Multifidelity Bayesian Verification

Gareth W. Molyneux and Alessandro Abate*

*Correspondence:
aabate@cs.ox.ac.uk
Department of Computer Science, University of Oxford, Oxford, UK
Full list of author information is available at the end of the article

Abstract

In this work we present a scalable approach to perform Bayesian verification over uncertain models of Chemical Reaction Networks (CRN). We blend two main procedures, one for data-based model verification and the other for model inference; we then expand their combination to a multifidelity setup. More precisely, we perform verification via (Bayesian) statistical model checking (SMC) based on model simulations; we at the same time update uncertain models by means of likelihood-free Approximate Bayesian Computation (ABC) schemes; finally, we push the scalability of the approach by newly leveraging a multifidelity setup (MF), whereby high- and low-fidelity models are traded off, also depending on their closeness. The possible different interleaving of these three parts comprise the core of the overall procedure, MF-ABC-(SMC)^2, which is tested on a number of diverse case studies for CRN.

Keywords: Stochastic Processes; Continuous-time Markov Chains; Stochastic Differential Equations; Formal Verification; Supervised Machine Learning; Likelihood-free Bayesian Inference; Multifidelity Models; Chemical Reaction Networks

Background

Problem under Study

Constructing accurate and explainable models of biological systems is a prevalent activity in systems and synthetic biology. Models allow to encompass prior knowledge, to extract more information from data, and lead to fine-grained analysis and prediction of the biological systems of interest. Attaining precise knowledge of many existing biological systems is however hard, making the formal analysis of their models a challenge and requiring models to encompass degrees of uncertainty. Data-driven, statistical approaches, such as Bayesian inference, allow to learn and update models as data becomes available. Similarly, statistical approaches can be leveraged to analyse and verify built models, particularly as their complexity and dimensionality grows (which is often the case for industrially-relevant biological systems), thus providing an alternative to formal mathematical approaches.

Contributions

We introduce a scalable framework that deploys multifidelity methods [1] combined with Bayesian inference and formal verification, which allows for model-based probabilistic verification of data-generating stochastic biological systems. Specifically, we extend earlier work integrating Bayesian inference and statistical model checking (SMC) [2], with new multifidelity methods recently developed for Bayesian learning [3, 4]. The reference technique in [2] allows for simultaneous inference and formal verification on a wide variety of stochastic models of interest, but possibly relies on

computationally costly simulations: here we make the use of multifidelity models to reduce the *computational cost* of the required simulations. Furthermore, in order to reduce not only the *cost* but also the required *number* of simulations, we make use of a Bayesian SMC algorithm based on a small subset of the sampled model parameters which are carefully chosen, rather than naïvely running a costly (frequentist) SMC algorithm for every sampled parameter, as in [2]. By running SMC on samples generated from the prior for instance, we are able to perform a parameter classification, which is needed for the overall procedure, with fewer simulations than in [2]. Also, we deploy a multifidelity (Bayesian) SMC algorithm allowing one to explore the statistical relation between the high- and low-fidelity models. By blending such techniques, we reduce the number of simulations as well as the computational cost of these simulations which culminates in the MF-ABC-(SMC)^2 algorithm, a new scalable multifidelity Bayesian verification approach.

Approach and Structure of this Article

Our new proposed framework can be crisply described as follows. We are given a property of interest (expressed as a formal specification in a relevant temporal logic), a class of high-fidelity (computationally expensive but accurate) and low-fidelity (computationally cheaper but less precise) parametrised models, and data from the underlying biological system (a CRN) - these ingredients are introduced in Section [1]. We rapidly infer parameters and perform statistical model checking,leveraging multifidelity methods and either Bayesian or frequentist techniques. We quantify the probability on whether or not the biological system that generates the data satisfies the given property, by integrating the results of the approximate (Bayesian) inference and of statistical model checking.We argue, both technically and through a number of case studies, that the approach requires fewer simulations, which are additionally computationally cheaper to generate, compared to earlier results in [2, 5]. We conclude the article with a discussion on the breadth of the proposed framework, and consequently on the many possible avenues for future work.

Related Recent Work

Learning models from data via Bayesian inference has been widely studied within the realms of biology [6, 7]. Due to the intractability of the likelihood function [8] of many models of interest, likelihood-free methods such as Approximate Bayesian Computation (ABC) [9, 10] have grown in popularity. In this work we build on the Bayesian verification framework introduced in [2, 5], which integrates likelihood-free inference and statistical model checking (SMC), incorporating multifidelity methods [1] into the framework to increase scalability. Beyond this extension of earlier work by the authors, this contribution also compares Bayesian [11, 12] vs. frequentist approaches to SMC, and further discusses how to possibly interleave the latter (SMC) step within the sample-based inference algorithms, thus trading off speed for generality and accuracy. Newer case studies and benchmarks further distinguish this contributions from [2, 5].

[1] We emphasise that, due to our likelihood-free inference technique, with this framework we are able to work with a wide variety of (high-fidelity) stochastic models, provided we have a low-fidelity model equivalent, and that we may also verify properties defined in various formal logics of our choice, provided that such properties can be assessed by means of a statistical model checking technique.

Multifidelity methods integrate information from multiple models of the same system to accelerate tasks such as inference and uncertainty quantification. Applications of multifidelity methods, such as Multilevel Monte Carlo [13, 14], have been practically successful, and have been recently exploited for likelihood-free inference using ABC [15]. For our work, we utilise multifidelity methods to accelerate the parameter inference part of our framework, specifically working with Multifidelity ABC [3] and its sequential Monte Carlo extension [4].

The Bayesian verification framework was proposed in [16], later extended to Markov models in [17, 18], and finally endowed with SMC techniques and applied to CRNs in [2, 5]. It leverages results in probabilistic and statistical model checking, which have been applied successfully to a wide variety of biological models [11, 19–23]. Parametric verification has been considered from a statistical approach underpinned by Gaussian Processes: smoothed Model checking [24] provides an estimate of the satisfaction probability with uncertainty estimates, and has been used for parameter estimation from Boolean observations [25] and for parameter synthesis [26]. Closely related work that integrates ABC-based inference with verification is the the Automaton-ABC presented in [27, 28]. This scheme similarly computes the satisfaction probability of a temporal specification as a function of the parameters of a given stochastic model. However quite distinctively, whilst in our work the distance metrics considered relate to distance between generated trace (or simulations) and the data observed from a data-generating system, in [27] the ABC framework relies on a distance measure that quantifies how far a generated trace from a stochastic model is from satisfying the property of interest. The use of an automaton-based formalism for properties, as in [29], allows to work with more sophisticated properties than those that can be defined using CSL.

Models and Properties

Although our methodology can be applied to a number of parametrised stochastic models, in view of the applications of interest we work with discrete-state, continuous-time Markov chains [30] and stochastic differential equations [31].

Parametric Continuous-Time Models

Let us first present the definition of continuous-time Markov chains.

Definition 1 (Continuous-time Markov Chain) *A continuous-time Markov chain (CTMC) $\mathcal{M}$ is a tuple $(\mathcal{X}, \mathcal{T}, x_0)$, where*

- *$\mathcal{X}$ is a finite, non-empty set of enumerable states,*
- *x_0 is the initial state of the CTMC,*
- *$\mathcal{T} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ is the transition rate matrix, where $\mathcal{T}(x, x')$ is the rate of transition from state x to state x'.*

We additionally define a labelling function $L : \mathcal{X} \to 2^{AP}$ which maps each state, $x \in \mathcal{X}$, to the set $L(x) \subseteq AP$ of atomic propositions in AP that hold true in x. This will be relevant to specify properties defined over models. The transition rate matrix $\mathcal{T}$ governs the dynamics of the overall model.

Definition 2 (Path of a CTMC) *Let $\mathcal{M} = (\mathcal{X}, \mathcal{T}, x_0)$ be a CTMC. An infinite path of a CTMC $\mathcal{M}$ is a non-empty sequence $\omega_\infty = x_0 t_0 x_1 t_1 \dots$ where $\mathcal{T}(x_i, x_{i+1}) > 0$ and $t_i \in \mathbb{R}_{>0}$ for all $i \geq 0$. In this work, we focus on finite paths, namely sequences $\omega = x_0 t_0 x_1 t_1 \dots \; x_{k-1} t_{k-1} x_k$ such that x_k is absorbing. The value t_i represents the amount of time spent in the state x_i before jumping to the next state in the chain, namely state x_{i+1}. We denote by $Path^{\mathcal{M}}(x)$ the set of all paths of the CTMC $\mathcal{M}$ starting in state x. A trace of a CTMC is the mapping of a path through the labelling function L.*

Parametric CTMCs extend the notion of CTMC by allowing transition rates to depend on a vector of parameters $\theta = (\theta_1, \dots, \theta_k) \in \mathbb{R}^k$. The range of possible values of each parameter θ_i is given by a closed and bounded real interval $[\theta_i^\perp, \theta_i^\top]$. The parameter space Θ is the hyper-rectangle defined as the Cartesian product of the individual intervals, $\Theta = \bigtimes_{i \in \{1,\dots,k\}} [\theta_i^\perp, \theta_i^\top]$. Finally, $\mathcal{T}_\theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}[\theta]$ is the parametric rate matrix, where $\mathbb{R}[\theta]$ denotes a set of polynomials over $\mathbb{R}^+$ with variables θ_k, $\theta \in \Theta$. Given a pCTMC and a parameter space Θ, we denote with $\mathcal{M}_\Theta$ the set $\{\mathcal{M}_\theta, \theta \in \Theta\}$ where $\mathcal{M}_\theta$ is the instantiated CTMC obtained by replacing the parameters in $\mathcal{T}$ with their valuation in θ. So a standard CTMC is induced by selecting a specific parameter $\theta \in \Theta$. For the models in this paper, we assume that x_0 is unique and deterministically given, although the parametrisation can be trivially extended to encode sets of initial conditions.

We now define stochastic differential equations (SDEs), which are analogous to CTMCs in depending on a continuous time variable, but which unlike CTMCs evolve over a continuous space. Let $(\Omega, \mathcal{F}, P)$ be a probability space where Ω is a sample space, $A \in \mathcal{F}$ an event in the sample space $\mathcal{F}$, and $\omega \in \Omega$ is a sample or outcome.

Definition 3 (Stochastic Differential Equation) *We define a stochastic process on the probability space $(\Omega, \mathcal{F}, P)$ as $\tilde{x}(t), t \geq 0 : \mathbb{R}_{\geq 0} \times \Omega \to \mathbb{R}^d$ and we describe the evolution of this continuous stochastic process with the stochastic differential equation (SDE) defined by*

$$\mathrm{d}\tilde{x}^i(t) = \varsigma_0^i(\tilde{x}(t))\ \mathrm{d}t + \sum_{j=1}^{J} \varsigma_j^i(\tilde{x}(t))\ \mathrm{d}W^j(t),\ \ i = 1, 2, \dots, d,$$

where $\tilde{x}(0) \in \mathbb{R}^d$ is the initial condition with state space of dimension d, $W(t) = [W^1(t), \dots, W^J(t)]^\top_{t \geq 0}$ is a standard Brownian Motion in $\mathbb{R}^J$ on $(\Omega, \mathcal{F}, P)$ and $\varsigma_j^i : \mathbb{R}^d \to \mathbb{R}$, $i = 1, 2, \dots, d$, $j = 0, 1, 2, \dots, J$ are Borel measurable functions under proper structural conditions, which ensure the existence and uniqueness of solutions to the SDEs [31].

We can similarly introduce a labelling map over the state space of the SDE, as done previously for the case of CTMCs. Later we shall employ the Chemical Langevin Equation (CLE) as a low-fidelity abstraction of CTMCs, and thus denote them as $\tilde{\mathcal{M}}$. We introduce the semantics of SDE over discrete time steps: an SDE can be numerically integrated in time using various techniques, for instance via the

known Euler-Maruyama scheme [32], which is presented in detail in the section on Methods.

In this work we shall exclusively focus on finite time horizons up to a terminal time T, $t \in [0, T]$: let $\Delta t = T/K$ for some $K \in \mathbb{N}$, then $\tau_k = k\Delta t$, where $k = 1, 2, \ldots, K$, is a uniform sequence of sampling times. With the Euler-Maruyama discretisation scheme,the definition of a path of the SDE follows intuitively as a sequence $\Omega = \tilde{x}_0 t_0 \tilde{x}_1 t_1 \ldots t_{K-1} \tilde{x}_K$ comprising sampled times $t_k = \tau_k$ and corresponding states $\tilde{x}_k \in \mathbb{R}^d$ for $k = 1, 2, \ldots, K$. We denote by $Path^{\tilde{\mathcal{M}}}(\tilde{x})$ the set of all finite (and, by extension, infinite) paths of the SDE $\tilde{\mathcal{M}}$ starting in state $\tilde{x}$. We can also work with parametrised SDEs, which is defined similarly to SDEs above and shall later be denoted as $\tilde{\mathcal{M}}_\theta$. Let $\theta = (\theta_1, \ldots, \theta_k)$ be the vector of parameters, taking values in $\Theta \subset \mathbb{R}^k_{\geq 0}$. A parametric stochastic differential equation (pSDE) over θ is defined similarly to the SDE in Definition 3, except that ς^i_j are now a function of the parameters, $\varsigma^i_j : \mathbb{R}^k \times \mathbb{R}^J \to \mathbb{R}$, $i = 1, 2, \ldots, d$, $j = 0, 1, 2, \ldots, J$, namely $\varsigma^i_j = \varsigma^i_j(\theta, \tilde{x}(t))$.

Chemical Reaction Networks

In this work we deal with Chemical Reaction Networks (CRNs), which are modelled as either CTMC or as SDE.

Definition 4 (Chemical Reaction Network) *A Chemical Reaction Network (CRN) is a tuple $(X, \mathcal{R}, K)$, where*

- *$X = (X_1, ..., X_{N_S})$ is a vector where each X_i represents the number of molecules of each species $i = 1, \ldots N_S$. $X \subseteq \mathbb{N}^{N_S}$ the state space,*
- *$\mathcal{R} = \{R_1, \ldots, R_{N_R}\}$ is the set of N_R chemical reactions, each of the form $R_j = (\boldsymbol{\upsilon}_j, a_j)$, with $\boldsymbol{\upsilon}_j$ the stoichiometry vector of size N_S for reaction j and $a_j(X) = a_j(X, k_j)$ is the propensity or rate function,*
- *$\Bbbk = (k_1, \ldots, k_{N_R})$ is the vector of (kinetic) parameters, taking values in a compact subset of $\mathbb{R}^{N_R}$.*

Each reaction j of the CRN is represented as

$$R_j : \sum_{i=1}^{N_S} \upsilon^-_{i,j} X_i \xrightarrow{k_j} \sum_{i=1}^{N_S} \upsilon^+_{i,j} X_i,$$

where $\upsilon^-_{i,j}$ ($\upsilon^+_{i,j}$) is the amount of species X_i consumed (produced) by reaction R_j. This results in the stoichiometric matrix $\upsilon_{i,j} = \upsilon^+_{i,j} - \upsilon^-_{i,j}$, which allows one to describe how the state vector $X(t)$ changes in time when reactions occur. Specifically, if a reaction j occurs, then the state vector $X(t)$ is updated by adding the stoichiometric vector $\boldsymbol{\upsilon}_j = [\upsilon_{1,j}, \upsilon_{2,j}, \ldots, \upsilon_{N_S,j}]^\top$. The probability of such an event is encapsulated within the propensity function $a_j(X(t))$, defined such that $a_j(X(t))dt$ is the probability, given $X(t)$, that reaction R_j will occur in an infinitesimal time interval $[t, t + dt)$ [33, 34].

CRN models: CTMC

CRNs can be modelled by CTMCs. In this work CTMCs are employed as "high-fidelity" models. A CTMC at time t models the state vector representing the number of molecules of each chemical species at that time, $X(t) = [X_1(t), X_2(t), \ldots, X_{N_S}(t)]^\top$ and the state space $\mathcal{X}$ comprises every possible combination of molecules. The rate of transitioning from state to state is governed by the kinetic parameters that relate to the rate of the reactions. The initial state of the CTMC is given as $x_0 = X(0) = [X_1(0), X_2(0), \ldots, X_{N_S}(0)]^\top$. We can additionally model a CRN as a pCTMC if we were to parametrise the kinetic reaction rates: we let $\mathcal{M}_\theta$ be the pCTMC, where θ is the vector comprising the unknown kinetic rates.

Simulations of a CRN, modelled by a pCTMC, can be generated using the Stochastic Simulation Algorithm (SSA) [33]. The SSA allows one to simulate independently a reaction index (among the possible N_R reactions) and a reaction time for each reaction that occurs, providing a highly detailed simulation at a high computational cost: at each iteration of the SSA algorithm, both a reaction time and index must be drawn to calculate the associated propensity function, which also updates the state vector at every stage. To circumvent this problem, we can simulate the number of reactions that can occur in a fixed time interval of length Δt. This method leads to the tau-leaping [35] and, core to our work, to the Chemical Langevin Equation (CLE) [34, 35], which is an approximation to the SSA algorithm that allows for a speed-up of simulation time, but which introduces errors that will be small as long as the updates are relatively small.

CRN models: Chemical Langevin Equation

In this work we utilise the chemical Langevin equations (CLEs) as the "low-fidelity" models of CRNs. The CLE models the CRN using N_S SDEs, where each SDE corresponds to the evolution of a chemical species over a time interval, $t \in [0, T]$. The full derivation of the CLE can be found in the Methods.The evolution of the ith chemical species, denoted by the ith element of the state vector, $X_i(t)$, is described by the corresponding variable $\tilde{x}^i(t)$,

$$\tilde{x}^i(t + \Delta t) = \tilde{x}^i(t) + \sum_{j=1}^{N_R} \upsilon_{i,j} a_j(Y(t))\Delta t + \sum_{j=1}^{N_R} \upsilon_{i,j} \sqrt{a_j(Y(t))\Delta t}\mathcal{Z}_j, \tag{1}$$

where $\mathcal{Z} \sim \mathcal{N}(0, 1)$. The SDE in (1) is in fact an Euler-Maruyama discretisation [36] with uniform time samples of duration Δt of an SDE of the form [31, 32]

$$\mathrm{d}\tilde{x}(t) = \sum_{j=1}^{N_R} \boldsymbol{\upsilon}_j a_j(\tilde{x}(t))\, \Delta t + \sum_{j=1}^{N_R} \boldsymbol{\upsilon}_j \sqrt{a_j(\tilde{x}(t))}\, \mathrm{d}W^j(t),$$

where $W^j(t)$ are independent scalar Brownian motions. We note that the dimensionality of the Brownian motion, d, depends on the number of different chemical species present in the CRN, thus $d = N_S$, and that dimensionality of the stochastic process depends on the total number of chemical reactions in the CRN, $J = N_R$, using d and J from Definition 3.

We have presented the CLE approximation to the evolution of a biochemical reaction network. This approximation is only appropriate when certain conditions are met (particularly when $a_j(X(t))\Delta t \gg 1$) and we go from discrete molecule counts, $X(t)$, to continuous molecule counts, $\tilde{x}(t)$, due to the use of the Normal approximation to the Poisson distribution. By taking advantage of this approximation we are able to simulate a larger number of reactions within a given timestep Δt, rather than simulating each individual reaction that fires, allowing for computational savings and faster simulation times. These savings and faster simulating times come at a price however, as the CLE is in practice a good approximation to the underlying dynamics only when the biological system presents a large number of molecules. Having defined the stochastic models of interest, we consider now how to formally define properties over these models.

Properties - Continuous Stochastic Logic

We wish to verify properties over CRNs and their pCTMC and pSDE interpretation, and employ a time-bounded, non-nested fragment of *continuous stochastic logic* (CSL) [19, 37, 38].

CSL is a temporal and quantitative logic that is sufficient for our purposes, as also evidenced by its wide use in systems biology literature. This is particularly so in terms of expressivity and ease of use within verification tasks.

Definition 5 *A CSL formula ϕ is interpreted over states $x \in \mathcal{X}$ of a model (e.g., a pCTMC $\mathcal{M}_\theta$), and φ is a formula over its paths, with syntax*

$$\phi := true \mid a \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid P_{\Cap\zeta}[\varphi] \ , \qquad \varphi := \ \phi_1 U^{[t,t']} \phi_2 \ ,$$

where $a \in AP$, $\Cap \in \{<, \leq, \geq, >\}$, $\zeta \in [0,1]$, and $t, t' \in \mathbb{R}_{\geq 0}, t \leq t'$.

Path formulae are defined by combining state formulae through the temporal operator $\phi_1 U^I \phi_2$, which is true if ϕ_2 is satisfied at some $\tau \in I$ and ϕ_1 holds at all preceding time instants [19]. Notice that, with very limited loss of expressivity, we have excluded the *next* operator, which can meaningfully apply to CTMCs, but not to SDEs.

We consider CSL formulae over models by associating atomic propositions to their state spaces via the model's labelling function L. Let $\mathcal{M}_\theta$ denote a parametrised stochastic model of choice, with associated initial state x_0, and set of paths $Path^{\mathcal{M}_\theta}(x_0)$. We qualitatively discuss the semantics of satisfaction of CSL formulae, noting that later we shall compute such satisfaction by means of statistical (rather than probabilistic) model checking approaches, namely by means of simulations of the models of interest.

The semantics of satisfaction of the propositional fragment of CSL is trivially related to the initial state of a path and to its associated label. Conversely, the semantics of the probabilistic operator can be explained as follows. Formula $P_{\Cap\zeta}[\varphi]$ holds if the probability of the path formula φ being satisfied by the set of paths originating from a given initial state meets the inequality $\Cap\ \zeta$. The path formula φ describes the bounded-horizon temporal requirement $\phi_1 U^{[t,t']} \phi_2$, widely known

as the "until" specification. A path satisfies such requirement if the latter label ϕ_2 is met by a path within the time interval $[t, t']$, whilst the earlier label ϕ_1 has continuously held beforehand. The measurability of the concerned (temporal) events depends on the underlying probability spaces of the model under study: for a CTMC this has been argued in [39] and can be similarly extended for SDE models, assuming labelling functions are measurable and have non-trivial co-domains, namely labels are associated to sets of non-zero measure, see [40] for further details.

The *satisfaction function* captures the probability of a property for a model (here a pCTMC, but this holds also for pSDEs) relates to its parameters and initial state. This function will be used later to classify the parameters (and the corresponding models) in the set Θ according to their property satisfaction. We note that the notation involved with the satisfaction function above is aligned with pCTMCs, however it is be straightforward to define an equivalent satisfaction function for pSDEs.

Definition 6 (Satisfaction Function) *Let ϕ be a CSL formula, $\mathcal{M}_\theta$ be a stochastic parametrised model over a space Θ, x_0 is the initial state, and $Path^{\mathcal{M}_\theta}(x_0)$ is the set of all paths generated by a parametrised stochastic model $\mathcal{M}_\theta$ with initial state x_0. Denote by $\Lambda_\phi : \theta \to [0, 1]$ the satisfaction function such that*

$$\Lambda_\phi(\theta) = P\left(\{\omega \in Path^{\mathcal{M}_\theta}(x_0) \models \varphi\} \mid \omega(0) = x_0\right),$$

where a path $\omega \models \varphi$ if its associated trace satisfies the path formula φ corresponding to the CSL formula ϕ. That is, $\Lambda_\phi(\theta)$ is the probability that the set of paths from a given parametrised stochastic model $\mathcal{M}_\theta$ satisfies a property φ. If $\Lambda_\phi(\theta) \Cap \zeta$, then we say that $\mathcal{M}_\theta \models \phi$.

Bayesian Inference

Given a set of observations or data, $y_{obs} \in \mathcal{Y}$, and a parametrised model $\mathcal{M}_\theta$, the task of Bayesian inference is to learn model parameters via probability distributions. Here $\mathcal{Y}$ is a general space that contains observations, which will be later related to the state space (and corresponding variables) of our models of interest. Prior beliefs about the model parameters, expressed through a probability distribution $\pi(\theta)$, are updated via y_{obs}, where assumptions on the model dynamics are encoded into the likelihood function $p(y_{obs} \mid \theta)$, which depends on the stochastic model $\mathcal{M}_\theta$. Using Bayes' theorem, the posterior distribution is calculated by $\pi(\theta \mid y_{obs}) = p(y_{obs} \mid \theta)\pi(\theta)/p(y_{obs})$.

For many complicated models that exhibit interesting dynamics, such as general CRNs, the likelihoods required to calculate the posterior probability distribution cannot be calculated directly, or are deemed to be computationally intractable. One can thus resort to likelihood-free methods, such as Approximate Bayesian Computation (ABC)[9], to approximate this posterior as $\pi_{ABC}(\theta \mid y_{obs}) \approx \pi(\theta \mid y_{obs})$. We resort to likelihood-free methods for inferring parameters of CRN networks from noisy data observed at discrete points in time. We assume that the observation of the state vector sample path is $y_{obs}(t) = O(X(t))$, where O is an arbitrary observation function on $X(t)$ that takes values in $\mathcal{Y}$, the space of observations. We shall

additionally leverage simulations from models that generate data that is appropriate for these observations - in particular, sampling CTMCs and SDEs at specific time instances generates data points that can be related to observations.

Approximate Bayesian Computation

In ABC methods [9], the likelihood is approximated by matching simulated data $y \sim p(y \mid \theta)$ from a model, where $y \in \mathcal{Y}$, with the observed data y_{obs}, according to some function of a distance metric, $d(y, y_{obs})$. Ideally, the physical observations y_{obs} are directly mapped to the simulations y generated by the model.

The simplest form of ABC is referred to as *rejection sampling*: samples are generated by $\theta^* \sim \pi(\theta)$, each of which is used to generate simulated data $y \sim p(y \mid \theta^*)$, where the proposed sample θ^* is accepted if $d(y, y_{obs}) \leq h$ for some $h \geq 0$, $h \in \mathbb{R}^+$, and rejected if $d(y, y_{obs}) > h$. This is equivalent to drawing a sample (θ, y) from the joint distribution $\pi_{ABC}(\theta, y \mid y_{obs}) \propto K_h(d(y, y_{obs}))p(y \mid \theta)\pi(\theta)$, where $K_h(\cdot)$ is a standard smoothing kernel function [41], here $K_h(d(y, y_{obs})) = \mathbb{1}_{\Omega_h}(y)$, where $\Omega_h(y) = \{y \in \mathcal{Y} \mid d(y, y_{obs}) \leq h\}$ and $d(y, y_{obs}) = \|y - y_{obs}\|$. The ABC approximation to the true posterior distribution is $\pi_{ABC}(\theta \mid y_{obs}) = \int \pi_{ABC}(\theta, y \mid y_{obs})\,\mathrm{d}y$. As $h \to 0$, samples from the true posterior distribution are obtained[9] as:

$$\lim_{h \to 0} \pi_{ABC}(\theta \mid y_{obs}) \propto \int \delta_{y_{obs}}(y) p(y \mid \theta)\pi(\theta)\,\mathrm{d}y = p(y_{obs} \mid \theta)\pi(\theta),$$

where $\delta_{y_{obs}}(y)$ is the Dirac delta measure. It is common and computationally advantageous to use sufficient summary statistics $s = S(y)$ for the datasets y_{obs} and y, ideally so that $\pi(\theta \mid y_{obs}) = \pi(\theta \mid s_{obs})$ [42]. We present the ABC rejection sampling scheme in the Methods, Algorithm 3.

Approximate Bayesian Computation - Sequential Monte Carlo

The major issue with the standard ABC procedure introduced previously is that if the prior $\pi(\theta)$ differs from the posterior distribution, $\pi(\theta \mid y_{obs})$, then the rates at which sampled parameters are accepted, will be low. Approximate Bayesian Computation - Sequential Monte Carlo (ABC-SMC) [10, 43] techniques are developed to mitigate this issue: they construct a sequence of intermediate distributions, $f_m(\theta)$, $m = 0, \ldots, M$, where $f_0(\theta) = \pi(\theta)$ is the initial sampling distribution and $f_M(\theta) = f(\theta)$ is the target distribution of interest, namely the approximated posterior, $\pi_{ABC}(\theta \mid s_{obs})$. A population of particles or samples from generation m, $\theta_m^{(i)}$, where $i = 1, \ldots, N$, are sequentially propagated between these intermediary distributions that act as importance sampling [9]. The success of importance sampling hinges on the choice of the importance probability distribution which generates sample proposals. Consider an importance probability distribution of the form $\hat{q}(\theta) = q(\theta)/Z_q$, where $q(\theta) > 0$ for all θ in the support of $\pi(\theta)$ and Z_q is a normalisation constant. The ABC importance sampling (ABC-IS) distribution first samples a parameter $\theta^* \sim \hat{q}(\theta)$ and simulates a corresponding trace $y \sim p(y \mid \theta^*)$ which is accepted or rejected with a corresponding weight $w(\theta^*, y) = [\pi(\theta^*)/q(\theta)] \cdot \mathbb{1}_{\Omega_h}(y) \geq 0$, to produce a weighted sample $\{\theta^{(i)}, w^{(i)}\}$. The ABC-RS algorithm (Algorithm 3) corresponds to the ABC-IS (Algorithm 4) when the importance distribution is chosen to be equal to the prior $\hat{q} = \pi$ and the number of simulations for each sampled

parameter is set to $B_m = 1$. The ABC-SMC technique (presented in Algorithm 5) attempts to bridge the gap between the prior $\pi(\theta)$ and the (unknown) posterior $\pi(\theta \mid s_{obs})$ or $\pi(\theta \mid y)$, where $s_{obs} = S(y_{obs})$, using importance sampling via intermediary distributions, chosen as

$$f_m(\theta) = \pi_{ABC}^{h_m}(\theta, s \mid s_{obs}) \propto \mathbb{1}_{\Omega_{h_m}}(s) p(y \mid \theta)\pi(\theta),$$

where again $s = S(y)$, $m = 0, \ldots, M$, and where h_m is a monotonically decreasing sequence, namely such that $h_m > h_{m+1} \geq 0$. We expect that the limit $\lim_{h_m \to 0} \pi_{ABC}^{h_m}(\theta \mid s_{obs}) = \pi(\theta \mid s_{obs})$ [9], and that the more samples N are generated, the more accurate the approximated quantity becomes.

A key part of the ABC-SMC scheme is the generation of samples θ^* and the setting of weights. Sample θ^* is initially ($m = 0$) taken from the prior and subsequently ($m > 0$) sampled from the intermediary distributions $f_{m-1}(\theta)$ through its corresponding weights (see below), as parameter $\theta_{m-1}^{(j)}$. If $m > 0$, θ^* is perturbed into θ^{**} by a kernel, $F_m(\theta^{**} \mid \theta^*)$ as demonstrated in Algorithm 5. For the perturbed parameter θ^{**}, a number B_m of simulations y_b (and, in turn, s_b), are generated from $p(y \mid \theta^{**})$, and the quantity $b_m(\theta^{**}) = \sum_{b=1}^{B_m} \mathbb{1}_{\Omega_{h_m}}(s_b)$ is calculated. If $b_m(\theta^{**}) = 0$, then θ^{**} is discarded and θ^* resampled again. Otherwise, the accepted θ^{**} results in the pair $\{\theta_m^{(i)}, w_m^{(i)}\}$, where the corresponding weights $w_m^{(i)}$ are set to

$$w_m^{(i)} = \begin{cases} b_m\left(\theta_m^{(i)}\right), & \text{if } m = 0 \\ \dfrac{\pi\left(\theta_m^{(i)}\right) b_m\left(\theta_m^{(i)}\right)}{\sum_{j=1}^{N} w_{m-1}^{(j)} F_m\left(\theta_m^{(i)} \mid \theta_{m-1}^{(j)}\right)}, & \text{if } m > 0 \end{cases}$$

and later normalised, after re-sampling each ith particle, $i = 1, \ldots, N$. Stopping rules for ABC-SMC schemes vary: here, we have opted for terminating the algorithm after a predetermined number $m = M$ of steps (a.k.a. epochs). The algorithm returns the weighted samples, representing the empirical posterior:

$$\left\{\theta_M^{(i)}, w_M^{(i)}\right\} \sim \pi_{ABC}^{h_M}(\theta \mid s_{obs}) \propto \int \mathbb{1}_{\Omega_{h_M}}(s) p(y \mid \theta)\pi(\theta)\, \mathrm{d}y.$$

Verification via Statistical Model Checking

Bayesian Statistical Model Checking

So far we have introduced inference techniques to learn models, such as ABC-SMC. We focus now on formally verifying properties on these models through statistical means, namely, via the means of Statistical Model Checking (SMC). The goal of SMC is to either calculate an estimate to the satisfaction probability $\Lambda_\phi(\theta) = P(\mathcal{M}_\theta \models \phi)$ for the model parametrised by θ, which we denote as $\hat{\Lambda}_\phi(\theta)$; alternatively, it may be used to set up a statistical hypothesis test for the given model and property. In this work, we are interested solely in estimating the satisfaction probability and refer the interested reader to [44] for a comprehensive review that covers SMC techniques based on both hypothesis testing or estimation. In this work, the statistical model checking problem can be stated as the estimation of a

Bernoulli random variable with unknown mean $\Lambda_\phi(\theta)$, which can be achieved by generating draws from such a Bernoulli random variable: akin to [2], the simulations that are generated within the ABC-SMC algorithm for the inference aspect of the work can be used to compute the estimate of $\Lambda_\phi(\theta)$.

Among the Bayesian approaches to SMC, we leverage here Bayesian interval estimation [12] to estimates the Bernoulli random variable $\mathcal{V}$ with unknown mean $\Lambda_\phi(\theta) = u$. The conditional probability density function associated with $\mathcal{V}$ is given by $p(v \mid u) = u^v(1-u)^{1-v}$, where $v = 1$ if $y \models \varphi$,where y is a simulation generated from the high-fidelity model, otherwise $v = 0$. A natural choice for the prior are Beta distributions $\pi(\cdot) = \pi(u, \alpha, \beta)$, as they are conjugate to Bernoulli ones. If we sample from a density $p(\cdot|u)$ where the unknown probability u is given by $\pi(\cdot)$, the posterior given observed Bernoulli r.v. $\mathcal{V} = \{v_1, \ldots, v_n\}$ is

$$\pi(u \mid \mathcal{V}) = \frac{p(\mathcal{V} \mid u)\pi(u)}{\int_0^1 p(\mathcal{V} \mid l)\pi(l)\,\mathrm{d}l}$$

and, as the v_i are iid, the likelihood factorises as $p(\mathcal{V} \mid u) = p(v_1, \ldots, v_n \mid u) = \Pi_{i=1}^n p(v_i \mid u)$. When sampling from a Bernoulli distribution with conjugate Beta prior, the integral in Equation (2) can be calculated analytically. If we define a coverage goal $c \in (\frac{1}{2}, 1)$, any interval (l_0, l_1) s.t.

$$\int_{l_0}^{l_1} \pi(u \mid v_1, \ldots, v_n)\,\mathrm{d}u = c, \tag{2}$$

is defined as the $100c$ percent Bayesian interval estimate for $u = \Lambda_\phi(\theta)$.

In the Bayesian statistical model checking algorithm (BSMC) (via Bayesian intervals), we define the half-size interval $\ell \in (0, \frac{1}{2})$ of the interval estimate for $\Lambda_\phi(\theta)$, a coverage parameter $c \in (\frac{1}{2}, 1)$ and coefficients α, β for the Beta prior, $\pi(\cdot)$. We generate traces $y_i \sim p(y|\theta)$ from model $\mathcal{M}_\theta$ and verify whether the trace satisfies a property of interest $y_i \models \varphi$, where φ is the path formula associated with ϕ. At each iteration i of the algorithm, we compute the posterior mean $\hat{\Lambda}_\phi(\theta)$ analytically, which is the Bayes estimator for $\Lambda_\phi(\theta)$, and then compute the posterior probability of the interval $(l_0, l_1) = (\hat{\Lambda}_\phi(\theta) - \ell, \hat{\Lambda}_\phi(\theta) + \ell)$, given by $\Upsilon = \int_{l_0}^{l_1} \pi(u \mid v_1, \ldots, v_i)\,\mathrm{d}u$. If $\Upsilon \geq c$, then the algorithm terminates and returns $\hat{\Lambda}_\phi(\theta)$ with corresponding upper and lower limits, $(l_0, l_1) = (\hat{\Lambda}_\phi(\theta) - \ell, \hat{\Lambda}_\phi(\theta) + \ell) = (\hat{\Lambda}_\phi^L(\theta), \hat{\Lambda}_\phi^U(\theta))$. If the posterior mean estimate, $\Lambda_\phi(\theta)$, converges to the extreme points of the interval $(0, 1)$, the algorithm returns $(l_0, l_1) = (0, 2\ell)$ if $l_0 < 0$ or $(l_0, l_1) = (1 - 2\ell, 1)$ if $l_1 > 1$.
As it will become clear shortly, this BSMC algorithm is used for parameter classification (next section), as well as to relate high- and low-fidelity models.

Approximate Parameter Classification via SMC

The aim of parameter classification is to partition the parameter space Θ according to the satisfaction of the CSL property ϕ. Parameter classification results in the feasible set of parameters $\Theta_\phi = \{\theta \in \Theta : \mathcal{M}_\theta \models \phi\} = \{\theta \in \Theta : \Lambda_\phi(\theta) \Cap \zeta\}$ which would ideally be computed via parameter synthesis techniques such as[23]. However, by classifying the outputs obtained from statistical model checking via

supervised machine learning, as is done in [2], one can rapidly perform a parameter classification over the entire parameter space for a general variety of models.

In this work we propose to employ again SMC and supervised machine learning for this task. Rather than perform SMC for every sampled point in the likelihood-free inference scheme as is done in [2], we consider carefully when to perform SMC to generate these parameter classification regions. While performing SMC for every sampled scheme allowed for a detailed parameter classification, by reducing the number of times we perform SMC, we speed up the simultaneous inference and verification scheme of [2]. In practice, we use the estimated lower $\hat{\Lambda}^L_\phi(\theta)$ and upper bounds $\hat{\Lambda}^U_\phi(\theta)$, such that $\Lambda_\phi(\theta) \in \left[\hat{\Lambda}^L_\phi(\theta), \hat{\Lambda}^U_\phi(\theta)\right]$, to partition the parameter space as:

- $\Theta_\phi = \{\theta \in \Theta : \hat{\Lambda}^L_\phi(\theta) > \zeta\}$,
- $\Theta_{\neg\phi} = \{\theta \in \Theta : \hat{\Lambda}^U_\phi(\theta) < \zeta\}$,
- $\Theta_{\mathcal{U}} = \Theta \backslash (\Theta_\phi \cup \Theta_{\neg\phi})$.

Notice that these sets are a function of $\theta \in \Theta$. Since in the inference procedure we generate a finite number of parameter samples, which are biased towards the sought posterior distribution, there might be areas of the parameter space Θ that are insufficiently covered. We thus resort to supervised learning techniques to classify parameter regions: we utilise support vector machines (SVMs) [45, 46] as a classification technique. Support vector machines are a simple yet effective supervised learning technique as they are effective in high-dimensional spaces and only require a subset of training points in the decision function. Given a set of training examples, each classified by the user, the SVM model assigns unseen data to one category or the other, making it a non-probabilistic binary linear classifier. The SVM constructs a hyper-plane or set of hyper-planes in a higher dimensional space that is trained to maximise the largest distance between the two (or more) classes - the larger this margin is, the lower the generalization error of the classifier. Given the relatively low number of data points required to train SVMs and the high number of simulations required to perform SMC that generate the data points we wish to classify, we make use of SVMs as our classification technique - alternative classifiers such as Stochastic Gradient Descent [47] make for better alternatives if we work with much larger datasets. We train the SVM classifier on the data produced from the overall algorithm, namely on the set $\{\theta^{(r)}, \hat{\Lambda}_\phi(\theta^{(r)}), \hat{\Lambda}^L_\phi(\theta^{(r)}), \hat{\Lambda}^U_\phi(\theta^{(r)})\}$ where $r = 1, \ldots, \bar{N}$ and $\bar{N}$ denotes the total number of samples generated, whether accepted or not.

The SVM, which is trained on the generated data, then provides a non-linear classification function, $\xi_\phi(\theta)$, where $\xi_\phi(\theta) = 1$ if $\theta \in \Theta_\phi$, $\xi_\phi(\theta) = -1$ if $\theta \in \Theta_{\neg\phi}$ and $\xi_\phi(\theta) = 0$ if $\theta \in \Theta_{\mathcal{U}}$. With this classification, we can then generalise over the entire parameters space $\theta \in \Theta$.

The Bayesian Verification Framework

Integration of Formal Verification and Inference

There are three aspects to the Bayesian Verification framework, first introduced in [2, 5] for CRNs: Bayesian inference, parameter classification (done via synthesis techniques [2] or approximately [5], as described above), and a probability (or credibility) calculation.

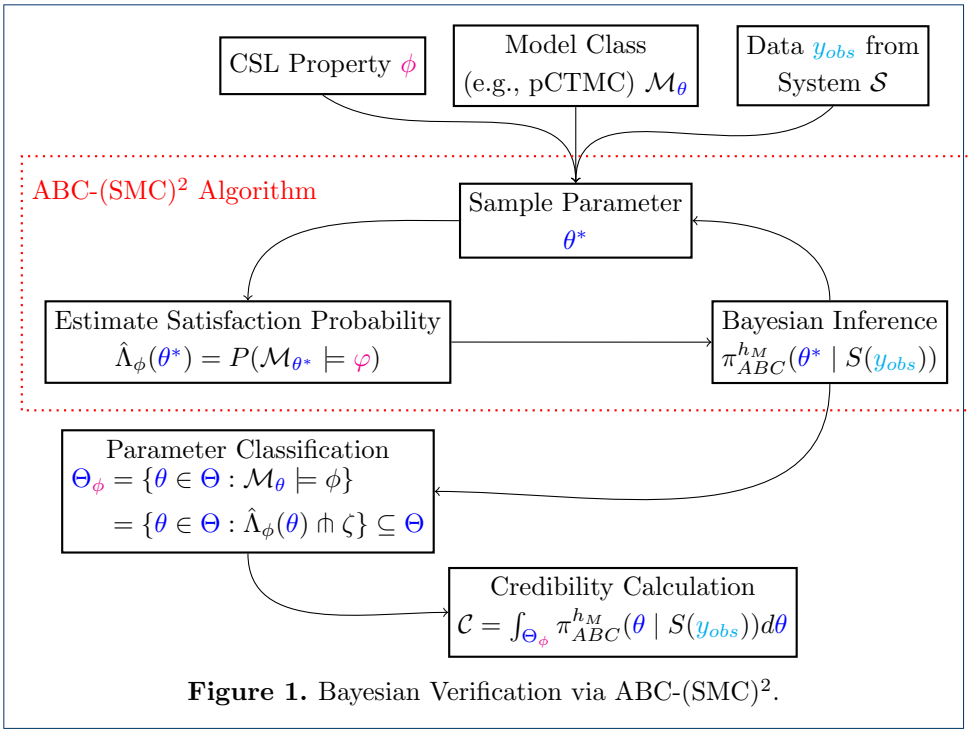


Figure 1. Bayesian Verification via ABC-(SMC)^2.

Regardless of how we achieve our parameter classification Θ_ϕ, we integrate the inferred posterior, $\pi_{ABC}(\theta \mid y_{obs})$ (or conditioned on $s_{obs} = S(y_{obs})$), over Θ_ϕ, the feasible set of parameters, to compute a probability estimate corresponding to the satisfaction of a CSL specification formula ϕ by the data-generating system $\mathcal{S}$, which we denote as the credibility $\mathcal{C}$:

$$\mathcal{C} = P(\mathcal{S} \models \phi \mid s_{obs}) = \int_{\Theta_\phi} \pi(\theta \mid s_{obs}) \, \mathrm{d}\theta. \tag{3}$$

The ABC-(SMC)^2 Algorithm

[2] introduced ABC-(SMC)^2, a technique that addresses the scalability limitations of previous Bayesian Verification frameworks [5]. Specifically, the work incorporates SMC (statistical model checking) within the Bayesian inference framework and approximate the classification of parameters: this allows the Bayesian Verification framework to be applied to a wider variety of models. The idea behind this framework, called Approximate Bayesian Computation - Sequential Monte Carlo with Statistical Model Checking: ABC-(SMC)^2 is presented in Fig. 1.

In the ABC-SMC scheme (Algorithm 5), a total of B_m simulations are performed for each sampled parameter θ^{**}, whether the sample is retained or not towards the approximate posterior $\pi^{h_M}_{ABC}(\theta \mid s_{obs})$: this leads to a considerable amount of wasted computational effort. In [2] we proposed instead to use these simulations to statistically model check each of the sampled parametrised models (whether or not the sample is retained towards the computation of the posterior) by means of the generated simulations. With the use of statistical and supervised machine learning approaches, one can then classify parameter regions using the information gathered with this simultaneous parameter inference and statistical model checking regime.

Recall that the output of the ABC-SMC algorithm is a set of samples $\theta_M^{(i)}$ with their corresponding weights $w_M^{(i)}$, which satisfy the following:

$$\{\theta_M^{(i)}, w_M^{(i)}\} \sim \pi_{ABC}^{h_M}(\theta \mid s_{obs}) \propto \int K_{h_M}\left(\|s - s_{obs}\|\right) p(y \mid \theta)\pi(\theta)dy, \tag{4}$$

where $i = 1, \ldots, N$ is the number of Monte Carlo samples required to approximate the posterior. For each parameter θ^{**}, simulation data is generated from the model $y_b \sim p(y \mid \theta^{**})$ to calculate $s^b = S(y_b)$, where $b = 1, \ldots, B_m$.

Extending the original ABC-SMC algorithm to the ABC-$(\text{SMC})^2$ algorithm, [2] set the number of simulations for each sampled parameter, B_m, to be the minimum number of samples estimated in the sequential Massart algorithm [48]: this is a frequentist approach which, as an alternative to the ealrier Bayesian algorithm, to calculate an estimated probability $\hat{\Lambda}_\phi(\theta^{**})$ (as defined above) with given accuracy and confidence. As a result, parameters θ^{**} are selected a total of $\bar{N}$ times, whether or not these samples are accepted at any generation m. For these sampled parameters, $\theta^{(r)}$, $r = 1, \ldots, \bar{N}$, corresponding mean probabilities, $\hat{\Lambda}_\phi\left(\theta^{(r)}\right)$, and $(1-\delta)$-uncertainty bounds are estimated: $\left\{\theta^{(r)}, \hat{\Lambda}_\phi\left(\theta^{(r)}\right), \hat{\Lambda}_\phi^L\left(\theta^{(r)}\right), \hat{\Lambda}_\phi^U\left(\theta^{(r)}\right)\right\}$, where $r = 1, \ldots, \bar{N}$. Here $\bar{N}$ depends on the acceptance rate of the sampled parameters $\theta^{(r)}$, where $\bar{N} \geq N \times M$, and where N is the number of Monte Carlo samples required and M is the total number of generations of the ABC-SMC scheme. From the complete algorithm, we obtain a set of weighted parameter vectors from the final generation M, $\{\theta_M^{(i)}, w_M^{(i)}\} \sim \pi_{ABC}^{h_M}(\theta \mid s_{obs})$ where $i = 1, \ldots, N$, as well as $\bar{N}$ sampled parameters and their corresponding estimated probabilities $\left\{\theta^{(r)}, \hat{\Lambda}_\phi\left(\theta^{(r)}\right), \hat{\Lambda}_\phi^L\left(\theta^{(r)}\right), \hat{\Lambda}_\phi^U\left(\theta^{(r)}\right)\right\}$, where again $r = 1, \ldots, \bar{N}$.

We recall that by increasing the number of Monte Carlo samples in the ABC-SMC scheme, we increase the accuracy of the posterior distribution estimate, and that this is indeed the case in the new framework too. However, as we increase the number of parameter samples to accept, we must also remember that we increase the number of times we perform statistical model checking for each sampled parameter. While this leads to a more accurate posterior distribution estimate and allows for a rigorous model verification across the parameter space, it also results in higher computational costs. When it comes to implementing the ABC-$(\text{SMC})^2$ framework on more complex models with a higher number of parameters, the algorithm has the same limitations as the ABC-SMC algorithm. While the ABC-SMC aspect of the framework provides one with a global parameter sensitivity analysis [10], as we increase the model complexity the probability of accepting the perturbed or sampled parameter θ^{**} decreases and we thus need to explore alternative and more efficient sampling schemes.

We present the ABC-$(\text{SMC})^2$ scheme introduced in [2] in Algorithm 1. This algorithm takes as inputs a property of interest, ϕ, a prior probability distribution $\pi(\theta)$ and additional parameters as inputs for the SMC algorithm. The estimated probabilities $\left\{\theta^{(r)}, \hat{\Lambda}_\phi(\theta)^{(r)}, \hat{\Lambda}_\phi^L(\theta)^{(r)}, \hat{\Lambda}_\phi^U(\theta)^{(r)}\right\}$, $r = 1, \ldots, \bar{N}$, will be utilised for approximate parameter classification, which is discussed in the next section.

Theoretical results on ABC-$(\text{SMC})^2$

We present a rationale for integrating SMC and Bayesian inference, and argue that the ABC-$(\text{SMC})^2$ algorithm converges asymptotically to the desired quantity $\mathcal{C} =$

Algorithm 1 ABC-$(\text{SMC})^2$

Input:

- CSL specification ϕ
- Prior $\pi(\theta)$ and data-generating likelihood function $p(y_{obs} \mid \theta)$
- A neighbourhood that evolves with distance thresholds $\Omega_{h_m} = \{y \in \mathcal{Y} \mid d(y, y_{obs}) \leq h_m\}$ with corresponding kernel $K_{h_m}(y)$ $(= \mathbb{1}_{\Omega_{h_m}}(y)$ in this paper)
- $N > 0$, number of particles used to estimate posterior distributions
- Sequence of perturbation kernels $F_m(\theta \mid \theta^*)$, $m = 0, 1 \ldots, M$
- A quantile $\upsilon \in [0, 1]$ to control the rate of decrease of h_m
- Summary statistic function $s = S(y)$
- $B_m > 0$, number of simulations per sampled particle. For stochastic systems $B_m > 1$
- Parameters for statistical model checking

Output:

- Set of weighted parameter vectors $\left\{\theta_M^{(i)}, w_M^{(i)}\right\}_{i=1}^N$ drawn from $\pi_{ABC}(\theta \mid s_{obs}) \propto \int K_{h_M}(s) p(y \mid \theta)\pi(\theta)\, \mathrm{d}s$
- $\{\theta^{(r)}, \hat{\Lambda}_\phi(\theta^{(r)}), \hat{\Lambda}_\phi^L(\theta^{(r)}), \hat{\Lambda}_\phi^U(\theta^{(r)})\}_{r=0}^{\bar{N}}$, where $\bar{N}$ is the total number of sampled parameters.

1: Set $r = 0$
2: **for** $m = 0, \ldots, M$: **do**
3: **for** $i = 0, \ldots, N$: **do**
4: **if** $m = 0$ **then**
5: Generate $\theta^{**} \sim \pi(\theta)$
6: **else**
7: Generate θ^* from the previous population $\{\theta_{m-1}^{(i)}\}$ with weights $\{w_{m-1}^{(i)}\}$ and perturb the particle to obtain $\theta^{**} \sim F_m(\theta \mid \theta^*)$
8: **end if**
9: **if** $\pi(\theta^{**}) = 0$ **then**
10: **goto** line 4
11: **end if**
12: Calculate $\left(\left\{\hat{\Lambda}_\phi\left(\theta^{**}\right), [\hat{\Lambda}_\phi^L\left(\theta^{**}\right), \hat{\Lambda}_\phi^U\left(\theta^{**}\right)\right\}, B_m, \sum_{b=1}^{B_m} K_{h_m}(\|s^b - s_{obs}\|), \bar{d}\right)$ from SMC algorithm of choice (frequentist, or Bayesian)
13: Calculate $b_m(\theta^{**}) = \sum_{b=1}^{B_m} K_{h_m}(s_b)$
14: Set $\left(\theta^{(r)}, \hat{\Lambda}_\phi\left(\theta^{(r)}\right), \hat{\Lambda}_\phi^L\left(\theta^{(r)}\right), \hat{\Lambda}_\phi^U\left(\theta^{(r)}\right)\right) = \left(\theta^{**}, \hat{\Lambda}_\phi(\theta^{**}), \hat{\Lambda}_\phi^L(\theta^{**}), \hat{\Lambda}_\phi^U(\theta^{**})\right)$
15: $r \leftarrow r + 1$
16: **if** $b_m(\theta^{**}) = 0$ **then**
17: **goto** line 4
18: **end if**
19: Set $\theta_m^{(i)} = \theta^{**}$, $\bar{d}_m^{(i)} = \frac{1}{B_m} \sum_{b=1}^{B_m} \|s_b - s_{obs}\|$ and calculate

20:

$$w_m^{(i)} = \begin{cases} b_m\left(\theta_m^{(i)}\right), & \text{if } t = 0 \\ \dfrac{\pi\left(\theta_m^{(i)}\right) b_m\left(\theta_m^{(i)}\right)}{\sum_{j=1}^N w_{m-1}^{(j)} F_m\left(\theta_m^{(i)} \mid \theta_{m-1}^{(j)}\right)}, & \text{if } t > 0 \end{cases}$$

21: **end for**
22: Generate $\{\theta_m^{(i)}, w_m^{(i)}\}_{i=1}^N$ using Algorithm 4 (ABC-IS) with importance distribution $f_m(\theta)$ and neighbourhood Ω_{h_m}
23: Normalise weights: $w_m^{(i)} \leftarrow w_m^{(i)} / \left(\sum_{i=1}^N w_m^{(i)}\right)$
24: Set $h_{m+1} = (\upsilon/N) \sum_{i=1}^N \bar{d}_m^{(i)}$
25: **end for**
26: **return** $\left\{\left(\theta_M^{(i)}, w_M^{(i)}\right)\right\}_{i=1}^N \left\{\theta^{(r)}, \hat{\Lambda}_\phi\left(\theta^{(r)}\right), \hat{\Lambda}_\phi^L\left(\theta^{(r)}\right), \hat{\Lambda}_\phi^U\left(\theta^{(r)}\right)\right\}_{r=1}^{\bar{N}}$

$P(\mathcal{S} \models \phi \mid s_{obs})$, which is core to the overall Bayesian Verification approach. Recall that N denotes the total number of accepted Monte Carlo samples, n is the number of simulations required for SMC estimates, with absolute error ϵ and confidence parameter δ leading to $(1-\delta)$-confidence intervals (or Bayesian equivalents) and h_m is the distance threshold for an ABC posterior $\pi_{ABC}^{h_m}(\theta \mid s_{obs})$. We let the credibility calculation that we compute from the ABC-(SMC)^2 algorithm be denoted as $\mathcal{C}_{m,N,n}$.

Theorem 0.1 (ABC-(SMC)^2 Convergence) *$\mathcal{C}_{m,N,n}$ converges in probability to $\mathcal{C}$ as the quantities $h_m \to 0$ $(m \to \infty)$ $N \to \infty$ and $n \to \infty$, namely:*

$$\forall \epsilon_{tot} > 0, \quad P(|C_{m,N,n} - \mathcal{C}| > \epsilon_{tot}) \to 0. \tag{5}$$

Proof To prove convergence, we work from the credibility calculation in Equation (3), applying the ABC approximation that converges to the true posterior [9]; the approximation scheme via Monte Carlo integration that converges in probability to the true integral [49]; and finally the SMC approximation to the satisfaction probability, given by $\hat{\Lambda}_\phi(\theta)$, which converges in probability to the true value $\Lambda_\phi(\theta)$. Starting from the credibility calculation, we note that

$$\mathcal{C} = \int_{\Theta_\phi} \pi(\theta \mid s_{obs}) \, \mathrm{d}\theta = \int_{\Theta} \mathbb{1}_{\Theta_\phi}(\theta) \pi(\theta \mid s_{obs}) \, \mathrm{d}\theta = \mathbb{E}_\pi[\mathbb{1}_{\Theta_\phi}(\theta)],$$

where $\mathbb{1}_{\Theta_\phi}(\theta) = 1$ if $\theta \in \Theta_\phi$, and 0 otherwise. Given that we work with likelihood-free methods, namely ABC, we have that

$$\begin{aligned} \mathcal{C} &= \int_{\Theta} \mathbb{1}_{\Theta_\phi}(\theta) \pi(\theta \mid s_{obs}) \, \mathrm{d}\theta \\ &= \lim_{m \to \infty} \int_{\Theta} \mathbb{1}_{\Theta_\phi}(\theta) \pi_{ABC}^{h_m}(\theta \mid s_{obs}) \, \mathrm{d}\theta \quad \text{(ABC approximation)}, \end{aligned}$$

where we have used the ABC approximation to the true posterior, specifically $\pi(\theta \mid s_{obs}) = \lim_{m \to \infty} \pi_{ABC}^{h_m}(\theta \mid s_{obs}) = \lim_{h_m \to 0^+} \pi_{ABC}^{h_m}(\theta \mid s_{obs})$[9]. We then apply the Monte Carlo approximation, taking the iid samples to be samples from the ABC posterior, $\pi_{ABC}(\theta \mid s_{obs})$,

$$\begin{aligned} \mathcal{C} &= \lim_{m \to \infty} \int_{\Theta} \mathbb{1}_{\Theta_\phi}(\theta) \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} \delta_{\theta^{(i)}}(\theta) \text{ (Monte Carlo approximation)} \\ &= \lim_{m \to \infty} \left[\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\Theta_\phi}(\theta^{(i)}) \right], \end{aligned}$$

where $\theta^{(i)}$ are iid samples from the ABC posterior, $\theta^{(i)} \sim \pi_{ABC}^{h_m}(\theta \mid s_{obs})$, $\delta_{\theta^{(i)}}(\theta)$ is the Dirac delta measure, where $\delta_x(A) = \mathbb{1}_A(x) = 1$ if $x \in A$, and $\delta_x(A) = \mathbb{1}_A(x) = 0$ otherwise, and we denote the Monte Carlo approximation to the integral $\mathcal{I} = \int_\Theta f(d\theta)$ as $\mathcal{I}_n = (1/n) \sum_{i=1}^N f(\theta^{(i)})$, where it can be shown [49] that $\mathcal{I}_n \to \mathcal{I}$ in probability, $P(|\mathcal{I}_n - \mathcal{I}| > \epsilon_{MC}) \to 0$ $\forall \epsilon_{MC} > 0$ as $N \to \infty$. The difficulty now is determining whether the sampled parameters $\theta^{(i)}$ lie within the set of parameters that satisfy the property, $\theta^{(i)} \in \Theta_\phi$. We note that Θ_ϕ is defined as $\Theta_\phi = \{\theta \in \Theta :$

$\Lambda_\phi(\theta) \pitchfork \zeta\}$, where ζ is the probability bound contained in a formula ϕ which allows us to present the function $\mathbb{1}_{\Theta_\phi}(\theta)$ as

$$\mathbb{1}_{\Theta_\phi}(\theta) = \mathbb{1}_{\Lambda_\phi(\theta)\pitchfork\zeta}(\theta) = \begin{cases} 1, & \Lambda_\phi(\theta) \pitchfork \zeta, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

which results in

$$\mathcal{C} = \lim_{m\to\infty} \left[\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\Lambda_\phi(\theta)\pitchfork\zeta}(\theta^{(i)}) \right].$$

However, we statistically estimate the satisfaction probability $\Lambda_\phi(\theta)$. Let V be a Bernoulli random variable with corresponding mean parameter $\Lambda_\phi(\theta^{(i)})$, that is, $v \sim \text{Bern}(\Lambda_\phi(\theta^{(i)}))$. Then $\Lambda_\phi(\theta)$ can be estimated using n sample draws from this distribution, $\hat{\Lambda}_\phi(\theta^{(i)}) \approx (1/n)\sum_{i=1}^{n} v_i$. This estimate has been shown (for both frequentist and Bayesian SMC [12, 44, 48]) to converge in probability to the true mean $\Lambda_\phi(\theta)$, that is, $P(|\hat{\Lambda}_\phi(\theta) - \Lambda_\phi(\theta)| > \epsilon_{SMC}) \to 0$, $\forall \epsilon_{SMC} > 0$ as $n \to \infty$. Thus, taking a further limit $n \to \infty$, this leads us to

$$\mathcal{C} = \lim_{m\to\infty} \left[\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \left[\lim_{n\to\infty} \mathbb{1}_{\hat{\Lambda}_\phi(\theta)\pitchfork\zeta}(\theta^{(i)}) \right] \right] \approx \mathcal{C}_{m,N,n}, \tag{7}$$

where $\mathcal{C}_{m,N,n} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\hat{\Lambda}_\phi(\theta)\pitchfork\zeta}(\theta^{(i)})$ and $\theta^{(i)}$ are iid draws from the ABC posterior, $\theta^{(i)} \sim \pi_{ABC}^{h_m}(\theta \mid s_{obs})$. Thus by taking the limits $\lim_{m,N,n\to\infty} \mathcal{C}_{m,N,n} = \mathcal{C}$. □

Results

Multifidelity Bayesian Verification

Multifidelity Bayesian Inference

Multifidelity Bayesian methods [3, 4] can reduce the computational burden of ABC by utilising simulations that are generated more cheaply, allowing one to construct the ABC posterior $\pi_{ABC}^{h_M}(\theta \mid y_{obs})$ with lower computational costs. In this section we integrate multifidelity ABC methods and SMC techniques, which results in a new, multifidelity Bayesian verification framework, the MF-ABC-(SMC)^2 scheme. This novel framework utilises low-fidelity simulations to construct the posterior at a lower computational cost than [2], allowing for a faster and cheaper way to calculate the credibility, namely quantity $\mathcal{C}$ in Equation (3). Of course, this might come with an error introduced by employing models that, whilst faster to simulate, are less aligned with data, and thus prone to result in more rejected samples.

Multifidelity Approximate Bayesian Computation

Whilst ABC methods are popular due to the ease with which they can be utilised to provide likelihood-free inference, they rely on a large number of simulations and can lead to high computational costs, especially if they require a high number of parameter proposals and subsequent simulations. Multifidelity Approximate Bayesian Computation (MF-ABC) algorithms [3] make use of multifidelity methods [1] to reduce the computational cost related to ABC sampling.

In multifidelity methods, we consider both a high-fidelity (HF) model of interest and a corresponding low-fidelity (LF) model, which is a model that approximates the high-fidelity model but is significantly cheaper to generate simulations from. The high-fidelity model $p(y \mid \theta)$ or $\mathcal{M}_\theta$, is a detailed model that is computationally expensive to generate simulations $y \sim p(y \mid \theta)$ from ($y \in \mathcal{Y}$ is the output space), where we denote this high computational cost as $c(\theta)$. The low-fidelity model, denoted by $\tilde{p}(\tilde{y} \mid \theta)$ or $\tilde{\mathcal{M}}_\theta$, maps parameter samples $\theta \in \Theta$ from a common parameter space to a distribution on an output space $\tilde{\mathcal{Y}}$. This model comes at a lower computational cost, say $\tilde{c}(\theta)$. Notice that in the multifidelity framework, we work directly with the simulations y and not with summary statistic thereof (previously denoted as $s = S(y)$): working with fewer simulations in the ABC scheme (and multifidelity variants) allows one to explore the parameter space more efficiently [50]. In our CRN setup, the high-fidelity model is a pCTMC $\mathcal{M}_\theta$, whereas the low-fidelity model is a pSDE $\tilde{\mathcal{M}}_\theta$. We assume the signals y and $\tilde{y}$ can be compared amongst each other and against the observations y_{obs}. In general, we can define a low-fidelity equivalent of y_{obs}, denoted by $\tilde{y}_{obs}$, which we obtain by mapping y_{obs} to the output space $\tilde{\mathcal{Y}}$. However in this work, for simplicity we work with the case where $\tilde{y}_{obs} = y_{obs}$. Similarly to how a distance function $d(y, y_{obs})$ and a neighbourhood Ω_h is defined for ABC, one can define a low-fidelity distance function $\tilde{d}(\tilde{y}, \tilde{y}_{obs})$ and a neighbourhood $\Omega_{\tilde{h}} = \{\tilde{y} \in \tilde{\mathcal{Y}} \mid \tilde{d}(\tilde{y}, \tilde{y}_{obs})\}$ with corresponding kernel $\mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y})$ associated with this new output space for the low-fidelity model. So whilst the MF-ABC framework can be quite general, for this paper we set $\tilde{y}_{obs} = y_{obs}$ and we use simple indicator functions for both the high- and low-fidelity kernels, $\mathbb{1}_{\Omega_h}(y)$ and $\mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y})$ respectively, where the distance metric is chosen to be the same, that is, $\tilde{d}(\cdot, \cdot) = d(\cdot, \cdot)$. These low-fidelity models come with an associated cost $\tilde{c}(\theta)$ and it is assumed that, on average, it is cheaper to generate simulations from the low-fidelity model as opposed to the high-fidelity model, $\mathbb{E}[\tilde{c}(\theta)] \ll \mathbb{E}[c(\theta)]$, where the expectation acts over the model's probability space. With this lower cost, the low-fidelity models, parameters and their associated weights, $\tilde{w}(\theta) = \mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y})$, $\{\theta^{(i)}, \tilde{w}^{(i)}\}_{i=1}^N$ can be sampled quickly compared to the high-fidelity equivalent $\{\theta^{(i)}, w^{(i)}\}_{i=1}^N$. However, the ABC approximations are not identical, $\mathbb{1}_{\Omega_h}(y)p(y \mid \theta)\pi(\theta) \neq \mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y})\tilde{p}(\tilde{y} \mid \theta)\pi(\theta)$ and using the low-fidelity model incurs a further bias when implementing the ABC scheme. [3] develops an MF-ABC algorithm that utilises both the low- and high-fidelity models to estimate $\pi_{ABC}(\theta \mid y_{obs})$. Doing so reduces the reliance of the ABC algorithm on generating simulations on the high-fidelity model, which thus reduces the computational burden of the algorithm. The key idea to the MF-ABC algorithm is the introduction of a multifidelity weight [3], namely

$$w_{\mathrm{mf}}(\theta) = \mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y}) + \frac{\mathbb{I}(U < \eta(\tilde{y}))}{\eta(\tilde{y})} \left[\mathbb{1}_{\Omega_h}(y) - \mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y})\right], \tag{8}$$

where $U \sim \mathcal{U}(0,1)$ and $\eta(\tilde{y}) \in (0,1)$ is a continuation probability, defined by $\eta(\tilde{y}) = \eta_1 \mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y}) + \eta_2(1 - \mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y}))$, where $\eta_1 \in [0,1]$ and $\eta_2 \in [0,1]$, determined at the discretion of the user. This is presented in Algorithm 6 with further particulars to be found in [3]. The multifidelity weight defined in Equation (8) combines ideas in alternative ABC techniques that make use of low-fidelity simulations to speed up the ABC algorithm. Specifically, the MF-ABC algorithm combines ideas from early

rejection ABC [51, 52] and early decision ABC. In early rejection ABC, the low fidelity simulation $\tilde{y}$ is used to determine whether or not to proceed and simulate $y \sim p(y \mid \theta)$ from the high-fidelity model, meanwhile in early decision ABC, the low-fidelity simulations are used to determine whether or not the proposed parameter is to be accepted without simulating the high-fidelity model. Combining these ideas for early rejection and early decision ABC into the multifidelity weight $w_{\mathrm{mf}}(\theta)$ gives rise to four possible values:

$$w_{\mathrm{mf}}(\theta) = \begin{cases} 1 & \tilde{y} \in \Omega_{\tilde{h}} \wedge U \geq \eta_1 \text{ (early accept)} \\ 0 & \tilde{y} \notin \Omega_{\tilde{h}} \wedge U \geq \eta_2 \text{ (early reject)} \\ 1 & \tilde{y} \in \Omega_{\tilde{h}} \wedge y \in \Omega_h \wedge U < \eta_1 \text{ (checked true positives)} \\ 0 & \tilde{y} \notin \Omega_{\tilde{h}} \wedge y \notin \Omega_h \wedge U < \eta_2 \text{ (checked true negative)} \\ 1 - 1/\eta_1 & \tilde{y} \in \Omega_{\tilde{h}} \wedge y \notin \Omega_h \wedge U < \eta_1 \text{ (checked false positive)} \\ 0 + 1/\eta_2 & \tilde{y} \notin \Omega_{\tilde{h}} \wedge y \in \Omega_h \wedge U < \eta_2 \text{ (checked false negative).} \end{cases} \tag{9}$$

Due to the possible negative weight values, the pairs $\{\theta^{(i)}, w_{\mathrm{mf}}^{(i)}\}_{i=1}^N$ cannot be treated as direct weighted samples from the ABC posterior $\pi_{ABC}(\theta \mid y_{obs})$. Still, we obtain an estimator for the ABC approximation of the posterior expectation of an arbitrary function $g(\theta)$ [3], as

$$\mathbb{E}_{ABC}(g(\theta) \mid y_{obs}) = \int g(\theta)\pi_{ABC}(\theta \mid y_{obs})\, \mathrm{d}\theta \approx \frac{\sum_{i=1}^N w_{\mathrm{mf}}^{(i)} g(\theta^{(i)})/N}{\sum_{j=1}^N w_{\mathrm{mf}}^{(i)}/N}.$$

Multifidelity Approximate Bayesian Computation - Sequential Monte Carlo

The Multifidelity approach of [3] presented in the previous section improves the performance of the native ABC algorithm (Algorithm 3) by allowing sampled parameters to be weighted and evaluated with a combination of low- and high-fidelity models. This method specifically allows one to take advantage of low-fidelity models to evaluate weights instead of relying solely on high-fidelity models, which means that a fixed number of Monte Carlo samples N from the ABC posterior $\pi_{ABC}(\theta \mid y_{obs})$ can be obtained for a lower computational cost, and thus faster, compared to high-fidelity models. Whilst this extension does not decrease the total number of simulations required, it does increase the speed at which weights can be evaluated. Similar to how sequential Monte Carlo approaches [10, 43] were introduced to reduce the total number of simulations required for ABC, [4] introduces a sequential Monte Carlo extension to the MF-ABC approach, aptly named MF-ABC-SMC, resulting in a framework that reduces the total number of simulations required as well as their computational cost. The MF-ABC part speeds up the evaluation of the weights given sampled parameters, whereas the sequential Monte Carlo approach (similar to the ABC-SMC extension to the ABC scheme) is expected to reduce the total number of sampled parameters (and in turn, the total number of simulations). The two approaches are integrated by extending the ABC importance sampling approach presented in Algorithm 4 to the multifidelity framework, resulting in a Multifidelity ABC importance sampler scheme (MF-ABC-IS). The continuation probabilities presented in the MF-ABC-IS algorithm, $\eta(\tilde{y})$, is now

an explicit function on both the low-fidelity simulation, $\tilde{y}$, and the parameter samples θ; $\eta = \eta(\theta, \tilde{y}) \in (0, 1]$. With the MF-ABC-IS algorithm, if we were to define both a sequence of decreasing thresholds $h_1 > \cdots > h_M$ that induce the neighbourhoods $\Omega_{h_1} \supseteq \cdots \supset \Omega_{h_M}$ for the high-fidelity model and a corresponding sequence of decreasing thresholds for the low-fidelity model, $\tilde{h}_1 > \cdots > \tilde{h}_M$ that induce the neighbourhoods $\Omega_{\tilde{h}_1} \supseteq \cdots \supseteq \Omega_{\tilde{h}_m}$, using an appropriate perturbation kernel $F_m(\theta^* \mid \theta)$, then the sequential Monte Carlo algorithm could proceed similarly to the ABC-SMC introduced previously.

Recall that the newly generated weights might be negative and cannot be sampled from directly: [4] proposes using non-negative weights $\hat{w}_m^{(i)} = |w_m^{(i)}|$, so that at each generation m, two weights for the same sampled parameters are produced at once, $\{\theta_m^{(i)}, \hat{w}_m^{(i)}\}$ and $\{\theta_m^{(i)}, w_m^{(i)}\}$. The weighted samples $\{\hat{w}_m^{(i)}, \theta_m^{(i)}\}$ are drawn from an alternative target distribution, $\varpi_m(\theta) \propto \pi_{ABC}^{h_m}(\theta \mid y_{obs}) + \delta_m(\theta)$, where the additional term $\delta_m(\theta)$ is the difference between this alternative target distribution and the ABC posterior - see [4] for details. Although this new importance distribution now targets $\varpi_m(\theta)$ at each generation m, instead of the desired $\pi_{ABC}^{h_m}(\theta \mid y_{obs})$, the MF-ABC-SMC scheme still produces an estimate of an arbitrary function $g : \Theta \to \mathbb{R}$ with the retained weights $w_m^{(i)}$, $\mathbb{E}_{ABC}^{h_m}(g(\theta) \mid y_{obs})$, which is a consistent Monte Carlo estimate under the ABC posterior at the m-th epoch [4].

Multifidelity Bayesian Statistical Model Checking

We propose to extend the MF-ABC-SMC to incorporate Statistical Model Checking. We work with Bayesian SMC, which theoretically has been shown to achieve the same statistical guarantees with fewer simulations - see the example in [11] and Figure 5. Whilst we can perform SMC separately for the high- and low-fidelity models, if we perform SMC simultaneously for the multifidelity model, we are able to seamlessly quantify the assessment via [53] of the low-fidelity model as an approximate abstraction of the high-fidelity model, as well as estimate parameter classification regions for Bayesian verification, akin to [2].

Let us recall the MF-ABC-SMC scheme, where the high-fidelity model is denoted as $\mathcal{M}_\theta$ with respective simulations y generated via $y \sim p(y \mid \theta)$ and we denote the low-fidelity model as $\tilde{\mathcal{M}}_\theta$ with respective simulations generated by $\tilde{y} \sim \tilde{p}(\tilde{y} \mid \theta)$. Let us denote the specification that we wish to verify on the high-fidelity model as ϕ (with respective threshold probability ζ), which we assume to be checkable also on the low-fidelity model. Both models are parametrised by the same set of parameters θ, but might lead to different satisfaction probabilities $\Lambda_\phi(\theta)$: we denote the high-fidelity satisfaction probability estimate as $\hat{\Lambda}_\phi(\theta)$, and the corresponding low-fidelity satisfaction probability estimate as $\tilde{\Lambda}_\phi(\theta)$.

The MF-BSMC method follows closely to the BSMC algorithm, where we assume in this case that we are interested in the same coverage (c), half-interval width size (ℓ) and prior Beta distribution parameters (α, β) for both models. By storing the information generated from both the high- and low-fidelity models, we can further assess the approximation quality of the low-fidelity model compared to the high-fidelity model. We quantify how well this low-fidelity model acts as a proxy to the high-fidelity model with the use of a metric, namely $\hat{\gamma}_\epsilon$, which is based on [53] and further discussed in detail in the Methods section.This multifidelity

approach to Bayesian SMC results in Algorithm 10, which performs BSMC on both models, returning the set $\{\hat{\Lambda}_\phi(\theta), \hat{\Lambda}^L_\phi(\theta), \hat{\Lambda}^U_\phi(\theta)\}$ for the high-fidelity model, and $\{\tilde{\Lambda}_\phi(\theta), \tilde{\Lambda}^L_\phi(\theta), \tilde{\Lambda}_\phi(\theta)^U\}$ for the low-fidelity model, and finally $\hat{\gamma}_\varepsilon$. We note here that the number of simulations required to estimate the high/low-fidelity satisfaction probability estimates might differ (n vs $\tilde{n}$). As a result, we set $n_\gamma = \min\{n, \tilde{n}\}$ for the quantification $\hat{\gamma}_\varepsilon$.

By integrating the MF-BSMC algorithm with the MF-ABC-SMC scheme, we obtain the multifidelity weighted samples $\{\theta^{(i)}_M, w^{(i)}_M\}^N_{i=1}$ and associated satisfaction probabilities, $\hat{\Lambda}_\phi(\theta)$ and $\tilde{\Lambda}_\phi(\theta)$. Recalling the definition of the credibility calculation in Equation (3), $\mathcal{C} = P(\mathcal{S} \models \phi \mid y_{obs}) = \mathbb{E}_\pi\left[\mathbb{1}_{\Theta_\phi}(\theta)\right]$, and noting the fact that the set Θ_ϕ is defined as $\Theta_\phi = \{\theta \in \Theta : \Lambda_\phi(\theta) \Cap \zeta\}$, we can set $\mathbb{1}_{\Theta_\phi}(\theta) = \mathbb{1}_{\Lambda_\phi(\theta)\Cap\zeta}(\theta)$, and using the multifidelity weights we can get an approximation of the credibility calculation of the form

$$\mathcal{C}_{\text{mf}} = \mathbb{E}^{h_M}_{ABC}(\mathbb{1}_{\Lambda_\phi(\theta)\Cap\zeta}(\theta) \mid y_{obs}) \approx \frac{\sum^N_{i=1} w^{(i)}_M \mathbb{1}_{\Lambda_\phi(\theta)\Cap\zeta}(\theta^{(i)}_M)}{\sum^N_{j=1} w^{(j)}_M}. \tag{10}$$

Recall that the satisfaction probabilities are statistically estimated with guarantees via SMC, namely by estimating the interval $\hat{\Lambda}_\phi(\theta) \in \left[\hat{\Lambda}^L_\phi(\theta), \hat{\Lambda}^U_\phi(\theta)\right]$. The lower and upper limits will be chosen to make a decision on whether $\theta \in \Theta_\phi$, depending on the sign of the threshold decision in $\Lambda_\phi(\theta) \Cap \zeta$. To improve the scalability of the Bayesian verification framework, we set the number of simulations associated with each sampled parameter to $B_m = 1$ according to [50] and thus no summary statistics are required. In conclusion, rather than performing rigorous SMC techniques for every sampled parameter, as in [2], we explore when to perform SMC and by which model at different stages of the algorithm, which results in a credibility calculation $\mathcal{C}_{\text{mf}}$ that is obtained with a fewer number of simulations that are of a lower (or at most equal) computational cost than the original ABC-(SMC)^2 scheme introduced in [2] and discussed above.

End Result: The MF-ABC-(SMC)^2 Algorithm

The overall MF-ABC-(SMC)^2 scheme is presented in Algorithm 2, and results from integrating MF-ABC-SMC with MF-BSMC. In this scheme we can decide where to input the MF-BSMC algorithm (Algorithm 10) into the MF-ABC-SMC algorithm (Algorithm 8). Arguably the optimal choice would be to perform SMC on the high-fidelity model when we have inferred the posterior probability distribution after M thresholds, $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$, as we are interested in the credibility calculation $\mathcal{C}_{\text{mf}} = P(\mathcal{S} \models \phi \mid y_{obs})$ as per (10). However, in the Case Studies we will explore alternative options of when to perform BSMC, such as on parameters sampled from the prior $\pi(\theta)$ rather than the posterior distribution, and on the low-fidelity model rather than the high-fidelity one.

Case Studies

Experimental Setup

All experiments have been run on an Intel(R) Xeon(R) CPU E5-1660 v3 @ 3.00GHz, 16 cores with 16GB memory. Both ABC-(SMC)^2 and MF-ABC-(SMC)^2 are coded in C++, while Python is used for the SVM classifier.

Algorithm 2 Multifidelity ABC-(SMC)^2 (MF-ABC-(SMC)^2)

Input:

- Observed data y_{obs}
- Prior Beta distribution $\pi(\theta)$ with parameters α, β
- High-fidelity model $p(y \mid \theta)$ and low-fidelity model $\tilde{p}(\tilde{y} \mid \theta)$
- Initial importance distribution $\hat{r}_0(\theta)$ (commonly set to $\hat{r}_0(\theta) = \pi(\theta)$)
- Sequence of high fidelity neighbourhoods $\Omega_{h_m} = \{y \in \mathcal{Y} \mid d(y, y_{obs}) \leq h_m\}$ and low-fidelity neighbourhoods $\Omega_{\tilde{h}_m} = \{\tilde{y} \in \tilde{\mathcal{Y}} \mid \tilde{d}(\tilde{y}, \tilde{y}_{obs}) \leq \tilde{h}_m\}$ with respective kernels $\mathbb{1}_{\Omega_{h_m}}(y)$ and $\mathbb{1}_{\Omega_{\tilde{h}_m}}(\tilde{y})$
- Initial continuation probability function $\eta_0 = \eta_0(\theta, \tilde{y}) = 1$
- Lower bounds on continuation probabilities, $\eta_1^L, \eta_2^L \in (0, 1)$
- Sequence of perturbation kernels $F_m(\theta|\theta^*)$
- Monte Carlo sample size $N > 0$
- Property defined on both high- and low-fidelity model ϕ
- Half-interval width size $\ell \in (0, \frac{1}{2})$, interval coverage parameter $c \in (\frac{1}{2}, 1)$
- Parameter ε to obtain $(1-\varepsilon)$-level approximate abstraction $\hat{\gamma}_\varepsilon$

Output:

- $\{\theta^{(i)}, w^{(i)}, \}_{i=1}^N$, accepted parameters with corresponding weights

1: **for** $m = 0, \ldots, M$: **do**
2: Produce $\{\theta_m^{(i)}, w_m^{(i)}\}_{i=1}^N$ and store information required to estimate optimal continuation probabilities from MF-ABC-IS algorithm (Algorithm 7) with inputs $\left(y_{obs}, \tilde{y}_{obs}, \Omega_{h_m}, \Omega_{\tilde{h}_m}, \pi(\theta), p(y \mid \theta), \tilde{p}(\tilde{y} \mid \theta), \eta_m(\theta, \tilde{y})\right)$ setting $\hat{q} = \hat{r}_m(\theta)$
3: Set $\hat{w}_m^{(i)} = |w_m^{(i)}|$ for $i = 1, \ldots, N$
4: Define $\hat{r}_{m+1}$ proportional to r_{m+1} (see [4])
5: Calculate (η_1^*, η_2^*) with lower bounds (η_1^L, η_2^L) using importance distribution $r_{t+1}(\theta)$, ABC thresholds h_{m+1} and $\tilde{h}_{m+1}$ with values stored from MF-ABC-IS algorithm at step 2 (see [4])
6: Set $\eta_{m+1}(\theta, \tilde{y}) = \eta_1^* \mathbb{1}_{\Omega_{\tilde{h}_m}}(\tilde{y}) + \eta_2^*(1 - \mathbb{1}_{\Omega_{\tilde{h}_m}}(\tilde{y}))$
7: **if** $(m = 0) \wedge$ {Perform MF-BSMC for $\pi(\theta)$ } **then**
8: Calculate $\{\hat{\Lambda}_\phi(\theta_0^{(i)}), \tilde{\Lambda}_\phi(\theta_0^{(i)}), \hat{\gamma}_\varepsilon^{(i)}\}_{i=1}^N$ from MF-BSMC algorithm (Algorithm 10)
9: **end if**
10: **end for**
11: **if** {Perform MF-BSMC for $\pi_{ABC}^{h_M}(\theta \mid y_{obs})$ } **then**
12: Calculate $\{\hat{\Lambda}_\phi(\theta_M^{(i)}), \tilde{\Lambda}_\phi(\theta_M^{(i)}), \hat{\gamma}_\varepsilon^{(i)}\}_{i=1}^N$ from MF-BSMC algorithm (Algorithm 10)
13: **end if**
14: **return** $\{\theta_M^{(i)}, w_M^{(i)}\}_{i=1}^N$

Application of the ABC-(SMC)^2 Algorithm

SIR System and Parametrised Stochastic Model

The SIR system describes the dynamics of three epidemic types, a susceptible group (S), an infected group (I), and a recovered group of individuals (R) - here we let S, I and R evolve via the two rules

$$S + I \xrightarrow{k_i} 2I, \quad I \xrightarrow{k_r} R.$$

A pCTMC model for the SIR system is governed by the rate parameters $\theta = (k_i, k_r)$, and each state of the pCTMC describes the combination of the number of each type (S, I, R) (equating to molecule/species counts in CRNs), where the state vector is given by $X(t) = [S(t), I(t), R(t)]^\top$, which denotes the number of molecules of each species at a given time t. For the complete details on the case study, please see the Methods.The initial state of the system/pCTMC is given by $x_0 = X(0) = [S_0, I_0, R_0]^\top = [95, 5, 0]^\top$. The property we are interested in verifying is defined as the CSL formula $\phi = P_{>0.1}((I > 0)U^{[100,150]}(I = 0))$, i.e. whether, with a probability greater than 0.1, the infection dies out within a time interval between $t = 100$ and $t = 150$ seconds. We confine our parameters to the set $\Theta = [k_i^\perp, k_i^\top] \times [k_r^\perp, k_r^\top] = [5 \times 10^{-5}, 0.003] \times [0.005, 0.2]$, generating observation data from the SIR model with three different parameter choices,θ_ϕ, $\theta_{\neg\phi}$ and $\theta_{\mathcal{U}}$, corresponding to the CTMCs $\mathcal{M}_{\theta_\phi}$, $\mathcal{M}_{\theta_{\neg\phi}}$ and $\mathcal{M}_{\theta_{\mathcal{U}}}$. These models will correspond to three "true" underlying stochastic systems $\mathcal{S}$, with associated observation data. For each instance, we work with observed data y_{obs} that is sampled at a finite

number of time steps. The observed data consists of only 5 simulated traces ($n_{sims} = 5$), observed at 10 time points ($n_{obs} = 10$). The summary statistics $S(y_{obs}) = s_{obs}$ corresponds to the average of the five traces and is used for the inference aspect of the algorithm. It is worth emphasising that with so few observation traces, black-box SMC (directly based on observation traces, not on model-generated simulations) would be hopeless.

The ABC-(SMC)^2 algorithm (Algorithm 1) outputs samples from the approximated posterior and their corresponding weights, $\left\{\theta_M^{(i)}, w_M^{(i)}\right\} \sim \pi_{ABC}^{h_M}(\theta \mid s_{obs})$ where $i = 1, \ldots, N$. We choose the number of Monte Carlo samples to be $N = 500$; the number of sequential steps (epochs) to be $M = 20$; the kernel function $K_h(d(y, y_{obs})) = \mathbb{1}_{\Omega_h}(y)$ to be a simple indicator function, i.e. $K_h(d(y, y_{obs})) = 1$ if $d(y, y_{obs}) = \|y - y_{obs}\| < h$, $K_h(d(y, y_{obs})) = 0$ otherwise; the rate at which the thresholds h_m decrease to be $\upsilon = 0.5$; and the summary statistic $s = S(y)$ is chosen to be the sample mean of the simulations and of the observations. We choose $\pi(\theta)$ to be a uniform prior over Θ, which we define as

$$\pi(\theta) = \begin{cases} \left(\Pi_{i=1}^{N_R}(\theta_i^{\top} - \theta_i^{\perp})\right)^{-1}, & \text{if } \theta \in \Theta = [\theta_1^{\perp}, \theta_2^{\top}] \times \cdots \times [\theta_{N_R}^{\perp}, \theta_{N_R}^{\top}] \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

For the remainder of this work, $\pi(\theta)$, the prior, is chosen to be a uniform prior over the parameter space, Θ, regardless of which model we work with. The perturbation kernel $F_m(\theta^{**} \mid \theta^*)$ is chosen to be a multivariate Normal distribution, so that $\theta^{**} \sim \mathcal{N}(\theta^*, 2\Sigma_{m-1})$, where the covariance is twice the second moment computed over the accepted weights and particles at step $m-1$, namely $\left\{\theta_{m-1}^{(i)}, w_{m-1}^{(i)}\right\}$, where $i = 1, \ldots, N$. For further details on alternative choices for threshold sequences, summary statistics and perturbation kernels, see [9, 54–57].

Parameter Classification: A Motivating Comparison

For the SMC component of the overall ABC-(SMC)^2 scheme, we run the Massart algorithm [48] with the following choice of parameters: $(\epsilon, \delta, \delta_c) = (0.01, 0.05, 0.001)$, which results in a maximum number of necessary simulations that equals $B_m \leq n_{\mathcal{O}} = \lceil \frac{1}{2\epsilon^2} \log \frac{2}{\delta} \rceil = 18445$. At the conclusion of the ABC-(SMC)^2 algorithm, we train the classifier over half of the sampled parameters (whether eventually accepted or rejected), with the corresponding estimated probabilities and test it on the other half, which results in the SVM classifier accuracy presented in Table 3 and further described below. We now briefly present the approximate parameter classification generated from the ABC-(SMC)^2 approach and compare that to the numerical approach of [22, 23], which we present in Figure 2.

Classification can be performed as a synthesis problem via PRISM: this technique dissects the parameter space into 14413 grid regions (Figure 2b), which results in calculating the satisfaction probability at 57652 points. Instead, to illustrate how we can generate an approximate parameter classification using SMC, we consider sampling 1000 points from the prior $\pi(\theta)$ (see Eq. 11). We run the Massart algorithm at each point to obtain an estimated probability with corresponding $(1 - \delta)$ confidence bounds, where $\delta = 0.05$. With these samples and probabilities, we classify parameter regions with an SVM, which results in Figure (2c), with corresponding

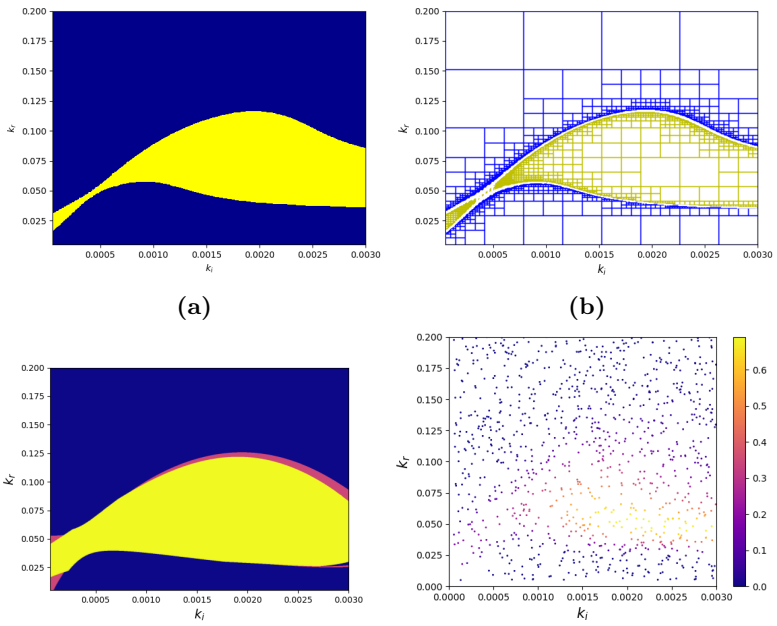


Figure 2. The set Θ_ϕ, is shown in yellow (lighter colour), meanwhile $\Theta_{\neg\phi}$, is shown in blue (darker colour) $\Theta_{\neg\phi}$. The undecided areas, $\Theta_{\mathcal{U}}$ (if any) are shown in magenta.(a) Parameter regions synthesised by GPU-Accelerated PRISM [23]. (b) Gridding scheme.(c) Parameter regions from SVM classification with 1000 samples from uniform distribution. (d) Estimated probabilities $\Lambda_\phi(\theta^*)$.

Table 1. Parameter classification - runtimes.

Parameter classification	Times [seconds]
PRISM-GPU	3096
SVM & SMC	1653.8

estimated probabilities in Figure (2d). The runtimes presented in Table 1 suggest that we obtain a good approximation of the parameter classification region in half the time of the GPU-accelerated PRISM tool, which could be further improved if we parallelised the computation [48]. These considerations have led us to embed the statistical parameter classification in the parameter inference algorithm.

Outcomes of the ABC-(SMC)^2 Algorithm

For the three models considered in the SIR case study, the inferred posterior mean $\hat{\theta}_M$, covariance Σ_M, total number of sampled parameters, $\bar{N}$, and resulting credibility calculation are given in Table 3, with corresponding runtimes in Table 4. Figures 3d, e and f plot the inferred posterior, showing the mean (denoted by $\times$) and two standard deviations from the mean (corresponding ellipse around the mean), as well as the true parameter value ($\triangle$). In Case θ_ϕ, we can assert, with a parameter classification based off a confidence of $(1-\delta) = 0.95$ and absolute-error $\epsilon = 0.01$, that the

Table 3. Bayesian Verification Results via ABC-(SMC)^2 for SIR case study.

Case	$\hat{\theta}_M$	Σ_M	Sampled Pars, $\bar{N}$	SVM Accuracy	Credibility Calculation
θ_ϕ	$\begin{bmatrix}0.00215\\0.07050\end{bmatrix}$	$\begin{bmatrix}1.46\cdot10^{-8} & 4.24\cdot10^{-7}\\4.24\cdot10^{-7} & 1.97\cdot10^{-5}\end{bmatrix}$	10952	99.6%	1
$\theta_{\neg\phi}$	$\begin{bmatrix}0.00072\\0.14519\end{bmatrix}$	$\begin{bmatrix}2.47\cdot10^{-8} & 3.41\cdot10^{-6}\\3.41\cdot10^{-6} & 9.22\cdot10^{-4}\end{bmatrix}$	10069	99.8%	0.0054
$\theta_{\mathcal{U}}$	$\begin{bmatrix}0.00193\\0.11337\end{bmatrix}$	$\begin{bmatrix}8.89\cdot10^{-8} & 5.86\cdot10^{-6}\\5.86\cdot10^{-6} & 4.21\cdot10^{-4}\end{bmatrix}$	10807	98.7%	0.6784

Table 4. Runtimes for ABC-(SMC)^2 algorithm on SIR case study.

	Times [seconds]		
Case	ABC(SMC)^2	SVM Optimisation	SVM Classification
θ_ϕ	64790	168	3.98
$\theta_{\neg\phi}$	8014	82	4.25
$\theta_{\mathcal{U}}$	35833	2166	5.12

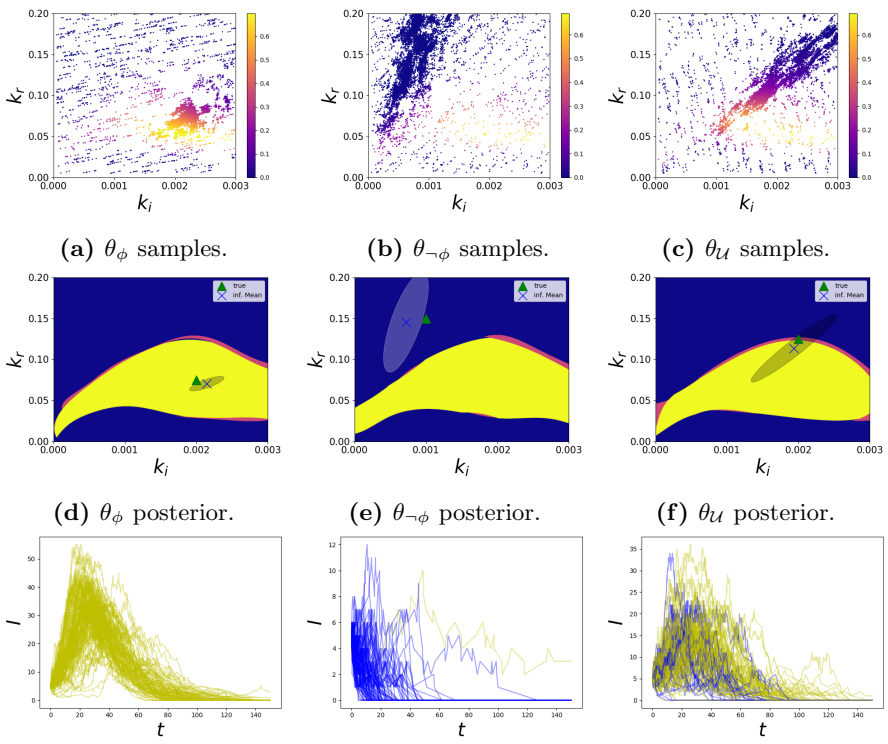
underlying stochastic system $\mathcal{S}$ does indeed satisfy the property of interest, as the credibility calculation gives $P(\mathcal{S} \models \phi \mid s_{obs}) = 1$. Case $\theta_{\neg\phi}$ has a low probability of satisfying the property of interest ($P(\mathcal{S} \models \phi \mid s_{obs}) = 0.0054$), whereas for Case $\theta_{\mathcal{U}}$ the inferred mean converges to the true mean that we would expect the estimated probability of satisfying the property to converge to, which is 0.5.

Table 2, and Figure 4 suggest that simulation times are largely dependent on the estimated probabilities, $\hat{\Lambda}_\phi(\theta)$: the closer the estimated probabilities are to 0.5, the larger the number of simulations required. To improve the runtime of Case $\theta_{\mathcal{U}}$, we would need to reduce variance and improve the accuracy of the inferred parameters, for instance by increasing the number of observed data points y_{obs} or with an alternative choice of either the summary statistics chosen or of the perturbation kernels [56]. The inferred posterior, total number of sampled parameters, SVM accuracy and output credibility calculation, are presented in Table 3, with the corresponding runtimes in Table 4.

Outcomes of the MF-ABC-(SMC)^2 Algorithm

In this section we illustrate the advantages of extending the ABC-(SMC)^2 algorithm to a multifidelity setup, whereby we can obtain a credibility calculation using computationally cheaper simulations. We also explore when to perform SMC, highlighting that we can achieve comparable results without running SMC for every sampled parameter, as proposed in the native ABC-(SMC)^2 algorithm. Furthermore, we provide motivation as to why we work with Bayesian SMC methods as opposed to a frequentist one, namely the Massart algorithm, which has been shown to be one of the best performing frequentist SMC schemes [58].

We consider three case studies, the production-degradation model, SIR model and the Michaelis-Menten model. See Methods for full details of variables and parame-



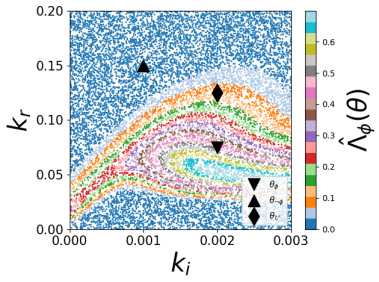
(a) θ_ϕ samples. **(b)** $\theta_{\neg\phi}$ samples. **(c)** $\theta_{\mathcal{U}}$ samples.

(d) θ_ϕ posterior. **(e)** $\theta_{\neg\phi}$ posterior. **(f)** $\theta_{\mathcal{U}}$ posterior.

(g) Samples traces from θ_ϕ posterior. **(h)** Samples traces from $\theta_{\neg\phi}$ posterior. **(i)** Samples traces from $\theta_{\mathcal{U}}$ posterior.

Figure 3. Bayesian verification results from ABC-(SMC)^2 for Case θ_ϕ (a and d), Case $\theta_{\neg\phi}$(b and e), and Case $\theta_{\mathcal{U}}$ (c and f). Sampled points θ with estimated probabilities $\hat{\Lambda}_\phi(\theta)$ (a, b and c). Inferred posterior $\pi^{h_M}_{ABC}(\theta \mid s_{obs})$ and parameter regions (d, e and f). Traces of I molecules simulated from $\mathcal{M}_\theta$, with θ sampled from the θ_ϕ posterior (g), the $\theta_{\neg\phi}$ posterior (h) and the $\theta_{\mathcal{U}}$ posterior (i). The set Θ_ϕ, is shown in yellow, whereas $\Theta_{\neg\phi}$ is shown in blue. The undecided areas $\Theta_{\mathcal{U}}$ is shown in magenta. For Fig (g-i) the traces colour represent which set the sampled parameters are members of.

ters chosen. For the multifidelity scheme, we will use pCTMCs as high-fidelity (HF) models $\mathcal{M}_\theta$, and pSDEs (based on the chemical Langevin equations, or CLE), as low-fidelity (LF) models $\tilde{\mathcal{M}}_\theta$. The CLE is known to be a good approximation for a CTMC a when the CRN has a large number of molecules [34]: this motivates to select an initial state for the SIR case study that differs from that considered in the previous experiments. We note that, for each study, a specific $\theta_\star$ is chosen to emulate an actual biological system such that $\mathcal{M}_{\theta_\star} \models \phi$, and thus $\mathcal{M}_{\theta_\star}$ is considered to be the "true" data-generating system. We summarise credibility calculations and runtimes in Table 5: we notice that time is mostly spent running MF-BSMC for the sampled parameters, whilst the parameter inference is quite rapid - this justifies the careful selection of SMC within the overall scheme discussed in this work. First,



Case	$\hat{\Lambda}_\phi(\theta)$	Total simulations
θ_ϕ	0.47254	18445
$\theta_{\neg\phi}$	0.00408719	2202
$\theta_{\mathcal{U}}$	0.100433	14775

Figure 4 & Table 2. True parameter values with corresponding estimated probabilities using SMC (15000 uniform samples), and number of SMC simulations used in ABC-(SMC)^2 .

Table 5. MF-ABC-(SMC)^2 Credibility Outcomes and Runtimes.

Case Study	$\mathcal{C}_{\text{mf}}$	Times [seconds]	
		MF-ABC-(SMC)^2	(MF-BSMC aspect)
Production-degradation	0.631	9621	(9582)
SIR	1.0	9572	(8825)
Michaelis-Menten	0.836	2423	(1701)

we provide a brief discussion as to our choice of working with Bayesian SMC as opposed to the Massart algorithm considered previously.

Frequentist vs Bayesian SMC

Both SMC techniques based on Chernoff-Hoeffding or Bayesian methods offer bounded confidence intervals, where for SMC methods utilising Chernoff-Hoeffding bounds, these bounds hold for any random variable with a bounded variance, meanwhile the Bayesian SMC approach explicitly constructs the posterior distribution based on the Bernoulli distribution. Through empirical studies, [11] shows that the performance of the BSMC algorithm depends more on the half-size ℓ of the interval than on the coverage of the interval. However, we note that the Bayesian SMC result indeed requires a fewer number of simulations in comparison with the frequentist Massart Algorithm previously considered. Whilst not formally proved here [12], we compare the frequentist SMC approach with the Bayesian approach by comparing the average number of simulations required to obtain results from BSMC (Algorithm 9) and from the Massart Algorithm [58]. Results are shown in Figure 5.

Production-degradation model

We work with a biochemical system that consists of a single chemical species X, involving production and degradation reactions of the form

$$\emptyset \xrightarrow{k_1} X, \quad X \xrightarrow{k_2} \emptyset,$$

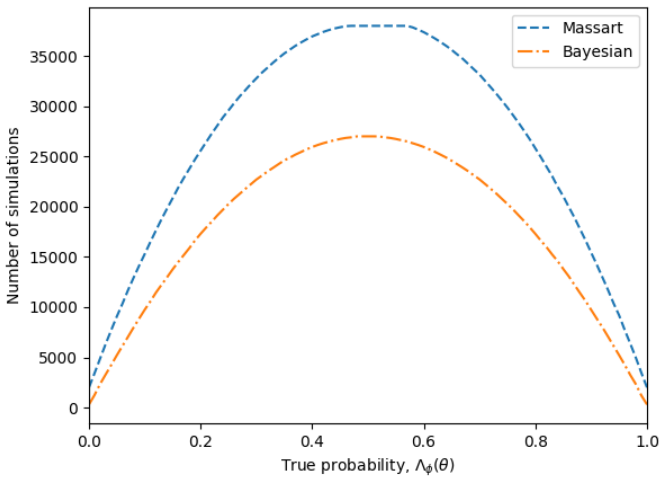


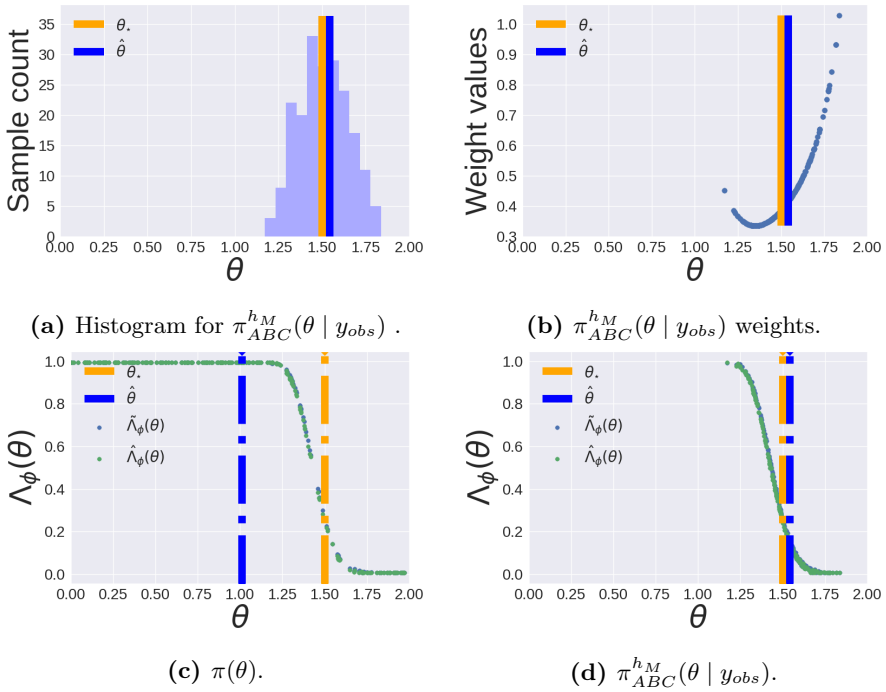
Figure 5. Comparison of average number of simulations required to estimate the true probability of satisfying a given property, $\Lambda_\phi(\theta)$ using Bayesian SMC or frequentist SMC (Massart Algorithm[48]) with coverage parameter $c = 0.999$ ($(1-\delta)$-confidence interval) and width parameter $\ell = 0.01$. ($\epsilon = 0.01$). The number of samples obtained is the mean of 100 repetitions of each SMC technique.

that is, X is produced at a rate $k_1 \in \mathbb{R}_{\geq 0}$, the parameter we wish to infer $\theta = k_1$, and X is degraded at a fixed rate of $k_2 = 0.01$. The propensity functions are given by $a_1(X(t)) = k_1$ and $a_2(X(t)) = k_2 X(t)$, where $X(t)$ is the number of X molecules at time t. The associated stoichiometric vectors are given by $\upsilon_1 = 1$ and $\upsilon_2 = -1$. We obtain the following CLE,

$$d\tilde{x}(t) = (k_1 - k_2\tilde{x}(t))dt + \sqrt{k_1}\,\mathrm{d}W^{(1)}(t) - \sqrt{k_2\tilde{x}(t)}\,\mathrm{d}W^{(2)}(t),$$

where $W^{(i)}(t)$, $i = 1, 2$ are independent scalar Wiener processes. The property we wish to verify is $\bar{\phi} = P_{\geq 0.7}(\square^{[200,500]}\,(120 \leq X \leq 180))$, that is, whether with a probability of at least 0.7 (inclusive), the number of X molecules remain between $120 \leq X \leq 180$ for $t \in [200, 500]$. This is equivalent to checking the dual property $\phi = P_{<0.3}(true\ U^{[200,500]}\,(X < 120 \cup X > 180)$, thus $\zeta = 0.3$. We present the inference and estimated satisfaction probabilities in Figure 6.

From Figure 6b we note that the weights are strictly positive, which imply that for most cases the LF model comes to the same conclusion as the HF model. (In the Methods section we instead explore the case when we obtain negative weights.) The satisfaction probabilities estimated via BSMC, for samples generated from the prior $\pi(\theta)$ and from the posterior $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$, are presented in Figures 6c and 6d, respectively. The credibility value $\mathcal{C}_{\text{mf}}$ in Table 5 is calculated by using the $\hat{\Lambda}_\phi(\theta)$ estimated from the HF model and the weights generated *from the posterior*: according to (10), the integral boils down to summing weights associated to pa-



(a) Histogram for $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$.

(b) $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$ weights.

(c) $\pi(\theta)$.

(d) $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$.

Figure 6. Production-degradation - Inference results (a) and weights (b) for sampled parameters $\theta \sim \pi^{h_M}_{ABC}(\theta \mid y_{obs})$ as well as estimated satisfaction probabilities for high-fidelity ($\hat{\Lambda}_\phi(\theta)$, blue points) and low-fidelity model ($\tilde{\Lambda}_\phi(\theta)$, green points), using samples from (c) $\pi(\theta)$ and (d) $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$. The posterior mean is denoted by $\hat{\theta}$, while the "true parameter" is denoted by $\theta_\star$.

rameters/models satisfying the property. From the satisfaction probabilities given in Figures 6c and 6d, we can see that the outcome of the Bayesian verification technique $\mathcal{C}_{\text{mf}}$ not only hinges on the accuracy and variance from the inference part, but also on the location of the posterior distribution with respect to the satisfaction probability $\Lambda_\phi(\theta)$: in this case study $\theta_\star$ lies close to the boundary of the feasible set of parameters, hence the obtained (somewhat uninformative) credibility of 0.631. As we increase the accuracy of the inference part, we expect the posterior to converge to the "true parameter" value $\theta_\star$, thus $\mathcal{C}_{\text{mf}} \to 1$ for the credibility.

SIR model

Recalling the SIR case study considered when investigating the ABC-(SMC)^2 regime, the CRN is governed by the following pair of rules

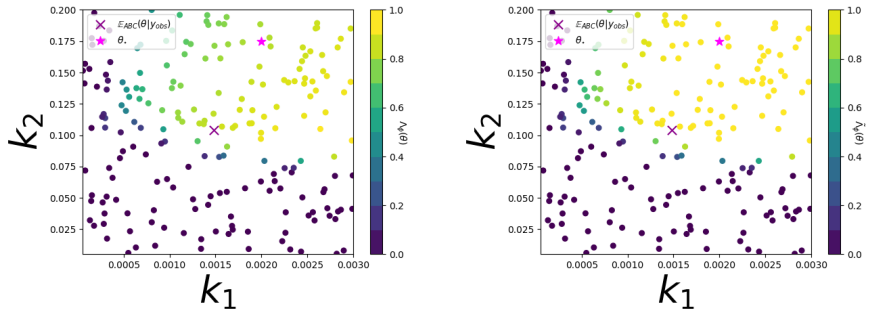
$$S + I \xrightarrow{k_1} 2I, \qquad I \xrightarrow{k_2} R,$$

where $\theta = (\theta_1, \theta_2) = (k_1, k_2)$. The above system has propensity functions $a_1(X(t)) = k_1 S(t) I(t)$ and $a_2(X(t)) = k_2 I(t)$ which have respective vectors $\boldsymbol{\upsilon}_1 = [-1, 1, 0]^\top$ and

$\boldsymbol{\upsilon}_2 = [0, -1, 1]^\top$, which leads us to the CLE approximation

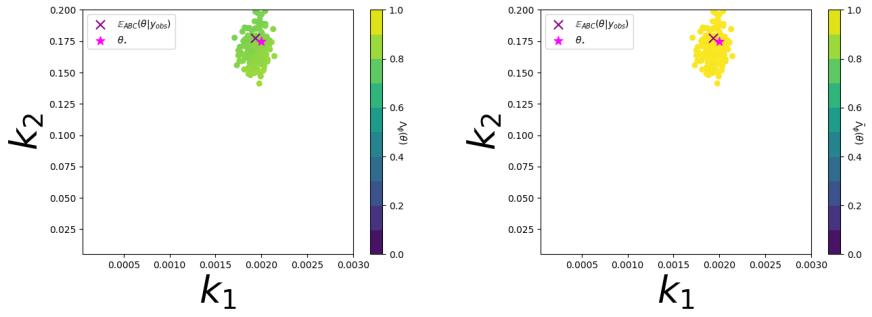
$$
\begin{aligned}
\mathrm{d}\tilde{x}^1(t) &= -k_1\tilde{x}^1(t)\tilde{x}^2(t)\,\mathrm{d}t - \sqrt{k_1\tilde{x}^1(t)\tilde{x}^2(t)}\,\mathrm{d}W^{(1)}(t),\\
\mathrm{d}\tilde{x}^2(t) &= \tilde{x}^2(t)\left(k_1\tilde{x}^1(t) - k_2\right)\mathrm{d}t + \sqrt{\tilde{x}^2(t)}\left(\sqrt{k_1\tilde{x}^1(t)}\,\mathrm{d}W^{(1)}(t) - \sqrt{k_2}\,\mathrm{d}W^{(2)}(t)\right),\\
\mathrm{d}\tilde{x}^3(t) &= \sqrt{k_2\tilde{x}^2(t)}\left(\sqrt{k_2\tilde{x}^2(t)}\,\mathrm{d}t + \mathrm{d}W^{(2)}(t)\right),
\end{aligned}
$$

where $\tilde{x}(t) = [\tilde{x}^1(t), \tilde{x}^2(t), \tilde{x}^3(t)]^\top$ are the continuous approximations to the state $X(t) = [S(t), I(t), R(t)]^\top$ and $W^{(j)}(t)$, $j = 1, 2$, are scalar independent Wiener processes. The property we consider is $\phi = P_{\geq 0.5}((I > I_0)U^{[7.5,10]}(I > R))$, expressing whether, with probability greater than or equal to $\zeta = 0.5$, the number of I molecules exceeds that of R molecules within a time interval $[7.5, 10]$.



(a) $\hat{\Lambda}_\phi(\theta)$ where $\theta \sim \pi(\theta)$.

(b) $\tilde{\Lambda}_\phi(\theta)$ where $\theta \sim \pi(\theta)$.



(c) $\hat{\Lambda}_\phi(\theta)$ where $\theta \sim \pi^{h_M}_{ABC}(\theta \mid y_{obs})$.

(d) $\tilde{\Lambda}_\phi(\theta)$ where $\theta \sim \pi^{h_M}_{ABC}(\theta \mid y_{obs})$.

Figure 7. SIR - Mean estimated satisfaction probabilities (colour bars) for high-fidelity model $\mathcal{M}_\theta$ (a and c) and low-fidelity model $\tilde{\mathcal{M}}_\theta$ (b and d) when the parameters are sampled from the prior $\pi(\theta)$ (a and b), and from the posterior $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$ (c and d).

We present the MF-BSMC results obtained from running the MF-ABC-(SMC)^2 algorithm, in Figure 7. The credibility reported in Table 5 is computed via the estimates $\hat{\Lambda}_\phi(\theta)$ where $\theta \sim \pi^{h_M}_{ABC}(\theta \mid y_{obs})$. We can see that we infer the parameters closely, however there seems to be a discrepancy between the estimated satisfaction probabilities of the HF and LF model based on the posterior distribution. If we were to consider running MF-BSMC on the prior alone, by utilising a supervised classifier akin to the parameter classification technique introduced in [2], we could

obtain a detailed feasibility set Θ_ϕ across the whole parameter space Θ, in addition to the credibility calculation $\mathcal{C}_{\text{mf}}$ - see Methods for further particulars. In addition to this, by performing MF-BSMC on the prior with an appropriate choice of distance metric (here we use $\hat{\gamma}_\epsilon$ from [53]), we can examine the relationship between $\hat{\Lambda}_\phi(\theta)$ and $\tilde{\Lambda}_\phi(\theta)$ across the parameter space Θ - see Figure 12.We conclude that whilst we can obtain an accurate credibility calculation $\mathcal{C}_{\text{mf}}$ by performing SMC for the HF model with samples generated from the posterior $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$, more information can be gathered by performing MF-BSMC with samples from the prior: this can be useful for parameter classification or to analyse the relationship between models (See Methods), in addition to calculating the overall credibility $\mathcal{C}_{\text{mf}}$.

Michaelis-Menten Case study

The Michaelis-Menten model [59, 60] comprises a state vector $X(t) = [E(t), S(t), C(t), P(t)]^\top$ with stoichiometry

$$E + S \xrightarrow{k_1} C, \quad C \xrightarrow{k_2} E + S, \quad C \xrightarrow{k_3} E + P.$$

The reactions above have the respective propensities $a_1(X(t)) = k_1 E(t)S(t)$, $a_2(X(t)) = k_2 C(t)$ and $a_3(X(t)) = k_3 C(t)$ where the respective stoichiometric vectors are given by $\boldsymbol{\upsilon}_1 = [-1, -1, 1, 0]^\top$, $\boldsymbol{\upsilon}_2 = [1, 1, -1, 0]^\top$ and $\boldsymbol{\upsilon}_3 = [1, 0, -1, 1]^\top$. With these propensity functions and stoichiometric vectors, the CLE approximation is given by

$$\begin{aligned}
\mathrm{d}\tilde{x}^1(t) &= \big((k_2 + k_3)\tilde{x}^3(t) - k_1\tilde{x}^1(t)\tilde{x}^2(t)\big)\,\mathrm{d}t - \sqrt{k_1\tilde{x}^1(t)\tilde{x}^2(t)}\,\mathrm{d}W^{(1)}(t) \\
&\quad + \sqrt{k_2\tilde{x}^3(t)}\,\mathrm{d}W^{(2)}(t) + \sqrt{k_3\tilde{x}^3(t)}\,\mathrm{d}W^{(3)}(t), \\
\mathrm{d}\tilde{x}^2(t) &= \big(k_2\tilde{x}^3(t) - k_1\tilde{x}^1(t)\tilde{x}^2(t)\big)\,\mathrm{d}t - \sqrt{k_1\tilde{x}^1(t)\tilde{x}^2(t)}\,\mathrm{d}W^{(1)}(t) + \sqrt{k_2\tilde{x}^3(t)}\,\mathrm{d}W^{(2)}(t), \\
\mathrm{d}\tilde{x}^3(t) &= \big(k_1\tilde{x}^1(t)\tilde{x}^2(t) - (k_2 + k_3)\tilde{x}^3(t)\big)\,\mathrm{d}t + \sqrt{k_1\tilde{x}^1(t)\tilde{x}^2(t)}\,\mathrm{d}W^{(1)}(t) \\
&\quad - \sqrt{k_2\tilde{x}^3(t)}\,\mathrm{d}W^{(2)}(t) - \sqrt{k_3\tilde{x}^3(t)}\,\mathrm{d}W^{(3)}(t), \\
\mathrm{d}\tilde{x}^4(t) &= k_3\tilde{x}^3(t)\,\mathrm{d}t + \sqrt{k_3\tilde{x}^3(t)}\,\mathrm{d}W^{(3)}(t),
\end{aligned}$$

where $\tilde{x}(t) = [\tilde{x}^1(t), \tilde{x}^2(t), \tilde{x}^3(t), \tilde{x}^4(t)]^\top = [\tilde{E}(t), \tilde{S}(t), \tilde{C}(t), \tilde{P}(t)]^\top$ and once again each $W^{(j)}(t)$ are independent scalar Weiner processes driving the reactions, $j = 1, 2, 3$. We are interested in verifying whether or not the true system satisfies $\phi = P_{\geq 0.8}(\Box^{[20,60]}(25 \leq E \leq 55)) = P_{\leq 0.2}(true\ U^{[20,60]}(E < 25) \cup (E > 25))$ (thus $\zeta = 0.8$), namely whether, with probability at least $\zeta = 0.8$, for time $t \in [20, 60]$ the number of enzyme molecules E remains between $25 \leq E \leq 55$. The inference results are presented in Figure 8. We generate a credibility $\mathcal{C}_{\text{mf}} = 0.836$ and expect it to converge to 1 as the inference accuracy improves.

Computational Considerations

With the MF-ABC-(SMC)^2 scheme we have succeeded in reducing both the total number of simulations required and the computational costs of said simulations by extending the ABC-(SMC)^2 with multifidelity techniques and by exploring when to perform SMC in a smarter way. With the multifidelity ABC framework, setting weights $w^{(i)}$ for sampled parameters $\theta^{(i)}$ may be calculated without incurring the

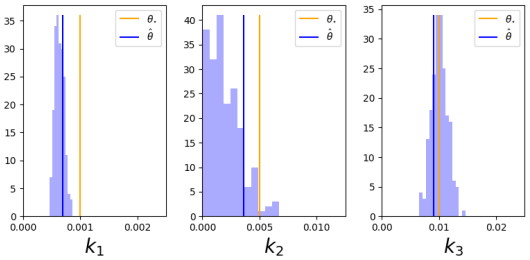


Figure 8. Michaelis-Mentern – Inference results ($\hat{\theta}$ posterior mean, $\theta_\star$ "true parameters").

computational cost of simulating y, whereas in the previous ABC-(SMC)^2 we are required to simulate $y \sim p(y \mid \theta)$ for every weight evaluation, where the computational cost of doing so is $\mathbb{E}[c(\theta)] \gg \mathbb{E}[\tilde{c}(\theta)]$, where $\mathbb{E}[\tilde{c}(\theta)]$ is the average cost of generating LF simulations $\tilde{y} \sim p(\tilde{y} \mid \theta)$ from models (here the expectation is defined over their probability spaces).

With regards to the number of simulations, recall that in the ABC-(SMC)^2 for a given model the worst-case scenario for the total number of simulations would be $\bar{N} \times n_{\mathcal{O}}$, where $n_{\mathcal{O}}$ is the maximum number of simulations required for the SMC scheme in ABC-(SMC)^2 (calculated in the Massart algorithm [58]) and $\bar{N} \geq N \times M$, where $\bar{N}$ denotes the total number of parameters sampled, whether they are accepted or not. In the MF-ABC-(SMC)^2 scheme however, rather than performing SMC for every sampled parameter $\theta^{(r)}, r = 1, \ldots, \bar{N}$, we run either SMC on parameter samples used to estimate the prior or the posterior. In this case, we would run SMC for a total of either N or $2N$ times, whether we use samples from the prior (N) or both prior and posterior ($2N$). Additionally, we see from Figure 5 that a fewer number of simulations are required to verify properties using Bayesian SMC compared to the Massart algorithm. By denoting the maximum number of simulations required for Bayesian SMC as $n_{\mathcal{B}}$, we note that $n_{\mathcal{B}} < n_{\mathcal{O}}$ and in the worst case scenario the total number of simulations required to estimate parameter regions using the MF-ABC-(SMC)^2 algorithm is equal to $N \times n_{\mathcal{B}} < 2N \times n_{\mathcal{B}} < N \times M \times n_{\mathcal{O}} \leq \bar{N} \times n_{\mathcal{O}}$.

Discussion

Discussion of ABC-(SMC)^2

The ABC-(SMC)^2 framework presented in [2] allows the Bayesian verification framework of [5] to be applied to a wide variety of models. The overall ABC-(SMC)^2 scheme can easily be parallelised over its components to drastically speed up the framework: each part of the scheme - the CRN simulations [61], the inference scheme [62] and the SMC scheme [48] - naturally allow for parallelisation. Besides parallelisation, our interests lay in reducing the overall workload: indeed no matter how the simulations are performed, either sequentially or in parallel, the ABC-(SMC)^2 framework leads to a large number of simulations and thus a possibly heavy computational burden.

Setting B_m to a small value (even $B_m = 1$), the acceptance probability is cheaper to evaluate [50] but can become highly variable, allowing for more time to ex-

plore the parameter space rather than obtaining more detailed information for a single sampled parameter that may be rejected. The trade-off is that, whilst the ABC-(SMC)^2 framework allows for a rigorous and informative Bayesian verification technique across the whole parameter space, for the credibility calculation one is solely interested in the posterior distribution, $\pi^{h_M}_{ABC}(\theta \mid s_{obs})$, and the associated satisfaction probabilities. In addition to this, by setting B_m to the minimum number of simulations required for SMC, in a worst case scenario this would lead to a total of $\bar{N} \times n_{\mathcal{O}}$ simulations, where $\bar{N} \geq N \times M$ is the total number of sampled parameters, N is the number of accepted particles, M is the total number of generations or epochs in the sequential Monte Carlo scheme, and $n_{\mathcal{O}}$ is the maximum number of simulations required for guarantees in SMC.

Whilst these proposals drastically can reduce the total number of simulations required, we look at further mitigating the computational burden of requiring a high number of expensive/slow simulations, by deploying multifidelity Monte Carlo methods [1], specifically Multifidelity ABC [3, 4].

Discussion on Multifidelity Bayesian Verification

Whilst the ABC-(SMC)^2 extension introduced in [5] greatly improves the applicability and scalability of the Bayesian verification framework, in this form the framework may require a restrictive number of simulations. Despite the ease as to which [2] can be easily parallelised, in this paper we looked to reduce the overall workload required from the ABC-(SMC)^2 technique. To reduce the workload, we looked at limiting both the number of simulations required and mitigate the computational cost of the required simulations. The former is achieved by choosing an alternative SMC technique which requires fewer simulations, as well as only performing SMC on samples generated either from the prior or posterior, allowing for a faster exploration of the parameter space [50]. The latter is achieved by extending the framework to utilise multifidelity methods, which allows one to use low-fidelity models that are cheaper and faster to simulate, in combination with high-fidelity models to perform Bayesian inference: we have thus introduced a multifidelity extension to the Bayesian verification framework of [2]. We also extended the use of multifidelity methods to allow for statistical model checking, which can be used to quantitatively assess (via the metric introduced in [53]) how well the LF model approximates the HF model.

With the introduction of multifidelity weights, however, there is a non-negative probability of obtaining negative weights - see further details in the Method section - these weights can however still be used to assess the overall credibility under the inferred posterior. Further extensions would be to consider how to avoid generating negative weights in the multifidelity framework, where mitigating the effects of negative Monte Carlo weights is the subject of current research [63, 64].

It is worth stressing the importance of selecting when to perform the multifidelity Bayesian SMC technique, MF-BSMC, as studied in our experiments. If one is interested in the credibility calculation alone, the most accurate method is to perform SMC using parameters sampled from the posterior distribution. Instead, if one is interested in generating an approximate parameter classification technique over the whole parameter space, and in exploring the relationship between the low-

and high-fidelity models (see Methods section), then one should consider performing MF-BSMC on models generated from the prior (or over the entire parameter space). By doing so, one could obtain an approximate parameter classification technique that can also be used to perform a credibility calculation, using SVMs similar to that proposed in [2]. Future work involves further investigation as to how one could formally verify properties defined on the high-fidelity model by using information gathered solely utilising the low-fidelity model.

Conclusions

In this paper we introduced a novel data-driven verification framework which utilises multifidelity methods, namely, Multifidelity Bayesian verification. Multifidelity methods for likelihood-free inference and statistical model checking, leverage both high-fidelity and low-fidelity models to reduce the computational burden of having to rely solely on a costly, high-fidelity model. The success of simulation-based techniques, whether used for inference, model checking or a wider range of Monte Carlo methods, hinge solely on the computational complexity of the underlying model. Simulation-based techniques remain popular however due to the higher degree of flexibility allowed when it comes to mathematical modelling, despite the high number of simulations required. By leveraging both low-fidelity and high-fidelity models, we can reduce the computational burden, and as such improve the scalability, of techniques such as [2], which rely entirely on simulation-based methods. With this multifidelity extension, the Bayesian verification technique introduced above can be used on a wider class of low- and high-fidelity models, allowing for a more rapid inference and verification.

Methods

In this section, we present the algorithms, concepts and further details on the case studies which is required to introduce our novel Multifidelity Bayesian Verification framework. Specifically we present the likelihood-free inference algorithms, statistical model checking and their multifidelity extensions. Many of the details of these algorithms are presented in the background and we present the novel framework and case studies in the Results section. To motivate our use of the low-fidelity models, we present a derivation of the chemical Langevin equation, followed by supplementary material regarding the case studies.

Algorithms

Algorithm 3 ABC Rejection Sampler (ABC-RS)

Input:

- Observed data y_{obs}
- Prior distribution $\pi(\theta)$ and data generating function $p(y \mid \theta)$ for model $\mathcal{M}_\theta$
- A neighbourhood $\Omega_h = \{y \in \mathcal{Y} \mid d(y, y_{obs}) \leq h\}$ with corresponding kernel $\mathbb{1}_{\Omega_h}(y)$
- $N > 0$, Monte Carlo sample size

Output:

- $\{\theta^{(i)}, w^{(i)}\}_{i=1}^N$, accepted parameters with corresponding weights

for $i = 0, \ldots, N$ **do**
  Generate $\theta^{(i)} \sim \pi(\theta)$
  Simulate $y \sim p(y \mid \theta^*)$
  Calculate $w^{(i)} = \mathbb{1}_{\Omega_h}(y)$
end for
return $\{\theta^{(i)}, w^{(i)}\}_{i=1}^N$

Algorithm 4 ABC Importance Sampler (ABC-IS)

Input:

- Observed data y_{obs}
- Prior distribution $\pi(\theta)$ and data generating function $p(y \mid \theta)$ for model $\mathcal{M}_\theta$
- Importance distribution $\hat{q}$ proportional to q
- A neighbourhood $\Omega_h = \{y \in \mathcal{Y} \mid d(y, y_{obs}) \leq h\}$ with corresponding kernel $\mathbb{1}_{\Omega_h}(y)$
- $N > 0$, Monte Carlo sample size
- $B_m > 0$ number of simulations

Output:

- $\{\theta^{(i)}, w^{(i)}\}_{i=1}^N$, accepted parameters with corresponding weights

for $i = 0, \ldots, N$ **do**
  Generate $\theta^{(i)} \sim \hat{q}(\theta)$
  for $b = 1, \ldots, B_m$: **do**
    Simulate $y_b \sim p(y \mid \theta^{(i)})$ (if working with summary statistics, set $s_b = S(y_b)$)
  end for
  Calculate $w^{(i)} = \left[\pi(\theta^{(i)})/q(\theta^{(i)})\right] \cdot \sum_{b=1}^{B_m} \mathbb{1}_{\Omega_h}(y_b)$
end for
return $\{\theta^{(i)}, w^{(i)}\}_{i=1}^N$

Algorithm 5 ABC-SMC

Input:

- Prior $\pi(\theta)$ and data-generating likelihood function $p(y_{obs} \mid \theta)$
- A neighbourhood that evolves with distance thresholds $\Omega_{h_m} = \{y \in \mathcal{Y} \mid d(y, y_{obs}) \leq h_m\}$ with corresponding kernel $K_{h_m}(y)$ $(= \mathbb{1}_{\Omega_{h_m}}(y)$ in this paper)
- $N > 0$, number of particles used to estimate posterior distributions
- Sequence of perturbation kernels $F_m(\theta \mid \theta^*)$, $m = 0, 1 \ldots, M$
- A quantile $\upsilon \in [0, 1]$ to control the rate of decrease of h_m
- Summary statistic function $s = S(y)$
- $B_m > 0$, number of simulations per sampled particle. For stochastic systems $B_m > 1$

Output:

- Set of weighted parameter vectors $\left\{\theta_M^{(i)}, w_M^{(i)}\right\}_{i=1}^N$ drawn from

$$\pi_{ABC}(\theta \mid s_{obs}) \propto \int K_{h_M}(s) p(y \mid \theta) \pi(\theta) \, \mathrm{d}s$$

1: **for** $m = 0, \ldots, M$: **do**
2: **for** $i = 0, \ldots, N$: **do**
3: **if** $m = 0$ **then**
4: Generate $\theta^{**} \sim \pi(\theta)$
5: **else**
6: Generate θ^* from the previous population $\{\theta_{m-1}^{(i)}\}$ with weights $\{w_{m-1}^{(i)}\}$ and perturb the particle to obtain $\theta^{**} \sim F_m(\theta \mid \theta^*)$
7: **end if**
8: **if** $\pi(\theta^{**}) = 0$ **then**
9: **goto** line 4
10: **end if**
11: **for** $b = 1, \ldots, B_m$: **do**
12: Generate $y_b \sim p(y \mid \theta^{**})$
13: Calculate $s_b = s(y_b)$
14: **end for**
15: Calculate $b_m(\theta^{**}) = \sum_{b=1}^{B_m} K_{h_m}(s_b)$
16: **if** $b_m(\theta^{**}) = 0$ **then**
17: **goto** line 4
18: **end if**
19: Set $\theta_m^{(i)} = \theta^{**}$, $\bar{d}_m^{(i)} = \frac{1}{B_m} \sum_{b=1}^{B_m} \|s_b - s_{obs}\|$ and calculate

20:

$$w_m^{(i)} = \begin{cases} b_m\left(\theta_m^{(i)}\right), & \text{if } t = 0 \\ \dfrac{\pi\left(\theta_m^{(i)}\right) b_m\left(\theta_m^{(i)}\right)}{\sum_{j=1}^N w_{m-1}^{(j)} F_m\left(\theta_m^{(i)} \mid \theta_{m-1}^{(j)}\right)}, & \text{if } t > 0 \end{cases}$$

21: **end for**
22: Generate $\{\theta_m^{(i)}, w_m^{(i)}\}_{i=1}^N$ using Algorithm 4 (ABC-IS) with importance distribution $f_m(\theta)$ and neighbourhood Ω_{h_m}
23: Normalise weights: $w_m^{(i)} \leftarrow w_m^{(i)} / \left(\sum_{i=1}^N w_m^{(i)}\right)$
24: Set $h_{m+1} = (\upsilon / N) \sum_{i=1}^N \bar{d}_m^{(i)}$
25: **end for**
26: **return** $\left\{\left(\theta_M^{(i)}, w_M^{(i)}\right)\right\}_{i=1}^N$

Algorithm 6 Multifidelity ABC Rejection Sampler (MF-ABC-RS)

Input:

- Observed data y_{obs} $(= \tilde{y}_{obs})$
- Prior distribution $\pi(\theta)$, high-fidelity model $p(y \mid \theta)$ and low-fidelity model $\tilde{p}(\tilde{y} \mid \theta)$
- High fidelity neighbourhood $\Omega_h = \{y \in \mathcal{Y} \mid d(y, y_{obs}) \leq h\}$ and low-fidelity neighbourhood $\Omega_{\tilde{h}} = \{\tilde{y} \in \tilde{\mathcal{Y}} \mid \tilde{d}(\tilde{y}, \tilde{y}_{obs}) \leq \tilde{h}\}$ with respective kernels $\mathbb{1}_{\Omega_h}(y)$ and $\mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y})$
- Continuation probabilities (η_1, η_2)
- Monte Carlo sample size $N > 0$

Output:

- $\{\theta^{(i)}, w^{(i)}\}_{i=1}^N$, accepted parameters with corresponding weights

for $i = 0, \ldots, N$ **do**
  Generate $\theta^{(i)} \sim \pi(\theta)$ and $U \sim \mathcal{U}(0,1)$
  Simulate $\tilde{y} \sim \tilde{p}(\tilde{y} \mid \theta^{(i)})$ from low-fidelity model
  Calculate $\tilde{w} = \mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y})$
  Set $w^{(i)} = \tilde{w}$ and $\eta = \eta_1 \tilde{w} + \eta_2(1 - \tilde{w})$
  if $U < \eta$ **then**
    Simulate $y \sim p(y \mid \theta^{(i)})$ from the high-fidelity model
    Calculate $w = \mathbb{1}_{\Omega_h}(y)$
    Update $w^{(i)} \leftarrow w^{(i)} + (w - w^{(i)})/\eta$
  end if
end for
(Optional) Calculate the approximation to $\mathbb{E}_{ABC}(g(\theta) \mid y_{obs})$, $\hat{\mathbb{E}}_{ABC}(g(\theta) \mid y_{obs}) = \sum_{i=1}^N w^{(i)} g(\theta^{(i)}) / \sum_{i=1}^N w^{(i)}$
return $\{\theta^{(i)}, w^{(i)}\}_{i=1}^N$

Algorithm 7 Multifidelity ABC Importance Sampler (MF-ABC-IS)

Input:

- Observed data y_{obs} $(= \tilde{y}_{obs})$
- Prior distribution $\pi(\theta)$, high-fidelity model $p(y \mid \theta)$ and low-fidelity model $\tilde{p}(\tilde{y} \mid \theta)$
- Importance distribution $\hat{q}(\theta) \propto q(\theta)$
- High fidelity neighbourhood $\Omega_h = \{y \in \mathcal{Y} \mid d(y, y_{obs}) \leq h\}$ and low-fidelity neighbourhood $\Omega_{\tilde{h}} = \{\tilde{y} \in \tilde{\mathcal{Y}} \mid \tilde{d}(\tilde{y}, \tilde{y}_{obs}) \leq \tilde{h}\}$ with respective kernels $\mathbb{1}_{\Omega_h}(y)$ and $\mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y})$
- Continuation probability function $\eta = \eta(\theta, \tilde{y})$
- Monte Carlo sample size $N > 0$

Output:

- $\{\theta^{(i)}, w^{(i)}, \}_{i=1}^N$, accepted parameters with corresponding weights

for $i = 0, \ldots, N$ **do**
  Generate $\theta^{(i)} \sim \hat{q}(\theta)$ and $U \sim \mathcal{U}(0,1)$
  Simulate $\tilde{y} \sim \tilde{p}(\tilde{y} \mid \theta^{(i)})$ from low-fidelity model
  Calculate $\tilde{w} = \mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y})$
  Set $w^{(i)} = \tilde{w}$
  if $U < \eta(\theta, \tilde{y})$ **then**
    Simulate $y \sim p(y \mid \theta^{(i)})$ from the high-fidelity model
    Calculate $w = \mathbb{1}_{\Omega_h}(y)$
    Update $w^{(i)} \leftarrow w^{(i)} + (w - w^{(i)})/\eta$
  end if
  $w^{(i)} \leftarrow \left[\pi(\theta^{(i)})/q(\theta^{(i)})\right] w^{(i)}$
end for
return $\{\theta^{(i)}, w^{(i)}\}_{i=1}^N$

Algorithm 8 Multifidelity ABC-SMC (MF-ABC-SMC)

Input:

- Observed data y_{obs} $(= \tilde{y}_{obs})$
- Prior distribution $\pi(\theta)$, high-fidelity model $p(y \mid \theta)$ and low-fidelity model $\tilde{p}(\tilde{y} \mid \theta)$
- Initial importance distribution $\hat{r}_0(\theta)$ (commonly set to $\hat{r}_0(\theta) = \pi(\theta)$)
- Sequence of high fidelity neighbourhoods $\Omega_{h_m} = \{y \in \mathcal{Y} \mid d(y, y_{obs}) \leq h_m\}$ and low-fidelity neighbourhoods $\Omega_{\tilde{h}_m} = \{\tilde{y} \in \tilde{\mathcal{Y}} \mid \tilde{d}(\tilde{y}, \tilde{y}_{obs}) \leq \tilde{h}_m\}$ with respective kernels $\mathbb{1}_{\Omega_h}(y)$ and $\mathbb{1}_{\Omega_{\tilde{h}}}(\tilde{y})$ and $m = 1, 2, \ldots, M$
- Initial continuation probability function $\eta_0 = \eta_0(\theta, \tilde{y}) = 1$
- Lower bounds on continuation probabilities, $\eta_1^L, \eta_2^L \in (0, 1)$
- Sequence of perturbation kernels $F_m(\theta|\theta^*)$
- Monte Carlo sample size $N > 0$

Output:

- $\{\theta^{(i)}, w^{(i)}, \}_{i=1}^N$, accepted parameters with corresponding weights

for $m = 0, \ldots, M$ **do**
  Produce $\{\theta_m^{(i)}, w_m^{(i)}\}_{i=1}^N$ and store information required to estimate optimal continuation probabilities ([4]) from MF-ABC-IS algorithm (Algorithm 7) with inputs $\left(y_{obs}, \tilde{y}_{obs}, \Omega_{h_m}, \Omega_{\tilde{h}_m}, \pi(\theta), p(y \mid \theta), \tilde{p}(\tilde{y} \mid \theta), \eta_m(\theta, \tilde{y})\right)$ setting $\hat{q} = \hat{r}_m(\theta)$
  Set $\hat{w}_m^{(i)} = |w_m^{(i)}|$ for $i = 1, \ldots, N$
  Define $\hat{r}_{m+1}$ proportional to r_{m+1} (See [4])
  Calculate (η_1^*, η_2^*) with lower bounds (η_1^L, η_2^L) using importance distribution $r_{t+1}(\theta)$, ABC thresholds h_{m+1} and $\tilde{h}_{m+1}$ with values stored from MF-ABC-IS algorithm at step 2 (See [4]).
  Set $\eta_{m+1}(\theta, \tilde{y}) = \eta_1^* \mathbb{1}_{\Omega_{\tilde{h}_m}}(\tilde{y}) + \eta_2^*(1 - \mathbb{1}_{\Omega_{\tilde{h}_m}}(\tilde{y}))$
end for
return $\{\theta_M^{(i)}, w_M^{(i)}\}_{i=1}^N$

Algorithm 9 Bayesian Statistical Model Checking (BSMC)

Input:

- Parametrised stochastic model $\mathcal{M}_\theta$
- Property ϕ defined in a formal logic, with path formula φ
- Half-interval width size $\ell \in (0, \frac{1}{2})$
- Interval coverage parameter $c \in (\frac{1}{2}, 1)$
- Prior Beta distribution with parameters α, β for (unknown) probability $\Lambda_\phi(\theta) = P(\mathcal{M}_\theta \models \phi)$

Output:

- Posterior mean estimate $\hat{\Lambda}_\phi(\theta)$ for true probability $\Lambda_\phi(\theta)$
- Interval of width 2ℓ, $(l_0, l_1) = (\hat{\Lambda}_\phi^L(\theta), \hat{\Lambda}_\phi^U(\theta))$ with posterior probability at least c

Initialise $n = 0$, total number of simulations.
Initialise $v = 0$, number of traces satisfying, $y \models \varphi$
while $\Upsilon < c$ **do**
  Generate trace $y \sim p(y \mid \theta)$ from $\mathcal{M}_\theta$
  $n \leftarrow n + 1$
  if $y \models \varphi$ **then**
    $v \leftarrow v + 1$
  end if
  Calculate $\hat{\Lambda}_\phi(\theta) = (v + \alpha)/(n + \alpha + \beta)$
  Set $(l_0, l_1) = (\hat{\Lambda}_\phi(\theta) - \ell, \hat{\Lambda}_\phi(\theta) + \ell)$
  if $l_1 > 1$ **then**
    $(l_0, l_1) = (1 - 2\ell, 1)$
  else if $l_0 < 0$ **then**
    $(l_0, l_1) = (0, 2\ell)$
  end if
  Compute posterior probability $\Upsilon = F_{(v+\alpha, n-v+\beta)}(l_1) - F_{(v+\alpha, n-v+\beta)}(l_0)$
end while
return Posterior mean estimate, $\hat{\Lambda}_\phi(\theta)$ and interval $(l_0, l_1) = \left(\hat{\Lambda}_\phi^L(\theta), \hat{\Lambda}_\phi^U(\theta)\right) = \left(\hat{\Lambda}_\phi(\theta) - \ell, \hat{\Lambda}_\phi(\theta) + \ell\right)$.

Algorithm 10 Multifidelity Bayesian Statistical Model Checking (MF-BSMC)

Input:

- Parametrised high-fidelity (HF) model $p(y \mid \theta)$ and low-fidelity (LF) model $\tilde{p}(\tilde{y} \mid \theta)$
- Property ϕ defined over HF and LF model, encoded as temporal logic formulae
- Half-interval width size $\ell \in (0, \frac{1}{2})$
- Interval coverage parameter $c \in (\frac{1}{2}, 1)$
- Prior Beta distribution with parameters α, β
- Parameter to obtain a $(1-\varepsilon)$-level approximate abstraction, ε

Output:

- Posterior mean estimates for HF satisfaction probability $\hat{\Lambda}_\phi(\theta)$ and LF sat. prob. $\tilde{\Lambda}_\phi(\theta)$
- Interval of width 2ℓ for HF model $(l_0, l_1) = (\hat{\Lambda}^L_\phi(\theta), \hat{\Lambda}^U_\phi(\theta))$, and for LF model $(\tilde{l}_0, \tilde{l}_1) = (\tilde{\Lambda}^L_\phi(\theta), \tilde{\Lambda}^U_\phi(\theta))$ with posterior probability at least c
- Measure of quality of the LF model compared to HF, $\hat{\gamma}_\varepsilon$ (Using [53] in this work)

Initialise $n = 0$, $\tilde{n} = 0$, total number of simulations
Initialise $v = 0$ and $\tilde{v} = 0$, number of traces satisfying, $y \models \phi$ and $\tilde{y} \models \phi$ respectively
Initialise $n_\gamma = 0$
while $\Upsilon < c$ or $\tilde{\Upsilon} < c$ **do**
  if $\Upsilon < c$ **then**
    Generate trace $y \sim p(y \mid \theta)$ from $\mathcal{M}_\theta$
    $n \leftarrow n + 1$
    if $y \models \varphi$ **then**
      $v \leftarrow v + 1$
    end if
    Calculate $\hat{\Lambda}_\phi(\theta) = (v + \alpha)/(n + \alpha + \beta)$
    Set $(l_0, l_1) = (\hat{\Lambda}_\phi(\theta) - \ell, \hat{\Lambda}_\phi(\theta) + \ell)$
    if $l_1 > 1$ **then**
      $(l_0, l_1) = (1 - 2\ell, 1)$
    else if $l_0 < 0$ **then**
      $(l_0, l_1) = (0, 2\ell)$
    end if
    Compute posterior probability $\Upsilon = F_{(v+\alpha, n-v+\beta)}(l_1) - F_{(v+\alpha, n-v+\beta)}(l_0)$
  end if
  if $\tilde{\Upsilon} < c$ **then**
    Generate trace $\tilde{y} \sim \tilde{p}(\tilde{y} \mid \theta)$ from $\tilde{\mathcal{M}}_\theta$
    $\tilde{n} \leftarrow \tilde{n} + 1$
    if $\tilde{y} \models \tilde{\varphi}$ **then**
      $\tilde{v} \leftarrow \tilde{v} + 1$
    end if
    Calculate $\tilde{\Lambda}_\phi(\theta) = (\tilde{v} + \alpha)/(\tilde{n} + \alpha + \beta)$
    Set $(\tilde{l}_0, \tilde{l}_1) = (\tilde{\Lambda}_\phi(\theta) - \ell, \tilde{\Lambda}_\phi(\theta) + \ell)$
    if $\tilde{l}_1 > 1$ **then**
      $(\tilde{l}_0, \tilde{l}_1) = (1 - 2\ell, 1)$
    else if $\tilde{l}_0 < 0$ **then**
      $(\tilde{l}_0, \tilde{l}_1) = (0, 2\ell)$
    end if
    Compute posterior probability $\tilde{\Upsilon} = F_{(\tilde{v}+\alpha, \tilde{n}-\tilde{v}+\beta)}(\tilde{l}_1) - F_{(\tilde{v}+\alpha, \tilde{n}-\tilde{v}+\beta)}(\tilde{l}_0)$
  end if
  if $n = \tilde{n}$ **then**
    $i \leftarrow i + 1$
    Calculate $\hat{\gamma}^{(i)} = d_T(\tilde{y}, y)$ and store
    $n_\gamma \leftarrow n_\gamma + 1$
  end if
end while
Calculate number of z values of $\hat{\gamma}^{(i)}$, $i = 1, \ldots, n_\gamma$ to discard via [53] with parameter ε
Set

$$\hat{\gamma}_\varepsilon = \max_{i \in \{1,2,\ldots,n_\gamma\} \setminus \{j_1, j_2, \ldots, j_z\}} \hat{\gamma}^{(i)},$$

where $\{j_1, j_2, \ldots, j_z\} \subset \{1, 2, \ldots, n_\gamma\}$ denote the indices of the discarded values of $\hat{\gamma}^{(i)}$
return HF estimates, $\{\hat{\Lambda}_\phi(\theta), \hat{\Lambda}^L_\phi(\theta), \hat{\Lambda}^U_\phi(\theta)\}$, LF estimates, $\{\tilde{\Lambda}_\phi(\theta), \tilde{\Lambda}^L_\phi(\theta), \tilde{\Lambda}^U_\phi(\theta)\}$ and $\hat{\gamma}_\varepsilon$

Derivation and Use of Chemical Langevin Equation

As mentioned in the main body of the text, SDEs can be integrated numerically via the Euler-Maruyama sheme [32], which is introduced next.

Definition 7 (Euler-Maruyama Method) *Given an SDE of the form*

$$\mathrm{d}\tilde{x}(t) = \varsigma_0(\tilde{x}(t))\ \mathrm{d}t + \sum_{j=1}^{J} \varsigma_j(\tilde{x}(t))\ \mathrm{d}W(t), \tag{12}$$

it can numerically simulated over an increasing sequence of sampling times $\{\tau_k\}_{k\in\mathbb{N}}$ (finite or infinite) via the Euler-Maruyama scheme [32, 65], which generates a stochastic difference equation of the following form:

$$\tilde{x}(\tau_{k+1}) = \tilde{x}(\tau_k) + \varsigma_0(\tilde{x}(\tau_k))(\tau_{k+1} - \tau_k) + \sum_{j=1}^{J} \varsigma_j(\tilde{x}(\tau_k))(W(\tau_{k+1}) - W(\tau_k)), \tag{13}$$

where $W(t)$ is the Brownian motion, evaluated at time $t \in \mathbb{R}_{\geq 0}$. In this work we shall exclusively focus on finite time horizons, $t \in [0, T]$ and assume the following: let $\Delta t = T/K$ for some $K \in \mathbb{N}$, then $\tau_k = k\Delta t$, where $k = 1, 2, \ldots, K$, is a uniform sequence of sampling times.

In this work we utilise the chemical Langevin equation (CLE) for the "low-fidelity" models of CRNs. We let the random variable $\mathcal{K}_j(X(t), \Delta t)$ denote the number of R_j reactions that occur over a given time interval $[t, t + \Delta t]$ from the molecule configuration $X(t)$, and for brevity we omit the explicit dependencies of the propensity function, $a_j = a_j(X(t))$. The number of molecules of chemical species i at time $t + \Delta t$ is then given as

$$X_i(t + \Delta t) = X_i(t) + \sum_{j=1}^{N_R} \mathcal{K}_j(X(t), \Delta t)\upsilon_{i,j}. \tag{14}$$

Calculating $\mathcal{K}_j(X(t), \Delta t)$ can be well approximated by imposing two conditions[34]. The first condition is to choose Δt small enough such that $a_j(X(t')) \cong a_j(X(t))$, $\forall t' \in [t, t + \Delta t],\ \forall j \in \{1, \ldots, N_R\}$. Since the reactions that occur during the time interval $[t, t+\Delta t]$ do not affect the propensity functions drastically, all of the reaction events occurring within this time interval can be considered to be independent of one another. Thus, each $\mathcal{K}_j(X(t), \Delta t)$ will be a statistically independent Poisson random variable, $\mathcal{P}_j(a_j, \Delta t)$, which allows for the approximation,

$$X_i(t + \Delta t) = X_i(t) + \sum_{j=1}^{N_R} \mathcal{P}_j(a_j, \Delta t),\ \text{for } i = 1, \ldots, N. \tag{15}$$

Conversely, the second condition we require is that Δt is large enough, such that the expected number of times that each reaction R_j is fired in $[t, t + \Delta t]$ is much larger than 1, $\mathbb{E}\left[\mathcal{P}_j(a_j, \Delta t)\right] = a_j \Delta t \gg 1$, $\forall j \in \{1, \ldots, N_R\}$. When both conditions are met, the Poisson random variable $\mathcal{P}_j(a_j, \Delta t)$ can be approximated by

a Gaussian r.v. with the same mean and variance, namely $\mathcal{P}_j(a_j(X(t)), \Delta t) \approx \mathcal{N}(a_j(X(t))\Delta t, a_j(X(t))\Delta t)$. Accordingly, the evolution of the state vector $X_i(t)$ is now described by the continuous approximation $\tilde{x}^i(t)$,

$$\tilde{x}^i(t + \Delta t) = \tilde{x}^i(t) + \sum_{j=1}^{N_R} \upsilon_{i,j} \mathcal{N}_j(a_j \Delta t, a_j \Delta t),\ i = 1, \ldots, N_S, \tag{16}$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes the Normal random variable with mean μ and variance σ^2, and where $a_j = a_j(\tilde{x}(t))$. Expressing $\mathcal{N}(\mu, \sigma^2) = \mu + \sigma\mathcal{Z}$, where $\mathcal{Z} \sim \mathcal{N}(0, 1)$, this equation now becomes

$$\tilde{x}^i(t + \Delta t) = \tilde{x}^i(t) + \sum_{j=1}^{N_R} \upsilon_{i,j} a_j(Y(t))\Delta t + \sum_{j=1}^{N_R} \upsilon_{i,j} \sqrt{a_j(Y(t))\Delta t} \mathcal{Z}_j. \tag{17}$$

The stochastic difference equation in (17) is in fact an Euler-Maruyama discretisation [36] with uniform time samples of duration Δt of an SDE of the form [31, 32]

$$\mathrm{d}\tilde{x}(t) = \sum_{j=1}^{N_R} \boldsymbol{\upsilon}_j a_j(\tilde{x}(t))\, \Delta t + \sum_{j=1}^{N_R} \boldsymbol{\upsilon}_j \sqrt{a_j(\tilde{x}(t))}\, \mathrm{d}W^j(t), \tag{18}$$

where $W^j(t)$ are independent scalar Brownian motions. We note that the dimensionality of the Brownian motion $d = N_S$ and that dimensionality of the stochastic process $J = N_R$ from Definition 3.

Supplementary Information for Case Studies

Practicalities

In this section we provide more detail behind the case studies, including generating traces, y_{obs}, the variables and choice of parameters. Drawing simulations $y \sim p(y \mid \theta)$ from $\mathcal{M}_\theta$ is equivalent to drawing a sample from the stochastic simulation algorithm (or Gillespie's algorithm) [66] up to a time T, whereas drawing $\tilde{y} \sim \tilde{p}(\tilde{p} \mid \theta)$ from $\tilde{\mathcal{M}}_\theta$ is equivalent to drawing samples traces from the CLE given in Equation (1) up to a time T, where we utilise the Euler-Maruyama scheme to integrate the SDEs, with a time step Δt chosen subject to the case study. We denote the observed data as y_{obs}, generating the data for y_{obs} using "true parameters" $\theta^\star$ (or θ_ϕ, $\theta_{\neg\phi}$ and $\theta_{\mathcal{U}}$ in the ABC-(SMC)^2 case study) and set $\tilde{y}_{obs} = y_{obs}$. The observed data is an average of a finite number of traces, denoted by n_{sims}, generated from the true model $\mathcal{M}_{\theta^\star}$, up to a time T_{obs} sampled at n_{obs} discrete points in time. Whilst the model used to generate the data y_{obs} is intrinsically noisy or stochastic, to emulate observation error in some case studies we perturb the observations further $y_{obs}(\bar{t}_k)$ with noise, given by σ^2, $y_{obs}(\bar{t}_k) \sim \mathcal{N}(y_{obs}(\bar{t}_k), \sigma^2)$ where $\bar{t}_k$ are the times we take observations. We set $d(a,b) = \tilde{d}(a,b) = \|a - b\|$ over time. We verify the same property on both the high-fidelity and low-fidelity model, ϕ, where the property is defined in CSL. In these case studies, we work with a fixed number of M thresholds for the algorithm to iterate through, and we set the initial thresholds $h_0 = \infty$ and $\tilde{h}_0 = \infty$ such that we accept every sample generated from the prior. Rather than inputting a predetermined sequence of neighbourhoods Ω_{h_m} and $\Omega_{\tilde{h}_m}$, we set h_{m+1} and $\tilde{h}_{m+1}$ to be equal to the average distances of the accepted sampled parameters in threshold h_m, such that $h_{m+1} < h_m$ and we will terminate the algorithm if this is not the case. The credibility calculations and runtimes for the MF-ABC-(SMC)^2 case studies are given in Table 5, which include the time taken to perform MF-BSMC for samples generated from the prior and the posterior, and in Table 7 we present all the parameters and variables for the case studies considered. For each of the case studies, the prior $\pi(\theta)$, is chosen to be a non-informative uniform over the parameter space Θ for each case study. We estimate the satisfaction probabilities for the HF and LF model using MF-BSMC (Algorithm 10) where the parameters are the multifidelity weighted parameters from the MF-ABC-SMC part of the algorithm, $\{\theta_M^{(i)}, w_M^{(i)}\}_{i=1}^N$.

The effect of negative weights

In the MF-ABC framework that we are now working in, there is a non-negative probability of obtaining negative weights. We now consider the case when negative weights are present and what effect they have on the results obtained from the MF-ABC-(SMC)^2 algorithm.

Considering the production-degradation case study in the main text, from Figure 9 we see that the algorithm infers a posterior mean of $\hat{\theta} = 1.3992$ and we see that the weights are mainly positive, meaning that the LF model comes to the same acceptance/rejection decision as the HF model. However, we note the magnitude of the singular negative weight - recalling the fact that negative weights arise when $\tilde{y} \in \Omega_{\tilde{h}}$ but $y \notin \Omega_h$. The magnitude of this negative weights influences the credibility calculation $\mathcal{C}$ as it influences the posterior mean parameter $\hat{\theta}$ and in turn $\mathcal{C}_{\text{mf}}$.

Variables/Parameters	SIR Case Study
Θ	$[5 \times 10^{-5}, 0.003]$ $\times [0.005, 0.2]$
θ_ϕ	$[0.002, 0.075]^\top$
$\theta_{\neg\phi}$	$[0.001, 0.15]^\top$
$\theta_{\mathcal{U}}$	$[0.002, 0.125]^\top$
x_0	$[95, 5, 0]^\top$
T	150
T_{obs}	150
n_{obs}	10
n_{sims}	5
σ	0
N	200
M	20
ϵ	0.01
δ	0.05
δ_c	0.001
$n_{\mathcal{O}}$	18445

Table 6. Variables and Parameters used in the ABC-(SMC)^2 Case Studies.

As we can see from Figure 9, the negative weight skews the posterior mean and the resulting credibility calculation, which is given by $\mathcal{C}_{\text{mf}} = 0.1935$. If we were to only consider the positive weights (that is, disregarding the only negative weight, not taking the absolute values), we obtain a posterior mean of $\hat{\theta}^+ = 1.5188$, which is much closer to the true value $\theta_\star$ and the resulting credibility calculation increases to $\mathcal{C}^+_{\text{mf}} = 0.5726$ when considering only the positive weights. When working with multifidelity methods, care must be taken when negative weights arise as the negative weights can influence the inferred posterior distribution. Whilst these negative weights could be thought of as penalising a region of the parameter space when $\tilde{y} \in \tilde{\Omega}_{\tilde{h}_m}$ but $y \notin \Omega_{h_m}$, in the MF-ABC-(SMC)^2 scheme the importance sampling distribution $\hat{r}_m(\theta)$ works with an absolute value of these negative weights, which might lead to additional density in regions where $w_m^{(i)} < 0$ is more likely. However, if the magnitude of these weights are large, the sampling scheme can lead to oversampling regions in the parameter space when the weight is negative which could further influence $\hat{\theta}$ and $\mathcal{C}_{\text{mf}}$. We plot the estimated satisfaction probabilities for the weighted sample that contains negative weights in Figure 9. We note the location of $\hat{\theta}$ and compare this with $\hat{\theta}^+$, where in Figure 8d the location of $\hat{\theta}^+$ is indistinguishable from the "true parameter" $\theta_\star$.
Future work will consider how to avoid generating negative weights in the multifidelity inference framework.

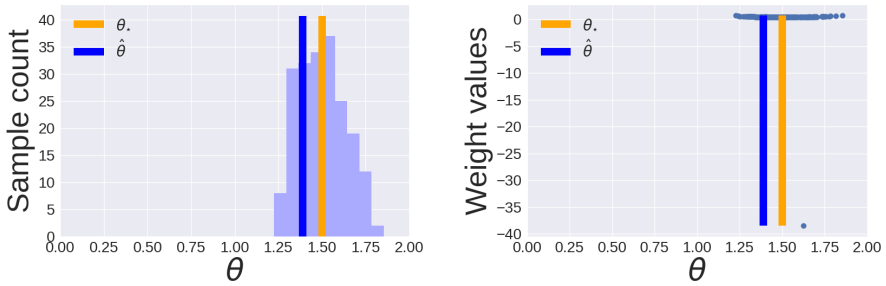
Parameter Classification
Akin to the work presented in [2] we can use estimated satisfaction probabilities to perform an approximate parameter classification, namely partitioning the pa-

Var./Par.	Case Studies		
	Prod-Deg	SIR	Michaelis-Menten
Θ	$[0, 2.0]$	$[5 \times 10^{-5}, 0.003]$ $\times[0.005, 0.2]$	$[0.0, 5.0 \times 10^{-3}]$ $\times[0.0, 2.5 \times 10^{-2}]$ $\times[0.0, 5.0 \times 10^{-2}]$
$\theta_\star$	1.5	$[0.002, 0.175]^\top$	$[0.001, 0.005, 0.01]^\top$
$\hat{\theta}$	1.54	$[0.001933, 0.17763]^\top$	$[0.00069, 0.00361, 0.0091]^\top$
x_0	50	$[500, 125, 0]^\top$	$[100, 100, 0, 0]^\top$
Δt	0.5	0.1	1.0
T	500	25	100
T_{obs}	500	5	100
n_{obs}	25	30	20
n_{sims}	5	5	5
σ	0	0	5.0
c	0.95	0.95	0.95
ℓ	0.01	0.01	0.01
$\hat{\Lambda}_\phi(\theta_\star)$	0.238791	0.874083	0.98693
$\hat{\Lambda}_\phi(\theta_\star)^L$	0.228791	0.864083	0.97693
$\hat{\Lambda}_\phi(\theta_\star)^U$	0.248791	0.884083	0.99693
n	6979	4223	685
$\tilde{\Lambda}_\phi(\theta_\star)$	0.258822	0.99333	0.98804
$\tilde{\Lambda}_\phi(\theta_\star)^L$	0.248822	0.98	0.97804
$\tilde{\Lambda}_\phi(\theta_\star)^U$	0.268822	1.00	0.99804
$\tilde{n}$	7366	148	566
N	200	200	200
M	5	20	20

Table 7. Variables & Parameters in the MF-ABC-(SMC)^2 Case Studies.

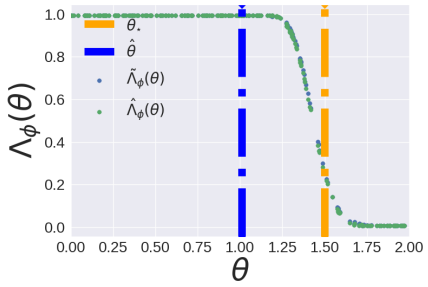
rameter space to a class of models that satisfy the property Θ_ϕ, and another that does not $\Theta_{\neg\phi}$. In Figure 10 we present the approximate parameter classification regions generated running SMC using samples from the prior probability distribution $\theta \sim \pi(\theta)$.

By performing the parameter classification on the prior, we understand how the parametrised model satisfies the property across the whole parameter space and the credibility calculation can be calculated using this approximate parameter classification. Performing a parameter classification using SVMs with samples generated from the posterior is not possible instead, as all the parameters have corresponding satisfaction probabilities that make their corresponding models satisfy the property: there is no decision boundary for the SVM to learn and thus it cannot partition the parameter regions. In Figure 11, we see that the posterior lies entirely within Θ_ϕ. So whilst running SMC over the posterior can be sufficient to compute the overall credibility $\mathcal{C}_{mf}$, it might not shed further light on the overall properties of the model set, which might be relevant when data varies with time, or when the posterior dis-

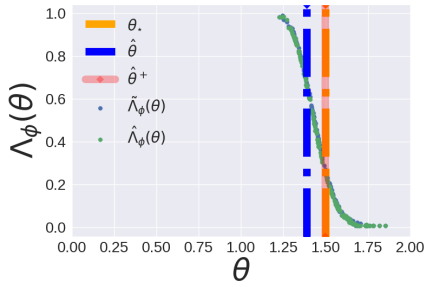


(a) Histogram for $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$.

(b) $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$ weights.

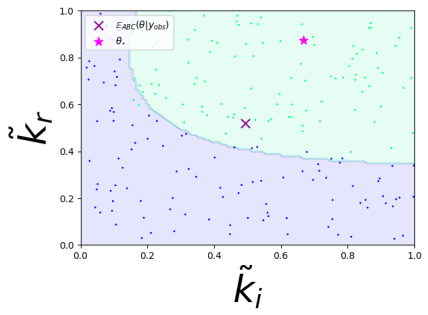


(c) $\pi(\theta)$.

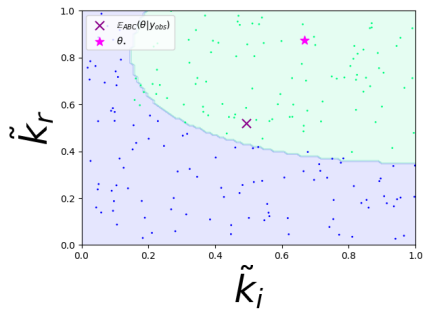


(d) $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$.

Figure 9. Inference results (including negative weights) (a) and weights (b) for sampled parameters $\theta \sim \pi^{h_M}_{ABC}(\theta \mid y_{obs})$ as well as estimated satisfaction probabilities obtained for both high-fidelity ($\hat{\Lambda}_\phi(\theta)$, denoted by blue points) and low-fidelity models ($\tilde{\Lambda}_\phi(\theta)$, denoted by green points), using samples from (c) $\pi(\theta)$ and (d) $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$. We highlight the "true parameter" ($\theta_\star = 1.5$), the posterior mean ($\hat{\theta} = 1.3992$) and the posterior mean considering only positive weights ($\hat{\theta}^+ = 1.5188$).



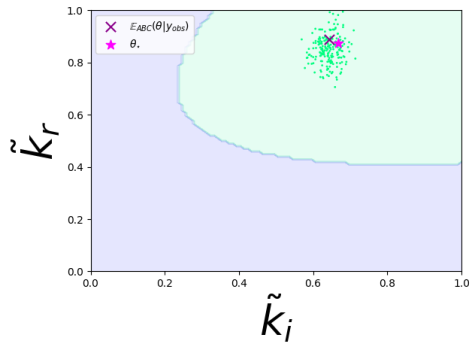
(a) Param. synthesis for $\mathcal{M}_\theta$.



(b) Param. synthesis for $\tilde{\mathcal{M}}_\theta$.

Figure 10. Parameter classification regions generated via SVMs, using estimated satisfaction probabilities from the prior, $\pi(\theta)$ for the high-fidelity model $\mathcal{M}_\theta$ (a), and for the low-fidelity model $\tilde{\mathcal{M}}_\theta$ (b).

tribution corresponds to models that are close to the probability threshold of the specification of interest.



(a) Param. synthesis for $\mathcal{M}_\theta$.

Figure 11. Parameter classification regions generated via SVMs, using estimated satisfaction probabilities $\hat{\Lambda}_\phi(\theta)$ from the prior $\pi(\theta)$, and where the (green) sample points are generated from the posterior $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$.

Statistical relationship between HF and LF models

From Figure 7, we can see that there is a difference in the estimated satisfaction probabilities for the HF and LF model - clearly there are regions in the parameter space where the LF model is a better approximation to the HF model. In this section we make an initial attempt as to how we could analyse the relation between HF and LF models using a well chosen metric in the MF-BSMC algorithm. We consider the metric introduced in [53] and presented here.

Definition 8 (Approximate Abstraction) *A model, $\tilde{\mathcal{M}}_\theta$, is a γ-approximate abstraction of $\mathcal{M}_\theta$ up to a level $1-\varepsilon$ if the following condition is satisfied:*

$$P(d_T(y, \tilde{y}) \leq \gamma) \geq 1 - \varepsilon, \tag{19}$$

where y and $\tilde{y}$ are simulations generated from $\mathcal{M}_\theta$ and $\tilde{\mathcal{M}}_\theta$ respectively, and $d_T(\cdot,\cdot)$ is a distance metric over a time-horizon with terminal time T chosen at the discretion of the user. Here we choose the metric

$$d_T(y, \tilde{y}) = \sup_{t \in [t_0, T]} \|y(t) - \tilde{y}(t)\|, \tag{20}$$

where t_0 is the initial time, T the horizon and $\tilde{y}(t)$ ($y(t)$) is the state of the low-fidelity model (high-fidelity model) at time t.

We note that the distance metric $d_T(\cdot,\cdot)$ is chosen independently of the distance metric $d(\cdot,\cdot)$ employed for the ABC schemes. Here we use this metric to explore regions of the parameter space Θ where $\tilde{\mathcal{M}}_\theta$ is a good approximation of $\mathcal{M}_\theta$, and we

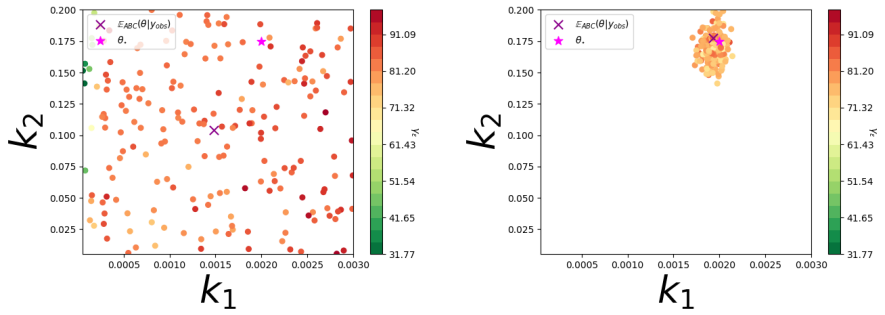
note that the choice for $d_T(\cdot,\cdot)$ is more appropriate for statistical model checking[53]. The quality γ of a low-fidelity model $\tilde{\mathcal{M}}_\theta$ as an approximate abstraction of $\mathcal{M}_\theta$ up to a level of $1-\varepsilon$ can be assessed by solving the following optimisation problem [53]

$$\min_{\gamma\in\mathbb{R}} \gamma, \text{ subject to } P(d_T(y,\tilde{y}) \leq \gamma) \geq 1-\varepsilon.$$

Let γ_ε be obtained from the equation above, then $\tilde{\mathcal{M}}_\theta$ is a γ_ε-approximate abstraction of $\mathcal{M}_\theta$ up to level $1-\varepsilon$. Computing γ_ε can be done using the randomised approach in [53], which provides an estimate of $\gamma_\varepsilon \approx \hat{\gamma}_\varepsilon$ with approximation guarantees.

Using this approximate abstraction metric and the resulting $\hat{\gamma}_\varepsilon$, in Figure 12 we quantify the difference over the parameter space between the LF model and the HF model. The results presented in Figure 12 are obtained by running MF-BSMC on parameter samples generated from the prior, $\theta \sim \pi(\theta)$ (Figure 12a) and samples generated from the posterior, $\theta \sim \pi^{h_M}_{ABC}(\theta \mid y_{obs})$ (Figure 12b).

Future work would involve exploring alternative choices of metric for $\hat{\gamma}_\epsilon$ that would allow one to perform verification on the LF model that can be formally translated to the HF model, in addition to exploring alternative distance metrics for both the inference and approximate abstraction schemes.



(a) $\pi(\theta)$. **(b)** $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$.

Figure 12. Estimated γ_ε values for LF and HF model, where θ is sampled from prior $\pi(\theta)$ (a) and posterior $\pi^{h_M}_{ABC}(\theta \mid y_{obs})$ (b) .

Acknowledgements
Gareth W. Molyneux acknowledges funding from the University of Oxford and the EPSRC & BBSRC Centre for Doctoral Training in Synthetic Biology (grant EP/L016494/1).

Author details
Department of Computer Science, University of Oxford, Oxford, UK.

References

1. Peherstorfer, B., Willcox, K., Gunzburger, M.: Survey of multifidelity methods in uncertainty propagation, inference, and optimization. Siam Review **60**(3), 550–591 (2018)
2. Molyneux, G.W., Abate, A.: ABC(SMC)^2: Simultaneous inference and model checking of chemical reaction networks. In: Abate, A., Petrov, T., Wolf, V. (eds.) Computational Methods in Systems Biology, pp. 255–279. Springer, Cham (2020)
3. Prescott, T.P., Baker, R.E.: Multifidelity approximate Bayesian computation. SIAM/ASA Journal on Uncertainty Quantification **8**(1), 114–138 (2020)
4. Prescott, T.P., Baker, R.E.: Multifidelity approximate Bayesian computation with sequential Monte Carlo parameter sampling. arXiv preprint arXiv:2001.06256 (2020)
5. Molyneux, G.W., Wijesuriya, V.B., Abate, A.: Bayesian verification of chemical reaction networks. arXiv preprint arXiv:2004.11321, SASB19 (2020). 2004.11321
6. Schnoerr, D., Sanguinetti, G., Grima, R.: Approximation and inference methods for stochastic biochemical kinetics: a tutorial review. Journal of Physics A: Mathematical and Theoretical **50**(9), 093001 (2017)
7. Wilkinson, D.J.: Stochastic Modelling for Systems Biology, Second Edition. Chapman & Hall/CRC Mathematical and Computational Biology. Taylor & Francis, (2011). https://books.google.co.uk/books?id=G3BaHtBrW68C
8. Broemeling, L.D.: Bayesian Inference for Stochastic Processes. CRC Press, (2017). https://books.google.co.uk/books?id=rtZCDwAAQBAJ
9. Sisson, S.A., Fan, Y., Beaumont, M.: Handbook of Approximate Bayesian Computation. Chapman and Hall/CRC, (2018)
10. Toni, T., Welch, D., Strelkowa, N., Ipsen, A., Stumpf, M.P.: Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. Journal of the Royal Society Interface **6**(31), 187–202 (2008)
11. Jha, S.K., Clarke, E.M., Langmead, C.J., Legay, A., Platzer, A., Zuliani, P.: A Bayesian approach to model checking biological systems. In: Degano, P., Gorrieri, R. (eds.) Computational Methods in Systems Biology, pp. 218–234. Springer, Berlin, Heidelberg (2009)
12. Zuliani, P., Platzer, A., Clarke, E.M.: Bayesian statistical model checking with application to simulink/stateflow verification. In: Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control, pp. 243–252 (2010)
13. Giles, M.B.: Multilevel Monte Carlo path simulation. Operations research **56**(3), 607–617 (2008)
14. Giles, M.B.: Multilevel Monte Carlo methods. Acta Numerica **24**, 259–328 (2015). doi:10.1017/S096249291500001X
15. Warne, D.J., Baker, R.E., Simpson, M.J.: Multilevel rejection sampling for approximate bayesian computation. Computational Statistics & Data Analysis **124**, 71–86 (2018)
16. Haesaert, S., den Hof, P.M.J.V., Abate, A.: Data-driven and model-based verification: a Bayesian identification approach. CoRR **abs/1509.03347** (2015). 1509.03347
17. Polgreen, E., Wijesuriya, V.B., Haesaert, S., Abate, A.: Data-efficient Bayesian verification of parametric Markov chains. In: Quantitative Evaluation of Systems - 13th International Conference, QEST 2016, Quebec City, QC, Canada, August 23-25, 2016, Proceedings, pp. 35–51 (2016). doi:10.1007/978-3-319-43425-4_3. https://doi.org/10.1007/978-3-319-43425-4_3
18. Polgreen, E., Wijesuriya, V.B., Haesaert, S., Abate, A.: Automated experiment design for data-efficient verification of parametric Markov decision processes. In: Quantitative Evaluation of Systems - 14th International Conference, QEST 2017, Berlin, Germany, September 5-7, 2017, Proceedings, pp. 259–274 (2017). doi:10.1007/978-3-319-66335-7_16. https://doi.org/10.1007/978-3-319-66335-7_16
19. Kwiatkowska, M., Norman, G., Parker, D.: Stochastic model checking. In: Bernardo, M., Hillston, J. (eds.) Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation (SFM'07). LNCS (Tutorial Volume), vol. 4486, pp. 220–270. Springer, (2007)
20. Kwiatkowska, M., Thachuk, C.: Probabilistic model checking for biology. In: Software Safety and Security. NATO Science for Peace and Security Series - D: Information and Communication Security. IOS Press, (2014)
21. Zuliani, P.: Statistical model checking for biological applications. International Journal on Software Tools for Technology Transfer **17**(4), 527–536 (2015). doi:10.1007/s10009-014-0343-0
22. Ceska, M., Dannenberg, F., Paoletti, N., Kwiatkowska, M., Brim, L.: Precise parameter synthesis for stochastic biochemical systems. Acta Inf. **54**(6), 589–623 (2014). doi:10.1007/s00236-016-0265-2
23. Ceska, M., Pilar, P., Paoletti, N., Brim, L., Kwiatkowska, M.Z.: PRISM-PSY: precise GPU-accelerated parameter synthesis for stochastic systems. In: Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings, pp. 367–384 (2016). doi:10.1007/978-3-662-49674-9_21. https://doi.org/10.1007/978-3-662-49674-9_21
24. Bortolussi, L., Milios, D., Sanguinetti, G.: Smoothed model checking for uncertain continuous-time Markov chains. Inf. Comput. **247**(C), 235–253 (2016). doi:10.1016/j.ic.2016.01.004
25. Bortolussi, L., Sanguinetti, G.: Learning and designing stochastic processes from logical constraints. In: International Conference on Quantitative Evaluation of Systems, pp. 89–105 (2013). Springer
26. Bortolussi, L., Silvetti, S.: Bayesian statistical parameter synthesis for linear temporal properties of stochastic models. In: Beyer, D., Huisman, M. (eds.) Tools and Algorithms for the Construction and Analysis of Systems,

pp. 396–413. Springer, Cham (2018)
27. Bentriou, M., Ballarini, P., Cournède, P.-H.: Reachability design through approximate Bayesian computation. In: International Conference on Computational Methods in Systems Biology, pp. 207–223 (2019). Springer
28. Bentriou, M., Ballarini, P., Cournède, P.-H.: Automaton-abc: A statistical method to estimate the probability of spatio-temporal properties for parametric markov population models. Theoretical Computer Science **893**, 191–219 (2021). doi:10.1016/j.tcs.2021.09.039
29. Ballarini, P., Barbot, B., Duflot, M., Haddad, S., Pekergin, N.: Hasl: A new approach for performance evaluation and model checking from concepts to experimentation. Performance Evaluation **90**, 53–77 (2015). doi:10.1016/j.peva.2015.04.003
30. Karlin, S., Taylor, H.E., Taylor, H.M., Taylor, H.M., Collection, K.M.R.: A First Course in Stochastic Processes vol. v. 1. Elsevier Science, (1975). https://books.google.co.uk/books?id=jkcTT07imDYC
31. Øksendal, B.: Stochastic Differential Equations: An Introduction with Applications. Hochschultext / Universitext. Springer, (2003). https://books.google.co.uk/books?id=kXw9hB4EEpUC
32. Kloeden, P.E., Platen, E.: Numerical Solution of Stochastic Differential Equations vol. 23. Springer, (2013)
33. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry **81**(25), 2340–2361 (1977). doi:10.1021/j100540a008. https://doi.org/10.1021/j100540a008
34. Gillespie, D.T.: The chemical Langevin equation. The Journal of Chemical Physics **113**(1), 297–306 (2000)
35. Higham, D.J.: Modeling and simulating chemical reactions. SIAM Review **50**(2), 347–368 (2008)
36. Maruyama, G.: Continuous markov processes and stochastic equations. Rendiconti del Circolo Matematico di Palermo **4**(1), 48 (1955)
37. Aziz, A., Sanwal, K., Singhal, V., Brayton, R.: Verifying continuous time Markov chains. In: Alur, R., Henzinger, T.A. (eds.) Computer Aided Verification, pp. 269–276. Springer, Berlin, Heidelberg (1996)
38. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.-P.: Model checking continuous-time markov chains by transient analysis. In: Emerson, E.A., Sistla, A.P. (eds.) Computer Aided Verification, pp. 358–372. Springer, Berlin, Heidelberg (2000)
39. Baier, C., Katoen, J.: Principles of Model Checking. MIT Press, (2008)
40. Davis, M.H.: Piecewise-deterministic markov processes: A general class of non-diffusion stochastic models. Journal of the Royal Statistical Society: Series B (Methodological) **46**(3), 353–376 (1984)
41. Sisson, S., Fan, Y., Beaumont, M.: Overview of abc. Handbook of Approximate Bayesian Computation, 3–54 (2018)
42. Prangle, D.: Summary statistics in approximate Bayesian computation. arXiv preprint arXiv:1512.05633 (2015)
43. Toni, T., Stumpf, M.P.: Simulation-based model selection for dynamical systems in systems and population biology. Bioinformatics **26**(1), 104–110 (2010)
44. Agha, G., Palmskog, K.: A survey of statistical model checking. ACM Trans. Model. Comput. Simul. **28**(1), 6–1639 (2018). doi:10.1145/3158668
45. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
46. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and computing **14**(3), 199–222 (2004)
47. Murphy, K.P.: Machine Learning - a Probabilistic Perspective. Adaptive computation and machine learning series. MIT Press, (2012)
48. Jegourel, C., Sun, J., Dong, J.S.: On the sequential Massart algorithm for statistical model checking. In: International Symposium on Leveraging Applications of Formal Methods, pp. 287–304 (2018). Springer
49. Durrett, R.: Probability: Theory and Examples vol. 49. Cambridge university press, (2019)
50. Bornn, L., Pillai, N.S., Smith, A., Woodard, D.: The use of a single pseudo-sample in approximate bayesian computation. Statistics and Computing **27**(3), 583–590 (2017)
51. Prangle, D.: Lazy ABC. Statistics and Computing **26**(1-2), 171–185 (2016)
52. Christen, J.A., Fox, C.: Markov chain monte carlo using an approximation. Journal of Computational and Graphical statistics **14**(4), 795–810 (2005)
53. Abate, A., Prandini, M.: Approximate abstractions of stochastic systems: A randomized method. In: 2011 50th IEEE Conference on Decision and Control and European Control Conference, pp. 4861–4866 (2011). IEEE
54. Del Moral, P., Doucet, A., Jasra, A.: An adaptive sequential Monte Carlo method for approximate Bayesian computation. Statistics and Computing **22**(5), 1009–1020 (2012)
55. Bonassi, F.V., West, M., *et al.*: Sequential Monte Carlo with adaptive weights for approximate Bayesian computation. Bayesian Analysis **10**(1), 171–187 (2015)
56. Filippi, S., Barnes, C.P., Cornebise, J., Stumpf, M.P.: On optimality of kernels for approximate Bayesian computation using sequential Monte Carlo. Statistical applications in genetics and molecular biology **12**(1), 87–107
57. Prangle, D., *et al.*: Adapting the ABC distance function. Bayesian Analysis **12**(1), 289–309 (2017)
58. Jegourel, C., Sun, J., Dong, J.S.: Sequential schemes for frequentist estimation of properties in statistical model checking. In: International Conference on Quantitative Evaluation of Systems, pp. 333–350 (2017). Springer
59. MICHAELIS, L.: Die kinetik der invertinwirkung. Biochem. Z. **49**, 333–369 (1913)
60. Qian, H., Elson, E.L.: Single-molecule enzymology: stochastic michaelis–menten kinetics. Biophysical chemistry **101**, 565–576 (2002)
61. Zhou, Y., Liepe, J., Sheng, X., Stumpf, M.P., Barnes, C.: GPU accelerated biochemical network simulation. Bioinformatics **27**(6), 874–876 (2011)
62. Jagiella, N., Rickert, D., Theis, F.J., Hasenauer, J.: Parallelization and high-performance computing enables automated statistical inference of multi-scale models. Cell systems **4**(2), 194–206 (2017)
63. Andersen, J.R., Gütschow, C., Maier, A., Prestel, S.: A positive resampler for monte carlo events with negative weights. The European Physical Journal C **80**(11) (2020). doi:10.1140/epjc/s10052-020-08548-w
64. Andersen, J.R., Maier, A.: Unbiased elimination of negative weights in monte carlo samples. arXiv preprint arXiv:2109.07851 (2021)
65. Higham, D.J.: An algorithmic introduction to numerical simulation of stochastic differential equations. SIAM

review **43**(3), 525–546 (2001)
66. Gillespie, D.T.: Stochastic simulation of chemical kinetics. Annual Review of Physical Chemistry **58**(1), 35–55 (2007). doi:10.1146/annurev.physchem.58.032806.104637. PMID: 17037977. https://doi.org/10.1146/annurev.physchem.58.032806.104637