Simulation-based inference for stochastic nonlinear mixed-effects models with applications in systems biology

Statistics and Computing (2026) 36:99
https://doi.org/10.1007/s11222-026-10850-8

ORIGINAL PAPER

Simulation-based inference for stochastic nonlinear mixed-effects models with applications in systems biology

Henrik Häggström[1] · Sebastian Persson[1] · Marija Cvijovic[1] · Umberto Picchini[1]

Received: 17 April 2025 / Accepted: 3 February 2026

Abstract
The analysis of data from multiple experiments, such as observations of several individuals, is commonly approached using mixed-effects models, which account for variation between individuals through hierarchical representations. This makes mixed-effects models widely applied in fields such as biology, pharmacokinetics, and sociology. In this work, we propose a novel methodology for scalable Bayesian inference in hierarchical mixed-effects models. Our framework first constructs amortized approximations of the likelihood and the posterior distribution, which are then rapidly refined for each individual dataset, to ultimately approximate the parameters posterior across many individuals. The framework is easily trainable, as it uses mixtures of experts but without neural networks, leading to parsimonious yet expressive surrogate models of the likelihood and the posterior. We demonstrate the effectiveness of our methodology using challenging stochastic models, such as mixed-effects stochastic differential equations emerging in systems biology-driven problems. However, the approach is broadly applicable and can accommodate both stochastic and deterministic models. We show that our approach can seamlessly handle inference for many parameters. Additionally, we applied our method to a real-data case study of mRNA transfection. When compared to exact pseudomarginal Bayesian inference, our approach proved to be both fast and competitive in terms of statistical accuracy.

Keywords Hierarchical models · Likelihood-free inference · Stochastic differential equations · Stochastic chemical reactions

1 Introduction

The analysis of data arising from multiple experiments – such as observations of individuals across various domains (e.g., humans, animals, cells, or trees) – has traditionally been approached using mixed-effects models (Diggle et al. 2002; Lavielle 2014). These models account for individual variability by treating model parameters as random variables that vary between subjects, allowing a hierarchical representation of the data. This structure effectively disentangles different sources of variability, distinguishing intra-individual fluctuations from between-individual differences. Mixed-effects modeling is widely applied across diverse fields, including biology, pharmacokinetics and pharmacodynamics, forestry, sociology, and many other disciplines where understanding variability across individuals is essential (Davidian and Giltinan 2003). The literature to tackle the inference problem for mixed-effects models is vast. However, options reduce substantially when stochastic dynamical models are considered due to considerably increased theoretical and computational difficulties. In this work, we present a new strategy for Bayesian inference in hierarchical nonlinear stochastic models with mixed-effects. Our approach provides fast to train *semi-amortized* approximations to both the likelihood function and the posterior distribution. We show that this makes our methodology scalable for an increasing number of individuals. In particular, we consider stochastic models with time-dynamics, with a focus on stochastic differential equations (SDEs) with mixed-effects. However, we consider SDE models merely to provide illustrative case studies, as the sta-

✉ Umberto Picchini
picchini@chalmers.se

Henrik Häggström
henhagg@chalmers.se

Sebastian Persson
sebastian.persson@crick.ac.uk

Marija Cvijovic
marija.cvijovic@chalmers.se

[1] Department of Mathematical Sciences, Chalmers University of Technology and the University of Gothenburg, Gothenburg, Sweden

Published online: 02 March 2026

tistical inference methodology we propose is agnostic to the specific type of model used to describe data, the only requirement being the availability of a generative model to simulate data.

Inference for stochastic modelling with mixed-effects is a challenging area due to the difficulty in (numerically) integrating out the latent quantities that enter the likelihood function. For example, models for stochastic chemical reactions that are common in systems biology involve either exact simulators (e.g. the Stochastic Simulation Algorithm, Gillespie, 1977, the Extrande method, Voliotis et al., 2016) or approximate simulators (Gillespie, 2000, Gillespie, 2007), and all these make the likelihood function *intractable*, that is, it is not possible to evaluate the likelihood exactly and is computationally hard to approximate. For example, when the model is an SDE, and observations are available at discrete time points, the unavailability of closed-form transition densities makes the likelihood function intractable (except for the simplest toy models). The statistical literature to tackle such difficulty is vast, see eg. Craigmile et al. (2023) for a recent review. On top of such difficulty, when SDEs are embedded in a mixed-effects framework, the problem becomes even harder due to the increased dimension of the integration problem. Although the literature for mixed-effect SDEs is available, as collected in Picchini (2025), this is very specialized, as methods either tackle very specific models, lacking generality and well-maintained software, or are general but computationally very intensive, e.g. when based on particle-filters (Botha et al. 2021; Wiqvist et al. 2021; Persson et al. 2022).

In this work, we provide a general approach for Bayesian inference in mixed-effects models having intractable likelihoods, using simulation-based inference (SBI) (see Cranmer et al., 2020 for a review). The appeal of SBI methods is that these only require forward-simulation of the model, rather than the evaluation of a potentially complicated expression for the likelihood function, assuming it is available. This allows for approximate frequentist and Bayesian inference, whenever running the model simulator at many values of a parameter $\boldsymbol{\theta}$ is computationally not too onerous. Indeed, in SBI, simulated data $\boldsymbol{y}$ are generated as $\boldsymbol{\theta} \rightarrow \mathcal{M}(\boldsymbol{\theta}) \rightarrow \boldsymbol{y}$, where the *simulator* $\mathcal{M}$ can be any generative model. Provided with many pairs of simulated $\boldsymbol{\theta}$'s and $\boldsymbol{y}$'s, it is possible to build inference for given observed data $\boldsymbol{y}_o$, in absence of a readily available expression for the likelihood function $p(\boldsymbol{y}_o|\boldsymbol{\theta})$, as we briefly summarize in Section 2. In recent years many SBI methods have focused on using deep neural networks to approximate conditional densities ("neural conditional density estimation"), for example to provide approximations to the posterior $p(\boldsymbol{\theta}|\boldsymbol{y}_o)$, the likelihood $p(\boldsymbol{y}_o|\boldsymbol{\theta})$, or both, see Section 2 for key references. In our work, we obtain scalable and accurate inference for complex stochastic models with mixed-effects, without using neural conditional density estimation (NCDE). The relevance of using probabilistic frameworks that are more parsimonious than neural networks can result in easier analytic tractability: in our case, we use Gaussian mixture models (GMMs) to approximate conditional densities, and the decades-long research on GMMs (Fruhwirth-Schnatter et al. 2019) provides estimators via ad-hoc algorithms for their fitting, such as expectation-maximization (Dempster et al. 1977). Moreover, the estimators obtained from a specific conditional density (say the likelihood function), when expressed via a GMM, can easily be transferred to other conditional densities (eg the posterior) using closed-form algebraic operations.

We build on the *SEquential Mixture Posterior and Likelihood Estimation* (SeMPLE) methodology (Häggström et al. 2024), which efficiently fits training data using a GMM via an expectation-maximization algorithm. The fitted Gaussian mixture provides, simultaneously, a closed-form deterministic approximation to both the likelihood and the posterior, which can both be evaluated and sampled from in a Gibbs sampler. For the case where M individuals are considered, with corresponding observed data $\boldsymbol{y}_o^{(i)}$ $(i = 1, ..., M)$, the surrogates of the likelihood and the posterior constructed by SeMPLE are called "semi-amortized", since an initial amortized approximation of the likelihood $p(\boldsymbol{y}|\boldsymbol{\theta})$ for a generic $\boldsymbol{y}$ is first obtained, and then rapidly adapted to the individual-specific data $\boldsymbol{y}_o^{(i)}$, providing an approximation to the individual $p(\boldsymbol{y}_o^{(i)}|\boldsymbol{\theta})$, without having to re-start separate fittings completely from scratch for every individual i, but instead starting from the amortized approximation.

We present two versions of SeMPLE, both delivering accurate inference, as demonstrated through comparisons with exact (pseudomarginal) Bayesian inference. The first version offers greater flexibility by allowing the specification of both fixed parameters and random effects, although at a higher computational cost. In contrast, the second version is designed for enhanced scalability, but requires all parameters to be treated as random effects. To illustrate our methodology, we applied it to three case studies based on two models: (i) a mixed-effects Ornstein-Uhlenbeck state-space model, and (ii) a mixed-effects SDE model (used to describe translation kinetics following mRNA transfection) which is tested using both simulated and real-world data. The code is available at https://github.com/henhagg/semple_mem.

2 Related work

Simulation-based inference (SBI) methods, reviewed e.g. in Cranmer et al. (2020) and Pesonen et al. (2023), have also been called "likelihood-free inference", where the latter has been used especially with reference to approximate Bayesian computation (ABC) (Marin et al. 2012; Sisson et al. 2018), synthetic likelihoods (Wood 2010; Price et al. 2018),

and pseudomarginal Markov chain Monte Carlo methods (Andrieu and Roberts 2009; Andrieu et al. 2010) when simple forward simulation is possible (as when the bootstrap filter is used to unbiasedly approximate the likelihood). The mentioned approaches have also been denoted as "statistical SBI" in Wang et al. (2024), to distinguish those from more recent methods exploiting neural networks (typically normalizing flows, Rezende and Mohamed, 2015; Papamakarios et al., 2021) to approximate conditional densities, so-called "neural conditional density estimation" (NCDE), which have gained considerable attention. NCDE has been used to approximate likelihoods (Papamakarios et al., 2019; Chen et al., 2021), posterior distributions (Papamakarios and Murray 2016; Greenberg et al. 2019; Durkan et al. 2020; Chen et al. 2021; Miller et al. 2021; Delaunoy et al. 2022), or the likelihood and the posterior simultaneously (Wiqvist et al. 2021; Radev et al. 2023). Moreover, NCDE approaches have been used both to sequentially refine inference conditionally on a specific observed data set $\boldsymbol{y}_o$, but also in an amortized way, see the review in Zammit-Mangion et al. (2024). For amortized approaches, the trained network does not depend on any specific $\boldsymbol{y}_o$ and therefore, once training has completed, it can be used to rapidly produce conditional density estimation for any $\boldsymbol{y}_o$, though this happens at a large upfront resource investment to obtain the amortized network in the first place. Regarding non-amortized approaches, comparisons between some of the methods are available, e.g., in Greenberg et al. (2019) and Häggström et al. (2024).

For the specific case of non-SBI inference for mixed-effects stochastic dynamic models, the range of inference options is large (Picchini 2025). However, this range shrinks considerably when SBI methods are considered: for the latter, methods for mixed-effects stochastic dynamic models revolve almost exclusively around pseudomarginal Markov chain Monte Carlo (pMCMC) (Whitaker et al. 2017; Wiqvist et al. 2021; Botha et al. 2021; Persson et al. 2022), and an exception within SBI is the NCDE-based posterior inference in Arruda et al. (2024). The advantage of pMCMC methods is that they produce exact Bayesian inference in the limit of an infinite number of MCMC iterations. Therefore, when it is possible to use pMCMC, this provides gold-standard Bayesian inference. However, in practice, for pMCMC to be effective, advanced proposal mechanisms for the solution paths of the models are often needed, to reduce the variance of Monte-Carlo based likelihood approximations (typically via particle filters) and hence reduce the runtime to properly explore the posterior surface. In pMCMC, constructing proposals for the solution's paths is a challenging and highly-specialized task (examples are Golightly and Wilkinson, 2011; Del Moral and Murray, 2015; Schauer et al., 2017), and bespoke constructions often need to be produced for any different attempted model, and often depend on specific assumptions on the measurement error (e.g, a linear observation model with Gaussian measurement error). Moreover, the tuning of the parameter proposal in pMCMC (and especially its initialization) can be tedious and prone to trial-and-error. This is why alternative SBI methods that rely solely on "simple" *forward* model simulation are particularly appealing, as they facilitate learning the mapping between simulated $\boldsymbol{\theta}$ and simulated $\boldsymbol{y}$. In our work, the goal is to construct surrogate deterministic approximations of the likelihood and posterior, rather than stochastic likelihood approximations as generated in pMCMC. In doing so, we do not employ NCDE, in contrast to Arruda et al. (2024), and instead provide a more parsimonious framework which is nevertheless expressive enough to produce accurate Bayesian inference. Before moving further, we wish to remind the reader that in SBI it is typical to conduct inference based on summaries $S(\boldsymbol{y})$ (provided by a data-reduction mapping) of $\boldsymbol{y}$, rather than inference based on $\boldsymbol{y}$ itself, where $S(\boldsymbol{y})$ is a set of statistics of $\boldsymbol{y}$ that are deemed informative about $\boldsymbol{\theta}$ (Fearnhead and Prangle 2012; Wiqvist et al. 2019; Åkesson et al. 2021) but are low-dimensional compared to $\boldsymbol{y}$. In our examples we do not employ summaries, and therefore we do not discuss this aspect further, however our methodology could accommodate inference based on some $S(\boldsymbol{y})$ should it be necessary, and in such case all the instances where $\boldsymbol{y}$ appears could be substituted with $S(\boldsymbol{y})$.

3 Stochastic differential equation mixed-effects models

As mentioned in the introduction, we may consider any generative model $\mathcal{M}$ to describe time-dynamics in the data. We choose to provide illustrations based on models employing stochastic differential equations, however these could be substituted with other models, for example solvers for Markov jump processes for stochastic chemical reactions. Consider data from a population of M individuals. Assume that the dynamics for each individual i are described by a stochastic process $\{\boldsymbol{X}_t^{(i)}\}_{t\geq 0}$ indexed by t (where t often indicates time though it can also be something different), where $\boldsymbol{X}_t^{(i)} \in \mathbb{R}^d$ for every t and every $i = 1, ..., M$. Assume dynamics governed by the following stochastic differential equations (SDEs)

$$\begin{cases} d\boldsymbol{X}_t^{(i)} = \boldsymbol{\mu}(\boldsymbol{X}_t, \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, t)dt + \boldsymbol{\sigma}(\boldsymbol{X}_t, \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, t)d\boldsymbol{B}_t^{(i)} \\ \boldsymbol{X}_0^{(i)} = \boldsymbol{x}_0^{(i)} \quad i = 1, ..., M, \\ \boldsymbol{c}^{(i)} \sim \pi(\boldsymbol{c} \mid \boldsymbol{\eta}), \quad i = 1, ..., M, \end{cases} \tag{1}$$

where $\boldsymbol{\mu}$ is a d-dimensional drift vector, the diffusion coefficient $\boldsymbol{\sigma}$ is a $d \times d$ positive definite matrix, each $\boldsymbol{B}_t^{(i)}$ denotes a vector of d independent Wiener processes. In

(1) we assume individual-specific parameters $\boldsymbol{c}^{(i)} \in \mathbb{R}^q$, while $\boldsymbol{\kappa} \in \mathbb{R}^p$ is common to all individuals and $\boldsymbol{\eta} \in \mathbb{R}^u$. For the individual-specific parameters we assume $\boldsymbol{c}^{(i)} \sim \pi(\boldsymbol{c} \mid \boldsymbol{\eta})$ $(i = 1, \dots, M)$ and we denote their collection across all subjects as $\boldsymbol{c} = (\boldsymbol{c}^{(1)}, \dots, \boldsymbol{c}^{(M)})$. The parameter $\boldsymbol{\eta}$ is called "population parameter" as it is underlying the distribution of all the $\boldsymbol{c}^{(i)}$, and as such it does not vary with i. Similarly, parameter $\boldsymbol{\kappa}$ is also assumed not to vary with i and as such is common to all subjects, however, unlike $\boldsymbol{\eta}$ for the $\boldsymbol{c}^{(i)}$'s, $\boldsymbol{\kappa}$ does not identify the distribution of any other random parameter.

The process $\{\boldsymbol{X}_t^{(i)}\}$ may be observed directly (ie without error) or indirectly: here we consider the general observational model (2) where it is also possible to have noisy observations $\boldsymbol{y}_t^{(i)}$ that are conditionally independent (given the latent process), and that are linked to $\{\boldsymbol{X}_t^{(i)}\}$ via

$$\boldsymbol{Y}_t^{(i)} = g(\boldsymbol{X}_t^{(i)}, \boldsymbol{\varepsilon}_t^{(i)}), \quad \boldsymbol{\varepsilon}_t^{(i)} \sim \pi_\varepsilon(\boldsymbol{\xi}) \quad i = 1, \dots, M, \tag{2}$$

where $\boldsymbol{Y}_t^{(i)} \in \mathbb{R}^{d_o}$, with $d_o \leq d$, $\boldsymbol{\varepsilon}_t^{(i)} \in \mathbb{R}^{d_o}$ represents measurement errors with distribution $\pi_\varepsilon(\boldsymbol{\xi})$ parameterised by the vector $\boldsymbol{\xi} \in \mathbb{R}^s$, and $g(\cdot)$ is a (possibly non-linear) function. We exemplify the dependence relationship between the introduced parameters and the stochastic processes in Figure 1. Evidently, if we assume no measurement error, then the observations $\boldsymbol{y}^{(i)}$ are direct (error-free) observations of $\{\boldsymbol{X}_t^{(i)}\}_{t \geq 0}$. Assume that noisy observations are collected at time-points $\{t_1, t_2, ..., t_n\}$, then we can have the case where at observational time t_j the observation $\boldsymbol{y}_{t_j}^{(i)}$ is a vector of length d_o, where having $d_o = d$ corresponds to the system being fully observed at t_j, whereas having $d_o < d$ opens up for $\{\boldsymbol{X}_t^{(i)}\}$ being "partially observed". A typical example would be $\boldsymbol{Y}_{t_j}^{(i)} = \boldsymbol{F}_{t_j}^{(i)} \boldsymbol{X}_{t_j}^{(i)} + \boldsymbol{\varepsilon}_{t_j}^{(i)}$, where the $\boldsymbol{F}_{t_j}^{(i)}$ are $d_o \times d$ matrices of known coefficients. In the notation introduced, we assumed for simplicity that the observational times are the same for all individuals, but we could have also used $\{t_1^{(i)}, t_2^{(i)}, ..., t_{n_i}^{(i)}\}$ and assumed that the set of observations are of different lengths n_i for different individuals: this can be handled in our framework but we decided to keep the notation lighter, for ease of reading. The vector of observed data for subject i is therefore $\boldsymbol{y}_o^{(i)} = (\boldsymbol{y}_{1,o}^{(i)}, ..., \boldsymbol{y}_{n,o}^{(i)}) \in \mathbb{R}^{d_o \times n}$ where we used the shorthand $\boldsymbol{y}_j^{(i)} \equiv \boldsymbol{y}_{t_j}^{(i)}$. The full set of observations stacks all individual observations as $\boldsymbol{y}_o = (\boldsymbol{y}_o^{(1)}, ..., \boldsymbol{y}_o^{(M)})^{\mathrm{T}}$. To simplify the reading, in next sections we will use $\boldsymbol{y}$ to denote a generic dataset, observed or simulated, and we will distinguish the two cases only when necessary.

Equations (1)-(2) define a very flexible model, a *stochastic differential equation mixed-effects model* (SDEMEM), which is able to represent (i) stochastic intra-individual variation (via the diffusion terms in the SDEs), (ii) between individuals variation (via the distribution $\pi(\boldsymbol{c}|\boldsymbol{\eta})$ of the individual parameters) and (iii) (optional) measurement variability (via $\pi_\varepsilon(\boldsymbol{\xi})$), if measurement error is at all considered. The model (1)-(2) is a state-space model, namely the latent process is Markovian and observations are assumed conditionally independent given the latent process. The Markovianity of $\{\boldsymbol{X}_t^{(i)}\}_{t \geq 0}$ implies that, in principle, exact simulation of solution paths to SDEs could arise either by sampling directly from the transition density $\boldsymbol{x}_t^{(i)} \sim \pi(\boldsymbol{x}_t^{(i)} | \boldsymbol{x}_s^{(i)}, \boldsymbol{c}^{(i)}, \boldsymbol{\kappa})$, for $s < t$, or using rejection sampling approaches (Beskos and Roberts 2005; Beskos et al. 2006). However, typically, and except for the simplest cases, exact approaches are not feasible, and numerical approximations (schemes) are implemented instead. Here we consider the simplest and most commonly used approximation scheme, which is the Euler-Maruyama scheme, where the solution $\boldsymbol{x}_t^{(i)}$ at time t is advanced to time $t + \Delta_t$ via

$$\begin{aligned} \boldsymbol{x}_{t+\Delta_t}^{(i)} &= \boldsymbol{x}_t^{(i)} + \boldsymbol{\mu}(\boldsymbol{x}_t^{(i)}, \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, t)\Delta_t \\ &\quad + \boldsymbol{\sigma}(\boldsymbol{x}_t^{(i)}, \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, t) \cdot \boldsymbol{u}_t^{(i)}, \end{aligned} \tag{3}$$

where $\boldsymbol{u}_t^{(i)} \sim \mathcal{N}(0, \Delta_t \boldsymbol{I}^{d \times d})$, for some step size $\Delta_t > 0$ and conditionally on $\boldsymbol{c}^{(i)}$ and $\boldsymbol{\kappa}$. Here and in the following we use $\mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$ to denote a multivariate Gaussian distributions with mean $\boldsymbol{m}$ and covariance matrix $\boldsymbol{\Sigma}$. However, whenever possible, more accurate numerical scheme should be considered for the solution of the SDE in (1) (a recent monograph is Higham and Kloeden, 2021), for example Buckwar et al. (2020) displays the inference bias in the posterior of the parameters when certain SDEs are solved using Euler-Maruyama. In conclusion, the generative model that in Section 1 was denoted $\mathcal{M}(\boldsymbol{\theta})$, here it is represented by (1)-(2). While we could also denote the generative model with $\mathcal{M}_\Delta(\boldsymbol{\theta})$, since a numerical scheme with stepsize Δ is required to solve the SDE, for simplicity we will keep using $\mathcal{M}(\boldsymbol{\theta})$. Importantly, while in this work the applications and case-studies focus on inference for SDE mixed-effects models, where observations arise from a state-space model formulation (1)-(2), this is only a possible example of application of the methodology offered in Sections 5-6. In fact, in Häggström et al. (2024) the original SeMPLE methodology, which was not specialized to mixed-effects state-space SDE models, was tested on examples that included both static and dynamic models, the only requirement being the ability to simulate data from a generative model $\mathcal{M}(\boldsymbol{\theta})$. Therefore, in this work we confirm that the methodology in Häggström et al. (2024) is flexible. As an additional possibility (not pursued in this work), it would not be difficult to show the possibility to accommodate, for example, observations arising from non-Markovian processes $\{X_t^{(i)}\}_{t \geq 0}$, or where the $Y_t^{(i)} | X_t^{(i)}$ are not conditionally independent.

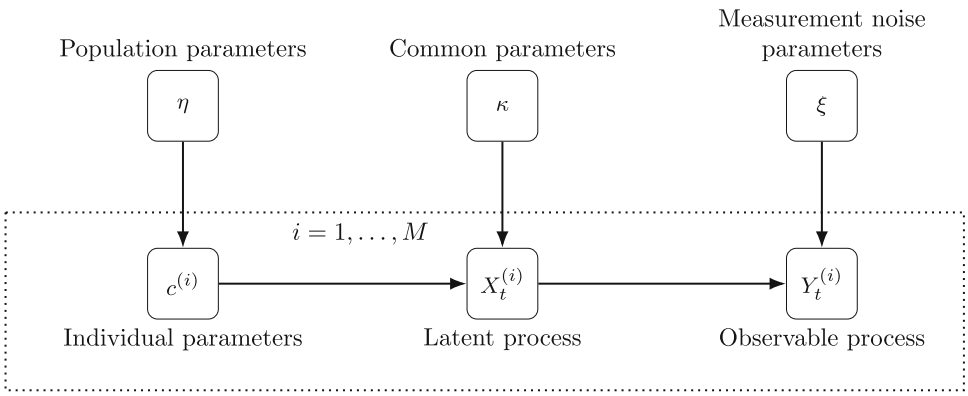

Fig. 1 Bayesian network model structure for a SDE mixed-effects model

4 Bayesian inference for SDEMEMs

As mentioned in the previous section, we will use $\boldsymbol{y}$ to denote both simulated data and observed data, and we will make use of the notation $\boldsymbol{y}_o$ for observed data only when necessary. Denoting with $\boldsymbol{y} = (\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(M)})^{\mathrm{T}}$ the stacked data from all M individuals, the full vector of parameters to learn is $\boldsymbol{\theta} = (\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\eta}, \boldsymbol{\xi})$, where $\boldsymbol{c} = (\boldsymbol{c}^{(1)}, ..., \boldsymbol{c}^{(M)})$. Notice, if data are not assumed to be affected with measurement error, then $\boldsymbol{\xi}$ is not included in $\boldsymbol{\theta}$. Below we consider the most general scenario. Assume that the $\boldsymbol{c}^{(i)}$'s are mutually independent and that data $\boldsymbol{y}^{(i)}$ are independent conditionally on the $\boldsymbol{c}^{(i)}$'s. We wish to sample from the full posterior distribution

$$\pi(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\eta}, \boldsymbol{\xi} \mid \boldsymbol{y}) \propto \pi(\boldsymbol{\kappa}, \boldsymbol{\eta}, \boldsymbol{\xi}) \prod_{i=1}^{M} \pi(\boldsymbol{y}^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi}) \pi(\boldsymbol{c}^{(i)} \mid \boldsymbol{\eta}), \tag{4}$$

where the individual likelihoods are obtained by the marginalization

$$\begin{aligned} \pi(\boldsymbol{y}^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi}) &= \int \pi(\boldsymbol{y}^{(i)}, \boldsymbol{x}^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi}) d\boldsymbol{x}^{(i)} \qquad (5) \\ &= \int \prod_{j=1}^{n} \pi(\boldsymbol{y}_j^{(i)} \mid \boldsymbol{x}_j^{(i)}, \boldsymbol{\xi}) \pi(\boldsymbol{x}_0^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}) \times \\ &\quad \prod_{j=1}^{n} \pi(\boldsymbol{x}_j^{(i)} \mid \boldsymbol{x}_{j-1}^{(i)}, \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}) d\boldsymbol{x}_0^{(i)} \cdots d\boldsymbol{x}_n^{(i)}, \qquad (6) \end{aligned}$$

where the $\boldsymbol{x}_j^{(i)} \equiv \boldsymbol{x}_{t_j}^{(i)}$ are values of the $\{\boldsymbol{X}_t^{(i)}\}_{t \geq 0}$ process at the observational times, and where $\boldsymbol{x}_0^{(i)}$ is a starting value for $\{\boldsymbol{X}_t^{(i)}\}_{t \geq 0}$ at time $t = 0$. The integral (6) is generally analytically intractable, except for simple cases, which motivates the need for either particle methods (sequential Monte Carlo) to get an unbiased estimate of the individual likelihood or, as we propose, a simulation-based method that provides a deterministic surrogate model as an approximation of the individual likelihood.

The graphical dependency in Figure 1, coupled with the state-space structure of the model (equations (1)-(2)), suggests a natural factorization of the full posterior (4), from which we sample using a Gibbs sampler, following Wiqvist et al. (2021), Botha et al. (2021), Persson et al. (2022), who used the same factorization for SDEMEMs. The Gibbs sampler iterates through the following steps

$$\text{step 1: } \pi(\boldsymbol{c} \mid \boldsymbol{\kappa}, \boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{y}) \propto \prod_{i=1}^{M} \pi(\boldsymbol{c}^{(i)} \mid \boldsymbol{\eta}) \pi(\boldsymbol{y}^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi}) \tag{7}$$

$$\text{step 2: } \pi(\boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{c}, \boldsymbol{\eta}, \boldsymbol{y}) \propto \pi(\boldsymbol{\kappa}, \boldsymbol{\xi}) \prod_{i=1}^{M} \pi(\boldsymbol{y}^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi}) \tag{8}$$

$$\text{step 3: } \pi(\boldsymbol{\eta} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi}, \boldsymbol{y}) \propto \pi(\boldsymbol{\eta}) \prod_{i=1}^{M} \pi(\boldsymbol{c}^{(i)} \mid \boldsymbol{\eta}). \tag{9}$$

Notice that the first step (equation (7)) is equivalent to (10), which in practice we employ in our implementation. That is, since we assume that data from each individual i are conditionally independent from the data of another individual i' ($i' \neq i$), then we can separately sample each individual $\boldsymbol{c}^{(i)}$ ($i = 1, ..., M$) from

$$\pi(\boldsymbol{c}^{(i)} \mid \boldsymbol{\kappa}, \boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{y}^{(i)}) \propto \pi(\boldsymbol{c}^{(i)} \mid \boldsymbol{\eta}) \pi(\boldsymbol{y}^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi}), \tag{10}$$

then stack the obtained draws into $\boldsymbol{c}$. The fact that the first step is broken-down into M independent contributions allows for parallelization, and in case of a large number of individuals this could reduce the runtime significantly. The second step in the Gibbs sampler pertains to the sampling of fixed effect parameters $\boldsymbol{\kappa}$ and $\boldsymbol{\xi}$, which are shared by all individuals, and the fact that their sampling is separated from that of the $\boldsymbol{c}^{(i)}$'s allows $(\boldsymbol{\kappa}, \boldsymbol{\xi})$ to be treated as common across all individuals. Notice that our treatment is different from, for example, Arruda et al. (2024) where such shared parameters are modeled as random effects with zero variance. A good reason for considering a Gibbs sampler is that its first step (7)

(or equivalently (10)), which requires a Metropolis-within-Gibbs approach, can benefit from an automatically learned proposal sampler that is particularly suited for multimodal targets, as we detail in Section 5. However, in practice, sampling from (7)-(8) when dealing with SDEMEMs is difficult, in that the individual likelihoods $\pi(\boldsymbol{y}^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ are typically intractable large-dimensional integrals (see (6)) that get approximated via Monte Carlo simulations, and we refer to Wiqvist et al. (2019), Botha et al. (2021) and Persson et al. (2022) for powerful but computer intensive approaches based on particle filters. Even worse, these computationally intensive Monte Carlo approximations, resulting in pseudo-marginal pMCMC methods, are difficult to tune and have to be re-executed for any considered $\boldsymbol{\theta}$ and whenever a new $\boldsymbol{y}^{(i)}$ is considered. Hence large values of individuals (M) can break down the feasibility of such approaches (but see the ameliorating strategies in Persson et al., 2022). To address this problem, in the next section, we show how to obtain both an amortized surrogate likelihood model, and an amortized surrogate posterior model in the form of two Gaussian mixture models ("mixture of experts"). Moreover, we anticipate that there can be computational advantages in eliminating the second Gibbs step (hence assuming that all parameters are random effects), which is discussed in Section 6.1.

5 Likelihood and posterior estimation via mixtures of experts

The main contribution of this work is to create a framework for Bayesian inference where the likelihood $\pi(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ is first approximated in an amortized way and then is specialized to a given observation $\boldsymbol{y}_o^{(i)}$ (for every $i = 1, ..., M$). We say that our method is "semi-amortized" because, unlike amortized procedures, we do not simply plug the observation $\boldsymbol{y}_o^{(i)}$ into an amortized approximation of $\pi(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ that can be evaluated, but instead start from an amortized likelihood, and this is sequentially refined for the specific observation $\boldsymbol{y}_o^{(i)}$, to provide the final $\pi(\boldsymbol{y}_o^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi})$. We will see that the same computations involved in producing the approximation to $\pi(\boldsymbol{y}_o^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ will return, as a by-product and without any further computation, an approximation to the posterior $\pi(\boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y}_o^{(i)})$. This is achieved via the *SEquential Mixtures Posterior and Likelihood Estimation* (SeMPLE) inference of Häggström et al. (2024). Once all individual likelihoods $\pi(\boldsymbol{y}_o^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ are approximated with *deterministic* surrogate functions, they are used in place of intractable likelihoods in (7) and (8). This is different from the pseudomarginal MCMC approach in Wiqvist et al. (2021), Botha et al. (2021) and Persson et al. (2022), that compute Monte Carlo estimates of the likelihoods $\pi(\boldsymbol{y}^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi})$. It is of interest to investigate whether the surrogate likelihoods from SeMPLE provide accurate approximations of the true intractable likelihood and, thus, accurate posterior inference. It is also of interest to discuss the computational footprint of using SeMPLE to estimate the likelihoods, compared to gold-standard pseudomarginal methods for SDEMEMs.

The remaining of this section (up until equation (23)) is not specific to stochastic mixed effects models, and here we use $\boldsymbol{y}$ to denote a generic observable and $\boldsymbol{\theta}$ to denote a generic model parameter. SeMPLE uses surrogates that are mixture-of-experts whose parameters are fitted via expectation-maximization (Xu et al. 1994) on training data obtained via an efficient Markov chain Monte Carlo (MCMC) algorithm. The SeMPLE approach makes repeated use, in a sequential way, of Gaussian locally linear mappings (GLLiM, Deleforge et al., 2014). GLLiM introduces a mixture of experts in the form of a GMM defined on the joint distribution for $(\boldsymbol{\theta}, \boldsymbol{y})$. The relationship between $\boldsymbol{y} \in \mathbb{R}^{d_o \times n}$ and $\boldsymbol{\theta} \in \mathbb{R}^l$ is assumed to be locally linear, defined using a latent variable $z \in \{1, ..., K\}$, via the following *surrogate generative model*

$$\boldsymbol{y} = \sum_{k=1}^{K} \mathbb{I}_{\{z=k\}} (\tilde{\boldsymbol{A}}_k \boldsymbol{\theta} + \tilde{\boldsymbol{b}}_k + \tilde{\boldsymbol{\epsilon}}_k), \tag{11}$$

where $\mathbb{I}$ denotes the indicator function, and $\tilde{\boldsymbol{A}}_k \in \mathbb{R}^{(d_o \times n) \times l}$ and $\tilde{\boldsymbol{b}}_k \in \mathbb{R}^{d_o \times n}$ are matrices and vectors, respectively, defining the affine transformation of $\boldsymbol{\theta}$ in (11), while $\tilde{\boldsymbol{\epsilon}}_k \in \mathbb{R}^{d_o \times n}$ corresponds to an error term capturing both the observational noise and the modelling error due to assuming an affine approximation for the data. In the following we consider Gaussian noise, $\tilde{\boldsymbol{\epsilon}}_k \sim \mathcal{N}(\boldsymbol{0}, \tilde{\boldsymbol{\Sigma}}_k)$, and assume $\tilde{\boldsymbol{\epsilon}}_k$ to not depend on $\boldsymbol{\theta}$, $\boldsymbol{y}$ nor z.

Conditionally on component $z = k$, an approximate generative model is given by

$$q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{y} \mid \boldsymbol{\theta}, z = k) = \mathcal{N}(\boldsymbol{y}; \tilde{\boldsymbol{A}}_k \boldsymbol{\theta} + \tilde{\boldsymbol{b}}_k, \tilde{\boldsymbol{\Sigma}}_k). \tag{12}$$

To complete the hierarchical model, $\boldsymbol{\theta}$ is assumed to follow a mixture of Gaussian distributions specified by

$$q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{\theta} \mid z = k) = \mathcal{N}_l(\boldsymbol{\theta}; \tilde{\boldsymbol{\nu}}_k, \tilde{\boldsymbol{\Gamma}}_k), \quad q_{\tilde{\boldsymbol{\phi}}}(z = k) = \pi_k. \tag{13}$$

The GLLiM hierarchical construction above (eq.(11) to (13)) defines a joint GMM on $(\boldsymbol{y}, \boldsymbol{\theta})$:

$$q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{y}, \boldsymbol{\theta}) = \sum_{k=1}^{K} q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{y} \mid \boldsymbol{\theta}, z = k) q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{\theta} \mid z = k) q_{\tilde{\boldsymbol{\phi}}}(z = k),$$

where the full vector of mixture model parameters is $\tilde{\boldsymbol{\phi}} = \{\pi_k, \tilde{\boldsymbol{\nu}}_k, \tilde{\boldsymbol{\Gamma}}_k, \tilde{\boldsymbol{A}}_k, \tilde{\boldsymbol{b}}_k, \tilde{\boldsymbol{\Sigma}}_k\}_{k=1}^K$. Given this joint distribution, we can easily deduce the conditional distributions $q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{y} \mid \boldsymbol{\theta})$ and

$q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid \boldsymbol{y})$ in closed-form. First, we have the surrogate likelihood

$$q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{y} \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} \tilde{\omega}_k(\boldsymbol{\theta}) \mathcal{N}(\boldsymbol{y}; \tilde{\boldsymbol{A}}_k \boldsymbol{\theta} + \tilde{\boldsymbol{b}}_k, \tilde{\boldsymbol{\Sigma}}_k), \tag{14}$$

with

$$\tilde{\omega}_k(\boldsymbol{\theta}) = \frac{\pi_k \mathcal{N}_l(\boldsymbol{\theta}; \tilde{\boldsymbol{\nu}}_k, \tilde{\boldsymbol{\Gamma}}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}_l(\boldsymbol{\theta}; \tilde{\boldsymbol{\nu}}_j, \tilde{\boldsymbol{\Gamma}}_j)}. \tag{15}$$

Notice the important distinction that $q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{y} \mid \boldsymbol{\theta})$ is the approximate (surrogate) likelihood, while $\pi(\boldsymbol{y}|\boldsymbol{\theta})$ is the true (unknown) likelihood that underlines the generative model that we denoted with $\mathcal{M}(\boldsymbol{\theta})$ in Section 1 and that, for SDEMEMs, corresponds to equations (1)-(2). As a consequence of assuming a mixture model as the joint model for $(\boldsymbol{y}, \boldsymbol{\theta})$, we immediately obtain also a surrogate posterior, without the need for further training, and this posterior is given by

$$q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid \boldsymbol{y}) = \sum_{k=1}^{K} \omega_k(\boldsymbol{y}) \mathcal{N}_l(\boldsymbol{\theta}; \boldsymbol{A}_k \boldsymbol{y} + \boldsymbol{b}_k, \boldsymbol{\Sigma}_k), \tag{16}$$

with

$$\omega_k(\boldsymbol{y}) = \frac{\pi_k \mathcal{N}(\boldsymbol{y}; \boldsymbol{\nu}_k, \boldsymbol{\Gamma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{y}; \boldsymbol{\nu}_j, \boldsymbol{\Gamma}_j)}, \tag{17}$$

where $\boldsymbol{\phi} = \{\pi_k, \boldsymbol{\nu}_k, \boldsymbol{\Gamma}_k, \boldsymbol{A}_k, \boldsymbol{b}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$. The surrogate (16) is obtained instantaneously and in closed-form, because we have the following algebraic relationships deduced from $\tilde{\boldsymbol{\phi}}$ (Deleforge et al. 2014):

$$\boldsymbol{\nu}_k = \tilde{\boldsymbol{A}}_k \tilde{\boldsymbol{\nu}}_k + \tilde{\boldsymbol{b}}_k, \tag{18}$$

$$\boldsymbol{\Gamma}_k = \tilde{\boldsymbol{\Sigma}}_k + \tilde{\boldsymbol{A}}_k \tilde{\boldsymbol{\Gamma}}_k \tilde{\boldsymbol{A}}_k^{\top}, \tag{19}$$

$$\boldsymbol{\Sigma}_k = \left(\tilde{\boldsymbol{\Gamma}}_k^{-1} + \tilde{\boldsymbol{A}}_k^{\top} \tilde{\boldsymbol{\Sigma}}_k^{-1} \tilde{\boldsymbol{A}}_k\right)^{-1}, \tag{20}$$

$$\boldsymbol{A}_k = \boldsymbol{\Sigma}_k \tilde{\boldsymbol{A}}_k^{\top} \tilde{\boldsymbol{\Sigma}}_k^{-1}, \tag{21}$$

$$\boldsymbol{b}_k = \boldsymbol{\Sigma}_k \left(\tilde{\boldsymbol{\Gamma}}_k^{-1} \tilde{\boldsymbol{\nu}}_k - \tilde{\boldsymbol{A}}_k^{\top} \tilde{\boldsymbol{\Sigma}}_k^{-1} \tilde{\boldsymbol{b}}_k\right). \tag{22}$$

Therefore, given training data of model parameters and simulated data $\{\boldsymbol{\theta}_j, \boldsymbol{y}_j\}_{j=1}^{N}$, $\tilde{\boldsymbol{\phi}}$ is first estimated via expectation-maximization (EM, Xu et al., 1994; Deleforge et al., 2014), to produce $q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{y} \mid \boldsymbol{\theta})$, and then the corresponding approximation for $\boldsymbol{\phi}$ is obtained at no cost via (18)–(22), so that the surrogate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid \boldsymbol{y})$ is also identified. To run EM, our implementation of SeMPLE uses the R `xLLiM` package (Perthame et al. 2022), which gives several options to parametrize the covariance matrices $\tilde{\boldsymbol{\Sigma}}_k$ (and hence, implicitly, also the $\boldsymbol{\Sigma}_k$), see Appendix B for more details. Before moving forward, notice that the number of components K used either in (14) or (16) can be set in a principled way, as described in Appendix B, where the Bayesian information criterion (BIC) is employed.

Rewriting the Gibbs sampler in equations (7)-(9) with the surrogate likelihood $q_{\tilde{\boldsymbol{\phi}}}$ results in the following approximated Gibbs steps (the third step is still exact as it does not involve likelihoods)

$$\hat{\pi}(\boldsymbol{c}^{(i)} \mid \boldsymbol{\kappa}, \boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{y}^{(i)}) \propto \pi(\boldsymbol{c}^{(i)} \mid \boldsymbol{\eta})\, q_{\tilde{\phi}}(\boldsymbol{y}^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi}), \quad i = 1, ..., M. \tag{23}$$

$$\hat{\pi}(\boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{c}, \boldsymbol{\eta}, \boldsymbol{y}) \propto \pi(\boldsymbol{\kappa}, \boldsymbol{\xi}) \prod_{i=1}^{M} q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{y}^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi}) \tag{24}$$

$$\pi(\boldsymbol{\eta} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi}, \boldsymbol{y}) \propto \pi(\boldsymbol{\eta}) \prod_{i=1}^{M} \pi(\boldsymbol{c}^{(i)} \mid \boldsymbol{\eta}). \tag{25}$$

Note that, in terms of the GLLiM notation, we have $\boldsymbol{\theta} = (\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ rather than $\boldsymbol{\theta} = (\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\eta}, \boldsymbol{\xi})$, and this is because, since the third Gibbs step does not involve likelihoods, GLLiM is not used to infer $\boldsymbol{\eta}$, given that $\boldsymbol{\eta}$ can be sampled in (25) in an exact way, either via conjugacy or using of-the-shelves algorithms such as the No-U-Turn Sampler (NUTS, Hoffman and Gelman, 2014), which we employed via the probabilistic framework `Stan` (Stan Development Team 2023), using default settings. As the surrogate likelihood provided by GLLiM is a mixture of Gaussians, this is differentiable, and therefore, it is possible to use gradient-informed MCMC methods such as NUTS, as implemented in `Stan`. Compared to the previously mentioned pseudo-marginal method found in Persson et al. (2022) and named PEPSDI (standing for `Particles Engine for Population Stochastic DynamIcs`), which requires non-trivial tuning of the covariance matrix of the proposal kernel, we found that with `Stan` the implemented warm-up scheme worked well without user input, when provided with a reasonable start-guess for $\boldsymbol{\theta}$.

However, as we motivate in Section 6, we will prefer to employ $q_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{y})$ as a multimodal proposal sampler in step 1 (eq. (23)), and instead employ HMC only for the other two steps. Furthermore, in Section 6.1, we illustrate another, more scalable, Gibbs sampler associated to assuming all parameters as random effects. To efficiently apply this sampler with the surrogate likelihood, we first train the GLLiM model on simulated data in a sequential manner, ensuring that the data become increasingly informative for the observed $\boldsymbol{y}_o$. The full procedure is described in the SeMPLE mixed-effects algorithm in the following section.

Algorithm 1 SeMPLE for mixed-effects models

Input: $\pi(\boldsymbol{c} \mid \boldsymbol{\eta})$, $\pi(\boldsymbol{\eta})$, $\pi(\boldsymbol{\kappa}, \boldsymbol{\xi})$, observed data $\{\boldsymbol{y}_o^{(i)}\}_{i=1}^M$, simulator $\pi(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$, positive integers R (number of SeMPLE rounds), N (number of prior predictive simulations at round $r = 0$), N_g (number of Gibbs samples).

Output: Posterior samples $\{\{\boldsymbol{c}_j^{(i)}\}_{i=1}^M, \boldsymbol{\kappa}_j, \boldsymbol{\xi}_j, \boldsymbol{\eta}_j\}_{j=1}^{N_g}$

Round $r = 0$

1: Sample iid $\boldsymbol{\eta}_j \sim \pi(\boldsymbol{\eta})$, $j = 1, ..., N$
2: Sample $\boldsymbol{c}_j \sim \pi(\boldsymbol{c} \mid \boldsymbol{\eta}_j)$, $j = 1, ..., N$
3: Sample $(\boldsymbol{\kappa}_j, \boldsymbol{\xi}_j) \sim \pi(\boldsymbol{\kappa}, \boldsymbol{\xi})$, $j = 1, ..., N$
4: Simulate N single-individual datasets $\boldsymbol{y}^{(j)} \sim \pi(\boldsymbol{y} \mid \boldsymbol{c}_j, \boldsymbol{\kappa}_j, \boldsymbol{\xi}_j)$, $j = 1, ..., N$
5: Collect $\mathcal{D}_0 = \{(\boldsymbol{c}_j, \boldsymbol{\kappa}_j, \boldsymbol{\xi}_j), \boldsymbol{y}_j\}_{j=1}^N$
6: Train single individual likelihood $q_{\tilde{\boldsymbol{\phi}}_0}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ on $\mathcal{D}_0$.
7: Obtain $q_{\boldsymbol{\phi}_0}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y})$ in closed form from $q_{\tilde{\boldsymbol{\phi}}_0}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$, using (18)–(22).
8:

Round $r = 1$

9: **for** $i = 1 : M$ **do**
10: Sample from surrogate posterior iid $(\boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_j^{(i)}, \boldsymbol{\xi}_j^{(i)}) \sim q_{\boldsymbol{\phi}_0}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y}_o^{(i)})$, $j = 1, ..., N/M$
11: Simulate $\boldsymbol{y}_j^{(i)} \sim \pi(\boldsymbol{y} \mid \boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_j^{(i)}, \boldsymbol{\xi}_j^{(i)})$, $j = 1, ..., N/M$
12: Collect $\mathcal{D}_1^i = \{(\boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_j^{(i)}, \boldsymbol{\xi}_j^{(i)}), \boldsymbol{y}_j^{(i)}\}_{j=1}^{N/M}$
13: **end for**
14: Train single individual likelihood $q_{\tilde{\boldsymbol{\phi}}_1}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ on $\mathcal{D}_1 = \bigcup_{i=1}^M \mathcal{D}_1^i$
15: Obtain $q_{\boldsymbol{\phi}_1}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y})$ in closed form from $q_{\tilde{\boldsymbol{\phi}}_0}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$, using (18)–(22).

16: Initialize $\boldsymbol{c}_1 \leftarrow \boldsymbol{c}_{N/M}$, $(\boldsymbol{\kappa}_1, \boldsymbol{\xi}_1) = (\bar{\boldsymbol{\kappa}}_{N/M}, \bar{\boldsymbol{\xi}}_{N/M})$, and $\boldsymbol{\eta}_1 \sim \pi(\boldsymbol{\eta})$
17: **for** $r = 2 : R$ **do**
18: **for** $j = 2 : N_g$ **do**
19: Sample new $\boldsymbol{c}_j^{(i)} \sim \pi(\boldsymbol{c}_{j-1}^{(i)} \mid \boldsymbol{\eta}_{j-1}) q_{\tilde{\boldsymbol{\phi}}_{r-1}}(\boldsymbol{y}_o^{(i)} \mid \boldsymbol{c}_{j-1}^{(i)}, \boldsymbol{\kappa}_{j-1}, \boldsymbol{\xi}_{j-1})$, $i = 1, ..., M$ via Algorithm 2
20: Sample new $(\boldsymbol{\kappa}_j, \boldsymbol{\xi}_j) \sim \pi(\boldsymbol{\kappa}_{j-1}, \boldsymbol{\xi}_{j-1}) \prod_{i=1}^M q_{\tilde{\boldsymbol{\phi}}_{r-1}}(\boldsymbol{y}_o^{(i)} \mid \boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_{j-1}, \boldsymbol{\xi}_{j-1})$ via HMC
21: Sample new $\boldsymbol{\eta}_j \sim \pi(\boldsymbol{\eta}_{j-1}) \prod_{i=1}^M \pi(\boldsymbol{c}_j^{(i)} \mid \boldsymbol{\eta}_{j-1})$ via HMC or conjugacy
22: Simulate $\boldsymbol{y}_j^{(i)} \sim \pi(\boldsymbol{y} \mid \boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_j, \boldsymbol{\xi}_j)$, $i = 1, ..., M$
23: **end for**
24: Collect $\mathcal{D}_r = \left\{\{\boldsymbol{c}_j^{(i)}\}_{i=1}^M, \boldsymbol{\kappa}_j, \boldsymbol{\xi}_j, \{\boldsymbol{y}_j^{(i)}\}_{i=1}^M\right\}_{j=1}^{N_g}$
25: Train single individual likelihood $q_{\tilde{\boldsymbol{\phi}}_r}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ on $\bigcup_{\tilde{r}=1}^r \mathcal{D}_{\tilde{r}}$
26: Obtain $q_{\boldsymbol{\phi}_r}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y})$ in closed form from $q_{\tilde{\boldsymbol{\phi}}_0}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$, using (18)–(22).
27: **end for**

6 SeMPLE for mixed-effects models

Now that we have outlined (in Section 5) the flexible structure of SeMPLE for a generic generative model $\mathcal{M}$, giving rise to simulated observations $\boldsymbol{y}$ or actual observations $\boldsymbol{y}_o$, we now specialize it to the case where $\mathcal{M}$ is a state-space model with mixed-effects. Therefore, by recalling Section 3, we now

Algorithm 2 Independence-Metropolis-Hastings for $\boldsymbol{c}$

Here round r is $r \geq 2$ and $q_{\boldsymbol{\phi}_{r-1}}(\boldsymbol{c} \mid \boldsymbol{y}_o^{(i)})$ produces independent samples. For the initial value $\tilde{\boldsymbol{c}}_1^{(i)}$, pick the last value $\boldsymbol{c}_{j-1}^{(i)}$ stored in the Markov chain from the previous Gibbs run.

1: Propose $\tilde{\boldsymbol{c}}_*^{(i)} \sim q_{\boldsymbol{\phi}_{r-1}}(\boldsymbol{c} \mid \boldsymbol{y}_o^{(i)})$ independently,
2: $\alpha = \min\left\{1, \frac{\pi(\tilde{\boldsymbol{c}}_*^{(i)} \mid \boldsymbol{\eta}_{j-1}) q_{\tilde{\boldsymbol{\phi}}_{r-1}}(\boldsymbol{y}_o^{(i)} \mid \tilde{\boldsymbol{c}}_*^{(i)}, \boldsymbol{\kappa}_{j-1}, \boldsymbol{\xi}_{j-1})}{\pi(\tilde{\boldsymbol{c}}_1^{(i)} \mid \boldsymbol{\eta}_{j-1}) q_{\tilde{\boldsymbol{\phi}}_{r-1}}(\boldsymbol{y}_o^{(i)} \mid \tilde{\boldsymbol{c}}_1^{(i)}, \boldsymbol{\kappa}_{j-1}, \boldsymbol{\xi}_{j-1})} \times \frac{q_{\boldsymbol{\phi}_{r-1}}(\tilde{\boldsymbol{c}}_1^{(i)} \mid \boldsymbol{y}_o^{(i)})}{q_{\boldsymbol{\phi}_{r-1}}(\tilde{\boldsymbol{c}}_*^{(i)} \mid \boldsymbol{y}_o^{(i)})}\right\}$
3: Sample $u \sim \mathcal{U}[0, 1]$
4: **if** $u \leq \alpha$ **then**
5: $\tilde{\boldsymbol{c}}_2^{(i)} = \tilde{\boldsymbol{c}}_*^{(i)}$
6: **else**
7: $\tilde{\boldsymbol{c}}_2^{(i)} = \tilde{\boldsymbol{c}}_1^{(i)}$
8: **end if**
9: $\boldsymbol{c}_j^{(i)} = \tilde{\boldsymbol{c}}_2^{(i)}$

have to consider that $\boldsymbol{y}_o$ consists of a collection of observations $\boldsymbol{y}^{(i)}$ across M individuals, hence $\boldsymbol{y}_o = (\boldsymbol{y}^{(1)}, ..., \boldsymbol{y}^{(M)})$, that individual data $\boldsymbol{y}^{(i)}$ are conditionally independent given random effects $\boldsymbol{c}^{(i)}$, and that for each individual the observed data $\boldsymbol{y}_t^{(i)}$, at time t, are considered as noisy observations of a latent process $\boldsymbol{X}_t^{(i)}$ at time t.

Our inference approach alternates between the following two tasks: (i) **sampling** via Gibbs a batch of size N_g (or N depending on cases) of parameters $\boldsymbol{\theta} = (\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\eta}, \boldsymbol{\xi})$ from (an approximation of) the full posterior, and (ii) **training** the surrogate posterior and surrogate likelihood by fitting a dataset $\mathcal{D} = \{\boldsymbol{\theta}_j, \boldsymbol{y}_j\}_{j=1}^{N_g \text{ or } N}$ using expectation-maximization (via GLLiM, as introduced in Section 5), where $\boldsymbol{\theta}_j \rightarrow \mathcal{M}(\boldsymbol{\theta}_j) \rightarrow \boldsymbol{y}_j$. The training step (ii) is repeated for R "rounds", each producing increasingly more accurate approximations of the likelihood and the posterior for the given observed data, and the latter approximations are then used in the sampling step, to provide posterior samples of increasing quality. The cycle alternating between sampling and training is repeated R times, and the samples obtained at the R-th round provide the final (SeMPLE) inference. In the following, for simplicity of notation, we assume N to be an integer multiple of M, so that N/M is integer.

The SeMPLE inference scheme for mixed-effects models is described in Algorithm 1. We will see that the surrogate likelihood and the surrogate posterior are initially amortized, and then become specialized to the specific observed data $\boldsymbol{y}_o$. At first, in round $r = 0$, an initial batch of N tuples $\mathcal{D}_0 = \{(\boldsymbol{c}_j, \boldsymbol{\kappa}_j, \boldsymbol{\xi}_j), \boldsymbol{y}_j\}_{j=1}^N$ are produced from the prior-predictive distributions (steps 1-4 of Algorithm 1). Notice, here each simulated $\boldsymbol{y}_j \in \mathbb{R}^{d_o \times n}$ has the same dimension of data from a single individual $\boldsymbol{y}_o^{(i)}$, and not the dimension $\mathbb{R}^{(d_o \times n) \times M}$ of the full dataset $\boldsymbol{y}_o = \{\boldsymbol{y}_o^{(i)}\}_{i=1}^M$, that is, $\boldsymbol{y}_j$ is not a collection of M simulated individual datasets. Similarly, each $\boldsymbol{c}_j \in \mathbb{R}^u$ inside $\mathcal{D}_0$ has the same dimension of an "individual $\boldsymbol{c}^{(i)}$". The set $\mathcal{D}_0$ constitutes the initial training data for GLLiM, and with this training data we apply the methodology in Section 5 to obtain an *amortized* surro-

gate likelihood $q_{\tilde{\boldsymbol{\phi}}_0}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ (step 6) with input argument $\boldsymbol{y} \in \mathbb{R}^{d_o \times n}$, that is a generic *single-individual* data. This is a key aspect of our methodology, because after training we can plug observations from any individual $\boldsymbol{y} \leftarrow \boldsymbol{y}_o^{(i)}$ into $q_{\tilde{\boldsymbol{\phi}}_0}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$, and therefore immediately evaluate each individual likelihood $q_{\tilde{\boldsymbol{\phi}}_0}(\boldsymbol{y}_o^{(i)} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ at observed data $\boldsymbol{y}_o^{(i)}$. For extra clarity, it is useful to remark once more that in $\mathcal{D}_0$ we have stacked many individual $\{\boldsymbol{c}_j\}_{j=1}^N$ parameters, with $\dim(\boldsymbol{c}_j) = \dim(\boldsymbol{c}^{(i)}) = u$, and therefore, with some abuse of notation[1], the dimension of the $\boldsymbol{c}$ that we write inside $q_{\tilde{\boldsymbol{\phi}}_0}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ and in $q_{\boldsymbol{\phi}_0}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y})$ (step 7), is the same dimension of an "individual $\boldsymbol{c}^{(i)}$", and this feature is preserved in the next rounds of SeMPLE. In steps 6-7 of Algorithm 1 the mixture model parameters $\tilde{\boldsymbol{\phi}}_0$ are obtained (via EM within GLLiM, as in Section 5) to produce the initial amortized single-individual likelihood $q_{\tilde{\boldsymbol{\phi}}_0}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$, and then by using using (18)–(22) we immediately deduce $\boldsymbol{\phi}_0$, and hence the corresponding amortized surrogate posterior $q_{\boldsymbol{\phi}_0}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y})$ where, again, $\boldsymbol{y} \in \mathbb{R}^{d_o \times n}$ has the dimensions of single-individual data. This concludes round $r = 0$ of SeMPLE.

Next comes round $r = 1$, where steps 10-11 of Algorithm 1 aim to refine the training data by collecting simulated parameters and simulated data that are conditional to the observed $\boldsymbol{y}_o^{(i)}$. This is done by sampling N/M times from the learned amortized posterior, by first inputing $\boldsymbol{y}^{(i)} \leftarrow \boldsymbol{y}_o^{(i)}$ inside $q_{\boldsymbol{\phi}_0}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y}^{(i)})$, and then sampling $(\boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_j^{(i)}, \boldsymbol{\xi}_j^{(i)}) \sim q_{\boldsymbol{\phi}_0}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y}_o^{(i)})$ (step 10). At this stage, sampling from $q_{\boldsymbol{\phi}_0}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y}_o^{(i)})$ is trivial to do (the surrogate posterior is a Gaussian mixture model), and then, conditionally on each of the N/M posterior draws, in step 11 we produce a corresponding simulated observation $\boldsymbol{y}_j^{(i)} \sim p(\boldsymbol{y} \mid \boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_j^{(i)}, \boldsymbol{\xi}_j^{(i)})$ (implicitly from the computer simulator for (2)). Note that we use the notation $\boldsymbol{\kappa}_j^{(i)}$ and $\boldsymbol{\xi}_j^{(i)}$ to emphasize that the shared parameters $\boldsymbol{\kappa}$ and $\boldsymbol{\xi}$ are sampled from the surrogate posterior conditional on $\boldsymbol{y}_o^{(i)}$. Hence, $\boldsymbol{\kappa}_j^{(i)}$ and $\boldsymbol{\xi}_j^{(i)}$ depend only on the observation $\boldsymbol{y}_o^{(i)}$ and not on all M individuals in the data $\boldsymbol{y}_o$. The surrogate posterior samples and the simulated observations are then collected into a new training data set $\mathcal{D}_1^{(i)} = \{(\boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_j^{(i)}, \boldsymbol{\xi}_j^{(i)}), \boldsymbol{y}_j^{(i)}\}_{j=1}^{N/M}$. This is repeated for each individual i, and GLLiM is then trained on the union of training data sets from all individuals $\mathcal{D}_1 = \bigcup_{i=1}^M \mathcal{D}_1^{(i)}$, to return new estimators $\tilde{\phi}_1$ and ϕ_1, and hence a new surrogate likelihood $q_{\tilde{\boldsymbol{\phi}}_1}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ (where any observed individual $\boldsymbol{y}_o^{(i)}$ can be plugged in place of $\boldsymbol{y}$) and a corresponding surrogate posterior $q_{\boldsymbol{\phi}_1}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y})$ (steps 14-15). This concludes round $r = 1$. Training GLLiM on the full data set $\mathcal{D}_1$ is preferable to training M specialized surrogates, separately, each on a corresponding data set $\mathcal{D}_1^{(i)}$. This is preferable since a single GLLiM model needs to be trained, and this improves scalability with respect to the number of individuals M. While fitting a single GLLiM model may not be particularly computationally demanding, instead doing so M times, independently, once for each $\mathcal{D}_1^{(i)}$, could results in a very high computational effort when M is very large. In addition, the single learned GLLiM model benefits from training on a larger dataset from the same underlying model, which could improve the approximation. Individual likelihood and posterior approximations can then be obtained simply by imputing data $\boldsymbol{y}_o^{(i)}$ and parameters $\boldsymbol{c}^{(i)}$ from a specific individual i. Notice that it is possible for samples from the initial surrogate $(\boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_j^{(i)}, \boldsymbol{\xi}_j^{(i)}) \sim q_{\boldsymbol{\phi}_0}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y}_o^{(i)})$ to "leak" outside of the prior's support, if this is bounded (the leaking is not possible in later steps where a Metropolis-Hastings regularization is implemented), therefore step 10 in Algorithm 1 could also be written to include a rejection procedure whenever $\pi(\boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_j^{(i)}, \boldsymbol{\xi}_j^{(i)}) = 0$, so that the sampling is repeated until N/M parameters having $\pi(\boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_j^{(i)}, \boldsymbol{\xi}_j^{(i)}) > 0$ are collected.

Next come the remaining rounds $r = 2, \ldots, R$, and unlike for $r = 1$, posterior sampling now becomes less immediate, since the product of the surrogate likelihood with the prior is not necessarily proportional to the density of a Gaussian mixture. This is why we now incorporate a Metropolis-within-Gibbs strategy. After initializing $\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi}$ and $\boldsymbol{\eta}$ (step 16), the remaining part of the algorithm concerns obtaining further refined surrogate likelihood and posterior, where training data includes N_g samples from the Gibbs steps in eq. (23)-(25), by utilizing the surrogate likelihood $q_{\tilde{\boldsymbol{\phi}}_1}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ and posterior $q_{\boldsymbol{\phi}_1}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y})$. First, the individual parameters $\boldsymbol{c}^{(i)}$ in (23) are sampled for subject i independently of other subjects (step 19) according to an independence-Metropolis-Hastings algorithm (Robert and Casella 2004), which is detailed in Algorithm 2 and that we justify further below. Note that the surrogate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{c} \mid \boldsymbol{y}_o^{(i)})$, where the components corresponding to $\boldsymbol{\kappa}$ and $\boldsymbol{\xi}$ have been marginalized out, is being used as a *self-tuned* proposal distribution in Algorithm 2 (it is "self-tuned" because (i) the means and covariance matrices of its components have been automatically provided by the EM procedure, and (ii) it is conditional to the observed data $\boldsymbol{y}_o^{(i)}$).

Proposing independently from $q_{\boldsymbol{\phi}}(\cdot \mid \boldsymbol{y}_o^{(i)})$ is a key feature of SeMPLE: since this proposal function is a mixture model, it is particularly suited for the exploration of multimodal posteriors, and the fact that it has been derived from the same training data as for the surrogate likelihood (and has the same number of components as the mixture model of the likelihood) makes it an appropriate proposal sampler. Moreover, $q_{\boldsymbol{\phi}}(\cdot \mid \boldsymbol{y}_o^{(i)})$ is conditional on the individual-specific

[1] In fact, here $\boldsymbol{c}$ is to be interpreted as a u-dimensional real variable, and not as the collection of M individual parameters $(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(M)})$ that we defined in section 3.

$\boldsymbol{y}_o^{(i)}$, and this makes it particularly well-informed for the task of sampling specifically $\boldsymbol{c}^{(i)}$ in (23). Next, in step 20, $\boldsymbol{\kappa}$ and $\boldsymbol{\xi}$ are updated by targeting the posterior proportional to $\pi(\boldsymbol{\kappa}_{j-1}, \boldsymbol{\xi}_{j-1}) \prod_{i=1}^{M} q_{\tilde{\boldsymbol{\phi}}_{r-1}}(\boldsymbol{y}_o^{(i)} \mid \boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_{j-1}, \boldsymbol{\xi}_{j-1})$, however here the sampling is instead carried out using NUTS, and we justify this specific choice further below. To run NUTS we use `Stan` (Stan Development Team 2023) through the interfaces `RStan` (Stan Development Team 2024) and `cmdstanr` (Gabry et al. 2024). The final step of the Gibbs sampler (step 21 in Algorithm 1) does not involve surrogate likelihoods, and can therefore be easily dealt via NUTS, or can be sampled exactly by using Normal-Gamma conjugate priors, as detailed in Appendix A. We plug the newly obtained samples into the computer simulator for (2) to obtain $\boldsymbol{y}_j^{(i)} \sim p(\boldsymbol{y} \mid \boldsymbol{c}_j^{(i)}, \boldsymbol{\kappa}_j, \boldsymbol{\xi}_j), i = 1, ..., M$. The procedure is then repeated to obtain N_g samples of each parameter and corresponding simulated observations. The Gibbs samples and corresponding simulated observations for all M individuals are collected into a data set $\mathcal{D}_r = \left\{ \{\boldsymbol{c}_j^{(i)}\}_{i=1}^{M}, \boldsymbol{\kappa}_j, \boldsymbol{\xi}_j, \{\boldsymbol{y}_j^{(i)}\}_{i=1}^{M} \right\}_{j=1}^{N_g}$, and GLLiM is then trained on all data sets up until this point $\bigcup_{\tilde{r}=1}^{r} \mathcal{D}_{\tilde{r}}$ (with the exception of the prior-predictive data $\mathcal{D}_0$ which is deemed too uninformative to refine surrogates for specific observed data), to obtain an updated surrogate likelihood and posterior $q_{\tilde{\boldsymbol{\phi}}_r}(\boldsymbol{y} \mid \boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ and $q_{\boldsymbol{\phi}_r}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y})$ respectively (steps 25-26). The full Gibbs procedure is then repeated with the new surrogate likelihood and posterior until the final round R is reached.

The reason for using NUTS in step 20, instead of using the surrogate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y}_o^{(i)})$ as independence proposal distribution, is that when new values for $\boldsymbol{\kappa}$ and $\boldsymbol{\xi}$ are sampled, they need to be proposed conditional on all observations in $\boldsymbol{y}_o = (\boldsymbol{y}_o^{(1)}, \ldots, \boldsymbol{y}_o^{(M)})$. This is not compatible with the individual surrogate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y}_o^{(i)})$ that we learn with SeMPLE, as each individual surrogate distribution depends only on the observation of one individual $\boldsymbol{y}_o^{(i)}$. To circumvent this, one would need to train an additional "global" surrogate model $q_{\boldsymbol{\phi}}(\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi} \mid \boldsymbol{y}_o)$ that is conditional on the entire observed data set of M subjects. However, for increasing M, the dimension of $\boldsymbol{y}_o$ may become too large to fit a GLLiM model to "raw data" (ie non-summarized data) in this way. Hence, we use NUTS to propose new values for $\boldsymbol{\kappa}$ and $\boldsymbol{\xi}$. This problem may be relaxed if the inference was based not on the raw observed data, but on summary statistics thereof, and hence reduce the data-dimensionality problem as commonly done in approximate Bayesian computation literature. In Section 6.1 we illustrate an opportunity to obtain a much more scalable Gibbs sampler, that corresponds to a slightly different model formulation.

Notice that the number of components K may progressively reduce during several rounds of Algorithm 1, as the implementation of the EM step in the `xLLiM` package automatically removes irrelevant mixture components having probability $\tilde{\omega}_k = 0$ (within floating point accuracy).

6.1 A scalable SeMPLE approach for mixed-effects models

The approach we have described thus far offers a compelling framework for Bayesian inference in mixed-effects models by automatically constructing surrogate likelihoods and posteriors, including MCMC proposal samplers, that are both efficient to evaluate and fast to sample from. However, it is possible to obtain considerable gains in terms of computational scalability, by slightly reducing the generality of the mixed-effects model, namely not assume any shared parameters $(\boldsymbol{\kappa}, \boldsymbol{\xi})$, and instead allow these to become part of the individual parameters $\boldsymbol{c}^{(i)}$, and hence be random effects. This means that the second Gibbs step can be avoided. This approach is similar to the "perturbed" model in Persson et al. (2022); however, while they fixed the variance of certain parameters, we infer the population variance for all parameters. The three-steps Gibbs sampler (23)-(25) is replaced by the following two-steps Gibbs sampler

$$\text{step 1: } \hat{\pi}(\boldsymbol{c}^{(i)} \mid \boldsymbol{\eta}, \boldsymbol{y}^{(i)}) \propto \pi(\boldsymbol{c}^{(i)} \mid \boldsymbol{\eta})\, q_{\tilde{\phi}}(\boldsymbol{y}^{(i)} \mid \boldsymbol{c}^{(i)}), \quad i = 1, ..., M. \tag{26}$$

$$\text{step 2: } \pi(\boldsymbol{\eta} \mid \boldsymbol{c}, \boldsymbol{y}) \propto \pi(\boldsymbol{\eta}) \prod_{i=1}^{M} \pi(\boldsymbol{c}^{(i)} \mid \boldsymbol{\eta}). \tag{27}$$

This "reduced" Gibbs sampler, where $\boldsymbol{c}^{(i)} = (\cdots, \boldsymbol{\kappa}^{(i)}, \boldsymbol{\xi}^{(i)})$, has the potential to scale much better with the number of individuals in the data set. The reason why it is beneficial for scaling to remove the fixed-effects is two-fold. First, this removes the need to target the full likelihood $\prod_{i=1}^{M} q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{y}^{(i)} \mid \boldsymbol{c}^{(i)}, \boldsymbol{\kappa}, \boldsymbol{\xi})$, which is the product of all individual likelihoods. In fact, when the number of individuals M grows, the automatic differentiation tool needed to perform NUTS will have to take care of differentiating with respect to the full likelihood, which is going to be a very long expression, where the many algebraic operations for its computation involve large matrices such as $\tilde{\boldsymbol{A}}_k$, $\tilde{\boldsymbol{\Sigma}}_k$ etc. Evaluating the gradient of such a large expression at every proposed parameter, will cause a considerable computational overhead. Of course, the complexity would be greatly reduced if the inference was based on data-summarization $S(\boldsymbol{y})$ rather than $\boldsymbol{y}$, see the end of Section 2, since the dimensions of eg. $\tilde{\boldsymbol{A}}_k$ and $\tilde{\boldsymbol{\Sigma}}_k$ depend on the dimensions of $\boldsymbol{y}$. Secondly, when using NUTS to propose fixed-effects $\boldsymbol{\kappa}$ and $\boldsymbol{\xi}$ in (24), it is not possible to utilize one of the main benefits of our methodology, which is the self-tuning proposal sampler (and surrogate posterior) $q_{\boldsymbol{\phi}}$, that has desirable abilities to explore multimodal surfaces (Häggström et al. 2024). On the other hand, when

considering all parameters to be individual as in (26)-(27), the proposal sampler $q_{\boldsymbol{\phi}}$ can be used to propose all the $\boldsymbol{c}^{(i)}$ (which include $(\boldsymbol{\kappa}^{(i)}, \boldsymbol{\xi}^{(i)})$).

7 Examples on simulated and real biological data

We evaluate the performance of SeMPLE for mixed-effects models through three case studies based on two models. First, we examine a state-space SDEMEM with latent dynamics driven by Ornstein-Uhlenbeck SDEs. This example is particularly relevant, as exact Bayesian inference is feasible without relying on pseudomarginal methods, providing a "gold-standard" reference posterior for comparison with SeMPLE inference. Next, we present two case studies where the reference posterior is obtained using a pseudomarginal (particle MCMC) sampler. These examples utilize the SDE model from Pieschner et al. (2022), which describes translation kinetics following mRNA transfection. One case involves simulated data, and in another case we use real data from Fröhlich et al. (2018). The model is two-dimensional, with only one observed component, and SeMPLE inference is compared to exact Bayesian (pseudomarginal) inference obtained using the PEPSDI framework (Persson et al. (2022)). PEPSDI (`Particles Engine for Population Stochastic DynamIcs`) provides a useful benchmark as, to the best of our knowledge, PEPSDI is the only software implementing pseudomarginal schemes for state-space SDEMEMs, including diagnostics for determining an appropriate number of particles, and several proposal kernels for pMCMC.

7.1 Ornstein-Uhlenbeck state-space model

The Ornstein-Uhlenbeck process is defined by the following SDE,

$$dX_t^{(i)} = c_1^{(i)}(c_2^{(i)} - X_t^{(i)})dt + c_3^{(i)} dW_t^{(i)}, \tag{28}$$

where the $\{W_t^{(i)}\}_{t\geq 0}$ are independent Wiener processes. In this example, $\{X_t^{(i)}\}_{t\geq 0}$ is one-dimensional. We consider the following state-space model, for $i = 1, ..., M$:

$$\begin{cases} dX_t^{(i)} = c_1^{(i)}(c_2^{(i)} - X_t^{(i)})dt + c_3^{(i)} dW_t^{(i)} \\ Y_t^{(i)} \quad = X_t^{(i)} + \epsilon_t^{(i)}, \quad \epsilon_t^{(i)} \sim \mathcal{N}(0, \xi^2), \end{cases} \tag{29}$$

where $\{Y_t\}_{t\geq 0}$ is the observed process. The transition densities for the latent dynamics are known, and hence the Euler-Maruyama discretization is not needed to simulate the SDE (28) numerically. Instead we use the following exact simulation scheme, which is induced by the exact transition density

$$X_{t+\Delta_t}^{(i)} = c_2^{(i)} + (X_t^{(i)} - c_2^{(i)})e^{-c_1^{(i)}\Delta_t} + c_3^{(i)} \sqrt{\frac{(1 - e^{-2c_1^{(i)}\Delta_t})}{2c_1^{(i)}}} \times u_t^{(i)}, \tag{30}$$

where $u_t^{(i)} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$.

7.1.1 Inference setup

We consider an inference setting similar to the one in Wiqvist et al. (2021) and Persson et al. (2022). Data were simulated for $M = 40$ individuals at 50 equidistant time points from $t = 0.2$ to $t = 10$ ($\Delta_t = 0.2$), and with initial value $X_0 = 0$ at $t = 0$. We set a Gaussian population distribution $\log(c_1^{(i)}, c_2^{(i)}, c_3^{(i)}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\tau}^{-1})$, where the true data generating values for the population parameters were set to $\boldsymbol{\mu} = (-0.7, 2.3, -0.9)$ and $\boldsymbol{\tau} = (4, 10, 4)$. Similarly to Wiqvist et al. (2021), the prior of $\boldsymbol{\eta} = (\mu_1, \mu_2, \mu_3, \tau_1, \tau_2, \tau_3)$ was set to be $\pi(\boldsymbol{\eta}) = \prod_{j=1}^{3} \pi(\mu_j|\tau_j)\pi(\tau_j)$, where the $\pi(\mu_j|\tau_j)$ and the $\pi(\tau_j)$ are in equation (31):

$$\begin{cases} \mu_j|\tau_j \sim \mathcal{N}(\mu_{0_j}, (\lambda_j\tau_j)^{-1}), \qquad j = 1, 2, 3, \\ \tau_j \sim Ga(\alpha_j, \beta_j), \end{cases} \tag{31}$$

where the Gamma distribution is parameterized by the shape α_j and rate β_j. The prior parameter values can be found in Supplementary Material Table S1. The "Normal-Gamma" prior (31) allows us to benefit from conjugacy, when sampling $\boldsymbol{\eta}$ directly from a Normal-Gamma distribution in the third Gibbs sampler step. The prior of $\boldsymbol{\tau}$ is shifted a bit from the setup in Wiqvist et al. (2021), to avoid having small precisions resulting in a large variance in $\boldsymbol{\mu}$, and consequentially unreasonable prior-predictive simulated data. The data-generating value of the noise parameter is $\log(\xi) = -1.2$, and we used as prior $\log(\xi) \sim \mathcal{N}(0, 1)$.

7.1.2 Settings for SeMPLE

We first determined an appropriate number of mixture components K, using the BIC criterion (Appendix B), see Figure 8. Consequently, we set the starting number of components to be $K = 10$. The covariance matrices $\boldsymbol{\Sigma}_k$ and $\tilde{\boldsymbol{\Sigma}}_k$ in the GLLiM models were set to be unconstrained, i.e. fully parameterized. The number of prior predictive samples is the same as the number of Gibbs samples, i.e. we set $N = N_g = 50,000$, and the number of SeMPLE round is $R = 4$.

7.1.3 Results

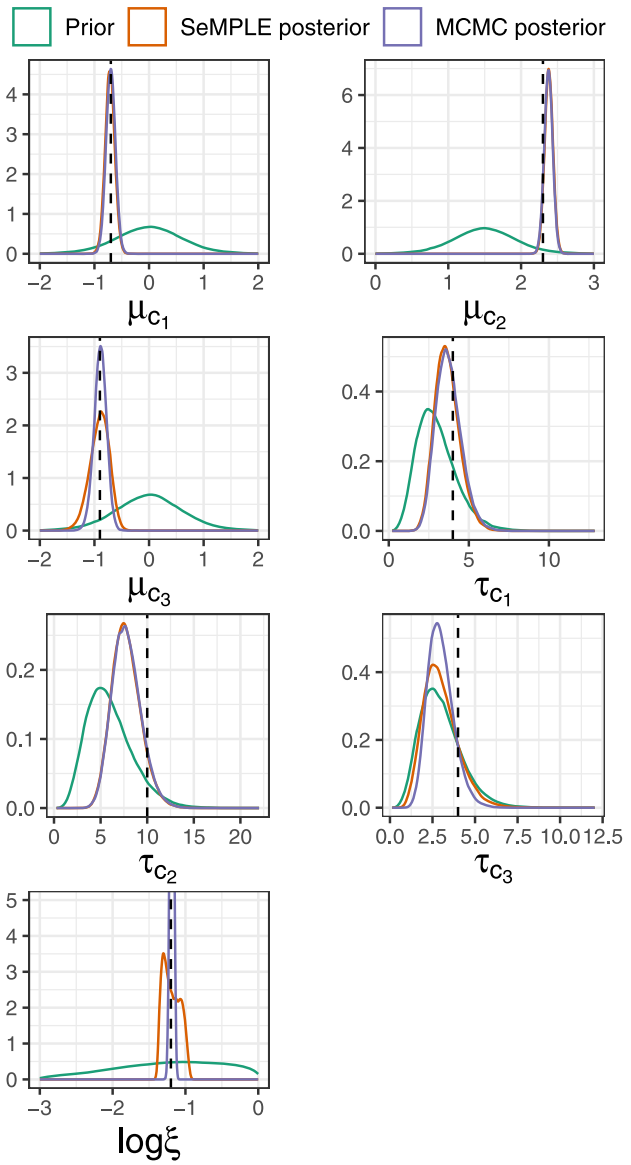

Fig. 2 Ornstein-Uhlenbeck: marginal posteriors from 10k posterior samples from MCMC using the exact likelihood (purple) and round $r = 4$ of SeMPLE (orange). Priors are in green. The dashed lines are the true parameter values

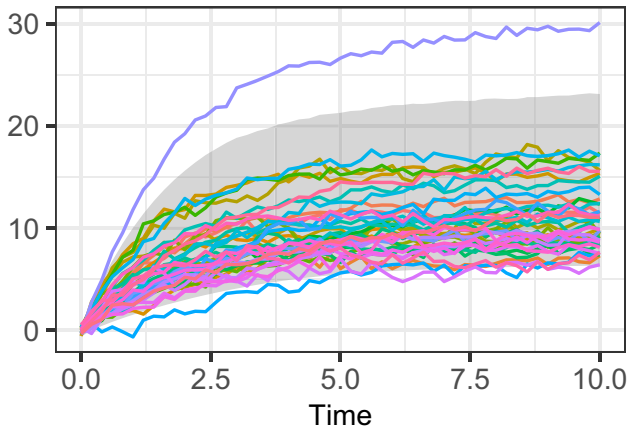

Fig. 3 Ornstein-Uhlenbeck: Posterior-predictive simulations from SeMPLE ($r = 4$) and data (colored lines, 40 individuals). In grey is the area between the 2.5th and 97.5th percentile from 10k posterior-predictive simulations obtained from SeMPLE

The prior and marginal posteriors for the "common parameters" (i.e. shared between subjects) $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ (Figure 2) are obtained using both SeMPLE and exact Bayesian inference (MCMC), where the latter does not employ particle methods. Exact inference is obtained using the three steps of the Gibbs sampler, where the exact likelihood provided by the Kalman filter is used in the Metropolis-within-Gibbs steps (7)-(8). Therefore, exact inference is used as a reference to asses the accuracy of the SeMPLE approach in Algorithm 1. For exact inference, 200k posterior samples were produced in total, where the first 150k were discarded as burn-in samples, and the remaining 50k samples were used for inference. Inference results obtained with SeMPLE (Figure 2) also used 50k posterior samples. Figure 2 reports results from both methods, and we conclude that inference obtained with SeMPLE is overall very satisfying, as even where differences from exact inference appear visually more marked, see the posterior for $\log \boldsymbol{\xi}$, in practical terms these differences are tiny, considering the magnitude of the observations in Figure 3. Moreover, posterior predictive checks in Figure 3 confirm the quality of the inference. Extra results are in Supplementary Material section S3.1: there, traceplots of the SeMPLE posterior samples show the excellent mixing of the chains from the last round of SeMPLE. Moreover, inference at the individual's level is also excellent, namely, SeMPLE captures the true values of the data-generating $\boldsymbol{c}^{(i)}$'s. We have also experimented with different numbers of individuals M ($M = 20$ in Figure S5 and $M = 100$ in Figure S6): as expected, the marginal posteriors contract around the true data generating parameters, and in particular, for $M = 100$, we do learn more about the population precisions $\boldsymbol{\tau}$.

7.2 mRNA transfection model

We consider the SDE model in a simulation study originally used to describe translation kinetics following mRNA transfection (Pieschner et al. (2022)). The dataset consists of time-lapse microscopy images capturing the fluorescence intensity of cells over at least 30 hours, with measurements taken every 10 minutes (Fröhlich et al. (2018)). During the first hour, cells were incubated with mRNA lipoplexes before being washed to prevent further uptake. As the released mRNA is translated into a green fluorescent protein (GFP), the cell fluorescence is tracked over time. The SDE below is used to study the translation kinetics of one cell, based on the observed fluorescence trajectory of individual cells

after transfection with mRNA encoding for GFP, where the two-dimensional stochastic process $(m(t), p(t))$ where $m(t)$ represents the amount of mRNA molecules at time t, and $p(t)$ is the amount of GFP molecules at time t. The SDEMEM is given by

$$d\begin{pmatrix} m^{(i)} \\ p^{(i)} \end{pmatrix}(t) = \begin{pmatrix} -\delta^{(i)} \cdot m(t)^{(i)} \\ k^{(i)} \cdot m^{(i)}(t) - \gamma^{(i)} \cdot p^{(i)}(t) \end{pmatrix} dt + \begin{pmatrix} \sqrt{\delta^{(i)} \cdot m^{(i)}(t)} & 0 \\ 0 & \sqrt{k^{(i)} \cdot m^{(i)}(t) + \gamma^{(i)} \cdot p^{(i)}(t)} \end{pmatrix} dB_t^{(i)}$$
$$(\delta^{(i)}, \gamma^{(i)}, k^{(i)}) \sim \pi(\delta, \gamma, k \mid \boldsymbol{\eta}), \quad i = 1, ..., M, \tag{32}$$

where the $B_t^{(i)}$ are two-dimensional standard Brownian motions. It is assumed that all mRNA molecules (within one cell) are released at once from the lipoplexes and denote this initial time point by t_0. Before t_0, there are neither mRNA nor GFP molecules, and at t_0, an amount m_0 of mRNA molecules is released, i.e

$$\begin{pmatrix} m^{(i)}(t) \\ p^{(i)}(t) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{for } t < t_0$$

while $m^{(i)}(t_0) = m_0$ and $p^{(i)}(0) = 0$. We take as observable mapping

$$y^{(i)}(t_j) = \log(\text{scale} \cdot p^{(i)}(t_j) + \text{offset}) + \varepsilon^{(i)}(t_j), \tag{33}$$

where $\varepsilon^{(i)}(t_j)$ is iid Gaussian measurement error $\varepsilon^{(i)}(t_j) \sim \mathcal{N}(0, \sigma^2)$, $j = 1, ..., n$, and $t_1, ..., t_n$ are observational time instants (see further below). Note that the observations depend only indirectly on process $\{m^{(i)}(t)\}$, and only $\{p^{(i)}(t)\}$ is observed. The model parameters (δ, γ, k) are treated as random effects parameters that vary between individuals, i.e. $\boldsymbol{c}^{(i)} = (\delta^{(i)}, \gamma^{(i)}, k^{(i)})$. The remaining ones, $(m_0, \text{scale}, \text{offset}, \sigma)$, are treated as fixed but unknown parameters, and we set $\boldsymbol{\kappa} = (m_0, \text{scale}, \text{offset})$ and $\boldsymbol{\xi} = \sigma$. The choice of individual parameters and individual-constant parameters is similar to Arruda et al. (2024). In Section 7.2.8 we also consider different assumptions where all parameters are set to be random effects, and discuss implications in terms of scalability.

7.2.1 Inference setup for simulated data

In this example, we fix $t_0 = 0$ and do not infer it. This choice enables a direct comparison with the pseudomarginal (particle MCMC) inference produced by the PEPSDI framework (Persson et al. 2022), which does not currently support inference of the initial simulation time t_0.

However, in the real-data case-study in next sections, we infer also t_0 (only with SeMPLE). Inference is performed on the log-scale and we assume $\log(\boldsymbol{c}^{(i)}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\tau}^{-1})$, where the true values used to generate data were set to $\boldsymbol{\mu} = (-0.694, -3, 0.027)$ and $\boldsymbol{\tau} = (10, 10, 10)$. It has been shown that SDE modelling improves identifiability of parameters, compared to using a corresponding ODE model (Pieschner et al. 2022). Nevertheless, some parameters are still unidentifiable (for details see supplementary section A.4.1 in Pieschner et al., 2022).

Here the exact transition densities are unavailable, and solutions to the SDE (32) are simulated using an Euler-Maruyama scheme implemented in `Rcpp` (Eddelbuettel et al. 2024) from $t = t_0$ to $t = 30$, with step size 0.01. The observed time series is interpolated from the Euler-Maruyama approximation at $n = 60$ equidistant time points from $t_1 = 0.5$ to $t_n = 30$. We simulate $M = 40$ individuals according to this setup. The prior distributions are set to be Normal-Gamma for the random effects and Gaussian on the log-scale for the fixed effects. Prior parameters can be found in Supplementary Material section S3.2.

7.2.2 Settings for SeMPLE

The starting number of mixture components for the mixture models was set to $K = 7$, according to the Bayesian information criterion (Figure 8), with covariance matrices for the mixture components specified to be full and unconstrained. The number of prior-predictive samples and Gibbs samples is $N = N_g = 50,000$ and the number of SeMPLE rounds is $R = 4$.

7.2.3 Setup for the pseudomarginal method PEPSDI

To assess the quality of the approximate inference obtained by SeMPLE, we run PEPSDI with the setup described in section 7.2.1. To avoid typical initialization problems affecting MCMC, when setting starting parameters far from the bulk of the posterior, we initialize PEPSDI at the same (true) parameter values used to generate the observed data. We produce 50,000 posterior samples, which is the same number of posterior samples produced with SeMPLE. The number of particles was set to 150 for every individual, to reduce the variance of the likelihood estimations and ensure accurate posterior inference. The initialization of the proposal covariance matrix for the fixed-effects $(\boldsymbol{\kappa}, \boldsymbol{\xi})$ was tuned manually to improve the mixing of the Markov chains of these parameters.

7.2.4 Results from simulated data

We compare posterior distributions based on 50,000 posterior samples from both PEPSDI and SeMPLE. The posterior distributions of the population means $\boldsymbol{\mu}$ and population precisions $\boldsymbol{\tau}$ returned by SeMPLE are virtually identical to exact (pseudomarginal) inference returned by PEPSDI (Fig-

ure 4). The same applies to the offset parameter, and while the marginal posterior for the measurement error's standard deviation σ is slightly wider than the PEPSDI posterior, we have noticed that posterior predictive simulations resulting from this are virtually indistinguishable from the observed data (plot not reported). The posterior plots indicate that the m_0 and the "scale" parameters are more challenging to infer, and this is not specific to SeMPLE, in fact the traceplots from PEPSDI for m_0 and scale (Figure S9) show difficulties with mixing, and therefore for these two parameters the comparison between SeMPLE and PEPSDI should be taken with a grain of salt. Such difficulty is also evident in the corresponding posterior plots in Arruda et al. (2024). In addition to the posterior density plots, posterior predictive checks for several exemplary individuals are given in Supplementary Material. The PEPSDI runtime to produce 50,000 posterior samples was approximately 149 hours, using the artificially favorable setup where we avoided investing time in the search for a suitable starting value for the parameters. The SeMPLE runtime throughout the $R = 4$ rounds was 77.4 hours. However, we have verified that already at $r = 2$ SeMPLE was producing accurate inference in only 10.4 hours (see Figure S12). In addition to this, the determination of the initial K via the BIC took 6 minutes. A comparison between effective sample sizes (ESS, the higher the better) can be found in Table S3. The univariate ESS was computed using the R-package `LaplacesDemon` (Statisticat and LLC. 2021), and we also reported the multivariate ESS, via the R-package `mcmcse` (Flegal et al. 2025). Table S3 shows that, overall, the ESS from SeMPLE are much larger than with pseudo-marginal (PEPSDI) inference, and that nearly independent samples are obtained much faster with SeMPLE (around five times faster). We emphasize that results from a slightly different model can be obtained much more rapidly via SeMPLE, see Section 7.2.8, corresponding to the two-steps Gibbs described in Section 6.1.

7.2.5 Inference setup with real data

We now use the same SDEMEM as in the simulated data example to test our method on the real-world dataset from Fröhlich et al. (2018), to which we refer the reader for details regarding the experimental procedures. This data has also been analysed in Pieschner et al. (2022), where the initial ODE model was extended to an SDE model with improved parameter identifiability, however in Pieschner et al. (2022) no new inference methodology was presented. We extend their work by applying our novel inference methodology. We consider the first $M = 40$ individuals from the data set labeled "20164027_mean_eGFP" in Fröhlich et al. (2018). The raw data is then log-transformed in accordance with the observable mapping (33). Note that the real data set has 180 measurements for each individual, as opposed to the

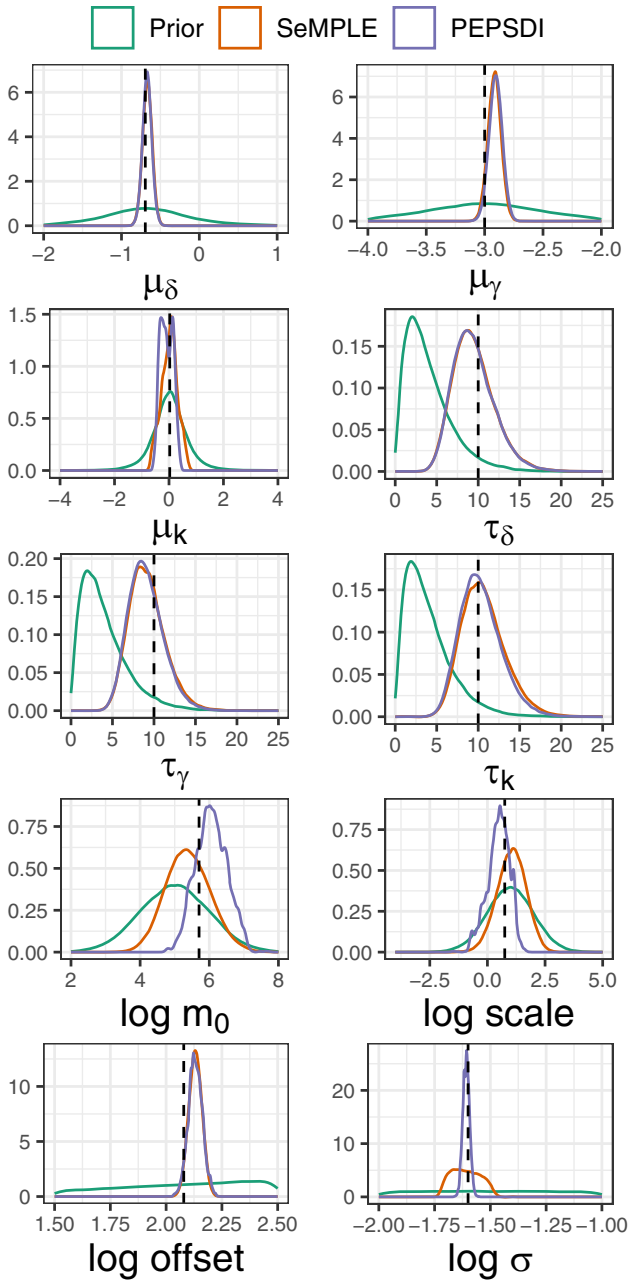

Fig. 4 mRNA model with 40 simulated individuals: marginal posteriors obtained with SeMPLE (orange, round $r = 4$) and with PEPSDI (purple). Priors are in green. The dashed lines are the true parameter values that were used to generate the observed data

60 measurements in the simulated data setup described in Section 7.2.1. To allow more flexibility, the prior distribution of the population parameters are set to be independent Gaussian for the mean $\boldsymbol{\mu}$ and Gamma-distributed precision $\boldsymbol{\tau}$, instead of a prior Normal-Gamma distribution. The prior parameters can be found in Supplementary Material section S4. Consequentially, it is no longer possible to sample the population parameters $\boldsymbol{\eta}$ directly, since we cannot exploit conjugacy here. Instead, we use NUTS to sample the popu-

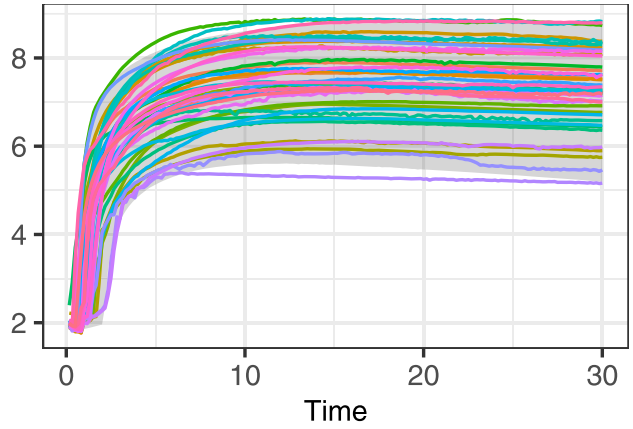

Fig. 5 mRNA model with real data: posterior-predictive simulations for 40 individuals using SeMPLE ($r = 4$, colored lines are observed data). In grey is the area between the 2.5th and 97.5th percentile from 1, 000 posterior-predictive simulations obtained from SeMPLE

lation parameters efficiently in the Gibbs sampler. Note that, as opposed to the setup with simulated data, we now infer the initial time point t_0 as a parameter that varies between individuals, i.e. $\boldsymbol{c}^{(i)} = (\delta^{(i)}, \gamma^{(i)}, k^{(i)}, t_0^{(i)})$; however we can only do so via SeMPLE, as PEPSDI currently does not allow to infer the starting time t_0, and therefore we cannot compare results from both methods. This is not a problem per-se, as we have already shown such a comparison and the SeMPLE reliability in the simulation study.

7.2.6 Settings for SeMPLE with real data

Similarly to the setting with simulated data in section 7.2.2, we set the GLLiM covariance matrices for the mixture components to be full and unconstrained. The number of mixture components was set to $K = 9$ according to the BIC (Figure 8). The number of prior-predictive samples was set to $N = 50,000$, we produce $N_g = 1,000$ posterior Gibbs samples, and the number of SeMPLE rounds is $R = 4$.

7.2.7 Results from real data example

To validate the quality of the inference, we provide posterior predictive simulations for all individuals (Figure 5). The uncertainty about the observed dynamics is well captured and the posterior inference is consistent with the observed data. Additional individual posterior predictive simulation plots can be found in the Supplementary Material section S4. The SeMPLE runtime was 24 hours. In addition to this, the runtime to determine the initial K via the BIC for this setup was 33 minutes. Note that the number of posterior samples was reduced, compared to the simulated data setup, to reduce the runtime.

7.2.8 Scalable approach using exclusively random-effects

As explained in Section 6.1, a major computational bottleneck in the framework presented so far is the second step of the Gibbs sampler (24), where the full likelihood (the product of all individual likelihoods) is used within NUTS. Here we consider inference with the two-steps "reduced" Gibbs sampler from Section 6.1. The population distribution for the translation kinetics model after mRNA transfection can then be rewritten as

$$\log(\delta^{(i)}, \gamma^{(i)}, k^{(i)}, t_0^{(i)}, m_0^{(i)}, \text{scale}^{(i)}, \text{offset}^{(i)}, \sigma^{(i)}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\tau}^{-1}), \quad i = 1, ..., M. \tag{34}$$

Therefore, compared to previous analyses, here ($m_0^{(i)}$, $\text{scale}^{(i)}$, $\text{offset}^{(i)}$, $\sigma^{(i)}$) are also random effects. We run SeMPLE with the same $M = 40$ individuals from the real data set as in Section 7.2.7. The runtime to produce 1, 000 posterior samples is 1.3 hours, a significant reduction compared to the corresponding runtime with fixed-effects of 24 hours in section 7.2.7. For the purpose of displaying inference results, Figure S15 shows marginal posterior distributions based on 10, 000 posterior samples. We note that for many of the population precisions $\boldsymbol{\tau}$, the posterior distributions are similar to the priors, except for the population precisions τ_{t_0} and τ_{offset}. This suggests that a data set of 40 individuals is not large enough to infer the population variance of all the model parameters, which we confirm in the next section, where we use larger values for M.

7.2.9 Runtime scaling

To further investigate how the runtime of the SeMPLE algorithm scales with the number of individuals M, we perform a simulation study for increasing M. We report the run times both with the most general SeMPLE algorithm for the three-steps Gibbs approach (with setup described in Section 7.2.5), and the corresponding setup with the scalable approach described in Section 7.2.8. However, to ease the calculations, here we use simulated data with 60 observations for each individual. The number of prior predictive samples is set to $N = 10,000$, and the number of Gaussian mixture components was set to $K = 10$ for both algorithm versions to make comparisons fair, even though previous BIC results (Figure 8) suggests a smaller K could be used. We measure the wall-clock runtime to obtain 1, 000 posterior samples from the corresponding Gibbs sampler, and this includes the training of the surrogate likelihood both on prior-predictive data and on samples from the surrogate posterior (Figure 6). Here, the "scalable approach" two-steps Gibbs is denoted with "only random-effects", and the three-steps Gibbs with "with fixed-effects". For the scalable approach the runtime

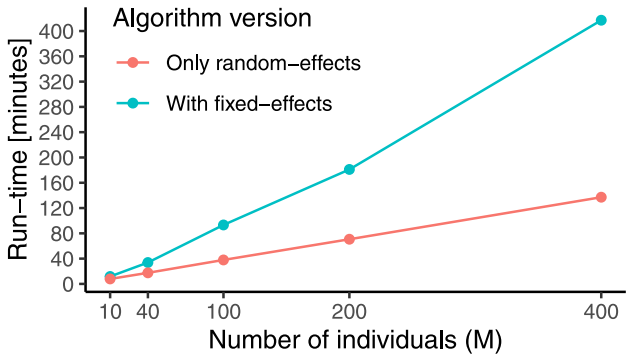

Fig. 6 SeMPLE runtime to obtain 1, 000 posterior samples as a function of the number of individuals. The blue line corresponds to the algorithm version that also include constant parameters (fixed-effects) and the red line to the version with only individual parameters (random-effects)

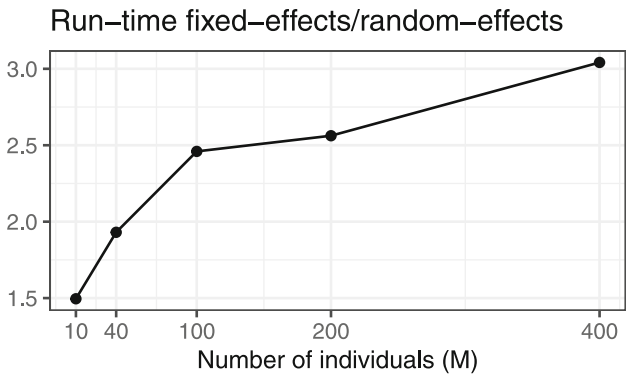

Fig. 7 Ratio of SeMPLE runtimes for a model that include fixed-effects and the one having only random-effects, corresponding to the data in Figure 6

clearly scales linearly with the number of individuals. This is to be expected since the sampling via Gibbs (i.e. excluding the mixtures fitting) typically makes up the majority of the runtime, and this scales linearly with M when no parallelization is performed. The runtime of fitting the surrogate models is typically negligible in comparison to the Gibbs sampling. Figure 7 shows the ratio between the runtimes from Figure 6, displaying the acceleration achieved when running the two-steps Gibbs instead of the three-steps Gibbs approach.

Connecting to the conjecture that the number of individuals ($M = 40$) in the data set was not large enough to infer the population precision (Figure S15), we refer to a simulation study in Supplementary Material using $M = 200$ (Figure S13). With $M = 200$ the posterior distributions of the population precisions are now very informative about the location of the true parameter values.

8 Discussion and conclusion

This study introduces a novel simulation-based (Bayesian) inference method for stochastic nonlinear mixed-effects models. More specifically, we focused on mixed-effects driven by SDEs (SDEMEMs). The method addresses the challenge of providing flexible, yet computationally efficient, methodology in this setting. Our SeMPLE methodology builds amortized approximations of the intractable likelihood and of the posterior using Gaussian mixtures of experts, and then proceeds at refining such approximations for given observed data, without using neural conditional estimation, unlike in state-of-the-art SBI methodology. This comes with some advantages, namely easier analytic tractability, the use of well-studied ad-hoc algorithms for their fitting (expectation-maximization), and finally the estimators obtained from a specific conditional density (say the likelihood function), when this is expressed via a Gaussian mixture, they can easily be transferred to other conditional densities (eg the posterior) using closed-form algebraic operations.

For the case studies we considered, including a SDEMEM for translation kinetics model after mRNA transfection, we compared our inference against gold-standard Bayesian inference, either using exact or asymptotically exact (pseudo-marginal) MCMC samplers. In all cases we show that with SeMPLE we obtain inference that is very similar to exact Bayes, but with the additional advantage of allowing inference for a much larger number of individuals than could be possible with particle-based MCMC approaches, as further discussed below. In terms of generality, SeMPLE provides full Bayesian inference with the option of treating any model parameter as either a fixed or a random effect, unlike methods where fixed effects are modelled as random effects with zero variance (Arruda et al. 2024). In addition, SeMPLE allows to infer the time point t_0 of the discrete jump in the SDE solution as a model parameter, instead of being forced to assume t_0 as known, a limitation present in both Pieschner et al. (2022) and Persson et al. (2022).

We have explored the scalability of our methodology with respect to an increasing number of individuals M (Section 7.2.9), and we have provided an additional (and slightly less general) version of our method that has an improved scalability (Section 7.2.8). We have shown that, with the more scalable version of SeMPLE, we can fit SDEMEMs to several hundreds of individuals using a standard laptop, which is particularly notable, and would not have been possible with the pseudomarginal particle MCMC (pMCMC) approaches in, e.g., Persson et al. (2022), without considerable tuning (e.g. tuning for the number of particles, and the usage of "guided" paths-solutions for the particle filters). Moreover, pMCMC is not an amortized approach and requires reruns for every new considered dataset. Still, for SeMPLE there is

room for improvement in terms of the scalability in the number of individuals M. For example, fully Bayesian inference for SDEMEMs, with several thousands of individuals, can be computationally demanding for SeMPLE when running on a standard laptop, but could of course be accommodated on a computer cluster. We note that the Gibbs sampler step with the individual parameters in (23) allows for parallelization, and with a large number of individuals this could reduce the runtime significantly. Another methodology, which targets scalability but using amortized neural density estimators, is in Arruda et al. (2024). However, in Arruda et al. (2024) the focus is not on fully Bayesian inference (even though a brief demonstration with Bayesian approaches is given), but instead on maximum likelihood estimation and uncertainty quantification through profile likelihood analysis. It is difficult to construct inference methods for SDEMEMs that are computationally efficient without making simplifications at the cost of generality. In this regard, we identify a need for further research in this field while providing a significant contribution in this direction.

Appendix A: Conjugate priors for the Ornstein-Uhlenbeck model

For the Ornstein-Uhlenbeck model we use conjugate priors, as a matter of convenience, as it makes it easier to compare the results of SeMPLE with the gold standard inference obtained using the Kalman filter. With conjugate priors we can sample explicitly from equation (25), and this is achieved by setting a Normal-Gamma prior distribution on the population parameters $\boldsymbol{\eta}$, as in the following

$$\begin{cases} \mu_j|\tau_j \sim \mathcal{N}(\mu_{0_j}, (\lambda_j\tau_j)^{-1}), & j = 1, 2, 3 \\ \tau_j \sim Ga(\alpha_j, \beta_j). \end{cases} \tag{A1}$$

We denote the latter prior by Normal-Gamma($\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \boldsymbol{\beta}$), and we let the population distribution be a Gaussian $\pi(\boldsymbol{c} \mid \boldsymbol{\eta}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\tau}^{-1})$. The distribution that we want to sample from in (25) is therefore a Normal-Gamma distribution (Murphy 2007) with the following parameters

$$\text{NG}\Big(\frac{\lambda_j\mu_{0j} + M\bar{\boldsymbol{c}}_j}{\lambda_j + M}, \lambda_j + M, \alpha_j + M/2, \\ \beta_j + \frac{1}{2}\sum_{i=1}^{M}(c_j^{(i)} - \bar{\boldsymbol{c}}_j)^2 + \frac{M\lambda_j}{\lambda_j + M}\frac{(\bar{\boldsymbol{c}}_j - \mu_{0j})^2}{2}\Big).$$

Appendix B: Determination of K

The number K of components in the Gaussian mixture model has to be specified prior to performing the EM procedure. A too large value of K may result in overfitting and unnecessary computational effort, while a too small value of K may limit the ability to represent the relationship between $\boldsymbol{\theta}$ and $\boldsymbol{y}$. The Bayesian Information Criterion (BIC), used in Deleforge et al. (2014) and Häggström et al. (2024) to guide the selection of K, is given by

$$BIC = -2\mathcal{L}(\hat{\boldsymbol{\phi}}) + D(\tilde{\boldsymbol{\phi}}) \log N$$

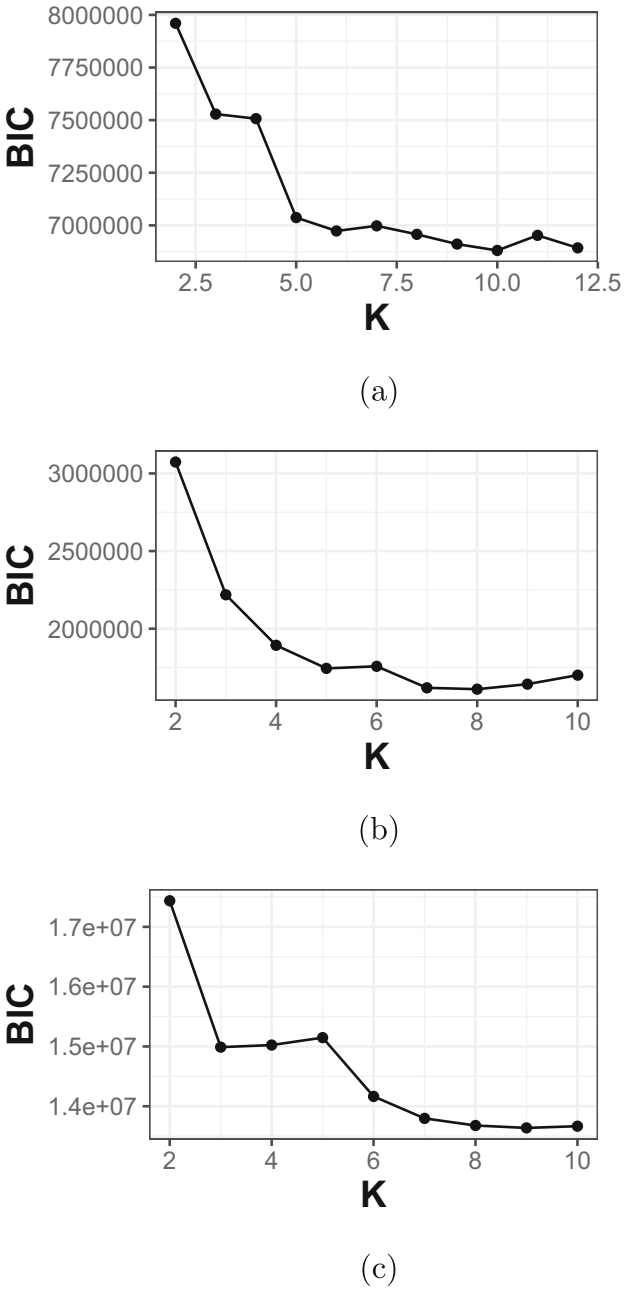

Fig. 8 Bayesian information criterion as a function of K. (a) Ornstein-Uhlenbeck model. (b) mRNA model with simulated data. (c) mRNA model with real data (40 subjects)

where $\mathcal{L}(\hat{\boldsymbol{\phi}})$ is the maximized value of the GLLiM log-likelihood function at the MLE $\hat{\boldsymbol{\phi}}$, $D(\tilde{\boldsymbol{\phi}})$ is the total number of parameters in the model and N is the number of observations in the training dataset. We wish to select a K returning a small BIC, when GLLiM is fitted with K components to a training dataset $\{\boldsymbol{\theta}_n, \boldsymbol{y}_n\}_{n=1}^N$ obtained by independently sampling parameters $\boldsymbol{\theta}_n \sim p(\boldsymbol{\theta})$ from the prior, and by simulating the corresponding $\boldsymbol{y}_n \sim p(\boldsymbol{y}|\boldsymbol{\theta}_n)$ from the generative model. When the training data is the set of the N independent $\{\boldsymbol{\theta}_n, \boldsymbol{y}_n\}_{n=1}^N$, the GLLiM log-likelihood is given by

$$\mathcal{L}(\tilde{\boldsymbol{\phi}}) = \sum_{n=1}^{N} \log q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{y}_n, \boldsymbol{\theta}_n),$$

with

$$q_{\tilde{\boldsymbol{\phi}}}(\boldsymbol{y}_n, \boldsymbol{\theta}_n) = \sum_{k=1}^{K} \mathcal{N}(\boldsymbol{y}_n; \tilde{\boldsymbol{A}}_k\boldsymbol{\theta}+\tilde{\boldsymbol{b}}_k, \tilde{\boldsymbol{\Sigma}}_k)\, \mathcal{N}(\boldsymbol{\theta}_n; \tilde{\boldsymbol{v}}_k, \tilde{\boldsymbol{\Gamma}}_k)\, \pi_k\ .$$

Note that within our framework $\boldsymbol{\theta} = (\boldsymbol{c}, \boldsymbol{\kappa}, \boldsymbol{\xi})$ and thus $l = q + p + s$, where $\boldsymbol{\theta} \in \mathbb{R}^l$, and $\boldsymbol{y} \in \mathbb{R}^{d_o \times n}$, where d_o is the dimension of the observation at a specific time point and n is the number of time points in the observation. The number of parameters that GLLiM needs to estimate is

$$\begin{aligned} D(\tilde{\boldsymbol{\phi}}) =& (K-1) + K\big(d_o n(q+p+s) + d_o n + \\ & (q+p+s) + \mathrm{nbpar}_\Sigma + \mathrm{nbpar}_\Gamma\big), \end{aligned} \tag{B2}$$

where nbpar_Σ and nbpar_Γ are the number of parameters in the covariance matrices $\tilde{\Sigma}_k$ and $\tilde{\Gamma}_k$, respectively. The covariance structure of the matrices $\tilde{\Sigma}_k$ and $\tilde{\Gamma}_k$ can be constrained to reduce the number of parameters that GLLiM needs to estimate. The `xLLiM` package (Perthame et al. 2022), that we use to run Expectation-Maximization when fitting GLLiM models, allows the $\tilde{\boldsymbol{\Sigma}}_k$'s to be set as isotropic, diagonal or full matrices, and set all equal or varying with k. The setup we considered for the covariance matrices can be found in the sections pertaining each of the considered examples.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11222-026-10850-8.

Acknowledgements HH, UP and MC acknowledge support from the Swedish Research Council (Vetenskapsrådet 2019-03924 and 2023-04319). SP and MC acknowledge support from the Swedish Foundation for Strategic Research (FFL15-0238). UP acknowledges support from the Chalmers AI Research Centre (CHAIR). The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

Author contributions H.H. developed the methodology, wrote a substantial part of the manuscript, wrote all the code and prepared all the figures. S.P. contributed to the interpretation of results, assisted with specific sections of the manuscript and provided support for the coding and the execution of the simulations with PEPSDI. M.C. provided feedback and some editorial contributions. U.P. conceptualized the study, conceived the core statistical methodology and wrote a substantial part of the manuscript. All authors reviewed the manuscript.

Funding Open access funding provided by University of Gothenburg.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

Andrieu, C., Doucet, A., Holenstein, R.: Particle Markov chain Monte Carlo methods. J. R. Stat. Soc. Ser. B Stat Methodol. **72**(3), 269–342 (2010)

Andrieu, C., Roberts, G.O.: The pseudo-marginal approach for efficient Monte Carlo computations. Ann. Stat. **37**(2), (2009)

Arruda, J., Schälte, Y., Peiter, C., Teplytska, O., Jaehde, U., Hasenauer, J.: An amortized approach to non-linear mixed-effects modeling based on neural posterior estimation. In: Proceedings of the 41st International Conference on Machine Learning **235**, 1865–1901 (2024)

Åkesson, M., Singh, P., Wrede, F., Hellander, A.: Convolutional neural networks as summary statistics for approximate Bayesian computation. IEEE/ACM Trans. Comput. Biol. Bioinf. **19**(6), 3353–3365 (2021)

Botha, I., Kohn, R., Drovandi, C.: Particle methods for stochastic differential equation mixed effects models. Bayesian Anal. **16**(2), 575–609 (2021)

Beskos, A., Papaspiliopoulos, O., Roberts, G.O., Fearnhead, P.: Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion). J. R. Stat. Soc. Ser. B Stat Methodol. **68**(3), 333–382 (2006)

Beskos, A., Roberts, G.O.: Exact simulation of diffusions. Annals of Applied Probability **15**(4), 2422–2444 (2005)

Buckwar, E., Tamborrino, M., Tubikanec, I.: Spectral density-based and measure-preserving ABC for partially observed diffusion processes. an illustration on hamiltonian SDEs. Stat. Comput. **30**(3), 627–648 (2020)

Cranmer, K., Brehmer, J., Louppe, G.: The frontier of simulation-based inference. Proc. Natl. Acad. Sci. **117**(48), 30055–30062 (2020)

Craigmile, P., Herbei, R., Liu, G., Schneider, G.: Statistical inference for stochastic differential equations. Wiley Interdisciplinary Reviews: Computational Statistics **15**(2), 1585 (2023)

Chen, Y., Zhang, D., Gutmann, M.U., Courville, A., Zhu, Z.: Neural approximate sufficient statistics for implicit models. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=SRDuJssQud

Deleforge, A., Forbes, F., Horaud, R.: High-dimensional regression with Gaussian mixtures and partially-latent response variables. Stat. Comput. **25**(5), 893–911 (2014)

Davidian, M., Giltinan, D.M.: Nonlinear models for repeated measurement data: an overview and update. J. Agric. Biol. Environ. Stat. **8**, 387–419 (2003)

Diggle, P.J., Heagerty, P., Liang, K.-Y., Zeger, S.: Analysis of Longitudinal Data. Oxford University Press, Oxford, UK; Published in the U.S. by Oxford University Press Inc., New York (2002)

Delaunoy, A., Hermans, J., Rozet, F., Wehenkel, A., Louppe, G.: Towards reliable simulation-based inference with balanced neural ratio estimation. Adv. Neural. Inf. Process. Syst. **35**, 20025–20037 (2022)

Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J. Roy. Stat. Soc.: Ser. B (Methodol.) **39**(1), 1–22 (1977)

Del Moral, P., Murray, L.M.: Sequential Monte Carlo with highly informative observations. SIAM/ASA Journal on Uncertainty Quantification **3**(1), 969–997 (2015)

Durkan, C., Murray, I., Papamakarios, G.: On contrastive learning for likelihood-free inference. In: International Conference on Machine Learning, pp. 2771–2781 (2020). PMLR

Eddelbuettel, D., Francois, R., Allaire, J., Ushey, K., Kou, Q., Russell, N., Ucar, I., Bates, D., Chambers, J.: Rcpp: Seamless R and C++ Integration. (2024). R package version 1.0.12. https://CRAN.R-project.org/package=Rcpp

Flegal, J.M., Hughes, J., Vats, D., Dai, N., Gupta, K., Maji, U.: Mcmcse: Monte Carlo Standard Errors for MCMC. Riverside, CA, and Kanpur, India (2025). R package version 1.5-1

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)

Fröhlich, F., Reiser, A., Fink, L., Woschée, D., Ligon, T., Theis, F.J., Rädler, J.O., Hasenauer, J.: Multi-experiment nonlinear mixed effect modeling of single-cell translation kinetics after transfection. NPJ systems biology and applications **4**(1), 42 (2018)

Fruhwirth-Schnatter, S., Celeux, G., Robert, C.P.: Handbook of Mixture Analysis. CRC Press, Boca Raton, FL (2019)

Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. **81**(25), 2340–2361 (1977)

Gillespie, D.T.: The chemical langevin equation. J. Chem. Phys. **113**(1), 297–306 (2000)

Gillespie, D.T.: Stochastic simulation of chemical kinetics. Annu. Rev. Phys. Chem. **58**(1), 35–55 (2007)

Greenberg, D., Nonnenmacher, M., Macke, J.: Automatic posterior transformation for likelihood-free inference. In: International Conference on Machine Learning, pp. 2404–2414 (2019). PMLR

Golightly, A., Wilkinson, D.J.: Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. Interface focus **1**(6), 807–820 (2011)

Gabry, J., Češnovar, R., Johnson, A., Bronder, S.: Cmdstanr: R Interface to 'CmdStan'. (2024). R package version 0.8.1, https://discourse.mc-stan.orghttps://mc-stan.org/cmdstanr/

Hoffman, M.D., Gelman, A.: The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. J. Mach. Learn. Res. **15**(1), 1593–1623 (2014)

Higham, D., Kloeden, P.: An Introduction to the Numerical Simulation of Stochastic Differential Equations. SIAM, Philadelphia, PA, USA (2021)

Häggström, H., Rodrigues, P.L.C., Oudoumanessah, G., Forbes, F., Picchini, U.: Fast, accurate and lightweight sequential simulation-based inference using Gaussian locally linear mappings. Transactions on Machine Learning Research (2024). https://openreview.net/forum?id=Q0nzpRcwWn

Lavielle, M.: Mixed Effects Models for the Population Approach: Models, Tasks. Methods and Tools. CRC Press, Boca Raton, FL, USA (2014)

Miller, B.K., Cole, A., Forré, P., Louppe, G., Weniger, C.: Truncated marginal neural ratio estimation. Adv. Neural. Inf. Process. Syst. **34**, 129–143 (2021)

Marin, J.-M., Pudlo, P., Robert, C.P., Ryder, R.J.: Approximate Bayesian computational methods. Stat. Comput. **22**(6), 1167–1180 (2012)

Murphy, K.P.: Conjugate Bayesian analysis of the Gaussian distribution. https://www.cs.ubc.ca/~murphyk/Papers/bayesGauss.pdf. Accessed: 1 October 2024 (2007)

Price, L.F., Drovandi, C.C., Lee, A., Nott, D.J.: Bayesian synthetic likelihood. J. Comput. Graph. Stat. **27**(1), 1–11 (2018)

Perthame, E., Forbes, F., Deleforge, A., Devijver, E., Gallopin, M.: xLLiM: High Dimensional Locally-Linear Mapping. (2022). R package version 2.2.1

Pieschner, S., Hasenauer, J., Fuchs, C.: Identifiability analysis for models of the translation kinetics after mRNA transfection. J. Math. Biol. **84**(7), 56 (2022)

Picchini, U.: Stochastic differential equations mixed-effects models. https://umbertopicchini.github.io/sdemem/. Accessed: 27 November 2025 (2025)

Papamakarios, G., Murray, I.: Fast ε-free inference of simulation models with Bayesian conditional density estimation. Advances in Neural Information Processing Systems **29** (2016)

Papamakarios, G., Sterratt, D., Murray, I.: Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows. In: Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, vol. 89, pp. 837–848. PMLR (2019)

Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference. The Journal of Machine Learning Research **22**(1), 2617–2680 (2021)

Pesonen, H., Simola, U., Köhn-Luque, A., Vuollekoski, H., Lai, X., Frigessi, A., Kaski, S., Frazier, D.T., Maneesoonthorn, W., Martin, G.M., Corander, J.: ABC of the future. Int. Stat. Rev. **91**(2), 243–268 (2023)

Persson, S., Welkenhuysen, N., Shashkova, S., Wiqvist, S., Reith, P., Schmidt, G.W., Picchini, U., Cvijovic, M.: Scalable and flexible inference framework for stochastic dynamic single-cell models. PLoS Comput. Biol. **18**(5), 1–24 (2022)

Robert, C., Casella, G.: Monte Carlo Statistical Methods. Springer (2004)

Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: International Conference on Machine Learning, pp. 1530–1538 (2015). PMLR

Radev, S.T., Schmitt, M., Pratz, V., Picchini, U., Koethe, U., Buerkner, P.: JANA: Jointly amortized neural approximation of complex Bayesian models. In: Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence, pp. 1695–1706 (2023). PMLR

Sisson, S.A., Fan, Y., Beaumont, M.: Handbook of Approximate Bayesian Computation. CRC Press, New York; London, UK (2018)

Statisticat, LLC.: LaplacesDemon: Complete Environment for Bayesian Inference. (2021). R package version 16.1.6. https://web.archive.org/web/20150206004624/http://www.bayesian-inference.com/software

Stan Development Team: Stan Modeling Language Users Guide and Reference Manual, version 2.32 (2023). https://mc-stan.org/docs/2_32/reference-manual/index.html

Stan Development Team: RStan: the R interface to Stan. R package version 2.32.6 (2024). https://mc-stan.org/

Schauer, M., Meulen, F., Zanten, H.: Guided proposals for simulating multi-dimensional diffusion bridges. Bernoulli **23**(4A), 2917–2950 (2017)

Voliotis, M., Thomas, P., Grima, R., Bowsher, C.G.: Stochastic simulation of biomolecular networks in dynamic environments. PLoS Comput. Biol. **12**(6), 1004923 (2016)

Wiqvist, S., Frellsen, J., Picchini, U.: Sequential neural posterior and likelihood approximation. arXiv preprint arXiv:2102.06522 (2021)

Whitaker, G.A., Golightly, A., Boys, R.J., Sherlock, C.: Bayesian inference for diffusion-driven mixed-effects models. Bayesian Anal. **2**(12), 435–463 (2017)

Wiqvist, S., Golightly, A., McLean, A.T., Picchini, U.: Efficient inference for stochastic differential equation mixed-effects models using correlated particle pseudo-marginal algorithms. Computational Statistics & Data Analysis **157**, 107151 (2021)

Wang, X., Kelly, R.P., Jenner, A.L., Warne, D.J., Drovandi, C.: A comprehensive guide to simulation-based inference in computational biology. arXiv preprint arXiv:2409.19675 (2024)

Wiqvist, S., Mattei, P.-A., Picchini, U., Frellsen, J.: Partially exchangeable networks and architectures for learning summary statistics in approximate Bayesian computation. In: Proceedings of the 36th International Conference on Machine Learning, vol. 97, pp. 6798–6807 (2019)

Wood, S.N.: Statistical inference for noisy nonlinear ecological dynamic systems. Nature **466**(7310), 1102–1104 (2010)

Xu, L., Jordan, M., Hinton, G.E.: An alternative model for mixtures of experts. Advances in neural information processing systems **7** (1994)

Zammit-Mangion, A., Sainsbury-Dale, M., Huser, R.: Neural methods for amortized inference. Annual Review of Statistics and Its Application **12** (2024)