Towards data-conditional simulation for ABC inference in stochastic differential equations

Citation for the original published paper (version of record):
Jovanovski, P., Golightly, A., Picchini, U. (2026). Towards data-conditional simulation for ABC inference in stochastic differential equations. Bayesian Analysis, 21(1): 309-339.
http://dx.doi.org/10.1214/24-BA1467

Bayesian Analysis (2026) **21**, Number 1, pp. 309–339

Towards Data-Conditional Simulation for ABC Inference in Stochastic Differential Equations

Petar Jovanovski*, Andrew Golightly†, and Umberto Picchini‡

Abstract. We develop a Bayesian inference method for the parameters of discretely-observed stochastic differential equations (SDEs). Inference is challenging for most SDEs, due to the analytical intractability of the likelihood function. Nevertheless, forward simulation via numerical methods is straightforward, motivating the use of approximate Bayesian computation (ABC). We propose a computationally efficient "data-conditional" simulation scheme for SDEs that is based on lookahead strategies for sequential Monte Carlo (SMC) and particle smoothing using backward simulation. This leads to the simulation of trajectories that are consistent with the observed trajectory, thereby considerably increasing the acceptance rate in an ABC-SMC sampler. As a result, our procedure rapidly guides the parameters towards regions of high posterior density, especially in the first ABC-SMC round. We additionally construct a sequential scheme to learn the ABC summary statistics, by employing an invariant neural network, previously developed for Markov processes, that is incrementally retrained during the run of the ABC-SMC sampler. Our approach achieves accurate inference and is about three times faster (and in some cases even 10 times faster) than standard (forward-only) ABC-SMC. We illustrate our method in five simulation studies, including three examples from the Chan–Karaolyi–Longstaff–Sanders SDE family, a stochastic bi-stable model (Schlögl) that is notoriously challenging for ABC methods, and a two dimensional biochemical reaction network.

Keywords: approximate Bayesian computation, biochemical reaction networks, invariant neural networks, sequential Monte Carlo, simulation-based inference, smoothing.

1 Introduction

Stochastic modelling is an important area of applied mathematics for the study of the evolution of systems driven by random dynamics. In this work we focus on Bayesian inference for stochastic differential equations (SDEs) observed at discrete time instants. SDEs (Oksendal, 2013; Särkkä and Solin, 2019) are fundamental tools for the modelling of continuous-time experiments that are subject to random dynamics. Application of SDEs are found in many areas including finance (Steele, 2001), population dynamics (Panik, 2017), systems biology (Wilkinson, 2018) and the wider applied sciences (Fuchs, 2013). However, with the exception of very few tractable cases, or the use of exact methods for a relatively small class of intractable diffusions (e.g. Beskos et al., 2006; Casella

*Dept. Mathematical Sciences, Chalmers University of Technology and the University of Gothenburg, Sweden, petarj@chalmers.se

†Department of Mathematical Sciences, Durham University, UK, andrew.golightly@durham.ac.uk

‡Dept. Mathematical Sciences, Chalmers University of Technology and the University of Gothenburg, Sweden, picchini@chalmers.se

 https://doi.org/10.1214/24-BA1467

and Roberts, 2011), SDEs require simulation-based approaches to conduct inference, since transition densities are unavailable in closed-form and hence the likelihood function cannot be obtained analytically. This prevents straightforward frequentist and Bayesian inference, and as a consequence, has generated much research on approximate methods for parameter inference; see Craigmile et al. (2022) for a recent review. However, intractable likelihoods are a common problem, far and beyond SDEs. For this reason, in the past twenty years, there has been increasing interest in simulation-based inference (SBI) methods (often named "likelihood-free" methods). The appeal of SBI methods is that they only require forward-simulation of the model, rather than the evaluation of a potentially complicated expression for the likelihood function, assuming it is available. SBI methods, whose most studied member is arguably approximate Bayesian computation (ABC, Sisson et al., 2018), are in principle agnostic to the complexity of the model, as long as enough time and computational resources necessary to forward simulate the model are available. Other SBI methods that follow this logic are, for example, synthetic likelihoods (Wood, 2010, Price et al., 2018), methods based on deep neural networks providing surrogates of the likelihood function and the posterior distribution (see the review in Cranmer et al., 2020), and the bootstrap particle filter when incorporated into pseudo-marginal methods, such as particle Markov chain Monte Carlo (pMCMC, Andrieu and Roberts, 2009; Andrieu et al., 2010). With reference to the latter, it can be difficult to initialize pMCMC algorithms at suitable parameter values, and this search may require a large number of model simulations ("particles") to obtain a reasonably mixing chain, when the tested model parameter is outside the bulk of the posterior. The latter issue is less problematic for ABC algorithms such as those based on sequential Monte Carlo samplers (SMC), which is the class of ABC algorithms we examine in this work (hence the acronym ABC-SMC). Inference returned by ABC methods can be used to initialize pMCMC algorithms, as well as select a covariance matrix for a random walk sampler, as done in Owen et al. (2015). While SBI methods are a rapidly expanding class of algorithms that, in many cases, can provide the only viable route to inference for stochastic modelling, they come at a cost. They are computationally demanding, where the bottleneck is the forward simulation of a model (SDEs in our case) conditionally on parameter values $\boldsymbol{\theta}$. Here, a "trajectory" $\mathbf{x}$ is a solution to the SDE (typically obtained via some numerical approximation), simulated conditionally on a given value of the model parameters, and in ABC $\mathbf{x}$ is compared to observed data $\mathbf{x}^{\mathrm{o}}$ via an appropriate distance metric $\|\mathbf{x}-\mathbf{x}^{\mathrm{o}}\|$. The procedure is iterated for many proposed values of $\boldsymbol{\theta}$ and the parameters generating a small distance $\|\mathbf{x}-\mathbf{x}^{\mathrm{o}}\|$ are retained as draws from a posterior distribution approximating the true posterior $\pi(\boldsymbol{\theta}\,|\,\mathbf{x}^{\mathrm{o}})$. However, the "forward simulation" of $\mathbf{x}$, which we denote as $\mathbf{x}\sim p(\mathbf{x}\,|\,\boldsymbol{\theta})$ where $p(\mathbf{x}\,|\,\boldsymbol{\theta})$ is the (unknown) likelihood function of $\boldsymbol{\theta}$, is usually "myopic" of data $\mathbf{x}^{\mathrm{o}}$. This implies that simulated trajectories can be very distant from data, even when simulated conditionally on plausible parameter values. This behaviour can persist even when observed and simulated data are compared via corresponding summary statistics as $\|\mathbf{s}-\mathbf{s}^{\mathrm{o}}\|$. Moreover, summary statistics $\mathbf{s}=S(\mathbf{x})$, for some function $S(\cdot)$, are often chosen in an ad-hoc manner, e.g. from domain knowledge or previous literature, though recent semi-automatic approaches have shown their effectiveness (Fearnhead and Prangle, 2012; Chen et al., 2020; Jiang et al., 2017; Forbes et al., 2021; Wiqvist et al., 2019).

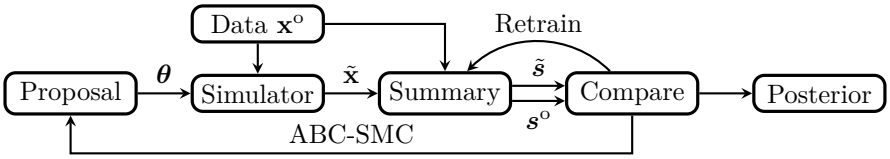

Figure 1: Diagram of dynamic ABC-SMC with data-conditional simulation. Note the dependence of the simulator on the observed trajectory $\mathbf{x}^o$.

In this work we propose, firstly, (i) a data-conditional scheme for SDE models that efficiently utilizes the information provided by the observed data $\mathbf{x}^o$, via the forward-backward simulation of multiple trajectories. Unlike the usual ("forward only") simulation scheme, our approach leads to simulated trajectories that are "adapted" to the observations, and thereby a considerable increase in the acceptance rate for ABC-SMC algorithms is obtained (and thus a more rapid convergence to an approximate posterior for $\boldsymbol{\theta}$). Secondly, (ii) we construct a sequential scheme to learn the ABC summary statistics of univariate SDE models via partially exchangeable networks (PENs, Wiqvist et al., 2019), a deep neural network that learns the posterior mean of a Markov process, and as such is particularly suitable for SDEs inference. Notably, and unlike most inference algorithms for SDEs where the simplest numerical scheme is employed (that is, the Euler-Maruyama discretization), our methodology allows the use of higher-order schemes for the time-forward pass, such as the Milstein scheme. In Figure 1 we present a diagram outlining the structure and flow of our proposed inference pipeline (note the dependence of the simulator on the observed trajectory $\mathbf{x}^o$). The data-conditional simulator, together with PEN, is embedded within an ABC-SMC algorithm. In each step, PEN is retrained on newly accepted trajectories, thus sequentially improving the summary statistics. Our approach dramatically speeds up the convergence to the true posterior, which requires much fewer inference rounds (as we illustrate using Wasserstein distances) and its running time is overall about 3 times faster (but for the marginal posteriors of some parameters it can be even 10 times faster, as in the Lotka-Volterra experiment) than the "standard" ABC-SMC approach where only the forward model $p(\mathbf{x}|\boldsymbol{\theta})$ is used to simulate trajectories. We find that three rounds of our proposed ABC-SMC scheme typically produces satisfactory results, and note that it could also be used in a "hybrid" algorithm, where the output of a few (2–3) rounds of our rapidly converging algorithm could be used to initialize the "standard" ABC-SMC approach for the final inference.

The paper is structured as follows. In Sections 1.1–1.2 we give some necessary background details of ABC methodology and in particular ABC-SMC. Section 2 introduces stochastic differential equations and data-augmentation considerations. Section 3 introduces our backward-smoothing methodology. Section 4 constructs the importance sampling weights for ABC-SMC. Section 5 gives the neural network methodology used to learn summary statistics for ABC that are suitable for SDEs. Finally all the previous concepts are combined in Section 6 to give the algorithm for ABC-SMC with data-conditional simulation. Section 7 contains simulation studies. A Supplementary Material section includes further simulation studies and methodological details (Jovanovski

et al., 2024). Accompanying code is available on GitHub.[1]

1.1 Approximate Bayesian computation

Let $p(\mathbf{x} \mid \boldsymbol{\theta})$ be the likelihood function of $\boldsymbol{\theta}$ corresponding to a given probabilistic model, and let $\pi(\boldsymbol{\theta})$ be the prior density ascribed to $\boldsymbol{\theta}$. The posterior density of $\boldsymbol{\theta}$ given the observed data $\mathbf{x}^{\mathrm{o}}$ is

$$\pi(\boldsymbol{\theta} \mid \mathbf{x}^{\mathrm{o}}) \propto p(\mathbf{x}^{\mathrm{o}} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}). \tag{1}$$

With the exception of a small subset of SDE models, the likelihood function $p(\mathbf{x}^{\mathrm{o}}|\boldsymbol{\theta})$ cannot be obtained in closed form, and this hinders the direct application of commonly used sampling algorithms for (1), such as Markov chain Monte Carlo (MCMC). Approximate Bayesian computation (ABC) bypasses the inability to evaluate the likelihood function analytically, by the ability to forward-simulate from it using a *model simulator*. Here, the model simulator is the SDE solver (up to a discretization error associated with the solver used) which produces a solution $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$ conditional on some value of $\boldsymbol{\theta}$. Using multiple calls from the model simulator at different parameter configurations (which can be accepted or, most often, rejected), ABC returns parameters from an approximate posterior $\pi_\epsilon(\boldsymbol{\theta} \mid \mathbf{x}^{\mathrm{o}})$, rather than the exact posterior in (1). Typically, as motivated in detail in Section 5, it is necessary to first summarize the data with low-dimensional summary statistics $S(\cdot)$, and sample from $\pi_\epsilon(\boldsymbol{\theta} \mid S(\mathbf{x}^{\mathrm{o}}))$. In this case the ABC posterior becomes

$$\pi_\epsilon(\boldsymbol{\theta} \mid S(\mathbf{x}^{\mathrm{o}})) \propto \pi(\boldsymbol{\theta}) \int \mathbb{1}(\|S(\mathbf{x}) - S(\mathbf{x}^{\mathrm{o}})\| \leq \epsilon) p(S(\mathbf{x}) \mid \boldsymbol{\theta})\, \mathrm{d}\mathbf{x}, \tag{2}$$

where $\|\cdot\|$ is an appropriate distance metric, $\mathbb{1}(\|S(\mathbf{x}) - S(\mathbf{x}^{\mathrm{o}})\| \leq \epsilon)$ is the indicator function, and $\epsilon > 0$ a tolerance value determining the accuracy of the approximation. The likelihood of the summary statistics $p(S(\mathbf{x})|\boldsymbol{\theta})$ is implicitly defined, namely samples from the latter are obtained by first sampling (simulating) $\mathbf{x} \sim p(\mathbf{x} \mid \boldsymbol{\theta})$ and then computing $S(\mathbf{x})$. If $S(\cdot)$ is highly informative for $\boldsymbol{\theta}$, then $\pi_\epsilon(\boldsymbol{\theta}|S(\mathbf{x}^{\mathrm{o}})) \simeq \pi_\epsilon(\boldsymbol{\theta}|\mathbf{x}^{\mathrm{o}})$. Therefore, for a small ϵ and an "informative" $S(\cdot)$, ABC may produce a reasonable approximation to the true posterior $\pi(\boldsymbol{\theta} \mid \mathbf{x}^{\mathrm{o}})$. Finding an appropriate ϵ is an exercise in balancing the increasing computational effort when ϵ is reduced (this in turn increases the rejection rate of the proposed parameters), against more accurate posterior inference. As a shorthand, we will denote the summary statistics of the observations by $\mathbf{s}^{\mathrm{o}} = S(\mathbf{x}^{\mathrm{o}})$.

The basic ABC algorithm for producing samples from (2) is the ABC with rejection-sampling (ABC-RS) algorithm (Tavaré et al., 1997; Pritchard et al., 1999), which can be described in three steps. Suppose we have an importance density $g(\boldsymbol{\theta})$ that is easy to sample from and has the same support as the prior. Propose (i) $\boldsymbol{\theta} \sim g(\boldsymbol{\theta})$ (where $g(\boldsymbol{\theta})$ can be the prior $\pi(\boldsymbol{\theta})$), (ii) simulate (implicitly) the summary statistic $\mathbf{s} \sim p(\mathbf{s} \mid \boldsymbol{\theta})$, and (iii) accept $\boldsymbol{\theta}$ as a sample from the ABC posterior with probability

$$\frac{\mathbb{1}(\|\mathbf{s} - \mathbf{s}^{\mathrm{o}}\| \leq \epsilon)\pi(\boldsymbol{\theta})}{C g(\boldsymbol{\theta})}, \quad \text{where } C = \max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \frac{\pi(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}. \tag{3}$$

[1] https://github.com/perojov/DataConditionalABC

Steps (i)–(iii) are iterated until a desired number M of parameter draws have been accepted. Since in steps (i) and (ii) both the parameter and the summary statistic are sampled, the actual sampling density of this algorithm is the joint density $g(\boldsymbol{\theta}, \mathbf{s}) = p(\mathbf{s}|\boldsymbol{\theta})g(\boldsymbol{\theta})$, and the target is the joint ABC posterior given by $\pi_\epsilon(\boldsymbol{\theta}, \mathbf{s}|\mathbf{s}^{\mathrm{o}}) \propto \mathbb{1}(\|\mathbf{s}-\mathbf{s}^{\mathrm{o}}\| \leq \epsilon)p(\mathbf{s}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$. The sampled summary statistic can be safely discarded since marginalizing $\pi_\epsilon(\boldsymbol{\theta}, \mathbf{s} \mid \mathbf{s}^{\mathrm{o}})$ over $\mathbf{s}$ gives $\pi_\epsilon(\boldsymbol{\theta} \mid \mathbf{s}^{\mathrm{o}}) = \int \pi_\epsilon(\boldsymbol{\theta}, \mathbf{s} \mid \mathbf{s}^{\mathrm{o}})\,\mathrm{d}\mathbf{s}$, the ABC posterior (2). The acceptance probability of ABC-RS is then given by

$$\frac{\pi_\epsilon(\boldsymbol{\theta}, \mathbf{s} \mid \mathbf{s}^{\mathrm{o}})}{Cg(\boldsymbol{\theta}, \mathbf{s})} \propto \frac{\mathbb{1}(\|\mathbf{s} - \mathbf{s}^{\mathrm{o}}\| \leq \epsilon)p(\mathbf{s} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{Cp(\mathbf{s} \mid \boldsymbol{\theta})g(\boldsymbol{\theta})} = \frac{\mathbb{1}(\|\mathbf{s} - \mathbf{s}^{\mathrm{o}}\| \leq \epsilon)\pi(\boldsymbol{\theta})}{Cg(\boldsymbol{\theta})}, \tag{4}$$

with C defined as in (3). The choice of the marginal importance density $g(\boldsymbol{\theta})$ is crucial to the efficiency of the algorithm. For example, if $g(\boldsymbol{\theta})$ is much more diffuse compared to the posterior density, most proposed parameter draws will be rejected, and if $g(\boldsymbol{\theta})$ is too concentrated, the tails of the posterior will not be thoroughly explored. To remedy the inefficiencies of ABC-RS, many modifications have been proposed, for which we refer to Sisson et al. (2018), and instead focus on sequential Monte Carlo ABC (Sisson et al., 2007; Toni et al., 2009; Beaumont et al., 2009; Del Moral et al., 2012; Picchini and Tamborrino, 2024), which is illustrated in the next section.

1.2 Approximate Bayesian computation – sequential Monte Carlo

Here we consider ABC implemented via a sequential Monte Carlo sampler, denoted ABC-SMC. In ABC-SMC, a population of M parameter samples (commonly referred to as "particles") are propagated along a sequence of T ABC posterior distributions with decreasing acceptance thresholds $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_T$. This has the effect of sampling parameter particles from regions of increasingly higher posterior density. The initialization works along the same lines as ABC-RS, however, rather than computing the normalizing constant as in (4), an importance weight $w(\boldsymbol{\theta}, \mathbf{s}) = \mathbb{1}(\|\mathbf{s}-\mathbf{s}^{\mathrm{o}}\| \leq \epsilon)\pi(\boldsymbol{\theta})/g(\boldsymbol{\theta})$ is assigned to each sample $(\boldsymbol{\theta}, \mathbf{s}) \sim g(\boldsymbol{\theta}, \mathbf{s})$. The ABC posterior π_{ϵ_1} at this initial step is constructed from a set of M weighted samples $(\boldsymbol{\theta}_1^{1:M}, W_1^{1:M})$ as $\pi_{\epsilon_1}(\boldsymbol{\theta} \mid \mathbf{s}^{\mathrm{o}}) = \sum_{j=1}^M W_1^j \mathcal{K}(\boldsymbol{\theta}_t^i \mid \boldsymbol{\theta}_{t-1}^j)$ (Del Moral et al., 2006). Here, $\mathcal{K}(\cdot \mid \cdot)$ serves as a transition kernel to move the particles into regions of (potentially) high posterior density, as well as increase the particle diversity in the population. In the subsequent steps, rather than proposing parameters from the initial importance density $g(\boldsymbol{\theta})$, they are proposed from a perturbation of the ABC posterior approximated in the previous step. More precisely, at step t, a particle is sampled from the previous population $\boldsymbol{\theta}_{t-1}^{1:M}$ with probabilities $W_{t-1}^{1:M}$, and then perturbed according to the transition kernel. Since the parameters are sampled from the previous ABC posterior, the importance weights for the new particle population $\boldsymbol{\theta}_t^{1:M}$ at this step take the form $W_t^i \propto \pi(\boldsymbol{\theta}_t^i)/\sum_{j=1}^M W_{t-1}^j \mathcal{K}(\boldsymbol{\theta}_t^i \mid \boldsymbol{\theta}_{t-1}^j)$. A typical choice for the transition kernel is the Gaussian density $\mathcal{K}(\boldsymbol{\theta}_t^i \mid \boldsymbol{\theta}_{t-1}^j) = \mathcal{N}(\boldsymbol{\theta}_t^i \mid \boldsymbol{\theta}_{t-1}^j, \boldsymbol{\Sigma}_t)$ (Beaumont et al., 2009; Toni et al., 2009) (here $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate normal density evaluated at $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$), where $\boldsymbol{\Sigma}$ can be specified in several ways. Algorithm 1 outlines ABC-SMC with $\boldsymbol{\Sigma}_t$ chosen to be twice the (weighted) covariance-matrix of the current particles population (as in Beaumont et al., 2009; Filippi et al., 2013). We note that novel, efficient parameter proposals for ABC-SMC have been recently introduced

in Picchini and Tamborrino (2024), though we will not use those in the present work. Finally, notice that the sequence of thresholds $\epsilon_1 > \cdots > \epsilon_T$ does not need to be fixed as an input to the algorithm, but can be dynamically determined at runtime, typically as a percentile of simulated distances.

Algorithm 1 ABC-SMC ($\epsilon_1 > \cdots > \epsilon_T$).

for $i = 1$ to M **do**
  while parameter not accepted **do**
    Sample parameter $\boldsymbol{\theta}_1^i \sim g(\boldsymbol{\theta})$ and summary $\mathbf{s}_1^i \sim p(\mathbf{s} \mid \boldsymbol{\theta}^i)$.
    if $\|\mathbf{s}_1^i - \mathbf{s}^{\mathrm{o}}\| \leq \epsilon_1$ **then**
      Accept $\boldsymbol{\theta}_1^i$ and compute $W_1^i = \pi(\boldsymbol{\theta}_1^i)/g(\boldsymbol{\theta}_1^i)$.
    end if
  end while
end for
Normalize $W_1^{1:M}$.
for $t = 2$ to T **do**
  Compute particle covariance $\boldsymbol{\Sigma}_t = 2 \times \mathrm{Cov}((\boldsymbol{\theta}_{t-1}^{1:M}, W_{t-1}^{1:M}))$.
  for $i = 1$ to M **do**
    while parameter not accepted **do**
      Sample $\boldsymbol{\theta}^*$ from $\boldsymbol{\theta}_{t-1}^{1:M}$ with probabilities $W_{t-1}^{1:M}$.
      Sample $\boldsymbol{\theta}_t^i \sim \mathcal{N}(\boldsymbol{\theta}^*, \boldsymbol{\Sigma}_t)$ and simulate $\mathbf{s}^i \sim p(\mathbf{s} \mid \boldsymbol{\theta}_t^i)$.
      if $\|\mathbf{s}^i - \mathbf{s}^{\mathrm{o}}\| < \epsilon_t$ **then**
        Accept $\boldsymbol{\theta}_t^i$ and compute $W_t^i = \pi(\boldsymbol{\theta}_t^i) / \sum_{j=1}^M W_{t-1}^j \mathcal{N}(\boldsymbol{\theta}_t^i \mid \boldsymbol{\theta}_{t-1}^j, \boldsymbol{\Sigma}_t)$.
      end if
    end while
  end for
  Normalize $W_t^{1:M}$.
end for
Output: Weighted sample $(\boldsymbol{\theta}_T^{1:M}, W_T^{1:M})$ of the ABC posterior density.

2 Inferential problem and numerical simulation of SDEs

Consider the time-homogeneous Itô diffusion $(\mathbf{X}_t)_{t \geq 0}$ satisfying the SDE

$$\begin{cases} \mathrm{d}\mathbf{X}_t = \boldsymbol{\mu}(\mathbf{X}_t, \boldsymbol{\theta})\mathrm{d}t + \boldsymbol{\sigma}(\mathbf{X}_t, \boldsymbol{\theta})\mathrm{d}\mathbf{B}_t, & \text{if } t > 0, \\ \mathbf{X}_0 = \mathbf{x}_0, & \text{if } t = 0, \end{cases} \tag{5}$$

with state space $\mathcal{X} \subseteq \mathbb{R}^d$, constant initial condition $\mathbf{x}_0 \in \mathcal{X}$, p-dimensional model parameter $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^p$ and d-dimensional standard Brownian motion $(\mathbf{B}_t)_{t \geq 0}$. The drift function $\boldsymbol{\mu} : \mathcal{X} \times \boldsymbol{\Theta} \to \mathbb{R}^d$ and diffusion coefficient $\boldsymbol{\sigma} : \mathcal{X} \times \boldsymbol{\Theta} \to \mathbb{R}^{d \times d}$ are assumed to be known in parametric form, and assumed to be sufficiently regular to ensure the existence and uniqueness of a solution for (5). We denote the transition density of the process $(\mathbf{X}_t)_{t \geq 0}$ by $p(\mathbf{x}_t \mid \mathbf{x}_s, \boldsymbol{\theta})$ for times $0 \leq s < t$.

The goal of our work is to perform statistical inference on the parameter $\boldsymbol{\theta}$, based on discrete-time observations of the diffusion at time instants $t_1 < t_2 < \cdots < t_n$, observed exactly (i.e. without additional measurement error). Due to the Markov property, the likelihood of $\boldsymbol{\theta}$ factorizes into the product of mutually independent transition densities. More precisely, for observation $\mathbf{x}^{\mathrm{o}} = (\mathbf{x}_0^{\mathrm{o}}, \mathbf{x}_1^{\mathrm{o}}, \ldots, \mathbf{x}_n^{\mathrm{o}})$, with a non-random initial state

$\mathbf{x}_0 = \mathbf{x}_0^{\text{o}}$, the likelihood is given by

$$p(\mathbf{x}_1^{\text{o}}, \ldots, \mathbf{x}_n^{\text{o}} \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} p(\mathbf{x}_i^{\text{o}} \mid \mathbf{x}_{i-1}^{\text{o}}, \boldsymbol{\theta}). \tag{6}$$

If the transition densities $p(\mathbf{x}_i^{\text{o}} \mid \mathbf{x}_{i-1}^{\text{o}}, \boldsymbol{\theta})$ are analytically tractable, statistical inference may proceed via maximum likelihood estimation, or via a Bayesian approach by computing the posterior distribution of $\boldsymbol{\theta}$ given the observation $\mathbf{x}^{\text{o}}$. However, the transition densities are tractable only for a handful of specific SDE models, and hence the likelihood of $\boldsymbol{\theta}$ cannot in general be evaluated. There exist several different approaches to deal with the problem of intractable likelihoods, see e.g. (Iacus, 2008; Fuchs, 2013; Craigmile et al., 2022) for a survey. In this work we focus on methods that utilize "data augmentation", implying the need to numerically discretize the solution of the SDE on a finer time-scale than the observational grid $(t_1, t_2, \ldots, t_n)$. We assume a regular observation grid with $t_i - t_{i-1} = \Delta$ for simplicity in notation. However, our method is also applicable to grids with irregular spacing, though special attention will need to be paid in these cases, see Section 2 in the Supplementary Material. Approximate solutions can be simulated using numerical methods that are based on the discrete-time approximation of the continuous process. Additionally, the quality of the ABC approximation to the posterior depends on the accuracy of the simulator. Sometimes exact simulation is possible (e.g. Beskos et al., 2006; Casella and Roberts, 2011), but in the case where the simulator is a numerical discretization method, e.g. Euler-Maruyama (EM), the accuracy will be determined by the fineness of the grid. In applications, the inter-observation times are often large, and direct simulation on that same scale will lead to bias. Bayesian data augmentation methods (Fuchs, 2013; van der Meulen and Schauer, 2017; Papaspiliopoulos et al., 2013; Golightly and Sherlock, 2022) introduce missing data between the observations such that the union of the observation and the missing data forms a dataset on a finer scale. Motivated by these approaches, we introduce a finer discretization of the interval $(t_0, t_1, \ldots, t_n)$ by $(\tau_0, \tau_1, \ldots, \tau_A, \ldots \tau_{nA})$, where $\tau_{iA} = t_i$, and such that $\tau_k - \tau_{k-1} = h = \Delta/A$, where $A \geq 2$ is the number of subintervals between two observational time instants. We denote by $N \equiv nA$ the total number of elements in the finer discretization, and for the discrete values of $(\mathbf{X}_t)_{t\geq 0}$ we adopt the shorthand $\mathbf{x}_{iA+k} \equiv \mathbf{x}_{\tau_{iA+k}}$. Similarly for the observed trajectory, we adopt the shorthand $\mathbf{x}_i^{\text{o}} \equiv \mathbf{x}_{t_i}^{\text{o}}$. The EM method is the most commonly used numerical scheme for SDEs: given the aforementioned finer partition, EM produces the discretization

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \boldsymbol{\mu}(\mathbf{X}_i, \boldsymbol{\theta})h + \boldsymbol{\sigma}(\mathbf{X}_i, \boldsymbol{\theta})(\mathbf{B}_{i+1} - \mathbf{B}_i), \tag{7}$$

for $i = 0, \ldots, N-1$. By the properties of the Brownian motion, the transition density $p(\mathbf{x}_{i+1} \mid \mathbf{x}_i, \boldsymbol{\theta})$ induced by the EM scheme is multivariate Gaussian with mean $\mathbf{x}_i + \boldsymbol{\mu}(\mathbf{x}_i, \boldsymbol{\theta})h$ and covariance matrix $\boldsymbol{\sigma}(\mathbf{x}_i, \boldsymbol{\theta})\boldsymbol{\sigma}^T(\mathbf{x}_i, \boldsymbol{\theta})h$.

In this paper we will consider two ways to simulate a trajectory: the first is "myopic" forward simulation, meaning that the trajectory is generated from $p(\mathbf{x}|\boldsymbol{\theta})$ and is therefore not conditioned on the observation (this is the standard approach in ABC, e.g. Picchini, 2014). The second way is one of our contributions which we term *data-conditional* simulation, where the sample path is generated from $p(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{x}^{\text{o}})$, and thereby conditional

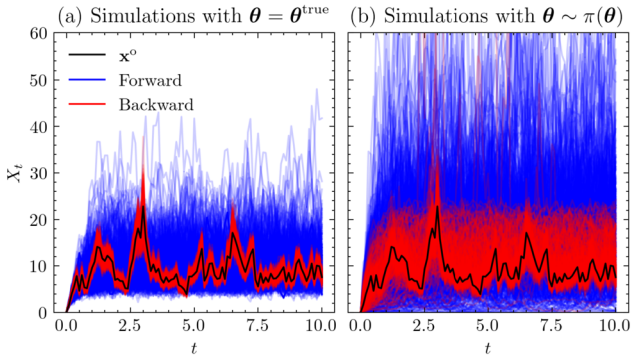

Figure 2: Illustration of simulated trajectories from both the forward and the data-conditional simulator with $P = 30$ particles for the CKLS SDE $\mathrm{d}X_t = \beta(\alpha - X_t)\mathrm{d}t + \sigma X_t^{\gamma}\mathrm{d}B_t$. The observation is shown in black, the forward trajectories in blue, and the backward trajectories in red. In (a) 500 trajectories (per simulator) are shown: these were simulated conditionally on the true parameter $\boldsymbol{\theta}^{\text{true}}$. In (b) 500 trajectories (per simulator) are shown: these were simulated from the prior-predictive distribution, with prior $\pi(\boldsymbol{\theta}) = \mathcal{U}(5, 30)\mathcal{U}(0, 3)\mathcal{U}(0.5, 2)\mathcal{U}(0.6, 1)$. This prior is chosen only for illustration, and it is different from the prior used for inference.

on the observed data. In Section 3 we will construct this scheme which is based on a forward/backward idea. The scheme possesses a "lookahead" mechanism for which the distribution of the intermediate points depends on the successive observations. In the forward direction, multiple trajectories are forward propagated and weighted, and then in the backward direction a single trajectory is simulated using a backward-simulation particle smoother. For an illustration of the difference between these two approaches, see Figure 2. There, for the CKLS SDE that we consider extensively in Section 7, we show that, due to the specific diffusion term, the trajectories produced from the forward model can vary substantially, even when simulating conditionally on the true parameter (Figure 2(a)). However using our forward-backward data-conditional approach, simulated trajectories are much more consistent with the observed data. Data-conditional simulation is illustrated in Section 3.

3 Data-conditional simulation

Backward simulation can be implemented via a sequential Monte Carlo (SMC) scheme that is based on processing the observation in the forward and then in the backward direction. In the forward direction, a numerical scheme is used to propagate multiple particles, which represent numerical solutions to the underlying SDE model. At this stage the particles are suitably weighted according to how close they are to the observed data $\mathbf{x}^{\text{o}}$. In the backward direction, a backward-simulation particle smoother (BSPS) is applied on the particle system that is obtained from the forward direction, in order to construct a single (backward) trajectory. This backward trajectory is a sample

from the conditional distribution $p(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{x}^{\text{o}})$. The development of the forward direction of our data-conditional simulator borrows ideas from the literature of simulation methods for diffusion bridges using SMC, such as Del Moral and Garnier (2005); Lin et al. (2010) and in particular, the bridge particle filter (BPF) of Del Moral and Murray (2015) (see also Golightly and Wilkinson, 2015). In the BPF, the simulation of a diffusion bridge $\mathbf{x}_{iA}, \ldots, \mathbf{x}_{(i+1)A}$ (recall that A is the number of subintervals) between two consecutive observations $(\mathbf{x}_i^{\text{o}}, \mathbf{x}_{i+1}^{\text{o}})$ relies on a suitably chosen particle weighting scheme, such that a set of P particles are guided from $\mathbf{x}_i^{\text{o}}$ to $\mathbf{x}_{i+1}^{\text{o}}$. Particles are simulated forward in time using only $p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \boldsymbol{\theta})$ for $iA \leq k \leq (i+1)A$, that is, the transition density arising from a numerical scheme such as Euler-Maruyama, or if desired, a higher order scheme (e.g. the Milstein scheme in $d = 1$). Even though the particles are propagated by the forward density, they are weighted in such a way that, by looking ahead to the next observation point $\mathbf{x}_{i+1}^{\text{o}}$, those particle trajectories that are not consistent with $\mathbf{x}_{i+1}^{\text{o}}$ will be given small weights and pruned out with a resampling step which occurs at every intermediate time point τ_k (unlike in standard particle filtering where resampling occurs at observation time points only). However, the BPF will result in a particle system $(\mathbf{x}_{1:N}^{1:P}, \omega_{1:N}^{1:P})$ for which $(\mathbf{x}_{1:N}^{j})$ is, at the observation time instants, exactly equal to the observed data, i.e. $\mathbf{x}_{t_i}^{j} \equiv \mathbf{x}_i^{\text{o}}$ for $i = 0, \ldots, n$. The reason is that for every subinterval $[t_i, t_{i+1}]$ the states $\mathbf{x}_{iA+1}, \ldots, \mathbf{x}_{(i+1)A-1}$ are randomly generated, whereas $\mathbf{x}_{iA} = \mathbf{x}_i^{\text{o}}$, $\mathbf{x}_{(i+1)A} = \mathbf{x}_{i+1}^{\text{o}}$ is fixed. The consequence is that, in the backward pass, the backward-simulation particle smoother will construct a backward trajectory that is exactly equal to $\mathbf{x}^{\text{o}}$ (at the observation time instants), regardless of the value of $\boldsymbol{\theta}$. Hence, the event $\{\|\mathbf{s} - \mathbf{s}^{\text{o}}\| = 0\}$ in (4) will occur with probability 1 for any $\boldsymbol{\theta}$, which is unwanted. Rather, we require acceptance for suitable values of $\boldsymbol{\theta}$, i.e. those that are representative of the posterior density. To this end, we will change the BPF as follows. In our adaptation we make two modifications. The first modification omits the resampling step in the BPF, in order for the particles (which represent numerical solutions to the underlying SDE model) to preserve their original sampling distribution $p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \boldsymbol{\theta})$. This will be of fundamental importance when we embed the data-conditional simulator within ABC-SMC, as well as in assembling the training data for the partially exchangeable neural network. Although we omit the resampling step, the particle weights are still computed, and they will be utilized once the forward phase is completed. The second modification is that we perform one more forward propagation step, from time $\tau_{(i+1)A-1}$ to $\tau_{(i+1)A}$ for every $i = 0, \ldots, n-1$, namely, the states $\mathbf{x}_{(i+1)A}$ are random, as opposed to be deterministically set to $\mathbf{x}_{i+1}^{\text{o}}$ as in the BPF of Del Moral and Murray (2015). The resulting algorithm reduces to sequential importance sampling (SIS) and will yield forward (myopic) trajectories that are weighted with respect to the observations. In order to utilize the weights, once the trajectories have been simulated up to time t_n, we complement SIS with a second recursion that evolves backward in time. In the backward pass, the (backward) trajectory is produced by firstly simulating $\tilde{\mathbf{x}}_N$, then $\tilde{\mathbf{x}}_{N-1}$, etc. until a complete trajectory $\tilde{\mathbf{x}}_0, \ldots, \tilde{\mathbf{x}}_N$ is generated. This forward-backward idea will allow us to develop a simulator that will generate trajectories which are consistent with the observation $\mathbf{x}^{\text{o}}$. The procedure is detailed in next sections.

3.1 Lookahead sequential importance sampling

We will begin by deriving the lookahead SIS algorithm for the first observation interval $[0, t_1]$, before considering the remaining intervals. As motivated above, the time interval $[0, t_1]$ is partitioned into A subintervals $0 = \tau_0 < \tau_1 < \cdots < \tau_A = t_1$, and the diffusion takes values $\mathbf{x}_0^o$ at $t = 0$ and $\mathbf{x}_1^o$ and $t = t_1$. Along this partition, multiple forward trajectories will be sampled recursively, by drawing from $p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \boldsymbol{\theta})$, and weighting according to

$$\tilde{W}_k(\mathbf{x}_{k-1}, \mathbf{x}_k) = \frac{q(\mathbf{x}_1^o \mid \mathbf{x}_k)}{q(\mathbf{x}_1^o \mid \mathbf{x}_{k-1})}, \quad \text{for } k = 1, \ldots, A, \tag{8}$$

where $q(\mathbf{x}_1^o \mid \mathbf{x}_k)$, for $k = 0, \ldots, A$ are positive functions. It is important to distinguish that the tilded W_k differs from the parameter weights used in ABC-SMC. This forms the basis of the BPF (Del Moral and Murray, 2015), albeit for the case of exact observations (i.e. when $\mathbf{x}_1^o$ is observed without noise). Note that the weighting function $q(\mathbf{x}_1^o \mid \mathbf{x}_A)$ at time τ_A can be omitted because $\mathbf{x}_A$ is fixed to be equal to $\mathbf{x}_1^o$. As alluded to previously, instead of fixing the last state as in the BPF, we perform one more forward propagation step, and hence include $q(\mathbf{x}_1^o \mid \mathbf{x}_A)$. A suitable choice for this weighting function is important because it greatly affects the efficiency of the inference. We discuss possible options later in this section. Through the use of the lookahead mechanism, the intermediate samples between two observations, $\mathbf{x}_1, \ldots, \mathbf{x}_{A-1}$ (excluding the final sample $\mathbf{x}_A$) are given weights according to how consistent they are with the subsequent observation $\mathbf{x}_1^o$. However, for the final sample $\mathbf{x}_A$, this approach is invalid because the transition density at $\tau_A \equiv t_1$ is a Dirac mass at the observation point. To this end we propose to take $q(\mathbf{x}_1^o \mid \mathbf{x}_A)$ to be equal to $q(\mathbf{x}_1^o \mid \mathbf{x}_{A-1})$, which is a plausible choice when the integration timestep $\tau_A - \tau_{A-1} = h$ is small. This way the particles $(\mathbf{x}_A^{1:P})$ that are close to the observational point $\mathbf{x}_1^o$ will be given greater weights, as opposed to those that are further away. Another possible choice for $q(\mathbf{x}_1^o \mid \mathbf{x}_A)$ may be a decaying function of the distance between $\mathbf{x}_1^o$ and $\mathbf{x}_A$, for example the Gaussian kernel, or the inverse of the (squared) Euclidean distance. The conditional distribution of the intermediate points, given the subsequent observation, admits the following form

$$\begin{aligned} p(\mathbf{x}_{1:A} \mid \mathbf{x}_0^o, \mathbf{x}_1^o) &\propto q(\mathbf{x}_1^o \mid \mathbf{x}_A) p(\mathbf{x}_{1:A} \mid \mathbf{x}_0^o) \frac{q(\mathbf{x}_1^o \mid \mathbf{x}_0^o)}{q(\mathbf{x}_1^o \mid \mathbf{x}_A)} \prod_{k=1}^{A} \tilde{W}_k(\mathbf{x}_{k-1}, \mathbf{x}_k) \\ &\propto \prod_{k=1}^{A} \tilde{W}_k(\mathbf{x}_{k-1}, \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{x}_{k-1}). \end{aligned} \tag{9}$$

Note that

$$\frac{q(\mathbf{x}_1^o \mid \mathbf{x}_0^o)}{q(\mathbf{x}_1^o \mid \mathbf{x}_{t_1})} \prod_{k=1}^{A} \tilde{W}_k(\mathbf{x}_{k-1}, \mathbf{x}_k) = 1.$$

For a generic observation interval $[t_i, t_{i+1}]$, the conditional distribution of the intermediate points $\mathbf{x}_{iA+1}, \ldots, \mathbf{x}_{(i+1)A}$ is similar to (9), except that the state $\mathbf{x}_{iA}$ at time τ_{iA} is not equal to the observation $\mathbf{x}_i^o$, but to the final value that was simulated on the previous interval $[t_{i-1}, t_i]$. This is different from the BPF case where the particles are

initialized to start from the observation at time t_i. Having derived the conditional distribution of the intermediate points on the first interval, we can easily extend (9) to the complete discretization $(\tau_0, \tau_1, \ldots, \tau_A, \ldots, \tau_N)$ as follows

$$p(\mathbf{x}_{1:N} \mid \mathbf{x}^{\mathrm{o}}) = \prod_{i=0}^{n-1} p(\mathbf{x}_{iA+1:(i+1)A} \mid \mathbf{x}_{iA}, \mathbf{x}^{\mathrm{o}}_{i+1}) \propto \prod_{i=0}^{n-1} \prod_{k=iA+1}^{(i+1)A} \tilde{W}_k(\mathbf{x}_{k-1}, \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{x}_{k-1}). \tag{10}$$

Equation (10) readily implies a sequential scheme where one can incrementally sample $\mathbf{x}_k \sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ and weight that sample by $\tilde{W}_k(\mathbf{x}_{k-1}, \mathbf{x}_k)$, for $k = 1, \ldots, N$. Assume that we have simulated a set of P particles up to time t_n, then we have the particle system $(\mathbf{x}_{1:N}^{1:P}, \omega_{1:N}^{1:P})$ approximating the lookahead densities

$$\hat{p}(\mathrm{d}\mathbf{x}_{iA+k} \mid \mathbf{x}^{\mathrm{o}}_{0:t_{i+1}}) = \sum_{j=1}^{P} \omega^j_{iA+k} \delta_{\mathbf{x}^j_{iA+k}}(\mathrm{d}\mathbf{x}_{iA+k}) \quad \text{for } k = 1, \ldots, A, \tag{11}$$

for $i = 0, \ldots, n-1$, where ω_k^j denote the normalized particle weights. Due to the way the weights $\tilde{W}_k(\mathbf{x}_{k-1}, \mathbf{x}_k)$ are defined in (8), it is worth noting that in the sequential-importance-sampling scheme, the weight of the sample $\mathbf{x}_{iA+k} \sim p(\mathbf{x}_{iA+k} \mid \mathbf{x}_{iA+k-1})$ is proportional to $q(\mathbf{x}^{\mathrm{o}}_{i+1} | \mathbf{x}_{iA+k})$ (see also Algorithm 3 in Del Moral and Murray, 2015 and the subsequent note on extending it to a time series of observations). In Algorithm 2 we present the lookahead SIS algorithm corresponding to equation (10) for a set of P particles, an illustration of which can be seen in Figure 3.

Having approximated the lookahead densities up to time t_n, we can now obtain an approximation of the so called backward kernel, to evolve backward in time and obtain a single trajectory that we use in ABC-SMC to decide on whether to accept or reject the associated parameter. Backward simulation methods for Monte Carlo (Lindsten et al., 2013), as well as the derivation of the backward simulation particle smoother for the finer discretization of $[0, t_n]$, is the topic of the following section.

Algorithm 2 Lookahead SIS $(\mathbf{x}^{\mathrm{o}}, \boldsymbol{\theta})$.

for $j = 1$ to P in parallel **do**
  for $i = 0$ to $n - 1$ **do**
    for $k = 1$ to A **do**
      Sample $\mathbf{x}^j_{iA+k} \sim p(\mathbf{x}_{iA+k} \mid \mathbf{x}^j_{iA+k-1}, \boldsymbol{\theta})$ with timestep h.
      Append the sample to the state history $\mathbf{x}^j_{0:iA+k} = \{\mathbf{x}^j_{0:iA+k-1}, \mathbf{x}^j_{iA+k}\}$.
      Calculate the weight of the particle according to $\omega^j_{iA+k} \propto q(\mathbf{x}^{\mathrm{o}}_{i+1} \mid \mathbf{x}^j_{iA+k})$.
    end for
  end for
end for
Output: Particle system $(\mathbf{x}_{1:N}^{1:P}, \omega_{1:N}^{1:P})$

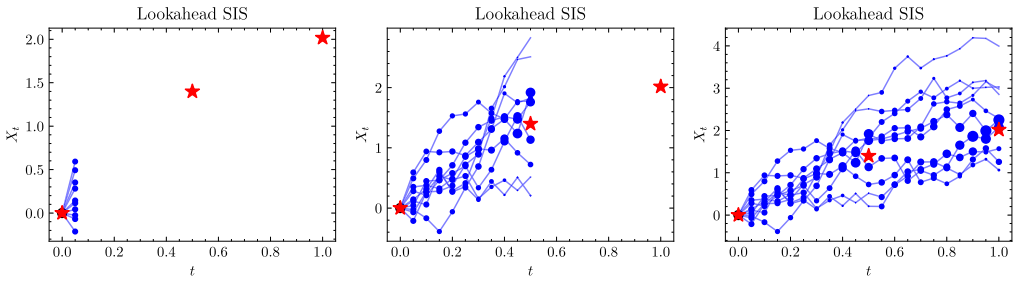

Figure 3: Evolution of a particle system obtained from the Lookahead SIS algorithm. Observations (red stars) and particles (blue). The size of each particle is proportional to its weight. Left: the particles are initialized at the initial state, and a single forward step is performed. Middle: the particles are simulated up to time t_1 and are weighted according to the observed state x_1^{o}. At time t_1, the particles start to look towards the subsequent observation x_2^{o}, and therefore are weighed in a different way, as seen in the right panel.

3.2 Extracting a single trajectory using particle smoothing

This section describes how to make use of the particles generated in the forward direction (see Algorithm 2) to select a suitable simulated trajectory for use within ABC-SMC. Backward simulation (BS) is a technique that is used within SMC to address the smoothing problem for models that have latent stochastic processes, i.e. state-space models (Lindsten et al., 2013; Särkkä, 2013). In a state-space model, it is assumed that a dynamical system evolves according to a Markovian (latent) stochastic process having state $\mathbf{x}_t$ at time t which is not directly measured; rather, $\mathbf{y}_t$ is observed as some function of the latent $\mathbf{x}_t$. Smoothing refers to the problem of estimating the distribution of the latent state at a particular time, given all of the observations up to some later time. More formally, given a set of observations $\mathbf{y} = (\mathbf{y}_{t_1}, \ldots, \mathbf{y}_{t_n})$, smoothing addresses the estimation of the marginal distributions $p(\mathbf{x}_{t_k} \mid \mathbf{y})$ for $k = 1, \ldots, n$, or sampling from the entire smoothing density $p(\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_n} \mid \mathbf{y})$. Backward simulation is based on the forward-backward idea, and assumes that Bayesian filtering has already been performed on the entire collection of observations, leading to an approximate representation of the densities $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ for each time step $t \in (t_1, \ldots, t_n)$, consisting of weighted particles $(\mathbf{x}_t^{1:P}, \omega_t^{1:P})$. The primary goal of BS is to obtain sample realizations from the smoothing density to gain insight about the latent stochastic process. The smoothing density can be factorized as

$$p(\mathbf{x}_{t_1:t_n} \mid \mathbf{y}) = p(\mathbf{x}_{t_n} \mid \mathbf{y}) \prod_{i=1}^{n} p(\mathbf{x}_{t_i} \mid \mathbf{x}_{t_{i+1}:t_n}, \mathbf{y}), \tag{12}$$

where, under the Markovian assumption of the model, one can write

$$p(\mathbf{x}_{t_i} \mid \mathbf{x}_{t_{i+1}:t_n}, \mathbf{y}) = p(\mathbf{x}_{t_i} \mid \mathbf{x}_{t_{i+1}}, \mathbf{y}_{t_1:t_i}) \propto p(\mathbf{x}_{t_i} \mid \mathbf{y}_{t_1:t_i}) p(\mathbf{x}_{t_{i+1}} \mid \mathbf{x}_{t_i}). \tag{13}$$

This equation additionally shows that, conditional on $\mathbf{y}$, $(\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_n})$ is an inhomogeneous Markov process. In our case we do not have a latent process as in the state-space model, but rather exact (without measurement error), discrete-time observations of an SDE. Still, we can utilize backward simulation because of the way that we have constructed the lookahead sequential importance sampling (Lookahead SIS, Algorithm 2) scheme. Moreover, the output of backward simulation will be the simulated trajectories that will be used in the ABC algorithm. Next, we focus on the backward simulation step where we derive the backward recursion along the finer discretization that will be used to generate smoothed trajectories. The key ingredient in this second step is the backward kernel

$$\mathcal{B}_{iA+k}(\mathrm{d}\mathbf{x} \mid \mathbf{x}_{iA+k+1}) = \mathbb{P}(\mathbf{x}_{iA+k} \in \mathrm{d}\mathbf{x} \mid \mathbf{x}_{iA+k+1}, \mathbf{x}^{\mathrm{o}}_{0:t_{i+1}}), \tag{14}$$

which admits the following density

$$p(\mathbf{x}_{iA+k} \mid \mathbf{x}_{iA+k+1}, \mathbf{x}^{\mathrm{o}}_{0:t_{i+1}}) \propto p(\mathbf{x}_{iA+k+1} \mid \mathbf{x}_{iA+k}) p(\mathbf{x}_{iA+k} \mid \mathbf{x}^{\mathrm{o}}_{0:t_{i+1}}). \tag{15}$$

Using the backward kernel we can get the following expression for the backward recursion

$$p(\mathbf{x}_{iA+k:N} \mid \mathbf{x}^{\mathrm{o}}_{0:t_n}) = p(\mathbf{x}_{iA+k} \mid \mathbf{x}_{iA+k+1}, \mathbf{x}^{\mathrm{o}}_{0:t_{i+1}}) p(\mathbf{x}_{iA+k+1:N} \mid \mathbf{x}^{\mathrm{o}}_{0:t_n}), \tag{16}$$

starting from the lookahead density $p(\mathbf{x}_N \mid \mathbf{x}^{\mathrm{o}}_{0:t_n})$ at time t_n. Equation (16) implies the following scheme: (i) sample

$$\tilde{\mathbf{x}}_N \sim p(\mathbf{x}_N \mid \mathbf{x}^{\mathrm{o}}_{0:t_n}), \tag{17}$$

and then going backwards in time, (ii) incrementally sample

$$\tilde{\mathbf{x}}_{iA+k} \sim p(\mathbf{x}_{iA+k} \mid \mathbf{x}_{iA+k+1}, \mathbf{x}^{\mathrm{o}}_{0:t_{i+1}}). \tag{18}$$

After a complete backwards sweep, the (backward) trajectory $(\tilde{\mathbf{x}}_0, \ldots, \tilde{\mathbf{x}}_N)$ is by construction a realization from the smoothing density $p(\mathbf{x} \mid \mathbf{x}^{\mathrm{o}})$. An important property of the backward kernel density is that at time τ_{iA+k}, it depends only on the transition density $p(\mathbf{x}_{iA+k+1} \mid \mathbf{x}_{iA+k})$, which we are able to approximate for a small timestep h, and on $p(\mathbf{x}_{iA+k} \mid \mathbf{x}^{\mathrm{o}}_{0:t_{i+1}})$, which is approximated by the weighted particle system obtained in the forward direction. To this end, to utilize the backward recursion, the lookahead densities $p(\mathbf{x}_{iA+k} \mid \mathbf{x}^{\mathrm{o}}_{0:t_{i+1}})$ must first be computed for $i = 0, \ldots, n-1$ and $k = 1, \ldots, A$. By substituting the particle approximation of the lookahead densities (11) into the backward density (15) we have the following approximation to the backward kernel

$$\hat{\mathcal{B}}_{iA+k}(\mathrm{d}\mathbf{x}_{iA+k} \mid \mathbf{x}_{iA+k+1}) = \sum_{j=1}^{P} \frac{\omega^j_{iA+k} p(\mathbf{x}_{iA+k+1} \mid \mathbf{x}^j_{iA+k})}{\sum_{l=1}^{P} \omega^l_{iA+k} p(\mathbf{x}_{iA+k+1} \mid \mathbf{x}^l_{iA+k})} \delta_{\mathbf{x}^j_{iA+k}}(\mathrm{d}\mathbf{x}_{iA+k}). \tag{19}$$

Algorithm 3 presents the complete backward recursion given a weighted particle system obtained from the lookahead SIS algorithm. A visual illustration is given in Figure 4. Notice that while the complete backward recursion results in a trajectory $\tilde{\mathbf{x}}$ of length $nA + 1$, we only need the values at the observational timepoints $(t_1, \ldots, t_n)$. As such,

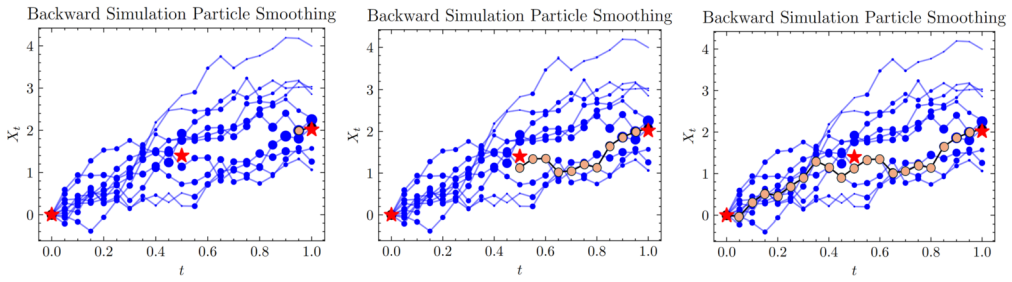

Figure 4: Evolution of a trajectory that is obtained via the backward simulation particle smoother on the particle system in Figure 3. Observations (red stars) and all particles (blue). Left: a particle is chosen at random according to equation (17) and a single step backward is taken according to equation (18) (the chosen particle is in apricot color). Middle: the backward recursion is taken up to time t_1 (in apricot). The right panel depicts the complete backward trajectory in apricot.

we downsample $\tilde{\mathbf{x}}$ at these points. Inspired by this, we explored taking larger time steps backward, despite deriving our recursions, (17)–(18), on a fine grid. For instance, rather than retracing steps on the fine grid at intervals of $\Delta t/A$ (which involves A steps), we considered intervals of larger lengths, for example $\Delta t/2$, transitioning from t_i to $t_{i-A/2}$, and then to t_{i-1} (a total of 2 steps). We experimented with different step sizes and found that stepping backward directly from t_i to t_{i-1} yielded the most consistent trajectories. As a result, our simulation studies adopt this method. Yet, the forward trajectories maintain their precision, given the forward simulator integrates at $h = \Delta t/A$.

Algorithm 3 BSPS: backward-simulation particle smoother $((\mathbf{x}_{1:N}^{1:P}, \omega_{1:N}^{1:P}), \boldsymbol{\theta})$.

1: Sample particle index $j \sim \mathcal{M}(\{\omega_N^i\}_{i=1}^P)$ and set $\tilde{\mathbf{x}}_N = \mathbf{x}_N^j$.
2: **for** $k = N-1$ to 1 **do**
3: **for** $j = 1$ to P **do**
4: Compute $\tilde{\omega}_k^j \propto \omega_k^j p(\tilde{\mathbf{x}}_{k+1} \mid \mathbf{x}_k^j, \boldsymbol{\theta})$.
5: **end for**
6: Normalize the smoothing weights $\{\tilde{\omega}_k^{1:P}\}$ to sum to unity.
7: Sample particle index $j \sim \mathcal{M}(\{\tilde{\omega}_k^{1:P}\})$ and set $\tilde{\mathbf{x}}_k = \mathbf{x}_k^j$.
8: Append the sample to the state history $\tilde{\mathbf{x}}_{k:N} = \{\tilde{\mathbf{x}}_k, \tilde{\mathbf{x}}_{k+1:N}\}$.
9: **end for**
10: **Output:** Backward trajectory $\tilde{\mathbf{x}}_{0:N}$.

In summary, given a parameter proposal $\boldsymbol{\theta}^*$ and the observation $\mathbf{x}^o$, the simulation of a backward trajectory $\tilde{\mathbf{x}}$ is achieved by executing Algorithms 2 and 3. Algorithm 2 returns the particle system that approximates the lookahead densities, and this same particle system is used as input to Algorithm 3 to approximate the backward kernel density. We denote by $\tilde{\mathbf{x}} \sim p(\mathbf{x} \mid \boldsymbol{\theta}^*, \mathbf{x}^o)$ the process of sampling a backward trajectory by executing the two algorithms. This $\tilde{\mathbf{x}}$ is the trajectory for which the ABC summary statistic $s_t^i = S(\tilde{\mathbf{x}})$ is calculated and evaluated in line 18 of the ABC-SMC Algorithm 5. In the next two sections we clarify further steps that are necessary for the adaptation of ABC-SMC to our specific context. In particular, Section 4 discusses how to appro-

priately weight the accepted parameter particles in ABC-SMC. Section 5 discusses the construction of ABC summary statistics that are suitable for SDE inference.

4 Intractable importance ratio and its approximation

Throughout section (3) we have addressed the problem of simulating paths $\tilde{\mathbf{x}}$ using a data-conditional forward-backward procedure. It now remains to be seen how to (i) construct suitable summary statistics that we apply to both $\tilde{\mathbf{x}}$ and data $\mathbf{x}^o$, and (ii) how to correct for the fact that $\tilde{\mathbf{x}}$ is not simulated from the forward model. We now consider (ii), and address (i) in Section 5. With a standard, forward only, simulator we would only need to care for $p(\mathbf{s} \mid \boldsymbol{\theta})$, with $\mathbf{s} = S(\tilde{\mathbf{x}})$ and $S(\cdot)$ a function still to be determined (to be addressed in Section 5). The utilization of the backward simulator effectively changes the distribution of the simulated trajectories. The summary of this backward trajectory is no longer distributed according to $p(\mathbf{s} \mid \boldsymbol{\theta})$, but rather according to another density that is conditional on observed data and which we denote by $p(\mathbf{s} \mid \boldsymbol{\theta}, \mathbf{x}^{\mathrm{o}})$ (this density is implicitly defined as that of the summary statistic of the backward trajectory sampled from the joint smoothing density $p(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{x}^{\mathrm{o}})$). Hence, the importance weight of the parameter-summary pair takes a different form and includes the intractable summary likelihoods of both the forward and the backward simulator. More precisely, the sampling density $g(\mathbf{s}, \boldsymbol{\theta})$ is no longer $p(\mathbf{s} \mid \boldsymbol{\theta})g(\boldsymbol{\theta})$, but rather $p(\mathbf{s} \mid \boldsymbol{\theta}, \mathbf{x}^{\mathrm{o}})g(\boldsymbol{\theta})$. We denote this new sampling density by $g(\mathbf{s}, \boldsymbol{\theta} \mid \mathbf{x}^{\mathrm{o}})$ to emphasize the dependence on the observed data. For a proposed parameter-summary pair the importance weight becomes

$$\frac{\pi_\epsilon(\boldsymbol{\theta}^*, \mathbf{s}^* \mid \mathbf{s}^{\mathrm{o}})}{g(\boldsymbol{\theta}^*, \mathbf{s}^* \mid \mathbf{x}^{\mathrm{o}})} \propto \frac{\mathbb{1}(\|\mathbf{s} - \mathbf{s}^{\mathrm{o}}\| \leq \epsilon)\pi(\boldsymbol{\theta}^*)}{g(\boldsymbol{\theta}^*)} \frac{p(\mathbf{s}^* \mid \boldsymbol{\theta}^*)}{p(\mathbf{s}^* \mid \boldsymbol{\theta}^*, \mathbf{x}^{\mathrm{o}})}. \tag{20}$$

Unlike the corresponding ratio for forward simulators in (4), the importance weight in (20) is intractable. Below, we focus on developing a computationally efficient procedure for approximating the ratio (20), by making use of the Synthetic Likelihood (SL) method. The SL method was first proposed in Wood (2010) to perform inference for parameters of computer-based models with an intractable likelihood. SL is characterized by the assumption that the summary statistic $\mathbf{s}$ follows a multivariate normal distribution with unknown mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$. These can in turn be estimated by implicitly sampling a set of P summary statistics for a particular parameter $\boldsymbol{\theta}$, $\mathbf{s}^j \sim p(\mathbf{s} \mid \boldsymbol{\theta})$ for $j = 1, \ldots, P$, and then computing their empirical mean $\boldsymbol{\mu}_{P,\boldsymbol{\theta}}$ and covariance $\boldsymbol{\Sigma}_{P,\boldsymbol{\theta}}$. This results in the approximation $p(\mathbf{s} \mid \boldsymbol{\theta}) \approx \mathcal{N}(\mathbf{s} \mid \boldsymbol{\mu}_{P,\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{P,\boldsymbol{\theta}})$. To construct an appropriate SL approximation of $p(\mathbf{s} \mid \boldsymbol{\theta}, \mathbf{x}^{\mathrm{o}})$, we need to look more closely into the particle approximation to the joint smoothing density of the backward trajectory. By plugging the approximations of the lookahead densities (11) and the backward kernels (14) into the joint smoothing density (12) along the fine discretization, we get the following approximation

$$\hat{p}(\mathrm{d}\mathbf{x} \mid \mathbf{x}^{\mathrm{o}}, \boldsymbol{\theta}) = \sum_{j_1=1}^{P} \cdots \sum_{j_n=1}^{P} \left(\prod_{i=1}^{n-1} \frac{\omega_{iA}^{j_i} p(\mathbf{x}_{(i+1)A}^{j_{i+1}} \mid \mathbf{x}_{iA}^{j_i}, \boldsymbol{\theta})}{\sum_{l=1}^{P} \omega_{iA}^{l} p(\mathbf{x}_{(i+1)A}^{j_{i+1}} \mid \mathbf{x}_{iA}^{l}, \boldsymbol{\theta})} \right) \omega_N^{j_n} \delta_{\mathbf{x}_A^{j_1}, \ldots, \mathbf{x}_N^{j_n}}(\mathrm{d}\mathbf{x}). \tag{21}$$

This equation defines a discrete distribution on $\mathcal{X}^n$, and can be understood as follows: for each observational time t_i, $i = 1, \ldots, n$ along the coarse discretization, the particles

$(\mathbf{x}_{iA}^{1:P})$ generated by Lookahead SIS is a set in $\mathcal{X}$ of cardinality P. By picking one particle at each time point, we obtain a particle trajectory, i.e. a point $(\mathbf{x}_A^{j_1}, \mathbf{x}_{2A}^{j_2}, \ldots, \mathbf{x}_N^{j_n}) \in \mathcal{X}^n$ with entries obtained at the n observational time points. We get P^n such trajectories by letting the indices $j_1, \ldots, j_n$ range from 1 to P. Clearly the form of (21) is computationally intensive to evaluate, however, it provides a connection to the backward simulator (Algorithm 3). Implicitly, the particle approximation to the joint smoothing density (21) depends on the particle system from the forward direction, i.e. $\hat{p}(\mathrm{d}\mathbf{x}|\mathbf{x}^{\mathrm{o}}, \boldsymbol{\theta}) = \hat{p}(\mathrm{d}\mathbf{x} \mid \mathbf{x}^{\mathrm{o}}, \boldsymbol{\theta}, (\mathbf{x}_{1:N}^{1:P}, \omega_{1:N}^{1:P}))$. Therefore, conditionally on the particle system from Lookahead SIS, the backward simulator generates i.i.d. Markovian samples from the distribution (21). Thus, to appropriately approximate $p(\mathbf{s} \mid \boldsymbol{\theta}, \mathbf{x}^{\mathrm{o}})$ by a Gaussian likelihood, the weighted particle system that is obtained from Lookahead SIS needs to remain fixed whilst the backward trajectories are being sampled. The larger the number of samples P, the better the approximation to the summary likelihood, albeit at a higher simulation cost. In our applications, we set the number of simulations for SL to be the same as the number of particles for the lookahead simulator, which we found to work well in the empirical examples. Hence if P is set to a high number, the cost per one forward-backward simulation will be increased. Similarly to the forward summary likelihood, the backward summary likelihood can be approximated by sampling a set of P data-conditional summary statistics $\tilde{\mathbf{s}}^j \sim p(\mathbf{s} \mid \boldsymbol{\theta}, \mathbf{x}^{\mathrm{o}})$ and then computing their empirical mean $\tilde{\boldsymbol{\mu}}_{P,\boldsymbol{\theta}}$ and covariance $\tilde{\boldsymbol{\Sigma}}_{P,\boldsymbol{\theta}}$. The resulting approximation is $p(\mathbf{s} \mid \boldsymbol{\theta}, \mathbf{x}^{\mathrm{o}}) \approx \mathcal{N}(\mathbf{s} \mid \tilde{\boldsymbol{\mu}}_{P,\boldsymbol{\theta}}, \tilde{\boldsymbol{\Sigma}}_{P,\boldsymbol{\theta}})$. The importance weight for a parameter-summary pair $(\boldsymbol{\theta}, \mathbf{s})$ can now be approximated as

$$w(\boldsymbol{\theta}, \mathbf{s}) \propto \frac{\mathbb{1}(\|\mathbf{s} - \mathbf{s}^{\mathrm{o}}\| \leq \epsilon)\pi(\boldsymbol{\theta})}{g(\boldsymbol{\theta})} \frac{\mathcal{N}(\mathbf{s} \mid \boldsymbol{\mu}_{P,\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{P,\boldsymbol{\theta}})}{\mathcal{N}(\mathbf{s} \mid \tilde{\boldsymbol{\mu}}_{P,\boldsymbol{\theta}}, \tilde{\boldsymbol{\Sigma}}_{P,\boldsymbol{\theta}})}. \tag{22}$$

Crucially, the form of the importance weight in (22) indicates that the ratio of the intractable likelihoods need only be computed when the simulated summary is close enough to the observed summary, i.e. $\mathbb{1}(\|\mathbf{s} - \mathbf{s}^{\mathrm{o}}\| \leq \epsilon) = 1$, otherwise the importance weight can be immediately set to zero without the need to compute the synthetic likelihoods. The procedure for computing the ratio is outlined in Algorithm 4.

Algorithm 4 Synthetic likelihood approximation $((\mathbf{x}_{1:N}^{1:P}, \omega_{1:N}^{1:P}), \boldsymbol{\theta}, \mathbf{s})$.

1: **for** $j = 1$ to P **do**
2: Summarize the genealogy of the j^{th} particle $\mathbf{s}^j = S(\mathbf{x}^j)$.
3: Sample $\tilde{\mathbf{x}}^j \sim \hat{p}(\mathbf{x} \mid \mathbf{x}^{\mathrm{o}}, \boldsymbol{\theta}, (\mathbf{x}_{1:N}^{1:P}, \omega_{1:N}^{1:P}))$ and summarize $\tilde{\mathbf{s}}^j = S(\tilde{\mathbf{x}}^j)$.
4: **end for**
5: Calculate the empirical means and covariance matrices from $\mathbf{s}^{1:P}$ and $\tilde{\mathbf{s}}^{1:P}$, respectively: $\boldsymbol{\mu}_{P,\boldsymbol{\theta}}$ (resp. $\tilde{\boldsymbol{\mu}}_{P,\boldsymbol{\theta}}$) and $\boldsymbol{\Sigma}_{P,\boldsymbol{\theta}}$ (resp. $\tilde{\boldsymbol{\Sigma}}_{P,\boldsymbol{\theta}}$).
6: **Output:** Approximate ratio $\mathcal{N}(\mathbf{s} \mid \boldsymbol{\mu}_{P,\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{P,\boldsymbol{\theta}})/\mathcal{N}(\mathbf{s} \mid \tilde{\boldsymbol{\mu}}_{P,\boldsymbol{\theta}}, \tilde{\boldsymbol{\Sigma}}_{P,\boldsymbol{\theta}})$.

In the Supplementary Material, we explore a crucial property of the approximate importance weight, as defined in (22), which allows for the exclusion of implausible parameters, independent of the ABC threshold ϵ value. This method hinges on the variance of the weights of the particle system, obtained from Lookahead SIS, and its impact on the covariance matrix $\tilde{\boldsymbol{\Sigma}}_{P,\boldsymbol{\theta}}$ of the synthetic likelihood found in the denominator of (22).

5 Automatic summary statistics for SDEs

The construction of the summary function $S(\cdot)$ has so far been left unspecified. We address this issue in this section and further details are in a Supplementary Material section, altogether with notable bibliography. In this work we employ a prediction-based approach via deep neural networks where, for a generic imputed dataset $\mathbf{x}$, the summaries are given by $S(\mathbf{x}) = \mathbb{E}(\boldsymbol{\theta} \mid \mathbf{x}) = f_\beta(\mathbf{x})$, with $f_{\boldsymbol{\beta}}(\cdot)$ a deep neural network parameterised via the parameter $\boldsymbol{\beta}$. Specifically, we take as $f_{\boldsymbol{\beta}}(\cdot)$ the partially exchangeable network (PEN) of Wiqvist et al. (2019). PENs offers a powerful way to learn ABC summary statistics for SDEs, at a much lower computational cost compared to other networks, since PENs do not have to "learn" the Markovian structure of the data, given that the network itself is designed to accommodate such a framework. The summary statistics estimator can be trained before running the inference algorithm. However, for this strategy to be effective, very many samples from the prior-predictive distribution are required, and this number is affected by how "vague" the prior is. This is the approach taken in Picchini (2014); Wiqvist et al. (2019); Fearnhead and Prangle (2012); Jiang et al. (2017); Chan et al. (2018). Recently, Chen et al. (2020) proposed a dynamic learning strategy, where the main idea is to learn the summary statistics and the posterior density at the same time, over multiple rounds, which we also advocate here. More precisely, at round t, the current summary statistics network $S_t(\cdot)$ is used to approximate the ABC posterior density, and then the summary statistics network is retrained on all the data obtained up to round t, and the newly trained network $S_{t+1}(\cdot)$ is used in round $t+1$. In order to learn the summary statistics in our approach, we denote with $\mathcal{D}$ the set containing training data. In the standard (forward) ABC-SMC approach, the dataset $\mathcal{D}$ is progressively populated with accepted parameters and their corresponding simulated trajectories. In our approach, however, this step warrants careful attention. For a given parameter, a set of P forward trajectories are sampled with Algorithm 2 and one backward trajectory is sampled with Algorithm 3. The backward trajectory is summarized and then passed to the ABC accept/reject step, but it is not this trajectory that is stored into $\mathcal{D}$. The distribution of the backward trajectories is markedly different from that of the forward trajectories, particularly for parameters that do not come from the posterior (see Section 1 of the Supplementary Material for a detailed discussion). Therefore, to keep the learning of the summary statistics consistent, we select one of the P *forward* trajectories according to how close it is to the observed trajectory. More precisely, we downsample the P forward trajectories and find the one with the minimum Euclidean distance to $\mathbf{x}^{\mathrm{o}}$, as follows

$$\mathbf{x} = \underset{\mathbf{x}\in\{\mathbf{x}^1,\ldots,\mathbf{x}^P\}}{\arg\min} \sqrt{\sum_{i=1}^{n} (\mathbf{x}_{iA} - \mathbf{x}_i^{\mathrm{o}})^2},$$

and this particular $\mathbf{x}$ is the trajectory that gets added into $\mathcal{D}$ altogether with its data-generating parameter. See also Algorithm 5 which is discussed in Section 6.

Algorithm 5 Dynamic ABC-SMC with data-conditional simulation (ABC-SMC-DC).

Initialize the dataset $\mathcal{D} = \emptyset$.
Generate R prior-predictive samples $(\mathbf{x}^{1:R}, \boldsymbol{\theta}^{1:R}) \overset{\text{iid}}{\sim} p(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$ and append $\mathcal{D} \coloneqq \mathcal{D} \cup (\mathbf{x}^{1:R}, \boldsymbol{\theta}^{1:R})$.
Train PEN on $\mathcal{D}$ to obtain $S_1(\cdot)$ and summarize the observation $\mathbf{s}_1^{\text{o}} = S_1(\mathbf{x}^{\text{o}})$.
for $i = 1$ to M **do**
  Sample $\boldsymbol{\theta}_1^i \sim \pi(\boldsymbol{\theta})$ and sample a particle system $(\mathbf{x}_{1:N}^{1:P}, \omega_{1:N}^{1:P})$ with Algorithm 2.
  Sample backward trajectory $\tilde{\mathbf{x}}$ with Algorithm 3 and compute $d_1^i = \|S_1(\tilde{\mathbf{x}}) - \mathbf{s}_1^{\text{o}}\|$.
  Store $\boldsymbol{\theta}_1^i$, compute the parameter weight w_1^i by $\mathcal{N}(S_1(\tilde{\mathbf{x}}) \mid \boldsymbol{\mu}_{P,\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{P,\boldsymbol{\theta}}) / \mathcal{N}(S_1(\tilde{\mathbf{x}}) \mid \tilde{\boldsymbol{\mu}}_{P,\boldsymbol{\theta}}, \tilde{\boldsymbol{\Sigma}}_{P,\boldsymbol{\theta}})$ using $(\mathbf{x}_{1:N}^{1:P}, \omega_{1:N}^{1:P})$.
  Pick a trajectory $\mathbf{x}$ from the forward trajectories $\mathbf{x}_{1:N}^{1:P}$ as detailed in Section 5, and append it to the dataset $\mathcal{D} \coloneqq \mathcal{D} \cup (\mathbf{x}, \boldsymbol{\theta}_1^i)$.
end for
for $t = 2$ to T **do**
  Normalize $w_{t-1}^{1:M}$. Retrain PEN on $\mathcal{D}$ to obtain $S_t(\cdot)$ and summarize $\mathbf{s}_t^{\text{o}} = S_t(\mathbf{x}^{\text{o}})$.
  Compute particle covariance $\boldsymbol{\Sigma}_t = 2 \times \text{Cov}((\boldsymbol{\theta}_{t-1}^{1:M}, w_{t-1}^{1:M}))$.
  Take ϵ_t to be the α-quantile of distances $d_{t-1}^{1:M}$ corresponding to accepted particles.
  for $i = 1$ to M **do**
    while parameter not accepted **do**
      Sample $\boldsymbol{\theta}^*$ from $\boldsymbol{\theta}_{t-1}^{1:M}$ with probabilities $w_{t-1}^{1:M}$ and perturb $\boldsymbol{\theta}_t^i \sim \mathcal{N}(\boldsymbol{\theta}^*, \boldsymbol{\Sigma}_t)$.
      Sample a particle system $(\mathbf{x}_{1:N}^{1:P}, \omega_{1:N}^{1:P})$ with Algorithm 2.
      Sample backward trajectory $\tilde{\mathbf{x}}$ with Algorithm 3 and compute $d_t^i = \|S_t(\tilde{\mathbf{x}}) - \mathbf{s}_t^{\text{o}}\|$.
      if $d_t^i \leq \epsilon_t$ **then**
        Accept $\boldsymbol{\theta}_t^i$, compute the parameter weight w_t^i by (23) using $(\mathbf{x}_{1:N}^{1:P}, \omega_{1:N}^{1:P})$.
        Pick a trajectory $\mathbf{x}$ from the forward trajectories $\mathbf{x}_{1:N}^{1:P}$ as detailed in Section 5, and append it to the dataset $\mathcal{D} \coloneqq \mathcal{D} \cup (\mathbf{x}, \boldsymbol{\theta}_t^i)$.
      end if
    end while
  end for
end for
Output: Weighted sample $(\boldsymbol{\theta}_T^{1:M}, w_T^{1:M})$ of the ABC posterior distribution.

6 Dynamic ABC-SMC with data-conditional simulation

We now combine the ideas presented in the previous sections to give a unified ABC-SMC framework for SDE inference. We sequentially re-learn the summary statistics with the PEN from Section 5, in every ABC-SMC round, echoing the method in Chen et al. (2020). Prior to ABC-SMC, we produce a dataset comprising of prior-predictive samples, and train PEN on this dataset. At the initial round the summary statistics function is denoted as $S_1(\cdot)$. With every new round (say, round t), the dataset is expanded with newly accepted parameters and trajectories, and PEN is re-trained on this dataset, yielding an updated summary statistics function that we denote by $S_t(\cdot)$. As outlined in Algorithm 1, a parameter $\boldsymbol{\theta}^*$ is proposed from the density $g_t(\cdot)$, and a summary statistic $\tilde{\mathbf{s}}$ is implicitly sampled from the data-conditional simulator $\tilde{\mathbf{s}}_t \sim p(\mathbf{s} \,|\, \mathbf{x}^{\text{o}}, \boldsymbol{\theta}^*)$ by sampling the conditional trajectory $\tilde{\mathbf{x}} \sim p(\mathbf{x} \mid \mathbf{x}^{\text{o}}, \boldsymbol{\theta}^*)$ and then computing $\tilde{\mathbf{s}}_t = S_t(\tilde{\mathbf{x}})$. The importance weight in this multi-round scenario takes the form

$$w_t(\boldsymbol{\theta}, S_t(\tilde{\mathbf{x}})) \propto \frac{\mathbb{1}(\|S_t(\tilde{\mathbf{x}}) - \mathbf{s}_t^{\text{o}}\| < \epsilon_t)\pi(\boldsymbol{\theta})}{\sum_{i=1}^M W_{t-1}^i \mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{t-1}^i, \boldsymbol{\Sigma}_{t-1})} \frac{\mathcal{N}(S_t(\tilde{\mathbf{x}}) \mid \boldsymbol{\mu}_{P,\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{P,\boldsymbol{\theta}})}{\mathcal{N}(S_t(\tilde{\mathbf{x}}) \mid \tilde{\boldsymbol{\mu}}_{P,\boldsymbol{\theta}}, \tilde{\boldsymbol{\Sigma}}_{P,\boldsymbol{\theta}})}. \tag{23}$$

The complete procedure is given in Algorithm 5, and our data-conditional inference method is denoted with ABC-SMC-DC, while we call ABC-SMC-F (where F stands for "forward simulation") a typical ABC-SMC procedure where simulated data is produced

by the forward model. We do not predefine the thresholding schedule $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_T$ of ABC-SMC, but rather utilize the adaptive schedule proposed in Prangle (2017), where the threshold ϵ_{t+1} is chosen as an α-quantile of the distances corresponding to accepted summaries at round t. Recall from end of Section 5 that the training data $\mathcal{D}$ for PEN is combined in a different way from that of the standard ABC-SMC algorithm. Because PEN is designed to learn from samples from the forward model, and not from the data-conditional model, $\mathcal{D}$ cannot include the accepted trajectories $\tilde{\mathbf{x}}$ where $\tilde{\mathbf{x}} \sim p(\mathbf{x} \,|\, \mathbf{x}^{\mathrm{o}}, \boldsymbol{\theta}^*)$.

7 Simulation studies

7.1 Chan–Karaolyi–Longstaff–Sanders family of models

The Chan–Karaolyi–Longstaff–Sanders (CKLS) family is a class of parametric SDE models that are widely used in finance applications, in particular to model interest rates and asset prices (Chan et al., 1992). The diffusion term of CKLS depends on two parameters and the state, taking the form σX_t^{γ}, where $\sigma > 0$ is the diffusion coefficient and $\gamma \in [0, 1]$. The CKLS family includes the Ornstein–Uhlenbeck process for $\gamma = 0$, the Cox–Ingersoll–Ross process for $\gamma = 1/2$, and the Black–Scholes process for $\alpha = 0$ and $\gamma = 1$. The state-dependent diffusion term can substantially increase the randomness of the SDE paths, and therefore CKLS presents a considerable inferential challenge for ABC algorithms when the forward simulator is used. The CKLS process satisfies the SDE

$$\begin{cases} \mathrm{d}X_t = \beta(\alpha - X_t)\mathrm{d}t + \sigma X_t^{\gamma}\mathrm{d}B_t, & \text{if } t > 0, \\ X_0 = x_0, & \text{if } t = 0, \end{cases} \tag{24}$$

for $\alpha, \beta \in \mathbb{R}$ and $\sigma, \gamma \in \mathbb{R}_+$. We will restrict ourselves to the case where $\alpha, \beta > 0$ and $0 \leq \gamma < 1$. This section describes numerical experiments for the different SDE models from the CKLS family, with tractable as well as intractable likelihoods. For the CKLS model, we already illustrated the benefits of generating trajectories by the data-conditional simulator in Figure 2. A similar figure for the Ornstein–Uhlenbeck, Cox–Ingersoll–Ross and an SDE model with nonlinear drift, can be seen in the Supplementary Material. We evaluate the efficiency of our proposed data-conditional Algorithm 5 (denoted ABC-SMC-DC) compared to ABC-SMC with Euler-Maruyama as a forward simulator (denoted ABC-SMC-F). For the data-conditional simulator we will take as weighing functions the Gaussian densities induced by the EM approximation of (24), namely,

$$q(x_i^{\mathrm{o}} \,|\, x_{iA-1}) = \mathcal{N}(x_i^{\mathrm{o}} \,|\, x_{iA-1} + \beta(\alpha - x_{iA-1})h, \sigma^2 x_{iA-1}^{2\gamma} h). \tag{25}$$

We run Algorithms 1 and 5 with $M = 10,000$ parameter particles and $T = 20$ rounds ($T = 10$ for the simpler case of the Ornstein–Uhlenbeck), but we preemptively stop the inference once the acceptance rate of the parameters falls below 1.5% when $t > 2$. In our experiments, we use the Wasserstein distance as a measure of similarity between probability distributions. If the true posterior is available, we use that as a reference posterior. In the case of an intractable likelihood, we choose as reference

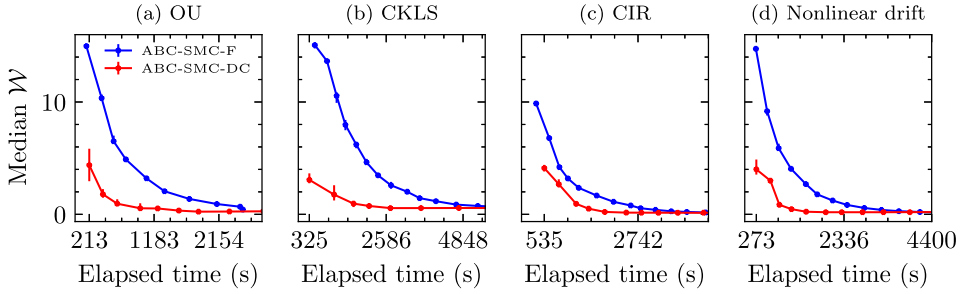

Figure 5: Results from the standard ABC-SMC-F, and our novel ABC-SMC-DC for (a) Ornstein–Uhlenbeck, (b) Chan–Karaolyi–Longstaff–Sanders, (c) Cox–Ingersoll–Ross and (d) a SDE with Nonlinear drift. The median Wasserstein distance between the ABC-posteriors at each round and a reference posterior is shown on the y-axis, and the median cumulative elapsed time per round is shown on the x-axis. A circle represents a round of ABC-SMC. Medians are taken over 20 runs.

the ABC posterior obtained from a run of the standard ABC-SMC algorithm with the same stopping condition and with fixed summary statistics, obtained by training PEN on 300,000 prior-predictive samples before starting ABC-SMC. Throughout this subsection, PEN is composed of dense layers and is structured as 2-100-100-100-dim($\boldsymbol{\theta}$).

Fixed $\gamma = 0$ (Ornstein-Uhlenbeck): inference for (α, β, σ)

In this example, the CKLS model is considered with γ fixed to $\gamma = 0$, yielding the ubiquitously utilized Ornstein–Uhlenbeck (OU) process. A notable characteristic of the OU process is that its transition densities are known (see the Supplementary Material), and hence it is an example of a tractable diffusion, because its likelihood function is available via the product of transition densities. Therefore exact inference via MCMC is possible. Details about the inference setup are given in Supplementary Material. For the inference, we treat γ as known and fixed at $\gamma = 0$, while (α, β, σ) are considered unknown. Our data-conditional ABC-SMC is denoted with ABC-SMC-DC, while pure forward simulation is denoted with ABC-SMC-F. The result of the analysis is in Figure 5(a) and is based on 20 independent runs on the same observed trajectory $\mathbf{x}^o$. The figure displays the computation time versus the Wasserstein distance between the ABC posteriors and the reference posterior (obtained via MCMC using the exact likelihood which is available for this model). In this case, the parameters of the OU model are easier to estimate compared to the general CKLS instance examined further below, where all parameters are unknown and the diffusion is state-dependent. However even in this simpler case, Figure 5(a) clearly displays that with our sampler we require about 3–4 rounds, and about 8–9 rounds with the standard sampler. Reducing the total number of rounds is important in our context, as at each round the PEN network is retrained, which brings a consequent computational overhead. In terms of running time, by comparing similar Wasserstein distances that are obtained at round 3 with our sampler and at round 8 with the standard sampler, it takes about 600 seconds to reach round 3 in

the former case and around 2,100 seconds to reach round 8 for the latter, thus with our sampler we have a 3.5-fold acceleration on average. In Table 1 we show that by round 3–4 the data-conditional ABC-SMC achieves a significant reduction of the Wasserstein distance, while showing a larger acceptance rate than the pure forward approach. For later rounds, the acceptance rate of our simulator decreases, however this is expected since very small Wasserstein distances are achieved.

Round	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
CKLS SDE with (α,β,σ) unknown and $\gamma=0$ (Ornstein–Uhlenbeck)										
Acc. rate (F/DC)	100/100	46/42	37/42	29/30	21/30	18/20	17/11	12/7	10/5	8/2
Wasser. (F/DC)	15/4.4	10/1.8	6.5/0.9	4.9/0.5	3.2/0.5	2/0.3	1.3/0.2	0.9/0.2	0.7/0.3	0.4/0.3
CKLS SDE with $(\alpha,\beta,\sigma,\gamma)$ unknown										
Acc. rate (F/DC)	100/100	54/52	39/19	26/19	22/5	18/2	14/1	12/1	9/1	6/NA
Wasser. (F/DC)	15/3	13.6/1.7	10.5/0.9	8/0.7	6.2/0.5	4.6/0.5	3.5/0.5	2.6/0.5	2/0.6	1.4/NA
CKLS SDE with (α,β,σ) unknown and $\gamma=1/2$ (Cox–Ingersoll–Ross)										
Acc. rate (F/DC)	100/100	54/78	34/50	31/55	22/40	17/33	15/25	13/14	11/8	9/5
Wasser. (F/DC)	9.8/4.1	6.8/2.7	4.2/0.9	3.2/0.5	2.3/0.2	1.6/0.1	1.1/0.1	0.8/0.1	0.5/0.1	0.4/0.1
SDE with nonlinear drift										
Acc. rate (F/DC)	100/100	50/85	36/53	30/59	25/46	21/37	18/24	15/14	13/8	11/4
Wasser. (F/DC)	14.7/4	9.2/3	5.9/0.8	4/0.5	2.7/0.2	1.8/0.2	1.2/0.2	0.8/0.2	0.5/0.2	0.4/0.2

Table 1: Comparison of median acceptance rates (in %) and median Wasserstein distances for both ABC-SMC algorithms, forward (F) and forward-backward data-conditional simulation (DC), in the first 10 rounds. Each cell in the table reports numbers as a/b, where a refers to the performance using F and b refers to B.

CKLS with four unknown parameters $(\alpha, \beta, \sigma, \gamma)$

In this example, we examine the SDE (24), where the parameters $(\alpha, \beta, \sigma, \gamma)$ are all unknown, and the likelihood is intractable. Figure 2 displays the observed data alongside simulated trajectories from both simulators. The analysis results are illustrated in Figure 5(b), derived from 20 independent runs on the observed trajectory depicted in Figure 2. Additionally, the posterior distributions resulting from these 20 runs are visualized in Figure 6. Figure 5 reveals a significant reduction in the Wasserstein distance even from the initial round using our method, as opposed to the standard ABC-SMC-F. Moreover, when viewed in conjunction with Figure 6, it is evident that our method attains satisfactory inference for all parameters by the third round. ABC-SMC-F takes roughly 2,500 seconds to reach a Wasserstein distance of 3.1, while the data-conditional ABC-SMC-DC starts at that value already at round one. Additionally, ABC-SMC needs about 4,645 seconds to attain a Wasserstein distance of 0.9, compared to our method, which requires 1,625 seconds to reach the same distance, thus with our sampler we have a 2.8-fold acceleration on average. See Table 1 for further results: there it is clear that at round 2 the large decrease in the Wasserstein distance brought by our approach is not paired with a drastic drop in the acceptance rate, since the latter is on par with the pure forward simulator. This is a strength of our simulator and, as previously mentioned, the run of our data-conditional simulator here could be halted at round 3.

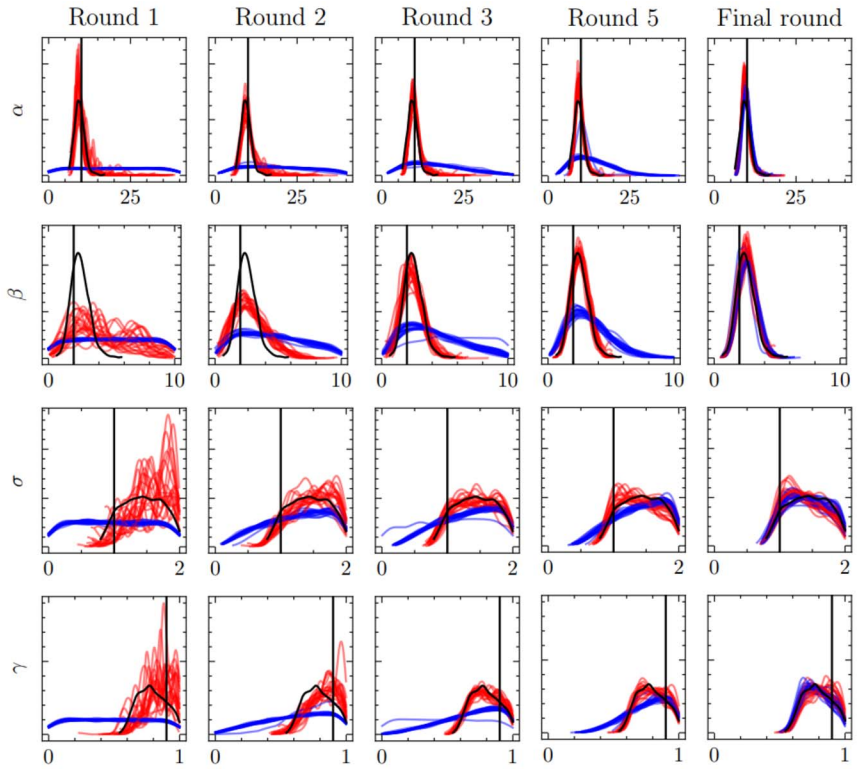

Figure 6: CKLS with four unknown parameters. Marginal posterior distributions across several rounds of ABC-SMC: marginals from our data-conditional ABC-SMC-DC are in red, and those from the forward ABC-SMC-F are in blue. The reference marginal posteriors are depicted in black, and the true parameter values as black vertical lines.

7.2 Biochemical reaction networks

A biochemical reaction network consists of a set of d chemical species, $X_1, \ldots, X_d$ that interact via a network of R reactions

$$\mathcal{R}_j : \sum_{i=1}^{d} \nu_{i,j}^{-} X_i \xrightarrow{\theta_j} \sum_{i=1}^{d} \nu_{i,j}^{+} X_i, \quad j = 1, 2, \ldots, R, \tag{26}$$

where $\nu_{i,j}^{-}$ and $\nu_{i,j}^{+}$ are the number of reactant and product molecules. When the system contains a large number of molecules and reactions occur frequently, the behavior of the biochemical reaction network can be closely approximated using the chemical Langevin equation (see e.g. Wilkinson, 2018). The chemical Langevin equation is an Itô SDE of the form

$$\mathrm{d}\mathbf{X}_t = \sum_{j=1}^{R} \boldsymbol{\nu}_j a_j(\mathbf{X}_t)\mathrm{d}t + \sum_{j=1}^{R} \boldsymbol{\nu}_j \sqrt{a_j(\mathbf{X}_t)}\mathrm{d}\mathbf{B}_t^{(j)}, \tag{27}$$

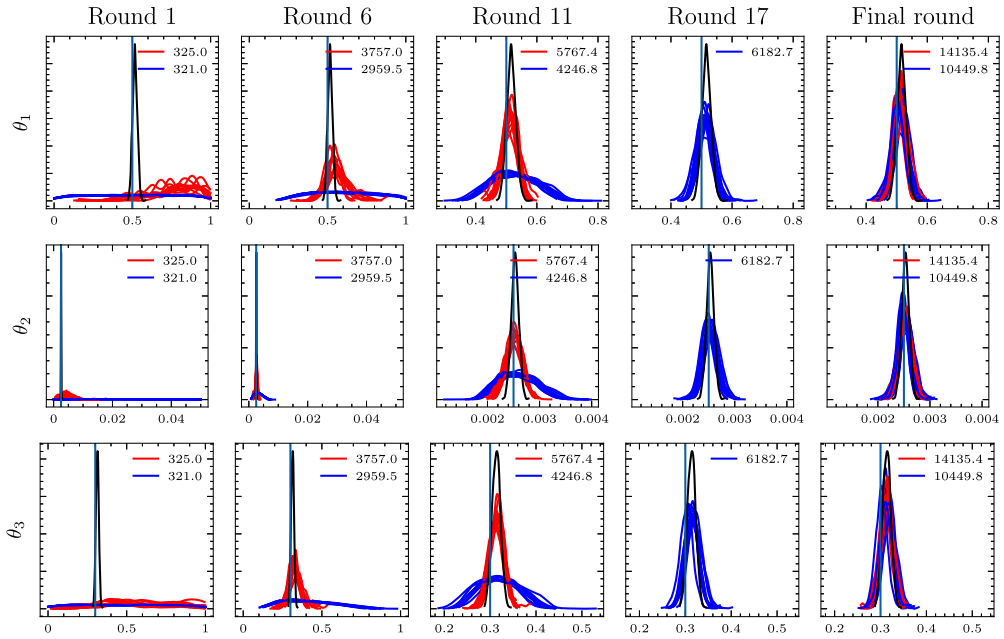

Figure 7: Lotka–Volterra model. Marginal posterior distributions at several rounds of ABC-SMC: for ABC-SMC-DC the marginals are displayed in red, and for ABC-SMC-F in blue (ABC-SMC-DC terminated before round 17). The reference marginal posteriors are depicted in black, and the true parameter values as black vertical lines. Each panel reports, in the upper right corner, the number of seconds since the algorithm started.

where $\boldsymbol{\nu}_j$ is the jth column of the stoichiometry matrix $\boldsymbol{\nu}$ with elements $\nu_{i,j} = \nu^+_{i,j} - \nu^-_{i,j}$, $a_j(\cdot)$ is the hazard of reaction j (typically computed via a mass-action rate law) and $\mathbf{B}^{(j)}_t$, $j = 1, \ldots, R$, are uncorrelated Brownian motion processes. For these models the parameter inference problem involves calculating the posterior distribution of the reaction rate constants $\theta_1, \ldots, \theta_R$. This section presents numerical experiments for two distinct types of biochemical reaction networks: the Schlögl model and the Lotka–Volterra model. The Schlögl model is a one-dimensional SDE that demonstrates stochastic bistability. The Lotka-Volterra model is a two-dimensional SDE and is explored for its characteristic oscillatory behavior. The Supplementary Material gives simulation details and the inference setup for both models.

Lotka–Volterra model

The Lotka–Volterra model is composed of two biochemical species, predator and prey, and three reactions (prey reproduction, prey death and predator reproduction, predator death). A reference posterior was returned by a run of (correlated particles) pseudo-marginal Metropolis-Hastings, with the likelihood function obtained using the modified

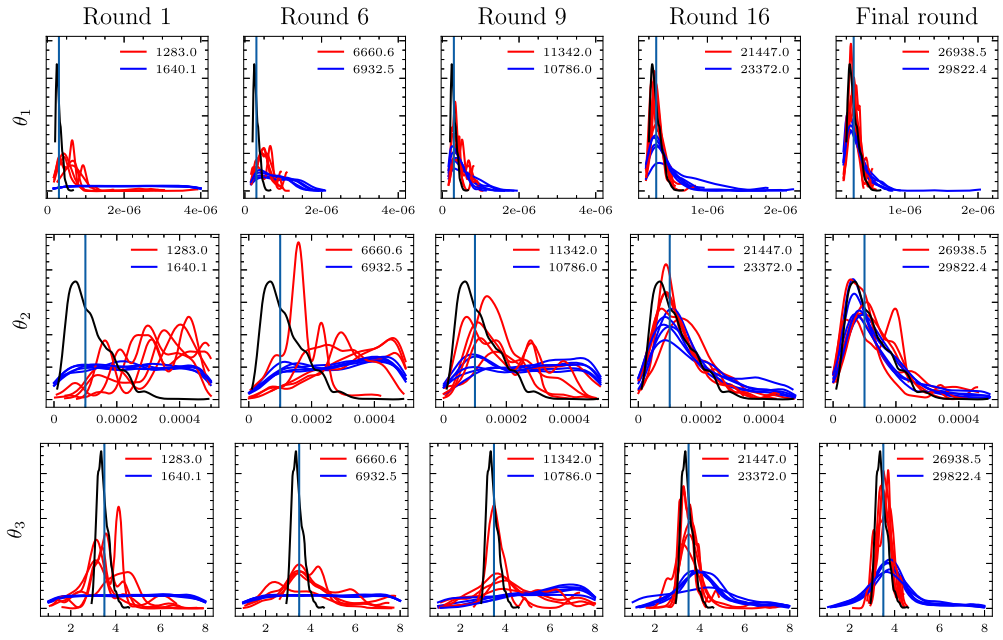

Figure 8: Schlögl model. Marginal posterior distributions for ABC-SMC-DC displayed in red, and ABC-SMC-F in blue. The reference marginal posteriors (via particle MCMC) are depicted in black, and the true parameter values as black vertical lines. Each panel reports, in the upper right corner, the number of seconds since the algorithm started.

diffusion bridge of Durham and Gallant (the version found in Whitaker et al., 2017) to impute points between observations. The results of the analysis can be seen in Figure 7. We notice that for θ_2, ABC-SMC-DC finds a region of high posterior density in the first round (318 seconds on average), whereas for ABC-SMC-F about 6 rounds are required (3206 seconds on average) to arrive at a similar posterior approximation, hence a 10-fold acceleration with ABC-SMC-DC. For ABC-SMC-DC, a crude posterior approximation is already obtained by round 6 with an average Wasserstein distance of 0.08 (0.28 for ABC-SMC-F), and displaying an inference quality that by round 11 is considerably more accurate than ABC-SMC-F, with an average Wasserstein distance of 0.015 (0.07 for ABC-SMC-F). We recall that a possible use of ABC-SMC-DC is to rapidly find the bulk of the posterior (say in six rounds in this case), and from there initialize the standard ABC-SMC-F.

Schlögl model

The Schlögl model is a notable instance of a reaction network showcasing bistability. It describes how, by a specific set of reactions, solutions of the deterministic representation gravitate towards one of two stable states and remain there indefinitely. In contrast, in stochastic versions of the model, the system can spontaneously alternate

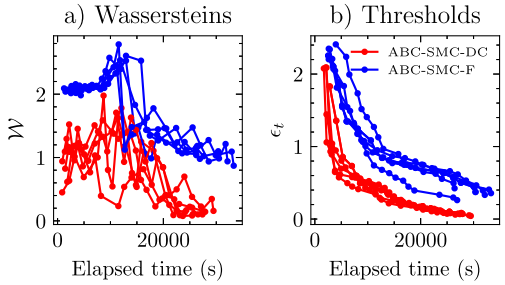

Figure 9: Schlögl model. a) the Wasserstein distance between the ABC-posteriors at each round and the reference posterior from particle MCMC is shown on the y-axis, and the cumulative elapsed time per round is shown on the x-axis. b) the ABC thresholds are shown on the y-axis, and the x-axis is the same as for a). A circle represents a round of ABC-SMC.

between these stable states due to inherent noise. The interval between switching events is unpredictable. This makes data from a single experiment challenging to replicate with simulated data in a particular experimental context, typically resulting in very low acceptance rates for ABC schemes. The Wasserstein results are presented in Figure 9(a), and the ABC thresholds are shown in Figure 9(b), based on five independent runs. Additionally, the posterior distributions from these five runs are visualized in Figure 8. ABC-SMC-DC effectively localizes the posterior region for θ_1 even in the first round. Conversely, ABC-SMC-F fails to find an accurate approximation of the θ_3 posterior within the given allocated time. ABC-SMC-DC achieves a much lower Wasserstein distance, and the ABC thresholds decrease much faster compared to ABC-SMC-F. This observation suggests a hybrid algorithm: once a sufficiently low threshold/Wasserstein distance is achieved, reverting to ABC-SMC-F can increase the inference accuracy at a lower computational cost.

8 Discussion

We have constructed a novel, efficient approach for parameter inference in stochastic differential equation (SDE) models when using approximate Bayesian computation (ABC) algorithms. Our sequential Monte Carlo ABC method with *data-conditional* simulation (ABC-SMC-DC) proposes trajectories resulting from smoothing approaches (using a carefully constructed combination of forward- and backward-simulated paths), producing a method converging much more rapidly to some reference posterior, as shown via Wasserstein distances. This is especially evident in the first round of our ABC-SMC-DC method, where the initial Wasserstein distance is much lower compared to the one produced from standard ABC-SMC where trajectories are instead myopically resulting from a forward model simulation. This often means that already at rounds 3–5 of our ABC-SMC-DC the posterior approximation is close to the bulk of the reference posterior, and moreover having to run very few rounds is a favorable feature, given that

at each round we also retrain a neural network to learn summary statistics, which is further discussed below. This could be exploited to create a "hybrid" algorithm, where our method is used for a few initial rounds allowing rapid converging to the bulk of the posterior, to then revert to purely forward ABC-SMC simulation for the remaining rounds, as the latter is computationally favorable when the approximated posterior is close to the target. Our method was tested on simulation studies that include several members of the Chan–Karaolyi–Longstaff–Sanders family of SDEs, and we found that the most dramatic display of the efficiency of our method comes from the more complex case studies, where the solution to the SDE is very erratic. We conjecture that more drastic accelerations can be achieved with even more challenging case studies with very erratic solution paths.

An important component of our proposed ABC-SMC-DC scheme with backward simulation is that the forward-pass in the procedure can be executed with any numerical discretization scheme, not just the customary Euler-Maruyama scheme. While in the present work we have illustrated our approach by using Euler-Maruyama in step 4 of the lookahead sampler (Algorithm 2), this need not be the case. Any higher-order scheme can be used in the forward-pass, and while this may imply that the scheme does not enjoy a closed-form expression for the associated transition density, the weights ω computed in step 4 of Algorithm 3 are only necessary towards obtaining a backward trajectory. Moreover, these weights could be computed with an approximate transition density such as the one resulting from the Euler-Maruyama or the Milstein schemes. This apparent contradiction means that, although when using weights ω from the latter two schemes to pick in the backward pass a trajectory from the cloud generated in the forward pass (which would instead use a higher-order method) we would have an inconsistency in the construction of the ω-weights in Algorithm 3, nevertheless this will allow to sample a backward trajectory that is picked from a set of accurate forward trajectories. This procedure could still be highly beneficial, as Buckwar et al. (2020) have shown that the ABC posterior can be inaccurate when employing the Euler-Maruyama scheme, for some SDE models. Another important feature of our ABC-SMC-DC is that it makes use of sequential learning of the summary statistics, which are refined after each round of ABC-SMC-DC, using the neural-network denoted PEN (Wiqvist et al., 2019), which is especially suited to Markov processes. However, we wish to emphasize that while PEN is a recommended choice to automatically construct summary statistics for SDEs, since PEN is by construction designed to exploit the Markovianity in the paths of SDEs solutions, the user may decide to use other ways to determine the summary statistics while still using our Algorithm 5. While this is entirely possible, the user should be careful in doing so, as in such case the synthetic likelihoods appearing in (23) may not be an appropriate approximation to the density of the summary statistics. The latter aspect is less risky when the summary statistic is an approximation to the parameters posterior mean (as with PEN, but also with the linear regression method of Fearnhead and Prangle, 2012, and the neural-network approaches of Jiang et al., 2017 and Akesson et al., 2021), thanks to central limit theorem arguments. In fact we only needed 30-50 paths to approximate the synthetic likelihoods via PEN. While PEN can so far be used only with one-dimensional SDEs, this does not prevent the use of our backwards simulation ABC-SMC with multidimensional SDEs, by constructing summary statistics using alternative methods.

Acknowledgments

We would like to express our gratitude to Dr. Massimiliano Tamborrino, University of Warwick, and Yanzhi Chen, University of Cambridge, for their comments and suggestions that enhanced the quality of this manuscript.

Funding

UP acknowledges funding from the Swedish National Research Council (Vetenskapsrådet 2019-03924). PJ and UP acknowledge funding from the Chalmers AI Research Centre.

Supplementary Material

Supplementary Material to: "Towards Data-Conditional Simulation for ABC Inference in Stochastic Differential Equations" (DOI: 10.1214/24-BA1467SUPP; .pdf). The supplementary material contains a rich amount of extra information, comprising both additional case studies and methodological details and figures

References

Akesson, M., Singh, P., Wrede, F., and Hellander, A. (2021). "Convolutional neural networks as summary statistics for approximate Bayesian computation." *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 334

Andrieu, C., Doucet, A., and Holenstein, R. (2010). "Particle Markov chain Monte Carlo methods." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342. MR2758115. doi: https://doi.org/10.1111/j.1467-9868.2009.00736.x. 310

Andrieu, C. and Roberts, G. O. (2009). "The pseudo-marginal approach for efficient Monte Carlo computations." *The Annals of Statistics*. MR2502648. doi: https://doi.org/10.1214/07-AOS574. 310

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. (2009). "Adaptive approximate Bayesian computation." *Biometrika*, 96(4): 983–990. MR2767283. doi: https://doi.org/10.1093/biomet/asp052. 313

Beskos, A., Papaspiliopoulos, O., Roberts, G. O., and Fearnhead, P. (2006). "Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion)." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3): 333–382. MR2278331. doi: https://doi.org/10.1111/j.1467-9868.2006.00552.x. 309, 315

Buckwar, E., Tamborrino, M., and Tubikanec, I. (2020). "Spectral density-based and measure-preserving ABC for partially observed diffusion processes. An illustration on Hamiltonian SDEs." *Statistics and Computing*, 30(3): 627–648. MR4065223. doi: https://doi.org/10.1007/s11222-019-09909-6. 334

Casella, B. and Roberts, G. O. (2011). "Exact simulation of jump-diffusion processes

with Monte Carlo applications." *Methodology and Computing in Applied Probability*, 13: 449–473. MR2822390. doi: https://doi.org/10.1007/s11009-009-9163-1. 309, 315

Chan, J., Perrone, V., Spence, J., Jenkins, P., Mathieson, S., and Song, Y. (2018). "A likelihood-free inference framework for population genetic data using exchangeable neural networks." *Advances in Neural Information Processing Systems*, 31. 325

Chan, K. C., Karolyi, G. A., Longstaff, F. A., and Sanders, A. B. (1992). "An empirical comparison of alternative models of the short-term interest rate." *The Journal of Finance*, 47(3): 1209–1227. 327

Chen, Y., Zhang, D., Gutmann, M., Courville, A., and Zhu, Z. (2020). "Neural approximate sufficient statistics for implicit models." *arXiv preprint arXiv:2010.10079*. 310, 325, 326

Craigmile, P., Herbei, R., Liu, G., and Schneider, G. (2022). "Statistical inference for stochastic differential equations." *Wiley Interdisciplinary Reviews: Computational Statistics*, e1585. MR4562291. doi: https://doi.org/10.1002/wics.1585. 310, 315

Cranmer, K., Brehmer, J., and Louppe, G. (2020). "The frontier of simulation-based inference." *Proceedings of the National Academy of Sciences*, 117(48): 30055–30062. MR4263287. doi: https://doi.org/10.1073/pnas.1912789117. 310

Del Moral, P., Doucet, A., and Jasra, A. (2006). "Sequential Monte Carlo samplers." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3): 411–436. MR2278333. doi: https://doi.org/10.1111/j.1467-9868.2006.00553.x. 313

Del Moral, P., Doucet, A., and Jasra, A. (2012). "An adaptive sequential Monte Carlo method for approximate Bayesian computation." *Statistics and Computing*, 22(5): 1009–1020. MR2950081. doi: https://doi.org/10.1007/s11222-011-9271-y. 313

Del Moral, P. and Garnier, J. (2005). "Genealogical particle analysis of rare events." *The Annals of Applied Probability*, 15(4): 2496–2534. MR2187302. doi: https://doi.org/10.1214/105051605000000566. 317

Del Moral, P. and Murray, L. M. (2015). "Sequential Monte Carlo with highly informative observations." *SIAM/ASA Journal on Uncertainty Quantification*, 3(1): 969–997. MR3403141. doi: https://doi.org/10.1137/15M1011214. 317, 318, 319

Fearnhead, P. and Prangle, D. (2012). "Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3): 419–474. MR2925370. doi: https://doi.org/10.1111/j.1467-9868.2011.01010.x. 310, 325, 334

Filippi, S., Barnes, C. P., Cornebise, J., and Stumpf, M. P. (2013). "On optimality of kernels for approximate Bayesian computation using sequential Monte Carlo." *Statistical Applications in Genetics and Molecular Biology*, 12(1): 87–107. MR3044402. doi: https://doi.org/10.1515/sagmb-2012-0069. 313

Forbes, F., Nguyen, H. D., Nguyen, T. T., and Arbel, J. (2021). "Approximate Bayesian computation with surrogate posteriors." *hal-03139256v4*. 310

Fuchs, C. (2013). *Inference for Diffusion Processes: With Applications in Life Sciences*. Springer Science & Business Media. MR3015023. doi: https://doi.org/10.1007/978-3-642-25969-2. 309, 315

Golightly, A. and Sherlock, C. (2022). "Augmented pseudo-marginal Metropolis–Hastings for partially observed diffusion processes." *Statistics and Computing*, 32(1): 21. MR4382657. doi: https://doi.org/10.1007/s11222-022-10083-5. 315

Golightly, A. and Wilkinson, D. J. (2015). "Bayesian inference for Markov jump processes with informative observations." *Statistical Applications in Genetics and Molecular Biology*, 14(2): 169–188. MR3331772. doi: https://doi.org/10.1515/sagmb-2014-0070. 317

Iacus, S. M. (2008). *Simulation and Inference for Stochastic Differential Equations: With R Examples*, volume 486. Springer. MR2410254. doi: https://doi.org/10.1007/978-0-387-75839-8. 315

Jiang, B., Wu, T.-y., Zheng, C., and Wong, W. H. (2017). "Learning summary statistic for approximate Bayesian computation via deep neural network." *Statistica Sinica*, 1595–1618. MR3701500. 310, 325, 334

Jovanovski, P., Golightly, A., and Picchini, U. (2024). "Supplementary Material for "Towards data-conditional simulation for ABC inference in stochastic differential equations"." *Bayesian Analysis*. doi: https://doi.org/10.1214/24-BA1467SUPP. 311

Lin, M., Chen, R., and Mykland, P. (2010). "On generating Monte Carlo samples of continuous diffusion bridges." *Journal of the American Statistical Association*, 105(490): 820–838. MR2724864. doi: https://doi.org/10.1198/jasa.2010.tm09057. 317

Lindsten, F., Schön, T. B., et al. (2013). "Backward simulation methods for Monte Carlo statistical inference." *Foundations and Trends® in Machine Learning*, 6(1): 1–143. 319, 320

Oksendal, B. (2013). *Stochastic Differential Equations: An Introduction with Applications*. Springer Science & Business Media. MR2001996. doi: https://doi.org/10.1007/978-3-642-14394-6. 309

Owen, J., Wilkinson, D. J., and Gillespie, C. S. (2015). "Scalable inference for Markov processes with intractable likelihoods." *Statistics and Computing*, 25: 145–156. MR3304916. doi: https://doi.org/10.1007/s11222-014-9524-7. 310

Panik, M. J. (2017). *Stochastic Differential Equations: An Introduction with Applications in Population Dynamics Modeling*. John Wiley & Sons. MR3644371. doi: https://doi.org/10.1002/9781119377399. 309

Papaspiliopoulos, O., Roberts, G. O., and Stramer, O. (2013). "Data augmentation for diffusions." *Journal of Computational and Graphical Statistics*, 22(3): 665–688. MR3173736. doi: https://doi.org/10.1080/10618600.2013.783484. 315

Picchini, U. (2014). "Inference for SDE models via approximate Bayesian computation." *Journal of Computational and Graphical Statistics*, 23(4): 1080–1100. MR3270712. doi: https://doi.org/10.1080/10618600.2013.866048. 315, 325

Picchini, U. and Tamborrino, M. (2024). "Guided sequential ABC schemes for intractable Bayesian models." *Bayesian Analysis*. doi: https://doi.org/10.1214/24-BA1451. 313, 314

Prangle, D. (2017). "Adapting the ABC distance function." *Bayesian Analysis*, 12(1). MR3620131. doi: https://doi.org/10.1214/16-BA1002. 327

Price, L. F., Drovandi, C. C., Lee, A., and Nott, D. J. (2018). "Bayesian synthetic likelihood." *Journal of Computational and Graphical Statistics*, 27(1): 1–11. MR3788296. doi: https://doi.org/10.1080/10618600.2017.1302882. 310

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., and Feldman, M. W. (1999). "Population growth of human Y chromosomes: a study of Y chromosome microsatellites." *Molecular Biology and Evolution*, 16(12): 1791–1798. 312

Särkkä, S. (2013). *Bayesian Filtering and Smoothing*. 3. Cambridge University Press. MR3154309. doi: https://doi.org/10.1017/CBO9781139344203. 320

Särkkä, S. and Solin, A. (2019). *Applied Stochastic Differential Equations*, volume 10. Cambridge University Press. MR3931353. 309

Sisson, S. A., Fan, Y., and Beaumont, M. (2018). *Handbook of Approximate Bayesian Computation*. CRC Press. MR3889281. 310, 313

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007). "Sequential Monte Carlo without likelihoods." *Proceedings of the National Academy of Sciences*, 104(6): 1760–1765. MR2301870. doi: https://doi.org/10.1073/pnas.0607208104. 313

Steele, J. M. (2001). *Stochastic Calculus and Financial Applications*, volume 1. Springer. MR1783083. doi: https://doi.org/10.1007/978-1-4684-9305-4. 309

Tavaré, S., Balding, D. J., Griffiths, R. C., and Donnelly, P. (1997). "Inferring coalescence times from DNA sequence data." *Genetics*, 145(2): 505–518. 312

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. (2009). "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems." *Journal of the Royal Society Interface*, 6(31): 187–202. 313

van der Meulen, F. and Schauer, M. (2017). "Bayesian estimation of discretely observed multi-dimensional diffusion processes using guided proposals." *Electronic Journal of Statistics*, 11: 2358–2396. MR3656495. doi: https://doi.org/10.1214/17-EJS1290. 315

Whitaker, G. A., Golightly, A., Boys, R. J., and Sherlock, C. (2017). "Improved bridge constructs for stochastic differential equations." *Statistics and Computing*, 27: 885–900. MR3627552. doi: https://doi.org/10.1007/s11222-016-9660-3. 332

Wilkinson, D. J. (2018). *Stochastic Modelling for Systems Biology*. Chapman and Hall/CRC, 3rd edition. MR2222876. 309, 330

Wiqvist, S., Mattei, P.-A., Picchini, U., and Frellsen, J. (2019). "Partially exchangeable networks and architectures for learning summary statistics in approximate Bayesian computation." In *International Conference on Machine Learning*, 6798–6807. PMLR. 310, 311, 325, 334

Wood, S. N. (2010). "Statistical inference for noisy nonlinear ecological dynamic systems." *Nature*, 466(7310): 1102–1104. 310, 323