Statistics and Computing (2025) 35:58
https://doi.org/10.1007/s11222-025-10584-z

ORIGINAL PAPER

Simulation based composite likelihood

Lorenzo Rimella[1,2] · **Chris Jewell**[3] · **Paul Fearnhead**[3]

Received: 17 July 2024 / Accepted: 6 February 2025

Abstract
Inference for high-dimensional hidden Markov models is challenging due to the exponential-in-dimension computational cost of calculating the likelihood. To address this issue, we introduce an innovative composite likelihood approach called "Simulation Based Composite Likelihood" (SimBa-CL). With SimBa-CL, we approximate the likelihood by the product of its marginals, which we estimate using Monte Carlo sampling. In a similar vein to approximate Bayesian computation (ABC), SimBa-CL requires multiple simulations from the model, but, in contrast to ABC, it provides a likelihood approximation that guides the optimization of the parameters. Leveraging automatic differentiation libraries, it is simple to calculate gradients and Hessians to not only speed up optimization but also to build approximate confidence sets. We present extensive empirical results which validate our theory and demonstrate its advantage over SMC, and apply SimBa-CL to real-world Aphtovirus data.

Keywords Hidden Markov model · Composite likelihood · Monte Carlo approximation · Individual-based models

1 Introduction

Discrete-state hidden Markov models (HMMs) are common in many applications, such as epidemics (Keeling and Rohani 2011), systems biology (Wilkinson 2018) and ecology (Glennie et al. 2023). Increasingly there is interest in individual-based models (e.g. Rimella et al. 2023b), in which the HMM explicitly describes the state of each individual agent in a population. For example, an individual-based epidemic model may characterize each person in a population as having a latent state, being either susceptible, infected, or recovered. This state is typically observed noisily, with a sample of individuals being detected as infected with a possibly imperfect diagnostic test (e.g. Cocker et al. 2023). Thus, whilst there may be only a small number of states for each individual, this corresponds to a latent state-space that grows exponentially with the number of individuals.

In theory, likelihood calculations for such discrete-state HMMs are tractable using the forward-backward recursions (Scott 2002). However, the computational cost of these recursions is at least linear in the size of the state-space of the HMM: this means that they are infeasible for individual-based models with moderate or larger population sizes. This has led to a range of approximate inference methods. These include Monte Carlo methods such as Markov chain Monte Carlo (MCMC) and sequential Monte Carlo (SMC). Whilst such methods can work well, often they scale poorly with the population size—which may lead to poor mixing of MCMC algorithms or large Monte Carlo variance of the weights in SMC. An alternative approach is approximate Bayesian computation (ABC), where one simulates from the model for different parameter values, and then approximates the posterior for the parameter based on how similar each simulated dataset is to the true data. Such a method needs informative, low-dimensional, summary statistics to be available so that one can accurately measure how close a simulated dataset is to the true data. Furthermore, ABC can struggle with complex models with many parameters, as the number of summary statistics needs to increase with the number of parameters (Fearnhead and Prangle 2012).

✉ Lorenzo Rimella
lorenzo.rimella@unito.it

Chris Jewell
c.jewell@lancaster.ac.uk

Paul Fearnhead
p.fearnhead@lancaster.ac.uk

[1] ESOMAS, University of Turin, Via Verdi 8, 10124 Turin, Italy

[2] Statistics Initiative, Collegio Carlo Alberto, Piazza Arbarello 8, 10122 Turin, Italy

[3] Mathematical Sciences, Lancaster University, Lancaster LA14YF, UK

Published online: 25 February 2025

In this paper, we consider individual-based HMMs where we have individual-level observations. We present a computationally efficient method for inference that is based on the simple observation: if we fix the state of all members of the population except one, then we can analytically calculate the conditional likelihood of that one individual using forward-backward recursions. This idea has been used before within MCMC algorithms that update the state of each individual in turn conditional on the states of the other individuals (Fintzi et al. 2017; Touloupou et al. 2020). Here we use it in a different way. By simulating multiple realizations of the states of the other individuals we can average the conditional likelihood to obtain a Monte Carlo estimate of the likelihood of the data for a given individual. We then sum the log of these estimated likelihoods over individuals to obtain a composite log-likelihood (Varin 2008) that can be maximized using, for example, stochastic gradient ascent, to estimate the parameters.

We introduce the general class of models we consider in Sect. 2. We then show how to obtain a Monte Carlo estimate of the likelihood for the observations associated with a single individual, which can be used as the basis of a composite likelihood for our model. The calculation of the likelihood for each individual involves accounting for feedback between the state of the individual in question, and the probability distribution of future states of the rest of the population. A computationally more efficient method can be obtained by ignoring this feedback—and we present theory that bounds the error of this approach, and shows that it can decay to zero as the population size tends to infinity. Then in Sect. 4 we show how we can get confidence regions around estimators based on maximizing our composite likelihood. We then demonstrate the efficiency for individual-based epidemic models both on simulated data and on data from the 2001 UK foot and mouth outbreak.

2 Model

2.1 Notation

Given the integer $t \in \mathbb{N}$, we denote the set of integers from 1 to t as $[t]$, and we use $[0:t]$ if we want to include 0. Additionally, we use $[t]$ as shorthand for indexing, for instance $x_{[t]}$ denotes the collection $x_1, \ldots, x_t$. If x is an N-dimensionsal vector, then given an index $n \in [N]$, we use x^n to denote the nth component of x and $x^{\backslash n}$ to denote the $(N-1)$-dimensional vector obtained by removing the nth component from x. If required, we augment the superscript notation and use $x^{(i)}$ to refer to the vector $x^{(i)}$ with components $x^{(i),n}$. For a finite and discrete set $\mathcal{S}$, we represent the cardinality of $\mathcal{S}$ as $\mathbf{card}\,(S)$, and we use the shorthand $\sum_x$ to express the sum over all elements of $\mathcal{S}$. We use bold font to denote random variables and regular font for deterministic quantities. For the underlying probability measure, we commonly use p and, for the sake of clarity, we focus on its functional form, for instance, we use $p\,(x_t|x_{t-1},\theta)$ for the probability of $\mathbf{x}_t = x_t$ given $\mathbf{x}_{t-1} = x_{t-1}$ and the parameters θ.

2.2 Hidden Markov models and likelihood computation

A hidden Markov model (HMM) $\left(\mathbf{x}_0, (\mathbf{x}_t, \mathbf{y}_t)_{t\geq 1}\right)$ is a stochastic process where the unobserved process $(\mathbf{x}_t)_{t\geq 0}$ is a Markov chain, and the observed process $(\mathbf{y}_t)_{t\geq 1}$ is such that, for any $t \geq 1$, $\mathbf{y}_t$ is conditionally independent of all the other variables given $\mathbf{x}_t$. See Chopin et al. (2020) for a comprehensive review of HMMs.

Within this paper, we focus on HMMs with finite-dimensional state-spaces. Precisely, we consider $(\mathbf{x}_t)_{t\geq 0}$ to take values on the state-space $\mathcal{X}^N$, which satisfies a product form, $\mathcal{X}^N = \bigtimes_{n\in[N]} \mathcal{X}$, where $\mathcal{X}$ is finite and discrete. We also consider $(\mathbf{y}_t)_{t\geq 1}$ to take values in the space $\mathcal{Y}^N$, which also satisfies a product form, $\mathcal{Y}^N = \bigtimes_{n\in[N]} \mathcal{Y}$, but here $\mathcal{Y}$ can be of any form.

Given a collection of parameters θ, an HMM is fully defined through its components: the initial distribution $p\,(x_0|\theta)$, which is the distribution of $\mathbf{x}_0$; the transition kernel $p\,(x_t|x_{t-1},\theta)$, which is the distribution of $\mathbf{x}_t$ given $\mathbf{x}_{t-1}$; and the emission distribution $p\,(y_t|x_t,\theta)$, which is the distribution of $\mathbf{y}_t$ given $\mathbf{x}_t$. For a given time horizon $T \in \mathbb{N}$, we may assume that the data sequence $y_1, \ldots, y_T$ is generated from the aforementioned hidden Markov model with parameters $\theta^\star$. Our primary interest is in inferring the parameter $\theta^\star$, responsible for generating the data or, in cases where the model is not fully identifiable, a set of parameters that are equally likely. Given the assumption that $\mathcal{X}$ is finite and discrete, the probability distribution $p\,(x_0|\theta)$ takes the form of a probability vector with $\mathbf{card}\,(\mathcal{X})^N$ elements, while $p\,(x_t|x_{t-1},\theta)$ corresponds to a $\mathbf{card}\,(\mathcal{X})^N \times \mathbf{card}\,(\mathcal{X})^N$ stochastic matrix. The computation of the likelihood for HMMs with discrete state-space is relatively straightforward and involves marginalization over the entire state-space:

$$p\left(y_{[T]}|\theta\right) = \sum_{x_{[0:T]}} p\,(x_0|\theta) \prod_{t\in[T]} p\,(x_t|x_{t-1},\theta)\, p\,(y_t|x_t,\theta)\,. \tag{1}$$

In practice, to avoid marginalizing on an exponential-in-time state-space, the likelihood is recursively computed using the forward algorithm, which recursively computes the filtering distribution $p(x_t|y_t,\theta)$ and the likelihood increments $p(y_t|y_{[t-1]},\theta)$. The $t+1$ step of the forward algorithm comprises two operations, namely, prediction:

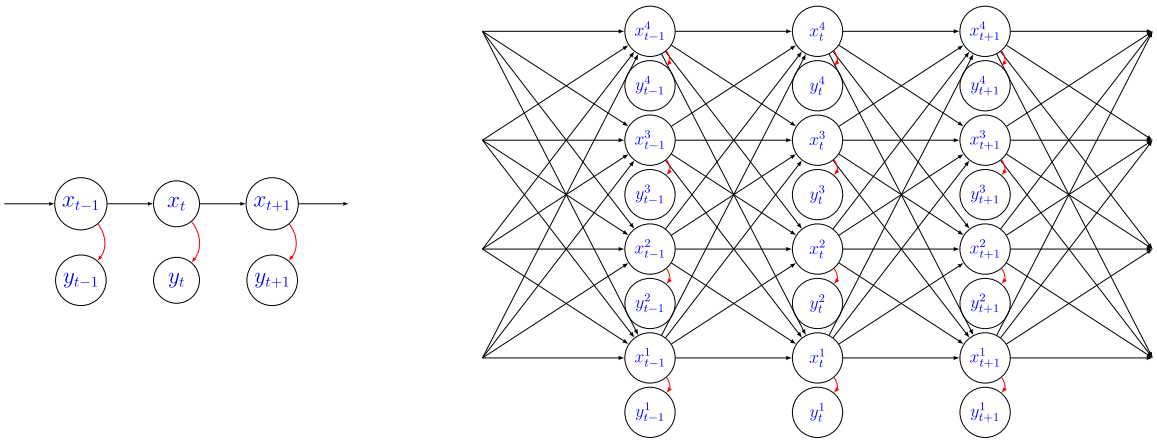

Fig. 1 Left: conditional independence structure of a standard HMM. Right: conditional independence structure of an HMM satisfying (2), with $N = 4$

$$\left\{\begin{array}{l} p\left(x_{t+1}|x_t,\theta\right) \\ p\left(x_t|y_{[t]},\theta\right) \end{array}\right\} \xrightarrow{\text{prediction}}$$
$$\left\{p\left(x_{t+1}|y_{[t]},\theta\right) = \sum_{x_t} p\left(x_{t+1}|x_t,\theta\right) p\left(x_t|y_{[t]},\theta\right)\right\},$$

where the transition kernel is applied to the previous filtering distribution, and correction:

$$\left\{\begin{array}{l} p\left(y_{t+1}|x_{t+1},\theta\right) \\ p\left(x_{t+1}|y_{[t]},\theta\right) \end{array}\right\} \xrightarrow{\text{correction}}$$
$$\left\{\begin{array}{l} p\left(x_{t+1}|y_{[t+1]},\theta\right) = \frac{p(y_{t+1}|x_{t+1},\theta)p(x_{t+1}|y_{[t]},\theta)}{p(y_{t+1}|y_{[t]},\theta)} \\ p\left(y_{t+1}|y_{[t]},\theta\right) = \sum_{x_{t+1}} p(y_{t+1}|x_{t+1},\theta)p\left(x_{t+1}|y_{[t]},\theta\right) \end{array}\right\}$$

from which the likelihood increments $p\left(y_t|y_{[t-1]},\theta\right)$, with $p\left(y_1|y_{[0]},\theta\right) := p\left(y_1|\theta\right)$, are then combined to compute the likelihood:

$$p\left(y_{[T]}|\theta\right) = \prod_{t\in[T]} p\left(y_t|y_{[t-1]},\theta\right).$$

Despite its simplicity, the forward algorithm necessitates a marginalization on the full state-space, incurring a computational cost that is, at worst, quadratic in the cardinality of the state-space. This translates to a complexity of $\mathcal{O}\left(\mathbf{card}\left(\mathcal{X}\right)^{2N}\right)$, making the forward algorithm unfeasible for large values of N.

2.3 Factorial structure

We consider HMMs with initial distribution, transition kernel, and emission distribution that satisfy the following factorizations:

$$\begin{aligned} p\left(x_0|\theta\right) &= \prod_{n\in[N]} p\left(x_0^n|\theta\right), \\ p\left(x_t|x_{t-1},\theta\right) &= \prod_{n\in[N]} p\left(x_t^n|x_{t-1},\theta\right), \\ p\left(y_t|x_t,\theta\right) &= \prod_{n\in[N]} p\left(y_t^n|x_t^n,\theta\right), \end{aligned} \tag{2}$$

which essentially says that we can decompose the initial distribution in N probability vectors of size $\mathbf{card}\left(\mathcal{X}\right)$, the transition kernel in N stochastic matrices that are $\mathbf{card}\left(\mathcal{X}\right) \times \mathbf{card}\left(\mathcal{X}\right)$, whose elements also depend on x_{t-1}, and the observation n is conditionally independent of all the other variables given $\mathbf{x}_t^n$, see Fig. 1. Note that the introduced factorization does not resolve our problems as the $p\left(x_t^n|x_{t-1},\theta\right)$ still depends on the whole space at time $t-1$, not allowing a full decoupling across the dimensions. Rather, it serves as an essential foundation upon which we construct our approximation. Furthermore, the factorization given by (2) is natural in several real-world applications, including epidemics (Rimella et al. 2023a, b), traffic modelling (Silva et al. 2015) and finance (Samanidou et al. 2007).

It is important to note that (2), apart from holding for many real-world discrete-time models, does not require evenly spaced observation. Indeed, we could equivalently work on a more general version of (2), where data are observed at $0 = \tau_0 < \tau_1 < \cdots < \tau_t$, and both the transition kernel and the emission distribution are time inhomogeneous (Rimella et al. 2023b; Duffield et al. 2023):

$$\begin{aligned} p_0\left(x_0|\theta\right) &= \prod_{n\in[N]} p_0\left(x_0^n|\theta\right), \\ p_{\tau_t}\left(x_{\tau_t}|x_{\tau_{t-1}},\theta\right) &= \prod_{n\in[N]} p_{\tau_t}\left(x_{\tau_t}^n|x_{\tau_{t-1}},\theta\right), \\ p_{\tau_t}\left(y_{\tau_t}|x_{\tau_t},\theta\right) &= \prod_{n\in[N]} p_{\tau_t}\left(y_{\tau_t}^n|x_{\tau_t}^n,\theta\right). \end{aligned}$$

To avoid cumbersome notation we use the formulation from (2) throughout the paper.

3 Simulation based composite likelihood: SimBa-CL

From the model structure shown in Sect. 2.3, if we fix all but one component of the latent process, $x^{\backslash n}_{[T]}$ say, we can leverage the factorization and calculate probabilities related to the time-trajectory of the remaining state, $x^n_{[T]}$, with a computational cost that is $\mathcal{O}(\mathbf{card}(\mathcal{X}))$. This idea has been used within Gibbs-style MCMC updates for epidemics see Fintzi et al. (2017); Touloupou et al. (2020). We show how to use this idea, together with using Monte Carlo to average over $x^{\backslash n}_{[T]}$, to calculate the marginal likelihoods $p(y^n_{[T]}|\theta)$. We can then use the product of these marginal likelihoods, $\prod_{n\in[N]} p(y^n_{[T]}|\theta)$, as a composite likelihood (Varin 2008) that can be maximized to estimate θ.

Using $p\left(y_{[T]}|\theta\right) \approx \prod_{n\in[N]} p\left(y^n_{[T]}|\theta\right)$ still falls short, as the computation of $p\left(y^n_{[T]}|\theta\right)$ continues to require a recursive marginalization on $\mathcal{X}^N$. Yet, we can express the marginal likelihood $p\left(y^n_{[T]}|\theta\right)$ as:

$$p\left(y^n_{[T]}|\theta\right) = \sum_{x^{\backslash n}_{[0:T-1]}} p\left(x^{\backslash n}_{[0:T-1]}|\theta\right) p\left(y^n_{[T]}|x^{\backslash n}_{[0:T-1]},\theta\right), \tag{3}$$

where:

$$\begin{aligned} &p\left(y^n_{[T]}|x^{\backslash n}_{[0:T-1]},\theta\right) \\ &= \sum_{x^n_{[0:T]}} p\left(x^n_T|x_{T-1},\theta\right) p\left(x^n_{[0:T-1]}|x^{\backslash n}_{[0:T-1]},\theta\right) \\ &\quad \prod_{t\in[T]} p\left(y^n_t|x^n_t,\theta\right). \end{aligned}$$

We have two necessary ingredients for calculating $p\left(y^n_{[T]}|\theta\right)$: firstly, $p\left(y^n_{[T]}|x^{\backslash n}_{[0:T-1]},\theta\right)$, which demands T recursive marginalizations on $\mathcal{X}$ given $x^{\backslash n}_{[0:T-1]}$; secondly, a marginalization on $\mathcal{X}^{N-1}$ through $p\left(x^{\backslash n}_{[0:T-1]}|\theta\right)$, see (3). (3) using Monte Carlo sampling.

We refer to this procedure as "Simulation Based Composite Likelihood", or "SimBa-CL" in short. In the following sections, we give an in-depth discussion on SimBa-CL and show how we can target the true marginals of the likelihood and build a likelihood approximation in $\mathcal{O}\left(N^2\right)$, see Sect. 3.1, how to approximate the marginals of the likelihood and build an approximation of the likelihood in $\mathcal{O}(N)$, see Sect. 3.2, and how to generalize SimBa-CL, see Sect. 3.4. For the sake of presentation, we remove the dependence on the parameter θ and focus on the filtering aspects of the algorithms for a fixed θ.

3.1 SimBa-CL with feedback

Given efficient sampling from $p\left(x^{\backslash n}_{[0:T-1]}\right)$ and low-cost evaluation of $p\left(y^n_{[T]}|x^{\backslash n}_{[0:T-1]}\right)$ for a given $x^{\backslash n}_{[0:T-1]}$, we can obtain a Monte Carlo estimate of the marginal likelihood from (3):

$$p\left(y^n_{[T]}\right) \approx \frac{1}{P}\sum_{i\in[P]} p\left(y^n_{[T]}|x^{(i),\backslash n}_{[0:T-1]}\right), \tag{4}$$

where $P \in \mathbb{N}$ is the number of Monte Carlo samples and $x^{(i),\backslash n}_{[0:T-1]} \sim p\left(x^{\backslash n}_{[0:T-1]}\right)$. Repeating (4) for all $n \in [N]$ and computing the product across n of these Monte Carlo estimates represents a reasonable strategy for approximating the likelihood of the model.

Two ingredients are pivotal in the computation of (4): (i) sampling from the model and (ii) calculating $p\left(y^n_{[T]}|x^{\backslash n}_{[0:T-1]}\right)$. Sampling from $p\left(x^{\backslash n}_{[0:T-1]}\right)$ can be achieved by sampling $x_{[0:T-1]}$ from $p\left(x_{[0:T-1]}\right)$ and then selecting the subset $x^{\backslash n}_{[0:T-1]}$. Sampling from the entire process is generally straightforward.

For the computation of $p\left(y^n_{[T]}|x^{\backslash n}_{[0:T-1]}\right)$, it is important to recognize that $p\left(x^n_{[0:T-1]}|x^{\backslash n}_{[0:T-1]}\right)$ can be reformulated as a product between the transition dynamics and the probability of observing a certain simulation outcome:

$$\begin{aligned} &p\left(x^n_{[0:T-1]}|x^{\backslash n}_{[0:T-1]}\right) \\ &\quad = p\left(x^n_0\right) \prod_{t\in[T-1]} p\left(x^n_t|x_{t-1}\right) f\left(x^n_{t-1}, x^{\backslash n}_{[0:t]}\right), \end{aligned} \tag{5}$$

where we refer to $f(x^n_{t-1}, x^{\backslash n}_{[0:t]}) := p(x^{\backslash n}_t|x^n_{t-1}, x^{\backslash n}_{[0:t-1]})$ as the simulation feedback, and so:

$$\begin{aligned} &f\left(x^n_{t-1}, x^{\backslash n}_{[0:t]}\right) \\ &\quad = \frac{\prod_{\bar{n}\in[N]\backslash n} p\left(x^{\bar{n}}_t|x_{t-1}\right)}{\sum_{\bar{x}^n_{t-1}} \prod_{\bar{n}\in[N]\backslash n} p\left(x^{\bar{n}}_t|\bar{x}^n_{t-1}, x^{\backslash n}_{[t-1]}\right) p\left(\bar{x}^n_{t-1}|x^{\backslash n}_{[0:t-1]}\right)}, \end{aligned} \tag{6}$$

where $p\left(x^n_0|x^{\backslash n}_{[0:0]}\right) = p\left(x^n_0\right)$ for the factorization of the initial distribution. The intuition is that this term accounts

for how changing x_{t-1}^n affects the probability of $x_t^{\backslash n}$. More details on the factorization (5) and the derivation of the simulation feedback (6) are available in Section A.1 of the supplementary material.

By reformulating $p\left(x_{[0:T-1]}^n | x_{[0:T-1]}^{\backslash n}\right)$ as depicted in (5), we arrive at the following expression:

$$p\left(y_{[T]}^n | x_{[0:T-1]}^{\backslash n}\right) = \sum_{x_{[0:T]}^n} p\left(x_0^n\right) \prod_{t\in[T-1]} f\left(x_{t-1}^n, x_{[0:t]}^{\backslash n}\right) \prod_{t\in[T]} p\left(x_t^n | x_{t-1}\right) p\left(y_t^n | x_t^n\right), \tag{7}$$

which resembles the likelihood of an HMM, see (1). Specifically, it comprises the usual transition dynamic term $p\left(x_t^n | x_{t-1}\right)$ accompanied by two likelihood terms: one originating from the simulation outcome $f\left(x_{t-1}^n, x_{[0:t]}^{\backslash n}\right)$, and another concerning the observation $p\left(y_t^n | x_t^n\right)$. We can then establish a forward algorithm involving two corrections, one that is correcting according to the emission distribution:

$$\left\{\begin{array}{c} p\left(y_t^n | x_t^n\right) \\ p\left(x_t^n | y_{[t-1]}^n, x_{[0:t]}^{\backslash n}\right) \end{array}\right\} \overset{\text{observation correction}}{\longrightarrow} \left\{\begin{array}{l} p\left(x_t^n | y_{[t]}^n, x_{[0:t]}^{\backslash n}\right) = \frac{p\left(y_t^n | x_t^n\right) p\left(x_t^n | y_{[t-1]}^n, x_{[0:t]}^{\backslash n}\right)}{p\left(y_t^n | y_{[t-1]}^n, x_{[0:t]}^{\backslash n}\right)} \\ p\left(y_t^n | y_{[t-1]}^n, x_{[0:t]}^{\backslash n}\right) \\ \quad = \sum_{x_t^n} p\left(y_t^n | x_t^n\right) p\left(x_t^n | y_{[t-1]}^n, x_{[0:t]}^{\backslash n}\right) \end{array}\right\}, \tag{8}$$

and the other that is correcting according to the simulation feedback:

$$\left\{\begin{array}{c} f\left(x_t^n, x_{[0:t+1]}^{\backslash n}\right) \\ p\left(x_t^n | y_{[t]}^n, x_{[0:t]}^{\backslash n}\right) \end{array}\right\} \overset{\text{feedback correction}}{\longrightarrow} \left\{\begin{array}{l} p\left(x_t^n | y_{[t]}^n, x_{[0:t+1]}^{\backslash n}\right) \\ \quad = \frac{f\left(x_t^n, x_{[0:t+1]}^{\backslash n}\right) p\left(x_t^n | y_{[t]}^n, x_{[0:t]}^{\backslash n}\right)}{p\left(x_{t+1}^{\backslash n} | y_{[t]}^n, x_{[0:t]}^{\backslash n}\right)} \\ p\left(x_{t+1}^{\backslash n} | y_{[t]}^n, x_{[0:t]}^{\backslash n}\right) \\ = \sum_{x_t^n} f\left(x_t^n, x_{[0:t+1]}^{\backslash n}\right) p\left(x_t^n | y_{[t]}^n, x_{[0:t]}^{\backslash n}\right) \end{array}\right\}.$$

The prediction follows as is in the basic HMM scenario with $p(x_t^n | x_{t-1})$ as transition kernel and $p\left(x_{t-1}^n | y_{[t-1]}^n, x_{[0:t]}^{\backslash n}\right)$ for the distribution to update:

$$\left\{\begin{array}{c} p(x_t^n | x_{t-1}) \\ p\left(x_{t-1}^n | y_{[t-1]}^n, x_{[0:t]}^{\backslash n}\right) \end{array}\right\} \overset{\text{feedback prediction}}{\longrightarrow} \left\{\begin{array}{l} p\left(x_t^n | y_{[t-1]}^n, x_{[0:t]}^{\backslash n}\right) \\ = \sum_{x_{t-1}^n} p(x_t^n | x_{t-1}) p\left(x_{t-1}^n | y_{[t-1]}^n, x_{[0:t]}^{\backslash n}\right) \end{array}\right\}. \tag{9}$$

The computation of the simulation feedback $f\left(x_{t-1}^n, x_{[0:t-1]}^{\backslash n}\right)$ relies on $p\left(x_{t-1}^n | x_{[0:t-1]}^{\backslash n}\right)$, the posterior distribution of x_{t-1}^n given the simulation output as observations. Consequently, this interpretation enables the employment of another forward algorithm to compute recursively these intermediate quantities, where the correction step is given by:

$$\left\{\begin{array}{c} \prod_{\bar{n}\in[N]\backslash n} p\left(x_t^{\bar{n}} | x_{t-1}\right) \\ p\left(x_{t-1}^n | x_{[0:t-1]}^{\backslash n}\right) \end{array}\right\} \overset{\text{correction}}{\longrightarrow} \left\{\begin{array}{l} p\left(x_{t-1}^n | x_{[0:t]}^{\backslash n}\right) \\ \quad = \frac{\prod_{\bar{n}\in[N]\backslash n} p\left(x_t^{\bar{n}} | x_{t-1}\right) p\left(x_{t-1}^n | x_{[0:t-1]}^{\backslash n}\right)}{p\left(x_t^{\backslash n} | x_{[0:t-1]}^{\backslash n}\right)} \\ p\left(x_t^{\backslash n} | x_{[0:t-1]}^{\backslash n}\right) \\ = \sum_{x_{t-1}^n} \prod_{\bar{n}\in[N]\backslash n} p\left(x_t^{\bar{n}} | x_{t-1}\right) p\left(x_{t-1}^n | x_{[0:t-1]}^{\backslash n}\right) \end{array}\right\}, \tag{10}$$

and the prediction follows:

$$\left\{\begin{array}{c} p\left(x_t^n | x_{t-1}\right) \\ p\left(x_{t-1}^n | x_{[0:t]}^{\backslash n}\right) \end{array}\right\} \overset{\text{prediction}}{\longrightarrow} \left\{p\left(x_t^n | x_{[0:t]}^{\backslash n}\right) = \sum_{x_{t-1}^n} p\left(x_t^n | x_{t-1}\right) p\left(x_{t-1}^n | x_{[0:t]}^{\backslash n}\right)\right\}. \tag{11}$$

An iterative application of the aforementioned steps provides a collection of likelihood increments on both the simulation output and the observations, enabling the computation of $p\left(y_{[T]}^n | x_{[0:T-1]}^{\backslash n}\right)$ as follows:

$$p\left(y_{[T]}^n | x_{[0:T-1]}^{\backslash n}\right) = p\left(y_T^n | y_{[T-1]}^n, x_{[0:T-1]}^{\backslash n}\right) \prod_{t\in[T-1]} p\left(y_t^n | y_{[t-1]}^n, x_{[0:t]}^{\backslash n}\right) p\left(x_t^{\backslash n} | y_{[t-1]}^n, x_{[0:t-1]}^{\backslash n}\right), \tag{12}$$

where $p\left(y_1^n|y_{[0]}^n, x_{[0:0]}^{\backslash n}\right) := p\left(y_1^n|x_0^{\backslash n}\right)$ and $p\left(x_1^{\backslash n}|y_{[0]}^n, x_{[0]}^{\backslash n}\right) := p\left(x_1^{\backslash n}|x_{[0]}^{\backslash n}, \theta\right)$.

Algorithm 1 SimBa-CL with feedback

Require: $p(x_0)$, $p(x_t|x_{t-1})$, $p(y_t|x_t)$ and their factorizations
for each $i \in [P]$ **do**
  $x_0^{(i),} \sim p(x_0)$
  for $t \in [T]$ **do**
    $x_t^{(i),} \sim p\left(x_t|x_{t-1}^{(i),}\right)$ and compute $f\left(x_{t-1}^n, x_{[0:t]}^{(i),\backslash n}\right)$
    for each $n \in [N]$ **do**
      if $t \neq T$, run the feedback correction (3.1) and get: $\left\{\begin{array}{c} p\left(x_{t-1}^n|y_{[t-1]}^n, x_{[0:t]}^{(i),\backslash n}\right) \\ p\left(x_t^{(i),\backslash n}|y_{[t-1]}^n, x_{[0:t-1]}^{(i),\backslash n}\right)\end{array}\right\}$
      Run the prediction (9) and get: $\left\{p\left(x_t^n|y_{[t-1]}^n, x_{[0:t]}^{(i),\backslash n}\right)\right\}$
      Run the observation correction (8) and get: $\left\{\begin{array}{c} p\left(x_t^n|y_{[t]}^n, x_{[0:t]}^{(i),\backslash n}\right) \\ p\left(y_t^n|y_{[t-1]}^n, x_{[0:t]}^{(i),\backslash n}\right)\end{array}\right\}$
      Run the correction (10) and get: $\left\{\begin{array}{c} p\left(x_{t-1}^n|x_{[0:t]}^{(i),\backslash n}\right) \\ p(x_t^{(i),\backslash n}|x_{[0:t-1]}^{(i),\backslash n})\end{array}\right\}$
      Run the prediction (11) and get: $\left\{p\left(x_t^n|x_{[0:t]}^{(i),\backslash n}\right)\right\}$
    end for
  end for
  Compute $p\left(y_{[T]}^n|x_{[0:T-1]}^{(i),\backslash n}\right)$ as in (12)
end for
Return $\frac{1}{P}\sum_{i\in[P]} p\left(y_{[T]}^n|x_{[0:T-1]}^{(i),\backslash n}\right)$

The final algorithm, named "SimBa-CL with feedback", is presented in Algorithm 1 and it shows a computational complexity of $\mathcal{O}\left(TN^2P\mathbf{card}\left(\mathcal{X}\right)^2\right)$, wherein P, T, N come from looping over number of simulations, time steps and dimensions, $\mathbf{card}\left(\mathcal{X}\right)^2$ comes from marginalizing over the state-space $\mathcal{X}$, and the extra N terms refers to the simulation feedback computation.
This cost can potentially be reduced to $\mathcal{O}\left(TNP\max_n\{\mathbf{card}\left(\mathbf{Neig}\left(n\right)\right)\}\mathbf{card}\left(\mathcal{X}\right)^2\right)$ if the transition kernel $p\left(x_t^n|x_{t-1}, \theta\right)$ presents some local structure. Precisely, if $p\left(x_t^n|x_{t-1}, \theta\right) = p\left(x_t^n|\bar{x}_{t-1}, \theta\right)$ for any $x_{t-1}^{\mathbf{Neig}(n)} = \bar{x}_{t-1}^{\mathbf{Neig}(n)}$, where $\mathbf{Neig}$ represents a function mapping any $n \in [N]$ onto a set in the power set of $[N]$. This indicates that computing the simulation feedback can be computationally cheaper if the inter-dimension interactions are sparse. Also, the algorithm can be readily parallelized across both the dimensions and the simulations, rendering the dependence concerning (P, N) less heavy in the presence of suitable hardware.

3.2 SimBa-CL without feedback

An important aspect of SimBa-CL with feedback is that it targets the true marginals of the likelihood. However, one could contemplate a strategy involving the removal of the simulation feedback and design a SimBa-CL that is more computationally efficient, while being "close" to SimBa-CL with feedback.

Looking back at (7), omitting the simulation feedback yields to:

$$\begin{aligned} p\left(y_{[T]}^n|x_{[0:T-1]}^{\backslash n}\right) &\approx \sum_{x_{[0:T]}^n} p\left(x_0^n\right) \prod_{t\in[T]} p\left(x_t^n|x_{t-1}\right) p\left(y_t^n|x_t^n\right) \\ &=: \tilde{p}\left(y_{[T]}^n|x_{[0:T-1]}^{\backslash n}\right), \end{aligned} \tag{13}$$

from which we have the following marginal likelihood approximation:

$$p\left(y_{[T]}^n\right) \approx \sum_{x_{[0:T-1]}^{\backslash n}} p\left(x_{[0:T-1]}^{\backslash n}\right) \tilde{p}\left(y_{[T]}^n|x_{[0:T-1]}^{\backslash n}\right) =: \tilde{p}\left(y_{[T]}^n\right),$$

where we emphasized that we are relying on an approximation by using the notation $\tilde{p}$. It can be seen that $\tilde{p}\left(y_{[T]}^n\right)$ is still a proper marginal likelihood, as it sums to one when marginalizing on $\mathcal{Y}$, but it is not the true marginal likelihood.

Upon scrutinizing (13), we can recognize the same likelihood structure as (1). This time, we are isolating our calculation to a single component n and fixing the others through simulation from the model. This suggests that a simple forward algorithm can be run in isolation on each dimension. Concretely, the corresponding forward algorithm will require a prediction step:

$$\begin{aligned} &\left\{\begin{array}{c} p\left(x_{t+1}^n|x_t\right) \\ \tilde{p}\left(x_t^n|y_{[t]}^n, x_{[0:t-1]}^{\backslash n}\right)\end{array}\right\} \xrightarrow{\text{prediction}} \\ &\left\{\begin{array}{c} \tilde{p}\left(x_{t+1}^n|y_{[t]}^n, x_{[0:t]}^{\backslash n}\right) \\ = \sum_{x_t^n} p\left(x_{t+1}^n|x_t\right) \tilde{p}\left(x_t^n|y_{[t]}^n, x_{[0:t-1]}^{\backslash n}\right)\end{array}\right\}, \end{aligned} \tag{14}$$

and a correction step:

$$\begin{aligned} &\left\{\begin{array}{c} p\left(y_{t+1}^n|x_{t+1}^n\right) \\ \tilde{p}\left(x_{t+1}^n|y_{[t]}^n, x_{[0:t]}^{\backslash n}\right)\end{array}\right\} \xrightarrow{\text{correction}} \\ &\left\{\begin{array}{c} \tilde{p}\left(x_{t+1}^n|y_{[t+1]}^n, x_{[0:t]}^{\backslash n}\right) \\ \quad = \frac{p\left(y_{t+1}^n|x_{t+1}^n\right)\tilde{p}\left(x_{t+1}^n|y_{[t]}^n, x_{[0:t]}^{\backslash n}\right)}{\tilde{p}\left(y_{t+1}^n|y_{[t]}^n, x_{[0:t]}^{\backslash n}\right)} \\ \tilde{p}\left(y_{t+1}^n|y_{[t]}^n, x_{[0:t]}^{\backslash n}\right) \\ = \sum_{x_{t+1}^n} p\left(y_{t+1}^n|x_{t+1}^n\right) \tilde{p}\left(x_{t+1}^n|y_{[t]}^n, x_{[0:t]}^{\backslash n}\right)\end{array}\right\}. \end{aligned} \tag{15}$$

Recursively applying (14) and (15) provides a sequence of approximate marginal likelihood increments which can be

Algorithm 2 Simulation likelihood without feedback

Require: $p(x_0)$, $p(x_t|x_{t-1})$, $p(y_t|x_t)$ and their factorization
for each $i \in [P]$ **do**
  $x_0^{(i),} \sim p(x_0)$
  for $t \in [T]$ **do**
    $x_t^{(i),} \sim p\left(x_t|x_{t-1}^{(i),}\right)$
    for each $n \in [N]$ **do**
      Run the prediction (14) and get: $\left\{\tilde{p}\left(x_t^n|y_{[t-1]}^n, x_{[0:t-1]}^{(i),\backslash n}\right)\right\}$
      Run the correction (15) and get: $\left\{\begin{array}{l}\tilde{p}\left(x_t^n|y_{[t]}^n, x_{[0:t-1]}^{(i),\backslash n}\right)\\ \tilde{p}\left(y_t^n|y_{[t-1]}^n, x_{[0:t-1]}^{(i),\backslash n}\right)\end{array}\right\}$
    end for
  end for
end for
Return $\frac{1}{P}\sum_{i\in[P]}\tilde{p}\left(y_{[T]}^n|x_{[0:T-1]}^{i,\backslash n}, \theta\right)$

then used to approximate the marginal likelihood for a fixed simulation in the usual way:

$$\tilde{p}\left(y_{[T]}^n|x_{[0:T-1]}^{\backslash n}\right) = \prod_{t\in[T]}\tilde{p}\left(y_t^n|y_{[t-1]}^n, x_{[0:t-1]}^{\backslash n}\right),$$

where $\tilde{p}\left(y_1^n|y_{[0]}^n, x_{[0:0]}^{\backslash n}\right) := \tilde{p}\left(y_1^n|x_{[0]}^{\backslash n}\right)$. The marginal likelihood approximation is then obtained as the mean of the Monte Carlo approximations, and we named this final algorithm SimBa-CL without feedback, see Algorithm 2.

When comparing Algorithm 2 with Algorithm 1, the simplicity of the latter becomes evident. The computational cost is reduced from $\mathcal{O}\left(TN^2P\mathbf{card}\left(\mathcal{X}\right)^2\right)$ to $\mathcal{O}\left(TNP\mathbf{card}\left(\mathcal{X}\right)^2\right)$. As with Algorithm 1, our new SimBa-CL procedure is parallelizable on both N and P.

3.3 KL-bound for SimBa-CL with and without feedback

To evaluate the impact of excluding the simulation feedback from SimBa-CL, and so understand when is worth including the feedback, a natural approach is to compare the two estimates of the marginal likelihood:

$$p\left(y_{[T]}^n\right) = \sum_{x_{[0:T]}^{\backslash n}} p\left(x_{[0:T]}^{\backslash n}\right) \sum_{x_{[0:T]}^n} p\left(x_{[0:T]}^n|x_{[0:T]}^{\backslash n}\right) \prod_{t\in[T]} p\left(y_t^n|x_t^n\right),$$

$$\tilde{p}\left(y_{[T]}^n\right) = \sum_{x_{[0:T]}^{\backslash n}} p\left(x_{[0:T]}^{\backslash n}\right)) \sum_{x_{[0:T]}^n} p\left(x_0^n\right) \prod_{t\in[T]} p\left(x_t^n|x_{t-1},\right) p\left(y_t^n|x_t^n\right).$$

As both $p\left(y_{[T]}^n\right)$ and $\tilde{p}\left(y_{[T]}^n\right)$ are probability distributions on $\mathcal{Y}^T$, a simple way of comparing them is to measure the Kullback–Leibler divergence, which we denote with $\mathbf{KL}\left[p\left(\mathbf{y}_{[T]}^n\right)||\tilde{p}\left(\mathbf{y}_{[T]}^n\right)\right]$. The objective of this section is then to establish an upper bound for the **KL**, demonstrating that under suitable assumptions, in the large-N limit, the "without feedback" approximate marginal likelihood is a good approximation of the true marginal likelihood.

Naturally, we must rely on technical assumptions for theoretical results, which we will strive to explain from an intuitive perspective as much as possible. Our result relies on some assumptions, which we now explain.

Assumption 1 For any $n, \bar{n} \in [N]$ and for any $x_t^{\bar{n}} \in \mathcal{X}$, if $x_{t-1}, \bar{x}_{t-1} \in \mathcal{X}^N$ are such that $x_{t-1}^{\backslash n} = \bar{x}_{t-1}^{\backslash n}$ then:

$$\left|p\left(x_t^{\bar{n}}|x_{t-1}\right) - p\left(x_t^{\bar{n}}|\bar{x}_{t-1}\right)\right| \leq \frac{1}{N}\left|d_{n,\bar{n}}\left(x_{t-1}^n\right) - d_{n,\bar{n}}\left(\bar{x}_{t-1}^n\right)\right|,$$

where $d_{n,\bar{n}} : \mathcal{X} \to \mathbb{R}_+$.

Assumption 1 ensures the boundness of the transition dynamic when altering the states of only $n \in [N]$ at time $t-1$. This essentially asserts that changing the state of a single component at time $t-1$ minimally impacts the dynamics of the other components. In essence, the impact is measured in terms of the function $d_{n,\bar{n}}$, which shows how changes in n affect any other dimension $\bar{n}$. This concept is similar in flavour to the decay of correlation property explained in Rebeschini and Van Handel (2015) and Rimella and Whiteley (2022), which ensures a weak sensitivity of the conditional distributions on any n given a perturbation on any other dimension. Conceptually, the main emphasis is that the dynamics are only impacted by single dimension-level perturbations at the scale N^{-1}. In systems of increasing size with interconnected dimensions, we expect that single dimension-level changes have diminishing effects on the dimension as a whole, which is also intuitively plausible in numerous real-world applications, like individual-based models for epidemiology (Rimella et al. 2023b; Cocker et al. 2023). Observe that we could redefine $d_{n,\bar{n}}$ as $d_{n,\bar{n}}/N$ and essentially mask the N^{-1} scaling. However, we made the N^{-1} dependence explicit. Our theoretical results will depend on the variability of $d_{n,\bar{n}}$, and will give $O(N^{-1})$ error bounds for models where the $d_{n,\bar{n}}$ functions have variance that is bounded as N increases.

Assumption 2 For any $n, \bar{n} \in [N]$, if $x_{t-1}, \bar{x}_{t-1} \in \mathcal{X}^N$ are such that $x_{t-1}^{\backslash \bar{n}} = \bar{x}_{t-1}^{\backslash \bar{n}}$ then there exists $0 < \epsilon < 1$ such that:

$$\sum_{x_t^n} p\left(x_t^n|x_{t-1}\right)\frac{1}{p\left(x_t^n|\bar{x}_{t-1}\right)^2} \leq \frac{1}{\epsilon^2}, \quad \text{and}$$

$$\sum_{x_t^n} p\left(x_t^n|x_{t-1}\right)\frac{1}{p\left(x_t^n|\bar{x}_{t-1}\right)^3} \leq \frac{1}{\epsilon^3}.$$

Assumption 2 states that given two initial states $x_{t-1}, \bar{x}_{t-1}$ which differ only in their nth element, any transition which is possible under x_{t-1} is not too unlikely under $\bar{x}_{t-1}$. Note that this assumption is not ruling out absorbing states, but rather ensures that the non-zero elements of $p\left(x_t^n | x_{t-1}\right)$ will stay not-zero in $p\left(x_t^n | \bar{x}_{t-1}\right)$.

Under assumptions 1 and 2 and the additional assumption that the effect of an interaction on a single dimension does not exceed N, we

Theorem 1 *If* $\left|d_{n,\bar{n}}\left(x^n\right) - d_{n,\bar{n}}\left(\bar{x}^n\right)\right| < N$ *for any* $x^n, \bar{x}^n \in \mathcal{X}$ *and assumptions 1, 2 hold, then for any* $n \in [N]$*:*

$$\mathbf{KL}\left[p\left(\boldsymbol{y}_{[T]}^n\right) || \tilde{p}\left(\boldsymbol{y}_{[T]}^n\right)\right] \leq \frac{a(\epsilon)}{N} \sum_{t\in[T]} \mathbb{E}\left\{\frac{1}{N} \sum_{\bar{n}\in[N], \bar{n}\neq n} \mathbb{V}ar\left[d_{n,\bar{n}}\left(\boldsymbol{x}_{t-1}^n\right) \middle| \boldsymbol{x}_{[0:t-1]}^{\setminus n}\right]\right\},$$

where $a(\epsilon) := 2\left[\frac{1}{2\epsilon^2} + \frac{1}{3\epsilon^3}\right]$.

Proof The proof requires the Data Processing inequality, a Taylor expansion of the function $f(z) = \log(1+z)$, and Jensen's inequality. The full proof is available in Section A.2 of the supplementary material. □

From Theorem 1 we can observe that the approximation improves when: (i) N increases; and (ii) the expected variance of the interaction term across dimensions decreases. Hence, our SimBa-CL without feedback will be more or less the same as the SimBa-CL with feedback whenever we are considering a sufficiently large N and when changing the state on one dimension does not affect the other dimensions too much.

3.4 SimBa-CL on general partitions

Up until now, we have implicitly assumed that approximating $p\left(y_{[T]}\right)$ involves a product of marginals across all dimensions. However, it is worth considering that a complete factorization across $[N]$ might not be the optimal choice. For example, when considering an epidemic model with households, it may be better to factorize over households rather than individuals, due to the strong dependencies within each household.

Consider a partition $\mathcal{K}$ on $[N]$ and reformulate our likelihood approximation as follows:

$$p\left(y_{[T]}\right) \approx \prod_{K\in\mathcal{K}} p\left(y_{[T]}^K\right), \quad \text{or} \quad p\left(y_{[T]}\right) \approx \prod_{K\in\mathcal{K}} \tilde{p}_{\mathcal{K}}\left(y_{[T]}^K\right),$$

where on the top we have the actual product of the true marginals and on the bottom $\tilde{p}_{\mathcal{K}}$ denotes the generalization of $\tilde{p}$. As for SimBa-CL with and without feedback we can reformulate our marginals and approximate marginals as a simulation from the model followed by an HMM likelihood:

$$p\left(y_{[T]}^K\right) := \sum_{x_{[0:T]}^{\setminus K}} p\left(x_{[0:T]}^{\setminus K}\right) \sum_{x_{[0:T]}^K} p\left(x_{[0:T]}^K | x_{[0:T]}^{\setminus K}\right) \prod_{t\in[T]} \prod_{n\in K} p\left(y_t^n | x_t^n\right), \tag{16}$$

which corrects according to the interactions inside K, and

$$\tilde{p}_{\mathcal{K}}\left(y_{[T]}^K\right) := \sum_{x_{[0:T]}^{\setminus K}} p\left(x_{[0:T]}^{\setminus K}\right) \sum_{x_{[0:T]}^K} \prod_{n\in K} p\left(x_0^n\right) \prod_{t\in[T]} p\left(x_t^n | x_{t-1}\right) p\left(y_t^n | x_t^n\right), \tag{17}$$

which does not use any feedback.

As seen in Sects. 3.1 and 3.2, (16) aims to approximate the true marginals over $K \in \mathcal{K}$, while (17) only offers approximations. Once again, akin to SimBa-CL with feedback, if we want to approximate the true marginals of the likelihood we need some form of simulation feedback. This time, the simulation feedback will be from $x_{[0:t]}^{\setminus K}$ onto x_{t-1}^K, as we are considering probability distributions on K.

We can then easily adapt Algorithms 1 and 2, transitioning from a full factorization to a factorization on the partition $\mathcal{K}$. The algorithm requires operations on the space $\mathcal{X}^K$ and so a computational cost that is exponential in the maximum number of components contained in K. More details and theoretical results are available in Section A.3 of the supplementary material.

We refer to Table 1 for a comparison of computational costs for a single time step. The final computational cost of each algorithm is then obtained by multiplying the values in Table 1 by T. SMC is used as the computational baseline. We observe that SMC is more expensive to run compared to SimBa-CL whenever the time required to compute the proposal distribution exceeds the time required to run the recursion (with or without feedback) in SimBa-CL. When considering SimBa-CL on partitions, the computational cost becomes linear in the number of partition elements, but exponential in the size of the largest element of the partition (in the worst-case scenario). This is because the dimensions within each element of the partition are now considered jointly. When including the feedback in SimBa-CL on partitions, we must iterate over all the elements external to the partition, i.e. $[N] \setminus K$, requiring $N - \max_{K\in\mathcal{K}} \mathbf{card}(K)$ (in the worst-case scenario).

Table 1 Computational cost for a single time step. We denote with $\mathcal{C}_{\mathcal{M}}(1)$ the cost of simulating from a single dimension of the model. For the SMC, $\mathcal{C}_q(1)$ is the cost of computing the proposal distribution q (see Rimella et al. (2023a) for an example). For SimBa-CL, we use "WF" to indicate "with feedback", "WOF" to indicate "without feedback", and $K_{max} := \max_{K \in \mathcal{K}} \mathbf{card}(K)$

Method	Computational cost
Simulation	$\mathcal{O}(N\mathcal{C}_{\mathcal{M}}(1))$
SMC	$\mathcal{O}(N\mathcal{C}_{\mathcal{M}}(P) + N\mathcal{C}_q(P))$
SimBa-CL WF	$\mathcal{O}(N\mathcal{C}_{\mathcal{M}}(P) + N^2 P\mathbf{card}(\mathcal{X})^2)$
SimBa-CL WOF	$\mathcal{O}(N\mathcal{C}_{\mathcal{M}}(P) + NP\mathbf{card}(\mathcal{X})^2)$
SimBa-CL WF on $\mathcal{K}$	$\mathcal{O}(N\mathcal{C}_{\mathcal{M}}(P) + \mathbf{card}(\mathcal{K})(N - K_{max})P\mathbf{card}(\mathcal{X})^{2K_{max}})$
SimBa-CL WOF on $\mathcal{K}$	$\mathcal{O}(N\mathcal{C}_{\mathcal{M}}(P) + \mathbf{card}(\mathcal{K})P\mathbf{card}(\mathcal{X})^{2K_{max}})$

4 Inference and confidence sets

Composite likelihoods are generally obtained as the product of likelihood components, whose structure is dependent on the considered model, see Varin et al. (2011) for a review of composite likelihood methods. Both SimBa-CL with and without feedback calculate a composite likelihood as the product of the marginals of the likelihood. We can use asymptotic theory for composite likelihood to get approximate confidence regions around the maximum composite likelihood estimator.

The first step is to find the maximum composite likelihood estimator $\hat{\theta}_{CL}$ by maximizing the composite likelihood $\mathcal{L}_{CL}\left(\theta; y_{[T]}\right) = \prod_{K \in \mathcal{K}} \mathcal{L}^K_{CL}\left(\theta; y^K_{[T]}\right)$ or, equivalently, the composite log-likelihood $\ell_{CL}\left(\theta; y_{[T]}\right) = \sum_{K \in \mathcal{K}} \ell^K_{CL}\left(\theta; y^K_{[T]}\right)$, where $\mathcal{L}^K_{CL}\left(\theta; y^K_{[T]}\right)$ and $\ell^K_{CL}\left(\theta; y^K_{[T]}\right)$ depends on the considered SimBa-CL.

The second step is to notice that in the composite likelihood literature, we have some form of asymptotic normality for $\hat{\theta}_{CL}$ (Varin 2008):

$$\hat{\theta}_{CL} \overset{d}{\approx} \mathcal{N}\left(\theta, G(\theta)^{-1}\right),$$

where $G(\theta)$ is the Godambe information matrix (Godambe 1960). It then follows that, upon estimating the Godambe information matrix, we can build confidence sets for $\hat{\theta}_{CL}$ as multidimensional ellipsoids.

The final step is to estimate the Godambe information matrix, which is given by $G(\theta) = S(\theta) V(\theta)^{-1} S(\theta)$, decomposed in terms of the sensitivity matrix and the variability matrix:

$$S(\theta) = \mathbb{E}_\theta\left\{-\mathrm{Hess}_\theta\left[\ell_{CL}\left(\theta; \mathbf{y}_{[T]}\right)\right]\right\} \quad \text{and}$$
$$(\theta) = \mathbb{V}\mathrm{ar}_\theta\left\{\nabla_\theta\left[\ell_{CL}\left(\theta; \mathbf{y}_{[T]}\right)\right]\right\},$$

where Hess_θ and ∇_θ are the Hessian and the gradient with respect to θ. Given that we can compute the Hessian and the gradient given θ, $y_{[T]}$, we can also estimate expectation and the variance via simulations from the model (expected information). More details on this aspect of SimBa-CL along with approximating $S(\theta)$ and $V(\theta)$ with the actual observations (observed information) are discussed in Section B of the supplementary material, with also more experiments available in Section C.

4.1 Bartlett identities

The computation of the Hessian can be resource-intensive and variance estimation might be noisy. We can then simplify the form of the sensitivity and variability matrix by invoking the first and second Bartlett identities. When considering SimBa-CL with feedback the identities hold asymptotically in the number of Monte Carlo samples P, while for SimBa-CL without feedback they hold only approximately. The sensitivity matrix and variability matrix can be then reformulated as:

$$S(\theta) = \sum_{K \in \mathcal{K}} \mathbb{E}_\theta\left[\nabla_\theta \ell^K_{CL}\left(\theta; y^K_{[T]}\right) \nabla_\theta \ell^K_{CL}\left(\theta; y^K_{[T]}\right)^\top\right],$$
$$V(\theta) = \sum_{K, \tilde{K} \in \mathcal{K}} \mathbb{E}_\theta\left[\nabla_\theta \ell^K_{CL}\left(\theta; y^K_{[T]}\right) \nabla_\theta \ell^{\tilde{K}}_{CL}\left(\theta; y^{\tilde{K}}_{[T]}\right)^\top\right],$$

where $=$ becomes $\approx$ if we do not include the feedback, and where both matrices can be once again estimated via simulation. More details are available in Section B.3 of the supplementary material.

5 Limitations and extensions

It is evident from Table 1 that considering SimBa-CL on a partition $\mathcal{K}$ results in a computational cost that is linear in the number of elements in the partition but exponential in the size of the elements within the partition, which quickly becomes infeasible. As shown in Sects. 6.1.2 and 6.1.3, choosing a coarser partition does not affect performance when considering a model with dimensions that exhibit low correlation.

This is the case, for instance, when using an individual-based model in epidemiology, where individuals have similar behaviours, e.g. they are homogeneous mixing (Ju et al. 2021; Rimella et al. 2023a). However, it is common for dimensions to cluster together in terms of correlation. For example, again in an individual-based model for epidemiology, individuals are often assigned to households and have a high likelihood of being infected by others within the same household (Rimella et al. 2023b). In such scenarios, the SimBa-CL approach described in Sect. 3.4 with $\mathcal{K}$ representing the partition induced by the households is preferable, but comes with an increased computational cost. As a solution, one could further partition households into smaller subsets and implement localized feedback to adjust accordingly. This hybrid SimBa-CL would eliminate feedback across elements of the partition while retaining feedback within the elements of the partition, which could potentially correct the approximation.

The factorial structure shown in (2) restricts SimBa-CL to discrete-time models with possibly unevenly timed observations. If we are interested in continuous-time models and the instantaneous dynamics are independent across dimensions given the current state, an Euler approximation of the continuous-time dynamics can be used to construct the discrete-time model. However, if the time between observations is large, such an Euler approximation may not be accurate. In such cases, we can use a sufficiently small discrete-time step to ensure the accuracy of the Euler approximation, with the additional time steps corresponding to points where no observations are available. This approach results in several prediction steps without correction, followed by a correction step when data are observed.

Unconditional simulation from the model has the benefit of making SimBa-CL almost as computationally efficient as simulating directly from the model. Indeed, after the simulation phase, only simple marginalizations on $\mathcal{X}$ are required, which can be even parallelized across the N dimensions. However, the simulation procedure does not incorporate information from the data, which makes it possible for SimBa-CL to generate realizations of $\mathbf{x}_{[0:t]}$ that are inconsistent with the data and might affect its performance via the transition kernel. Fortunately, Assumption 2 guarantees that all the non-zero elements of the transition kernel remain non-zero, ruling out the possibility of SimBa-CL collapsing to zero likelihood estimates. That said, mismatches can still occur if there is significant uncertainty about the trajectory of the latent process. However, increasing the number of simulations P reduces the probability of such mismatches. Note that we do not imply that there are no advantages to incorporating data information during the simulation process. On the contrary, the closer the simulations are to the actual hidden trajectories, the better the performance (Whiteley and Lee 2014; Rimella et al. 2023a).

One option, for example, could be to simulate from $p\left(x_t^n | y_{[t]}^n, x_{[0:t]}^{\setminus n}, \theta\right)$ for SimBa-CL with feedback, or from $\tilde{p}\left(x_t^n | y_{[t]}^n, x_{[0:t-1]}^{\setminus n}, \theta\right)$ for SimBa-CL without feedback. Including the observations should make the simulations more consistent with the data, possibly reducing both the bias and the variance of SimBa-CL. From a theoretical perspective, this could even result in a better bound compared to the one from Theorem 1, which currently increases linearly with time. We provide a small experiment on this "conditional" SimBa-CL in Section C.6 of the supplementary material.

Alternatively, one could even attempt to avoid simulations from the model altogether by performing the prediction step with the modified transition kernel $\prod_{n\in[N]} p\left(x_t^n | x_{t-1}^n, \mathbb{E}_{\rho^{\setminus n}}\left[\mathbf{x}_{t-1}^{\setminus n}\right], \theta\right)$, i.e. substituting the contribution from $x_{t-1}^{\setminus n}$ with its expected contribution under a distribution $\rho^{\setminus n}$, which could be either unconditional or conditional on the data. This approach would still exploit the factorial structure (2), and the computational cost could be further reduced whenever closed-form solutions are available (Rimella et al. 2025).

It can be observed from Sects. 3.1 and 3.2 that both SimBa-CL with and without feedback require simple operations that combine the initial distribution, transition kernel, and emission distribution. If these three quantities are continuous with respect to the parameters θ, then, given the Monte Carlo simulations, it is possible to compute gradients via automatic differentiation. This process is readily implemented in multiple Machine Learning libraries and can be combined with any gradient descent technique to optimize the parameters (Zeiler 2012; Kingma et al. 2014). However, the Monte Carlo simulation itself depends on the parameters θ, which complicates the differentiation process. To be more precise, we consider quantities of the form $\sum_{n\in[N]} \log\left(p(y_{[T]}^n|\theta)\right)$, with $\tilde{p}$ for SimBa-CL without feedback, meaning that we need to compute:

$$\frac{\partial\, p(y_{[T]}^n|\theta)}{\partial\theta} = \frac{\partial}{\partial\theta}\mathbb{E}_{\mathcal{M}}\left[p\left(y_{[T]}^n|\mathbf{x}_{[0:T-1]}^{\setminus n}, \theta\right)\right],$$

where $\mathcal{M}$ stands for simulating from the model. In our experiments we consider the approximation:

$$\frac{\partial\, p(y_{[T]}^n|\theta)}{\partial\theta} \approx \mathbb{E}_{\mathcal{M}}\left[\frac{\partial}{\partial\theta} p\left(y_{[T]}^n|\mathbf{x}_{[0:T-1]}^{\setminus n}, \theta\right)\right].$$

which computes derivatives by conditioning on the results of the Monte Carlo simulation, and essentially approximates the derivative of an expectation with the expectation of a derivative. On the one hand, this approximation reduces computational complexity by ignoring part of the derivative. On the other hand, it produces biased estimates of the gradient.

As we require simulation from a finite state-space and therefore simulation from categorical random variables, we could alternatively use a continuous relaxation of categorical random variables known as the Gumbel-Softmax trick (Maddison et al. 2016; Jang et al. 2016). This technique substitutes the arg max of i.i.d. Gumbel random variables (Gumbel 1954) with a Softmax function. In this way, sampling from a categorical distribution is expressed as a continuous function of i.i.d. random variables and suitable to the reparameterization trick (Blundell et al. 2015), which allows to exchange expectations with derivatives without any approximation. In our case we get:

$$\frac{\partial\, p(y^n_{[T]}|\theta)}{\partial\theta} \approx \mathbb{E}_{\mathcal{G}}\left[\frac{\partial}{\partial\theta}\, p\left(y^n_{[T]}|s(\mathbf{G},\theta,\tau),\theta\right)\right],$$

where $\mathbf{G}$ is a collection of i.i.d. Gumbel random variables with distribution $\mathcal{G}$, and s is a composition of continuous functions involving the Gumbel random variables, the parameter θ, and the temperature parameter τ.

Even though this approximation produces unbiased gradient estimates as $\tau \to 0$, it is computationally heavier and challenging to use when considering large P and when computing confidence regions. We refer to Section C.6 of the supplementary material for a detailed discussion and additional experiments.

6 Experiments

The experiments centre on the field of epidemiological modelling, and specifically they focus on individual-based models (IBMs). Individual-based models arise when we want to describe an epidemic from an individual perspective. The complexity of IBMs lies in their high-dimensional state-space, making closed-form likelihood computationally unfeasible. However, these models satisfy the factorization outlined in (2), making them the perfect candidates for our SimBa-CL.

SimBa-CL is implemented in Python using the library TensorFlow and is available at the GitHub repository: https://github.com/LorenzoRimella/SimBa-CL. All the experiments were run on a 32gb Tesla V100 GPU available on "The High End Computing" (HEC) facility at Lancaster University. For gradient computation we use TensorFlow's default automatic differentiation, see Sect. 5 for a deeper discussion.

In Section C of the supplementary material, interested readers can find more details about the experiments reported in the paper, as well as additional experiments on the following topics: a spatial SIS individual-based model; a comparison between SimBa-CL and its extensions; the role of P in terms of the quality of the approximation.

6.1 The effect of feedback and partition's choice on SimBa-CL

In this section we perform a cross-comparison on a susceptible-infected-susceptible (SIS) IBM on SimBa-CL with feedback on $\mathcal{K} = \{\{1\}, \ldots, \{N\}\}$ ("fully factorized SimBa-CL with feedback"), SimBa-CL without feedback on $\mathcal{K} = \{\{1\}, \ldots, \{N\}\}$ ("fully factorized SimBa-CL without feedback"), and SimBa-CL without feedback on $\mathcal{K} = \{\{1, 2\}, \ldots, \{N-1, N\}\}$ ("coupled SimBa-CL without feedback"), where N is assumed to be even.

6.1.1 Model

Building upon the framework introduced by Ju et al. (2021) and Rimella et al. (2023a), where n represents an individual and w_n is an observed vector of covariates. We consider a $p\left(x^n_0|\theta\right)$ with an initial probability of infection of $1/1+\exp\left(-\beta_0^\top w_n\right)$ and a transition kernel $p\left(x^n_t|x_{t-1},\theta\right)$ with a probability of transitioning from S to I of $1-\exp\left[-\lambda_n\left(\sum_{n\in[N]}\mathbb{I}\left(x^n_{t-1}=2\right)/N+\iota\right)\right]$ and a probability of transitioning from I to S of $1-\exp\left(-\gamma_n\right)$, where $\lambda_n = 1/(1+\exp\left(-\beta_\lambda^\top w_n\right))$ and $\gamma_n = 1/(1+\exp\left(-\beta_\gamma^\top w_n\right))$ and with $w_n, \beta_0, \beta_\lambda, \beta_\gamma \in \mathbb{R}^2$. Moreover we consider $p\left(y^n_t|x^n_t,\theta\right) = q^{x^n_t}\mathbb{I}\left(y^n_t\neq 0\right) + \left(1-q^{x^n_t}\right)\mathbb{I}\left(y^n_t=0\right)$, with $q \in [0,1]^2$. Unless specify otherwise, our baseline model employs $N = 1000$, $T = 100$, w_n to be such that $w^1_n = 1$ and $w^2_n \sim$ **Normal** $(0,1)$, and the data-generating parameters $\beta_0 = [-\log\left((1/0.01)-1\right), 0]^\top$, $\beta_\lambda = [-1,2]^\top$, $\beta_\gamma = [-1,-1]^\top$, $q = [0.6, 0.4]^\top$ and $\iota = 0.001$. It is also important to mention that the considered SIS satisfies Assumption 1 and Assumption 2, see Section A.1 of the supplementary material.

6.1.2 Empirical evaluation of the Kullback–Leibler divergence

We start by comparing our SimBa-CL methods in terms of empirical Kullback–Leibler divergence (KL) on a set of simulated data. Different settings are considered: an increasing population size $N = [10, 100, 1000]$; either "high" $\beta_0 = [-6.9, 0]^\top$ or "base" $\beta_0 = [-4.60, 0]^\top$ or "low" $\beta_0 = [-2.20, 0]^\top$, i.e. around either 0.1% or 1% or 10% of initial infected; and "base" and "low" $\iota = [0.001, 0.01]$. Note that different β_0 and ι control the variance of the process, as having more infected at the beginning of the epidemic or including more environmental effects results in an epidemic that is closer to the equilibrium. We set $P = 1024$ for all scenarios and for both SimBa-CL with feedback, SimBa-CL without feedback and coupled SimBa-CL without feedback. All the versions of SimBa-CL were run under the data-generating parameter.

Table 2 Comparing empirical KL between SimBa-CL and under different scenarios. All the numerical values have been multiplied by 10^9 to improve visualization

N	1000	100	100	100	100	10
β_0	Base	High	Low	Base	Base	Base
ι	Base	Base	Base	Base	Low	Base
KL on feedback	0.3 (0.2)	7.5 (3.2)	0.8 (0.4)	5.9 (2.9)	1.4 (1)	49.6 (17.2)
KL on partition	1.4 (0.3)	36.1 (6.6)	3 (0.9)	35.1 (7)	5.6 (2)	177.3 (36.5)

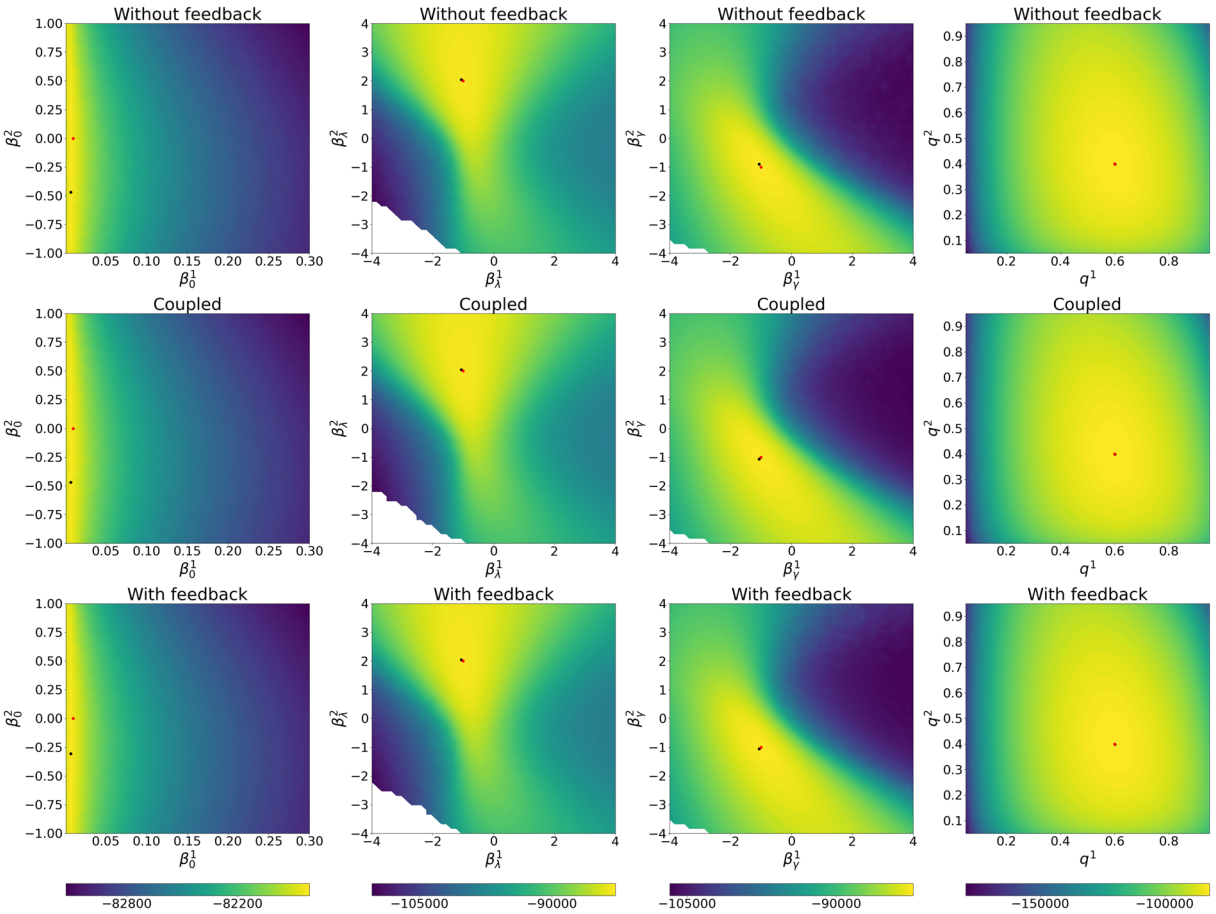

Fig. 2 Profile log-likelihood surfaces for $\beta_0, \beta_\lambda, \beta_\gamma, q$. From top to bottom, fully factorized SimBa-CL without feedback, coupled SimBa-CL without feedback, and fully factorized SimBa-CL with feedback. Dots locate the data-generating parameter and the maximum on the grid

Table 2 reports the mean and standard deviations of the empirical KL per each scenario. Focusing on the fourth row, which contrasts the fully factorized SimBa-CL with and without feedback, we can notice that: increasing N decreases the KL, and decreasing the variance decreases the KL. Similar conclusions can be drawn for the fifth row, which compares the fully factorized SimBa-CL without feedback with the coupled SimBa-CL without feedback. These comments suggest less and less differences across the methods when increasing N and decreasing the variance, which is in line with our theoretical results.

6.1.3 Comparing likelihood surfaces

We proceed to undertake a comparison of profile likelihood surfaces for the baseline SIS IBM using the following protocol: (i) choose one among β_0, β_λ, β_γ and q; (ii) simulate using the baseline model, and ensure at least 10 infected in the epidemic realization; (iii) create a bi-dimensional grid on the chosen parameter; (iv) per each element of the grid compute our SimBa-CL methods by fixing the other parameters to their real values. As for the previous section, we set $P = 1024$ when computing all surfaces and for both SimBa-CL with feedback, SimBa-CL without feedback and coupled SimBa-CL without feedback.

The outcomes of this experiment are illustrated in Fig. 2. Interestingly, all the considered SimBa-CL exhibit a consistent shape, meaning that, in the SIS scenario, including the feedback or choosing a coarser partition has a limited impact on the overall likelihood. Furthermore, it becomes evident that all these methods effectively recover the data-generating parameter, except for β_0. Here there is an obvious identifiability issue with β_0 as $\beta_0^2 = 0$ implies covariate w_n^2 to not be used. However, given that w_n^2 is random, we could have more initial infection associated with $w_n^2 < 0$, which gives a higher likelihood to models where $\beta_0^2 < 0$. Similar reasoning can be replicated for $w_n^2 > 0$, which explains the symmetry of the likelihood surface of β_0 on the vertical axis.

6.2 Asymptotic properties of SimBa-CL

We can now turn to the problem of computing the maximum composite likelihood estimator and the corresponding confidence sets. Remark that for a sufficiently "regular" model and a sufficiently large N, including the simulation feedback and using a coarser partition bears marginal significance for SimBa-CL. We hence narrow our studies to the asymptotic properties of the fully factorized SimBa-CL without feedback. As a toy model, we consider again the SIS IBM described in Sect. 6.1.1.

Crucially, it is worth emphasizing once more that gradients and Hessians can be computed through automatic differentiation as we have implemented SimBa-CL using TensorFlow, see Sect. 5. This computational capability grants us access to the estimates of the variability and sensitivity matrix, as from Sect. 4, thereby avoiding any manual derivation of derivatives.

Remark that in both Sects. 6.2.1 and 6.2.2, we set $P = 500$ for parameter optimization and $P = 100$ for computing confidence regions. These values of P were selected to avoid out-of-memory errors.

6.2.1 Maximum Simba-CL convergence and coverage in two dimensions

We start our exploration by looking at the bi-dimensional parameter β_λ, given all the other parameters fixed to their baseline values. The hope is that β_λ is easily identifiable as it highly influences the evolution of the epidemics, and so we can test the asymptotic properties on a well-behaved parameter.

To explore the asymptotics of SimBa-CL we investigate four scenarios with an increasing amount of data: (i) $N = 100, T = 100$; (ii) $N = 100, T = 300$; (iii) $N = 1000, T = 100$; (iv) $N = 1000, T = 300$. Per each scenario, we simulate 100 epidemics and per each dataset we optimize β_λ through Adam optimization (Kingma et al. 2014), aiming to minimize the negative log-likelihood. After optimization, we have a sample of 100 bi-dimensional parameters per each scenario, which can be turned into box plots as shown in Fig. 3. As both T and N increase, we can observe an evident shrinkage towards the true parameter, suggesting consistency of the maximum SimBa-CL estimator when N and T increase.

Taking the investigation a step further, we analyse the empirical coverage of confidence sets built as explained in Sect. 4. We consider $N = 1000$, $T = 300$, and the optimized parameters from the previous experiment. We start by calculating the Godambe information matrix without using the Bartlett identities, and we build 95% 2-dimensional confidence sets for our parameter β_λ. The procedure results in a coverage of 1 and so an overestimation of uncertainty. However, when repeating the same procedure using the Bartlett identities, the coverage now aligns with the theoretical coverage of 0.95. This favourable outcome can be attributed to less noisy estimates, as we are exploiting the factorization in the model and so computing expectations on lower dimensional spaces, see Section C.3 of the supplementary materials.

6.2.2 Maximum Simba-CL convergence and coverage in nine dimensions

Transitioning to a substantially more intricate scenario, we now estimate all the parameters of the model $\theta = (\beta_0, \beta_\lambda, \beta_\gamma, q, \iota)$. Analogous to the 2-dimensional case, we simulate from the model 100 times, and per each simulation, we optimize the parameters using Adam optimizer. The outcomes are reported in Fig. 4.

Foremost, it becomes apparent that increasing the value of T does not influence β_0. This arises because observations in the later periods carry scarce information on the initial condition. Moreover, recall that $w_n^2 \sim \textbf{Normal}(0, 1)$ and $\beta_0^2 = 0$. This makes the parameter not identifiable, as commented in Sect. 6.1. This ill-posed model definition implies that increasing N will not improve the uncertainty around our estimate as $\beta_0^2 < 0$ and $\beta_0^2 > 0$ can be equally likely, while the unbiasedness is preserved due to the symmetry of the set of equally likely parameters. At the same time, the parameters ι and β_λ are also hard to identify as smaller (or bigger) estimates of β_λ will lead to bigger (or smaller) estimates of ι. Indeed, β_λ governs the infection rates from the community, while ι represents the environmental effect. It is then clear that generating an epidemic from β_λ, ι is equivalent to generating one by decreasing β_λ and increasing accordingly ι. This correlation is especially vivid in Fig. 4, as an overestimation of ι (log-scale) leads to an underestimation of β_λ.

Clearly, in a 9-dimensional scenario, the process of recovering empirically the theoretical coverage is substantially more complicated. Jointly, we find 0 coverage of the 9-dimensional 95% confidence sets, irrespective of whether the Bartlett identities are employed or not. We then compute

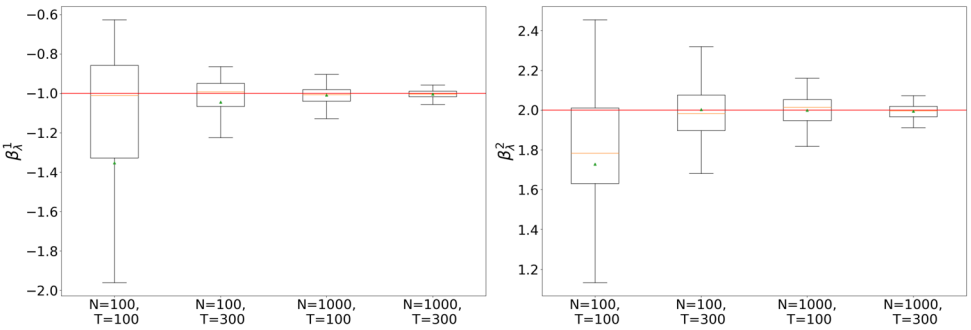

Fig. 3 Box plots on the optimized β_λ. On the left, β_λ^1, on the right, β_λ^2. Horizontal solid lines within the boxes show the medians and triangles are used for the means. Horizontal solid lines show the true parameters

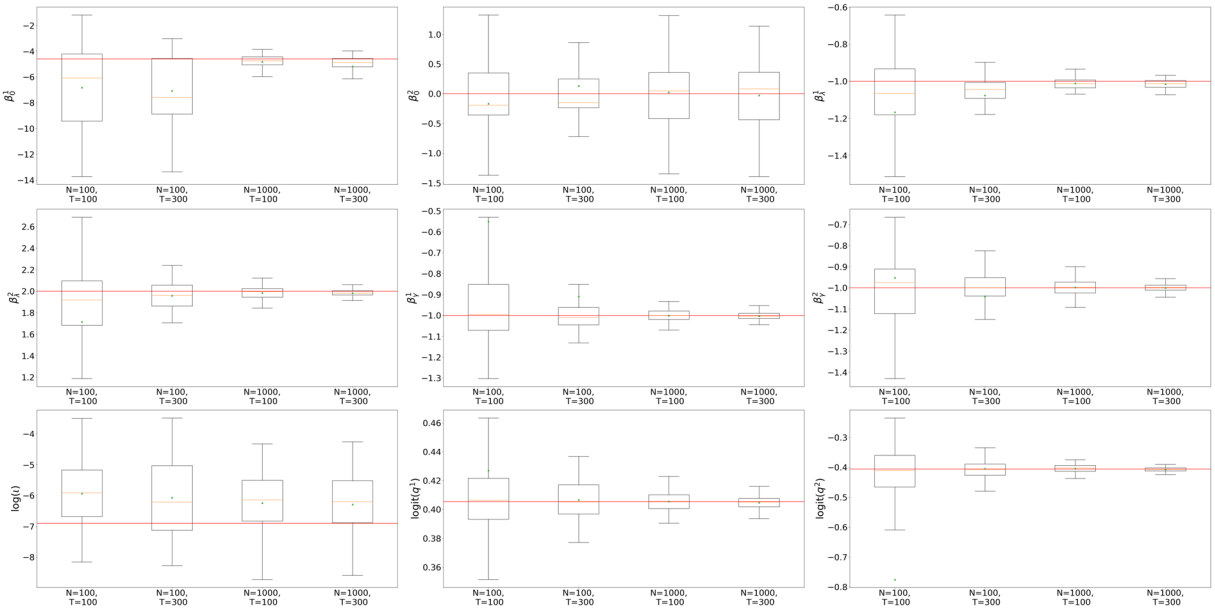

Fig. 4 Box plots on the optimized $\theta = (\beta_0, \beta_\lambda, \beta_\gamma, q, \iota)$. Parameter labels are reported on the y-axis. Horizontal solid lines within the boxes show the medians and triangles are used for the means. Horizontal solid lines show the true parameters

Table 3 Empirical coverage per each parameter when computing the Godambe information matrix with and without the approximate Bartlett identities. Whenever the parameter is bi-dimensional the coverage per each component is reported in the same cell separated by "and"

Parameter	β_0	β_λ	β_γ	q	ι
Without Bartlett	0.17 and 0.05	0.61 and 0.87	0.8 and 1.	0.87 and 0.5	0.02
With Bartlett	0.98 and 0.89	0.99 and 0.75	0.97 and 0.97	1. and 0.98	0.92

the confidence intervals on single parameters by marginalizing the 9-dimensional Gaussian distribution. Marginally, we find that using the approximate Bartlett identities improves the coverages, see Table 3 for numerical values, with comments similar to the previous section for the quality of the estimates.

6.3 Comparing SimBa-CL with sequential Monte Carlo

As SimBa-CL methods provide biased estimates of the likelihood, the objective of this section is to compare SimBa-CL methods with both sequential Monte Carlo (SMC) and block sequential Monte Carlo (BSMC) algorithms. While SMC provides an unbiased particle estimate of the likelihood (Chopin et al. 2020), this quantity can suffer high variance in high-dimensional scenarios. On the other hand, BSMC deals with the curse of dimensionality by providing a factorized,

albeit biased, particle estimate of the likelihood (Rebeschini and Van Handel 2015).

Building upon the previous sections, we compare SMC and BSMC with fully factorized SimBa-CL without feedback. Regarding the SMC comparison, we consider two approaches: the auxiliary particle filter (APF) and the SMC with proposal distribution given by Rimella et al. (2023a). Due to the curse of dimensionality, we expect poor performances of the APF, hence we include the Block APF in our analysis. The block APF works as the Block particle filter (Rebeschini and Van Handel 2015), a BSMC algorithm, but it proposes particles according to the transition kernel informed by the current observation.

6.3.1 Model

In the subsequent sections, we work again on the SIS IBM from Sect. 6.1.1. However, we also analyse a susceptible-exposed-infected-removed (SEIR) IBM. Specifically, we still have some bi-dimensional covariates w_n, while $p(x_0^n|\theta)$ is now 4-dimensional with the second and the fourth components being zero and the first and the third components being as in SIS IBM. Similarly, the transition kernel $p(x_t^n|x_{t-1}, \theta)$ is a 4 by 4 matrix with the same dynamics of SIS IBM when considering transitions from S to E and from I to R, with the addition of a transition E to I with probability $1 - \exp(-\rho)$. Unless specified otherwise, we consider as the baseline model the one with $N = 1000$, $T = 100$ and the data-generating parameters set to $\beta_0 = [-\log((1/0.01) - 1), 0]^\top$, $\beta_\lambda = [-1, 2]^\top$, $\rho = 0.2$, $\beta_\gamma = [-1, -1]^\top$, $q = [0, 0, 0.6, 0.4]^\top$. Observe, that the first two elements of q are both zero, meaning that we do not observe any susceptible and exposed.

6.3.2 SimBa-CL and SMC for an individual-based SIS model

We consider the baseline SIS model with $N = 1000$, and precisely: we generate the data; we run SimBa-CL and the baseline algorithms 100 times on the given data using the data-generating parameters; and we estimate the mean and standard deviation of the log-likelihood. The results are reported in Table 4.

Notably, the method proposed by Rimella et al. (2023a) emerges as the best method in terms of log-likelihood mean and variance, as it yields unbiased estimates of the likelihood and reduces the variance. Our SimBa-CL exhibits superior computational efficiency, with a running time that is also almost three times faster than vanilla Block APF. It is worth noting that even though the bias of SimBa-CL is significant the log-likelihood variance is considerably lower than vanilla APF and comparable with the Block APF. It can be observed that the Block APF run out of memory when increasing the number of particles. This is attributed to the necessity of maintaining P particles for each individual and, at the same time, performing a multinomial resampling for each of these particles.

We also run a paired comparison on the profile likelihood surfaces as in Sect. 6.1, and report them in Fig. 5. It can be noticed that both SimBa-CL and Block APF generate similar likelihood surfaces whose maxima are close to the data-generating parameter. Both P and the number of particles were set to 1024. Unfortunately, we could not include in our studies the SMC from Rimella et al. (2023a) for computational reasons.

6.3.3 SimBa-CL and SMC for an individual-based SEIR model

The section concludes with a comparison of the baseline SEIR IBM. As for the SIS model, we simulate synthetic data, we run our SimBa-CL along with the baseline algorithms using the data-generating parameter, and we estimate the mean and variance of the resulting log-likelihood computations. The outcomes are reported in Table 5.

Note that the SEIR scenario is considerably more complex than the SIS scenario. In the SEIR case, observing only infected and removed makes it difficult for the SMC algorithms to prevent particle failure, without the use of very informative proposal distributions. The underlying intuition is that if we propose a susceptible individual at time t and then we observe that individual to be infected, or removed, at time $t + 1$, it becomes impossible to recover from our incorrect proposal at time t.

Table 5 clearly shows that, to avoid failure of the SMC, we need a smart proposal distribution as the one proposed by Rimella et al. (2023a), and also a large h to reach a reasonable log-likelihood variance. On the other hand, our SimBa-CL is able to reach comparable log-likelihood variance almost ten times faster than the SMC.

6.4 2001 UK foot and mouth disease outbreak

In the year 2001, the United Kingdom experienced an outbreak of foot and mouth disease, a highly contagious virus affecting cloven-hoofed animals. Over an 8-month period, 2026 farms out of 188361 in the UK were infected, concentrated in the North and South West of England, and costing an estimated £8 billion to public and private sectors (UK National Audit Office 2022).

Data on the outbreak are covered by copyright, see Defra (Department for environment, food and rural affairs) website (http://www.defra.gov.uk). From the full dataset we extracted: farm coordinates, farm notification status with date, number of cattle and number of sheep in the farm. We also localised our studies in the surroundings of Cumbria and selected a total of 8791 farms, which integrates the studies by Jewell et al. (2009), see Fig. 6 for a graphical representation.

Table 4 Log-likelihood means and log-likelihood standard deviations for the baseline SIS model with $N = 1000$. h is the number of future observations included in Rimella et al. (2023a) ($h = 0$ correspond to APF)

P	512	1024	2048	Time (s)
APF	−81103.37 (46.04)	−81046.57 (49.65)	−80976.34 (36.08)	1.05
h = 5	−79551.92 (1.79)	−79552.24 (1.6)	−79552.81 (1.57)	3.78
h = 10	−79551.9 (1.81)	−79552.22 (1.47)	−79553.01 (1.56)	5.61
Block APF	−79565.69 (5.84)	−79558.44 (3.95)	Out of memory	2.97
SimBa-CL	−79612.74 (3.4)	−79612.31 (2.37)	−79612.34 (1.55)	1.03

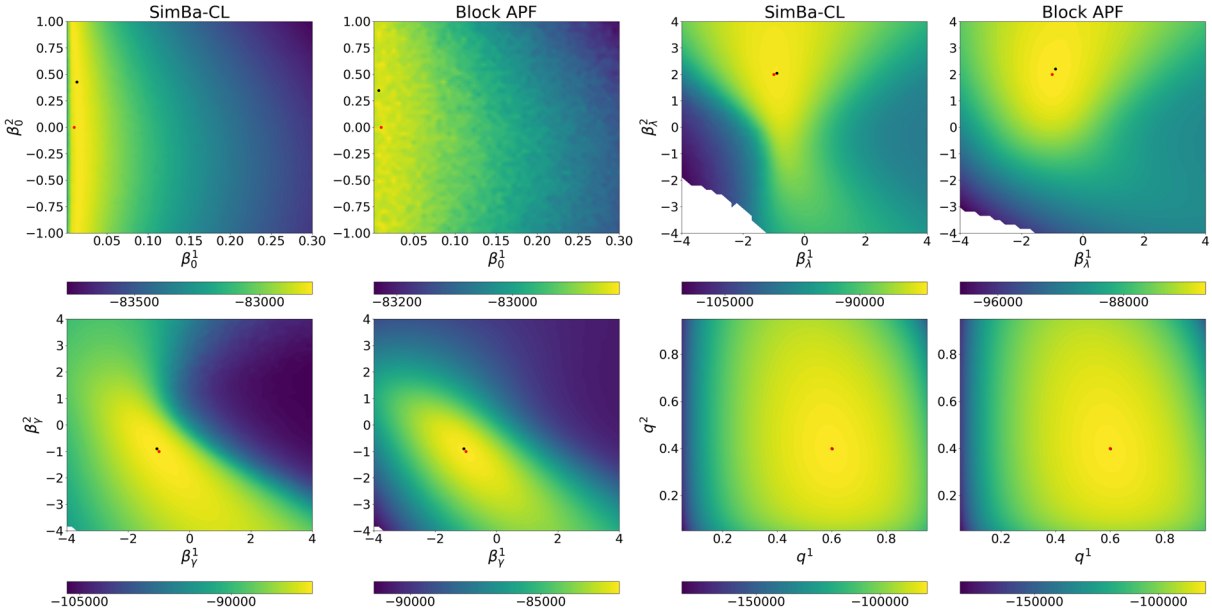

Fig. 5 Profile log-likelihood surfaces for $\beta_0, \beta_\lambda, \beta_\gamma, q$ from fully factorized SimBa-CL without feedback (first and third column) and Block APF (second and the fourth columns). Dots are used for the data-generating parameter and the maximum on the grid

Table 5 Log-likelihood means and log-likelihood standard deviations for the baseline SEIR model with $N = 1000$. h is the number of future observations included in Rimella et al. (2023a) ($h = 0$ correspond to APF)

P	512	1024	2048	Time (s)
APF	Failed	Failed	Failed	1.2
h = 5	−43447.56 (52.04)	−43419.52 (51.08)	−43391.0 (52.41)	4.44
h = 20	−43004.55 (5.38)	−43001.9 (4.65)	−42999.76 (3.7)	11.08
h = 50	−42999.93 (3.44)	−42998.13 (2.72)	−42996.74 (2.39)	20.88
Block APF	Failed	Failed	Failed	2.09
SimBa	−43683.85 (9.54)	−43683.67 (7.35)	−43683.76 (5.16)	1.25

6.4.1 Model

Similar to previous models, we consider an IBM with farms as the individual. We assume farms exist in Susceptible, Infected, Notified (i.e. quarantined on detection), and Removed states (the SINR model). Transitions from S to I and I to N follow a discrete-time stochastic process, with infected farms immediately quarantined, and N to R (farm culling) occurs deterministically after 1 day.

We consider an initial probability of infection for farm n of $1 - \exp\left\{-\tau \sum_{\tilde{n}\in[N]} \lambda_{\tilde{n},n}/N\right\}$, with parameter $\tau > 0$. We assume transition probabilities from I to N of $1 - \exp\{-\gamma\}$, $\gamma > 0$, and from N to R of 1. We assume individual infection probabilities, i.e. transitions from S to I, of $1 - \exp\left\{-\sum_{\tilde{n}\in[N]} \lambda_{\tilde{n},n} \mathbb{I}\left(x_{t-1}^{\tilde{n}} = 2\right)\right\}$, where $\lambda_{\tilde{n},n}$ is the infection pressure exerted by an infected farm $\tilde{n}$ to a susceptible farm n and formulated as:

$$\lambda_{\tilde{n},n} = \frac{\delta}{N}\left[\zeta\left(w_{\tilde{n}}^{c}\right)^{\chi} + \left(w_{\tilde{n}}^{s}\right)^{\chi}\right]\left[\xi\left(w_{n}^{c}\right)^{\chi} + \left(w_{n}^{s}\right)^{\chi}\right] \frac{\psi}{E_{\tilde{n},n}^{2} + \psi^{2}}, \tag{18}$$

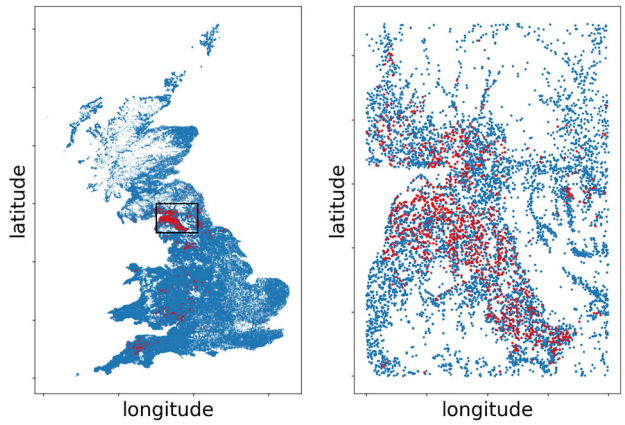

Fig. 6 Graphical representation of the FMD outbreak. Top: the full outbreak in Great Britain. Bottom: the analysed farms. Red dots are used to indicate farms that got infected during the outbreak. The black rectangle encloses the analysed farms in the full plot

with $\delta, \xi, \zeta, \chi, \psi$ positive parameters, w_n^c number of cattle in the n-th farm, w_n^s number of sheep in the n-th farm and $E_{\tilde{n},n}$ the Euclidean distance in kilometres between farm $\tilde{n}$ and farm n. This SINR model is an example of heterogeneous mixing IBM as the infectious contacts are not homogeneous in space. The emission distribution follows the usual formulation $p\left(y_t^n | x_t^n, \theta\right) = q^{x_t^n} \mathbb{I}\left(y_t^n \neq 0\right) + \left(1 - q^{x_t^n}\right) \mathbb{I}\left(y_t^n = 0\right)$, with $q = [0, 0, 1, 0]^\top$ as we observe perfectly the notified.

We set $P = 500$ when performing the optimization and $P = 200$ when computing confidence regions.

6.4.2 Inference

We run 100 optimization using Adam on our fully factorized SimBa-CL without feedback and select the "best" simulation according to its final SimBa-CL score. We then estimate the Godambe information matrix using the approximate Bartlett identities. As we learned the parameters in a log-scale, we need to use log-Normal when plotting the parameters distributions, see Fig. 7.

The parameter τ can be intuitively understood as the time interval between the first infection and the first notified infections, marking the onset of the notification process. In Fig. 7 we can recognize an optimal τ of about 40, suggesting a relatively slow start in notifying farms. Furthermore, we can observe a mean time before notification of about 2.5 days, leading to an estimate of 3.5 days for the mean infection period, encompassing the period from farm infection to culling. This implies a relatively fast intervention once the notification process is implemented. Also, from the last row of Fig. 7, we can notice a decrease of over 60% of the infectivity after just 2 km, which could be used to define containment zones around infected farms.

Parameters ζ, χ, ξ are more difficult to interpret as they regulate the susceptibility and infectivity of farms according to the number of animals. To help visualize their effect we produce Table 6, which shows the average susceptibility and infectivity of a medium-sized farm with only cattle, a medium-sized farm with only sheep, a large-sized farm, and a small-sized farm. From Table 6 we can deduce that the

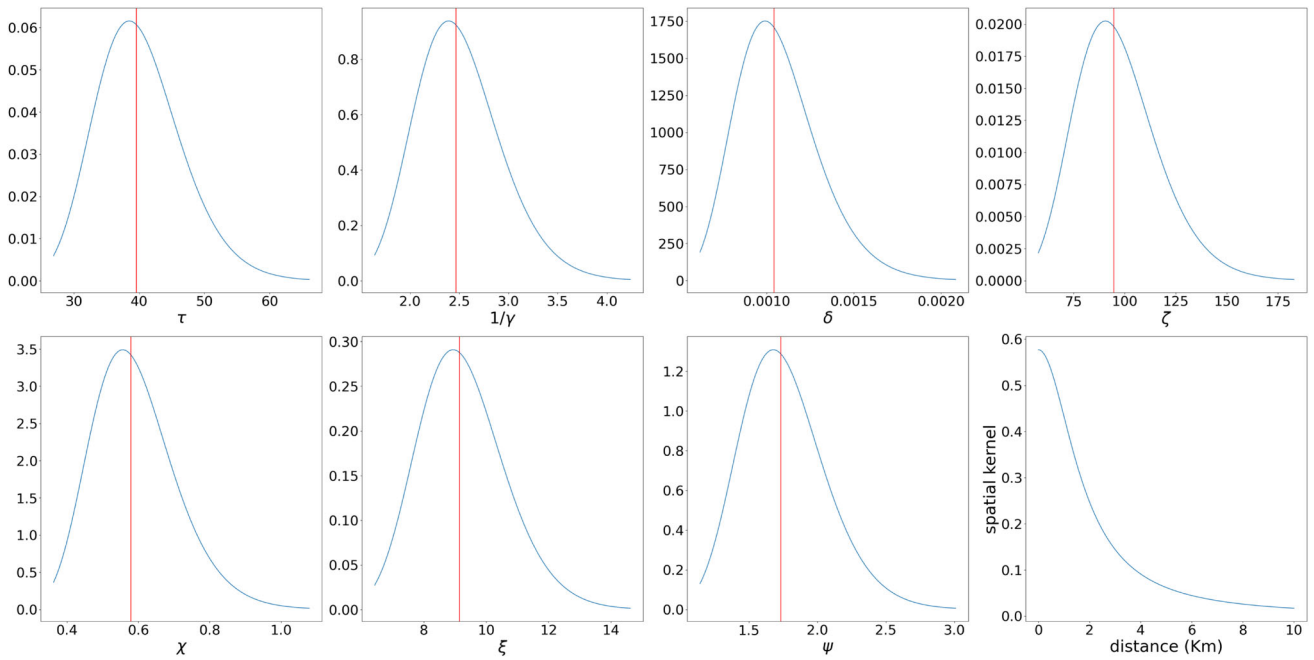

Fig. 7 FMD parameters' distributions and spatial kernel decay. Parameters' labels are reported on the x-axis. Solid lines represent the maximum SimBa-CL estimator. The last plot of the second row shows the spatial decay of infectivity in Km

Table 6 Mean susceptibility and mean infectivity for four farms conformations

Nr. cattle	Nr. sheep	Mean susc	Mean infec
100	0	131.83	1364.99
0	1000	54.69	54.69
50	500	124.84	950.17
2	6	16.49	144.37

effect of owing cattle is significantly higher than the one of owing sheep for both infectivity and susceptibility, and that even small farms can affect the epidemic spread. This agrees with the study from Jewell et al. (2013) and also with the Directive of the Council of the European Union (2003).

7 Discussion

In this work, we propose SimBa-CL, a novel composite likelihood approach for inference in high-dimensional HMM, which relies on simulations from the model and basic forward recursions to compute Monte Carlo approximations of the marginals of the likelihood. The computational cost of the algorithm is quadratic in the dimension N when considering simulations with feedback, targeting the marginals of the likelihood exactly, and can be reduced to linear when ignoring them, albeit by introducing an approximation. For well-mixing models where the effect of changing the state of one individual has a decreasing impact on the dynamics of the other individuals, we provide Kullback–Leibler divergence bounds showing that the effect of the feedback becomes negligible as N increases. We demonstrate that SimBa-CL is competitive with state-of-the-art SMC algorithms in terms of likelihood variance while being significantly faster and suited to automatic differentiation, allowing straightforward optimization of the parameters via Machine Learning libraries.

Although our experiments with SimBa-CL are limited to individual-based models for epidemiology, we have presented the methodology in the context of general HMMs. Indeed, SimBa-CL could be applied to standard compartmental models for epidemics (Chowell et al. 2004; Lekone and Finkenstädt 2006), where individual counts are modelled instead of individual states. Here, conjugacy properties of the model could be exploited to simplify the SimBa-CL filter and feedback (Whitehouse et al. 2023). Furthermore, state-space models and HMMs are used to model unknown skill levels in competitive sports based on game outcomes (Herbrich et al. 2006; Štrumbelj and Vračar 2012; Duffield et al. 2023). For instance, SimBa-CL on a partition of football teams could be applied to estimate the skilled players across teams over time. Also, the factorization in (2) can be recognized in traffic congestion applications (Silva et al. 2015; Rimella and Whiteley 2022), where business states are estimated using data from preallocated sensors. In this context, SimBa-CL appears to be a promising approach, in particular, SimBa-CL with feedback would allow spatial correlations to be properly accounted for, ensuring that no information is lost.

Our empirical results show good performance for parameter estimation, particularly for large N. We believe this stems from the fact that the approximation of ignoring feedback decreases as $N \to \infty$ and composite likelihood methods have good asymptotic properties, albeit these are shown for simpler data models (Varin 2008; Varin et al. 2011). In essence, we are conjecturing that the likelihood surface induced by SimBa-CL might have some asymptotic shape whose maximum is coalescing around the data-generating parameter or a set of equally likely parameters. However, a rigorous proof of consistency is beyond the scope of this article and we leave it to future works.

Supplementary information

Code is open-source and available at the Github repository https://github.com/LorenzoRimella/SimBa-CL. Proofs and additional details are available in the supplementary material.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11222-025-10584-z.

Acknowledgements The authors acknowledge the contributions of the editor and two anonymous reviewers of the journal "Statistics and Computing", whose insightful feedback significantly improved the paper. The authors also thank the High-End Computing (HEC) facility at Lancaster University for providing the computational resources necessary to run the experiments. This work was supported by the EPSRC grants EP/R018561/1 (Bayes4Health) and EP/R034710/1 (CoSInES). PF acknowledges funding from the EPSRC grant EP/Y028783/1 (Prob_AI).

Author Contributions All authors contributed to the main ideas and the writing of the paper. Lorenzo Rimella also performed the experiments.

Funding Open access funding provided by Università degli Studi di Torino within the CRUI-CARE Agreement.

Data Availability The data and the code to reproduce the study are open-source and available at the Github repository https://github.com/LorenzoRimella/SimBa-CLThe real data are subject to copyright.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd international conference on machine learning. proceedings of machine learning research, vol. 37, pp. 1613–1622. PMLR, Lille, France (2015). https://proceedings.mlr.press/v37/blundell15.html

Chopin, N., Papaspiliopoulos, O., et al.: An Introduction to Sequential Monte Carlo, vol. 4. Springer, Berlin (2020)

Chowell, G., Hengartner, N.W., Castillo-Chavez, C., Fenimore, P.W., Hyman, J.M.: The basic reproductive number of Ebola and the effects of public health measures: the cases of Congo and Uganda. J. Theor. Biol. **229**(1), 119–126 (2004)

Cocker, D., Chidziwisano, K., Mphasa, M., Mwapasa, T., Lewis, J.M., Rowlingson, B., Sammarro, M., Bakali, W., Salifu, C., Zuza, A., et al.: Investigating one health risks for human colonisation with extended spectrum beta-lactamase producing *E. coli* and *K. pneumoniae* in Malawian households: a longitudinal cohort study. Lancet Microbe (2023)

Cocker, D., Sammarro, M., Chidziwisano, K., Elviss, N., Jacob, S., Kajumbula, H., Mugisha, L., Musoke, D., Musicha, P., Roberts, A., Rowlingson, B., Singer, A., Byrne, R., Edwards, T., Lester, R., Wilson, C., Hollihead, B., Thomson, N., Jewell, C., Morse, T., Feasey, N.: Drivers of Resistance in Uganda and Malawi (DRUM): a protocol for the evaluation of One-Health drivers of Extended Spectrum Beta Lactamase (ESBL) resistance in Low-Middle Income Countries (LMICs). Wellcome Open Res. **7**, 55 (2023). https://doi.org/10.12688/wellcomeopenres.17581

Directive of the Council of the European Union: COUNCIL DIRECTIVE 200385EC of 29 September 2003 on Community measures for the control of foot-and-mouth disease repealing Directive 85511EEC and Decisions 89531EEC and 91665EEC and amending Directive 9246EEC, Official Journal of the European Union (2003)

Duffield, S., Power, S., Rimella, L.: A state-space perspective on modelling and inference for online skill rating. arXiv preprint arXiv:2308.02414 (2023)

Fearnhead, P., Prangle, D.: Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. J. R. Stat. Soc. Ser. B Stat Methodol. **74**(3), 419–474 (2012)

Fintzi, J., Cui, X., Wakefield, J., Minin, V.N.: Efficient data augmentation for fitting stochastic epidemic models to prevalence data. J. Comput. Graph. Stat. **26**(4), 918–929 (2017)

Glennie, R., Adam, T., Leos-Barajas, V., Michelot, T., Photopoulou, T., McClintock, B.T.: Hidden Markov models: Pitfalls and opportunities in ecology. Methods Ecol. Evol. **14**(1), 43–56 (2023)

Godambe, V.P.: An optimum property of regular maximum likelihood estimation. Ann. Math. Stat. **31**(4), 1208–1211 (1960)

Gumbel, E.J.: Statistical Theory of Extreme Values and Some Practical Applications: A Series of Lectures, vol. 33. US Government Printing Office, New York (1954)

Herbrich, R., Minka, T., Graepel, T.: TrueSkill: A Bayesian Skill Rating System. Advances in neural information processing systems **19** (2006)

Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)

Jewell, C., Brown, J., Keeling, M., Green, L., Roberts, G., : Bayesian epidemic risk prediction-knowledge transfer and usability at all levels. In: Society for veterinary epidemiology and preventive medicine. Proceedings of a Meeting Held in Madrid, Spain, 20-22 March 2013, pp. 127–142 (2013). Society for Veterinary Epidemiology and Preventive Medicine

Jewell, C.P., Kypraios, T., Neal, P., Roberts, G.O.: Bayesian analysis for emerging infectious diseases. Bayesian Anal. **4**, 465–496 (2009)

Ju, N., Heng, J., Jacob, P.E.: Sequential monte carlo algorithms for agent-based models of disease transmission. arXiv preprint arXiv:2101.12156 (2021)

Keeling, M.J., Rohani, P.: Modeling Infectious Diseases in Humans and Animals. Princeton University Press, Princeton (2011)

Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

Lekone, P.E., Finkenstädt, B.F.: Statistical inference in a stochastic epidemic SEIR model with control intervention: Ebola as a case study. Biometrics **62**(4), 1170–1177 (2006)

Maddison, C.J., Mnih, A., Teh, Y.W.: The concrete distribution: A continuous relaxation of discrete random variables. arXiv preprint arXiv:1611.00712 (2016)

Rebeschini, P., Van Handel, R.: Can local particle filters beat the curse of dimensionality? Ann. Appl. Probab. **25**, 2809–2866 (2015)

Rimella, L., Whiteley, N., Jewell, C., Fearnhead, P., Whitehouse, M.: Scalable calibration for partially observed individual-based epidemic models through categorical approximations. arXiv preprint arXiv:2501.03950 (2025)

Rimella, L., Whiteley, N.: Exploiting locality in high-dimensional factorial hidden Markov models. J. Mach. Learn. Res. **23**(1), 134–167 (2022)

Rimella, L., Alderton, S., Sammarro, M., Rowlingson, B., Cocker, D., Feasey, N., Fearnhead, P., Jewell, C.: Inference on extended-spectrum beta-lactamase Escherichia coli and Klebsiella pneumoniae data through SMC2. J. R. Stat. Soc. Ser. C Appl. Stat. (2023). https://doi.org/10.1093/jrsssc/qlad055

Rimella, L., Jewell, C., Fearnhead, P.: Approximating optimal SMC proposal distributions in individual-based epidemic models. Stat. Sin. (2023). https://doi.org/10.5705/ss.202022.0198

Samanidou, E., Zschischang, E., Stauffer, D., Lux, T.: Agent-based models of financial markets. Rep. Prog. Phys. **70**(3), 409 (2007)

Scott, S.L.: Bayesian methods for hidden Markov models: recursive computing in the 21st century. J. Am. Stat. Assoc. **97**(457), 337–351 (2002)

Silva, R., Kang, S.M., Airoldi, E.M.: Predicting traffic volumes and estimating the effects of shocks in massive transportation systems. Proc. Natl. Acad. Sci. **112**(18), 5643–5648 (2015)

Štrumbelj, E., Vračar, P.: Simulating a basketball match with a homogeneous Markov model and forecasting the outcome. Int. J. Forecast. **28**(2), 532–542 (2012). https://doi.org/10.1016/j.ijforecast.2011.01.004

Touloupou, P., Finkenstädt, B., Spencer, S.E.: Scalable Bayesian inference for coupled hidden Markov and semi-Markov models. J. Comput. Graph. Stat. **29**(2), 238–249 (2020)

UK National Audit Office: Report by the Comptroller and auditor general: The 2001 outbreak of foot and mouth disease (2022). https://www.nao.org.uk/reports/the-2001-outbreak-of-foot-and-mouth-disease

Varin, C., Reid, N., Firth, D.: An overview of composite likelihood methods. Stat. Sin. 5–42 (2011)

Varin, C.: On composite marginal likelihoods. Asta Adv. Stat. Anal. **92**(1), 1–28 (2008)

Whitehouse, M., Whiteley, N., Rimella, L.: Consistent and fast inference in compartmental models of epidemics using Poisson Approximate Likelihoods. J. R. Stat. Soc. Ser. B Stat. Methodol. (2023). https://doi.org/10.1093/jrsssb/qkad065

Whiteley, N., Lee, A.: Twisted particle filters. Ann. Stat. **42**(1), 115–141 (2014). https://doi.org/10.1214/13-AOS1167

Wilkinson, D.J.: Stochastic Modelling for Systems Biology. CRC Press, New York (2018)

Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)