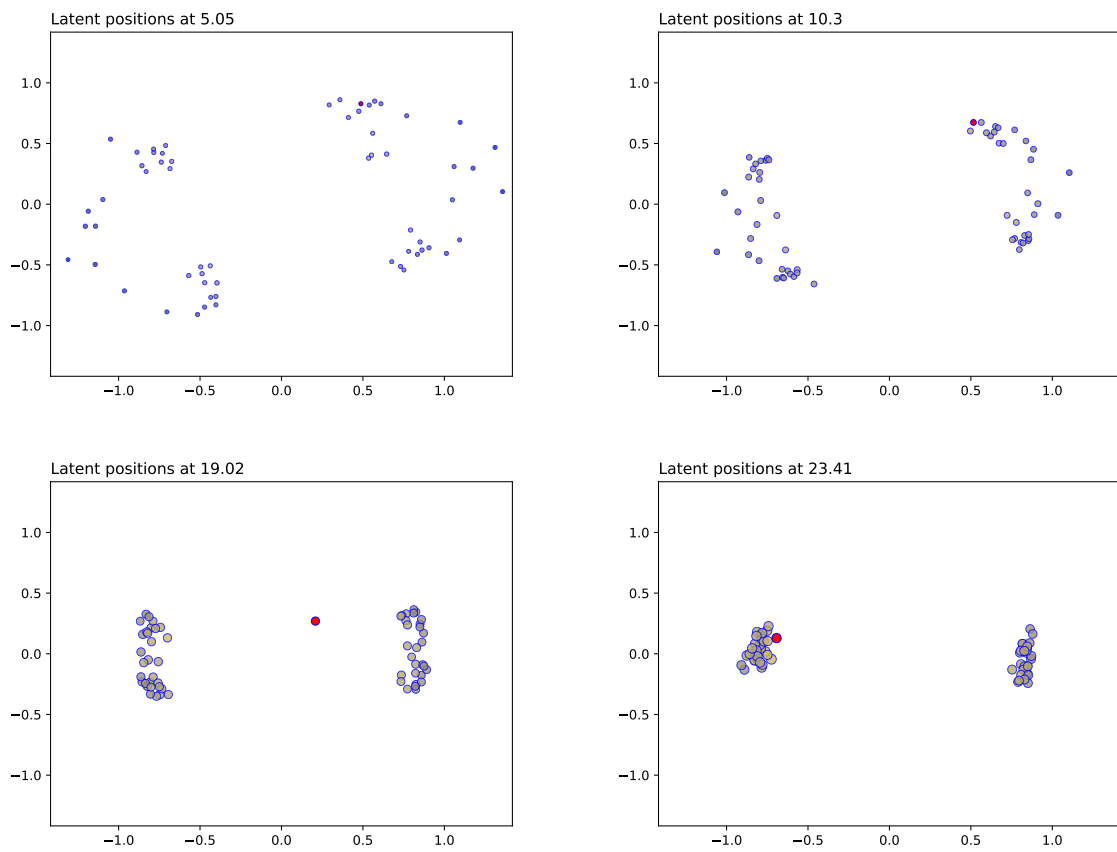


Figure 4: *Simulation study 2: snapshots for the distance model.*

5.2 Comparison with the static LPM

For simulation study 2, we propose a comparison of our results with a static LPM, as per Hoff et al. (2002). We use an implementation of the static LPM available from the R package `latentnet`.

In order to make the results comparable, we divide the time interval of 40 seconds into 80 sub-intervals of 0.5 seconds each. Then, within each sub-interval, we aggregate the interaction data by creating an edge between all those nodes that have at least one interaction. By doing so, we obtain a sequence of 80 binary undirected networks, on which we fit the distance model of Hoff et al. (2002).

We propose the visual results for 4 sub-intervals in Figure 5. From these results, we make the following observations:

- The static LPM does not capture the transition of the special node from one community to the other.
- The static LPM does not provide a model-based framework to make the snapshots comparable across time-frames, due to rotations, translations and reflections.
- The static LPM does not provide an initial strong separation of the communities, due to the adaption that is made in discretizing the data over time.

On the other hand, the CLPM can address these issues directly by providing a continuous-time evolution and thus a more accurate representation of the trajectories, without using any ad-hoc data transformation.

5.3 Distance Model

Simulation study 3. In this simulation study, we generate data from the latent distance model itself (Section 2.2.2). In this case, our goal can be more ambitious and thus we aim at reconstructing the individual trajectory of each of the nodes, at every point in time, as accurately as possible. To make the reading of the results easier, we assume that the nodes move along some pre-determined trajectories that are easy to visualize. The $N = 20$ nodes start on a ring which is centered at the origin of the space, and has radius equal to 1. The nodes are located consecutively and in line along the ring, with equal space in between any two consecutive nodes. Then, they start to move at constant speed towards the centre of the space, which they reach after 5 seconds. After reaching

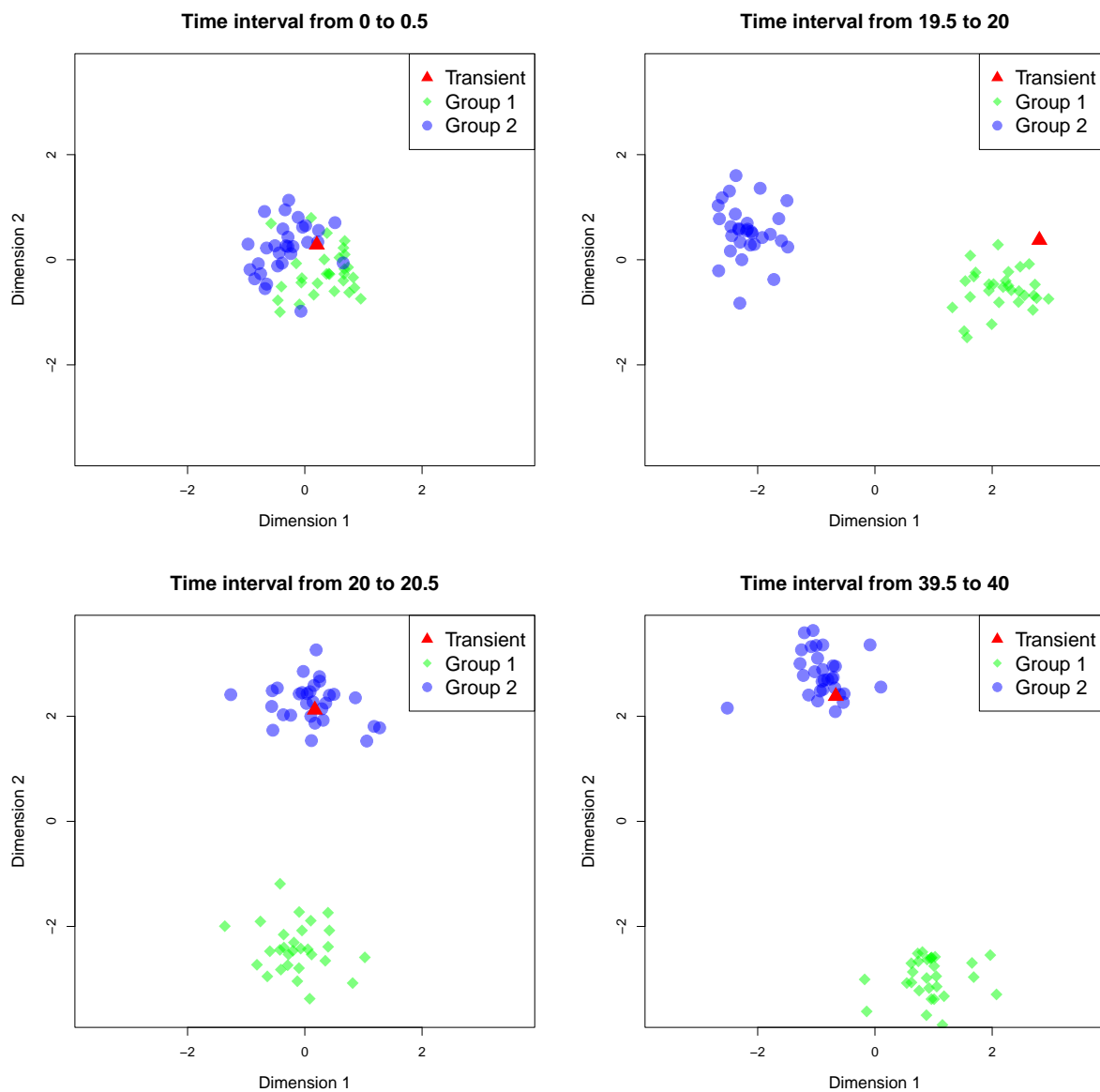


Figure 5: *Simulation study 2: fitted static LPM on 4 sub-intervals. Colors indicate the cluster membership, with one node in red being the transient node that changes community.*

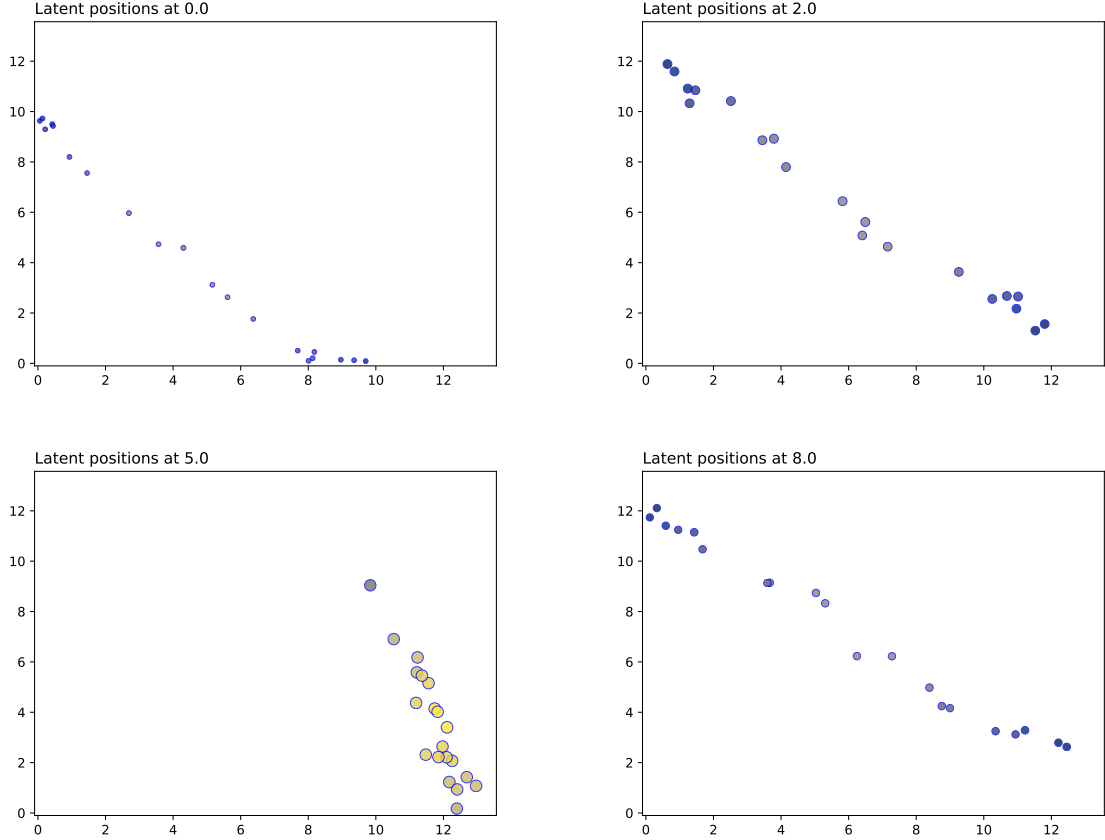


Figure 6: *Simulation study 3: snapshots for the projection model.*

the centre, they perform the same motion backwards, and they are back at their initial positions after 5 more seconds. The trajectories of the nodes make it so that, when the nodes are along the largest ring, their rate of interaction is essentially zero, however the rate increases as they are closer and closer to the center of the space.

Figure 6 shows a collection of snapshots for the projection model. The nodes are approximately equally spaced along a line and they progress outwards from the centre of the space. As they get far apart from the centre and from each other, their dot products increase and so do their interaction rates. The projection model, which is not the same model that has generated the data, tends to spread out the nodes on the space, which is ideal and expected from these data. However, this means that some of the nodes almost point in perpendicular directions, which is at odds with the fact that, half-way through the study, all nodes should interact with all others.

As concerns the results for the distance model, these are shown in Figure 7, and they highlight that the true trajectories are essentially accurately recovered. The model can

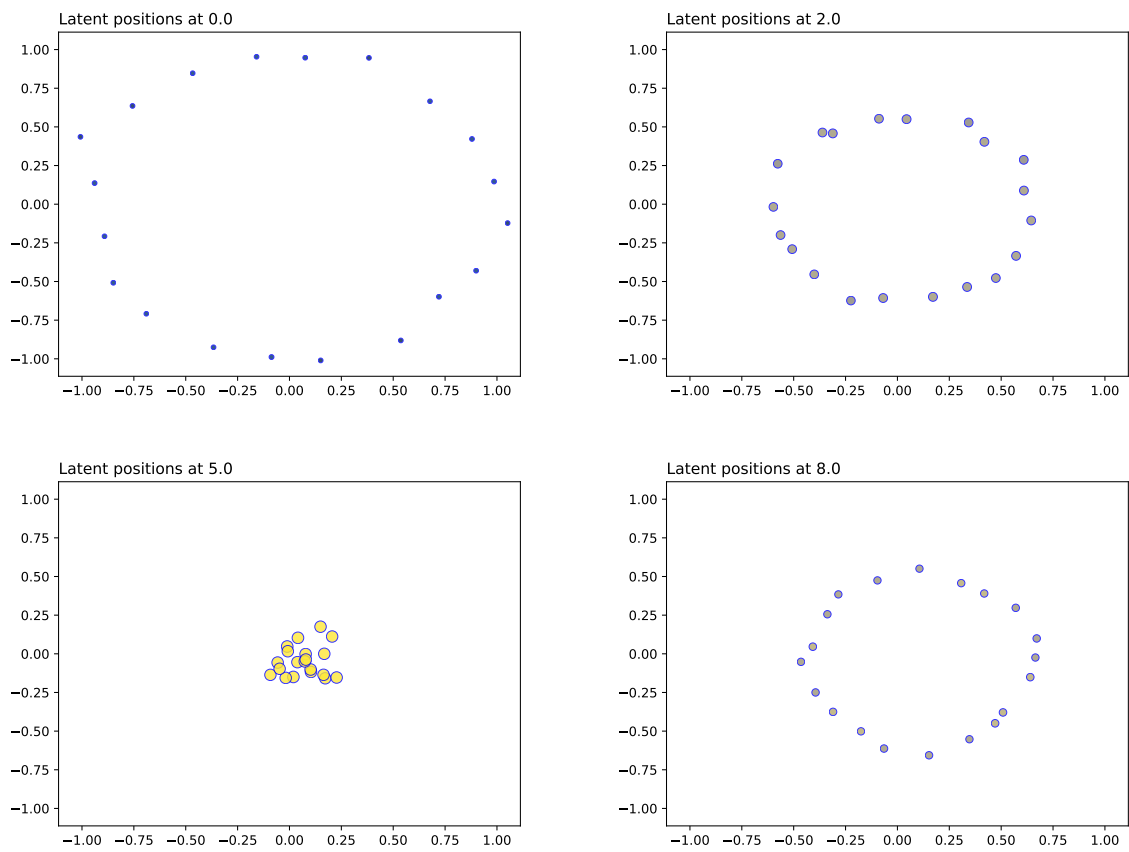


Figure 7: *Simulation study 3: snapshots for the distance model.*

capture really well the contraction and expansion of the latent space, and the individual trajectories of the nodes are closely following the theoretical counterparts. The scale of the latent space is also correctly estimated since the largest ring has approximately radius 1.

There are some important remarks to make. First, after 5 seconds, i.e. when all nodes are located close to the centre, it is understandable that a rotation or reflection (with respect to the origin of the space) may happen. This is inevitable since the solution can only be recovered up to a rotation/reflection of all the latent trajectories, but also because the first 5 seconds and the last 5 seconds can technically be seen as two independent problems. The collapse to zero can be seen as a reset in terms of orientation of the latent space. That is because the penalization terms only work with two consecutive change points, so, if we view them as identifiability constraints, they would lose their effectiveness when all the nodes collapse to zero for some time. A second fundamental remark is that the estimation procedure can lead to good results only if we observe an appropriate number of interactions. This is a specific trait of latent position models in general, since we can only guess the position of one node accurately when we know to whom it connects (or, in this context, how frequently), as we would tend to locate it close to its neighbors. In our simulated setting, there are few to no interactions when nodes are along the largest ring, so it makes sense that the results seem a bit more noisy in those instants.

5.4 Running times

We report some results aiming at quantifying the time gain due to the use of the stochastic gradient descent (SGD) algorithm detailed in Section 3. We simulated dynamic networks according to the setup of Simulation study 2 (Section 5.1) with number of nodes varying in the range $\{30, 60, \dots, 180\}$. For each number of nodes, the distance model was first fit to the data with (mini) batch size $N/10$ and 25 epochs⁴. This settings were checked to be sufficient to numerically reach a stationary point (\mathbf{Z}, β) stabilizing the log-likelihood. The average log-likelihood (say l) on the last epoch was computed and a full-batch gradient descent (GD) algorithm was independently run on the same data and stopped either once reaching l or after 250 epochs. The same initial (random) values for \mathbf{Z} and β were adopted for both SGD and GD, with a learning rate of 1.0^{-4} for \mathbf{Z} and 1.0^{-7} for β . The running

⁴In more details, groups of $N/10$ nodes were selected uniformly at random in the graph. The k -th epoch is concluded once all nodes were selected at least k times.

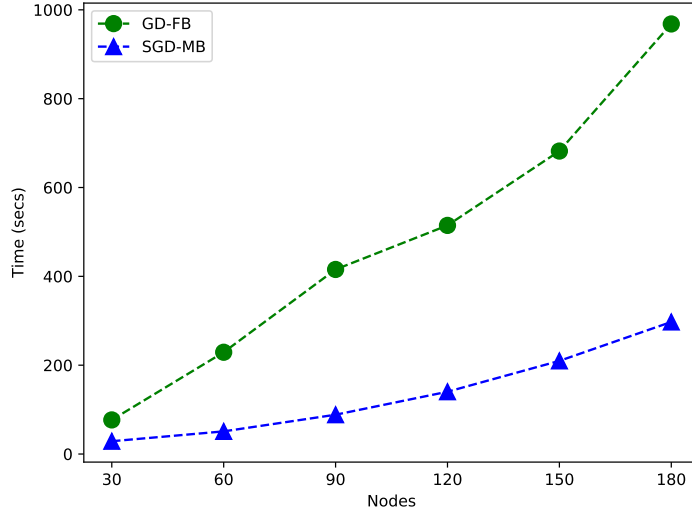


Figure 8: time (in seconds) needed to maximize the penalized log-likelihood with full-batch GD (green) and mini-batch SGD (blue).

times (seconds) needed to reach l for each number of nodes are reported in Figure 8. As it can be seen, with $N = 180$ nodes, the full-batch GD algorithm roughly needs 16 minutes to converge versus 4 minutes required by the mini-batch SGD counterpart. The experiment was run on a DELL server PowerEdge T640, equipped with an Intel Xeon processor, 28 cores (12 cores and 27 of RAM memory available on a dedicated virtual machine) and a NVIDIA GeForce RTX 2080 graphic card. The optimization algorithms (GD and SGD) were coded in PyTorch and exploited GPU acceleration.

6 Applications

In this section, we illustrate our approach over 3 real datasets, highlighting how we can characterize the trajectories of individual nodes, the formation and dissolution of communities, and other types of connectivity patterns. From the simulation studies, we have pointed out that the distance model generally provides a more convenient and appropriate framework to study these aspects of the data. In addition, it is also easier to interpret, so, we only show the results for the distance model and redirect the reader to the associated code repository where the complete results can be found.

6.1 ACM Hypertext conference dataset

The ACM Hypertext 2009 conference was held over three days in Turin, Italy, from June 29 to July 1. At the conference, 113 attendees wore special badges which recorded an

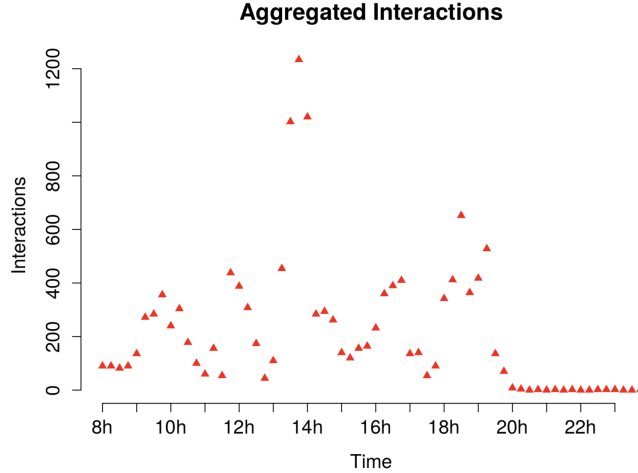


Figure 9: *ACM application*: cumulative number of interactions for each quarter hour (first day).

interaction whenever two badges were facing each other at a distance of 1.5 meters or less, for at least 20 seconds. For each of these interactions, a timestamp was recorded as well as the identifiers of the two personal badges.

This interaction dataset was first analyzed by Isella et al. (2011), and is publicly available from *Hypertext 2009 network dataset – KONECT* (2017). Similarly to Corneli et al. (2016), we focus our analysis on the first day of the conference. On the first day, the main events that took place included a poster session in the morning (starting from 8 a.m.), a lunch break around 1 p.m., and a cheese and wine reception in the evening between 6 p.m. and 7 p.m. We use our distance CLPM to provide a graphical representation of these data, and to note how the model responds to the various gatherings that happened during the day. We set a change point (η_k) every fifteen minutes. Figures 10 and 11 provide a number of snapshots highlighting some of the relevant moments of the day. The complete results, shown as a video, can be found in the code repository that accompanies this paper.

We can see that, in the morning, there is a high level of mixing between the attendees. The visitors tend to merge and split into different communities that change very frequently and very randomly. These communities reach a high level of clusteredness, which signals that the participants of the study are mixing into different groups. This is perfectly in agreement with the idea that the participants are moving from one location to another, as it usually happens during poster sessions and parallel talk sessions. The nodes exhibit different types of patterns and behaviours, in that some nodes are central and tend to

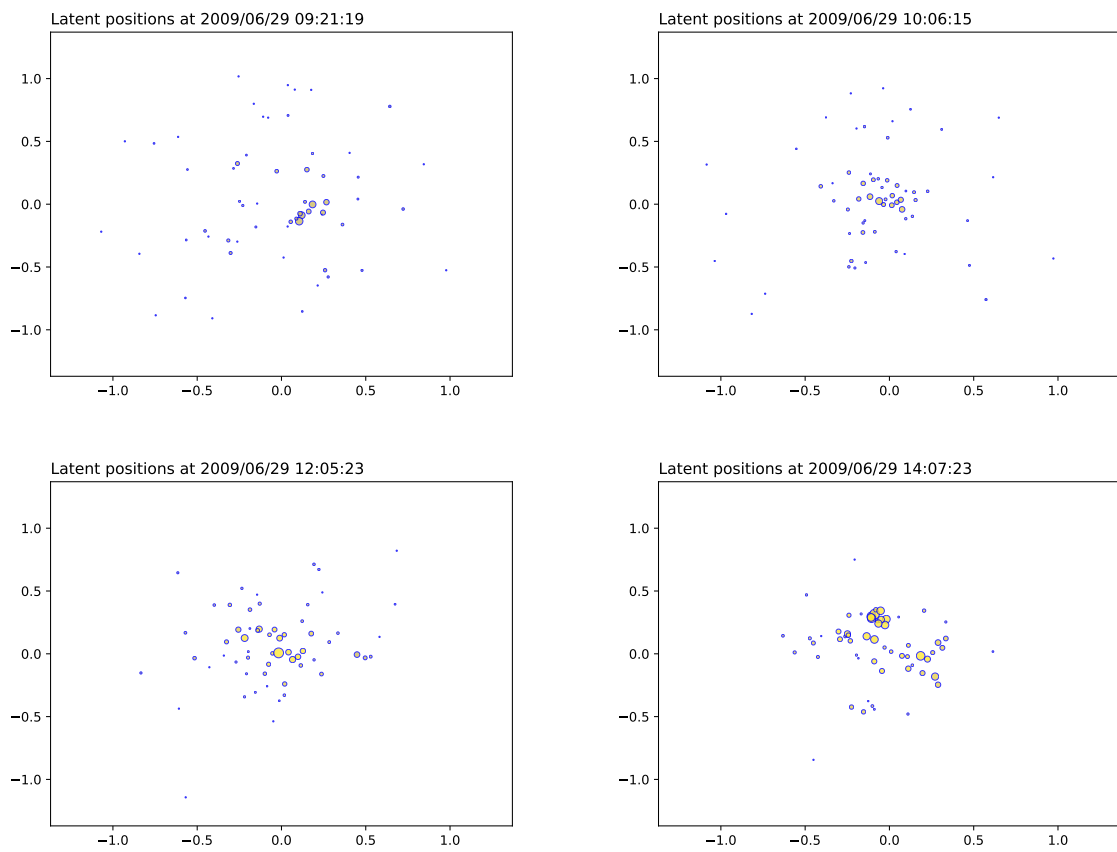


Figure 10: **ACM application**: snapshots for the distance model (morning hours).

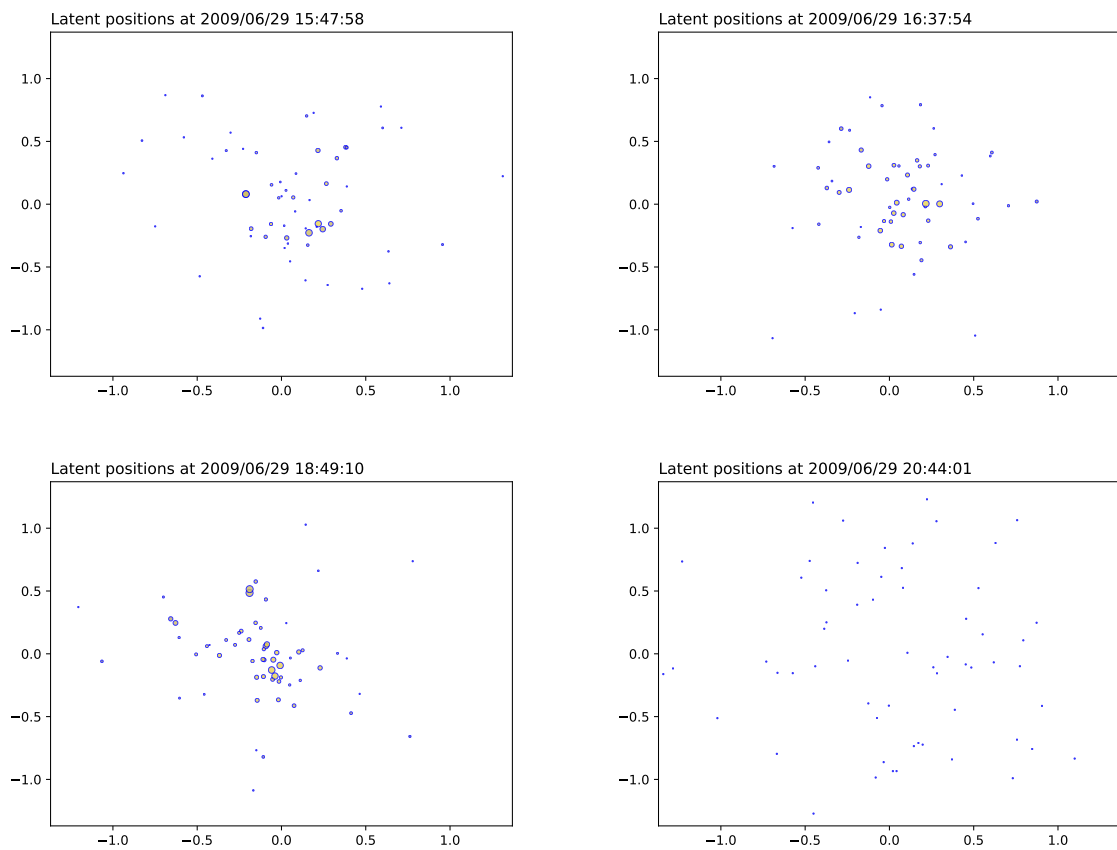


Figure 11: *ACM application: snapshots for the distance model (afternoon hours).*

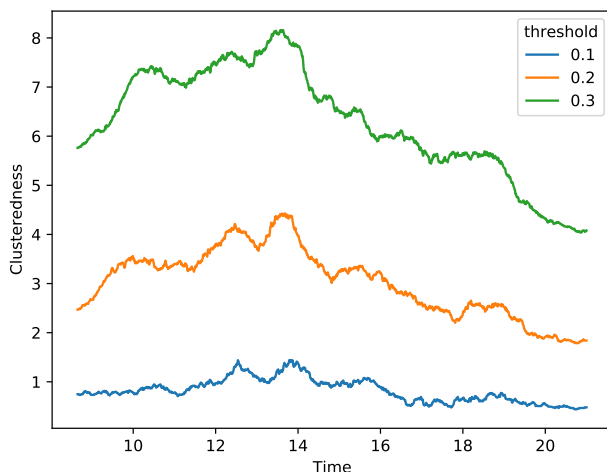


Figure 12: **ACM application:** clusteredness measure for various threshold values. The x-axis shows the hour of the day, whereas the y-axis shows the average number of nodes that a random node would have within the threshold distance.

join many communities, whereas others have lower levels of participation and remain at the outskirts of the space.

In the late morning, we see a clear close gathering around 12 p.m., whereby almost all nodes move towards the centre of the space. This is emphasized even more at 2 p.m., which corresponds to the lunch break. It is especially interesting that, even though the space becomes more contracted at this time, we can still clearly see a strong clustering structure.

In the afternoon, we go back to the same patterns as in the morning, whereby the participants mix in different groups and move around the space. The wine reception is also clearly captured around 6 p.m. where we see again some level of contraction of the space, to signal a large gathering of the participants.

After this event, the overall rate of interactions diminishes sharply, and as a consequence we see the nodes spreading out in the space.

In terms of clustering, we use the index introduced in Section 4 on the results for the distance model, for various threshold levels. The results are shown in Figure 12, where we can appreciate strong time-dynamic patterns. All threshold values highlight several peaks for the clustering measure, confirming the aggregations happening throughout the morning, at lunch time, and in the evening. In addition, we can also highlight a number of smaller cycles, which are congruent with the creation and dissolution of small communities as may be observed during parallel sessions or poster presentations.

6.2 Reality mining

The reality mining dataset (Eagle and Pentland 2006) is derived from the Reality Commons project, which was run at the Massachusetts Institute of Technology (MIT) from 14 September 2004 to 5 May 2005. The dataset describes proximity interactions in a group of 96 students, collected primarily through bluetooth devices. An overview of this network dataset is also given by Rastelli (2019).

In the context of this paper, the proximity interactions can be reasonably considered as instantaneous interactions, due to the study being 9 months long. With our latent space representation, we aim at highlighting the patterns of connections of the students during the study, and any social communities that arise and how these change over time.

Figure 13 shows a few snapshots of our fitted distance CLPM. The complete results, shown as a video, can be found in the code repository that accompanies this paper.

We observe that, in general, the students are quite separated and few communities arise. This does not necessarily mean that the nodes do not interact, but it is a sign that there are no subgroups of students with an uncommonly high interaction rate. Also, it is important to point out that the latent space is very expanded: coupled with an estimated intercept value of 3.9, this signals that the latent space has a strong effect on modeling the interaction rates and that it can capture well the variability in the data.

Over time, the students tend to mix in different social groups, thus quickly forming and undoing communities. This could be explained by the interactions that the students have due to college activities or other daily activities. Near the end of the study, a large cluster appears, signaling a large gathering to which the students participated. This may correspond to the period before a deadline, as outlined in Eagle and Pentland (2006).

6.3 London bikes

Infrastructure networks provide an excellent example of instantaneous interaction data. In this section we consider a network of bike hires which is collected and publicly distributed by *Transport for London: Cycle hire usage data 2012–2015*. We focus on a specific weekend day (Sunday 6 September 2015), and study the patterns of interactions between all bike hire stations in London. The bike hire stations correspond to the nodes of our network, whereas an instantaneous interaction between two nodes at time t simply means that a bike started a journey from one station towards the other, at that time (we consider undirected connections).

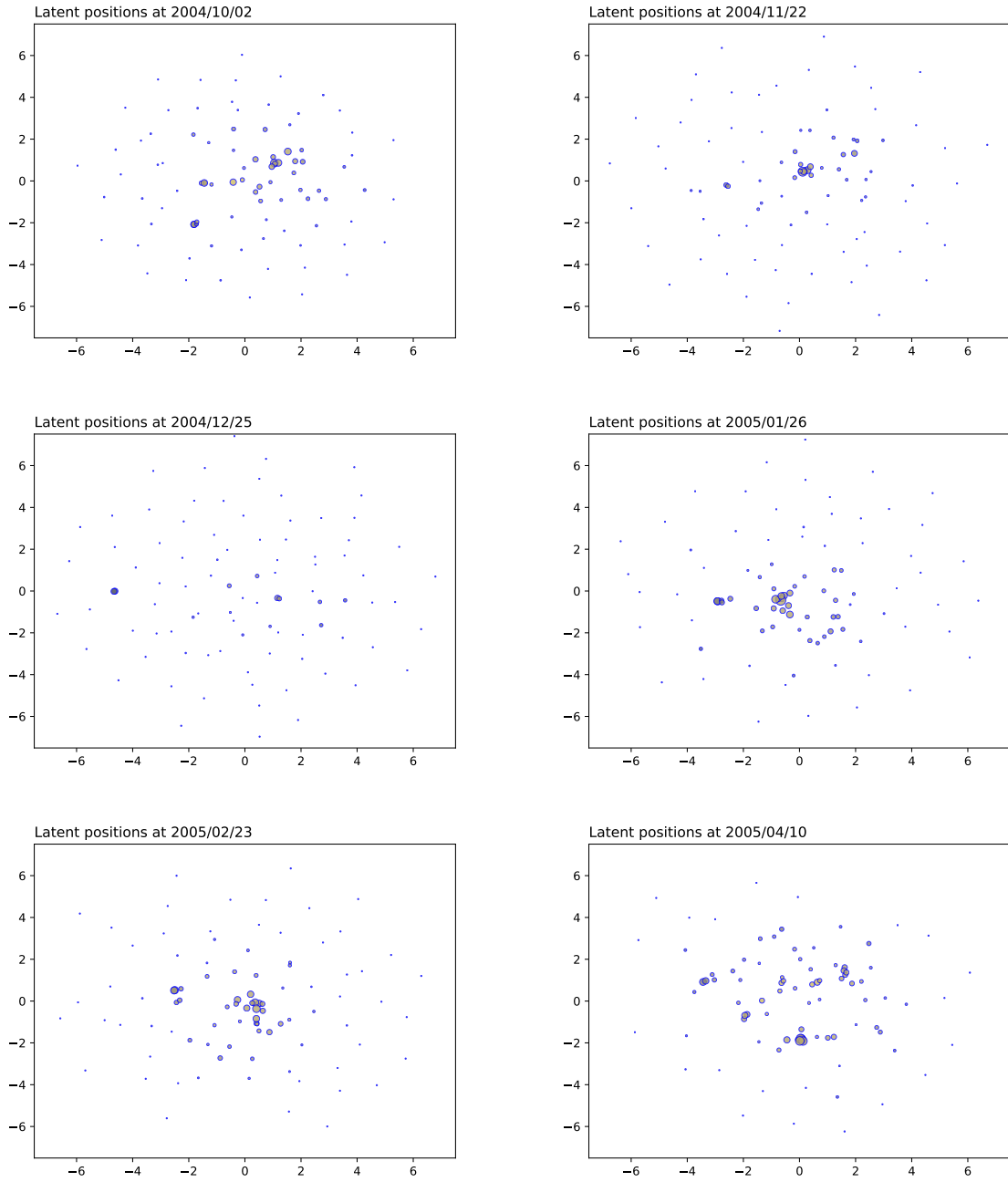


Figure 13: MIT application: snapshots for the distance model.

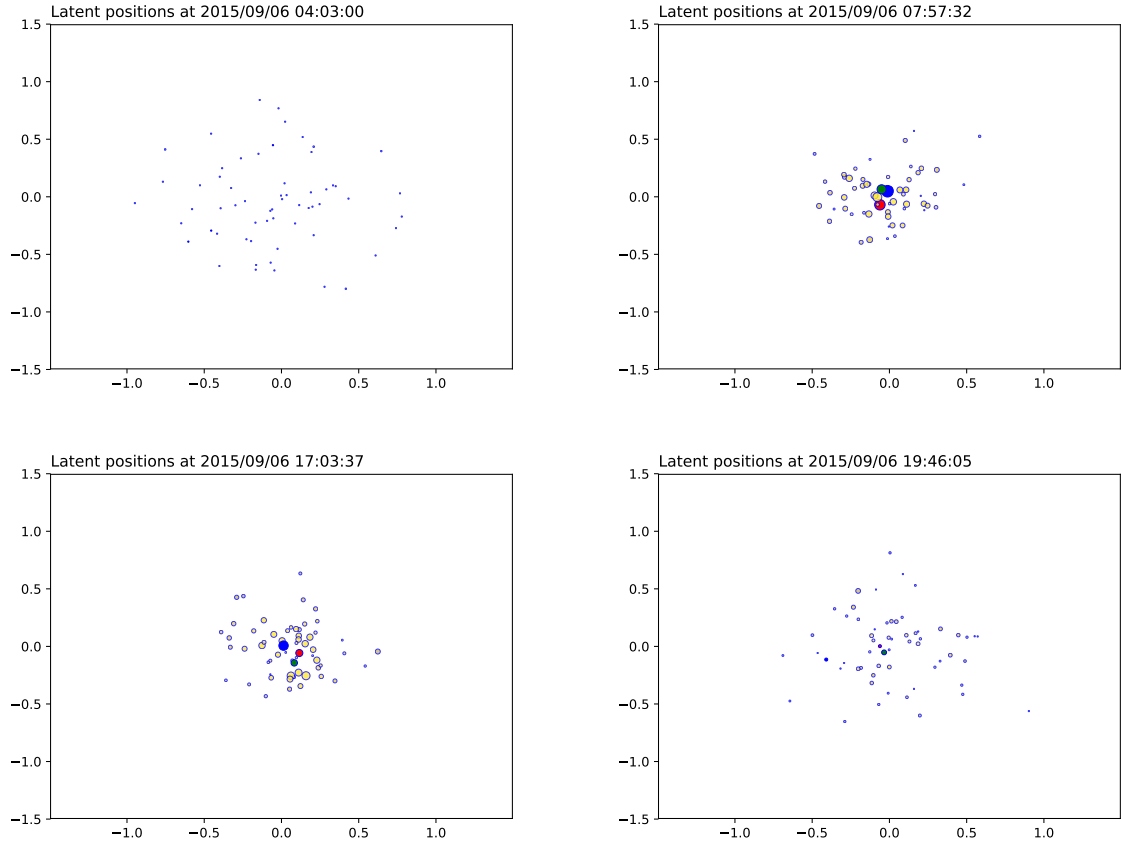


Figure 14: *London bikes application: snapshots for the distance model.*

In Figure 14 we show a collection of snapshots at some critical time points during the day, for the distance model. The complete results, shown as a video, can be found in the code repository that accompanies this paper.

Although there are a total of 818 stations that are active in this dataset, we provide a visualization for the 60 most active stations only. However, we emphasize that the results were obtained using the whole dataset. In addition, we highlight with a different color the 3 stations with the highest number of interactions, overall. These stations are:

- Belgrove Street, King’s Cross, situated next to King’s Cross Square (shown in red);
- Finsbury Circus, Liverpool Street, situated next to Liverpool Street station (shown in blue);
- Newgate Street, St. Paul’s, situated next to St. Paul’s Cathedral (shown in green).

The first aspect that we notice is that the latent space expands during inactive times,

and it contracts during busy hours. The contractions and expansions are not homogeneous, rather they highlight the presence of dense and less dense clusters of stations.

The estimate for the intercept parameter is -5.2 , and the dispersion of the points in the latent space is not particularly large. This signals that the latent space characterization is not having a very strong effect on the rate of interactions, and the model does not capture much variability in the rates of interactions. This highlights that these connectivity data follow patterns that cannot be completely explained by the purely geometrical nature of our model. That is, the connections are determined by a variety of factors that cannot be framed into this latent positions context, and the geographical information on bike hiring accounts for only a part of the problem.

7 Conclusions

We have introduced a new time-continuous version of the well known and widely used latent position model, as an extension which can model instantaneous interactions between entities. We have proposed a new methodology which provides good flexibility while also allowing for an efficient inferential framework. The methodology is implemented in our software CLPM which accompanies this paper and is publicly available. This provides an essential additional tool for practitioners that are interested in deriving latent space visualizations from observed instantaneous interaction data.

The framework that we propose is highly inspired by the work of Hoff et al. (2002), and by the vast literature that has followed in this direction. Our work combines some crucial theoretical and statistical aspects of latent position modeling, with a pragmatic approach to inference and visualization of the results. Crucially, we provide simulation studies and real data applications to demonstrate how our method leads to sensible and accurate results, with low computational demands.

As regards extensions and future work, our research opens up several new directions, to address and potentially change some crucial parts of our procedure. A fundamental challenge is related to the geometric nature of the latent space. In this paper, and in the literature cited here, affine latent spaces are considered, endowed with the standard dot product, which, in turn, induces the Euclidean distance. However, some important works in the literature of the *static* latent position model consider the latent space to be spherical (McCormick and Zheng 2015) or hyperbolic (Krioukov et al. 2010; Asta and Shalizi 2015). As expected, since latent position models are generative models, the

geometry of the latent space has crucial consequences on the properties of the simulated network (Smith et al. 2019). Indeed, recently, Lubold et al. (2020) introduced a method to consistently estimate the manifold type, dimension, and curvature from a class of latent spaces. Addressing these topics in the context of dynamic latent position models is a promising avenue of research that can extend our work. Another challenging aspect of our methodology regards inference: in this paper, we propose an optimization approach to maximize a penalized likelihood. An interesting alternative would be to consider a different approach that could allow one to also quantify uncertainty around the parameter estimates. Finally, in terms of modeling, we use piece-wise continuous trajectories due to their flexibility and easy tractability, however, alternatives to this parametrization may also be considered.

Acknowledgements

The authors would like to thank Prof. Charles Bouveyron for useful feedback in the early stages of this work, but also for supporting and coordinating a research visit which fostered this project.

References

- Asta, D. M. and C. R. Shalizi (2015). “Geometric network comparisons”. In: *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pp. 102–110.
- Barrat, A. and C. Cattuto (2013). “Temporal networks of face-to-face human interactions”. In: *Temporal Networks*. Springer, pp. 191–216.
- Baydin, A. G., B. A. Pearlmutter, A. A. Radul, and J. M. Siskind (2018). “Automatic differentiation in machine learning: a survey”. In: *Journal of machine learning research* 18.
- Bottou, L. (2010). “Large-scale machine learning with stochastic gradient descent”. In: *Proceedings of COMPSTAT’2010*. Springer, pp. 177–186.
- Cattuto, C., W. Van den Broeck, A. Barrat, V. Colizza, J.-F. Pinton, and A. Vespignani (2010). “Dynamics of person-to-person interactions from distributed RFID sensor networks”. In: *PloS one* 5.7, e11596.
- CLPM GitHub repository* (Mar. 2021). URL: <https://github.com/marcogenni/CLPM>.
- Corneli, M., P. Latouche, and F. Rossi (2016). “Exact ICL maximization in a non-stationary temporal extension of the stochastic block model for dynamic networks”. In: *Neurocomputing* 192, pp. 81–91.
- Corneli, M., P. Latouche, and F. Rossi (2017). “Multiple change points detection and clustering in dynamic networks”. In: *Statistics and Computing*, pp. 1–19.
- Durante, D. and D. B. Dunson (2014). “Nonparametric Bayes dynamic modelling of relational data”. In: *Biometrika* 101.4, pp. 883–898.

- Durante, D. and D. B. Dunson (2016). “Locally adaptive dynamic networks”. In: *The Annals of Applied Statistics* 10.4, pp. 2203–2232.
- Durante, D., D. B. Dunson, and J. T. Vogelstein (2017). “Nonparametric Bayes modeling of populations of networks”. In: *Journal of the American Statistical Association*, pp. 1–15.
- Eagle, N. and A. S. Pentland (2006). “Reality mining: sensing complex social systems”. In: *Personal and ubiquitous computing* 10.4, pp. 255–268.
- Fosdick, B. K., T. H. McCormick, T. B. Murphy, T. L. J. Ng, and T. Westling (2019). “Multiresolution network models”. In: *Journal of Computational and Graphical Statistics* 28.1, pp. 185–196.
- Friel, N., R. Rastelli, J. Wyse, and A. E. Raftery (2016). “Interlocking directorates in Irish companies using a latent space model for bipartite networks”. In: *Proceedings of the National Academy of Sciences* 113.24, pp. 6629–6634.
- Gollini, I. and T. B. Murphy (2014). “Joint modelling of multiple network views”. In: *Journal of Computational and Graphical Statistics*.
- Gormley, I. C. and T. B. Murphy (2007). “A latent space model for rank data”. In: *Statistical Network Analysis: Models, Issues, and New Directions*. Springer, pp. 90–102.
- Griewank, A. (1989). “On automatic differentiation”. In: *Mathematical Programming: recent developments and applications* 6.6, pp. 83–107.
- Halko, N., P.-G. Martinsson, and J. A. Tropp (2011). “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions”. In: *SIAM review* 53.2, pp. 217–288.
- Handcock, M. S., A. E. Raftery, and J. M. Tantrum (2007). “Model-based clustering for social networks”. In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 170.2, pp. 301–354.
- Hoff, P. D. (2005). “Bilinear mixed-effects models for dyadic data”. In: *Journal of the American Statistical Association* 100.469, pp. 286–295.
- Hoff, P. D. (2018). “Additive and multiplicative effects network models”. In: *arXiv preprint arXiv:1807.08038*.
- Hoff, P. D., A. E. Raftery, and M. S. Handcock (2002). “Latent space approaches to social network analysis”. In: *Journal of the American Statistical Association* 97.460, pp. 1090–1098.
- Hypertext 2009 network dataset – KONECT* (Oct. 2017). URL: <http://konect.cc/networks/sociopatterns-hypertext>.
- Isella, L., J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck (2011). “What’s in a Crowd? Analysis of Face-to-Face Behavioral Networks”. In: *J. of Theoretical Biology* 271.1, pp. 166–180.
- Junuthula, R., M. Haghdan, K. S. Xu, and V. Devabhaktuni (2019). “The block point process model for continuous-time event-based dynamic networks”. In: *The World Wide Web Conference*, pp. 829–839.
- Kim, B., K. H. Lee, L. Xue, and X. Niu (2018). “A review of dynamic network models with latent variables”. In: *Statistics surveys* 12, p. 105.
- Klimt, B. and Y. Yang (2004). “The enron corpus: A new dataset for email classification research”. In: *European Conference on Machine Learning*. Springer, pp. 217–226.

- Krioukov, D., F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguná (2010). “Hyperbolic geometry of complex networks”. In: *Physical Review E* 82.3, p. 036106.
- Krivitsky, P. N., M. S. Handcock, A. E. Raftery, and P. D. Hoff (2009). “Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models”. In: *Social networks* 31.3, pp. 204–213.
- Lee, D. D. and H. S. Seung (1999). “Learning the parts of objects by non-negative matrix factorization”. In: *Nature* 401.6755, pp. 788–791.
- Lubold, S., A. G. Chandrasekhar, and T. H. McCormick (2020). *Identifying the latent space geometry of network models through analysis of curvature*. Tech. rep. National Bureau of Economic Research.
- Matias, C., T. Rebafka, and F. Villers (2018). “A semiparametric extension of the stochastic block model for longitudinal networks”. In: *Biometrika* 105.3, pp. 665–680.
- McCormick, T. H. and T. Zheng (2015). “Latent surface models for networks using Aggregated Relational Data”. In: *Journal of the American Statistical Association* 110.512, pp. 1684–1695.
- Nowicki, K. and T. A. B. Snijders (2001). “Estimation and prediction for stochastic blockstructures”. In: *Journal of the American statistical association* 96.455, pp. 1077–1087.
- Park, H.-J. and K. Friston (2013). “Structural and functional brain networks: from connections to cognition”. In: *Science* 342.6158.
- Raftery, A. E. (2017). “Comment: extending the latent position model for networks”. In: *Journal of the American Statistical Association* 112.520, pp. 1531–1534.
- Raftery, A. E., X. Niu, P. D. Hoff, and K. Y. Yeung (2012). “Fast inference for the latent space network model using a case-control approximate likelihood”. In: *Journal of Computational and Graphical Statistics* 21.4, pp. 901–919.
- Rahimi, A. and B. Recht (2007). “Random Features for Large-Scale Kernel Machines.” In: *NIPS*. Vol. 3. 4. Citeseer, p. 5.
- Rastelli, R. (2019). “Exact integrated completed likelihood maximisation in a stochastic block transition model for dynamic networks”. In: *J French Stat Soc* 160.1, pp. 35–56.
- Rastelli, R., N. Friel, and A. E. Raftery (2016). “Properties of latent variable network models”. In: *Network Science* 4.4, pp. 407–432. DOI: 10.1017/nws.2016.23.
- Rastelli, R., F. Maire, and N. Friel (2018). “Computationally efficient inference for latent position network models”. In: *arXiv preprint arXiv:1804.02274*.
- Salter-Townshend, M. and T. H. McCormick (2017). “Latent space models for multiview network data”. In: *The Annals of Applied Statistics* 11.3, pp. 1217–1244.
- Salter-Townshend, M., A. White, I. Gollini, and T. B. Murphy (2012). “Review of statistical network analysis: models, algorithms, and software”. In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 5.4, pp. 243–264.
- Sarkar, P. and A. W. Moore (2006). “Dynamic social network analysis using latent space models”. In: *Advances in Neural Information Processing Systems*, pp. 1145–1152.
- Sewell, D. K. and Y. Chen (2015a). “Analysis of the formation of the structure of social networks by using latent space models for ranked dynamic networks”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 64.4, pp. 611–633.
- Sewell, D. K. and Y. Chen (2015b). “Latent space models for dynamic networks”. In: *Journal of the American Statistical Association* 110.512, pp. 1646–1657.

- Sewell, D. K. and Y. Chen (2016). “Latent space models for dynamic networks with weighted edges”. In: *Social Networks* 44, pp. 105–116.
- Shortreed, S., M. S. Handcock, and P. Hoff (2006). “Positional estimation within a latent space model for networks”. In: *Methodology* 2.1, pp. 24–33.
- Smith, A. L., D. M. Asta, and C. A. Calder (2019). “The geometry of continuous latent space models for network data”. In: *Statistical science: a review journal of the Institute of Mathematical Statistics* 34.3, p. 428.
- Sosa, J. and L. Buitrago (2020). “A Review of Latent Space Models for Social Networks”. In: *arXiv preprint arXiv:2012.02307*.
- Tafakori, L., A. Pourkhanali, and R. Rastelli (2019). “Measuring systemic risk and contagion in the European financial network”. In: *arXiv preprint arXiv:1911.11488*.
- Transport for London: Cycle hire usage data 2012–2015*. <http://cycling.data.tfl.gov.uk/>. Accessed: 2015-09-6.

A Log-likelihood for the projection model

To make the forthcoming results clearer, we introduce the following notation for the dot products:

$$S_{ij}^{gh} := \langle \mathbf{z}_i(\eta_g), \mathbf{z}_j(\eta_h) \rangle \quad (11)$$

for some nodes i and j , and for any $g, h \in \{1, \dots, K\}$. Most commonly, g and h will correspond to the labels of two consecutive change points, thus we will use them to identify two breakpoints for the trajectories, and to reconstruct the positions in between. Also, we denote:

$$S_{ij}(t) := \langle \mathbf{z}_i(t), \mathbf{z}_j(t) \rangle$$

for the dot product at a generic time $t \in [0, T]$.

Proposition 2. *Under the projection model, the log-likelihood is exactly equal to:*

$$\log \mathcal{L}(\mathbf{Z}) = \sum_{i,j: i < j} \left\{ \sum_{e \in \mathcal{E}_{ij}} \log S_{ij}(\tau_e) - \frac{1}{6} \sum_{g=1}^{K-1} (\eta_h - \eta_g) \left(2S_{ij}^{gg} + S_{ij}^{gh} + S_{ij}^{hg} + 2S_{ij}^{hh} \right) \right\} \quad (12)$$

where $h = g + 1$.

Proof. The only non-trivial part in formula (4) regards the integral:

$$\int_0^T \langle \mathbf{z}_i(s), \mathbf{z}_j(s) \rangle ds$$

where we have that, for $s \in [\eta_g, \eta_{g+1}]$:

$$\mathbf{z}_i(s) = (1-t)\mathbf{z}_i(\eta_g) + t\mathbf{z}_i(\eta_{g+1})$$

and, thanks to Eq. (3), $t \in [0, 1]$ is such that $s = (1-t)\eta_g + t\eta_{g+1}$. Back to the calculation of the integral, we can decompose this into $K-1$ integrals using the K change points as follows:

$$\int_0^T \langle \mathbf{z}_i(s), \mathbf{z}_j(s) \rangle ds = \sum_{g=1}^{K-1} \int_{\eta_g}^{\eta_{g+1}} \langle \mathbf{z}_i(s), \mathbf{z}_j(s) \rangle ds$$

and then, in each integral, we apply the transformation $s = (1-t)\eta_g + t\eta_{g+1}$ to obtain:

$$\begin{aligned} \int_0^T \langle \mathbf{z}_i(s), \mathbf{z}_j(s) \rangle ds &= \sum_{g=1}^{K-1} (\eta_{g+1} - \eta_g) \int_0^1 \langle \mathbf{z}_i((1-t)\eta_g + t\eta_{g+1}), \mathbf{z}_j((1-t)\eta_g + t\eta_{g+1}) \rangle dt \\ &= \sum_{g=1}^{K-1} (\eta_{g+1} - \eta_g) \int_0^1 \left[(1-t)^2 S_{ij}^{gg} + t(1-t)(S_{ij}^{gh} + S_{ij}^{hg}) + t^2 S_{ij}^{hh} \right] dt \end{aligned}$$

where S_{ij}^{gh} is defined in Eq. (11) and $h = g + 1$. Since the S terms do not depend on t , the integral can be solved analytically:

$$\begin{aligned} \int_0^T \langle \mathbf{z}_i(s), \mathbf{z}_j(s) \rangle ds &= \sum_{g=1}^{K-1} (\eta_{g+1} - \eta_g) \left[S_{ij}^{gg} \int_0^1 (1-t)^2 dt + (S_{ij}^{gh} + S_{ij}^{hg}) \int_0^1 t(1-t) dt + S_{ij}^{hh} \int_0^1 t^2 dt \right] \\ &= \sum_{g=1}^{K-1} (\eta_{g+1} - \eta_g) \left[\frac{S_{ij}^{gg}}{3} + \frac{S_{ij}^{gh} + S_{ij}^{hg}}{6} + \frac{S_{ij}^{hh}}{3} \right] \end{aligned}$$

which is equivalent to the result of the integral appearing in Eq. (12). \square

B Log-likelihood for the distance model

We focus on the integral in Eq. (6) and prove that it can be explicitly solved.

Proof. In force of Eq. (3), it reads

$$\begin{aligned}
& e^\beta \left(\sum_{g=1}^{K-1} \int_0^1 \exp \left\{ - \left\| (1-t)(\mathbf{z}_i(\eta_g) - \mathbf{z}_j(\eta_g)) + t(\mathbf{z}_i(\eta_{g+1}) - \mathbf{z}_j(\eta_{g+1})) \right\|_2^2 \right\} (\eta_{g+1} - \eta_g) dt \right) \\
&= e^\beta \left(\sum_{g=1}^{K-1} (\eta_{g+1} - \eta_g) \int_0^1 \exp \left\{ - \left\| t(\Delta_i^g - \Delta_j^g) + (\mathbf{z}_i^g - \mathbf{z}_j^g) \right\|_2^2 \right\} dt \right)
\end{aligned} \tag{13}$$

where the variable change $t = \frac{s-\eta_g}{\eta_{g+1}-\eta_g}$ was performed and the following notations were adopted to simplify the exposition

$$\begin{aligned}
\Delta_i^g &:= \mathbf{z}_i(\eta_{g+1}) - \mathbf{z}_i(\eta_g) \\
\Delta_j^g &:= \mathbf{z}_j(\eta_{g+1}) - \mathbf{z}_j(\eta_g) \\
\mathbf{z}_i^g &:= \mathbf{z}_i(\eta_g) \\
\mathbf{z}_j^g &:= \mathbf{z}_j(\eta_g)
\end{aligned}$$

By denoting $g(t) := - \left\| t(\Delta_i^g - \Delta_j^g) + (\mathbf{z}_i^g - \mathbf{z}_j^g) \right\|_2^2$ the exponent inside the integral, we can “complete the square” as follows

$$\begin{aligned}
g(t) &= - \left\| \Delta_i^g - \Delta_j^g \right\|^2 \left(t^2 + 2t \left\langle \frac{\Delta_i^g - \Delta_j^g}{\left\| \Delta_i^g - \Delta_j^g \right\|}, \frac{\mathbf{z}_i^g - \mathbf{z}_j^g}{\left\| \Delta_i^g - \Delta_j^g \right\|} \right\rangle + \frac{\left\| \mathbf{z}_i^g - \mathbf{z}_j^g \right\|^2}{\left\| \Delta_i^g - \Delta_j^g \right\|^2} \right) \\
&= - \left\| \Delta_i^g - \Delta_j^g \right\|^2 \left(t - \left\langle \frac{\Delta_i^g - \Delta_j^g}{\left\| \Delta_i^g - \Delta_j^g \right\|}, \frac{\mathbf{z}_j^g - \mathbf{z}_i^g}{\left\| \Delta_i^g - \Delta_j^g \right\|} \right\rangle \right)^2 \\
&\quad - \left(\left\| \mathbf{z}_i^g - \mathbf{z}_j^g \right\|^2 - \left(\left\langle \frac{\Delta_i^g - \Delta_j^g}{\left\| \Delta_i^g - \Delta_j^g \right\|}, \mathbf{z}_j^g - \mathbf{z}_i^g \right\rangle \right)^2 \right) \\
&= - \frac{1}{2(\sigma^2)_{ij}^g} (t - \mu_{ij}^g)^2 - \left(\left\| \mathbf{z}_i^g - \mathbf{z}_j^g \right\|^2 - (\left\| \Delta_i^g - \Delta_j^g \right\| \mu_{ij}^g)^2 \right)
\end{aligned}$$

where

$$\begin{aligned}
\mu_{ij}^g &:= \left\langle \frac{\Delta_i^g - \Delta_j^g}{\left\| \Delta_i^g - \Delta_j^g \right\|}, \frac{\mathbf{z}_j^g - \mathbf{z}_i^g}{\left\| \Delta_i^g - \Delta_j^g \right\|} \right\rangle, \\
(\sigma^2)_{ij}^g &:= \frac{1}{2 \left\| \Delta_i^g - \Delta_j^g \right\|^2}.
\end{aligned} \tag{14}$$

By plugging all this into Eq. 13 it follows that

$$\begin{aligned}
& \int_0^T e^{\beta - \left\| \mathbf{z}_i(t) - \mathbf{z}_j(t) \right\|_2^2} dt = \\
&= \sqrt{2\pi} e^\beta \left\{ \sum_{l=1}^{K-1} e^{-\left(\left\| \mathbf{z}_i^l - \mathbf{z}_j^l \right\|^2 - (\left\| \Delta_i^l - \Delta_j^l \right\| \mu_{ij}^l)^2 \right)} \sigma_{ij}^l (\eta_{l+1} - \eta_l) \left[\Phi \left(\frac{1 - \mu_{ij}^l}{\sigma_{ij}^l} \right) - \Phi \left(\frac{0 - \mu_{ij}^l}{\sigma_{ij}^l} \right) \right] \right\}
\end{aligned} \tag{15}$$

□