

Sequential Estimation of Temporally Evolving Latent Space Network Models

Kathryn Turnbull^{a,*}, Christopher Nemeth^a, Matthew Nunes^b, Tyler McCormick^c

^a*Department of Mathematics and Statistics, Lancaster University, Lancaster, UK,*

^b*School of Mathematical Sciences, University of Bath, Bath, UK,*

^c*Department of Statistics, University of Washington, Seattle, Washington, USA,*

Abstract

Dynamic network data describe interactions among a fixed population through time. This data type can be modelled using the latent space framework, where the probability of a connection forming is expressed as a function of low-dimensional latent coordinates associated with the nodes, and sequential estimation of model parameters can be achieved via Sequential Monte Carlo (SMC) methods. In this setting, SMC is a natural candidate for estimation which offers greater scalability than existing approaches commonly considered in the literature, allows for estimates to be conveniently updated given additional observations and facilitates both online and offline inference. A novel approach to sequentially infer parameters of dynamic latent space network models is proposed by building on techniques from the high-dimensional SMC literature. The scalability and performance of the proposed approach is explored via simulation, and the flexibility under model variants is demonstrated. Finally, a real-world dataset describing classroom contacts is analysed using the proposed methodology.

Keywords: Statistical Network Analysis, Sequential Monte Carlo, Latent Space, Dynamic Networks

*Corresponding author at Department of Mathematics and Statistics, Fylde College, Lancaster University, Lancaster, LA1 4YF, United Kingdom.

Email addresses: k.turnbull@lancaster.ac.uk (Kathryn Turnbull),
c.nemeth@lancaster.ac.uk (Christopher Nemeth), m.a.nunes@bath.ac.uk (Matthew Nunes), tylermc@uw.edu (Tyler McCormick)

1. Introduction

Network data describing the temporal evolution of interactions among a population of interest occur in a diverse range of applications, and particular examples of this data type include social ties among friend groups [41], academic coauthorship [76] and co-evolution of financial indices [21]. The latent space network approach of [32], where low-dimensional latent coordinates associated with the nodes of a network express the propensity for ties to form, is a popular approach for modelling such data (see for example [60, 66, 21]). In this article, we focus our attention on exact estimation of a dynamic latent space network (DLSN) model, in which the latent states evolve over time according to a Markov process, via sequential Monte Carlo (SMC) methods. Our aim is to be able to dynamically estimate latent state positions efficiently, without relying on approximations. This is challenging in this context due to the dimensionality of the state space and the dependence inherent to networked data. This approach is appealing in this context since we expect SMC to be more scalable than existing techniques when the number of observations in time is large, SMC facilitates both online and offline inference and SMC estimates can be conveniently updated given additional observations.

Latent space network models for temporal data are well-established in the literature, and both discrete time (see [60, 66, 25]) and continuous time (see [21, 57]) models have been proposed. It is typical in this approach to assume that a tie is more likely to form between two nodes whose latent coordinates are close in terms of Euclidean distance. This assumption allows a practitioner to determine an intuitive visualisation of the data through the latent positions and encourage transitive relationships to occur. Posterior samples for parameters of a DLSN model are typically obtained via MCMC [66, 21] which allows for asymptotically exact inference. However, standard algorithms become increasingly computationally costly as the number of nodes in the network or the number of observations in time grow. To improve scalability, several authors have proposed likelihood approximations [56, 58] whereas others rely on variational inference which targets a computationally cheaper approximation of the posterior (see [68, 45]).

In this article, we present a SMC scheme for joint state and parameter estimation of a DLSN model in which the latent trajectories for each node are assumed stationary. Our approach combines the methodology for high-dimensional state spaces presented in [51] with the gradient-based parameter

estimation presented in [49]. Our SMC scheme allows for both online and offline inference, and we explore the performance of our approach on simulated and real data examples. We present two simulation studies, the first of which examines the scalability of our approach as the number of nodes or the number of observations in time grows and the second of which explores the performance of our methodology under model misspecification. Additionally, we consider the performance of our approach on a real-world dataset describing classroom contacts in comparison and demonstrate how it is straightforward to consider model adaptations within our framework.

Our proposed methodology is closely related to the surrounding literature, but differs in two key ways. Firstly, our proposed DLSN model builds upon the SSM formulation of [60, 66] but instead assumes stationarity in the latent trajectories so that properties of the network interactions are preserved through time. This is advantageous in data settings when the properties of the observed networks do not change significantly between time points. Secondly, whilst SMC methodology has previously been used in the context of latent variable network data [61, 75], we present an inference procedure which does not rely on particular model forms or approximations, such as in [61, 75], by appealing to methodology proposed for high-dimensional state spaces. As a consequence of this, it is straightforward to apply our approach on modifications of our proposal model, including a model variant in which the stationarity assumption on the latent coordinates is relaxed. Finally, we note that our work differs from [9] who focus on the application for SMC to a sequence of growing graphs.

The remainder of this paper is organised as follows. In Section 2, we introduce our proposed model and comment further on its context within the broader literature. In Section 3, we review SMC methodology and detail our inference procedure. We present two simulation studies in Section 4 and examine a real world data example describing classroom contacts in Section 5. Finally, we conclude with a discussion in Section 6.

2. Dynamic Latent Space Network Modelling

In this section, we present a latent space model for dynamic networks in which the latent trajectories are assumed to follow a stationary first order Markov process and highlight the flexibility of this model by describing a number of possible variants. When coupled with the assumption that the latent coordinates follow a Gaussian random walk in the generative model,

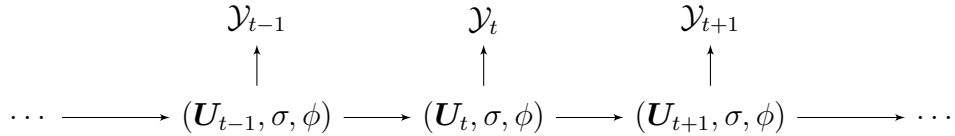


Figure 1: SSM for DLSNs. The observed adjacency matrices are modelled independently conditional on the latent coordinates, and the latent coordinates are modelled with a first order Markov process that is governed by static parameters (σ, ϕ) .

stationarity implies that the variance of the latent coordinate random variables does not accumulate over time. This allows network properties to be preserved over time and prevents degeneracy in which networks sampled from the model are empty given a sufficiently large time horizon.

Our proposed model is most similar to the work of [60] and [66]. However, in contrast to these works, we impose stationarity on the latent trajectories. This relates to the work of [31] who consider a latent factor model with stationarity, but differs since our formulation is based on latent Euclidean distances. A myriad of extensions have also been discussed in the literature such as the inclusion of community structures [66], weighted interactions [67], ranked interactions [65], multiple interaction types [22] and bipartite networks [25], and we note that our model may also be extended analogously as appropriate. For a detailed review of dynamic latent space network modelling we refer to [40].

2.1. Proposed Model and Identifiability

Suppose that we observe interactions among a population of size N at times $t = 1, 2, \dots, T$. We represent this information as a collection of time-indexed adjacency matrices $\mathcal{Y}_t = \{y_{ijt}\}_{i,j=1,2,\dots,N}$, where y_{ijt} indicates the connection between nodes i and j at time t . We will consider binary connections, so that $y_{ijt} \in \{0, 1\}$ where $y_{ijt} = 1$ indicates the presence of the $(i, j)^{th}$ edge. Additionally, we will assume that the connections are symmetric, so that $y_{ijt} = y_{jit}$, and that there are no self ties, so that $y_{iit} = 0$.

In the latent space framework, the probability of a connection forming is expressed as a function of d -dimensional coordinates associated with the nodes. We let $\mathbf{U}_t \in \mathbb{R}^{N \times d}$ represent the $N \times d$ matrix of latent coordinates at time t where the i^{th} row of \mathbf{U}_t corresponds to the latent coordinate of the i^{th} node at time t , $u_{it} \in \mathbb{R}^d$. Following [60, 66], we model the observations $\{\mathcal{Y}_t\}_{t=1}^T$ using a state space model (SSM) (see Figure 1), where we assume that

the interactions at time t occur independently conditional on the latent states $\{\mathbf{U}_t\}_{t=0}^T$ which follow a first order Markov process. We further assume that the latent trajectories for each node follow independent stationary Gaussian random walks so that

$$p(\mathbf{U}_0|\sigma, \phi) = \prod_{i=1}^N \mathcal{N}_d\left(0, \frac{\sigma^2}{1-\phi^2} I_d\right), \quad (1)$$

$$p(\mathbf{U}_t|\mathbf{U}_{t-1}, \sigma, \phi) = \prod_{i=1}^N \mathcal{N}_d(\phi u_{i,t-1}, \sigma^2 I_d), \quad \text{for } t = 1, 2, \dots, T, \quad (2)$$

where $\sigma \in \mathbb{R}_{>0}$, and $|\phi| < 1$.

Given the latent trajectories, we can now express the probability of observing connections \mathcal{Y}_t . Following [32], we assume that, conditional on \mathbf{U}_t , each edge can be modelled independently via logistic regression such that nodes whose latent coordinates are close in terms of Euclidean distance are more likely to be connected. More specifically, for $t = 1, 2, \dots, T$, we let

$$p(\mathcal{Y}_t|\mathbf{U}_t, \alpha) = \prod_{i<j} p_{ijt}^{y_{ijt}} (1 - p_{ijt})^{1-y_{ijt}}, \quad (3)$$

$$p_{ijt} = \frac{1}{1 + e^{-\eta_{ijt}}}, \quad (4)$$

$$\eta_{ijt} = \alpha - \|u_{it} - u_{jt}\|, \quad (5)$$

where $\alpha \in \mathbb{R}$ controls the global tendency for connections to form and $\|\cdot\|$ denotes the ℓ^2 -norm.

Specifying the connection probabilities as a function of the Euclidean distance has two primary advantages. Firstly, this implies that the coordinates \mathbf{U}_t provide an intuitive visualisation of the network at time t and, secondly, this encourages transitive relationships which are often observed in social networks. Additionally, there exist many possible variants of this model. For example, we may mimic the projection model of [32] by expressing (5) as a function of $u_i^T u_j$ or we may adapt (3) and (4) to express non-binary connections. These adaptations will be illustrated in Section 5.

Since (3) only depends on \mathbf{U}_t through the pairwise distances, $p(\mathcal{Y}_t|\mathbf{U}_t, \sigma, \phi)$ is invariant to distance-preserving transformations of \mathbf{U}_t . This is a key issue associated with latent space network models and, to resolve this, many authors rely on a Procrustes transformation (for example see [32], [66]) which finds the coordinates $\hat{\mathbf{U}}$ that minimise the sum of squared differences between

\mathbf{U} and some pre-specified reference coordinates \mathbf{U}_0 . Following [21], we opt to examine the model in terms of the connection probabilities which do not suffer from non-identifiability.

3. Model Inference

In this section we present details of our estimation procedure for the DSLN detailed in Section 2.1. The state space model formulation presents a natural setting for the application of SMC methodology and this is appealing for three key reasons. Firstly, this class of algorithms is based on recursive estimation and so SMC is more scalable than MCMC methods when the number of time steps T is large. Secondly, in contrast to variational Bayes techniques, SMC does not require model approximations. Finally, SMC methods facilitate both online and offline inference and allow posterior samples to be updated given additional observations.

A brief review of SMC methodology is presented in Appendix A for the unfamiliar reader. We obtain our inference procedure by combining the Guided Intermediate Resampling Filter of [51] with the gradient ascent approach of [49] for static parameter estimation. Details of our state and parameter estimation procedure are presented in Sections 3.1 and 3.2, respectively. We note here that, since the dimension of the state spaces for our model of interest is equal to $N \times d$, we must consider methodology for high-dimensional state space even for moderate N . Furthermore, as commented in Appendix A.4, the dependence present in network data restricts the set of high-dimensional SMC methodology that is appropriate for our model.

3.1. State Estimation

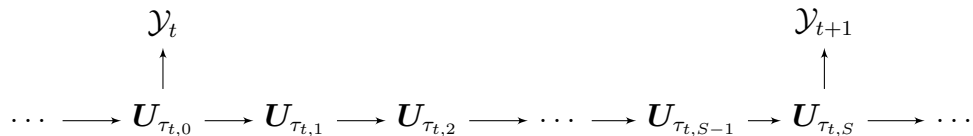


Figure 2: Intermediary steps $\{\mathbf{U}_{\tau_{t,s}}\}_{s=0}^S$ between observations \mathcal{Y}_t and \mathcal{Y}_{t+1} , where $\mathbf{U}_{\tau_{t,0}} = \mathbf{U}_t$ and $\mathbf{U}_{\tau_{t,S}} = \mathbf{U}_{t+1}$.

To estimate the latent coordinates we rely on the Guided Intermediate Resampling Filter (GIRF) of [51]. This scheme introduces artificial intermediary states in order to guide particles to regions of the state space with high

probability via a guide function, typically taken as the predictive likelihood. We opt to use this scheme since it is appropriate for our modelling assumptions, is straightforward to implement and can be adapted to model variants, such as those described in Section 2. It is important to stress that the dependence in our likelihood (3) makes it challenging to consider many alternative high-dimensional particle filtering schemes proposed in the literature, such as [59, 7].

Our goal is to estimate $\{\mathbf{U}_t\}_{t=1}^T$ given the observed connections $\{\mathcal{Y}_t\}_{t=1}^T$. For each pair of observations $\{\mathcal{Y}_t, \mathcal{Y}_{t+1}\}_{t=1}^{T-1}$, the GIRF introduces $S - 1$ intermediary time steps which we denote by $\{\tau_{t,s}\}_{s=0}^S$. Then, we rely upon the following decomposition for the latent transitions.

$$p_\theta(\mathbf{U}_{t+1}|\mathbf{U}_t) = \tilde{p}_\theta(\mathbf{U}_{\tau_{t,1}}|\mathbf{U}_t)\tilde{p}_\theta(\mathbf{U}_{\tau_{t,2}}|\mathbf{U}_{\tau_{t,1}})\cdots\tilde{p}_\theta(\mathbf{U}_{t+1}|\mathbf{U}_{\tau_{t,S-1}}), \quad (6)$$

where θ denotes model parameters and $\{\tau_{t,s}\}_{s=0}^S$ satisfy $\tau_{t,0} := t < \tau_{t,1} < \cdots < \tau_{t,S-1} < \tau_{t,S} := t + 1$, and $\tau_{t,0}$ and $\tau_{t,S}$ correspond to the times t and $t + 1$, respectively. Figure 2 provides a depiction of the intermediary states and a derivation of $\tilde{p}_\theta(\mathbf{U}_s|\mathbf{U}_{\tau_{t,s-1}})$ for the transition equation in (2) is given in Appendix B. Note that, for this intermediary transition to be valid, we require $\phi \in (0, 1)$ but this does not affect the expressivity of our model since connections are modelled in terms of the latent distances.

At each intermediary time step, the particles are weighted according to an assessment function $\nu_{\tau_{t,s}}(u)$ which guides particles towards future observations. This function must be specified so that $\nu_{\tau_{t,0}}(u) = 1$ and $\nu_{\tau_{t,S}}(u) = p(\mathcal{Y}_T|u)$, and we will discuss particular choices below. Given this function, particles at time step $\tau_{t,s}$ are then weighted according to

$$\omega_{\tau_{t,s}}(\mathbf{U}_{\tau_{t,s}}, \tilde{\mathbf{U}}_{\tau_{t,s-1}}) = \begin{cases} \frac{\nu_{\tau_{t,s}}(\mathbf{U}_{\tau_{t,s}})}{\nu_{\tau_{t,s-1}}(\tilde{\mathbf{U}}_{\tau_{t,s-1}})} & \text{if } \tau_{t,s-1} \notin 1:T \\ \frac{\nu_{\tau_{t,s}}(\mathbf{U}_{\tau_{t,s}})}{\nu_{\tau_{t,s-1}}(\tilde{\mathbf{U}}_{\tau_{t,s-1}})} p(\mathcal{Y}_t|\tilde{\mathbf{U}}_{\tau_{t,s-1}}) & \text{if } \tau_{t,s-1} \in 1:T \end{cases}. \quad (7)$$

Using (7), we see that the likelihood $\ell(\mathcal{Y}_{1:T}) = \mathbb{E} \left[\prod_{t=1}^T p(\mathcal{Y}_t|\mathbf{U}_t) \right]$ can be approximated by $\hat{\ell} = \prod_{t=0}^T \prod_{s=1}^S \frac{1}{M} \sum_{i=1}^M \omega_{\tau_{t,s}}(\mathbf{U}_{\tau_{t,s}}^{(i)}, \tilde{\mathbf{U}}_{\tau_{t,s-1}}^{(i)})$.

Algorithm 1 details the GIRF for the DLSN model in Section 2.1 where $L = \log \hat{\ell}$. To implement this procedure we must specify $\nu_{\tau_{t,s}}(x)$ and the number of intermediary states S . [51] suggest choosing $\nu_{\tau_{t,s}}(u)$ so that particles are guided towards B future observations and then demonstrate that

Algorithm 1 GIRF for DLSN with θ assumed known

Initialise: $L = 0$, sample $\tilde{\mathbf{U}}_{\tau_{0,0}}^{(i)} \sim p(\cdot|\theta)$ for $i \in 1 : M$
For $t = 0 : T - 1$
 For $s \in 1 : S$
 $\mathbf{U}_{\tau_{t,s}}^{(i)} \sim p(\cdot|\tilde{\mathbf{U}}_{\tau_{t,s-1}}^{(i)}, \theta)$ for $i \in 1 : M$
 $w_{\tau_{t,s}}^{(i)} = \omega_{\tau_{t,s}}(\mathbf{U}_{\tau_{t,s}}^{(i)}, \tilde{\mathbf{U}}_{\tau_{t,s-1}}^{(i)})$ for $i \in 1 : M$
 $L = L + \log(\sum_j w_{\tau_{t,s}}^{(j)})$
 Sample $b^{(i)}$ such that $\mathbb{P}(b^{(i)} = j) \propto w_{\tau_{t,s}}^{(j)}$ for $i \in 1 : M$
 Set $\tilde{\mathbf{U}}_{\tau_{t,s}}^{(i)} = \mathbf{U}_{\tau_{t,s}}^{(b^{(i)})}$
 End
End

S should scale linearly according to the dimension of the latent states. In our setting, a straightforward option is to take $\nu_{\tau_{t,s}}(\mathbf{U}) = p(\mathcal{Y}_{t+1}|\mathbf{U})$ or, to incorporate B future observations,

$$\nu_{\tau_{t,s}}(\mathbf{U}) = \prod_{b=1}^{\min\{B, T-t\}} (\nu_{\tau_{t,s}, \tau_{t+b}}(\mathbf{U}))^{\eta_{\tau_{t,s}, \tau_{t+b}}}, \quad (8)$$

where $\nu_{\tau_{t,s}, \tau_{t+b}}(\mathbf{U})$ approximates $p_{\mathcal{Y}_{t+b}|\mathbf{U}_{\tau_{t,s}}}(\mathcal{Y}_{t+b}|\mathbf{U})$ and $\eta_{\tau_{t,s}, \tau_{t+b}}$ controls the contribution of $\nu_{\tau_{t,s}, \tau_{t+b}}(\mathbf{U})$ in the assessment function. We let

$$\eta_{\tau_{t,s}, \tau_{t+b}} = 1 - \frac{(bS - s)}{S[(t+b) - \max(t+b-B, 0)]}, \quad (9)$$

so that the contribution of observations decreases as a function of distance from $\tau_{t,s}$. This ensures that the potentially less accurate approximations have a smaller contribution to $\nu_{\tau_{t,s}}(\mathbf{U})$. It may be possible to approximate $p_{\mathcal{Y}_{t+b}|\mathbf{U}_{\tau_{t,s}}}(\mathcal{Y}_{t+b}|\mathbf{U})$ via simulation, however this will be computationally expensive for our setting. Instead we take $\nu_{\tau_{t,s}, \tau_{t+b}}(\mathbf{U}) = g(\mathcal{Y}_{t+b}|m_{\tau_{t,s}}(\mathbf{U}))$, where $m_{\tau_{t,s}}(\mathbf{U}) = \mathbb{E}[\mathbf{U}_{t+b}|\mathbf{U}_{\tau_{t,s}} = \mathbf{U}] = \phi\mathbf{U}$, which can be conveniently calculated.

Appendix D provides an example of the GIRF on simulated data with known θ . From this, we find that the choice of S significantly impacts the performance of the filter, with larger values improving the estimated effective sample size (ESS) given by $\widehat{ESS} = 1/\sum_{i=1}^M (w_t^{(i)})^2$ at time t . We also found little difference in practice between choosing $p(\mathcal{Y}_{t+1}|\mathbf{U})$ versus (8) as the assessment function for this example.

3.2. Parameter Estimation

We now consider estimation of static parameters $\theta = (\sigma, \phi, \alpha)$ using the gradient ascent approach of [49]. This allows both online and offline estimation, is flexible to variants of the model and has been shown to exhibit favourable performance when compared with other well known algorithms. We begin by discussing their approach for offline estimation of θ .

A gradient ascent procedure obtains the k^{th} estimate of θ by taking

$$\theta_k = \theta_{k-1} + \gamma_k \nabla \log p(\mathcal{Y}_{1:T} | \theta) |_{\theta=\theta_{k-1}}, \quad (10)$$

where γ_k is a sequence of decreasing steps such that $\sum_k \gamma_k = \infty$ and $\sum_k \gamma_k^2 < \infty$. A typical choice for this sequence is $\gamma_k = k^{-\alpha}$ with $0.5 < \alpha < 1$.

Following [49], we can rely on a particle approximation of the score $\zeta_T = \nabla \log p(\mathcal{Y}_{1:T} | \theta) |_{\theta=\theta_{k-1}}$ to evaluate (10). Details of this approximation are given in Algorithm 2, where $\{m_t^{(i)}, w_t^{(i)}\}_{i=1}^M$ denotes the particle approximation of $\zeta_t = \nabla \log p(\mathcal{Y}_{1:t} | \theta)$. An implementation of Algorithm 2 returns an estimate of ζ_T which can then be used to update θ according to (10). An offline procedure for joint estimation of θ and the latent states therefore requires N_θ runs of the PF, and an outline of our implementation using the GIRF is presented in Algorithm 3.

An online implementation can be obtained by assuming

$$\nabla \log p(\mathcal{Y}_t | \mathcal{Y}_{1:t-1}, \theta_t) \approx \nabla \log p(\mathcal{Y}_{1:t} | \theta_t) - \nabla \log p(\mathcal{Y}_{1:t-1} | \theta_{t-1}), \quad (11)$$

which typically holds for small changes between θ_{t-1} and θ_t . Then, at the t^{th} iteration, we approximate the score ζ_T by $\nabla \log \hat{p}(\mathcal{Y}_t | \mathcal{Y}_{1:t-1}, \theta_t) = \zeta_t - \zeta_{t-1}$, where ζ_t is obtained via Algorithm 2. The details of this online procedure for joint estimation of θ and the latent states are given in Algorithm 4.

Since we require $\sigma > 0$ and $\phi \in (0, 1)$ we opt to estimate $\tilde{\theta} = (\tilde{\sigma}, \tilde{\phi}, \alpha)$ where

$$\tilde{\sigma} = \log(\sigma) \in \mathbb{R} \quad \text{and} \quad \tilde{\phi} = \log\left(\frac{\phi}{1-\phi}\right) \in \mathbb{R}. \quad (12)$$

Details of the necessary gradient calculations and parameter initialisations are provided in Appendix C and Appendix E, respectively.

4. Performance for Simulated Examples

We now explore the performance of the GIRF when additionally estimating static parameters, as detailed in Section 3. First, we will consider the

Algorithm 2 Estimate the score ζ_t via Rao-Blackwellisation

Initialise: set $m_0^{(i)} = 0$ for $i = 1, 2, \dots, M$, $\zeta_0 = 0$.

For $t = 1, 2, \dots, T$

1) Run one iteration of a PF to obtain $\{\mathbf{U}_t^{(i)}\}_{i=1}^M$, $\{a_{t-1}^{(i)}\}_{i=1}^M$ and $\{w_t^{(i)}\}_{i=1}^M$ (see Algorithm 1 or 5)

2) Update the mean approximation

$$m_t^{(i)} = \lambda m_{t-1}^{(a_{t-1}^{(i)})} + (1 - \lambda) \zeta_{t-1} + \nabla \log p(\mathcal{Y}_t | \mathbf{U}_t^{(i)}, \alpha) + \nabla \log p(\mathbf{U}_t^{(i)} | \mathbf{U}_{t-1}^{(a_{t-1}^{(i)})}, \theta)$$

3) Update the score vector $\zeta_t = \sum_{i=1}^M w_t^{(i)} m_t^{(i)}$

End

Algorithm 3 Offline joint estimation of θ and $\{\mathbf{U}_t\}_{t=1}^T$ using the GIRF

Initialise: set θ_0 and number of iterations N_θ

For $l \in 1 : N_\theta$

1) Run the GIRF (Algorithm 1) with Rao-Blackwellisation (Algorithm 2) to estimate $\zeta_T = \nabla \log p(\mathcal{Y}_{1:T} | \theta) |_{\theta=\theta_{l-1}}$

2) Update theta according to $\theta_l = \theta_{l-1} + \gamma_l \zeta_T$

End

performance of this approach when the data are simulated according to different mechanisms in Section 4.1. Then, we will consider the scalability of this approach as N and T increase in Section 4.2.

4.1. Simulated Data

In this section our focus will be on the performance of the GIRF for networks simulated according to the following data generating mechanisms. Throughout, we fix $N = 30$, $d = 2$ and $T = 25$.

(S1) Simulate from the model in Section 2.1 with $\alpha = 0.75$, $\sigma = 0.4$ and $\phi = 0.9$.

(S2) Partition the nodes into two groups with latent centres $\mu_c \in \mathbb{R}^d$, for $c \in \{1, 2\}$. Suppose that the i^{th} node belongs to group c , then we let

$$u_{it}^c = (1 - q)u_{i,t-1}^c + q\mu_c + \epsilon, \quad (13)$$

Algorithm 4 Online joint estimation of θ and $\{\mathbf{U}_t\}_{t=1}^T$ using the GIRF

Initialise: set $\zeta_0 = 0, \theta_0, m_0^{(i)} = 0$ and $\tilde{\mathbf{U}}_{\tau_0,0}^{(i)} \sim p(\cdot|\theta)$ for $i = 1, 2, \dots, M$.

For $t = 1, 2, \dots, T$

1) Run S intermediary steps of the GIRF to obtain $\{\mathbf{U}_t^{(i)}\}_{i=1}^M$,
 $\{a_{t-1}^{(i)}\}_{i=1}^M$ and $\{w_t^{(i)}\}_{i=1}^M$ (innermost loop in Algorithm 1)

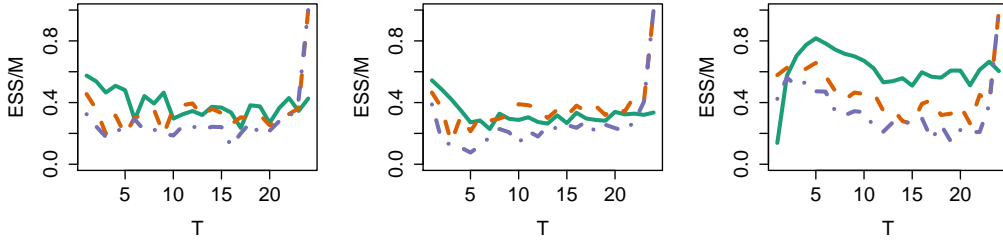
2) Update the mean approximation

$$m_t^{(i)} = \lambda m_{t-1}^{(i)} + (1 - \lambda) \zeta_{t-1} + \nabla \log p_{\theta_{t-1}}(\mathcal{Y}_t | \mathbf{U}_t^{(i)}, \alpha) \\ + \nabla \log f_{\theta_{t-1}}(\mathbf{U}_t^{(i)} | \mathbf{U}_{t-1}^{(i)}, \theta)$$

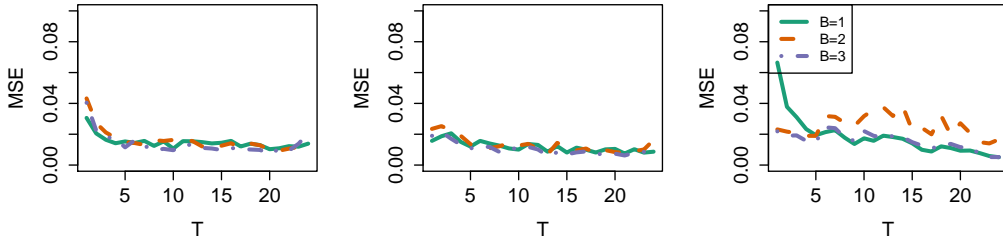
3) Update the score vector $\zeta_t = \sum_{i=1}^M w_t^{(i)} m_t^{(i)}$

4) Update theta according to $\theta_t = \theta_{t-1} + \gamma_k(\zeta_t - \zeta_{t-1})$

End

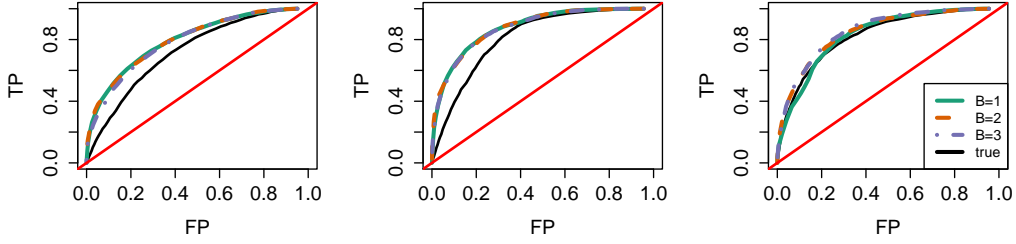


(a) Effective sample size.

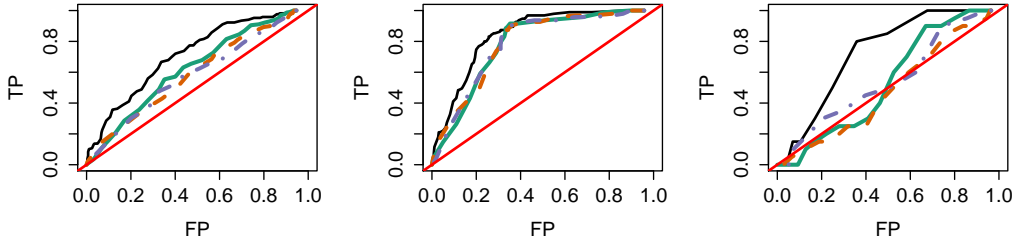


(b) Mean square error in probability.

Figure 3: Figure 3a shows the effective sample size and Figure 3b shows the mean square error in probability (see (15)) for the online estimation procedure run with $B = 1$ (green, solid), $B = 2$ (orange, dashed) and $B = 3$ (purple, dot-dashed) look ahead steps. For each figure, left, middle and right correspond to data simulated according to models (S1), (S2) and (S3), respectively.



(a) ROC curve for observations at time 1 to time $T - 1$.



(b) ROC curve for observations at time T .

Figure 4: Figures 4a and 4b show the ROC curves for observations 1 to $T - 1$ and for the predicted probabilities at time T , respectively. Each figure reports the ROC for the online estimation procedure run with $B = 1$ (green, solid), $B = 2$ (orange, dashed) and $B = 3$ (purple, dot-dashed) look ahead steps. For each figure, left, middle and right correspond to data simulated according to models (S1), (S2) and (S3), respectively. The line $y = x$ is shown in red and the ROC curve for the true probabilities is shown in black.

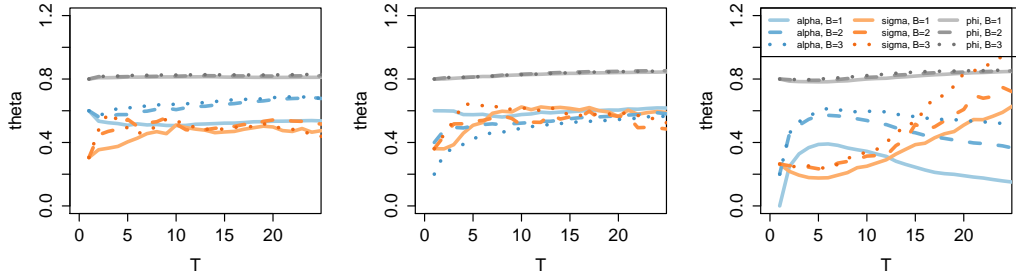


Figure 5: Estimates of α (blue), σ (orange) and ϕ (grey) obtained via online procedure with $B = 1$ (solid), $B = 2$ (dashed), $B = 3$ (dot-dashed). Left, middle and right figures correspond to the data generated according to (S1), (S2) and (S3), respectively.

where $q \in (0, 1)$ and $\epsilon \sim \mathcal{N}_d(0, \sigma^2 I_d)$. We take $\alpha = 0.75, \sigma = 0.4, q = 0.25, \mu_1 = (2, 0)$ and $\mu_2 = (-2, 0)$, and note that this is similar to the

model of [68].

(S3) Let the density α decrease over time, so that (5) is replaced by

$$\eta_{ijt} = \alpha_t - \|u_{it} - u_{jt}\|, \quad (14)$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_T)$ is a decreasing sequence starting at 2 and ending at -2 . We also take $\sigma = 0.4$ and $\phi = 0.9$.

(S1) corresponds to simulating from the generative model, (S2) corresponds to simulating from a variant of the generative model in which the nodes belong to one of two communities that evolve jointly, and (S3) corresponds to simulating from a variance of the generative model in which the density of the networks evolves from more to less connected. Note that (S1) allows us to verify our approach is behaving as expected, whereas (S2) and (S3) allow us to explore the performance of our approach under two specific forms of model misspecification.

For each of (S1), (S2) and (S3), we fit our model using both an online and offline procedure. We take $S = 1.5N$, $M = 5000$ and $d = 2$, and consider the two different choices of assessment function discussed in Section 3.1. For $B = 1$, we take $\nu_{\tau_{t,s}}(\mathbf{U}) = p(\mathcal{Y}_{t+1}|\mathbf{U})$, and for $B \in \{2, 3\}$ we take the assessment function given by (9). For the online procedure, the results are summarised in Figures 3, 4 and 5 and corresponding figures for the offline procedure can be found in Appendix F. To assess the performance Figure 3 depicts the ESS, mean square error in probability, given by

$$MSE_{prob}^{(t)} = \frac{1}{\binom{N}{2}} \sum_{i < j} (p_{ijt} - \hat{p}_{ijt})^2, \quad (15)$$

and Figure 4 shows ROC curves. Figure 5 shows the parameter estimates.

Overall, we see that increasing the number of look-ahead steps B does not offer much improvement in terms of the performance of the filter. This is likely due to the observations being binary, and setting $B > 1$ may offer improvement for more informative observations. The cases (S1) and (S2) exhibit the most stable performance throughout and this is particularly clear in Figure 5 where we see that the estimates of α and σ for case (S3) vary according to the decreasing density of the observed networks. We also see in Figure 3b that the MSE in probability is larger for case (S3), however this is somewhat obscured by the variability in the parameter estimates. In the

offline setting, we find that the parameter estimates are more stable which results in a larger MSE (see Appendix F). Online estimation is challenging in general, and so it is unsurprising that we find that the performance is generally much more stable in the offline setting. Finally, we comment that suitable modifications of our model may be made in order to express the structures present in (S2) and (S3) if appropriate.

4.2. Scalability

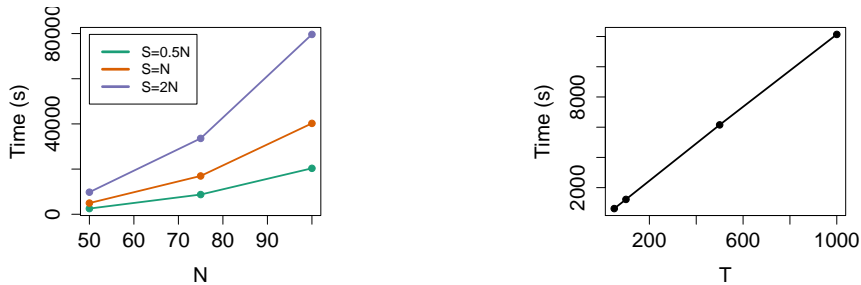
In this section we explore the scalability of our approach as the dimension of the state space increases in terms N and T . We simulate data according to the model in Section 2.1 under the following two settings.

1. **Increasing T :** Simulate a dataset with $T = 1000$ and $N = 20, d = 2, \alpha = 1.25, \sigma = 0.2, \phi = 0.9$, and restrict the number of observations to $t = 50, 100, 500$, and 1000 .
2. **Increasing N :** Simulate a dataset for each of $N \in \{50, 75, 100\}$ and $T = 25, d = 2, \alpha = 1, \sigma = 0.2, \phi = 0.9$.

For each case we implement online inference with $M = 5000$ and the timings are shown in Figure 6. We choose the number of intermediary states as $S = N$ for increasing T and $S = 0.5N, N$, and $2N$ for increasing N . From Figure 6a we see that the computational cost grows quadratically with the number of nodes N and this is due to the $\binom{N}{2}$ terms in the expression (3). Whilst we may reduce the overall computational cost through decreasing S , this will affect the performance of the filter (see Figure G.13a in Appendix G). From Figure 6b we see that there is a linear increase in the computational cost as T grows. The filter is much more stable in this setting (see Figure G.13b in Appendix G), and so is much more appropriate for network data with a large number of observations in time.

5. Classroom Contact Data

In this section we consider the application of our proposed methodology to a dataset describing face-to-face contact among primary school children. Students are equipped with sensors and a connection is recorded if two students face each other within a 20 second interval. The data are available from www.sociopatterns.org and have been published in [71] and [27]. We analyse interactions among a class of 25 school children on an aggregate level,



(a) Computational time as a function of N for $T = 25$. The number of intermediary states is set to $0.5N, N$ and $2N$.

(b) Time as a function of T for $N = 20$. The number of intermediary states is set to $S = N$.

Figure 6: Time taken for online SMC procedure for increasing N (left) with $S = 0.5N$ (green), $S = N$ (orange) and $S = 2N$ (purple), and increasing T (right).

where we record whether each pair of students interacted within a 4 minute interval. In Section 5.1 we analyse the data with binary interactions whereas in Section 5.2 we analyse the data with integer interactions. Furthermore, a discussion of the stationarity assumption for this data example is presented in Appendix H.

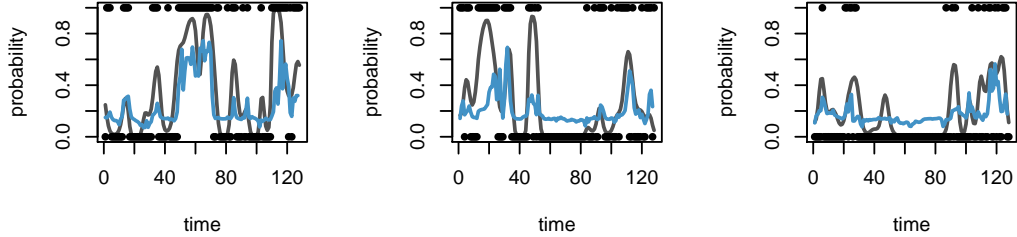
5.1. Binary Interactions

We now apply our model to the contact network where the connections are binary, so that $y_{ijt} = 1$ if students i and j interact within the t^{th} interval and $y_{ijt} = 0$ otherwise. We compare our model to the approach of [21] who model the latent trajectories according to a Gaussian process and, throughout this section, we report estimates obtained via the offline procedure. In addition to the formulation presented in Section 2.1, we also fit our model with the linear predictor (5) replaced by

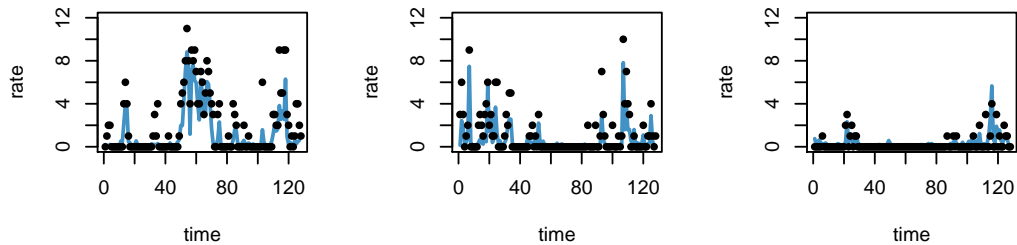
$$\eta_{ijt} = \alpha + u_{it}^T u_{jt}. \quad (16)$$

Similarly to the Euclidean distance formulation, u_{it} captures the probability of the i^{th} node interacting with the remaining nodes at time t relative to the base-rate parameter α . In contrast to the Euclidean distance assumption, the dot-product assumption imposes that nodes whose latent positions lie in similar directions from the origin are more likely to be connected.

Since [21] model the connection probabilities as a function of the dot-product between latent variables, the formulation (16) makes for a more reasonable comparison than the Euclidean distance formulation in presented



(a) Posterior mean connection probabilities for the dot-product DLSN model (blue) and the GP model of [21] (grey).



(b) Connection rates for the dot-product DLSN model.

Figure 7: Fitted model obtained via offline procedure for a selection of student pairs. Figure 7a shows the connection probabilities for binary data and Figure 7b shows the rate for count data. Left, middle and right correspond to the same node pairs for each plot.

in Section 2.1. We note that these models differ in how the latent trajectories and the base-rate parameter α are modelled. More specifically, [21] assume that the latent coordinates evolve according to a Gaussian process and allow α to vary with time.

We fit our model with $S = 2N$, $M = 5000$ and $d = 2$ and, similarly to [21], we assess the model on the connection probabilities to avoid identifiability issues associated with the latent representation. Figure 7a shows the estimated connection probabilities for the first $T - 1$ observations for a selection of node pairs, where the blue line represents the dot-product DLSN model fitted and the grey line represents the model of [21]. We see that there is a reasonable correspondence between the two models, however the model of [21] returns a smoother estimate of the probability trajectory. Our approach instead is able to obtain estimates at a lower computational cost, particularly when online inference is implemented. More precisely, the MCMC scheme of [21] involves update steps for each time point and node at every interaction, whereas our online GIRF scheme requires $S \times M$ likelihood evaluations at

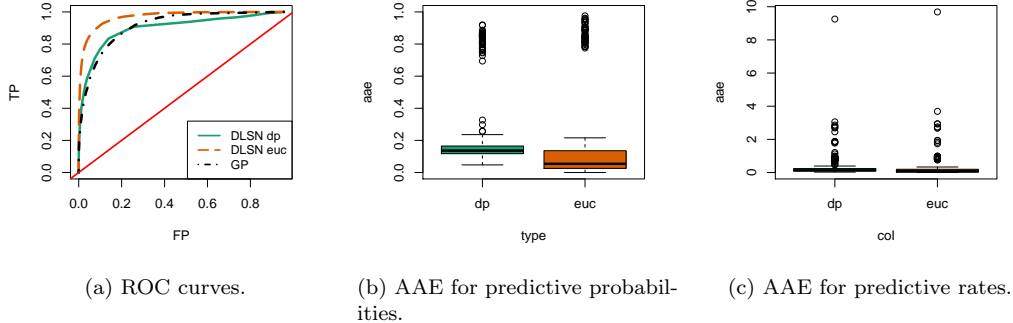


Figure 8: Summary of fit for the dot-product DLSN, Euclidean distance DLSN and the model of [21]. Figure 8a shows the ROC curve for each model. Figures 8b and 8c compare the average absolute error (AAE) (17) for our model with a dot-product and Euclidean distance formulations when the interactions are binary and weighted, respectively. We note the predictive measures are only considered for the DLSN models.

each time point.

Figure 8a reports the ROC curve for the DLSN model and the model of [21], and the dot-product and Euclidean distance formulations of the DLSN model have both been included for comparison. We see from this figure that all the models perform comparably well in terms of this measure, however we see that the Euclidean distance model is a marginally better classifier.

Finally, we explore the predictive distribution for the T^{th} observation. To assess the quality of predictions, we simulate $R_{\text{rep}} = 5000$ observations according to \hat{p}_{ijT} and record the average absolute error. For $\{i, j\} \in \{i, j \in \{1, 2, \dots, N\} | i < j\}$, this is given by

$$AAE_{ij} = \frac{1}{R_{\text{rep}}} \sum_{r=1}^{R_{\text{rep}}} |y_{ijT} - \hat{y}_{ijT}^r|, \quad (17)$$

where \hat{y}_{ijT}^r is simulated from a Bernoulli distribution with success probability \hat{p}_{ijT} . Note that (17) is equal to 0 when the data are predicted correctly. The distribution of the AAE over all node pairs is depicted in Figure 8b for the dot-product and Euclidean distance DLSN model, and from this we see that most observations are well predicted. Additionally, we observe that the Euclidean distance formulation performs better in terms of this measure.

5.2. Weighted Interactions

Our approach also allows us to straightforwardly analyse networks with non-binary interactions and, in this section, we consider the classroom data where y_{ijt} represents the number of interactions between nodes i and j in the t^{th} time interval. To adapt the model in Section 2.1, we model the probability of connections forming as

$$p(\mathcal{Y}_t | \mathbf{U}_t, \alpha) = \prod_{i < j} e^{-\lambda_{ijt}} \lambda_{ijt}^{y_{ijt}} / y_{ijt}!, \quad (18)$$

$$\lambda_{ijt} = \exp\{\alpha - \|u_{it} - u_{jt}\|\}, \quad (19)$$

with expressions for the latent transitions (1) and (2) specified as before.

The formulation (19) can also be adapted to express λ_{ijt} in terms of the dot-product $u_i^T u_j$, similar to the model considered in Section 5.1. Here, we consider both formulations and explore the model fit when estimates are obtained via the offline procedure. Throughout, we fix $S = 2N, M = 5000$ and $d = 2$. The estimated rates for a subset of node pairs for the dot-product formulation is shown in Figure 7b, where each panel corresponds to the same set of node pairs in Figure 7a. It is clear that modelling the data in this way allows us to obtain an understanding of the interactions on a finer-scale. Finally, we assess the predictive distribution for the T^{th} observations in Figure 8c. We see that the majority of observations are predicted well and that there is no clear difference between the dot-product and Euclidean distance models. Note that we have not compared to the approach of [21] for this example. Since their inference procedure relies on a data augmentation scheme for logistic models, it is not immediately straightforward to adapt their approach to the non-binary setting.

6. Discussion

In this article we have considered sequential estimation of a dynamic latent space network model and our approach relies on the algorithms introduced in [51] and [49]. Since standard SMC methodology scales poorly as the dimension of the state space increases, we consider the GIRF of [51] which allows us to estimate networks with moderate N . Our approach is most appropriate in settings where there are a large number of observations in time, allows convenient estimation of predictive distributions and, unlike MCMC, can be updated with additional observations without rerunning the entire

inference mechanism. We have also shown that, in contrast to related approaches, our procedure can be adapted to model variations, such as weighted interactions, and is suited to both online and offline inference. To the best of our knowledge, we have presented a novel approach for sequential estimation of dynamic latent space networks which, in contrast to existing methodology, does not rely on specific model forms or approximations. Since the likelihood of a latent space network model is of $O(N^2)$ computation complexity and a basic MCMC scheme requires $n_{it} \times T$ evaluations of this likelihood, we expect our SMC scheme to be more computationally efficient when $M \times S$ is much less than n_{it} .

In practical applications, we anticipate our proposed methodology to be particularly appealing in either data streaming settings or for data examples with a large number of observations in time. SMC is well-suited to these settings since the sequential nature of SMC allows estimates to be updated given additional observations, therefore making this approach amenable to online inference. In contrast, we stress that estimates obtained via either MCMC or variational methods would require reestimation of model parameters under further observations. Furthermore, since our inference procedure is simulation based, our methodology is not sensitive to modifications of the model. Whilst this is also the case for MCMC methods, we note that for some model modifications it may be difficult to derive an expression for the evidence lower bound, needed to update estimates in variational methods, and this would make application of such techniques challenging.

When implementing our methodology, there are a several choices which require the practitioners input. Firstly, they must specify the number of particles M and the number of intermediary steps S given their available computing resources. As shown in Section 4.2, increasing S will require more computational resources but, from Appendix D we see that a larger S does improve the performance of the GIRF. We advise taking M and S as large as possible and not choosing S less than $1.5N$. In our simulations and data examples we let $S = 1.5N$ and $S = 2N$, respectively, and found this to be sufficient. Secondly, depending on the specifics of the data example, a practitioner must choose between online and offline inference. Here we note that, as evidenced by the study in Section 4.1, parameter estimation is less stable for our online implementation and so care must be taken. Furthermore, we also note that the noise in the log-likelihood approximation utilised in our gradient descent algorithm may give rise to local modes. However, we did not find evidence of this in our examples with known static parameters.

A key limitation of our approach is the scalability in terms of the number of nodes N and this is due to both properties of SMC methodology and the $O(N^2)$ calculations needed to evaluate the likelihood. The latter contribution has been addressed via likelihood approximations (see [56] and [58]), however it is not straightforward to adapt this approach within the context of SMC. Alternatively, the modelling approach taken in [24], in which the nodes are partitioned into communities and the within-community connection probabilities are modelled via a latent space, may allow us to consider networks with larger N . In this setting, regions of independence in the latent space may allow us to develop a more scalable approach through a partition of the state space (for example, as in [59]).

This work may also be considered in the context of changepoint or anomaly detection, similarly to [42], where the authors rely on approximate inference via variational methods. However, in contrast to existing DLSN models, our approach models the latent nodes via a stationary process and so is likely more appropriate for determining changes in network behaviour. For example, the latent space approach may allow for the detection of changes in joint distributions of motif counts resulting from a change in the variance of the latent representation.

Acknowledgements

KT gratefully acknowledges the support of the EPSRC funded EP/L015692/1 STOR-i Centre for Doctoral Training and CN gratefully acknowledges the support of EPSRC grants EP/S00159X/1, EP/R01860X/1 and EP/V022636/1. We thank Daniele Durante for providing code for the Gaussian Process network model used in Section 5.1, and we thank Brendan Murphy and Marco Battiston for helpful comments on an earlier version of this work.

Code to implement our methodology can be found at <https://github.com/kturnbull/DLSN-SMC/> and the real data analysed in Section 5 is available at <http://www.sociopatterns.org/datasets/primary-school-temporal-network-data/>.

References

- [1] Ades, M. and van Leeuwen, P. J. (2015). The equivalent-weights particle filter in a high-dimensional system. *Quarterly Journal of the Royal Meteorological Society*, 141(687):484–503.

- [2] Akyildiz, Ö. D. and Míguez, J. (2019). Nudging the particle filter. *Statistics and Computing*.
- [3] Ala-Luhtala, J., Whiteley, N., Heine, K., and Piche, R. (2016). An introduction to twisted particle filters and parameter estimation in non-linear state-space models. *IEEE Transactions on Signal Processing*, 64(18):4875–4890.
- [4] Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.
- [5] Bengtsson, T., Bickel, P., Li, B., et al. (2008). Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and statistics: Essays in honor of David A. Freedman*, pages 316–334. Institute of Mathematical Statistics.
- [6] Beskos, A., Crisan, D., and Jasra, A. (2014a). On the stability of Sequential Monte Carlo methods in high dimensions. *The Annals of Applied Probability*, 24(4):1396–1445.
- [7] Beskos, A., Crisan, D., Jasra, A., Kamatani, K., and Zhou, Y. (2017). A stable particle filter for a class of high-dimensional state-space models. *Advances in Applied Probability*, 49(1):24–48.
- [8] Beskos, A., Crisan, D. O., Jasra, A., and Whiteley, N. (2014b). Error bounds and normalising constants for Sequential Monte Carlo samplers in high dimensions. *Advances in Applied Probability*, 46(1):279–306.
- [9] Bloem-Reddy, B. and Orbanz, P. (2018). Random-walk models of network formation and Sequential Monte Carlo methods for graphs. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(5):871–898.
- [10] Brockwell, A., Del Moral, P., and Doucet, A. (2012). Sequentially interacting Markov chain Monte Carlo methods. *ArXiv e-prints*.
- [11] Cappé, O. (2011). Online EM algorithm for hidden Markov models. *Journal of Computational and Graphical Statistics*, 20(3):728–749.
- [12] Carvalho, C. M., Johannes, M. S., Lopes, H. F., and Polson, N. G. (2010). Particle learning and smoothing. *Statistical Science*, 25(1):88–106.
- [13] Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). SMC2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426.
- [14] Crisan, D. and Míguez, J. (2018). Nested particle filters for online pa-

- parameter estimation in discrete-time state-space Markov models. *Bernoulli*, 24(4A):3039–3086.
- [15] Del Moral, P. (2004). *Feynman-Kac Formulae*, chapter Genealogical and Interacting Particle Systems With Applications. Springer-Verlag New York.
- [16] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.
- [17] Douc, R. and Cappé, O. (2005). Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69.
- [18] Doucet, A., De Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo methods in practice*. Statistics for Engineering and Information Science. Springer, New York, NY.
- [19] Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.
- [20] Doucet, A., Johansen, A. M., et al. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3.
- [21] Durante, D. and Dunson, D. B. (2014). Nonparametric Bayes dynamic modelling of relational data. *Biometrika*, 101(4):883–898.
- [22] Durante, D., Mukherjee, N., and Steorts, R. C. (2017). Bayesian learning of dynamic multilayer networks. *Journal of Machine Learning Research*, 18(1):1414–1442.
- [23] Fearnhead, P. (2002). Markov Chain Monte Carlo, sufficient statistics and particle filters. *Journal of Computational and Graphical Statistics*, 11(4):848–862.
- [24] Fosdick, B. K., McCormick, T. H., Murphy, T. B., Ng, T. L. J., and Westling, T. (2019). Multiresolution network models. *Journal of Computational and Graphical Statistics*, 28(1):185–196.
- [25] Friel, N., Rastelli, R., Wyse, J., and Raftery, A. E. (2016). Interlocking directorates in irish companies using a latent space model for bipartite networks. *Proceedings of the National Academy of Sciences*, 113(24):6629–6634.
- [26] Gao, M. and Zhang, H. (2012). Sequential Monte Carlo methods for

- parameter estimation in nonlinear state-space models. *Computers and Geosciences*, 44:70 – 77.
- [27] Gemmetto, V., Barrat, A., and Cattuto, C. (2014). Mitigation of infectious disease at school: targeted class closure vs school closure. In *BMC Infectious Diseases*.
- [28] Gilks, W. R. and Berzuini, C. (2001). Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146.
- [29] Godsill, S. and Clapp, T. (2001). Improvement strategies for Monte Carlo particle filters. In *Sequential Monte Carlo methods in practice*, pages 139–158. Springer.
- [30] Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings F, Radar and Signal Processing*, 140(2):107–113.
- [31] He, Y. and Hoff, P. D. (2019). Multiplicative coevolution regression models for longitudinal networks and nodal attributes. *Social Networks*, 57:54 – 62.
- [32] Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.
- [33] Hürzeler, M. and Künsch, H. R. (2001). *Approximating and Maximising the Likelihood for a General State-Space Model*, pages 159–175. Springer New York, New York, NY.
- [34] Ionides, E. L., Bhadra, A., Atchadé, Y., and King, A. (2011). Iterated filtering. *The Annals of Statistics*, 39(3):1776–1802.
- [35] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45.
- [36] Kantas, N., Doucet, A., Singh, S., and Maciejowski, J. (2009). An overview of Sequential Monte Carlo methods for parameter estimation in general state-space models. *IFAC Proceedings Volumes*, 42(10):774 – 785. 15th IFAC Symposium on System Identification.
- [37] Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., and Chopin, N. (2015). On particle methods for parameter estimation in state-space models. *Statistical Science*, 30(3):328–351.
- [38] Katzfuss, M., Stroud, J. R., and Wikle, C. K. (2016). Understanding the ensemble Kalman filter. *The American Statistician*, 70(4):350–357.

- [39] Khan, Z., Balch, T., and Dellaert, F. (2005). Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819.
- [40] Kim, B., Lee, K. H., Xue, L., and Niu, X. (2018). A review of dynamic network models with latent variables. *Statistics Surveys*, 12:105–135.
- [41] Krivitsky, P. N. and Handcock, M. S. (2014). A separable model for dynamic networks. *Journal of the Royal Statistical Society Series B*, 76(1):29–46.
- [42] Lee, W., McCormick, T. H., Neil, J., and Sodja, C. (2019). Anomaly detection in large scale networks with latent space models.
- [43] LeGland, F. and Mevel, L. (1997). Recursive estimation in hidden Markov models. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 4, pages 3468–3473 vol.4.
- [44] Liu, J. and West, M. (2001). *Combined Parameter and State Estimation in Simulation-Based Filtering*, pages 197–223. Springer New York, New York, NY.
- [45] Liu, Y. and Chen, Y. (2023). Variational inference for latent space models for dynamic networks. *Statistica Sinica*.
- [46] Lopes, H. F. and Tsay, R. S. (2011). Particle filters and Bayesian inference in financial econometrics. *Journal of Forecasting*, 30(1):168–209.
- [47] Malik, S. and Pitt, M. K. (2011). Particle filters for continuous likelihood evaluation and maximisation. *Journal of Econometrics*, 165(2):190 – 209.
- [48] Naesseth, C. A., Lindsten, F., and Schön, T. B. (2019). High-dimensional filtering using nested Sequential Monte Carlo. *IEEE Transactions on Signal Processing*, 67(16):4177–4188.
- [49] Nemeth, C., Fearnhead, P., and Mihaylova, L. (2016). Particle approximations of the score and observed information matrix for parameter estimation in state–space models with linear computational cost. *Journal of Computational and Graphical Statistics*, 25(4):1138–1157.
- [50] Nguyen, T. L. T., Septier, F., Peters, G. W., and Delignon, Y. (2016). Efficient Sequential Monte-Carlo samplers for Bayesian inference. *IEEE Transactions on Signal Processing*, 64(5):1305–1319.
- [51] Park, J. and Ionides, E. L. (2020). Inference on high-dimensional implicit dynamic models using a guided intermediate resampling filter. *Statistics and Computing*, 30(5):1497–1522.
- [52] Petersen, K. B. and Pedersen, M. S. (2012). The matrix cookbook. Version 20121115.

- [53] Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599.
- [54] Polson, N. G., Stroud, J. R., and Müller, P. (2008). Practical filtering with sequential parameter learning. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(2):413–428.
- [55] Poyiadjis, G., Doucet, A., and Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80.
- [56] Raftery, A. E., Niu, X., Hoff, P. D., and Yeung, K. Y. (2012). Fast inference for the latent space network model using a case-control approximate likelihood. *Journal of Computational and Graphical Statistics*, 21(4):901–919.
- [57] Rastelli, R. and Corneli, M. (2021). Continuous latent position models for instantaneous interactions. *arXiv preprint arXiv:2103.17146*.
- [58] Rastelli, R., Maire, F., and Friel, N. (2018). Computationally efficient inference for latent position network models. *arXiv e-prints*, page arXiv:1804.02274.
- [59] Rebeschini, P. and van Handel, R. (2015). Can local particle filters beat the curse of dimensionality? *The Annals of Applied Probability*, 25(5):2809–2866.
- [60] Sarkar, P. and Moore, A. W. (2006). Dynamic social network analysis using latent space models. In Weiss, Y., Schölkopf, B., and Platt, J. C., editors, *Advances in Neural Information Processing Systems 18*, pages 1145–1152. MIT Press.
- [61] Sarkar, P., Siddiqi, S. M., and Gordon, G. J. (2007). A latent space approach to dynamic embedding of co-occurrence data. In Meila, M. and Shen, X., editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 420–427, San Juan, Puerto Rico. PMLR.
- [62] Septier, F., Pang, S. K., Carmi, A., and Godsill, S. (2009). On MCMC-based particle methods for Bayesian filtering: Application to multitarget tracking. In *2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 360–363.
- [63] Septier, F. and Peters, G. W. (2015). An Overview of Recent Advances in Monte-Carlo Methods for Bayesian Filtering in High-Dimensional Spaces. In Peters, G. W. and Matsui, T., editors, *Theoretical Aspects of*

- Spatial-Temporal Modeling*. SpringerBriefs - JSS Research Series in Statistics.
- [64] Septier, F. and Peters, G. W. (2016). Langevin and Hamiltonian based Sequential MCMC for efficient Bayesian filtering in high-dimensional spaces. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):312–327.
 - [65] Sewell, D. K. and Chen, Y. (2015a). Analysis of the formation of the structure of social networks by using latent space models for ranked dynamic networks. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 64(4):611–633.
 - [66] Sewell, D. K. and Chen, Y. (2015b). Latent space models for dynamic networks. *Journal of the American Statistical Association*, 110(512):1646–1657.
 - [67] Sewell, D. K. and Chen, Y. (2016). Latent space models for dynamic networks with weighted edges. *Social Networks*, 44:105 – 116.
 - [68] Sewell, D. K. and Chen, Y. (2017). Latent space approaches to community detection in dynamic networks. *Bayesian Analysis*, 12(2):351–377.
 - [69] Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640.
 - [70] Snyder, C., Bengtsson, T., and Morzfeld, M. (2015). Performance bounds for particle filters using the optimal proposal. *Monthly Weather Review*, 143(11):4750–4761.
 - [71] Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J.-F., Quaggiotto, M., Van den Broeck, W., Régis, C., Lina, B., and Vanhems, P. (2011). High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE*, 6(8):1–13.
 - [72] Storvik, G. (2002). Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing*, 50(2):281–289.
 - [73] van Leeuwen, P. J. (2010). Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*, 136(653):1991–1999.
 - [74] Wigren, A., Murray, L., and Lindsten, F. (2018). Improving the particle filter in high dimensions using conjugate artificial process noise. *IFAC-PapersOnLine*, 51(15):670 – 675. 18th IFAC Symposium on System Identification SYSID 2018.

- [75] Xu, K. S. and Hero, A. O. (2014). Dynamic stochastic blockmodels for time-evolving social networks. *IEEE Journal of Selected Topics in Signal Processing*, 8:552–562.
- [76] Yang, T., Chi, Y., Zhu, S., Gong, Y., and Jin, R. (2011). Detecting communities and their evolutions in dynamic social networks - a Bayesian approach. *Machine Learning*, 82(2):157–189.

Appendix A. Background: SMC review

SMC methods are a class of simulation-based algorithms designed to iteratively estimate a posterior distribution (see [20], [46] and [18]). We split our discussion into state estimation in Appendix A.2, where we focus on particle filtering, and static parameter estimation in Appendix A.3. To motivate the algorithm considered in Section 3.1, we briefly discuss high-dimensional SMC methodology Appendix A.4.

Appendix A.1. General SSM

A general SSM is governed by the following equations

$$\begin{aligned}
 X_0 &\sim \mu_\theta(x_0), \\
 X_t|X_{t-1} = x_{t-1} &\sim f_\theta(x_t|x_{t-1}), & \text{for } t = 1, 2, \dots, T, \\
 Z_t|X_t = x_t &\sim g_\theta(z_t|x_t), & \text{for } t = 1, 2, \dots, T,
 \end{aligned} \tag{A.1}$$

where $\{X_t\}_{t=1}^T$ are the latent states, $\{Z_t\}_{t=1}^T$ are the observations and θ is a vector of parameters. The distributions $\mu_\theta(\cdot)$ and $f_\theta(\cdot|\cdot)$ govern the latent states and $g_\theta(\cdot|\cdot)$ controls the observations. Note that the model in Section 2.1 is a special case of (A.1) with $\theta = (\alpha, \phi, \sigma)$, $Z_t = \mathcal{Y}_t$ and $X_t = \mathbf{U}_t$ and densities as specified in (1), (2) and (3).

Appendix A.2. Generic Particle Filter

A particle filtering (PF) scheme estimates the latent states $\{x_t\}_{t=0}^T$ given observations $\{z_t\}_{t=1}^T$ by sequentially targeting the filtering densities $\{p_\theta(x_t|z_{1:t})\}_{t=1}^T$. In general, we cannot obtain analytic expressions for the filtering densities and so a PF scheme instead relies on approximations obtained via importance sampling (IS). However we note that, in the special case when $f_\theta(x_t|x_{t-1})$ and $g_\theta(z_t|x_t)$ are Gaussian, we may determine $p_\theta(x_t|z_{1:t})$ analytically and this scheme is the well-known Kalman filter [35].

Algorithm 5 Generic Particle Filter to obtain estimates $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^M$ of $p_\theta(x_t|z_{1:t})$, for $t = 1, 2, \dots, T$.

- *Iteration* $t = 0$:

Sample M particles $\{x_0^{(i)}\}_{i=1}^M$ from $\mu_\theta(\cdot)$ and assign weights $w_0^{(i)} = 1/M$.

- *Iteration* $t = 1, \dots, T$:

Assume particles $\{x_{t-1}^{(i)}\}_{i=1}^M$ with weights $\{w_{t-1}^{(i)}\}_{i=1}^M$ that approximate $p_\theta(x_{t-1}|z_{1:t-1})$.

a) Sample parent indices $\{a_{t-1}^{(i)}\}_{i=1}^M$ from $\mathcal{F}(\cdot|W_{t-1})$, where $W_{t-1} = (w_{t-1}^{(1)}, w_{t-1}^{(2)}, \dots, w_{t-1}^{(M)})$.

b) Propagate the particles according to $x_t^{(i)} \sim q_\theta(\cdot|z_t, x_{t-1}^{a_{t-1}^{(i)}})$, for $i = 1, 2, \dots, M$, and set $x_{1:t}^{(i)} = (x_{1:t-1}^{a_{t-1}^{(i)}}, x_t^{(i)})$

c) Calculate the weights

$$w_t^{(i)} \propto p_\theta(x_{1:t}^{(i)}|z_{1:t}) / q_\theta(x_t^{(i)}|z_t, x_{1:t-1}^{a_{t-1}^{(i)}}) p_\theta(x_{1:t-1}^{a_{t-1}^{(i)}}|z_{1:t-1})$$

for $i = 1, 2, \dots, M$ and normalise.

A generic PF scheme for approximating $\{p_\theta(x_t|z_{1:t})\}_{t=1}^T$ is outlined in Algorithm 5. We let $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^M$ denote the particle approximation to $p_\theta(x_t|z_{1:t})$, where $x_t^{(i)}$ and $w_t^{(i)}$ represent the i^{th} particle and i^{th} weight, respectively, for $i = 1, 2, \dots, M$. This scheme updates the particles at time $t - 1$ according to a proposal distribution $q_\theta(x_t|x_{t-1}, z_t)$ and then adjusts the weights to account for the discrepancy between the t^{th} and $(t - 1)^{\text{th}}$ filtering densities. Algorithm 5 contains a resampling step where parent indices $a_{t-1}^{(i)}$ are resampled according to $\mathcal{F}(\cdot|\cdot)$. This step mitigates against particle degeneracy in which the weights concentrate onto a small number of particles causing the quality of the approximations to degrade. We refer to [17] and [20] for details of standard resampling schemes.

Many well-established particle filtering algorithms can be obtained as a special case of the scheme in Algorithm 5. For instance, we may obtain the standard SIR filter of [30] by taking $q_\theta(\cdot|x_{t-1}, z_t) = f_\theta(\cdot|x_{t-1})$ and $w_t = g_\theta(z_t|\cdot)$. This filter has the advantage of being simple to implement, however the particles are propagated according to the model dynamics and so do not

incorporate information about future observations. It has been shown that the proposal which obtains optimal performance in terms of the variance of the importance weights is [19] $q_\theta(x_t|z_t, x_{t-1}) = p_\theta(x_t|z_t, x_{t-1})$. However, for many practical applications, we cannot find a tractable expression for this proposal. Alternative approaches, such as the auxiliary particle filter (APF) of [53], instead aim to approximate $p(x_t|z_t, x_{t-1})$.

To assess the performance of a PF, we can approximate the effective sample size (ESS) at time t as $\widehat{ESS} = 1/\sum_{i=1}^M (w_t^{(i)})^2$. Additionally, a PF allows us to obtain an estimate of the marginal likelihood $p_\theta(y_{1:T})$ from the importance weights using

$$\hat{p}_\theta(y_{1:T}) = \prod_{t=1}^T \left[\frac{1}{M} \sum_{i=1}^M w_t^{(i)} \right]. \quad (\text{A.2})$$

Evaluating (A.2) does not incur an additional computational cost and this estimator has been shown to be unbiased (see Theorem 7.4.2 in [15]).

Appendix A.3. Static Parameter Estimation

The procedure in Algorithm 5 estimates the latent states $\{x_t\}_{t=1}^T$ conditional on known static parameters θ . However, in practice, we also wish to estimate static parameters θ , given by (σ, ϕ, α) in our application. The existing literature can be divided into offline and online approaches, and we will briefly discuss these below. For a more in depth discussion, we refer the reader to [36], [26], [37] and [46].

Offline estimation can be considered within the Bayesian or Frequentist paradigm. In the Bayesian setting, particle MCMC [4] methods allow joint state and parameter estimation by utilising an SMC approximation of the the likelihood which leaves the target distribution invariant. This avoids calculation of complex proposal distributions and, due to the crucial role of the estimation of the marginal likelihood, attention has been focused on developing PF schemes which improve the quality of this estimate (for example, see [3]). Alternatively, [13] develop an iterated procedure which relies on nested particle filters to estimate the state and parameters jointly in the Bayesian setting. From the Frequentist perspective, estimating θ is viewed as a likelihood maximisation problem (see [47] and [33]). This has been addressed via gradient ascent in which estimates of θ are updated according to the gradient of the log-likelihood (for example, see [34]). [55] consider estimating the score and information matrix, which can be used within gradient ascent,

and [49] extend this to develop a procedure which has a linear computational cost. Alternatively, θ can be estimated using an expectation maximisation procedure as in [16].

In the online setting, a natural approach for estimating θ would be to find a particle approximation to the joint density $p(x_t, \theta | z_{1:t})$, similar to the procedure for estimating $\{x_t\}_{t=1}^T$ outlined in Algorithm 5. However, since θ does not evolve in time, the particle set will degenerate under repeated resampling. Online estimation of θ therefore presents a challenging task and remains an open problem in the literature. Several Bayesian strategies have been proposed, such as [30], who include artificial dynamics to reduce the degeneracy, and [28], who rely on MCMC kernels to add diversity to the particle set. Other approaches include practical filtering (see [54]), kernel approximations (see [44]) and estimating θ using sufficient statistics (see [72], [23] and [12]). More recently, [14] introduced a procedure which relies on nested particle filters, similarly to the approach of [13]. In the Frequentist setting, θ can be estimated via likelihood maximisation. For example, see [43], [55] and [49], for methodology which relies on gradient ascent. Alternatively, [11] consider estimation via an expectation maximisation procedure.

Appendix A.4. High-Dimensional Particle Filtering

It is well-understood that the performance of SMC methods degrade as the dimension of the state space increases [5, 69, 6, 8] and it has been shown that the number of particles must increase exponentially with the state dimension to avoid particle degeneracy [70]. Developing methodology for high-dimensional state spaces remains an open problem in the literature, and in recent years a body of work has developed to address this issue.

A popular strategy involves the utilisation of MCMC techniques, such as in [29] where MCMC moves are introduced to mitigate against particle degeneracy. More broadly, MCMC methodology has appeared as part of sequential MCMC schemes [39, 62, 10] and this idea has been explored further in, for example, [50] and [64]. Other authors have considered approximate algorithms, such as the ensemble Kalman filter (see [38]) which approximates a Kalman filter through a sample, or ‘ensemble’, from the target distribution. Alternatively, [59] explore local particle filters such as the block particle filter in which the state space is partitioned into non-overlapping blocks. Finally, [2] generalise the nudging approach, where a subset of particles are artificially moved towards the observation (see also [73] and [1]).

Several algorithms have also been developed for particular classes of models. [7] introduce the space-time filter which assumes that the likelihood can be decomposed so that knowledge of the observations can be incrementally incorporated. [74] introduce artificial process noise to improve the performance of the particle filter in high-dimensions for a class of state space models. [48] consider a nested approach in which an additional particle filter is introduced to approximate the proposal distribution relied upon in the fully adapted filter of [53]. [51] introduce artificial intermediary states in order to handle high-dimensional state spaces. We note that in our application of interest the form of our network model restricts which of these algorithms are appropriate. Finally, for a selective review of high-dimensional particle filtering algorithms, we refer the reader to [63].

Appendix B. Derivation of $\tilde{p}_\theta(\mathbf{U}_{\tau_t,s}|\mathbf{U}_{\tau_t,s-1})$

In this appendix we derive the intermediary transition kernels required to implement the GIRF. Since all nodes follow the same distribution independently, we drop the notation i and derive the transition equation for a single node.

For u_t we have $U_t|U_{t-1} = u_{t-1} \sim \mathcal{N}(\phi u_{t-1}, \sigma^2 I_d)$. We require an expression for $\tilde{p}_\theta(u_{\tau_t,s}|u_{\tau_t,s-1})$ so that

$$p_\theta(u_{t+1}|u_t) = \tilde{p}_\theta(u_{\tau_{t,1}}|u_t)\tilde{p}_\theta(u_{\tau_{t,2}}|u_{\tau_{t,1}})\dots\tilde{p}_\theta(u_{t+1}|u_{\tau_{t,S-1}}). \quad (\text{B.1})$$

By properties of the Normal distribution, we know that $\tilde{p}_\theta(u_{\tau_t,s}|u_{\tau_t,s-1}) = \mathcal{N}(m u_{\tau_t,s-1}, v I_d)$. We determine the values of m and v below.

Determining m

By the law of total expectation we have

$$\mathbb{E}[U_t|U_{t-1}] = \mathbb{E}[U_{\tau_t,S}|U_{\tau_t,0}] = \phi u_{t-1}, \quad (\text{B.2})$$

$$= \mathbb{E}[\mathbb{E}[\dots\mathbb{E}[\mathbb{E}[U_{\tau_t,S}|U_{\tau_t,S-1}]|U_{\tau_t,S-2}]\dots|U_{\tau_t,1}]|U_{\tau_t,0}], \quad (\text{B.3})$$

$$= m^S u_{t-1}. \quad (\text{B.4})$$

$$\Rightarrow m = \phi^{1/S}. \quad (\text{B.5})$$

We have $\tilde{p}_\theta(u_{\tau_t,s}|u_{\tau_t,s-1}) = \mathcal{N}(\phi^{1/S} u_{\tau_t,s-1}, v I_d)$ and we note that this is valid for $\phi \in (0, 1)$.

Determining v

The law of total variance tells us

$$\text{Var} [U_{\tau_t, S} | U_{\tau_t, 0}] = \mathbb{E} [\text{Var} [U_{\tau_t, S} | U_{\tau_t, 1}] | U_{\tau_t, 0}] + \text{Var} [\mathbb{E} [U_{\tau_t, S} | U_{\tau_t, 1}] | U_{\tau_t, 0}]. \quad (\text{B.6})$$

For the expectation, we have

$$\mathbb{E} [U_{\tau_t, s} | U_{\tau_t, s'}] = \phi^{(s-s')/S} u_{\tau_t, s} \text{ for } s' < s. \quad (\text{B.7})$$

We then obtain a recursion for (B.6) by writing

$$\text{Var} [U_{\tau_t, S} | U_{\tau_t, s}] = \mathbb{E} [\text{Var} [U_{\tau_t, S} | U_{\tau_t, s+1}] | U_{\tau_t, s}] + \text{Var} [\mathbb{E} [U_{\tau_t, S} | U_{\tau_t, s+1}] | U_{\tau_t, s}], \quad (\text{B.8})$$

for $s = S - 2, S - 3, \dots, 0$.

Starting from $s = S - 2$, we have

$$\text{Var} [U_{\tau_t, S} | U_{\tau_t, S-2}] = \mathbb{E} [\text{Var} [U_{\tau_t, S} | U_{\tau_t, S-1}] | U_{\tau_t, S-2}] + \text{Var} [\mathbb{E} [U_{\tau_t, S} | U_{\tau_t, S-1}] | U_{\tau_t, S-2}], \quad (\text{B.9})$$

$$= \mathbb{E} [vI_d | U_{\tau_t, S-2}] + \text{Var} [\phi^{1/S} U_{\tau_t, S-1} | U_{\tau_t, S-2}], \quad (\text{B.10})$$

$$= vI_d + \phi^{2/S} vI_d. \quad (\text{B.11})$$

Then from $s = S - 3$ we have

$$\text{Var} [U_{\tau_t, S} | U_{\tau_t, S-3}] = \mathbb{E} [\text{Var} [U_{\tau_t, S} | U_{\tau_t, S-2}] | U_{\tau_t, S-3}] + \text{Var} [\mathbb{E} [U_{\tau_t, S} | U_{\tau_t, S-2}] | U_{\tau_t, S-3}], \quad (\text{B.12})$$

$$= \mathbb{E} [vI_d + \phi^{2/S} vI_d | U_{\tau_t, S-3}] + \text{Var} [\phi^{2/S} U_{\tau_t, S-2} | U_{\tau_t, S-3}], \quad (\text{B.13})$$

$$= vI_d + \phi^{2/S} vI_d + \phi^{4/S} vI_d. \quad (\text{B.14})$$

Applying this procedure iteratively, we obtain

$$\sigma^2 = \text{Var} [U_{\tau_t, S} | U_{\tau_t, 0}] = vI_d [1 + \phi^{2/S} + \phi^{4/S} + \dots + \phi^{2(S-1)/S}], \quad (\text{B.15})$$

$$= vI_d \sum_{r=0}^{S-1} (\phi^{2/S})^r = vI_d \left(\frac{1 - (\phi^{2/S})^S}{1 - \phi^{2/S}} \right) = vI_d \left(\frac{1 - \phi^2}{1 - \phi^{2/S}} \right), \quad (\text{B.16})$$

$$\Rightarrow v = \sigma^2 \left(\frac{1 - \phi^{2/S}}{1 - \phi^2} \right) I_d. \quad (\text{B.17})$$

Finally, we have

$$\tilde{p}_\theta(u_{\tau_t, s} | u_{\tau_t, s-1}) = \mathcal{N} \left(\phi^{1/S} u_{\tau_t, s-1}, \sigma^2 \left(\frac{1 - \phi^{2/S}}{1 - \phi^2} \right) I_d \right). \quad (\text{B.18})$$

Appendix C. Gradient derivation for parameter estimation

To estimate $\theta = (\alpha, \sigma, \phi)$ using the scheme detailed in Section 3, we must find expressions for $\frac{d}{d\theta} \log p(\mathbf{U}_t | \mathbf{U}_{t-1}, \theta)$ and $\frac{d}{d\theta} \log p(\mathcal{Y}_t | \mathbf{U}_t, \theta)$. For the model detailed in Section 2.1 we obtain

$$\frac{\partial}{\partial \alpha} \log p(\mathcal{Y}_t | \mathbf{U}_t, \theta) = \sum_{i < j} \left\{ y_{ijt} - \frac{1}{1 + \exp(\|u_{it} - u_{jt}\| - \alpha)} \right\}, \quad (\text{C.1})$$

$$= \sum_{i < j} (y_{ijt} - p_{ijt}). \quad (\text{C.2})$$

$$\frac{\partial}{\partial \sigma} \log p(\mathcal{Y}_t | \mathbf{U}_t, \theta) = \frac{\partial}{\partial \phi} \log p(\mathcal{Y}_t | \mathbf{U}_t, \theta) = \frac{\partial}{\partial \alpha} \log p(\mathbf{U}_t | \mathbf{U}_{t-1}, \theta) = 0. \quad (\text{C.3})$$

$$\frac{\partial}{\partial \sigma} \log p(\mathbf{U}_t | \mathbf{U}_{t-1}, \theta) = \sum_{i=1}^N \left\{ -\frac{d}{\sigma} + \frac{1}{\sigma^3} (u_{it} - \phi u_{i,t-1})^T (u_{it} - \phi u_{i,t-1}) \right\}. \quad (\text{C.4})$$

$$\frac{\partial}{\partial \phi} \log p(\mathbf{U}_t | \mathbf{U}_{t-1}, \theta) = \sum_{i=1}^N \left\{ \frac{1}{\sigma^2} u_{i,t-1}^T (u_{it} - \phi u_{i,t-1}) \right\}. \quad (\text{C.5})$$

where (C.5) follows from (84) in [52].

Since $\sigma > 0$ and $\phi \in (0, 1)$, we opt to estimate $\tilde{\sigma}$ and $\tilde{\phi}$ as specified in (12). From the chain rule we have

$$\frac{\partial}{\partial \tilde{\sigma}} = \frac{\partial}{\partial \sigma} \frac{\partial \sigma}{\partial \tilde{\sigma}} \quad \text{and} \quad \frac{\partial}{\partial \tilde{\phi}} = \frac{\partial}{\partial \phi} \frac{\partial \phi}{\partial \tilde{\phi}}. \quad (\text{C.6})$$

Then, we obtain the required expressions using

$$\frac{\partial \tilde{\sigma}}{\partial \sigma} = \frac{1}{\sigma} \quad \text{and} \quad \frac{\partial \tilde{\phi}}{\partial \phi} = \frac{1}{\phi(1 - \phi)}. \quad (\text{C.7})$$

$$(\text{C.8})$$

For stability reasons, we scale each of the gradients by a $1/Nd$.

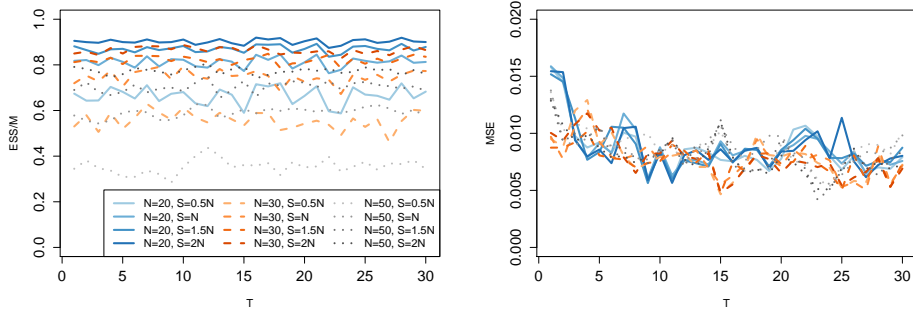


Figure D.9: Performance of GIRF as N increases for varying number of intermediary states S . The ESS and average MSE in probability are shown in the left and right panels, respectively, and the number of particles was fixed at $M = 10000$. Blue, red and grey lines correspond to $N = 20$, $N = 30$ and $N = 50$, respectively.

Appendix D. Suitability of the GIRF

To assess the suitability of this approach, we fit the GIRF to data simulated from the model in Section 2.1 with $\alpha = 1.2$, $\sigma = 0.2$ and $\phi = 0.9$. Figure D.9 shows the performance of the GIRF for different choices of S and varying N when the assessment function is taken as $v_{\tau_{t,s}}(\mathbf{U}) = p(\mathcal{Y}_{t+1}|\mathbf{U})$ and θ is assumed known. The left plot depicts the ESS (see Section Appendix A.2) divided by the number of particles and the right hand side shows the mean square error in probability. For $t = 1, 2, \dots, T$, this is given by

$$MSE_{prob}^{(t)} = \frac{1}{\binom{N}{2}} \sum_{i < j} (p_{ijt} - \hat{p}_{ijt})^2, \quad (\text{D.1})$$

where p_{ijt} and \hat{p}_{ijt} represent the true and estimated probability of nodes i and j sharing a tie at time t , respectively.

Figure D.9 indicates that the accuracy of the filter remains reasonably stable as the dimension of the state space increases. We also see that estimation is more challenging as N increases and that S significantly impacts the performance of the filter. It is clear that too few intermediary states will result in poor quality approximations. Finally, we comment here that little difference in the performance was observed when the assessment function was given by (8) and it is not clear whether this is true more generally. For example, we may find that incorporating future observations is more worthwhile for non-binary interactions.

Algorithm 6 Procedure for initialising σ

Input: $\mathcal{Y}_1, \mathcal{Y}_2$ and d

For $t \in \{1, 2\}$:

- Calculate the $N \times N$ distance matrix $D_t = \{d_{ijt}\}_{i,j \in 1:N}$, where d_{ijt} is the graph distance between nodes i and j . We set $d_{iit} = 0$ and, if $d_{ijt} = \infty$, we set d_{ijt} to be the maximum finite entry of D_t plus some constant $\gamma > 0$.
- Calculate the MDS coordinates with latent space dimension d , \hat{U}_t
- Calculate $\sigma_t = \frac{1}{Nd} \sum_{i=1}^N \hat{u}_{it}$

Let $\sigma = (\sigma_1 + \sigma_2)/2$

Appendix E. Static parameter initialisation

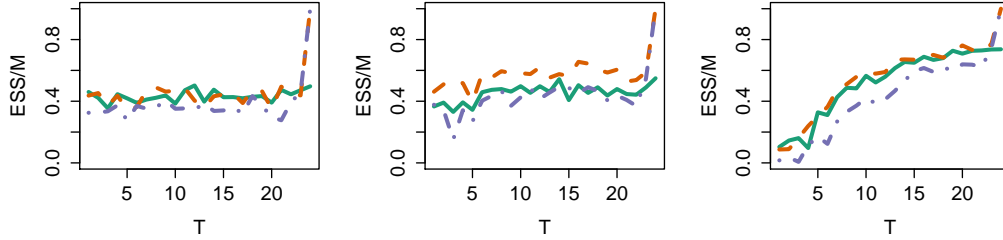
To initialise σ we follow [66] and adapt the GMDS approach considered in [60] and our procedure is detailed in Algorithm 6. We initialise ϕ at 0.8, and then choose α via a grid search conditional on σ and ϕ . More specifically, for a sequence of candidate α values, we simulate a series of latent trajectories given σ and ϕ . Then, we initialise α by taking the value which results in a network with density that is most similar to the average observed density $\sum_{t=1}^T \sum_{i < j} y_{ijt} / T \binom{N}{2}$.

Appendix F. Alternative Scenarios

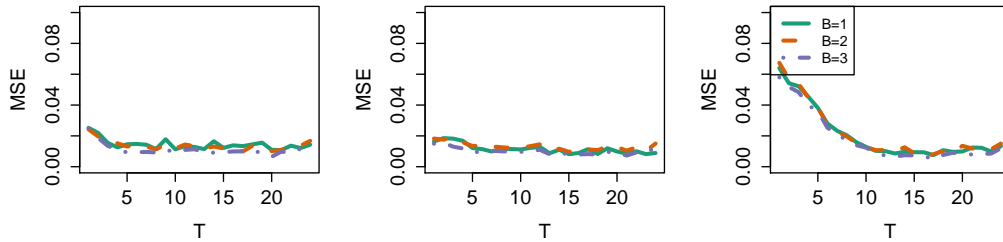
Figures F.10, F.11 and F.12 summarise the fit for offline estimation procedure for the data scenarios considered in Section 4.1. Throughout we fix $S = 1.5N, M = 5000, d = 2$ and $N_\theta = 20$. Corresponding figures for the offline procedure are presented in Section 4.1.

Appendix G. Performance Plots for Scalability Simulation

Figure G.13 shows the performance of the filter as N and T increase and the data were generated according to the description in Section 4.2. From Figure G.13a, we see that the problem becomes more challenging as N grows and choosing S too small will decrease the performance of the filter. From Figure G.13b we see that the performance of the filter is stable at T increases. As expected, increasing the dimension of the state space is the more challenging aspect.



(a) Effective sample size.

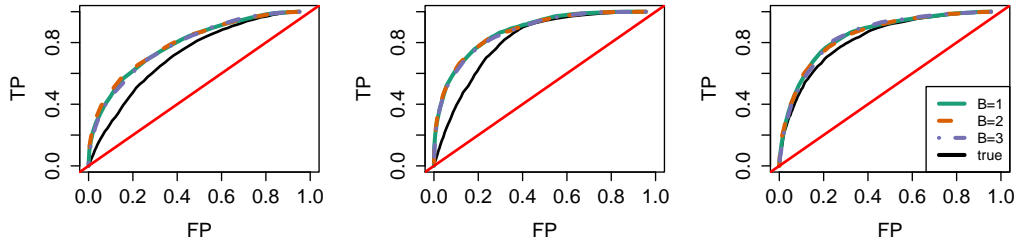


(b) Mean square error in probability (15).

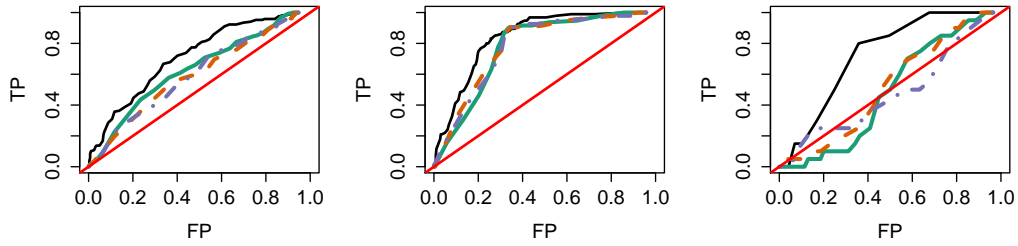
Figure F.10: Figure F.10a shows the effective sample size and Figure F.10b shows the mean square error in probability (see (15)) for the offline estimation procedure run with $B = 1$ (green, solid), $B = 2$ (orange, dashed) and $B = 3$ (purple, dot-dashed) look ahead steps. For each figure, left, middle and right correspond to data simulated according to models (S1), (S2) and (S3), respectively.

Appendix H. Stationarity assumption

The assumption of stationarity over the latent trajectories in our model ensures that the network properties are preserved through time. Without this assumption, the generative model gives rise to empty networks as T grows since the variance of the latent positions accrues over time. For the data example presented in Section 5, we plot some network summary statistics in Figure H.14 to justify this assumption. Whilst there is some variability in these summaries, there do not exist significant trends which suggest the assumption of stationarity to be unreasonable.



(a) ROC curve for observations at time 1 to time $T - 1$.



(b) ROC curve for observations at time T .

Figure F.11: Figures F.11a and F.11b show the ROC curves for observations 1 to $T - 1$ and for the predicted probabilities at time T , respectively. Each figure reports the ROC for the offline estimation procedure run with $B = 1$ (green, solid), $B = 2$ (orange, dashed) and $B = 3$ (purple, dot-dashed) look ahead steps. For each figure, left, middle and right correspond to data simulated according to models (S1), (S2) and (S3), respectively. The line $y = x$ is shown in red and the ROC curve for the true probabilities is shown in black.

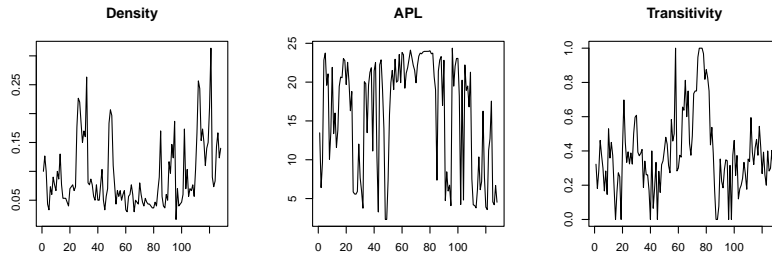


Figure H.14: Summary of the classroom contacts dataset considered in Section 5. Left, middle and right panels plot the density, average path length (APL) and transitivity of the observed graphs over time, respectively.

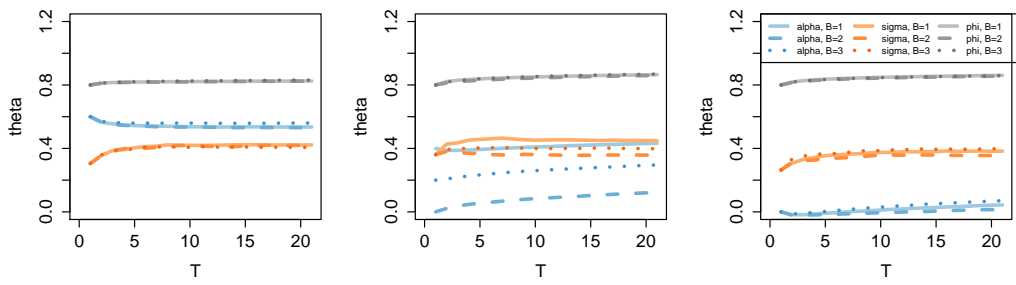
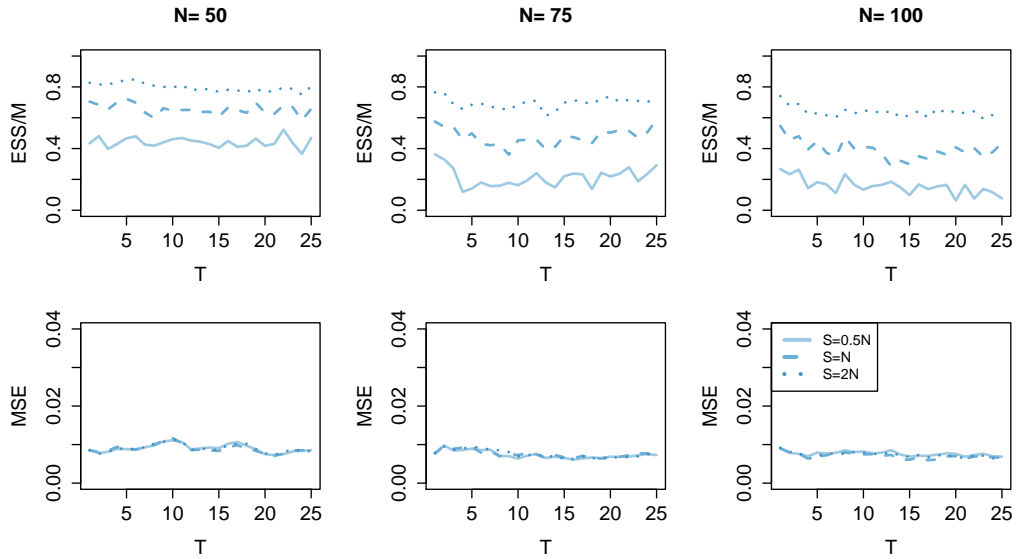
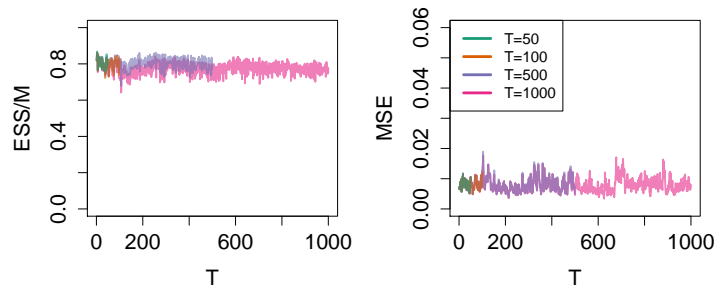


Figure F.12: Estimates of α (blue), σ (orange) and ϕ (grey) obtained via offline procedure. Left, middle and right figures correspond to the data generated according to (S1), (S2) and (S3), respectively.



(a) Summary of fit for increasing N with $T = 25$ and $S = 0.5N, N,$ and $2N$. The first row shows the ESS scaled by M and the second row shows the MSE in probability.



(b) Summary of fit for increasing T with $N = 20$ and $S = N$.

Figure G.13: Summary of performance of online SMC scheme as the dimension of the data increases in terms of N and T .