Online Likelihood-free Inference for Markov State-Space Models Using Sequential Monte Carlo

by

Tyler Verhaar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Statistics

Waterloo, Ontario, Canada, 2025

© Tyler Verhaar 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Sequential Monte Carlo (SMC) methods (a.k.a particle filters) refer to a class of algorithms used for filtering problems in non-linear state-space models (SSMs). SMC methods approximate posterior distributions over latent states by propagating and resampling particles, where each particle is associated with a weight representing its relative importance in approximating the posterior. Through iteratively updating particles and/or weights SMC methods gradually refine the particle-based approximation to reflect true posterior distribution. A key challenge in SMC methods arises when the likelihood, responsible for guiding particle weighting, is intractable. In such cases, Approximate Bayesian Computation (ABC) methods can approximate the likelihood, bypassing the need for closed-form expressions. The particle SMC method of interest in this thesis is Chopin's SMC^2 framework Chopin et al. [2013] which uses a nested SMC approach. An "outer" operates on the parameter space θ, and an "inner" particle filter estimates the likelihood for a fixed parameter θ, facilitating joint inference over the parameters and states. The framework proposed by Chopin required closed-form likelihoods and was intended for offline learning. This thesis proposes an ABC-SMC^2 algorithm for online-inference in SSMs with intractable likelihoods. Our method uses Approximate Bayesian Computation (ABC) in the inner particle filter to approximate likelihoods via an ABC kernel, thus enabling inference without closed-form observation likelihoods. To address the challenges of online learning, we introduce an adaptive ε-scheduler for dynamically selecting the ABC kernel's tolerance levels and a likelihood recalibration mechanism that retroactively refines posterior estimates using previously observed data. We validate our approach on three case studies using compartment models governed by an ODE system: a toy linear ODE system, the non-linear Lotka–Volterra equations, and a high-dimensional SEIR model with real-world covariates. In these experiments, ABC-SMC^2 outperforms fixed and adaptive ε-schedulers in terms of credible interval coverage, posterior accuracy, RMSE.

Acknowledgements

I would like to thank Dr. Jeffrey Negrea and Dr. Glen McGee for their time and for serving on the Examining Committee. I would like to thank my supervisor Dr. Samuel Wong for his inestimable mentorship over the years. Furthermore, I would like to thank the faculty and staff of the Department of Statistics and Actuarial Sciences for their support throughout my time here.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

List of Figures vii

List of Tables ix

1 Introduction 1

1.1 The Filtering Problem . . . 2

1.2 Approximate Bayesian Computation . . . 2

1.3 Joint Parameter and State Inference . . . 3

1.4 SMC^2 . . . 3

1.5 ABC-SMC^2 for Online-Inference . . . 4

2 Methodology 5

2.1 Compartment Models . . . 5

2.1.1 Continuous-Time Compartment Models . . . 6

2.1.2 Discrete-Time Compartment Models . . . 8

2.2 Markov State Space Models . . . 10

2.2.1 Markov State-Space Model Framework . . . 10
2.2.2 Compartment Models as Markov State-Space Models . . . 12
2.3 ABC-SMC^2 Algorithm . . . 14
2.3.1 Approximate Bayesian Computation (ABC) . . . 14
2.3.2 Sequential Monte Carlo (SMC) . . . 15
2.3.3 ABC-SMC^2 Algorithm . . . 20
2.4 Novel Contributions . . . 27
2.4.1 Online Learning . . . 27
2.4.2 Adaptive Epsilon Selection Strategy . . . 31
2.4.3 Likelihood Recalibration . . . 35

3 Results **39**
3.1 Toy Example . . . 39
3.1.1 Setup . . . 39
3.1.2 Experiment Parameters . . . 42
3.1.3 Results . . . 43
3.2 Lotka–Volterra . . . 48
3.2.1 Setup . . . 48
3.2.2 Experiment Parameters . . . 50
3.2.3 Results . . . 51
3.3 SEIR Model . . . 54
3.3.1 Setup . . . 54
3.3.2 Experiment Parameters . . . 57
3.3.3 Results . . . 58

4 Conclusion **63**
4.1 Limitations . . . 63
4.2 Future Directions . . . 64

References **66**

List of Figures

3.1 Left: time-series of the covariates informing flow rates τ_{ij}. $Z_{t,i}$ is the ith covariate at time t. Right: time-series of the compartments $X_t^{(i)}$ vs. time t. 42

3.2 Simulated trajectories with 95% credible intervals compared to observed data for four ε scheduling methods: constant $\varepsilon = 0.001$ (top left), constant $\varepsilon = 0.01$ (top right), Del Moral scheduler (bottom left), and the Novel method (bottom right). The dashed lines represent the true observed data $D_{1:T}$, the shaded regions indicate the 95% credible intervals for the trajectories $X_t^{(1)}$ (blue) and $X_t^{(2)}$ (orange) generated from the posterior distribution at time T. 47

3.3 Posterior densities for each parameter across 20 simulation runs using the novel adaptive method on the toy example. Each subplot corresponds to one of the parameters, showing the density estimates from each individual simulation run. The red dashed line represents the mean of the posterior estimates across all runs, the black dashed line indicates the true parameter value. 48

3.4 Left: time-series of the covariates informing flow rates τ_{ij}. $Z_{t,i}$ is the ith covariate at time t. Right: time-series of the compartments $X_t^{(i)}$ vs. time t. 51

3.5 Simulated SEIR compartment observations for a single seed. The figure illustrates the dynamics of the Susceptible (S_t), Exposed (E_t), Infected (I_t), and Recovered (R_t) compartments over time (t) under the specified experimental parameters. 59

3.6 This figure displays the effective sample size (ESS) and compartment observations over time for the SMC sampler. The black lines represent the ESS paths for different sampler attempts, with dots marking the points of collapse where ESS reaches 1, signifying sampler failure. The coloured lines on the right axis correspond to the compartment observations (S_t, E_t, I_t, R_t). 60

3.7 95% credible intervals for SEIR compartment trajectories under the $\varepsilon = 0.01$ and novel methods. Shaded regions reflect posterior uncertainty, with narrower intervals observed for the novel method. 62

List of Tables

3.1 Experimental Settings for ABC-SMC^2 Simulation 44

3.2 Mean and RMSE (in parentheses) of parameter estimates for each ε scheduling method. Columns denote the different ε scheduling approaches: two constant schedules (0.001 and 0.01), the Del Moral scheduler, and our Novel method. **Bolded** RMSE values represent the smallest RMSE across all methods. 44

3.3 Comparison of SMC sampler performance for different ε scheduling methods. The collapse rate indicates the proportion of runs that failed, that is $\text{ESS}_t = 1$ (all weight is concentrated on single particle and no new particles adding in the subsequent PMH-step). 45

3.4 Mean and credible intervals (in parentheses) of parameter estimates for the Constant (0.01) and Novel ε scheduling methods. 46

3.5 Experimental Settings for ABC-SMC^2 Simulation for the LV-Equations . . 52

3.6 Mean and RMSE (in parentheses) of parameter estimates for three ε scheduling methods: Constant (0.01), Del Moral, and Novel for the LV-equations. **Bolded** RMSE values represent the smallest RMSE across all methods. . 53

3.7 Mean and credible intervals [in square brackets] of parameter estimates for three ε scheduling methods: Constant (0.01), Del Moral, and Novel for the LV-equations. 53

3.8 True Parameters for SEIR Model by Covariate and Transition 57

3.9 This table outlines the true parameters used for simulating data in the SEIR model. Each column corresponds to a transition between compartments, while each row details a specific parameter and the covariate it is associated with in time-varying transition rates (τ). Parameters marked with a star (*) are fixed and not learned, leaving a total of 20 parameters to be estimated. The final column, "Start Time," indicates when each parameter begins to influence the simulation; before this time, the corresponding covariate is zero (hence its coefficient θ_i will have no influence on τ). 57

3.10 Experimental Settings for ABC-SMC^2 Simulation for the SEIR model. . . . 58

3.11 Mean, RMSE (in parentheses), and credible intervals [in brackets] of parameter estimates for the Constant (0.01) and Novel ε scheduling methods. **Bolded** RMSE values indicate the method-wise minimum RMSE achieved. Due to the fixed intercepts (see Table 3.8) the second subscript $\beta_{ij,1}$ does not indicate the intercept as in the previous examples. 61

Chapter 1

Introduction

State-Space Models (SSMs) provide a probabilistic framework for modelling dynamic systems. These models are defined by (1), a sequence of hidden states $\{\mathbf{X}_t : t = 0, 1, \dots\}$ representing latent dynamics of the system, and (2) observed data $\{\mathbf{D}_t : t = 1, 2, \dots\}$ which, through an observation model, are conditionally independent of each other given the hidden states [Doucet et al., 2001]. Typically, the hidden states are assumed to follow a Markov process. The following three probabilities define the structure of a SSM

$p(\mathbf{X}_0)$	Initial distribution of the hidden states
$p(\mathbf{X}_{t+1} \mid \mathbf{X}_t, \theta)$	State-transition model governing the evolution of $\mathbf{X}_t$, for $t \geq 0$
$p(\mathbf{D}_t \mid \mathbf{X}_t, \theta)$	Observation model relating hidden states to observed data, for $t \geq 1$

where θ denotes model parameters that govern both transitions ($\mathbf{X}_t \to \mathbf{X}_{t+1}$) and observations ($D_t$). This general framework allows SSMs to capture complex and uncertain dynamics underlying real-world processes. SSMs are widely used in areas such as epidemiology [Liu et al., 2020a], where they are used to model infectious disease dynamics, finance [Mergner, 2009, Zeng and Wu, 2013], where they capture latent market trends, and ecology [Allen, 2008], where they model population dynamics in stochastic environments. We now discuss an integral problem in SSMs: the filtering problem.

1.1 The Filtering Problem

The filtering problem consists of sequentially estimating $p(\mathbf{X}_t \mid \mathbf{D}_{1:t}, \theta)$, the probability distribution over the latent states at time t, given a parameter θ and observed data $\mathbf{D}_{1:t}$. This distribution is often referred to as the filtering distribution. Sequential Monte Carlo (SMC) methods, also known as particle filters, can approximate the filtering distribution given the model parameter θ [Arulampalam et al., 2002, Doucet et al., 2001]. These methods achieve this by maintaining and updating a collection of weighted particles over time through a sequence of propagation, weighting, and resampling steps. SMC methods are versatile, requiring fewer assumptions about the underlying system than traditional approaches like the Kalman filter [Kalman, 1960]. This versatility enables SMC methods to handle systems with non-linear dynamics or non-Gaussian distributions. In many practical settings, such as epidemiological models with complex transmission mechanisms [Liu et al., 2020a], the likelihood $p(\mathbf{D}_t \mid \mathbf{X}_t, \theta)$ or state-transition model $p(\mathbf{X}_{t+1} \mid \mathbf{X}_t, \theta)$ may be intractable, meaning they cannot be expressed or computed in closed form [Chopin and Papaspiliopoulos, 2020, Doucet et al., 2001]. For example, when observations depend on non-linear relationships with the states, the likelihood may be analytically or computationally intractable. In such cases, Approximate Bayesian Computation (ABC) methods, also known as likelihood-free methods, can approximate the likelihood, bypassing the need for closed-form expressions.

1.2 Approximate Bayesian Computation

Approximate Bayesian Computation (ABC) methods bypass the direct evaluation of a likelihood by using a simulation model and comparing the observed data $\mathbf{D}_t$ to simulated data $\mathbf{D}_t^{\text{sim}}$ from the model given parameters θ. This comparison between observed and simulated data is conducted via a distance metric $\rho(\cdot, \cdot)$, a summary statistic S, and a kernel function $K_\varepsilon(\cdot)$, where $\varepsilon > 0$ is the kernel tolerance. The likelihood is approximated through this comparison, which typically takes the form $K_\varepsilon\left(\rho\left(S(\mathbf{D}_t), S(\mathbf{D}_t^{\text{sim}})\right)\right)$. Such an approach can be adapted for acceptance-rejection sampling: if the distance is below ε, the proposed θ is deemed plausible. While ABC is a powerful tool for likelihood-free inference, particularly in genetics and epidemiology [Toni, 2011], combining ABC with sequential state and parameter inference is non-trivial [Moral et al., 2012]. For instance, selecting the sequence of kernel tolerances $\varepsilon_{1:t}$ is challenging, as it requires balancing approximation accuracy (achieved by taking $\varepsilon \to 0$) and computational efficiency. This trade-off has been the focus of significant research [Douc et al., 2011, Silk et al., 2012].

1.3 Joint Parameter and State Inference

Thus far, we have assumed the model parameter θ to be known. In practice, however, this is often not the case, and θ must also be inferred from data alongside the latent states, introducing a joint parameter-state inference problem. Naively, one could treat θ as part of the latent state vector and run a single high-dimensional particle filter. However, this approach often leads to particle degeneracy and computational intractability, as the dimensionality of the latent space grows with both time t and the number of parameters [Arulampalam et al., 2002], exacerbating the curse of dimensionality.

Another potential solution to the joint parameter-state inference problem is to perform offline parameter inference via standard MCMC methods. This can be achieved by constructing Markov chains that sample from the joint posterior $p(\theta, \mathbf{X}_{1:T} \mid \mathbf{D}_{1:T})$ of parameters and latent states based on all available observations. In some practical settings, such as online learning, recalibrating the entire MCMC process for a single new observation is computationally infeasible, as MCMC must reprocess all past observations to ensure accurate posterior sampling. One compromise is offered by particle MCMC (PMCMC) techniques [Andrieu and Doucet, 2010], which combine MCMC moves in the parameter space with SMC. PMCMC has theoretical guarantees of convergence to the correct posterior, but in practice, it can struggle, particularly in high-dimensional settings [Cui et al., 2016, Endo et al., 2019].

1.4 SMC^2

To address the challenges outlined in the preceding sections, [Chopin et al., 2013] proposed the SMC^2 algorithm. The SMC^2 algorithm is a nested SMC approach that allows for joint parameter θ and state $\mathbf{X}_t$ inference. It consists of two key components:

- An "outer" SMC that propagates and resamples particles in the parameter space, approximating $p(\theta \mid \mathbf{D}_{1:t})$.
- For each outer particle $\theta^{(i)}$, an "inner" particle filter approximates the latent state posterior $p(\mathbf{X}_t \mid \mathbf{D}_{1:t}, \theta^{(i)})$ and estimates the likelihood increments $p(\mathbf{D}_t \mid \mathbf{D}_{1:t-1}, \theta^{(i)})$.

This approach disentangles the high-dimensional problem of joint inference into two layers of SMC, mitigating the degeneracy that arises from naive state-space augmentation.

Moreover, by effectively recycling the likelihood estimates from the inner filters, SMC^2 updates the parameter distribution in an online fashion, processing data sequentially without resorting to re-running a full MCMC scheme at each step. However, SMC^2, as proposed in [Chopin et al., 2013], makes two assumptions that can restrict its applicability in practice. Firstly, each inner filter requires repeated evaluation of $p(\mathbf{D}_t \mid \mathbf{X}_t, \theta)$ or $p(\mathbf{X}_{t+1} \mid \mathbf{X}_t, \theta)$. In the original framework, they assume there is a closed-form likelihood and transition density. Secondly, while observations are processed sequentially by the SMC^2 algorithm, it is not designed for online learning where the choice of algorithmic hyperparameters (i.e., resampling thresholds, MCMC kernel settings) may need to be adaptive.

1.5 ABC-SMC^2 for Online-Inference

In this thesis, we present a novel online-ABC-SMC^2 algorithm for Markov SSMs that integrates Approximate Bayesian Computation within the "inner" SMC framework, enabling online inference in SSMs with intractable likelihoods, building on the work in [Hou, 2024]. Our method processes time-series data sequentially, allowing for online updates to the posterior approximation. By incorporating an ABC kernel $K_\varepsilon\left(\rho\left(S(\mathbf{D}_t), S(\mathbf{D}_t^{\text{sim}})\right)\right)$ into the inner filters, we eliminate the need for closed-form observation density, making SMC^2 applicable to complex models. We introduce an adaptive ABC-kernel ε-scheduler that tunes the ABC tolerance ε_t as new data arrive, balancing particle diversity and convergence to the true posterior. To mitigate the risk of posterior drift when tolerances change, our method includes a recalibration step that retroactively refines particle weights based on previously observed data. We benchmark our ABC-SMC^2 approach on three compartmental models governed by ordinary differential equations (ODEs): a toy linear system, the classic Lotka-Volterra predator-prey model, and a high-dimensional SEIR model incorporating real-world covariates. Results demonstrate that our algorithm outperforms both fixed and standard adaptive ε-schedulers in terms of credible interval coverage, posterior accuracy, and RMSE.

Chapter 2

Methodology

2.1 Compartment Models

In this section, we investigate a motivating example to which our ABC-SMC^2 method can be applied: compartment models. Compartment models, initially introduced by Ross [Ross, 1916, 1917, Ross and Hudson, 1917], are widely used in the mathematical modelling of infectious diseases to understand the spread and control of epidemics. However, they are not limited to this context; other examples include modelling biochemical reactions [Anderson, 2013], pulmonary gas exchange [Tsoukias and George, 1998], the dynamics of opinion spread in social networks [Toscani and Zanella, 2023], and traffic flow analysis to model vehicle movements across road networks [Daganzo, 1994]. For instance, the well-known SIR model [Beckley et al., 2013, Harko et al., 2014, Kröger and Schlickeiser, 2020, Schlickeiser and Kröger, 2021] divides a population into compartments, such as Susceptible (S), Infectious (I), and Recovered (R), with individuals progressing between compartments over time.

There are two main classes of compartment models: continuous-time and discrete-time, each with its own advantages and disadvantages. Continuous-time models are often represented with a system of ordinary differential equations (ODEs) and provide a deterministic framework well-suited for analyzing the underlying structure of system dynamics (general trends, average behaviour). In contrast, discrete-time models can capture stochastic effects, making them more realistic for modelling various phenomena but more complex to implement and analyze. We consider the continuous-time case in Section 2.1.1 and discuss the discrete-time model in Section 2.1.2.

2.1.1 Continuous-Time Compartment Models

Consider a compartmental model that describes the rates of change in a system of compartments representing different states or groups in a population. Let the N compartments at time t be denoted by

$$\mathbf{X}_t = (X_t^{(1)}, X_t^{(2)}, \ldots, X_t^{(N)}) \in \mathbb{R}^N,$$

where each $X_t^{(i)}$ represents the quantity in the i-th compartment at time t. The evolution of each compartment is governed by a set of differential equations of the form:

$$\frac{dX_t^{(i)}}{dt} = f_i(\mathbf{X}_t, t, \theta) \quad i \in [N], \tag{2.1}$$

where $[N] = \{1, 2, \ldots, N\}$ and $f_i(\mathbf{X}_t, t, \theta)$ denotes the rate of change of compartment $X_t^{(i)}$, depending on all other compartments in $\mathbf{X}_t$, time t, and the model parameters $\theta \in \mathbb{R}^d$.

We briefly remark on the structure of the f_i-functions, as it will become relevant in subsequent sections. The goal of these functions is to capture the general structure of dependencies and transitions between compartments. To achieve this, we use the following formulation:

$$f_i(\mathbf{X}_t, t, \theta) = \underbrace{\sum_{j=1}^{N} \alpha_{ji}(t, \theta) \cdot h_{ji}(\mathbf{X}_t, t)}_{\text{inflow}} - \underbrace{\sum_{k=1}^{N} \alpha_{ik}(t, \theta) \cdot h_{ik}(\mathbf{X}_t, t)}_{\text{outflow}} + g_i(\mathbf{X}_t, t, \theta), \tag{2.2}$$

where:

- $\alpha_{ji}(t, \theta) \geq 0$ is the rate of transition from compartment j to compartment i, which can depend on time t and parameter(s) θ.
- $h_{ji}(\mathbf{X}_t, t)$ is a function that captures the influence of compartment j on the rate of change of compartment i. The function h_{ji} allows for non-linear combinations of the compartments, which could include products, ratios, or other complex dependencies between compartments. For instance, one could have $h_{ji}(\mathbf{X}_t, t) = X_t^{(i)} \cdot X_t^{(j)}$.
- $g_i(\mathbf{X}_t, t, \theta)$ captures additional factors affecting compartment $X_t^{(i)}$ at time t that cannot be captured by the h_{ji}'s, such as the impact of covariates.

The ODE setup above provides a flexible framework for modelling various compartmental dynamics in systems such as epidemiological models, ecological models, and other population-based models. The ODE system provides a continuous-time index for all compartments, which can be solved numerically (via an ODE solver) to obtain a deterministic trajectory for each compartment. This procedure is generally performed as follows: let the observed data be denoted by $\mathbf{D} = \{D_{t_k}\}_{k=1}^{K}$ or $D_{t_1:t_K}$ at discrete times $t_1, \ldots, t_K$. For instance, in an epidemiological model, D_{t_k} can represent the count data of susceptible, exposed, infected, and recovered individuals at time t_k.

To connect the ODE system to these observed data, we utilize a likelihood function. At each time t_k, the ODE system produces a deterministic solution $\hat{\mathbf{X}}_t(\theta)$, which represents the predicted state of the compartments at time t_k under model parameters θ. Ideally, the predicted state $\hat{\mathbf{X}}_t(\theta)$ should agree with the observed data D_{t_k}. However, due to noisy measurements or unmodelled dynamics, there may be discrepancies, introducing uncertainty. This uncertainty is taken into account by defining a likelihood function $L(\mathbf{D}|\theta)$ given by:

$$L(\mathbf{D}|\theta) = \prod_{k=1}^{K} p(D_{t_k}|\hat{\mathbf{X}}_{t_k}(\theta)), \tag{2.3}$$

where p denotes the probability distribution of the observed data given the predicted state. The specific form of p depends on the context. For instance, if D_{t_k} represents count data, a common choice [Anderson et al., 2020, Tang and Tu, 2012] is:

$$D_{t_k} \sim \text{NegBin}\left(\hat{\mathbf{X}}_{t_k}(\theta), \sigma^2\right).$$

Additionally, if D_{t_k} is continuous data [Aitkin, 2010, Pawitan, 2001, Tanner, 1993], one could have:

$$D_{t_k} \sim \mathcal{N}\left(\hat{\mathbf{X}}_{t_k}(\theta), \sigma^2\right),$$

where in both cases σ^2 is a dispersion parameter of the measurement noise.

Now that we've established how a continuous-time compartment model is typically constructed and implemented, we remark on its two key drawbacks. The discussion here focuses on traditional deterministic compartment models based on ODEs, which omit stochastic components by design. Extensions to stochastic compartment models do exist (i.e., stochastic differential equations (SDEs) or piecewise deterministic Markov processes), but they are not considered in this context.

1. **Lack of Stochasticity**: The deterministic nature of the ODE solution limits its ability to model systems where inherent randomness affects the transitions between compartments [Gillespie, 1977, Kermack and McKendrick, 1927]. For example, in an epidemiological model, the transition rate between compartments (i.e., susceptible to infected) may depend on social behaviour, which can fluctuate due to events like public gatherings. A purely deterministic model would miss the random spikes in infection rates caused by these unpredictable gatherings, resulting in overly smooth predictions [Allen, 2008].

2. **Likelihood Over-reliance**: Without explicitly modelled stochastic dynamics, the likelihood function must account for all sources of uncertainty through the chosen observation model. This can lead to biased estimates if the assumed distribution does not effectively capture the underlying variability [McLachlan and Krishnan, 2007]. Referring back to our previous example, suppose the measurement process is assumed to follow a normal distribution centred around the predicted number of infected individuals. If the actual measurement noise is heavy-tailed (i.e., due to under-reporting or sudden outbreaks), the normal distribution assumption might lead to an underestimation of uncertainties, causing biased parameter estimates.

2.1.2 Discrete-Time Compartment Models

A popular [Allen, 2003, Barnes and Fulford, 2002, Chiang, 1980] circumvention of the two problems posed above is to discretize time by inducing difference equations from the ODE system:

$$X^{(i)}_{t_{k+1}} = X^{(i)}_{t_k} + f_i(\mathbf{X}_{t_k}, t_k, \theta)\Delta t + \varepsilon^{(i)}_{t_k} \quad \forall i \in [N], \tag{2.4}$$

where $\Delta t = t_{k+1} - t_k$ and $\varepsilon^{(i)}_{t_k}$ are perturbation noise terms.

The role of stochasticity in enhancing compartment models has been investigated in [Ball et al., 1997, Lekone and Finkenstädt, 2006]. The incorporation of the perturbation term into the model effectively addresses this. Firstly, it allows for the modelling of inherent randomness in transitions between compartments, addressing the first limitation of the purely deterministic ODE approach. Secondly, with the stochastic perturbations being incorporated directly into the system, we no longer rely solely on the observation model $L(\mathbf{D}|\theta)$ to capture all sources of variability. This alleviates the burden on the likelihood function and reduces the risk of biased estimates arising from unmodelled dynamics.

The discretized compartmental model framework offers additional computational benefits, especially when data arrives at uniform time intervals (i.e., $\Delta t = 1$), simplifying difference equations into a recursive form:

$$X_{t_{k+1}}^{(i)} = X_{t_k}^{(i)} + f_i(\mathbf{X}_{t_k}, t_k, \theta) + \varepsilon_{t_k}^{(i)}.$$

This recursive formulation is computationally advantageous as it eliminates the need for solving complex systems of equations at each time step, allows for direct computation of the system's state at t_{k+1} from t_k, facilitates iterative algorithms (particle filters, MCMC, and forward simulations), and supports efficient parallelization and memory management. Various works have explored introducing stochasticity into compartmental models. For instance, [Lekone and Finkenstädt, 2006] developed a stochastic compartmental model for the transmission of Ebola, which effectively captured stochastic transmission dynamics. However, their model did not incorporate external covariates to inform the transmission process over time, limiting its ability to reflect real-world variability influenced by external factors such as policy changes and mobility patterns.

Subsequent work in [Liu et al., 2020a] incorporated covariates into discrete-time compartmental models for COVID-19. Let a time-series of covariates be written as $\mathbf{Z} = \{Z_{t_k}\}_{k=1}^{K}$ (or $Z_{t_1:t_K}$), where each $Z_{t_k} \in \mathbb{R}^p$ represents observed covariates at discrete time points $t_1, t_2, \ldots, t_K$. In [Liu et al., 2020a], these covariates included mobility patterns and policy stringency, allowing the model to dynamically adapt to changing external conditions by modelling the transition rates $\alpha_{ij}(t, \theta)$ as functions of these covariates using log-normal transitions:

$$\alpha_{ij}(t, Z_{t_k}, \theta) \sim \mathcal{LN}\left(\beta_{ij}^T Z_{t_k}, \sigma_{ij}^2\right),$$

where $\beta_{ij} \in \mathbb{R}^p$ are the coefficients that determine the influence of the covariates on the transition rates, and σ_{ij}^2 accounts for noise, ensuring non-negativity. However, this incorporation introduces additional complexity. Specifically, achieving such integration within a traditional ODE framework often necessitates assuming deterministic dependence on covariates, which may not fully capture inherent variability. Extending the model to incorporate covariate-dependent stochastic transitions addresses this limitation, creating a more realistic and adaptive framework for modelling real-world systems. This approach results in transition probabilities between compartments that depend on both the states $\mathbf{X}(t)$ and the covariates $\mathbf{Z}(t)$, leading to a nonlinear and potentially non-Markovian system.

In this work, we adopt and build on this framework by incorporating the same type of time-series covariates $\mathbf{Z}$ into our model $\alpha_{ij}(t, \theta) = \alpha_{ij}(t, Z_t, \theta)$. This extension allows for

a comprehensive representation of dynamic transitions, incorporating both covariates and stochastic noise into the time-varying coefficients. By doing so, we effectively absorb both the perturbation term from Equation 2.4 and the g_i-term from Equation 2.2.

In their study, [Liu et al., 2020a] applied this discrete-time modelling approach over short-term horizons and linked observations to a negative binomial likelihood (2.3), estimating the posterior distribution of parameters θ using adaptive Sequential Monte Carlo (SMC) methods coupled with Hamiltonian Monte Carlo (HMC) kernels. While effective for short-term applications, their methodology was tailored to specific settings and did not fully address the complexities associated with longer time horizons or more generalised scenarios. Motivated by this, we developed a sampling method incorporating SMC that is adapted to a different modelling context.

2.2 Markov State Space Models

State-space models (SSMs) provide a probabilistic framework for systems evolving over time, characterized by sequences of hidden states, latent variables, and their relationship to observed data. In this section, we define the SSM framework used throughout this paper and show how the discrete-time compartment models introduced earlier can be mapped to this framework. This section will read slightly differently than the description given in the introduction 1, as it will simultaneously set up the framework for our computational methods.

2.2.1 Markov State-Space Model Framework

Consider a time series of observed data $\mathbf{D} = \{D_t\}_{t=1}^{T}$, where $D_t \in \mathbb{R}^M$, and a corresponding time series of covariates $\mathbf{Z} = \{Z_t\}_{t=0}^{T-1}$, where $Z_t \in \mathbb{R}^p$, representing external factors influencing system dynamics. The time index $t = 0, 1, \ldots, T$ spans the system's evolution, with T denoting the terminal time; we assume $T < \infty$.

The system's dynamics are governed by a sequence of hidden states $\mathbf{X}_{0:T} = \{\mathbf{X}_t\}_{t=0}^{T}$, where $\mathbf{X}_t \in \mathbb{R}^N$, representing the hidden state configuration at each time step. When the hidden states obey the Markov property, we may refer to the SSM as a Markov SSM. The hidden states $\mathbf{X}_t$ need not be directly observable but are linked to observed data D_t through an observation function $G : \mathbb{R}^N \to \mathbb{R}^M$, which maps the hidden state to an observable output $\mathbf{Y}_t = G(\mathbf{X}_t)$, allowing for comparisons between the modelled/simulated

and observed data. In addition, latent variables $\mathrm{V}_{0:T}$, where $\mathrm{V}_t \in \mathbb{R}^K$, encode auxiliary stochastic processes or intermediate variables influencing state transitions. The latent variable is parameterized by the model parameters $\theta \in \mathbb{R}^d$ and incorporates the covariates. This combined structure of the hidden states and latent variables facilitates more flexible modelling of systems. For clarity, we refer to the tuple $(\mathbf{X}_t, \mathrm{V}_t)$ as the augmented state of the system at time t.

The transition between hidden states from $\mathbf{X}_t$ to $\mathbf{X}_{t+1}$ is determined by a transition function $\mathbf{F} : \mathbb{R}^N \times \mathbb{R}^K \to \mathbb{R}^N$ given by

$$\mathbf{X}_{t+1} = \mathbf{F}(\mathbf{X}_t, \mathrm{V}_t),$$

which maps the augmented state $(\mathbf{X}_t, \mathrm{V}_t)$ to the next hidden state $\mathbf{X}_{t+1}$. Note that the function $\mathbf{F}$ is implicitly parameterized by θ through V_t. The corresponding probabilistic transition density is given by

$$p(\mathbf{X}_{t+1} \mid \mathbf{X}_t, Z_t, \theta) = \int \mathbb{I}\{\mathbf{F}(\mathbf{X}_t, \mathrm{V}_t) = \mathbf{X}_{t+1}\} p(\mathrm{V}_t \mid Z_t, \theta)\, d\mathrm{V}_t, \tag{2.5}$$

where $\mathbb{I}$ is an indicator function enforcing the consistency of the state transition with the model. The transition density may be intractable but can be approximated if we can simulate from the density. Intractable state transition densities that can be simulated from have been extensively investigated in [Aushev et al., 2024, Dean et al., 2014, Higgs, 2011, Jovanovski et al., 2024, Lindsten et al., 2015]. The covariates Z_t reflect the set of information available at time t. This formulation naturally satisfies the Markov property because the transition density $p(\mathbf{X}_{t+1} \mid \mathbf{X}_t, Z_t, \theta)$ depends only on the current hidden state $\mathbf{X}_t$, the covariates Z_t, and the parameters θ. Formally:

$$p(\mathbf{X}_{t+1} \mid \mathbf{X}_t, \mathrm{V}_t, Z_t, \theta) = p(\mathbf{X}_{t+1} \mid \mathbf{X}_t, Z_t, \theta),$$

implying that the future state $\mathbf{X}_{t+1}$ is conditionally independent of all past states $\mathbf{X}_{0:t-1}$ given the present state $\mathbf{X}_t$, covariates Z_t, and model parameters θ. The Markov property ensures that the future state is conditionally independent of past states given the current state and inputs. The theoretical properties of such processes in the context of SMC have been discussed at length in [Douc et al., 2011, Ghahramani and Hinton, 1996]. Lastly, errors are conditionally independent given the latent variables, as the Markov property ensures each state transition depends only on the current hidden state and latent variables, not past states or errors. Furthermore, the observation model assumes that measurement noise at each time step is independent, given the current hidden state, ensuring no dependencies between errors across time steps.

The joint probability distribution over all variables in the SSM is factorized as:

$$p(\mathbf{X}_{0:T}, \mathrm{V}_{1:T}, \mathbf{D}_{1:T} \mid \mathbf{Z}_{0:T-1}, \theta) = p(\mathbf{X}_0 \mid \theta) \prod_{t=0}^{T-1} p(\mathbf{X}_{t+1} \mid \mathbf{X}_t, \mathrm{V}_t, Z_t, \theta) p(\mathrm{V}_t \mid Z_t, \theta) \prod_{t=1}^{T} p(D_t \mid \mathbf{X}_t, \theta),$$

where:

- $p(\mathbf{X}_0 \mid \theta)$: The prior distribution over the initial state.
- $p(\mathbf{X}_{t+1} \mid \mathbf{X}_t, \mathrm{V}_t, Z_t, \theta)$: The state transition density.
- $p(\mathrm{V}_t \mid Z_t, \theta)$: The distribution of the latent variables.
- $p(D_t \mid \mathbf{X}_t, \theta)$: The observation likelihood, mapping hidden states to observed data.

The integral structure of $p(\mathbf{X}_{t+1} \mid \mathbf{X}_t, Z_t, \theta)$, due to marginalization over V_t, often renders exact inference intractable, requiring approximate methods, such as Sequential Monte Carlo (SMC), which we discuss in subsequent sections. The resulting model structure is illustrated below:

$$\begin{array}{ccccccc} (\mathbf{X}_0, V_0) & \xrightarrow{\theta, Z_0} & (\mathbf{X}_1, V_1) & \xrightarrow{\theta, Z_1} & \cdots & \xrightarrow{\theta, Z_{T-1}} & (\mathbf{X}_T, V_T) \\ & & \downarrow & & & & \downarrow \\ & & D_1 & & & & D_T \end{array}$$

This formulation is sufficiently general to encompass a wide range of systems, including Hidden Markov Models, dynamic ecological systems, stochastic financial processes, and non-linear biological systems. It provides a foundation for the Bayesian inference techniques described in subsequent sections.

2.2.2 Compartment Models as Markov State-Space Models

Here we show how the state-space model framework 2.2.1 established above can be applied to the discrete-time compartment model detailed in 2.1.2. We let $\Delta t = 1$ in Equation 2.4, resulting in time-indexing $t = 0, 1, 2, \ldots, T$. Assume we have access to time series of observed data $\mathbf{D}_{1:t} = \{D_t\}_{t=1}^{T} = D_{1:T}$, where $D_t \in \mathbb{R}^M$, and covariate data $\mathbf{Z} = \{Z_t\}_{t=0}^{T-1} = Z_{0:T-1}$, where $Z_t \in \mathbb{R}^p$. Let the state of the compartments at time t be given by

$$\mathbf{X}_t = (X_t^{(1)}, X_t^{(2)}, \ldots, X_t^{(N)}) \in \mathbb{R}^N.$$

We write the transition rates from compartment i to compartment j at time t as $\alpha_{ij}(t, Z_t, \theta)$, which are modelled using log-normal distributions as described in [Liu et al., 2020a]:

$$\alpha_{ij}(t, Z_t, \theta) \sim \mathcal{LN}\left(\beta_{ij}^T Z_t, \sigma_{ij}^2\right),$$

where $\beta_{ij} \in \mathbb{R}^p$ are the coefficients in the multiplication $\beta_{ij}^T Z_t$, drawn from a prior distribution $\beta_{ij} \sim \mathcal{N}(0, 1)$, and σ_{ij}^2 are the variances modelled by $\sigma_{ij}^2 \sim \Gamma(k_\Gamma, \theta_\Gamma)$ (shape-scale parameterization). The full parameter set is denoted $\theta = (\boldsymbol{\beta}, \boldsymbol{\sigma})$. The log-normal distribution ensures that the transition rates α_{ij} remain non-negative, aligning with the physical interpretation of compartment transitions. The choice of priors for β_{ij} helps to regularize the model and reflect prior knowledge, while the gamma distribution for σ_{ij}^2 supports modelling non-negative variances and potential heteroscedasticity.

As noted in 2.1.2, this approach embeds stochastic noise terms within the GLMs at each time t, making the compartment transitions inherently stochastic. In the context of our SSM framework above, the latent variable takes the form

$$\mathrm{V}_t = (\alpha_{11}, \alpha_{12}, \ldots, \alpha_{1N}, \alpha_{21}, \ldots, \alpha_{N1})^T.$$

The stepping equations for the ODE system are expressed as:

$$\begin{aligned} X_{t+1}^{(i)} &= X_t^{(i)} + f_i(\mathbf{X}_t, t, \theta) \quad \forall i \in [N], \\ f_i(\mathbf{X}_t, t, \theta) &= \sum_{j=1}^{N} \alpha_{ji}(t, Z_t, \theta) \cdot h_{ji}(\mathbf{X}_t, t) - \sum_{k=1}^{N} \alpha_{ik}(t, Z_t, \theta) \cdot h_{ik}(\mathbf{X}_t, t), \end{aligned}$$

which are used to construct the state transition function $\mathbf{F}$. The functions $h_{ij}(\mathbf{X}_t, t)$ represent the state-dependent interactions that influence transitions between compartments. Fix any time step t, and for shorthand write $h_{ij} = h_{ij}(\mathbf{X}_t, t)$, representing the influence of the current state $\mathbf{X}_t$ on transitions from compartment $i \to j$ at time t. We define the matrix $\mathbf{M}_t$ to capture these transition effects:

$$\mathbf{M}_t = \begin{pmatrix} 0 & -h_{12} & \cdots & -h_{1N} & 0 & h_{21} & \cdots & h_{N1} \\ -h_{21} & 0 & \cdots & -h_{2N} & h_{12} & 0 & \cdots & h_{N2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -h_{N1} & -h_{N2} & \cdots & 0 & h_{1N} & h_{2N} & \cdots & 0 \end{pmatrix}.$$

This matrix reflects the state-dependent interactions that govern transitions between compartments. We can write the state transition function $\mathbf{F}$ as follows:

$$\mathbf{X}_{t+1} = \mathbf{F}(\mathbf{X}_t, V_t) = \mathbf{X}_t + \mathbf{M}_t \mathrm{V}_t. \tag{2.6}$$

Hence, $\mathbf{X}_{t+1}$ is a linear transformation of V_t given $\mathbf{X}_t$.

Lastly, the compartment model adheres to the Markov property. Specifically, the latent variable V_t, which encapsulates the transition rates $\alpha_{ij}(t, Z_t, \theta)$, depends only on the covariates Z_t at time t and the model parameters θ. The next state $\mathbf{X}_{t+1}$ depends only on the current state $\mathbf{X}_t$ and the latent variable V_t through the state transition function $\mathbf{X}_{t+1} = \mathbf{F}(\mathbf{X}_t, \mathrm{V}_t)$. As a result, $\mathbf{X}_{t+1}$ is conditionally independent of all past states $\mathbf{X}_{0:t-1}$ given $\mathbf{X}_t$, Z_t, and θ, thereby satisfying the Markov property.

2.3 ABC-SMC^2 Algorithm

In this section, we will review approximate Bayesian computation (ABC) and sequential Monte Carlo (SMC), after which we will outline how these can be used in tandem for solving problems with intractable likelihoods, such as the one presented in 2.2.2. Finally, we will outline the ABC-SMC^2 algorithm. Throughout this section, we will introduce notation that will be used for the remainder of this paper.

2.3.1 Approximate Bayesian Computation (ABC)

Approximate Bayesian Computation (ABC) originated in the 1980s, first noted by *Rubin* [Rubin, 1984], who proposed a sampling mechanism to draw from the posterior distribution, aligning with the principles of the ABC-rejection scheme. This concept emphasized computational methods for posterior estimation, enabling the analysis of models beyond those with analytically tractable likelihoods [Sunnåker et al., 2013]. Concurrently, *Diggle and Gratton* [Diggle and Gratton, 1984] introduced a simulation-based approach for approximating the likelihood function. These early contributions highlight ABC's utility in bypassing direct likelihood evaluations, making it suitable for implicit models [Beaumont et al., 2009], such as the one described above.

The standard ABC approach goes as follows: suppose we wish to estimate the likelihood $L(\theta \mid y)$, where $y \in \mathbb{R}^n$ is some observed data and $\theta \in \mathbb{R}^d$ is the parameter. We sample parameters from the prior $\theta \sim \pi(\cdot)$ and simulate data x from some mechanism $p(\cdot \mid \theta)$ that depends on θ [Sisson et al., 2007a]. Now, we wish to measure how "close" the simulated data x is to the observed data y. To do this, we define a summary statistic $S : \mathbb{R}^n \to \mathbb{R}^p$ and choose a metric $\rho : \mathbb{R}^p \times \mathbb{R}^p \to [0, \infty)$ to measure $\rho(S(x), S(y))$ [Beaumont et al., 2009]. This

"distance" determines whether a sample θ is accepted or rejected via a "tolerance" level denoted ε, where we assume $\varepsilon \geq 0$, and the accept-reject condition is $\rho(S(x), S(y)) \leq \varepsilon$.

Under this framework, the ABC approximation of the likelihood $L(\theta \mid y)$ can be expressed as follows:

$$L_{\text{ABC}}(\theta \mid y, \varepsilon) \propto \int K_\varepsilon \left(\rho(S(x), S(y))\right) p(x \mid \theta)\, dx, \tag{2.7}$$

where $p(x \mid \theta)$ denotes the data generation mechanism based on the parameter θ, and K_ε is a kernel function [Sunnåker et al., 2013]. A well-chosen prior is crucial for ABC, focusing sampling on regions of parameter space plausible given the observed data. ABC struggles in higher dimensions, where matching summary statistics becomes harder, amplifying complexity and reducing efficiency. A critical challenge of ABC methods is the arbitrary choice of the tolerance level ε. Given the expression in 2.7 we observe there is a trade-off, smaller values of ε lead to better approximations, at the cost of lower acceptance rates, and vice versa, large ε-values will yield higher acceptance rates but at the cost of less accurate approximations which can introduce substantial bias in the estimate. This trade-off has been the subject of extensive research [Lintusaari et al., 2017, Marin et al., 2012, Sisson et al., 2007b], we will revisit this in the context of our ABC-SMC^2 algorithm in Section 2.3.

2.3.2 Sequential Monte Carlo (SMC)

Sequential Monte Carlo (SMC) methods, commonly known as particle filters, are a powerful set of algorithms used to approximate probability distributions, especially in contexts where data arrives sequentially and distributions evolve over time [Arulampalam et al., 2002, Doucet et al., 2001]. Originally coined by *Del Moral* in 1996, SMC methods emerged to solve nonlinear filtering problems in dynamic systems by representing probability distributions as a set of weighted samples, or "particles" [Gordon et al., 1993]. These methods are particularly useful in state-space models, such as Hidden Markov Models (HMMs) and nonlinear dynamical systems, where the goal is to estimate the internal states of a system based on noisy or partial observations [Doucet et al., 2006]. Unlike linear filtering techniques like the Kalman filter [Kalman, 1960], which assume Gaussian distributions, particle filters can approximate distributions of arbitrary form, making them highly versatile in real-world applications such as signal processing [Arulampalam et al., 2002, Candy, 2016], target tracking [Khan et al., 2004, Orton and Fitzgerald, 2002], and computer vision [Rao and Satyanarayana, 2013].

The fundamental concept of Sequential Monte Carlo (SMC) is to maintain a collection of N particles, where each particle corresponds to a parameter θ and represents: (1) a possible state of the system $\mathbf{X}$, and (2) an associated weight W which indicates the relative importance of each particle in approximating the posterior distribution. For the remainder of this paper we will adopt the standard SMC particle notation. In this notation, $\mathbf{X}_t^{(i)}$ represents the i-th particle at "time" t, and $W_t^{(i)}$ denotes its associated weight. The superscript i identifies individual particles within the collection at each step t. The subscript t indexes a sequential progression, which may correspond to time in dynamic systems or to iterations in general inference problems. While SMC is often applied to sequential data, it is not restricted to such settings. In cases where data is fully available, t may instead index artificial steps, such as data partitioning or tempering. We will refer to the "time"-index colloquially but it should be noted that this is a contextual term. We write the collection of particle states at time t as $\{\mathbf{X}_t^{(i)}\}_{i=1}^N$, and the corresponding weights are $\{W_t^{(i)}\}_{i=1}^N$. Lastly, the collection of parameters associated with each particle are written as $\{\theta^{(i)}\}_{i=1}^N$, the lack of a subscript t it due to the parameters being time-independent. In cases where a combined representation is needed we may write $\{(\theta^{(i)}, W_t^{(i)})\}_{i=1}^N$, in general the context will clarify the meaning of these collections and/or notations.

At each time step t, the particle weights $W_{t-1}^{(i)}$ are updated based on the new observations D_t using sequential importance sampling (SIS) [Chen et al., 2006, Doucet et al., 2001, Liu et al., 1998, MacEachern et al., 1999, Siegmund, 1976]. We will provide a brief overview of the procedure here, given new observation data D_t we seek to adjusts the weights to reflect the posterior distribution, denoted as $W_t^{(i)}$. This update process can be expressed mathematically as:

$$W_t^{(i)} \propto W_{t-1}^{(i)} \cdot \frac{p(D_t \mid \mathbf{X}_t^{(i)}, \theta^{(i)})}{q(\mathbf{X}_t^{(i)} \mid \mathbf{X}_{t-1}^{(i)}, D_t)} \tag{2.8}$$

where $p(D_t \mid \mathbf{X}_t^{(i)}, \theta^{(i)})$ is the likelihood of the new observation D_t given the particle state $\mathbf{X}_t^{(i)}$ and parameter $\theta^{(i)}$, and $q(\mathbf{X}_t^{(i)} \mid \mathbf{X}_{t-1}^{(i)}, D_t)$ is the proposal distribution used for sampling the next state of the particles. The weights are subsequently normalized so that $\sum_{i=1}^N W_t^{(i)} = 1$. It should be noted that while not the case in this paper, there are contexts in which weights are not normalized [Doucet et al., 2001].

As the simulation progresses in t, a common issue known as weight degeneracy may arise. As the simulation handles additional observations by updating weights, it may be the case that a small subset of particles retain significant weights, while the remaining particles

receive negligible weights, this phenomenon is know as particle/weight degeneracy. This results in a poor representation of the target distribution.

A common measure of particle degeneracy is the effective sample size (ESS) [Doucet et al., 2001, Liu, 2001, Liu and Chen, 1998], which is defined as:

$$\mathrm{ESS}_t\left(\{W_t^{(i)}\}_{i=1}^N\right) = \frac{\left(\sum_{i=1}^N W_t^{(i)}\right)^2}{\sum_{i=1}^N \left(W_t^{(i)}\right)^2} \tag{2.9}$$

Note that under uniform weighting (maximal weight entropy), $\mathrm{ESS}_t = N$, in contrast, if all but one particle has zero weight then $\mathrm{ESS}_t = 1$. For any collection of normalized weights $\{W_t^{(i)}\}_{i=1}^N$, ESS_t lies in the range $[1, N]$, providing an intuitive and bounded measure of weight degeneracy.

The issue of particle degeneracy is often addressed through a resampling step [Li et al., 2021] where particles with low weights are discarded, and particles with high weights are duplicated. Often an additional propagation step is used to maintain diversity within the resampled particle set. This process helps mitigate weight collapse and ensures that the set of particles provides a better representation of the posterior distribution. It is important to note that resampling alone may not fully resolve challenges associated with weight degeneracy [Hou and Wong, 2022]. While resampling reduces the variance in particle weights it can exacerbate the problem of sample impoverishment, where duplicated particles reduce the diversity of the particle set. Further, the propagation step following resampling may not always counterbalance this loss of diversity, particularly in high-dimensional parameter spaces or settings with strongly peaked posterior distributions [Finke and Thiery, 2023, Septier and Peters, 2015]. To determine when resampling should be performed one needs to measure the degree of weight degeneracy exhibited by the current collection of particles. A common technique is to monitor the effective sample size (ESS) 2.9. When the ESS falls below a specified threshold γN, with $\gamma \in (0, 1)$, resampling is triggered to prevent degeneracy and maintain an accurate representation of the target distribution [Liu and Chen, 1998].

In some SMC applications, the likelihood $p(D_t \mid \mathbf{X}_t, \theta)$ is intractable due to complex or implicit model dynamics. Instead, we propagate particles through a deterministic transition function $\mathbf{X}_t = \mathbf{F}(\mathbf{X}_{t-1}, V_t)$, and approximate the likelihood using an ABC kernel $K_\varepsilon\left(\rho\left(S(\mathbf{D}_t), S(G(\mathbf{X}_t))\right)\right)$. For notational brevity, for the remainder of this paper we will write $K_{\varepsilon_t}(\mathbf{Y}_t^{(i)}, D_t)$ or $K_{\varepsilon_t}(G(\mathbf{X}_t^{(i)}), D_t)$ in place of the kernel expression in the preceding

sentence, this notation adopt the metric's $(\cdot, \cdot)$ notation and removes the explicit dependence on the summary statistic S and metric ρ. This substitution transforms the standard SMC update equation 2.8 into the ABC-SMC form:

$$W_t^{(i)} \propto W_{t-1}^{(i)} \cdot K_{\varepsilon_t}(\mathbf{Y}_t^{(i)}, D_t),$$

enabling inference in models with intractable likelihoods. This setup is generally referred to as an ABC-SMC algorithm [Everitt and Rowińska, 2021, Molyneux and Abate, 2020, Toni, 2011], which operates as follows:

- **Initialization:** Sample $\{\theta^{(i)}\}_{i=1}^N$ from the prior distribution $\pi(\theta)$ and assign uniform initial weights $W_0^{(i)} = \frac{1}{N}$ for all particles.
- **Importance Sampling:** For each time step t, simulate $\mathbf{X}_t^{(i)}$ from the state transition model $p(\mathbf{X}_t \mid \mathbf{X}_{t-1}, \theta^{(i)})$. Update the particle weights $W_t^{(i)}$ based on the ABC criterion:
$$W_t^{(i)} \propto W_{t-1}^{(i)} \cdot K_{\varepsilon_t}(\mathbf{Y}_t^{(i)}, D_t),$$
where K_{ε_t} is the ABC kernel with tolerance ε_t and $\mathbf{Y}_t^{(i)} = G(\mathbf{X}_t^{(i)})$ represents the mapped observable state.
- **Resampling:** If $\text{ESS}_t < \gamma N$, resample particles $\{\theta^{(i)}, \mathbf{X}_t^{(i)}\}_i$ based on the current weights $\{W_t^{(i)}\}_i$ and reset the weights to $W_t^{(i)} = \frac{1}{N}, i \in [N]$ for all resampled particles.

This approach flexibly approximates posterior distributions in intractable models. By combining ABC for likelihood approximation with SMC for sequential updating and particle management, the ABC-SMC algorithm ensures that the particle set continues to represent the evolving posterior effectively. A formal description of this algorithm is provided in Algorithm 1.

In the standard ABC-SMC algorithm, each particle at time t corresponds to a single state $\mathbf{X}_t^{(i)}$ with weight $W_t^{(i)}$, determined by (1) the previous weight and (2) the fit of $\mathbf{X}_t^{(i)}$ to observations D_t, via the kernel $K_{\varepsilon_t}(\mathbf{Y}_t^{(i)}, D_t)$. While efficient, this single-state-per-particle approach can be problematic in high-dimensional or constrained state-space models [Blum et al., 2013, Marin et al., 2012]. The transition model $p(\mathbf{X}_t \mid \mathbf{X}_{t-1}, \theta)$, incorporating latent variables, is inherently stochastic, meaning samples $\mathbf{X}_t^{(i)}$ may differ significantly. Limiting each particle to a single sampled state discards other plausible transitions, hindering the ability to represent the posterior. This issue is particularly acute in ABC, where K_{ε_t} evaluates the fit of simulated outputs to the data, increasing variance,

Algorithm 1 ABC-SMC Algorithm [Toni, 2011]

Require: Prior distribution $\pi(\theta)$, state prior $p(\mathbf{X}_0|\theta)$, latent sampler $p(V_t|\theta, Z_t)$, transition function $\mathbf{F}$, observation function G, ABC-kernel K_ε, N number of particles, observations $D_{1:T}$, covariates $Z_{0:T-1}$, tolerance schedule $\varepsilon_{1:T}$

for each particle $i = 1, 2, \ldots, N$ **do**
  Sample $\theta^{(i)} \sim \pi(\theta)$
  Sample $\mathbf{X}_0^{(i)} \sim p(\mathbf{X}_0|\theta^{(i)})$
  Set $W_0^{(i)} = \frac{1}{N}$
end for
for each time $t = 1, 2, \ldots, T$ **do**
  for each particle $i = 1, 2, \ldots, N$ **do**
    Sample $V_t^{(i)} \sim p(V_t|\theta^{(i)}, Z_t)$
    Propagate $\mathbf{X}_{t+1}^{(i)} = \mathbf{F}(\mathbf{X}_t^{(i)}, V_t^{(i)})$
    Define observable state $\mathbf{Y}_{t+1}^{(i)} = G(\mathbf{X}_{t+1}^{(i)})$
    Update weight $W_{t+1}^{(i)} \propto W_t^{(i)} \cdot K_{\varepsilon_t}\left(\mathbf{Y}_{t+1}^{(i)}, D_{t+1}\right)$
  end for
  Normalize $\{W_{t+1}^{(i)}\}_{i=1}^N$ such that $\sum_{i=1}^N W_{t+1}^{(i)} = 1$
  if a resampling criterion is satisfied **then**
    Resample particles $\{\theta^{(i)}, \mathbf{X}_{t+1}^{(i)}\}_{i=1}^N$ based on weights $\{W_{t+1}^{(i)}\}_{i=1}^N$
    Reset weights $W_{t+1}^{(i)} = \frac{1}{N}$
  end if
end for
Log-likelihood estimate:

$$\log\left(\hat{p}\left(D_{1:T} \mid \theta, Z_{0:T-1}\right)\right) = \sum_{t=1}^{T} \log\left(\frac{1}{N}\sum_{i=1}^{N} W_t^{(i)}\right)$$

bias, and particle degeneracy [Marin et al., 2012, Moral et al., 2006]. Allowing particles to represent distributions over states could address these limitations.

To address these issues, we take inspiration from *Chopin's* proposed SMC^2 method [Chopin et al., 2013] and adopt an ABC-SMC^2 approach, where each parameter $\theta^{(i)}$ corresponds to multiple simulated states $\{\mathbf{X}_t^{(i,j)}\}_{j=1}^{N_i}$. Specifically, we employ an "outer-SMC" in which "outer" particles represent parameters $\{\theta^{(i)}\}_{i=1}^{N_o}$, and an "inner-SMC" where each parameter particle $\theta^{(i)}$ is associated with a collection of "inner" states $\{\mathbf{X}_t^{(i,j)}\}_{j=1}^{N_i}$. Here we use doubly-indexed superscript (i,j) to denote the ith outer-particle's jth inner-particle, we let $N_o, N_i \in \mathbb{N}$ denote the number of outer and inner particles, respectively. With respect to notation we establish a few more conventions: we denote the weights of the outer-SMC particles as upper-case $W_t^{(i)}$ and the weights of the inner-SMC particles as lower-case $w_t^{(i,j)}$. As above, when referring to a collection of particles or weights, we use $\{\cdot\}_{(\cdot)}$. For instance, $\{w_t^{(i,j)}\}_{j=1}^{N_i}$ represents the inner weights for the ith outer particle, and $\{W_t^{(i)}\}_{i=1}^{N_o}$ represents the set of all outer weights.

2.3.3 ABC-SMC^2 Algorithm

Originally proposed by [Chopin et al., 2013], the SMC^2 method for state-space models utilizes an outer SMC for the estimation of unknown parameters and an inner SMC for the propagation of hidden states, along with a Particle Markov Chain Monte Carlo (PMCMC) kernel that stochastically updates parameters by incorporating the inner SMC [Andrieu et al., 2010]. In Chopin's original formulation, the inner SMC involved a closed-form likelihood $L(\theta^{(i)} \mid x)$. However, in the context of the compartment models discussed above 2.1, as well as the broader proposed framework 2.2.1 where the transition density $p(\mathbf{X}_{t+1} \mid \mathbf{X}_t, Z_t, \theta)$ is intractable standard SMC^2 methods do not apply. As detailed above in 2.3.2, we may bypass the intractable likelihood by constructing an inner ABC-SMC. Additionally, a PMCMC kernel [Golightly and Wilkinson, 2011] is used for stochastically updating θ by incorporating the results of the inner ABC-SMC, resulting in a hybrid method ABC-PMCMC. The integration of the outer SMC for θ, the inner ABC-SMC, and the ABC-PMCMC forms the full ABC-SMC^2 algorithm [Hou, 2024] that is utilized in this paper.

The concept of an ABC-SMC^2 method was discussed by [Kerama et al., 2022]. Their work explored the use of a decreasing sequence of tolerance levels $(\varepsilon_n)_n$ where $\infty > \varepsilon_n > \varepsilon_{n+1}$ (following [Moral et al., 2012]) in the ABC kernel, resulting in a sequence of distributions $(\pi_n)_n$ that approach the true target posterior over multiple iterations. Their

work considered an offline setting and they did not fit to a time-series of observations (or covariates) which underpins the novelties in our approach.

In contrast, we present our ABC-SMC^2 framework for online inference, which accommodates the general state space framework established in 2.2.1 This approach allows for real-time parameter updates as new data becomes available, making it suitable for dynamic systems with sequentially arriving observations $\mathbf{D}_{1:T} = \{D_t\}_{t=1}^{T}$ and covariates $\mathbf{Z} = \{Z_t\}_{t=0}^{T-1}$.

We use an ABC-based bootstrap filter 2 for the inner SMC to estimate the marginal propagation log-likelihood for a given fixed parameter θ (drawn from the outer SMC process). At each time t, we sample $V_t^{(i)} \sim p(V_t \mid \theta, \mathbf{X}_t^{(i)})$ for each particle i, then we generate an updated state $\mathbf{X}_{t+1}^{(i)}$ via the transition function, $\mathbf{X}_{t+1}^{(i)} = \mathbf{F}(\mathbf{X}_t^{(i)}, V_t^{(i)})$. Now, to update our marginal propagation likelihood estimate at time t we compute the ABC kernel evaluation $K_{\varepsilon_t}(D_t \mid G(\mathbf{X}_{t+1}^{(i)})) = \hat{p}\left(D_t|\mathbf{X}_t^{(i)}, \theta\right)$, subject to a predefined tolerance ε_t, we update the particle weights $w_t^{(i)} \propto w_{t-1}^{(i)} \cdot \hat{p}\left(D_t|\mathbf{X}_t^{(i)}, \theta\right)$. The randomness of the transitions introduced by the latent variables V_t ensures the particles remain diverse, even if the weights $\{w_t^{(i)}\}_i$ concentrate on a small subset of particles. After this has been completed for $1:t$ we compute the estimated marginal propagation log-likelihood for $D_{1:t}$ given θ as

$$\log\left(\hat{p}\left(D_{1:T} \mid \theta\right)\right) = \sum_{t=1}^{T} \log\left(\frac{1}{N}\sum_{i=1}^{N} w_t^{(i)}\right)$$

A formal algorithm is written in Algorithm 2.

Following [Chopin et al., 2013], we implement a Particle Metropolis-Hastings (PMH) kernel to stochastically update the parameters θ to ensure that the parameter space is explored effectively while preserving the correct posterior distribution.

The PMH kernel proposes a new candidate parameter θ_n from a proposal distribution $H(\theta_n \mid \theta_c)$, where θ_c denotes the current parameter. For each proposal, we compute the associated log-likelihoods l_n and l_c for the new and current parameters, respectively, using the ABC-based marginal likelihood estimation (Algorithm 2). The error in the likelihood estimates introduced by ABC can distort the accuracy of the Metropolis-Hastings acceptance probability, particularly in higher-dimensional settings where the likelihood becomes less informative, potentially leading to acceptance rates that deviate from the true posterior dynamics. Afterwhich, we evaluate the log-prior probabilities p_n and p_c for

Algorithm 2 ABC Bootstrap Filter

Require: parameter θ, state prior $p(\mathbf{X}_0|\theta)$, latent sampler $p(V_t|\theta, Z_t)$, transition function $\mathbf{F}$, observation function G, ABC-kernel K_ε, N number of particles, observations $D_{1:T}$, covariates $Z_{0:T-1}$, tolerance schedule $\varepsilon_{1:T}$

for each particle $i = 1, 2, \ldots, N$ **do**
  Sample $\mathbf{X}_0^{(i)} \sim p(\mathbf{X}_0|\theta)$
  Sample $V_0^{(i)} \sim p(V_0|\theta, Z_0)$
  $\mathbf{X}_1^{(i)} = \mathbf{F}\left(\mathbf{X}_0^{(i)}, V_0^{(i)}\right)$
  Set $w_1^{(i)} \propto K_{\varepsilon_t}\left(G(\mathbf{X}_1^{(i)}), D_1\right)$
end for
normalize $\{w_1^{(i)}\}_i$
for each time $t = 1, \ldots, T-1$ **do**
  Resample $\{\mathbf{X}_t^{(i)}\}_i$ from $\{w_t^{(i)}\}_i$
  for each particle $i = 1, 2, \ldots, N$ **do**
    Sample $V_t^{(i)} \sim p\left(V_t|\theta, Z_t\right)$
    $\mathbf{X}_{t+1}^{(i)} = \mathbf{F}\left(\mathbf{X}_t^{(i)}, V_t^{(i)}\right)$
    Set $w_{t+1}^{(i)} \propto K_{\varepsilon_t}\left(G(\mathbf{X}_{t+1}^{(i)}), D_{t+1}\right)$
  end for
  normalize $\{w_{t+1}^{(i)}\}_i$
end for
Log-likelihood estimate

$$\log\left(\hat{p}\left(D_{1:T}|\theta, Z_{0:T-1}\right)\right) = \sum_{t=1}^{T} \log\left(\frac{1}{N}\sum_{i=1}^{N} w_t^{(i)}\right)$$

θ_n and θ_c, as well as the log of the proposal densities s_n and s_c. The Metropolis-Hastings acceptance probability α is given by:

$$\alpha = \exp\left\{(l_n - l_c) + (p_n - p_c) + (s_n - s_c)\right\}.$$

A uniform random variable $u \sim \text{Uni}(0, 1)$ is sampled, and the candidate θ_n is accepted if $u < \alpha$; otherwise we reject and continue with θ_c as the current parameter. This procedure is formalized in Algorithm 3.

Algorithm 3 ABC Particle Metropolis-Hastings Kernel

Require: Propagation kernel H, current parameter θ_c, inputs to Algorithm 2.

Propagate new parameters $\theta_n \sim H(\theta \mid \theta_c)$
$p_n = \log p(\theta_n)$
$l_n =$ run 2 with θ_n
$s_n = \log H(\theta_c \mid \theta_n)$
$p_c = \log p(\theta_c)$
$l_c =$ run 2 with θ_c
$s_c = \log H(\theta_n \mid \theta_c)$
$\alpha = \exp\left\{(l_n - l_c) + (p_n - p_c) + (s_n - s_c)\right\}$
Sample $u \sim \text{Uni}(0, 1)$
if $u < \alpha$ **then**
    $\theta = \theta_n$
else
    $\theta = \theta_c$
end if
return θ

The complete ABC-SMC^2 algorithm incorporates both the inner ABC-SMC and the Particle Metropolis-Hastings (PMH) kernel to manage and update parameters θ based on sequential observations $D_{1:T}$. The general procedure is as follows: define a set of N_0 initial parameter samples $\{\theta^{(i)}\}_{i=1}^{M}$ drawn from the prior distribution $p(\theta)$. Each parameter sample is associated with a collection of N_i inner particles, representing potential state trajectories $\{\mathbf{X}_t^{(i,j)}\}_{j=1}^{N}$ for each parameter $\theta^{(i)}$.

For each observation at time t we propagate each outer particle $\theta^{(i)}$'s inner state $\{\mathbf{X}_t^{(i,j)}\}_{j=1}^{N_i}$ and estimate the marginal propagation likelihood using the t time-step of the

inner ABC bootstrap filter algorithm (Algorithm 2) to compute the weighted marginal likelihood estimate:

$$\hat{p}_{\varepsilon_t}(\mathbf{Y}_t^{(i)} \mid \mathbf{Y}_{t-1}^{(i)}, \theta^{(i)}) = \frac{1}{N_i} \sum_{j=1}^{N_i} w_t^{(i,j)},$$

where $w_t^{(i,j)}$ represents the weight of each inner particle at time t, calculated based on the ABC kernel $K_{\varepsilon_t}(\mathbf{Y}_t^{(i,j)}, D_t)$. We then update the weight of each outer particle $\theta^{(i)}$:

$$W_t^{(i)} \propto W_{t-1}^{(i)} \cdot \hat{p}_{\varepsilon_t}(\mathbf{Y}_t^{(i)} \mid \mathbf{Y}_{t-1}^{(i)}, \theta^{(i)}),$$

and update the log-likelihood for each particle:

$$\ell_t^{(i)} = \ell_{t-1}^{(i)} + \log\Big(\hat{p}_{\varepsilon_t}(\mathbf{Y}_t^{(i)} \mid \mathbf{Y}_{t-1}^{(i)}, \theta^{(i)})\Big).$$

Next, we compute the effective sample size (ESS_t) to check particle diversity, if $\text{ESS}_t < \gamma N_o$, we perform resampling of the outer particles $\{\theta^{(i)}\}_{i=1}^{N_o}$ using the weights $\{W_t^{(i)}\}_{i=1}^{N_o}$. The resampling threshold γ is a common choice in the Sequential Monte Carlo (SMC) literature, typically set as a constant proportion of the particle count (γN_o). It is a hyperparameter that can be tuned based on the specific context to balance computational cost and particle diversity. The resampled particles are then updated using the Particle Metropolis-Hastings (PMH) kernel (Algorithm 3), which proposes new parameter values $\theta^{(i)}$ for each particle to ensure exploration of the parameter space. After resampling, the weights are reset uniformly:

$$W_t^{(i)} = \frac{1}{N_o}, \quad \forall i \in \{1, \ldots, N_o\}.$$

This iterative process continues for each time step, $t = 1, \ldots, T$ finalizing with approximate posterior samples $\{\theta^{(i)}\}_{i=1}^{N_o}$ and log-likelihoods $\{\ell_T^{(i)}\}_{i=1}^{N_o}$. The complete algorithm is given in Algorithm 4.

We briefly remark on the theoretical structure of the target distribution, which we write as π_t of ABC-SMC^2 algorithm presented above. We adapt the structure given in [Chopin et al., 2013] (Proposition 1) into a suitable form.

To do so, we now introduce some additional notation following [Chopin et al., 2013] for writing out the target density π_t. We consider a single parameter θ with N_i inner states $\mathbf{X}_t^{(j)}$ at time t. Let $a_t^{(j)} \in [N_i]$ denote the ancestor particle index of the jth inner particle at time t. For instance, if $N_i = 3$ and the inner particle indices $(1, 2, 3)$ at time $t = 0$ are

Algorithm 4 ABC-SMC^2 Algorithm

Require: N_o number of outer particles, N_i number of inner particles, resampling criterion, observations $D_{1:T}$, covariates $Z_{0:T-1}$, all inputs to Algorithms 3

for each outer particle $i = 1, 2, \ldots, N_o$ **do**
  Sample $\theta^{(i)} \sim p(\theta)$
  Sample $\mathbf{X}_0^{(i,j)} \sim p\left(\mathbf{X}_0|\theta\right)$ for $j \in [N_i]$
  Sample $V_0^{(i,j)} \sim p\left(V_0^{(i)}|\theta, Z_0\right)$ for $j \in [N_i]$
  $\mathbf{X}_1^{(i,j)} = \mathbf{F}\left(\mathbf{X}_0^{(i,j)}, V_0^{(i,j)}\right)$ for $j \in [N_i]$
  $w_1^{(i,j)} \propto K_{\varepsilon_1}\left(G\left(\mathbf{X}_1^{(i,j)}\right), D_1\right)$ for $j \in [N_i]$
  Normalize $\{w_1^{(i,j)}\}_j$
end for
for $t = 1, 2, \ldots, T-1$ **do**
  for each outer particle $i = 1, 2, \ldots, N_o$ **do**
    Resample $\{\mathbf{X}_t^{(i,j)}\}_j$ from $\{w_t^{(i,j)}\}_j$
    Sample $V_t^{(i,j)} \sim p\left(V_t^{(i)}|\theta, Z_t\right)$ for $j \in [N_i]$
    $\mathbf{X}_{t+1}^{(i,j)} = \mathbf{F}\left(\mathbf{X}_t^{(i,j)}, V_t^{(i,j)}\right)$ for $j \in [N_i]$
    $w_{t+1}^{(i,j)} \propto K_{\varepsilon_t}\left(G\left(\mathbf{X}_{t+1}^{(i,j)}\right), D_{t+1}\right)$ for $j \in [N_i]$
    Normalize $\{w_t^{(i,j)}\}_j$
    Set $\hat{p}\left(\mathbf{X}_{t+1}^{(i)}|\mathbf{X}_t^{(i)}, \theta\right) = N_i^{-1} \sum_{j=1}^{N_i} w_{t+1}^{(i,j)}$
    $W_{t+1}^{(i)} \propto W_t^{(i)} \cdot \hat{p}\left(\mathbf{X}_{t+1}^{(i)}|\mathbf{X}_t^{(i)}, \theta\right)$
    $\ell_{t+1}^{(i)} = \ell_{t+1}^{(i)} + \log \hat{p}\left(\mathbf{X}_{t+1}^{(i)}|\mathbf{X}_t^{(i)}, \theta\right)$
  end for
  Normalize $\{W_{t+1}^{(i)}\}_i$
  if resampling criterion satisfied **then**
    Resample $\{\theta^{(i)}\}$ from $\{W_{t+1}^{(i)}\}_i$
    for each outer particle $i = 1, 2, \ldots, N_o$ **do**
      Call Algorithm 3 with $\theta^{(i)}, D_{1:t+1}, Z_{0:t}, \varepsilon_{1:t+1}$
    end for
    $W_{t+1}^{(i)} = 1/N_o$ for $i \in [N_o]$
  end if
end for

resampled at time $t = 1$, yielding $(2, 3, 2)$ then $a_1^{(1)} = 1, a_1^{(i,2)} = 2, a_1^{(i,3)} = 3$ and furthermore $a_2^{(i,1)} = 2, a_2^{(i,2)} = 3, a_2^{(i,3)} = 2$. Using the $a_t^{(j)}$'s we can construct an index history for each $\mathbf{X}_t^{(j)}$. Let $\mathcal{H}_t^{(j)} = (h_t^{(j)}(1), \ldots, h_t^{(j)}(t))$ denote the index history defined by $h_t^{(j)}(t) = n$ and $h_t^{(j)}(s) = a_s^{(h_t^{(j)}(s+1))}$ for $s \in [t-1]$. Similarly, let $\mathcal{X}_{1:t}^{(j)} = (\mathbf{X}_{1:t}^{(j)}(1), \ldots, \mathbf{X}_{1:t}^{(j)}(t))$ denote the state trajectory of the inner particle $\mathbf{X}_t^{(j)}$ defined by $\mathcal{X}_{1:t}^{(j)} = \mathbf{X}_s^{h_t^{(j)}}, s \in [t]$. Finally, let $q_{t,\theta}(\cdot|\mathbf{X}_{t-1}^{(i)})$ be the transition density conditional on $\mathbf{X}_{t-1}^{(i)}$ and parameter θ. Chopin shows in [Chopin et al., 2013] (Proposition 1) that the probability density π_t can be written as follows

$$\begin{aligned}\pi_t\left(\theta, \mathbf{X}_{1:t}^{1:N_i}, a_{1:t-1}^{1:N_i}\right) =& p\left(\theta \mid D_{1:t}\right) \cdot \left[\tfrac{1}{N_i} \sum_{j=1}^{N_i} p\big(\mathcal{X}_{1:t}^{(j)} \,|\, \theta,\ D_{1:t}\big)\right]\ N_i^{-(t-1)} \\ &\times \left(\prod_{\substack{k=1 \\ k \neq H_t^{(j)}(1)}}^{N_i} q_{1,\theta}(x_1^{(i)})\right) \cdot \left(\prod_{s=1}^{t} \prod_{\substack{k=1 \\ k \neq H_t^{(j)}(s)}}^{N_i} w_{s-1}^{(a_{s-1}^i)} q_{s,\theta}\left(\mathbf{X}_s^{(i)} | \mathbf{X}_{s-1}^{(a_{s-1}^i)}\right)\right)\end{aligned} \tag{2.10}$$

The first term is posterior in θ, note the marginal distribution of π_t w.r.t. θ is $p(\theta|D_{1:t})$. The second term encodes the "mixture over genealogical paths," acknowledging that each trajectory $\mathbf{X}_{1:t}^{(j)}$ could be the actual realized path with probability $1/N_i$, and that each time step's ancestor is (theoretically) chosen uniformly from the previous generation, the $N_i^{-(t-1)}$ appears due to the repeated resampling steps over times $1, 2, \ldots, t-1$. The third term represents the initial proposal/transition probabilities for all particles that are not on the chosen genealogical path. Lastly, the final term ensures that all particles (including those not on the chosen genealogical path) are properly accounted for in the transition-and-weight factors. The index $k = H_t^{(j)}(s)$ is excluded because its contribution is already accounted for in the second term.

Chopin [Chopin et al., 2013] (see also [Andrieu and Doucet, 2010, Fearnhead et al., 2008]) observes that SMC^2 remains valid even when one or more densities in the state-space model (e.g. the transition or observation densities) are *intractable*. Sufficient conditions are (1) one can simulate from the distribution in question, which facilitates the inner-particle propagation is feasible and (2) an unbiased estimator of the corresponding likelihood (or incremental weight) is available. Therefore, our ABC-SMC^2 methods may inherit the theoretical guarantees outlined in [Chopin et al., 2013] (Propositions 1 & 2).

2.4 Novel Contributions

In this section, we will outline what we believe are novel contributions to the literature.

2.4.1 Online Learning

The time-series nature of our data $D_{1:T}$ and covariates $Z_{1:T}$ introduces unique challenges for traditional ABC-SMC^2 methods. Other implementations of ABC-SMC^2-like algorithms [Kerama et al., 2022] do not fit to a time-series in an online fashion. Rather, these methods use a sequence of decreasing tolerance levels $\varepsilon_1 > \varepsilon_2 > \cdots > \varepsilon_T$ (where t does not denote 'time' but rather the iterations) to iteratively refine posterior approximations. However, these are not designed for scenarios where data arrives incrementally.

In contrast, our proposed ABC-SMC^2 algorithm supports online inference, where observations arrive sequentially. This adaptation offers a significant computational advantage: by updating particle weights $(\theta^{(i)}, W_t^{(i)})$ incrementally via the inner-SMC, we avoid recomputing weights by considering the entire time series $D_{1:t}$ at each step. Consequently, the computational cost scales linearly with the outer-SMC process, allowing for more efficient real-time analysis.

Moreover, the flexibility of the online-learning approach allows for the incorporation of new covariates during the time period $t = 1, \ldots, T$. When new covariates are introduced, the corresponding parameters can be sampled from the prior distribution $\pi(\theta)$ at time t, while the remaining parameters retain their more informed positions in the posterior distribution.

This online-learning framework introduces a subtle yet significant novelty in our algorithm, which we explore further in the following subsections. Each particle $(\theta^{(i)}, W_t^{(i)})$ accumulates its log-likelihood estimate $\ell_t^{(i)}$ at each time step t based on the inner ABC-SMC kernel's evaluation of the observed data D_t. Specifically:

$$\ell_t^{(i)} = \ell_{t-1}^{(i)} + \log\left(\hat{p}(\mathbf{Y}_t^{(i)} \mid \mathbf{Y}_{t-1}^{(i)}, \theta^{(i)})\right), \tag{2.11}$$

where $\hat{p}(\mathbf{Y}_t^{(i)} \mid \mathbf{Y}_{t-1}^{(i)}, \theta^{(i)})$ is computed using the kernel $K_{\varepsilon_t}(\cdot, \cdot)$ at time t.

Notably, this introduces a dependency of the log-likelihood $\ell_t^{(i)}$ on the sequence of tolerance levels, denoted $\varepsilon_{1:t} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_t)$. This dependency means that the log-likelihood

at time t reflects not just the current tolerance ε_t but the entire history of tolerance levels used up to that point. This dependency on the entire history is not present in the outer-SMC weights, which are updated as follows:

$$W_t^{(i)} \propto W_{t-1}^{(i)} \cdot \hat{p}(\mathbf{Y}_t^{(i)} \mid \mathbf{Y}_{t-1}^{(i)}, \theta^{(i)}), \tag{2.12}$$

where $W_{t-1}^{(i)}$ is fixed at time t, and $\hat{p}(\mathbf{Y}_t^{(i)} \mid \mathbf{Y}_{t-1}^{(i)}, \theta^{(i)})$ depends only on ε_t.

These two observations; the log-likelihood dependence on the entire ε-history and the outer-SMC weights dependence only on the current ε_t—form the basis for our additional novelties detailed in the following subsections. Before we discuss these, we will explore the underlying motivation by examining the behaviour of the ABC-SMC^2 algorithm in an online-learning context.

When fitting a time series using our ABC-SMC^2 method, two features of time-series data pose challenges for the particle system, potentially leading to weight degeneracy.

Firstly, time-series data can often exhibit regime changes, abrupt shifts in behaviour at specific time points. A regime change occurs when the observed data D_t at time t reflects a significant shift in the underlying process (relative to $D_{t-k:t-1}$ for some $k > 1$. Said shifts effectively "push" the posterior distribution in a new direction, requiring parameter values to adjust accordingly. As a result, potentially large subsets of the particles which previously fit the data well may no longer align with the new behaviour, leading to many particles receiving negligible weights.

Secondly, the data generating process for a time-series $D_{1:T}$ may depend on parameters that only begin to influence the observations at certain time points $t \geq \tau$, where $\tau > 0$ is some time index. For example, consider a coefficient in a generalized linear model (GLM) that is multiplied by a covariate Z_t which is zero for the first $\tau - 1$ time steps. Such parameters do not contribute informatively to the posterior until their associated covariates are non-zero or relevant data becomes available. To accommodate this, we allow for the dynamic introduction of parameters into the model at the times they become relevant. When a subset of "new" parameters $\theta_N = (\theta_{i_1}, \ldots, \theta_{i_k})^T$ is first introduced at time τ, its posterior $\pi(\theta_N \mid D_t)$ initially resembles the prior $\pi(\theta_N)$ due to the lack of informative observations. Meanwhile, sufficiently "learned" parameters $\theta_L = (\theta_{j_1}, \ldots, \theta_{j_p})^T$, which have been iteratively updated from earlier time steps will be more concentrated in regions of the parameter space informed by the data, reflecting a tightened posterior distribution. As a result, the introduction of new parameters can lead to situations where only particles with favourable prior samples for $\theta_{i_1}, \ldots, \theta_{i_k}$ achieve a good fit to the observed data.

The ramifications of both scenarios described above are similar: a small subset of particles fitting the data well cause the weights to concentrate on these particles while the others receive negligible weights, hence resulting in particle degeneracy. Furthermore, in the case of the dynamic introduction of parameters, the posterior of the learned parameters $\pi(\theta_L \mid D_t)$ will be reshaped by the performance of the prior samples $\theta_{i_1}, \ldots, \theta_{i_k}$, which effectively "slice" through the posterior of previously well-learned parameters. This slicing reduces particle diversity and biases the posterior toward regions dictated by the favourable prior samples of the new parameters. This interplay tightens the posterior of θ_L in a direction dominated by the new parameters, which can skew inference and hinder exploration. In both scenarios, weight degeneracy undermines the particle set's ability to effectively represent the posterior distribution. This not only compromises the diversity of the particle system but also the overall quality of the inference, particularly in high-dimensional or dynamically evolving models.

The consequences of weight degeneracy further exacerbate these challenges Firstly, when particle degeneracy occurs, the ESS drops significantly, triggering the resampling mechanism. Resampling is intended to redistribute weights effectively, allowing the particle system to recover diversity. However, in practice, if weights are overly concentrated, resampling may disproportionately replicate these particles. This replication results in a narrow parameter space dominated by a small subset of particles, further reducing diversity. While various resampling strategies exist, such as multinomial, stratified, and residual resampling, they all remain susceptible to particle replication when the particle diversity is sufficiently low [Doucet et al., 2000, Li et al., 2021]. A potential theoretical fix involves using the subsequent PMCMC move to "spread out" the resampled particles. In practice, however, this approach is often insufficient, particularly in high-dimensional settings where PMCMC moves struggle to replenish particle diversity effectively [Andrieu and Doucet, 2010, Finke and Thiery, 2021, Naesseth et al., 2016]. Without gradient information, as is the case in ABC, the proposal distribution lacks an informed direction for exploration, further limiting its ability to improve particle diversity. This limitation reflects an inherent challenge in ABC-based methods, where the absence of posterior gradients restricts the effectiveness of resampling and PMCMC updates.

Furthermore, resampling coupled with a propagation kernel relies on defining a proposal distribution to propagate particles. Typically, said proposal distribution is parameterized by an estimated mean $\hat{\mu}$ and covariance matrix $\hat{\Sigma}$, which are often weighted (as is the case in our method) to incorporate additional information regarding the quality of the particles in terms of their fit to the data [Del Moral et al., 2006]. However, under particle degeneracy, $\hat{\Sigma}$ is estimated from a small subset of particles, which can lead to negligible

diagonal entries and an overly narrow covariance structure [Briol, 2018]. This narrowing confines the propagation kernel to a small region of the parameter space, inhibiting exploration and creating a feedback loop: newly generated particles remain clustered in the same narrow region, leading to low acceptance rates and perpetuating low ESS despite resampling [Filippi et al., 2013].

One could consider alternative constructions of the propagation kernel, such as using unweighted covariance estimates, a fixed covariance, or a random walk. However, these approaches have significant drawbacks. Unweighted covariance estimates fail to leverage particle quality, often resulting in less accurate proposals. Fixed covariance structures, while theoretically able to prevent degeneracy, are highly rigid and poorly suited for dynamic models where the posterior evolves over time [Del Moral et al., 2006, Tobar et al., 2015]. Additionally, in high-dimensional settings, fixed proposals tend to have abysmal acceptance rates due to the inefficiency of exploring vast parameter spaces with limited flexibility [Filippi et al., 2013]. Random walk proposals similarly struggle in high dimensions, as they lack the ability to adapt to the posterior's local geometry, leading to poor exploration and low acceptance rates [Gunawan et al., 2018].

Adaptive MCMC strategies could mitigate these issues by dynamically adjusting the proposal structure based on particle performance [Filippi et al., 2013]. However, these methods are computationally expensive and impractical within the framework of online SMC, where efficiency is critical. For instance, running multiple iterations of MCMC for each resampling step would dramatically increase computational overhead, undermining the scalability of the method for large datasets or high-dimensional parameter spaces [Andrieu and Doucet, 2010].

While alternatives, such as using pre-specified or independent proposal distributions, could address some of these challenges, they also introduce their own limitations. A pre-specified covariance matrix may fail to adapt to the evolving posterior, particularly in dynamic time-series models, where the parameter space can shift significantly over time [Schuster et al., 2017]. Similarly, independent proposals often falter in high dimensions due to their inability to capture the complex structure of the posterior effectively [Briol, 2018, Filippi et al., 2013]. Empirical evidence supports the use of weighted covariance estimates, which generally provide better performance despite their limitations, particularly when compared to unweighted or fixed proposals [Andrieu and Doucet, 2010, Tobar et al., 2015]. These approaches strike a balance between flexibility and computational efficiency, making them a practical choice for many applications, though challenges remain in ensuring robust particle diversity under degeneracy [Filippi et al., 2013].

Both of the phenomena discussed above can lead to SMC sampler collapse, which, for the remainder of this paper, we define as a scenario where (1) there is a single particle with weight $W_t^{(i)} = 1$, and (2) after the resampling and propagation steps, all particles are identical, i.e., $\forall i, j \in [N_o], \theta^{(i)} = \theta^{(j)}$.

The occurrence of a regime change in an online learning setting or the introduction of new parameters demands a more sophisticated approach to managing the effective sample size (ESS). One counterintuitive yet effective strategy is to increase the tolerance parameter ε_t during periods of low ESS to prevent particle collapse. This approach contrasts with existing methods [Kerama et al., 2022, Moral et al., 2012], as well as the primary objective of ABC-SMC^2, which is to sequentially reduce ε_t to achieve a progressively better approximation of the posterior distribution.

For example, the adaptive scheme in [Moral et al., 2012] restricts the search for the new tolerance ε_t to the interval $[0, \varepsilon_{t-1}]$, thereby enforcing the monotonicity of the sequence $(\varepsilon_t)_t$. However, if this upper bound ε_{t-1} is too restrictive in an online-learning context, it may result in a poor ESS and could ultimately lead to particle degeneracy and algorithmic collapse.

To address this, we propose relaxing the monotonicity constraint on ε_t, allowing it to increase when necessary to maintain particle diversity. This adaptive adjustment of the tolerance helps mitigate the risk of particle degeneracy, especially in scenarios with abrupt regime changes or newly introduced parameters. Details of our adaptive strategy for managing ε_t are outlined in Section 2.4.2.

However, this adjustment introduces the need for a recalibration step later in the simulation. Once the particles have settled into more stable regions of the parameter space, recalibrating ε_t ensures that the tolerance parameter can be gradually reduced again to refine the posterior approximation. This recalibration process is discussed in Section 2.4.3.

2.4.2 Adaptive Epsilon Selection Strategy

The selection of a sequence of ABC-kernel tolerances $(\varepsilon_t)_t$ has been explored extensively in the ABC-SMC literature [Albert et al., 2015, Moral et al., 2012, Silk et al., 2012]. As noted in the preceding section, the outer-SMC weights at time t functionally depend only on the current ε_t. Since the effective sample size (ESS) is determined by these weights, we can view the ESS as a function of ε_t, illustratively written as $\text{ESS}_t(\{W_t^{(i)}\}_i, \varepsilon_t$. This concept

was first introduced by [Moral et al., 2012], who sought to control the ESS by adjusting the tolerance parameter by solving the following equation for ε_t:

$$\text{ESS}_t\left(\{W_t^{(i)}\}_i, \varepsilon_t\right) = \alpha \text{ESS}_{t-1}\left(\{W_{t-1}^{(i)}\}_i, \varepsilon_{t-1}\right), \tag{2.13}$$

where $\alpha \in (0,1)$ is referred to as the "quality" index. Values of α close to 1 favour slower but more accurate convergence, while values close to 0 result in faster but potentially less accurate outcomes. At time t, the quantity ESS_{t-1} is fixed, as it has already been computed based on the weights and tolerance at the previous step. Therefore, we are solving Equation (2.13) solely as a function of ε_t, without introducing dependencies on earlier tolerances beyond those encapsulated in the fixed ESS_{t-1}.

Since ESS_t is monotone in ε_t a bisection search can be used to solve (2.13) numerically. Since ESS_t is not continuous (due to the finite sample size N_o of the outer-MC particles) this method can introduce small inaccuracies into the solution of (2.13). The method in [Moral et al., 2012] attempts to solve (2.13) at time t within the interval $\mathcal{E} = [0, \varepsilon_{t-1}]$, we will refer to $\mathcal{E}$ as the "search space" in the context of solving 2.13 numerically. If no solution exists, the algorithm sets $\varepsilon_t = \varepsilon_{t-1}$ which ensures the monotonicity of the sequence $(\varepsilon_t)_t$. Given the online-learning context of our problem and the potential effects on ESS_t, we cannot apply this approach directly, as it may lead to particle degeneracy. To mitigate this risk, we allow for increases in ε_t by extending the search space $\mathcal{E}$ to increase the likelihood of finding numerical solutions to 2.13 The remainder of this section details our approach.

To ensure the selected ε_t remains "reasonable" we define an upper bound $\varepsilon_{\max}$ such that $\varepsilon_t \leq \varepsilon_{\max}, \forall t$. This removes the dependency on ε_{t-1} and hence lifts the monotonicity constraint. One could allow for a lower bound $\varepsilon_{\min}$ as well but this can generally be set to 0. Given some initial ε_t value from a deterministic scheduler we can compute a naive ESS_t estimate under ε_t and compare it to the target value $\alpha \text{ESS}_{t-1}\left(\{W_{t-1}^{(i)}\}_i, \varepsilon_{t-1}\right)$ to determine the search space $\mathcal{E}$ The exact procedure for the adaptive epsilon selection strategy is outlined in Algorithm 5.

The use of inf and sup in the adaptive epsilon scheduler addresses the monotonic but non-continuous nature of $\text{ESS}_t\left(\{W_t^{(i)}\}_i, \varepsilon_t\right)$ due to the finite number of particles. This approach accounts for potential discontinuities more robustly than solving (2.13) via bisection alone. The characterization using inf and sup ensures that the chosen $\varepsilon_t^\star$ maintains the correct directional relationship between the target ESS and the realized $\text{ESS}_t(\varepsilon_t^\star)$.

Algorithm 5 Adaptive Epsilon Scheduler for Online-Learning

Require: Initial ε_t, quality index factor α, maximum and minimum epsilon bounds $\varepsilon_{\max}, \varepsilon_{\min}$ (generally $\varepsilon_{\min} = 0$)

Determine target ESS $\tau = \alpha \mathrm{ESS}_{t-1}\left(\{W_{t-1}^{(i)}\}_i, \varepsilon_{t-1}\right)$.

Compute baseline $\mathrm{ESS}_t\left(\{W_t^{(i)}\}_i, \varepsilon_t\right)$

if $\mathrm{ESS}_t\left(\{W_t^{(i)}\}_i, \varepsilon_t\right) < \tau$ **then**

    Define $\mathcal{E} = (\varepsilon_t, \varepsilon_{\max}]$.

    Find (via bisection) $\varepsilon_t^\star$ such that

$$\varepsilon_t^\star = \inf\left\{\varepsilon \in \mathcal{E} : \mathrm{ESS}_t\left(\{W_t^{(i)}\}_i, \varepsilon_t\right) \geq \tau\right\}.$$

else $\mathrm{ESS}_t\left(\{W_t^{(i)}\}_i, \varepsilon_t\right) \geq \tau$

    Define $\mathcal{E} = [\varepsilon_{\min}, \varepsilon_t]$.

    Find (via bisection) $\varepsilon_t^\star$ such that

$$\varepsilon_t^\star = \sup\left\{\varepsilon \in \mathcal{E} : \mathrm{ESS}_t\left(\{W_t^{(i)}\}_i, \varepsilon_t\right) \leq \tau\right\}.$$

end if

return $\varepsilon_t^\star$

We briefly comment on the computational cost of this approach. The ABC-kernel is the first algorithmic component that requires selecting ε_t; everything prior to this step can be pre-computed. In particular, the propagation of each outer-particle's inner states ($\mathbf{X}_t^{(i,j)} \to \mathbf{X}_{t+1}^{(i,j)}$) which is the most computationally intensive does not depend on ε_t, hence computing ESS_t under a given ε_t amounts to the following operations, all of which can be in parallel and/or vectorized excluding the final ESS_t. Hence, each ε_t-evaluation of ESS_t scales linearly with N_o in terms of computational cost:

$$\begin{aligned}
\forall i \in [N_o] \quad & \hat{p}(\mathbf{Y}_t^{(i)} \mid \mathbf{Y}_{t-1}^{(i)}, \theta^{(i)}) = \frac{1}{N_i} \sum_{j=1}^{N_i} K_{\varepsilon_t}(\mathbf{Y}_t^{(i,j)}, D_t) \\
\forall i \in [N_o] \quad & W_t^{(i)\star} = W_{t-1}^{(i)} \cdot \hat{p}(\mathbf{Y}_t^{(i)} \mid \mathbf{Y}_{t-1}^{(i)}, \theta^{(i)}) \\
\forall i \in [N_o] \quad & W_t^{(i)} = \frac{W_t^{(i)\star}}{\sum_{i=1}^{N_o} W_t^{(i)\star}} \\
& \text{ESS}_t\left(\{W_t^{(i)}\}_i, \varepsilon_t\right) = \left(\sum_{i=1}^{N_o} \left(W_t^{(i)}\right)^2\right)^{-1}
\end{aligned}$$

The adaptive epsilon selection strategy is integral to the success of our online learning framework. Without an adaptive approach, the algorithm would frequently collapse due to particle degeneracy, especially when new parameters are introduced or the observed data behaviour changes. Furthermore, the incremental nature of weight updates in our framework, as described in 2.3, allows for efficient adaptive epsilon adjustments without re-evaluating the likelihoods (which depend on the entire ε-history).

However, this approach does introduce some potential issues. Suppose some new data $D_{t_1:t_k}$ being learned over time steps $t_1 < \cdots < t_k$ does not fit the current parameters well. Consequently, Algorithm 5 will increase tolerance levels $\varepsilon_{t_1}, \ldots, \varepsilon_{t_k}$ to prevent particle collapse and promote exploration of the parameter space. By Equation 2.11 the log-likelihoods of the particles will be updated with these potentially large ε values. When temporarily large ε values are introduced, particles gain wider distributions that may not necessarily represent the best fit for the data specifically at $D_{t_1:t_k}$. Firstly, this results in less informative outer weights $\{W_t^{(i)}\}_i$, reducing the effectiveness of the resampling step, which relies on these weights to prioritize higher-quality particles. Secondly, the log-likelihoods $\{\ell_t^{(i)}\}_i$ used in the propagation step will struggle to differentiate between particles, as the high $\varepsilon_{t_1}, \ldots, \varepsilon_{t_k}$ values dampen the variance in their incremental contributions at each time step t (Equation 2.11). Now consider time time-step $t, t_k < t$, and suppose the particle parameters $\{\theta^{(i)}\}_i$ have "settled" into more stable regions of the parameter space. If the new data

D_t does not contain significant information with respect to the poorly fitting parameters at times $t_1, \ldots, t_k$, the parameter estimates may remain largely unchanged, perpetuating a posterior that fails to adequately fit earlier observations. By allowing for increases in ε we accept temporarily less-informative weights that prevent weight degeneracy, however, to avoid the issues outlined above additional information can be extracted from the data by retroactively tightening $\varepsilon_{t_1}, \ldots, \varepsilon_{t_k}$.

To that end, we propose a log-likelihood "recalibration step" that adjusts the outer weights $\{W_t^{(i)}\}_i$ and likelihood estimates $\{\ell_t^{(i)}\}_i$ at time t by recomputing the particle weights under a "new" epsilon history $\varepsilon_{1:t}^\star$. Intuitively this can be thought of as updating ABC-approximation of the transition density under smaller kernel tolerance values $\varepsilon_{t_p}^\star < \varepsilon_{t_p}$

$$\hat{p}_{\varepsilon_{t_p}}\left(\mathbf{Y}_{t_p}^{(i)} \mid \mathbf{Y}_{t_p-1}^{(i)}, \theta^{(i)}\right) \to \hat{p}_{\varepsilon_{t_p}^\star}\left(\mathbf{Y}_{t_p}^{(i)} \mid \mathbf{Y}_{t_p-1}^{(i)}, \theta^{(i)}\right) \quad \text{for } p \in [k]$$

The smaller ε-values make the transition density approximations more informative, increasing variance and decreasing entropy. Said "recalibration" will result in more informative outer weights $\{W_t^{(i)}\}_i$ and likelihood estimates $\{\ell_t^{(i)}\}_i$ at time t. Consequently, the resampling step will prioritize higher-quality particles, while the PMH kernel will better differentiate between "good" and "bad" parameter values, improving both the propagation and resampling steps. We formalize this idea in the next section.

2.4.3 Likelihood Recalibration

The goal of the likelihood recalibration is to derive "new" weights and likelihoods under a new (and "smaller") epsilon history $\varepsilon_{1:t}^\star$. Suppose we are at time t in our simulation and we wish to replace ε_k with a new $\varepsilon_k^\star < \varepsilon_k$ value where $k < t$. Recall the structure of the target distribution π_t at time t given in Equation 2.10, in particular note the dependence on the inner-particle ancestry $a_{1:t-1}^{1:N_i}$. Now consider the case of our ABC-SMC^2 algorithm at any time s we (1) resample inner particles $\{\mathbf{X}_s^{(i,j)}\}_{j=1}^{N_i}$ at time s based on the inner particle weights $\{w_{s-1}^{(i,j)}\}_{j=1}^{N_i}$ from the preceding iteration and (2) we assign new inner weights $w_s^{(i,j)} \propto K_\varepsilon\left(G(\mathbf{X}_s^{(i,j)},), D_s\right)$. These inner-weights are then used to approximate the transition density $\hat{p}\left(\mathbf{Y}_t^{(i)} | \mathbf{Y}_{t-1}^{(i)}\right)$ used to update the outer particle weights.

Now consider our goal of replacing ε_k with a new $\varepsilon_k^\star < \varepsilon_k$. At time k we resample the inner particles $\{\mathbf{X}_k^{(i,j)}\}_{j=1}^{N_i}$ based on weights $\{w_{k-1}^{(i,j)}\}_{j=1}^{N_i}$, propagate via the transition function, and induce new (denoted via superscript $\star$) weights $w_k^{(i,j)^\star} \propto K_{\varepsilon_k^\star}\left(G(\mathbf{X}_k^{(i,j)},), D_k\right)$

under the new $\varepsilon_k^\star$. These new weights $w_k^{(i,j)^\star}$ will then be used for resampling at time $k+1$, and hence the original inner-particle ancestry $a_{1:t-1}^{1:N_i}$ is no longer valid, and hence we are not longer targeting the correct target distribution π_t. Therefore, to ensure a valid inner-particle ancestry, and hence a correct target distribution π_t at time t we may change ε_k to $\varepsilon_k^\star$ but we must simulate the inner particles from time k to time t to obtain a valid inner-particle ancestry. This induces a new sequence $w_{k:t}^{(i,j)}$ for the ith outer particle which resulting induces a new sequence $W_{k:t}^{(i)}$ as its weight updates (Algorithm 2 line 20) depend on new inner particles. Therefore, to obtain the correct target distribution we must run lines 10-22 in Algorithm 4 from time k to time t which will yield a valid inner particle ancestry for each outer-particle, and correctly defined outer-weights $\{W_t^{(i)}\}_{i=1}^{N_o}$.

The argument above can be extended to any finite number $p < t$ ABC-tolerance values $\varepsilon_{k_i} \to \varepsilon_{k_i}^\star$ where $\{k_1, \ldots, k_p\} \subseteq [t-1], k_i < k_{i+1}, i = 1, 2, \ldots, p-1$ by simulating particles from time k_1 to time t. Note that this "recalibration" of weights under the new epsilon history $\varepsilon_{1:t}^\star$ is at most as expensive as the potential PMCMC step at time t during the resampling procedure. The reason being, in the worst case the recalibration involves simulating the N_i inner particles from time $k = 1$ to time t for N_o particles, which is equivalent to computing the log-likelihoods for each of the N_o proposal generated during the PMCMC step. The additional outer-weight calculation in the recalibration adds minimal overhead as it merely aggregates the particles inner-weights.

One consequence of simulating the inner and outer particles over long-time horizons $k : t$ is particle degeneracy as there are no resampling steps. Therefore, as $t \to T$ and for $k \ll t$ the outer particle weights $\{W_t^{(i)}\}_{i=1}^{N_o}$ may exhibit significant particle degeneracy. Allowing for resampling while recalibrating the weights would mitigate this issue but is computationally infeasible as it would effectively result in new initialization of ABC-SMC^2 algorithm with the current posterior as the prior. To circumvent this we propose a single resampling step at time $t-1$ that allows for the outer-weights to be reset to uniform $1/N_o$ preventing the resulting outer-weights $\{W_t^{(i)}\}_{i=1}^{N_o}$ at time t from exhibiting severe particle degeneracy. This makes our recalibration step doubly as expensive as a simple PMH-step, however, we can control the rate at which we perform the recalibration and hence the overall computational cost is manageable. Note this resampling step does not compromise the inner-particle ancestry and hence our target distribution π_t remains valid. We detail the recalibration step in Algorithm 6.

Note the inner-most j-loop over the inner particles can be parallelized. The remaining i-loop amounts to aggregating inner weights per outer particle, this operation, as well as the subsequent reweighting can be vectorized allowing for efficient updates. Therefore,

Algorithm 6 Epsilon History Recalibration

Require: Recalibration frequency f, dominating kernel tolerance schedule values $\eta(t)$, current epsilon history $\varepsilon_{1:t-1}$

Ensure: Recalibrated epsilon history $\varepsilon^{\star}_{1:t-1}$, inner particles and weights $\{\mathbf{X}^{(i,j)}_{t-1}\}_{i,j}, \{w^{(i,j)}_{t-1}\}_{i,j}$

if $t \mod f = 0$ **then**
  Construct $\varepsilon^{\star}_{1:t-1}$ according to

$$\varepsilon^{\star}_k = \min\{\eta(k), \varepsilon_k\}$$

  Identify starting index τ given by

$$\tau = \inf\{k : \varepsilon_k > \varepsilon^{\star}_k\}$$

  for each $k = \tau, \tau+1, \ldots, t-1$ **do**
    for each outer particle $i = 1, \ldots, N_o$ **do**
      Resample inner particles $\{\mathbf{X}^{(i,j)}_{k-1}\}_j$ based on weights $\{w^{(i,j)}_{k-1}\}_j$.
      for each inner particle $j = 1, \ldots, N_i$ **do**
        Sample $V^{(i,j)}_k \sim p\left(V_k \mid \theta^{(i)}, Z_{k-1}\right)$.
        Propagate $\mathbf{X}^{(i,j)}_k = \mathbf{F}\left(\mathbf{X}^{(i,j)}_{k-1}, V^{(i,j)}_k\right)$.
        Define $\mathbf{Y}^{(i,j)}_k = G\left(\mathbf{X}^{(i,j)}_k\right)$.
        Compute inner weight $w^{(i,j)}_k \propto K_{\varepsilon^{\star}_k}\left(\mathbf{Y}^{(i,j)}_k, D_k\right)$.
      end for
      Compute $\hat{p}_{\varepsilon^{\star}_k}(\mathbf{Y}^{(i)}_k \mid \mathbf{Y}^{(i)}_{k-1}, \theta^{(i)}) = \frac{1}{N_i}\sum_{j=1}^{N_i} w^{(i,j)}_k$.
      Update outer weight $W^{(i)}_k = W^{(i)}_{k-1} \cdot \hat{p}_{\varepsilon_t}(\mathbf{Y}^{(i)}_k \mid \mathbf{Y}^{(i)}_{k-1}, \theta^{(i)})$.
    end for
  end for
  Resample $\{\tilde{\theta}^{(i)}_{t-1}\}$ from $\{(\theta^{(i)}_{t-1}, W^{(i)}_{t-1})\}_i$
  for each outer particle $i = 1, \ldots, N_o$ **do**
    Update $\theta^{(i)}$ by running Algorithm 3 with $\tilde{\theta}^{(i)}$ as input.
    Reset outer weight $W^{(i)}_t = 1/N_o$.
  end for
end if
return $\varepsilon^{\star}_{1:t-1}, \{\mathbf{X}^{(i,j)}_{t-1}\}_{i,j}, \{w^{(i,j)}_{t-1}\}_{i,j}$

this routine is not prohibitively expensive. Furthermore, note that the weights used for resampling (line 17) are generally more informative than the outer-weights used in the main ABC-SMC^2 routine as they will have accumulated notably more updates with no resampling as $t \to T$.

The choice of the dominating scheduler η is arbitrary, we used a log-linear scheduler due to its slow convergence to its target value η_T as $t \to T$

$$\eta_{\text{log-linear}}(t) = \exp\left(\log(\varepsilon_0) + \log\left(\frac{\varepsilon_T}{\varepsilon_0}\right) \cdot \frac{t}{T}\right)$$

where η_0, η_T are hyperparameters.

The recalibration mechanism operates independently of the adaptive epsilon scheduler, which evaluates only the current iteration's ESS and weights. Retroactive adjustments to past ε values do not interfere with the adaptive scheduler's functioning. This separation allows the adaptive scheduler to optimize ε_t freely at each time step t, while the recalibration mechanism refines the ε-history for all previous steps up to $t-1$.

The choice of the recalibration frequency f is largely arbitrary and has a direct impact on the balance between computational efficiency and posterior refinement. A larger f results in less frequent updates, while a smaller f increases the computational cost due to more frequent recalibrations. An f of 1 would reduce to offline learning by recalibrating at every step which is not practical for an online-learning setting. Although the fixed f used in this work yields effective results, this choice is heuristic and may not be optimal. Future research could explore dynamic recalibration frequencies that adapt based on convergence metrics or ESS, potentially inspired by principles underlying the Del Morat scheduler. However, designing such an approach in an online learning context presents significant computational challenges and remains an open area for further investigation (see Section 4.2).

Chapter 3

Results

We present three applications of our ABC-SMC^2 algorithm involving dynamic systems governed by systems of ordinary differential equations (ODEs). The first application will serve as our "toy example" which models a a simplified two-tank flow system, characterized by a simple linear ODE system. Secondly, we apply our method to the Lotka–Volterra equations [Cooke et al., 1981, Hoppensteadt, 2006], a non-linear ODE system used to model dynamics in biological systems. Lastly, we apply our methods to the SEIR model [Harko et al., 2014] used in epidemiology to model the spread of disease throughout a population, here we also incorporate real-world covariates to capture the dynamics of infectious disease spread. Each section will proceed as follows: first the ODE system will be presented and mapped to the SSM model defined in 2.2.1, second the experimental settings will be detailed, and lastly we will examine the results.

3.1 Toy Example

3.1.1 Setup

We begin with a simple toy example involving a two-tank flow model. Consider two tanks containing $X_t^{(1)}$ and $X_t^{(2)}$ units of water at time t. Water flows from tank 1 to tank 2 at a rate given by $\alpha_{12}(t)$ (transition $1 \to 2$), and from tank 2 to tank 1 at a rate $\alpha_{21}(t)$ (transition $2 \to 1$). Additionally, tank 2 has an outflow at rate into the abyss $\alpha_{20}(t)$ (transition $2 \to 0$) (preventing a trivial ODE solution as $-dX_t^{(1)} = dX_t^{(2)}$). This system can be described by the following system of ODEs:

$$
\begin{aligned}
dX_t^{(1)} &= -\alpha_{12}(t)X_t^{(1)} + \alpha_{21}(t)X_t^{(2)} \\
dX_t^{(2)} &= \alpha_{12}(t)X_t^{(1)} - \alpha_{21}(t)X_t^{(2)} - \alpha_{20}(t)X_t^{(2)}
\end{aligned}
$$

We discretize the system of ODEs with a uniform time step $\Delta t = 1$ (this could be any unit of time, seconds, minutes, etc.), resulting in:

$$
\begin{aligned}
X_{t+1}^{(1)} &= X_t^{(1)} - \frac{X_t^{(1)}}{\tau_{12}(t)} + \frac{X_t^{(2)}}{\tau_{21}(t)} \\
X_{t+1}^{(2)} &= X_t^{(2)} + \frac{X_t^{(1)}}{\tau_{12}(t)} - \frac{X_t^{(2)}}{\tau_{21}(t)} - \frac{X_t^{(2)}}{\tau_{20}(t)}
\end{aligned}
\tag{3.1}
$$

For notational consistency inline with the latter sections, we express the flow rates using $1/\tau$, where $\alpha_{ij}(t) = 1/\tau_{ij}(t)$. The subscript ij in τ_{ij} indicates flow from compartment i to j, and τ_{20} represents the outflow from tank 2 as there is no compartment 0. As described in 2.1.2 we inform these flow rates using a time-series of covariates $Z_{0:T-1} \in \mathbb{R}^{T\times 3}$, with each flow rate associated with one specific covariate.

We fix $T = 400$ for the time-frame. The covariate time-series $Z_{0:T-1}$ (let $Z_t^{(i)}$ represent the ith covariate at time t) was generated as follows:

$$
\begin{aligned}
Z_t^{(1)} &= \begin{cases} 0 & t \le 200 \\ 0.05\sin(10\pi t) + \varepsilon_t & t > 200 \end{cases} \\
Z_t^{(2)} &= \begin{cases} 0 & t \le 200 \\ 0.2\frac{t}{T} + \sum_{s=200}^{t} \epsilon_t & t > 200 \end{cases} \\
Z_t^{(3)} &= \begin{cases} 0 & t \le 200 \\ 0.5\sin\left(\frac{10\pi t}{T}\right) + \zeta_t & t > 200 \end{cases}
\end{aligned}
$$

where $\varepsilon_t, \epsilon_t \sim \mathcal{N}(0,1)$ and $\zeta_t \sim \text{Uni}(-0.05, 0.05)$. Furthermore, we inject "shocks" into the covariate $\{Z_t^{(2)}\}_t$ for $t = 200, 225, 250$ by setting $Z_t^{(2)} = 10$. All covariates were min-max normalized to lie in $[0,1]$ so that $Z_t^{(i)} \in [0,1]$ (aside from the shocks for $Z_{(2)}$). The

covariate time-series were constructed specifically to ensure that the resulting observation series exhibits significant changes in behaviour, namely after $t = 200$ due to the coefficients' β interaction with the covariates. These shifts in the underlying observation trajectories are designed to reflect real-world time series dynamics, where shocks or structural changes in the system occur. Such changes often coincide with the introduction of new parameters into the data-generating process, driven by the influence of "new" covariates. These scenarios highlight the limitations of existing methods and static ε-schedules, which we will show can be ill-suited for online learning contexts and prone to sampler collapse under these conditions.

Inline with the notation established in 2.1.2 τ's are time-varying parameters defined as follows:

$$\begin{aligned} \log(\tau_{12}(t)) &\sim \mathcal{N}\left(\boldsymbol{\beta}_{12}^T Z_t, \sigma_{12}^2\right), \\ \log(\tau_{21}(t)) &\sim \mathcal{N}\left(\boldsymbol{\beta}_{21}^T Z_t, \sigma_{21}^2\right) \\ \log(\tau_{20}(t)) &\sim \mathcal{N}\left(\boldsymbol{\beta}_{20}^T Z_t, \sigma_{20}^2\right), \end{aligned} \tag{3.2}$$

with priors $\beta_{ij,k} \sim \mathcal{N}(0,1)$ on the mean-covariate coefficients and $\sigma_{ij} \sim \Gamma(k_\sigma, \theta_\sigma)$ (shape-scale parameterization) for the variances. We write $\beta_{ij,k}$ to mean the kth coefficient in the coefficient vector $\boldsymbol{\beta}_{ij}$ corresponding to the transition from $i \to j$ and σ_{ij}^2 to mean the variance parameter. We allow for an intercept term in the log-normal means, so there is an augmented 1-entry at the start of these covariate vectors Z_t so this operation $\boldsymbol{\beta}_{ij}^T$ is well-defined. Lastly, in this example $Z_t^{(1)}$ interacts with transition $1 \to 2$, $Z_t^{(2)}$ interacts with transition $2 \to 1$ and so on for $Z_t^{(3)}$ and transition $2 \to 0$. In the true value vector we write later we do not explicitly list the 0s corresponding to "unused" covariate information provided by $Z_t^{(i)}$. The time-series of both the covariates $Z_{0:T-1}$ as well as the observations generated from a single simulation are show in Figure 3.1.

Using the SSM notation in 2.2.1 we write the parameter θ, and the augmented latent state $(\mathbf{X}_t, V_t)$ as follows

$$\begin{aligned} \theta &= (\boldsymbol{\beta}_{12}, \sigma_{12}, \boldsymbol{\beta}_{21}, \sigma_{21}, \boldsymbol{\beta}_{20}, \sigma_{20}), \\ \mathbf{X}_t &= \left(X_t^{(1)}, X_t^{(2)}\right)^T, \\ V_t &= \left(\frac{1}{\tau_{12}(t)}, \frac{1}{\tau_{21}(t)}, \frac{1}{\tau_{20}(t)}\right)^T. \end{aligned}$$

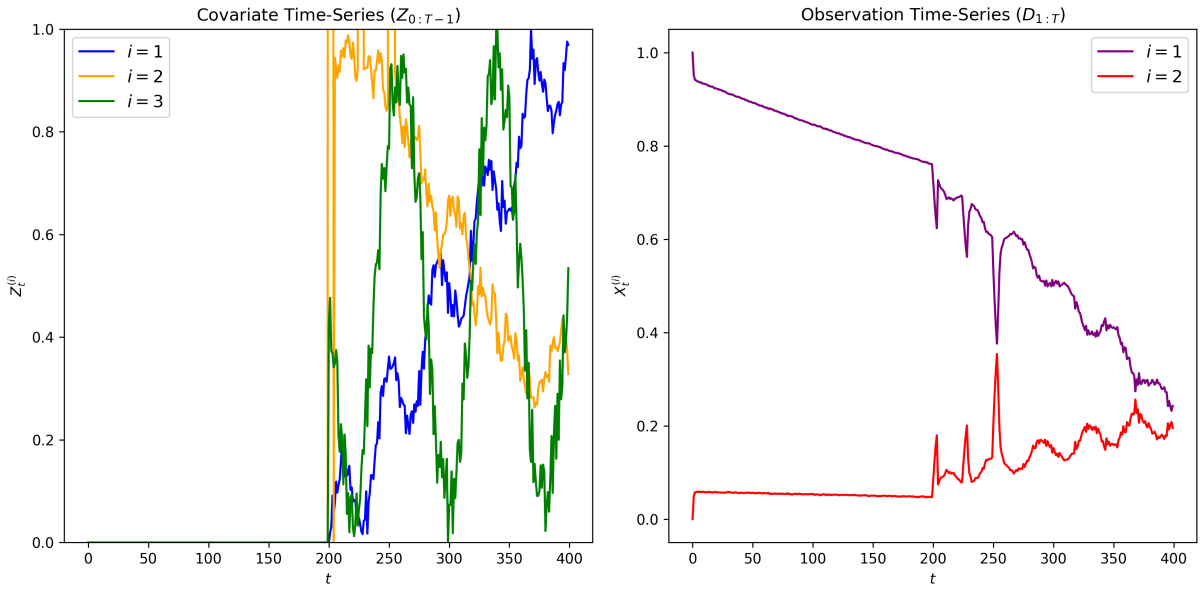


Figure 3.1: Left: time-series of the covariates informing flow rates τ_{ij}. $Z_{t,i}$ is the ith covariate at time t. Right: time-series of the compartments $X_t^{(i)}$ vs. time t.

Lastly, using the discreteized version of the ODE system given in 3.1 we may write the transition function $\mathbf{F}$ as follows

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \begin{pmatrix} -X_t^{(1)} & X_t^{(2)} & 0 \\ X_t^{(1)} & -X_t^{(2)} & X_t^{(2)} \end{pmatrix} \begin{pmatrix} \frac{1}{\tau_{12}(t)} \\ \frac{1}{\tau_{21}(t)} \\ \frac{1}{\tau_{20}(t)} \end{pmatrix} = \mathbf{X}_t + M_t V_t = \mathbf{F}\left(\mathbf{X}_t, V_t\right)$$

3.1.2 Experiment Parameters

The settings for the numerical experiment are as follows: we run 20 independent seeds with $N_o = 5,000$ outer particles and $N_i = 100$ inner particles over a time frame of $T = 400$ time steps. The compartments are initialized as $\mathbf{X}_0 = (1, 0)^T$, and the true parameter vector θ was chosen to be:

$$\begin{aligned}\theta &= (3.0, -2.5, 0.01, 0.25, 0.5, 0.01, 4.0, 1.0, 0.01)^T \\ &= (\beta_{12,1}, \beta_{12,2}, \sigma_{12}, \beta_{21,1}, \beta_{21,2}, \sigma_{21}, \beta_{20,1}, \beta_{20,2}, \sigma_{20})^T\end{aligned}$$

The log-normal mean coefficients follow standard normal priors, $\beta_{ij,k} \sim \mathcal{N}(0,1)$, and the variances are drawn from a gamma distribution, $\sigma \sim \text{Gamma}(0.2, 0.5)$. Adding to our note above, $\beta_{12,2}$ is multiplied by the covariate $Z_t^{(1)}$, we do not write would-be coefficients $\beta_{12,3}, \beta_{12,3}$ for $Z_t^{(2)}, Z_t^{(3)}$ in θ. For the ABC-kernel, we use a Gaussian kernel defined as $K_\varepsilon(x,y) = \phi(\|x-y\|_2)/\varepsilon$, where ϕ represents the density of $\mathcal{N}(0,1)$ and $\|\cdot\|_2$ is the L^2-norm. The observation function G is taken as the identity mapping, $G(\mathbf{X}_t) = \mathbf{X}_t$. The proposal kernel for the coefficients is a multivariate normal distribution ($\boldsymbol{\beta} \sim \mathcal{MVN}(\hat{\mu}_\beta, \hat{\Sigma}_\beta)$), while a log-normal distribution is used for the variances ($\sigma_k \sim \mathcal{LN}(\hat{\mu}_k, \hat{\sigma}_k)$). Here both $\hat{\mu}_\beta, \hat{\Sigma}_\beta$ are weighted estimates using the outer-particle weights following [Chopin et al., 2013]. Lastly, each variance parameter is proposed independently based on estimates $\hat{\mu}_i, \hat{\sigma}_i$ which are also weighted estimates. Note that in dimensional spaces more complex means of estimation may be required, particularly in the presence of weight degeneracy. No additional operations were done on the covariance matrix $\hat{\Sigma}_\beta$ to induce independence between coefficients effecting different transitions. The resampling threshold for the effective sample size is set at $\gamma = 0.95$. The ε-schedule varies between runs, and specific settings for comparison are detailed later. A summary of the experimental configurations is provided in Table 3.1.

A single "run" or simulation constitutes generating observations $D_{1:T}$ based on a single unique seed, and then running the SMC sampler on said observation data. Should the SMC sampler collapse, which we define as the case that $\text{ESS}_t = 1$ for some $t \in [T]$ and there are no newly accepted particles in the PMH-step we restart the SMC sampler from $t = 1$, on the same observations $D_{1:T}$.

3.1.3 Results

We compare four methods: two constant ε-schedules ($\varepsilon_t = 0.01, \varepsilon_t = 0.001$) serving as heuristic baselines, the Del Moral scheduler from [Moral et al., 2012], and our novel adaptive ε-scheduler coupled with the likelihood recalibration mechanism detailed in Section 2.4. Table 3.2 summarizes the mean and RMSE values of parameter estimates for each method.

Parameters $\beta_{12,1}, \beta_{21,1}, \beta_{20,1}$ correspond to the intercept terms of the coefficients modelling the time-varying parameters τ's mean which directly influencing the structure of the data-generating process from $t = 0$. Further, parameters $\beta_{12,2}, \beta_{21,2}, \beta_{20,2}$ correspond to the

Table 3.1: Experimental Settings for ABC-SMC^2 Simulation

Setting	**Value**
Number of seeds (runs)	20
True parameter θ	$(0.2, 0.85, 0.01, 0.2, 0.15, 0.01, 4.0, 2, 0.01)$
Priors	$\beta_{ij,k} \sim \mathcal{N}(0,1)$, $\sigma_k \sim \text{Gamma}(0.2, 0.5)$
Outer particles N_o	5,000
Inner particles N_i	100
Timeframe T	400
ABC-Kernel	$K_\varepsilon(x,y) = \phi(\|x-y\|_2)/\varepsilon$
ESS threshold γ	0.95
Proposal kernel	$\boldsymbol{\beta} \sim \mathcal{MVN}(\hat{\mu}_\beta, \hat{\Sigma}_\beta), \sigma_k \sim \mathcal{LN}(\hat{\mu}_k, \hat{\sigma}_k)$
Initial state $\mathbf{X}_0$	$(1,0)^T$
Quality Index	0.975
Recalibration Scheduler	$(\eta_0, \eta_T) = (0.01, 0.0005)$
Recalibration Frequency	20

θ	True Value	Constant (0.001)		Constant (0.01)		Del Moral		Novel	
$\beta_{12,1}$	3.000	3.009	(0.020)	2.999	(0.012)	3.000	(0.023)	3.000	(**0.007**)
$\beta_{12,2}$	-2.500	-2.507	(0.009)	-2.498	(**0.008**)	-2.504	(0.010)	-2.500	(**0.008**)
σ_{12}	0.025	0.030	(**0.006**)	0.001	(0.024)	0.026	(0.010)	0.009	(0.021)
$\beta_{21,1}$	0.250	0.261	(0.022)	0.249	(0.014)	0.252	(0.022)	0.25	(**0.007**)
$\beta_{21,2}$	0.500	0.490	(0.012)	0.499	(0.009)	0.494	(0.014)	0.499	(**0.007**)
σ_{21}	0.025	0.009	(0.018)	0.001	(0.024)	0.014	(0.016)	0.025	(**0.012**)
$\beta_{20,1}$	4.000	4.161	(0.331)	4.00	(0.005)	4.004	(0.006)	4.999	(**0.002**)
$\beta_{20,2}$	1.000	0.930	(0.202)	1.001	(0.015)	1.011	(0.078)	1.002	(**0.008**)
σ_{20}	0.025	0.447	(0.813)	0.011	(**0.014**)	0.033	(0.065)	0.001	(0.024)

Table 3.2: Mean and RMSE (in parentheses) of parameter estimates for each ε scheduling method. Columns denote the different ε scheduling approaches: two constant schedules (0.001 and 0.01), the Del Moral scheduler, and our Novel method. **Bolded** RMSE values represent the smallest RMSE across all methods.

coefficients on the covariate data in τ. Note from Figure 3.1 there is no information to be learned regarding the shape of these parameters posterior until $t > 200$, making these parameters harder to learn than the intercepts $\beta_{12,1}, \beta_{21,1}, \beta_{20,1}$. Therefore, these coefficients tend to have higher RMSE values due to the reduced observational data available to

inform their posterior distributions, as well as potentially having negligible influence on the τ values when their covariate values $Z_t^{(i)}$ are small. Lastly, $\sigma_{12}, \sigma_{21}, \sigma_{20}$ govern the variance of the time-varying parameters τ. These variance parameters are the most challenging to estimate because their influence on the time-varying parameters is indirect.

Our method achieves the best overall performance in terms of estimation accuracy, consistently attaining lowest RMSE values compared to other methods. The constant $\varepsilon = 0.001$ scheduler achieves some reasonable estimates for some parameters but suffers from a high SMC sampler collapse rate of 53.4% (see Table 3.3). The small ABC-kernel tolerance severely constrains particle diversity across all timesteps t, resulting in consistently low effective sample size (ESS) and frequent SMC sampler collapse, particularly during regime shifts in the data-generating process $D_{1:T}$ (i.e., starting at $t > 200$). Similarly, the Del Moral scheduler, despite being adaptive, also has a high collapse rate of 63.6% due to its monotone sequence of $(\varepsilon_t)_t$ values. While this gradual refinement ensures posterior accuracy in stable or low-variability data, it fails to accommodate regime shifts in the underlying observations due to its monotonic constraint which inhibits particle diversity, causing ESS degeneration and sampler collapse. These issues are more pronounced in higher-dimensional settings, where the rapid narrowing of the posterior distribution disproportionately limits exploration, compounding inference errors for new parameters while destabilizing estimates for previously learned ones.

Tag	Total Initializations	Collapse Rate
Constant (0.001)	43	0.534
Constant (0.01)	20	0.000
Del Moral	55	0.636
Novel	20	0.000

Table 3.3: Comparison of SMC sampler performance for different ε scheduling methods. The collapse rate indicates the proportion of runs that failed, that is $\text{ESS}_t = 1$ (all weight is concentrated on single particle and no new particles adding in the subsequent PMH-step).

as shown in Table 3.4, this method generates significantly wider posterior credible intervals compared to our approach. The broader ε threshold allows a larger range of particles to pass the ABC kernel, inflating posterior uncertainty. While this can produce reasonable point estimates for some parameters, it compromises the precision of the posterior distribution, reducing its utility for inference and decision-making. In contrast, our method achieves tighter credible intervals while maintaining low RMSE values.

Figure 3.2 illustrates the fit of each method to the underlying data by plotting the true

θ_i	True Value	Constant (0.01)		Novel	
$\beta_{12,1}$	3.000	2.999	(2.965, 3.033)	3.000	(2.994, 3.005)
$\beta_{12,2}$	-2.500	-2.498	(-2.527, -2.469)	-2.500	(-2.508, -2.491)
σ_{12}	0.025	0.001	(0.000, 0.008)	0.009	(0.006, 0.012)
$\beta_{21,1}$	0.250	0.249	(0.211, 0.286)	0.250	(0.244, 0.256)
$\beta_{21,2}$	0.500	0.499	(0.474, 0.525)	0.499	(0.493, 0.504)
σ_{21}	0.025	0.001	(0.000, 0.008)	0.025	(0.019, 0.034)
$\beta_{20,1}$	4.000	4.000	(3.971, 4.027)	3.999	(3.995, 4.003)
$\beta_{20,2}$	1.000	1.001	(0.915, 1.093)	1.002	(0.989, 1.015)
σ_{20}	0.025	0.011	(0.000, 0.071)	0.001	(0.000, 0.002)

Table 3.4: Mean and credible intervals (in parentheses) of parameter estimates for the Constant (0.01) and Novel ε scheduling methods.

trajectory for a single seed alongside the 95% credible intervals of trajectories generated by the posterior particles at time T. For consistency, the true trajectory shown here is the same as that in Figure 3.1. The $\varepsilon = 0.01$ scheduler produces the widest credible intervals, reflecting the increased posterior variance associated with its less restrictive ABC-kernel tolerance which weakens the alignment with the observed data. In contrast, the $\varepsilon = 0.001$ scheduler generates narrower credible intervals due to its stringent tolerance, which enforces stricter filtering, however due to its strict ε-value the learned parameters at time T do not reflect the true parameters well resulting in an inaccurate trajectory.

The Del Moral scheduler produces credible intervals that are comparable to those of our method, due to its monotone decreasing $(\varepsilon_t)_t$ sequence, which gradually tightens the posterior fit over time. However, as noted before, this method exhibits frequent sampler collapse, as shown in Table 3.3, compromising its reliability.

Our method achieves the tightest credible intervals, outperforming all other methods in balancing posterior precision and fit to the observed data. By dynamically adjusting ε_t, it maintains particle diversity during transitions and enforces a stricter posterior fit when needed. The recalibration mechanism further ensures that earlier large ε_t values do not inflate posterior variance, enabling precise and stable inference across all simulations without any sampler collapse.

Figure 3.3 presents the posterior densities for each parameter across 20 simulation runs

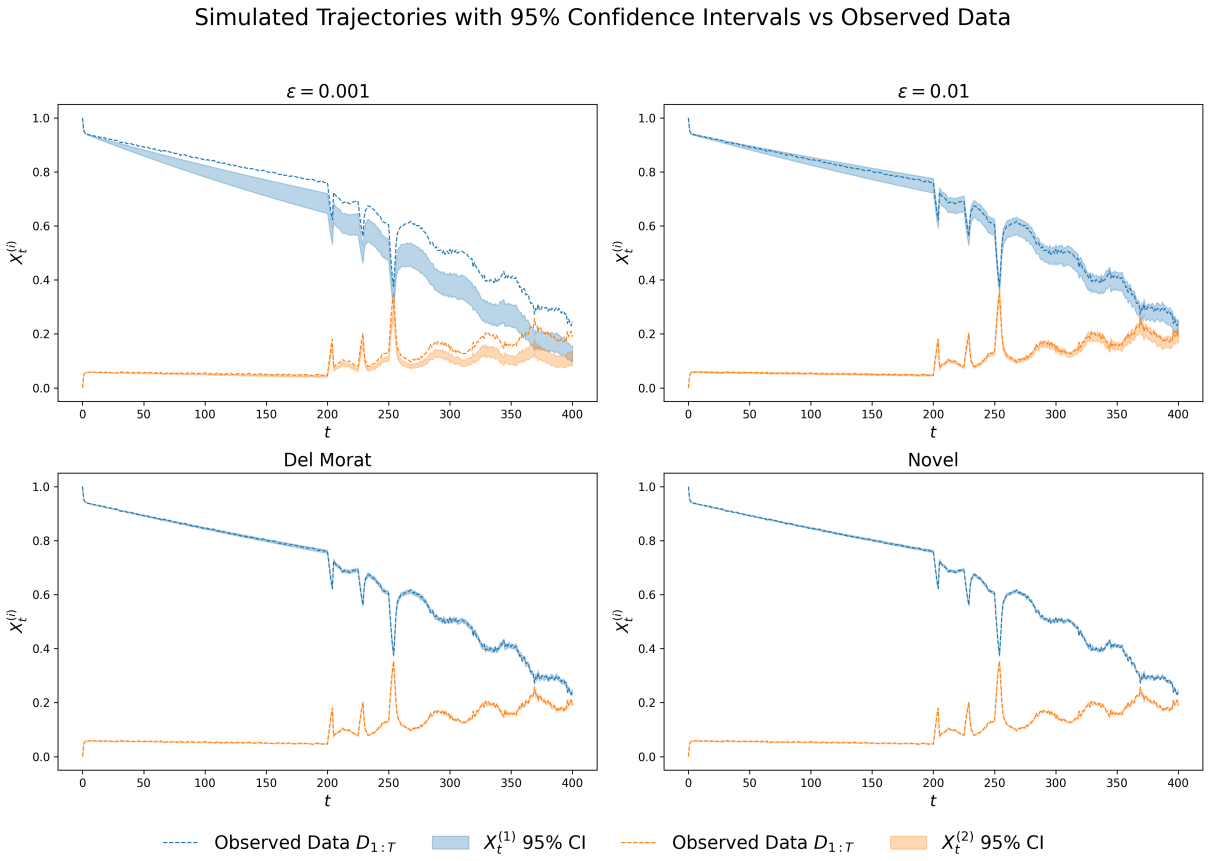


Figure 3.2: Simulated trajectories with 95% credible intervals compared to observed data for four ε scheduling methods: constant $\varepsilon = 0.001$ (top left), constant $\varepsilon = 0.01$ (top right), Del Moral scheduler (bottom left), and the Novel method (bottom right). The dashed lines represent the true observed data $D_{1:T}$, the shaded regions indicate the 95% credible intervals for the trajectories $X_t^{(1)}$ (blue) and $X_t^{(2)}$ (orange) generated from the posterior distribution at time T.

obtained using our novel adaptive method.

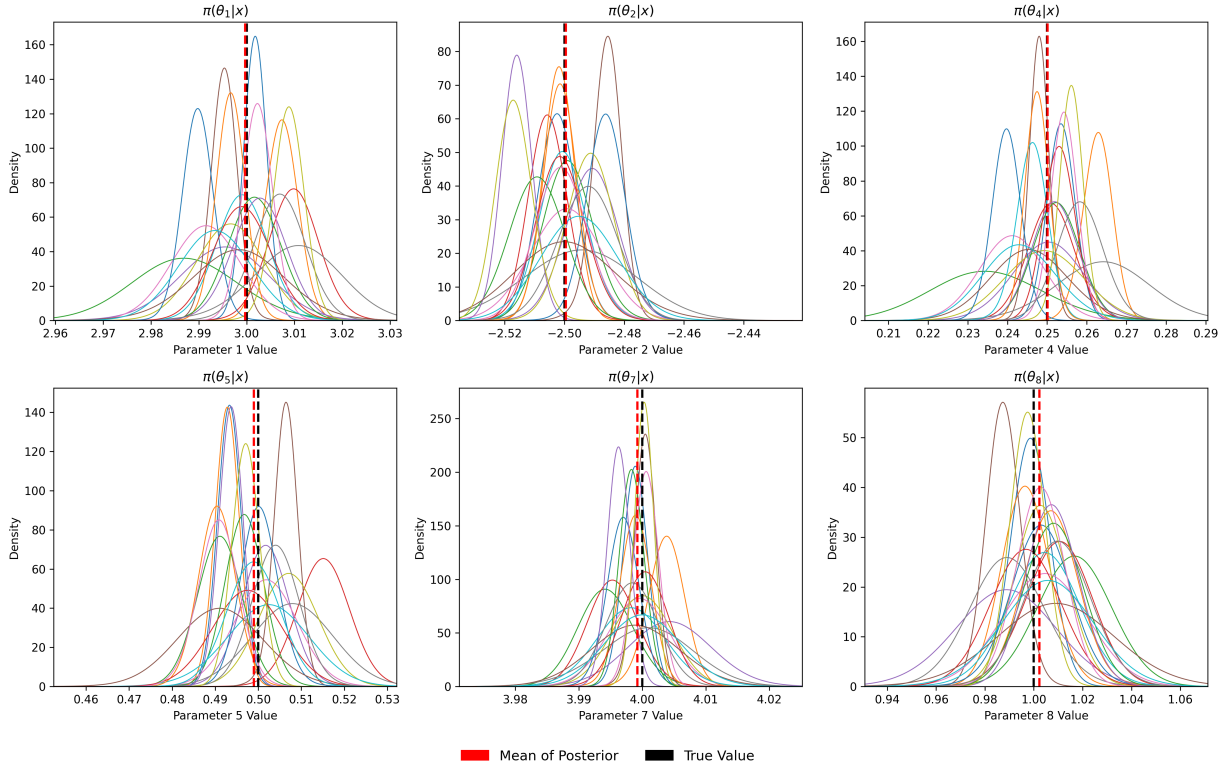


Figure 3.3: Posterior densities for each parameter across 20 simulation runs using the novel adaptive method on the toy example. Each subplot corresponds to one of the parameters, showing the density estimates from each individual simulation run. The red dashed line represents the mean of the posterior estimates across all runs, the black dashed line indicates the true parameter value.

3.2 Lotka–Volterra

3.2.1 Setup

The preceding Section 3.1.1 covered a simple linear ODE system, we now consider a more complex system known as the Lotka-Volterra predator-prey equations [Brauer and Castillo-

Chavez, 2000, Cooke et al., 1981, Freedman, 1980] commonly used to model predator-prey dynamics in biological systems [Beals et al., 1999, Hoppensteadt, 2006] market dynamics [Hung et al., 2017, Prasolov, 2016]. For ease of interpretation, we will consider the biological interpretation. Let X_t and Y_t denote the population densities of prey and predators at time t, the classical Lotka-Volterra system can be written as follows

$$\begin{aligned}\frac{dX_t}{dt} &= \alpha(t)X_t - \beta(t)X_tY_t \\ \frac{dY_t}{dt} &= -\gamma(t)Y_t + \delta(t)X_tY_t\end{aligned}$$

where $\alpha(t)$ ($\gamma(t)$) represents the prey (predator) per-capita birth rate absent predators (prey), $\beta(t)$ represents how strongly predators Y_t reduce the prey population X_t, and $\delta(t)$ represents how feeding on the prey population X_t increases the predator population's growth. Each time-varying parameter listed above is assumed to be strictly positive. Note the X_tY_t interactions add a non-linearity to the system which can result in more complex dynamics relative to a linear ODE system.

As in 3.1 in Section 3.1.1, we discretize over time steps $\Delta t = 0.01$. We again adopt the $1/\tau$-representation of the time-varying parameters

$$\alpha(t) = \frac{1}{\tau_1(t)}, \quad \beta(t) = \frac{1}{\tau_2(t)}, \quad \gamma(t) = \frac{1}{\tau_3(t)}, \quad \delta(t) = \frac{1}{\tau_4(t)}.$$

The resulting discretized updates are then:

$$\begin{aligned}X_{t+1} &= X_t \; + \; \left(\tfrac{X_t}{\tau_1(t)} - \tfrac{X_tY_t}{\tau_2(t)}\right)\Delta t \\ Y_{t+1} &= Y_t + \left(-\tfrac{Y_t}{\tau_3(t)} + \tfrac{X_tY_t}{\tau_4(t)}\right)\Delta t\end{aligned} \tag{3.3}$$

As before we let the time-varying parameters $\tau_i(t)$ represent the latent processes sampled from log-normals defined with the covariates $Z_{0:T-1}$

$$\begin{aligned}\log(\tau_1(t)) &\sim \mathcal{N}\big(\boldsymbol{\beta}_0^T Z_t,\, \sigma_0^2\big) \quad \log(\tau_2(t)) \sim \mathcal{N}\big(\boldsymbol{\beta}_1^T Z_t,\, \sigma_1^2\big) \\ \log(\tau_3(t)) &\sim \mathcal{N}\big(\boldsymbol{\beta}_2^T Z_t,\, \sigma_2^2\big) \quad \log(\tau_4(t)) \sim \mathcal{N}\big(\boldsymbol{\beta}_3^T Z_t,\, \sigma_3^2\big)\end{aligned}$$

The same priors for the coefficients $\beta_{i,j} \sim \mathcal{N}(0,1)$ and $\sigma_i \sim \text{Gamma}(k_\sigma, \theta_\sigma)$ are used, here we fix the variance parameters so no priors are required. A covariate time-series $Z_{0:T-1}$ was

generated to as follows, each observation $Z_t \in \mathbb{R}^2$ corresponds to (1) food supply ($Z_t^{(1)}$) and (2) seasonal weather ($Z_t^{(2)}$) which were generated as follows:

$$Z_t^{(1)} = \begin{cases} 0.5 + 0.05\varepsilon_t + J_t & t \geq 200 \\ 0 & t < 200 \end{cases}$$
$$Z_t^{(2)} = \begin{cases} 0.5\cos\left(\frac{2\pi t}{100}\right) + 0.05\epsilon_t & t \geq 500 \\ 0 & t < 500 \end{cases}$$

where are I.I.D. $\varepsilon_t, \epsilon_t \sim \mathcal{N}(0,1)$ and J_t is a jump occurring with probability $p = 0.02$ with size $S \sim \text{Uni}(-0.2, 0.2)$. We let the food supply $Z_t^{(1)}$ impact τ_1, τ_3, τ_4, and the seasonal weather $Z_t^{(2)}$ impact τ_1, τ_2, τ_4. the covariate time-series, as well as a single simulated path $D_{1:t}$ can be seen in Figure 3.4. Using the SSM notation in 2.2.1 we write the parameter θ, augmented latent state $(\mathbf{X}_t, V_t)$ and transition function $\mathbf{F}$ as follows

$$\begin{aligned} \theta &= (\boldsymbol{\beta}_1, \sigma_1, \boldsymbol{\beta}_2, \sigma_2, \boldsymbol{\beta}_3, \sigma_3, \boldsymbol{\beta}_4, \sigma_4) \\ \mathbf{X}_t &= (X_t, Y_t)^T, \\ V_t &= \left(\frac{1}{\tau_1(t)}, \frac{1}{\tau_2(t)}, \frac{3}{\tau_1(t)}, \frac{1}{\tau_4(t)}\right)^T \\ \mathbf{X}_{t+1} &= \mathbf{X}_t + \begin{pmatrix} X_t & -X_t Y_t & 0 & 0 \\ 0 & 0 & -Y_t & X_t Y_t \end{pmatrix} V_t \Delta t = \mathbf{F}(\mathbf{X}_t, V_t) \end{aligned}$$

3.2.2 Experiment Parameters

The settings are very similar to before, 20 independent seeds with $N_o = 5,000$ outer particles and $N_i = 100$ inner particles over a time frame of $T = 1000$ time steps with states initialized as $\mathbf{X}_0 = (10, 5)^T$, and the true parameter vector θ was chosen to be:

$$\begin{aligned} \theta &= (-0.75, 0.15, 0.5, 1.2, -0.15, -0.5, 0.35, 1.7, 0.5, 0.1, 0.05^\star, 0.05^\star, 0.05^\star, 0.05^\star)^T \\ &= (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\beta}_4, \sigma_1, \sigma_2, \sigma_3, \sigma_4)^T \end{aligned} \tag{3.4}$$

The parameters marked $\star$ are fixed, in this case time-varying parameters' variances are fixed parameters. The coefficients follow standard normal priors, $\beta_{ij,k} \sim \mathcal{N}(0,1)$. The Gaussian

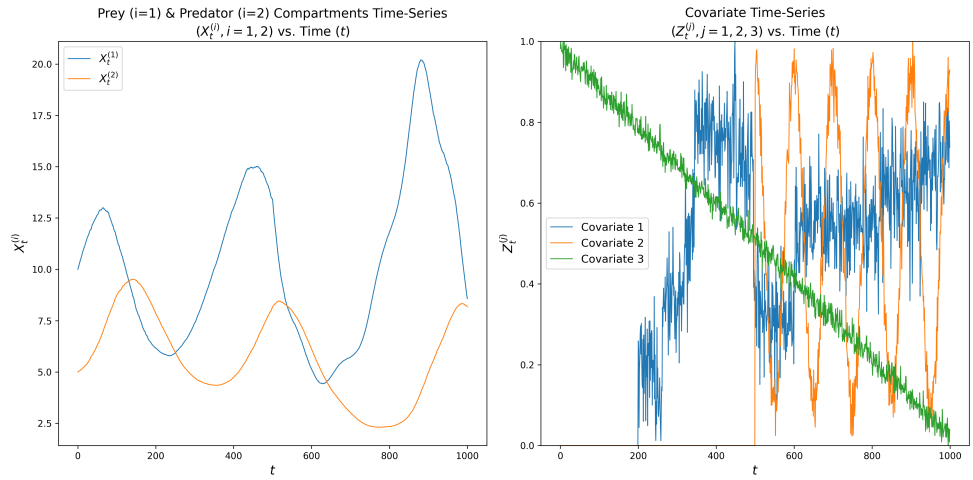


Figure 3.4: Left: time-series of the covariates informing flow rates τ_{ij}. $Z_{t,i}$ is the ith covariate at time t. Right: time-series of the compartments $X_t^{(i)}$ vs. time t.

kernel with bandwidth ε was used with an $\|\cdot\|_2$-norm measuring the distance between the observed and simulated data with an identity observation function G. The proposal kernel for the coefficients is a multivariate normal distribution ($\boldsymbol{\beta} \sim \mathcal{MVN}(\hat{\mu}_\beta, \hat{\Sigma}_\beta)$), the estimators are as defined in 3.1. The resampling threshold used was $\gamma = 0.95$. A summary of the experimental configurations is provided in Table 3.5.

We note that the LV-equations do not lie in $[0, 1]$ as our previous example, this means we have to aaccommodatefor larger ε values.

3.2.3 Results

The results of three ε-scheduling methods; Constant ($\varepsilon = 0.01$), Del Moral, and our novel method was assessed for parameter estimation in the context of high-magnitude and highly variable Lotka–Volterra (LV) dynamics. We cannot access the performance of the $\varepsilon = 0.001$ scheduler as it failed to complete a single simulation, exhibiting a 100% collapse rate showing such a small initial tolerance cannot accommodate significant shifts in magnitude in the observed data $D_{1:T}$. Lastly, we do assess the performance of the Del Moral scheduler here, but we note it exhibited a high 84% collapse rate.

Table 3.6 shows the performance of the remaining three methods listed above, showing varying levels of estimation accuracy. The difficulties exhibited by all methods are primarily

Table 3.5: Experimental Settings for ABC-SMC^2 Simulation for the LV-Equations

Setting	**Value**
Number of seeds (runs)	20
True parameter θ	see 3.4
Priors	$\beta_{ij,k} \sim \mathcal{N}(0,1)$, $\sigma_k \sim \text{Gamma}(0.2, 0.5)$
Outer particles N_o	5,000
Inner particles N_i	100
Timeframe T	1000
ABC-Kernel	$K_\varepsilon(x,y) = \phi(\|x-y\|_2)/\varepsilon$
ESS threshold γ	0.95
Proposal kernel	$\boldsymbol{\beta} \sim \mathcal{MVN}(\hat{\mu}_\beta, \hat{\Sigma}_\beta), \sigma_k \sim \mathcal{LN}(\hat{\mu}_k, \hat{\sigma}_k)$
Initial state $\mathbf{X}_0$	$(10,5)^T$
Quality Index	0.975
Recalibration Scheduler	$(\eta_0, \eta_T) = (0.1, 0.01)$
Recalibration Frequency	20

attributed to the large variability in the LV data (as seen in Figure 3.4), particularly in magnitude, which inhibits accurate posterior estimation for all ε-scheduling strategies.

We see that our approach consistently achieves the lowest RMSE across most parameters, outperforming the Constant ($\varepsilon = 0.01$) and Del Moral schedulers. The adaptive ε_t mechanism effectively balances particle diversity and posterior refinement. The Del Moral scheduler, while occasionally producing credible intervals comparable to the Novel scheduler, suffers from a high collapse rate (84%) due to its monotonic $(\varepsilon_t)_t$ sequence. The Constant ($\varepsilon = 0.01$) scheduler performs reasonably well for some parameters but fails for those highly sensitive to covariate variability, such as $\beta_{4,2}$ and $\beta_{3,2}$. Table 3.7 shows the average 95% for each parameter estimate credible intervals.

The Novel scheduler does yield narrower credible intervals, although said intervals do not always include the true parameter values. This limitation is common to the other two methods as well, again we attribute this to the variability in magnitude inherent in the LV model. Overall, the wider credible intervals of the Constant ($\varepsilon = 0.01$) scheduler indicate weaker posterior fits relative to the narrower intervals resulting from the Novel scheduler.

θ	True Value	Constant (0.01)		Del Moral		Novel	
$\beta_{1,1}$	-0.75	-0.669	(0.152)	-0.600	(0.199)	-0.711	(**0.088**)
$\beta_{1,2}$	0.15	0.115	(0.072)	0.075	(0.096)	0.135	(**0.034**)
$\beta_{1,3}$	0.50	0.435	(0.106)	0.395	(0.161)	0.460	(**0.089**)
$\beta_{2,1}$	1.20	1.273	(0.134)	1.329	(0.170)	1.237	(**0.080**)
$\beta_{2,2}$	-0.15	-0.232	(0.144)	-0.281	(0.170)	-0.201	(**0.102**)
$\beta_{3,1}$	-0.50	-0.368	(0.205)	-0.429	(0.136)	-0.468	(**0.071**)
$\beta_{3,2}$	0.35	0.248	(0.150)	0.414	(0.114)	0.339	(**0.060**)
$\beta_{4,1}$	1.70	1.833	(0.203)	1.761	(0.129)	1.738	(**0.083**)
$\beta_{4,2}$	0.50	0.386	(0.160)	0.534	(0.075)	0.478	(**0.068**)
$\beta_{4,3}$	0.10	0.107	(0.020)	0.120	(0.041)	0.105	(**0.017**)

Table 3.6: Mean and RMSE (in parentheses) of parameter estimates for three ε scheduling methods: Constant (0.01), Del Moral, and Novel for the LV-equations. **Bolded** RMSE values represent the smallest RMSE across all methods.

θ_i	True Value	Constant (0.01)		Del Moral		Novel	
$\beta_{1,1}$	-0.75	-0.669	[-0.683, -0.654]	-0.600	[-0.602, -0.596]	-0.711	[-0.735, -0.687]
$\beta_{1,2}$	0.15	0.115	[0.095, 0.132]	0.075	[0.073, 0.076]	0.135	[0.111, 0.160]
$\beta_{1,3}$	0.50	0.435	[0.403, 0.473]	0.395	[0.382, 0.401]	0.460	[0.417, 0.503]
$\beta_{2,1}$	1.20	1.273	[1.258, 1.288]	1.329	[1.326, 1.333]	1.237	[1.214, 1.257]
$\beta_{2,2}$	-0.15	-0.232	[-0.264, -0.200]	-0.281	[-0.295, -0.272]	-0.201	[-0.238, -0.163]
$\beta_{3,1}$	-0.50	-0.368	[-0.389, -0.353]	-0.429	[-0.432, -0.427]	-0.468	[-0.499, -0.435]
$\beta_{3,2}$	0.35	0.248	[0.200, 0.289]	0.414	[0.408, 0.426]	0.339	[0.275, 0.402]
$\beta_{4,1}$	1.70	1.833	[1.812, 1.850]	1.761	[1.760, 1.764]	1.738	[1.705, 1.773]
$\beta_{4,2}$	0.50	0.386	[0.341, 0.422]	0.534	[0.527, 0.538]	0.478	[0.418, 0.540]
$\beta_{4,3}$	0.10	0.107	[0.085, 0.136]	0.120	[0.118, 0.125]	0.105	[0.077, 0.139]

Table 3.7: Mean and credible intervals [in square brackets] of parameter estimates for three ε scheduling methods: Constant (0.01), Del Moral, and Novel for the LV-equations.

3.3 SEIR Model

3.3.1 Setup

The COVID-19 pandemic, driven by the SARS-CoV-2 virus, has significantly influenced public health and policy since its emergence in late 2019 [Chen et al., 2020]. To manage and mitigate the spread of the disease, public health measures such as mobility restrictions and policy stringency were implemented, with their effectiveness supported by various studies [Tian et al., 2021]. The evolution of the virus into different variants introduced new complexities into policy making, as later variants had the capacity to infect previously recovered individuals, contributing to the phenomenon of reinfections [Burki, 2021, Choi and Smith, 2021, Harvey et al., 2021].

In this section, we apply our compartment model framework to the well-known SEIR-model [Beckley et al., 2013, Harko et al., 2014, Kröger and Schlickeiser, 2020, Schlickeiser and Kröger, 2021] and a COVID-19 dataset spanning a 2.5 year time-period and incorporating a diverse set of covariates. We utilize daily infection data from Public Health Ontario. Furthermore, we incorporate various covariate data, such as mobility [Google LLC, 2022], stringency of public policies [Bank of Canada, 2022], wastewater, vaccination rates, and variant prevalence. This detailed dataset allows us to inform the transition rates $\tau_{ij}(t)$ (modelled via GLMs) more accurately. This approach benefits from embedding these diverse covariates directly into the stochastic framework, thereby capturing the dynamic nature of disease transmission more realistically.

By incorporating these additional data points, our approach aims to enhance the precision and flexibility of compartmental models in capturing real-world dynamics. This method strives to avoid simplistic assumptions often present in conventional epidemiological modelling, supporting a more nuanced and robust representation of disease transmission [Ioannidis et al., 2022, Liu et al., 2020b, Sartorius et al., 2021].

The SEIR model assumes four compartments; those who are susceptible S who may be infected after being exposed, those who have been exposed E but are not yet infected, those who have been infected I, and those who have recovered R. If we assume the population has a total of N individuals then $S + E + I + R = N$, dividing each compartment by N gives a more numerical tractable model where we consider the portion of the population in each state, the model is governed by the following non-linear system of ODEs.

$$
\begin{aligned}
\frac{dS}{dt} &= -\frac{S \cdot I}{\tau_{SE}} + \frac{R}{\tau_{RS}} \\
\frac{dE}{dt} &= \frac{S \cdot I}{\tau_{SE}} - \frac{E}{\tau_{EI}} \\
\frac{dI}{dt} &= \frac{E}{\tau_{EI}} - \frac{I}{\tau_{IR}} \\
\frac{dR}{dt} &= \frac{I}{\tau_{IR}} - \frac{R}{\tau_{RS}}
\end{aligned}
$$

The corresponding discretized system (with uniform time steps of $\Delta t = 1$ as we have daily data), is given below:

$$
\begin{aligned}
S_{t+1} &= S_t + \frac{R_t}{\tau_{RS}(t)} - \frac{S_t I_t}{\tau_{SE}(t)} \\
E_{t+1} &= E_t + \frac{S_t I_t}{\tau_{SE}(t)} - \frac{E_t}{\tau_{EI}(t)} \\
I_{t+1} &= I_t + \frac{E_t}{\tau_{EI}(t)} - \frac{I_t}{\tau_{IR}(t)} \\
R_{t+1} &= R_t + \frac{I_t}{\tau_{IR}(t)} - \frac{R_t}{\tau_{RS}(t)}
\end{aligned}
$$

where the τ's are the time-varying parameters. We set the initial condition of the system to be $S_0 = \frac{N-1}{N}, E_0 = 0, I_0 = \frac{1}{N}, R_0 = 0$ as per [Beckley et al., 2013, Harko et al., 2014, Kröger and Schlickeiser, 2020]. Below we detail the covariate data utilized to inform the transitions via the time-varying τ's, no covariate data was generated so the underlying covariate data is more reflective on real-world data.

- Mobility - changes in mobility patterns will affect contact rates in the population, in turn increasing the number of interactions where the disease is transmitted, hence it influences the $S \mapsto E$ transition.
- Stringency - policy affecting the degree to which the population may move and interact will directly impact transmission rates as well, hence it influences the $S \mapsto E$ transition.

- Wastewater - an indicator of the prevalence and level of the virus in a given community, serves as a proxy for the prevailing infection rate so it influences the $E \mapsto I$ transition.
- Variants - transmission dynamics and reinfection rates differ under different strains/variants, therefore this covariate influences transitions $E \mapsto I, I \mapsto R, R \mapsto S$.
- Vaccination Rate - a direct form of immunity against the virus which influences the $E \mapsto I$ transition. Vaccines are designed based on the known strains of the virus, but may not be as effective against newer strains and therefore we account for vaccination rate in the $R \mapsto S$ transition as it may have an offsetting effect.

The time-varying transmission periods τ's are assumed to follow a log-normals with means determined by coefficients β and variances σ^2's as:

$$\tau_{SE}(t) \sim \text{LogNormal}(\boldsymbol{\beta}_{SE}^T Z_{SE,t}, \sigma_{SE}^2) \tag{3.5}$$
$$\tau_{EI}(t) \sim \text{LogNormal}(\boldsymbol{\beta}_{EI} Z_{EI,t}, \sigma_{EI}^2) \tag{3.6}$$
$$\tau_{IR}(t) \sim \text{LogNormal}(\boldsymbol{\beta}_{IR}^T Z_{IR,t}, \sigma_{IR}^2) \tag{3.7}$$
$$\tau_{RS}(t) \sim \text{LogNormal}(\boldsymbol{\beta}_{RS}^T Z_{RS,t}, \sigma_{RS}^2) \tag{3.8}$$

where $\beta_{ij,k} \sim \mathcal{N}(0, \sigma_\beta^2)$ and $\sigma \sim \Gamma(k_\Gamma, \theta_\Gamma)$ with k_Γ for shape and θ_Γ for scale, for all σ_{ij}'s as the priors.

We formalize this model using the notation from section 2.2.1

$$\begin{aligned}
\theta &= (\boldsymbol{\beta}_{SE}, \sigma_{SE}, \boldsymbol{\beta}_{EI}, \sigma_{EI}, \boldsymbol{\beta}_{IR}, \sigma_{IR}, \boldsymbol{\beta}_{RS}, \sigma_{RS}) \\
\mathbf{X}_t &= (S_t, E_t, I_t, R_t)^T \\
V_t &= \left(\frac{1}{\tau_{SE}(t)}, \frac{1}{\tau_{EI}(t)}, \frac{1}{\tau_{IR}(t)}, \frac{1}{\tau_{RS}(t)} \right)^T \\
\mathbf{X}_{t+1} &= \mathbf{X}_t + \begin{pmatrix} -S_t I_t & 0 & 0 & R_t \\ S_t I_t & -E_t & 0 & 0 \\ 0 & E_t & -I_t & 0 \\ 0 & 0 & I_t & -R_t \end{pmatrix} V_t = \mathbf{F}(\mathbf{X}_t, V_t)
\end{aligned}$$

3.3.2 Experiment Parameters

The settings for the numerical experiment are as follows: 20 independent seeds with $N_o = 5,000$ outer particles and $N_i = 100$ inner particles over a time frame of $T = 884$ time steps. The compartments are initialized at $\mathbf{X}_0 = \left(\frac{N-1}{N}, 0, \frac{1}{N}, 0\right)^T, N = 14,000,000$ (approximate population of Ontario during 2020-2023 timeframe), and the true parameter vector θ has 28 parameters, please see Table 3.8 for a complete listing.

Table 3.8: True Parameters for SEIR Model by Covariate and Transition

Covariate	**S → E**	**E → I**	**I → R**	**S → R**	**Start Time**
Intercept	0.0⋆	0.0⋆	0.0⋆	0.0⋆	0
Mobility	0.20	–	–	–	0
Stringency	0.10	–	–	–	0
Wastewater	–	2.25	–	–	266
Non-VOC	–	1.50	1.00	1.00	0
Alpha	–	1.20	0.70	0.60	323
Gamma	–	0.90	0.60	0.40	354
Delta	–	0.60	0.50	0.20	413
Omicron	–	0.30	0.40	0.10	658
Vaccination Rate	–	-0.75	–	1.00	335
Variances	0.01⋆	0.01⋆	0.01⋆	0.01⋆	0

Table 3.9: This table outlines the true parameters used for simulating data in the SEIR model. Each column corresponds to a transition between compartments, while each row details a specific parameter and the covariate it is associated with in time-varying transition rates (τ). Parameters marked with a star (*) are fixed and not learned, leaving a total of 20 parameters to be estimated. The final column, "Start Time," indicates when each parameter begins to influence the simulation; before this time, the corresponding covariate is zero (hence its coefficient θ_i will have no influence on τ).

In this experiment, we fix the variances and intercepts. This assumption is reasonable as the baseline effects for these transitions are largely absorbed by the dynamics of the non-variant-of-concern (non-VOC) strain, capturing the natural infection and recovery rates without significant influence from covariates. As before, the coefficients follow standard normal priors, $\beta \sim \mathcal{N}(0, 1)$, and the variances are drawn from a gamma distribution, $\sigma \sim \text{Gamma}(0.2, 0.5)$. The same ABC-kernel, and PMH proposal kernel were used, as well as the resampling threshold $\gamma = 0.95$.

Table 3.10: Experimental Settings for ABC-SMC^2 Simulation for the SEIR model.

Setting	**Value**
Number of seeds (runs)	20
True parameter θ	See Table 3.8
Priors	$\beta_{ij,k} \sim \mathcal{N}(0,1)$
Outer particles N_o	5,000
Inner particles N_i	100
Timeframe T	884
ABC-Kernel	$K_\varepsilon(x,y) = \phi(\|x-y\|_2)/\varepsilon$
ESS threshold γ	0.95
Proposal kernel	$\boldsymbol{\beta} \sim \mathcal{MVN}(\hat{\mu}_\beta, \hat{\Sigma}_\beta)$
Initial state $\mathbf{X}_0$	$(\frac{N-1}{N}, 0, \frac{1}{N}, 0)^T, N = 14,000,000$
Quality Index	0.975
Recalibration Scheduler	$(\eta_0, \eta_T) = (0.01, 0.001)$
Recalibration Frequency	20

Figure 3.5 shows the a sample trajectory of SEIR compartments for a single seed, providing a representative example of the observation data that the SMC sampler will learn from. For consistency, all subsequent figures shown in this section involving a single-simulation (single seed) are based on this seed.

3.3.3 Results

This experiment extends the our ABC-SMC^2 method to a higher dimensional setting compared to the previous two sections as outlined in Table 3.8. The increased dimensionality introduces additional complexity into the observation data dynamics and parameter learning process. Moreover, the influence of parameters on the transitions (the τ's) begins at varying times throughout the simulation (see column "Start Time" in Table 3.8) as some parameters are unavailable early in the timeframe (e.g., due to data collection limitations, such is the case for wasterwater), while others only become relevant at later stages (e.g., parameters associated with the emergence new variants, such as Omicron).

As in the earlier section, we fixed 20 unique seeds to generate 20 distinct observation trajectories and ran the SMC sampler on each trajectory. As before, should an SMC sampler collapse (as defined in 3.1), we restart the sampler from time $t = 1$ on the same

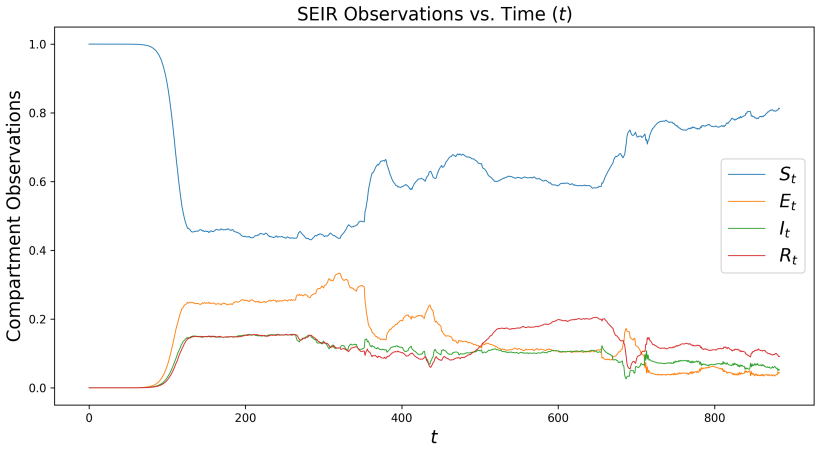


Figure 3.5: Simulated SEIR compartment observations for a single seed. The figure illustrates the dynamics of the Susceptible (S_t), Exposed (E_t), Infected (I_t), and Recovered (R_t) compartments over time (t) under the specified experimental parameters.

observation path $D_{1:T}$. To prevent infinite restarts, we imposed a maximum of 10 restart attempts per seed. Both the $\varepsilon = 0.001$ method and the Del Moral scheduler failed to complete a single simulation across all 20 seeds. For these methods, the sampler repeatedly collapsed after 10 restart attempts, failing to progress to the end of the simulation.

The poor performance of the $\varepsilon = 0.001$ method and the Del Moral scheduler arises from their inability to dynamically adjust to evolving data regimes. The observed data $D_{1:T}$ (see Figure 3.5) provides the best demonstration as two why these methods are not suitable for time-series with regime changes. As in the figure, as $t \to 100$ the data undergoes a significant behaviour change from being nearly constant to increasing/decreasing dramatically. The Del Moral method's monotonic decreasing $(\varepsilon_t)_t$ schedule reduces ε too quickly during the early phases of the simulation, as the initial observations are relatively stable and simple to fit, resulting in reductions in the ε value. As new variability emerges in the data, the small ε-value causes the SMC sampler to struggle to maintain particle diversity, leading to severe degeneration in the ESS resulting in frequent collapse.

Figure 3.6 illustrates the evolution of ESS values over time for a single seed, showing multiple attempts of the SMC sampler under the Del Moral scheduler. Notably, the initial stability of the compartment trajectories allows ε to shrink rapidly, but this overly aggressive reduction ultimately compromises the sampler's robustness when the data dynamics

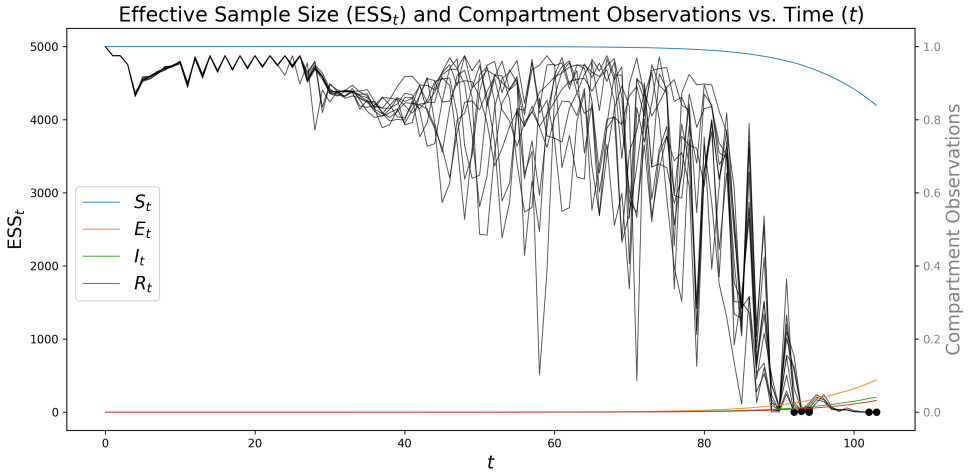


Figure 3.6: This figure displays the effective sample size (ESS) and compartment observations over time for the SMC sampler. The black lines represent the ESS paths for different sampler attempts, with dots marking the points of collapse where ESS reaches 1, signifying sampler failure. The coloured lines on the right axis correspond to the compartment observations (S_t, E_t, I_t, R_t).

become more complex.

In contrast, the $\varepsilon = 0.01$ method successfully completes all simulations due to its large and constant ε value, which prevents particle collapse. However, this large tolerance comes at the cost of reduced posterior precision. As shown in Table 3.11, the $\varepsilon = 0.01$ method produces reasonable point estimates for some parameters but exhibits significantly higher RMSE values compared to our novel method. Furthermore, Table 3.11 demonstrates that the posterior credible intervals for the $\varepsilon = 0.01$ method are substantially wider, reflecting inflated posterior uncertainty.

The performance differences between the $\varepsilon = 0.01$ method and our novel approach are further illustrated in Figure 3.7. The figure compares the 95% credible intervals for the simulated SEIR compartments across both methods. For all compartments, the novel method exhibits narrower credible intervals, reflecting lower posterior uncertainty. This is particularly evident in compartments governed by parameters introduced at later time steps, where the novel method's adaptive scheduling and recalibration provide a significant advantage in estimating parameters that have less observational data to inform their posteriors. Additionally, the variance in the trajectories appears to correlate with the number of parameters influencing each compartment. For example, the susceptible compartment,

θ_i	True Value	Constant (0.01)			Novel		
		Estimate	RMSE	95% CI	Estimate	RMSE	95% CI
$\beta_{SE,1}$	0.20	0.201	(**0.005**)	[0.195, 0.208]	0.202	(**0.005**)	[0.198, 0.206]
$\beta_{SE,2}$	0.10	0.100	(**0.006**)	[0.095, 0.105]	0.100	(**0.006**)	[0.096, 0.103]
$\beta_{EI,1}$	2.25	2.241	(0.032)	[2.184, 2.301]	2.251	(**0.008**)	[2.238, 2.264]
$\beta_{EI,2}$	1.50	1.501	(0.005)	[1.495, 1.507]	1.501	(**0.004**)	[1.498, 1.503]
$\beta_{EI,3}$	1.20	1.190	(0.048)	[1.115, 1.267]	1.196	(**0.018**)	[1.184, 1.209]
$\beta_{EI,4}$	0.90	0.899	(0.069)	[0.799, 0.998]	0.903	(**0.016**)	[0.879, 0.928]
$\beta_{EI,5}$	0.60	0.610	(0.023)	[0.576, 0.641]	0.600	(**0.007**)	[0.593, 0.607]
$\beta_{EI,6}$	0.30	0.291	(0.053)	[0.202, 0.387]	0.294	(**0.021**)	[0.280, 0.309]
$\beta_{EI,7}$	-0.75	-0.731	(0.070)	[-0.853, -0.619]	-0.742	(**0.027**)	[-0.760, -0.724]
$\beta_{IR,1}$	1.00	1.001	(**0.005**)	[0.996, 1.004]	1.000	(**0.005**)	[0.998, 1.003]
$\beta_{IR,2}$	0.70	0.701	(**0.005**)	[0.696, 0.706]	0.701	(**0.005**)	[0.698, 0.705]
$\beta_{IR,3}$	0.60	0.603	(0.016)	[0.576, 0.629]	0.605	(**0.014**)	[0.586, 0.623]
$\beta_{IR,4}$	0.50	0.500	(**0.006**)	[0.492, 0.509]	0.500	(**0.007**)	[0.494, 0.507]
$\beta_{IR,5}$	0.40	0.401	(**0.003**)	[0.396, 0.405]	0.401	(**0.003**)	[0.398, 0.404]
$\beta_{SR,1}$	1.00	1.001	(0.007)	[0.992, 1.011]	1.000	(**0.004**)	[0.998, 1.004]
$\beta_{SR,2}$	0.60	0.644	(0.061)	[0.574, 0.714]	0.604	(0.019)	[0.589, 0.617]
$\beta_{SR,3}$	0.40	0.415	(0.070)	[0.286, 0.538]	0.412	(**0.017**)	[0.382, 0.440]
$\beta_{SR,4}$	0.20	0.202	(0.024)	[0.161, 0.243]	0.199	(**0.008**)	[0.191, 0.206]
$\beta_{SR,5}$	0.10	0.153	(0.075)	[0.068, 0.235]	0.103	(**0.021**)	[0.088, 0.120]
$\beta_{SR,6}$	1.00	0.936	(0.091)	[0.833, 1.041]	0.997	(**0.026**)	[0.977, 1.016]

Table 3.11: Mean, RMSE (in parentheses), and credible intervals [in brackets] of parameter estimates for the Constant (0.01) and Novel ε scheduling methods. **Bolded** RMSE values indicate the method-wise minimum RMSE achieved. Due to the fixed intercepts (see Table 3.8) the second subscript $\beta_{ij,1}$ does not indicate the intercept as in the previous examples.

which depends on fewer parameters than the others, shows smaller credible intervals for both methods.

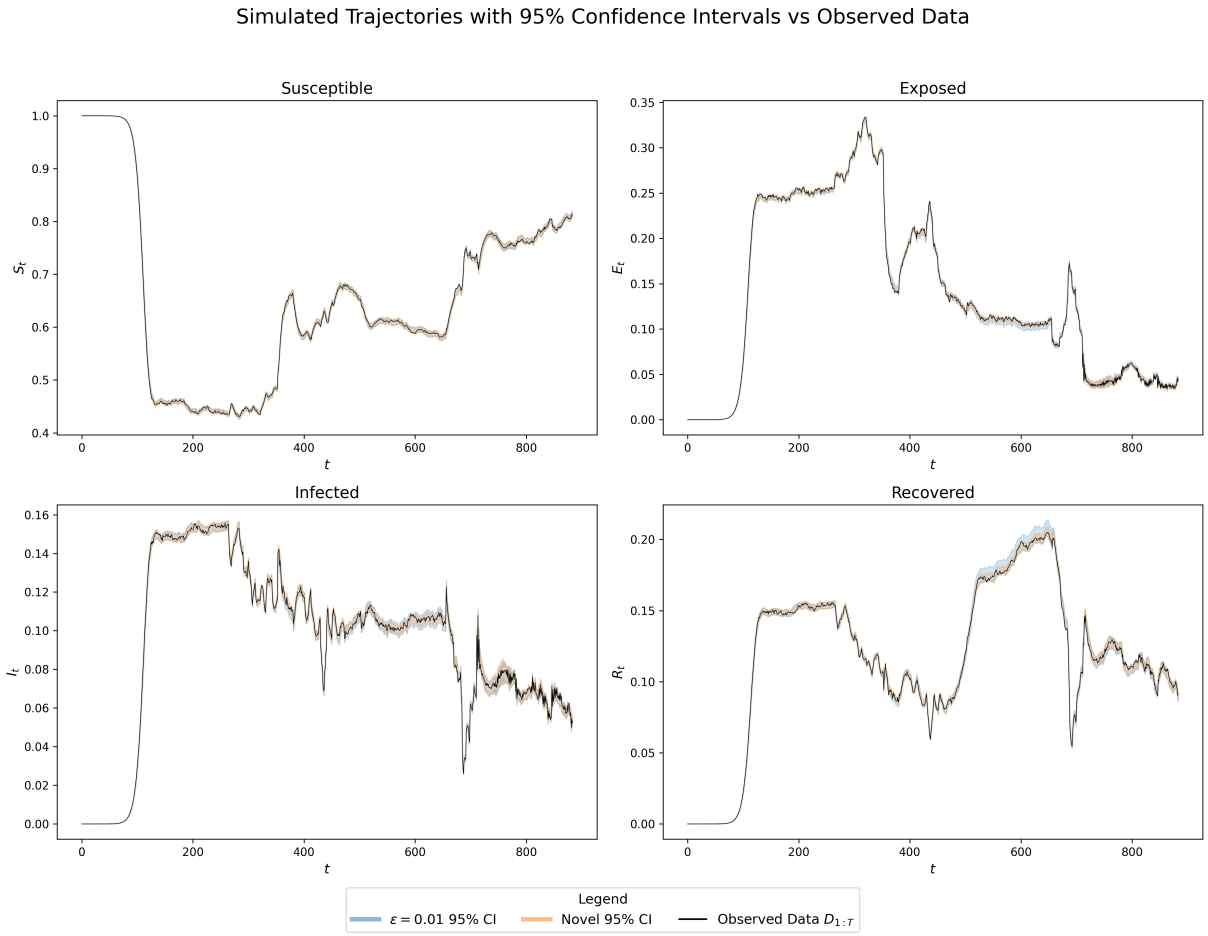


Figure 3.7: 95% credible intervals for SEIR compartment trajectories under the $\varepsilon = 0.01$ and novel methods. Shaded regions reflect posterior uncertainty, with narrower intervals observed for the novel method.

Chapter 4

Conclusion

In this thesis we proposed a novel online-ABC-SMC^2 algorithm for Markov state-space models. We introduced an adaptive ε-scheduler for time-series data that effectively manages the particle diversity throughout the simulation while promoting convergence to the target posterior. Lastly, we introduced a likelihood recalibration that allows for retroactive adjustments to the ABC-kernel history $\varepsilon_{1:t}$ to promote more accurate posterior convergence. We briefly remark on some limitations of our approach as well as potential directions for future research.

4.1 Limitations

Here we remark on some of the limitations of our approach. Firstly, in online learning setting, achieving accurate parameter estimates during early iterations is difficult due to limited (and potentially non-informative) observed data and iterations to approximate the posterior, therefore, the resulting in posterior distribution will still resemble the prior. A potential solution is to incorporate an initial offline phase where the ABC-SMC^2 can sequentially approach the target distribution in an offline context. Upon running the offline phase, the resulting parameter estimates can then be incorporated into a more informed prior distributions for the online-learning phase to follow. This approach is computationally feasible as well due to the small set of observations.

Furthermore, our proposed method introduces dependence on many additional hyper-parameters; the search space for ε (although $\varepsilon_{\min}$ can be set to 0 in most cases), the quality

index α and the recalibation frequency f, and the initial and target values for the dominating scheduler η_0, η_T. Furthermore, the general ABC-SMC^2 algorithm also depends on largely arbitrary choices for N_i, N_o, γ as well. The choice of these hyperparameters can significantly impact the performance and posterior convergence of the algorithm hence suitable calibration strategies may be required, which are computationally expensive in an SMC context. While some work has been done in an offline context [Chopin et al., 2015] making optimal choices for the hyperparameters could resulting in significantly more efficient future implementations.

Furthermore, our method inherits the typical limitations of ABC, in particular the lack of a gradient. The gradient-free nature of ABC methods in SMC algorithms can introduce convergence challenges, particularly in high-dimensional parameter spaces. Without gradient information, the algorithm cannot effectively leverage the local geometry of the parameter space, resulting in a "blind" search process that can struggle when the parameter space includes ridges, valleys, or highly correlated parameters [Beaumont, 2010, Sisson et al., 2018]. The absence of a curvature measure also limits the ability to make adaptive step size adjustments and align with the posterior distribution, unlike gradient-based approaches such as HMC [Carpenter et al., 2017, Duane et al., 1987, Neal, 2011]. Recent works such as synthetic likelihoods and gradient-mimicking techniques have been proposed to improve convergence and facilitate more efficient exploration of the parameter space [Wilkinson, 2014, Wood, 2010]. Another consequence of using ABC to approximate the likelihood is the dependence on arbitrarly chosen kernel functions.

4.2 Future Directions

Several potential areas for future work could enhance ABC-SMC^2 methods for both offline and online learning contexts.

Investigating smooth likelihood approximations, such as Gaussian mixture models (GMMs), could be beneficial. Fitting a GMM to particle weights or observed data could provide a continuous representation of the likelihood, allowing for approximation of the gradient of the posterior $\nabla\pi(\theta|x)$. Gradient information could offer insights into the local geometry of the parameter space, improving convergence, particularly in high-dimensional settings.

Another potential direction is the use of context-specific proposal propagation kernels. For example, during exploration, removing off-diagonal dependencies from the covariance matrix may encourage broader sampling, while in phases of exploitation, using weighted

covariance matrices and means could guide particles more effectively. The choice of proposal kernel could be adapted based on data behavior, likelihood characteristics, or particle weight distributions.

Adapting distance metrics or kernel functions through machine learning techniques is another area of interest. Adjusting these components in real time based on data characteristics could enable more effective modeling of complex, multimodal posteriors. However, this approach would likely introduce additional computational costs, requiring efficient implementation strategies.

Recall from 2.4.3 that the choice of recalibration frequency f is largely heuristic and could potentially be improved. Dynamic approaches to recalibration, for instance, inspired by methods such as the Del Morat [Moral et al., 2012] epsilon scheduler, could adapt the frequency in response to convergence metrics or other algorithmic indicators. Such an approach might help balance computational efficiency with the benefits of recalibration, particularly in an online learning context.

References

M. Aitkin. *Statistical inference: An integrated Bayesian/likelihood approach*. Chapman and Hall/CRC, 2010. URL https://books.google.com/books?id=YZvMBQAAQBAJ.

C. Albert, H. R. Künsch, and A. Scheidegger. A simulated annealing approach to approximate bayes computations. *Statistics and Computing*, 25(4):863–870, 2015. doi: 10.1007/s11222-014-9507-8.

L. J. S. Allen. *An introduction to stochastic processes with applications to biology*. Chapman and Hall/CRC, 2008. URL https://www.routledge.com/An-Introduction-to-Stochastic-Processes-with-Applications-to-Biology/Allen/p/book/9781584885405.

L.J.S. Allen. *Introduction to Stochastic Processes with Applications to Biology*. Pearson Education, Upper Saddle River, NJ, 2003. ISBN 978-0135839535.

D.H. Anderson. *Compartmental modeling and tracer kinetics*. Springer, Berlin, Germany, 2013. URL https://books.google.com/books?hl=en&lr=&id=HAfpCAAAQBAJ.

Sean C. Anderson et al. Quantifying the impact of covid-19 control measures using a bayesian model of physical distancing. *PLoS Comput. Biol.*, 16(e1008274), 2020.

C. Andrieu and A. Doucet. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010. URL http://webpages.iust.ac.ir/hajirajabi/phd/particle%20filter/Particle%20Markov%20chain%20Monte%20Carlo%20methods.pdf.

C. Andrieu, A. Doucet, and R. Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010. doi: 10.1111/j.1467-9868.2009.00736.x.

M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. doi: 10.1109/78.978374.

Alexander Aushev, Hanna Pesonen, and Andrew Howes. Likelihood-free inference in state-space models with unknown dynamics. *Statistics and Computing*, 2024. doi: 10.1007/s11222-023-10339-8. URL https://link.springer.com/article/10.1007/s11222-023-10339-8.

F. Ball, D. Mollison, and G. Scalia-Tomba. Epidemics with two levels of mixing. *The Annals of Applied Probability*, pages 46–89, 1997. doi: 10.1214/aoap/1034625254.

Bank of Canada. Covid-19 stringency index. Technical report, Bank of Canada, 2022. https://www.bankofcanada.ca/markets/market-operations-liquidity-provision/covid-19-actions-support-economy-financial-system/covid-19-stringency-index/.

B. Barnes and G.R. Fulford. *Mathematical Modelling with Case Studies: Using Maple and MATLAB*. CRC Press, Boca Raton, FL, 2nd edition, 2002. ISBN 978-1584882977.

M. Beals, L. Gross, and C. R. Townsend. Predator-prey dynamics. http://www.tiem.utk.edu, 1999. Archived from the original on 2012-12-15. Retrieved 2018-01-09.

M. A. Beaumont, J. M. Cornuet, J. M. Marin, and C. P. Robert. Approximate bayesian computation (abc) in practice. *Genetics*, 181(2):339–409, 2009. doi: 10.1534/genetics.108.100323.

Mark A. Beaumont. Approximate bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41:379–406, 2010.

R. Beckley, C. Weatherspoon, M. Alexander, M. Chandler, A. Johnson, and G. S. Bhatt. Modeling epidemics with differential equations. Internal report, Tennessee State University, 2013. URL https://www.tnstate.edu/research/pdf/Modeling%20Epidemics.pdf. Retrieved July 19, 2020.

M. G. B. Blum, M. A. Nunes, D. Prangle, and S. A. Sisson. A comparative review of dimension reduction methods in approximate bayesian computation. *Statistical Science*, 28(2):189–208, 2013. doi: 10.1214/12-STS406.

F. Brauer and C. Castillo-Chavez. *Mathematical Models in Population Biology and Epidemiology*. Springer-Verlag, 2000.

F. X. Briol. *Statistical computation with kernels*. PhD thesis, UCL, 2018. URL https://discovery.ucl.ac.uk/id/eprint/10158241/1/PhDthesis_fxbriol.pdf.

T. Burki. Understanding variants of sars-cov-2. *The Lancet*, 397(10273):462, 2021.

J. V. Candy. *Bayesian Signal Processing: Classical, Modern, and Particle Filtering Methods*. John Wiley & Sons, 2016. ISBN 978-1119125383.

B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1):1–32, 2017. doi: 10.18637/jss.v076.i01.

Y. Chen, I. H. Dinwoodie, and S. Sullivant. Sequential importance sampling for multiway tables. *The Annals of Statistics*, 34(1):523–545, 2006. URL https://projecteuclid.org/journals/annals-of-statistics/volume-34/issue-1/Sequential-importance-sampling-for-multiway-tables/10.1214/009053605000000822.short.

Y. Chen, Q. Liu, and D. Guo. Emerging coronaviruses: genome structure, replication, and pathogenesis. *Journal of Medical Virology*, 92(4):418–423, 2020.

C.C. Chiang. *Stochastic Processes in Epidemiology: HIV/AIDS, Other Infectious Diseases and Computers*. World Scientific, Singapore, 1980. ISBN 978-9810224273.

J. Y. Choi and D. M. Smith. Sars-cov-2 variants of concern. *Yonsei Medical Journal*, 62 (11):961, 2021.

N. Chopin, P. E. Jacob, and O. Papaspiliopoulos. Smc^2: An efficient algorithm for sequential analysis of state-space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426, 2013. doi: 10.1111/rssb.12016.

Nicolas Chopin and Omiros Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer, 2020. URL https://link.springer.com/content/pdf/10.1007/978-3-030-47845-2.pdf.

Nicolas Chopin, James Ridgway, Mathieu Gerber, and Omiros Papaspiliopoulos. Towards automatic calibration of the number of state particles within the SMC^2 algorithm. *ArXiv preprint 1506.00570*, Jun 2015.

D. Cooke, R. W. Hiorns, and et al. *The Mathematical Theory of the Dynamics of Biological Populations*, volume II. Academic Press, 1981.

Tiangang Cui, Kody J H Law, and Youssef M Marzouk. Dimension-independent likelihood-informed mcmc. *Journal of Computational Physics*, 304:109–137, 2016. doi: 10.1016/j.jcp.2015.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999115006701.

C.F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994. doi: 10.1016/0191-2615(94)90002-7.

T. A. Dean, S. S. Singh, and A. Jasra. Parameter estimation for hidden markov models with intractable likelihoods. *Scandinavian Journal of Statistics*, 41:707–731, 2014. URL https://arxiv.org/pdf/1103.5399.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

Peter J. Diggle and Richard Gratton. Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society, Series B*, 46(2):193–227, 1984. doi: 10.1111/j.2517-6161.1984.tb01290.x.

R. Douc, A. Garivier, E. Moulines, and J. Olsson. Sequential monte carlo smoothing for general state-space hidden markov models. *The Annals of Applied Probability*, 21(6): 2109–2145, 2011. doi: 10.1214/10-AAP735. Retrieved from https://projecteuclid.org/journals/annals-of-applied-probability/volume-21/issue-6/Sequential-Monte-Carlo-smoothing-for-general-state-space-hidden-Markov/10.1214/10-AAP735.

A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer, 2001. ISBN 978-1441930909.

A. Doucet, M. Briers, and S. Sénécal. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Signal Processing Magazine*, 23(6):53–81, 2006. doi: 10.1109/MSP.2006.248123.

Arnaud Doucet, Nando De Freitas, and Neil Gordon. Sequential monte carlo methods in practice. *Statistics for Engineering and Information Science*, 2000.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, 1987. doi: 10.1016/0370-2693(87)91197-X.

Akira Endo, Edwin Van Leeuwen, and Marc Baguelin. Introduction to particle markov-chain monte carlo for disease dynamics modellers. *Epidemics*, 27:1–15, 2019. doi: 10.1016/j.epidem.2018.09.001. URL https://www.sciencedirect.com/science/article/pii/S1755436519300301.

R. G. Everitt and P. A. Rowińska. Delayed acceptance abc-smc. *Journal of Computational and Graphical Statistics*, 30(1):1–13, 2021. URL https://www.tandfonline.com/doi/abs/10.1080/10618600.2020.1775617.

Paul Fearnhead, Omiros Papaspiliopoulos, and Gareth O Roberts. Particle filters for partially observed diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):755–777, 2008.

S. Filippi, C. P. Barnes, J. Cornebise, and M. P. H. Stumpf. On optimality of kernels for approximate bayesian computation using sequential monte carlo. *Statistical Applications in Genetics and Molecular Biology*, 12(1):1–23, 2013. URL https://arxiv.org/pdf/1106.6280.

A. Finke and A. H. Thiery. Conditional sequential monte carlo in high dimensions. *Annals of Statistics*, 51(2):395–421, 2023. URL https://projecteuclid.org/journals/annals-of-statistics/volume-51/issue-2/Conditional-sequential-Monte-Carlo-in-high-dimensions/10.1214/22-AOS2252.short.

Axel Finke and Alexandre H. Thiery. Conditional sequential monte carlo in high dimensions, 2021. URL https://arxiv.org/abs/2108.10277.

H. I. Freedman. *Deterministic Mathematical Models in Population Ecology*. Marcel Dekker, 1980.

Z. Ghahramani and G. E. Hinton. Switching state-space models. Technical report, University of Toronto Technical Report CRG-TR, 1996. Retrieved from http://www.gatsby.ucl.ac.uk/publications/papers/11-1999.pdf.

D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977. URL https://pubs.acs.org/doi/abs/10.1021/j100540a008.

Andrew Golightly and Darren J. Wilkinson. Bayesian parameter inference for stochastic biochemical network models using particle markov chain monte carlo. *Interface Focus*, 1(6):807–820, 2011. doi: 10.1098/rsfs.2011.0047. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsfs.2011.0047.

Google LLC. Google covid-19 community mobility reports. Technical report, Google LLC, 2022. https://www.google.com/covid19/mobility/ Accessed: ¡date¿.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F - Radar and Signal Processing*, 140(2):107–113, 1993. doi: 10.1049/ip-f-2.1993.0015.

D. Gunawan, R. Kohn, and M. N. Tran. Robust particle density tempering for state space models. *arXiv preprint arXiv:1805.00649*, 2018. URL https://arxiv.org/pdf/1805.00649.

T. Harko, F. S. Lobo, and M. K. Mak. Exact analytical solutions of the susceptible-infected-recovered (sir) epidemic model and of the sir model with equal death and birth rates. *Applied Mathematics and Computation*, 236:184–194, 2014. doi: 10.1016/j.amc.2014.03.030. URL https://doi.org/10.1016/j.amc.2014.03.030.

W. T. Harvey, A. M. Carabelli, B. Jackson, R. K. Gupta, E. C. Thomson, E. M. Harrison, C. Ludden, R. Reeve, A. Rambaut, S. J. Peacock, et al. Sars-cov-2 variants, spike mutations and immune escape. *Nature Reviews Microbiology*, 19(7):409–424, 2021.

M. C. Higgs. *Approximate inference for state-space models*. PhD thesis, University College London, 2011. Retrieved from https://discovery.ucl.ac.uk/id/eprint/1335828/1/1335828.pdf.

F. Hoppensteadt. Predator-prey model. *Scholarpedia*, 1(10):1563, 2006. doi: 10.4249/scholarpedia.1563.

Zhaoran Hou. *Sequential Monte Carlo for Applications in Structural Biology, Financial Time Series and Epidemiology*. Phd thesis, University of Waterloo, Waterloo, Ontario, Canada, 2024.

Zhaoran Hou and Samuel W. K. Wong. Estimating boltzmann averages for protein structural quantities using sequential monte carlo, 2022. URL https://arxiv.org/abs/2210.14216.

Hui-Chih Hung, Yu-Chih Chiu, and Muh-Cherng Wu. A modified lotka–volterra model for diffusion and substitution of multigeneration dram processing technologies. *Mathematical Problems in Engineering*, 2017:1–12, 2017. ISSN 1024-123X. doi: 10.1155/2017/3038203.

J. P. Ioannidis, S. Cripps, and M. A. Tanner. Forecasting for covid-19 has failed. *International Journal of Forecasting*, 38(2):423–438, 2022.

Petar Jovanovski, Andrew Golightly, and Umberto Picchini. Towards data-conditional simulation for abc inference in stochastic differential equations. *Bayesian Analysis*, 2024. URL https://arxiv.org/pdf/2310.10329. In Press.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. doi: 10.1115/1.3662552.

Ivis Kerama, Thomas Thorne, and Richard G. Everitt. Rare event abc-smc^2, 2022. URL https://arxiv.org/abs/2211.02172.

W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772):700–721, 1927. URL https://royalsocietypublishing.org/doi/10.1098/rspa.1927.0118.

Z. Khan, T. Balch, and F. Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. In *European Conference on Computer Vision (ECCV)*, pages 279–290, 2004. doi: 10.1007/978-3-540-24673-2_23.

M. Kröger and R. Schlickeiser. Analytical solution of the sir-model for the temporal evolution of epidemics. part a: Time-independent reproduction factor. *Journal of Physics A*, 53(50):505601, 2020. doi: 10.1088/1751-8121/abc65d. URL https://doi.org/10.1088/1751-8121/abc65d.

P. E. Lekone and B. F. Finkenstädt. Statistical inference in a stochastic epidemic seir model with control intervention: Ebola as a case study. *Biometrics*, 62(4):1170–1177, 2006. doi: 10.1111/j.1541-0420.2006.00609.x.

Yichao Li, Wenshuo Wang, Ke Deng, and Jun S Liu. Stratification and optimal resampling for sequential monte carlo. *Biometrika*, 109(1):181–194, 02 2021. ISSN 1464-3510. doi: 10.1093/biomet/asab004. URL https://doi.org/10.1093/biomet/asab004.

Fredrik Lindsten, Philip Bunch, Sumeetpal S. Singh, and Thomas B. Schön. Particle ancestor sampling for near-degenerate or intractable state transition models. *arXiv preprint arXiv:1505.06356*, 2015. URL https://arxiv.org/abs/1505.06356.

J. Lintusaari, M. U. Gutmann, R. Dutta, S. Kaski, and J. Corander. Approximate bayesian computation: a review. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 10(1):2–18, 2017. doi: 10.1002/sam.11320.

J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2001. ISBN 978-0387952308.

J. S. Liu and R. Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998. doi: 10.1080/01621459.1998.10473765.

J. S. Liu, R. Chen, and W. H. Wong. Rejection control and sequential importance sampling. *Journal of the American Statistical Association*, 93(443):1022–1031, 1998. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1998.10473764.

L. Liu, S. Vikram, J. Lao, X. Ben, A. D'Amour, S. O'Banion, M. Sandler, R. A. Saurous, and M. D. Hoffman. Estimating the changing infection rate of covid-19 using bayesian models of mobility. *medRxiv*, pages 2020–08, 2020a. URL https://www.medrxiv.org/content/early/2020/08/01/2020.08.01.20166822. Preprint.

L. Liu, S. Vikram, J. Lao, X. Ben, A. D'Amour, S. O'Banion, M. Sandler, R. A. Saurous, and M. D. Hoffman. Estimating the changing infection rate of covid-19 using bayesian models of mobility. *medRxiv*, pages 2020–08, 2020b.

S. N. MacEachern, M. Clyde, and J. S. Liu. Sequential importance sampling for non-parametric bayes models: The next generation. *Canadian Journal of Statistics*, 27(2): 251–267, 1999. URL https://onlinelibrary.wiley.com/doi/abs/10.2307/3315637.

J. M. Marin, P. Pudlo, C. P. Robert, and R. J. Ryder. Approximate bayesian computation schemes. *Biometrika*, 98(3):485–502, 2012. doi: 10.1093/biomet/asr076.

G. McLachlan and T. Krishnan. *The EM algorithm and extensions*. John Wiley & Sons, 2007. URL https://onlinelibrary.wiley.com/doi/book/10.1002/9781119993544.

Sebastian Mergner. *Applications of state-space models in Finance: An empirical analysis of the time-varying relationship between macroeconomics, fundamentals and Pan-European stock indices*. Springer, 2009. URL https://books.google.com/books?id=b_bXPlWz3FgC.

G. W. Molyneux and A. Abate. Abc (smc): Simultaneous inference and model checking of chemical reaction networks. In *Computational Methods in Systems Biology*,

pages 153–170. Springer, 2020. URL https://link.springer.com/chapter/10.1007/978-3-030-60327-4_14.

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

P. Del Moral, A. Doucet, and A. Jasra. An adaptive sequential monte carlo method for approximate bayesian computation. *Statistics and Computing*, 22(5):1009–1020, 2012. doi: 10.1007/s11222-011-9271-y.

Christian A. Naesseth, Fredrik Lindsten, and Thomas B. Schön. High-dimensional filtering using nested sequential monte carlo, 2016. URL https://arxiv.org/abs/1612.09162.

R. M. Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2:113–162, 2011.

M. Orton and W. Fitzgerald. A bayesian approach to tracking multiple targets using sensor arrays and particle filters. *IEEE Transactions on Signal Processing*, 50(2):216–223, 2002. doi: 10.1109/978377.

Y. Pawitan. *In all likelihood: Statistical modelling and inference using likelihood*. Oxford Science Publications, 2001. URL https://henriquelaureano.github.io/stuff/likelihood/likelihood.pdf.

Alexander V. Prasolov. *Some quantitative methods and models in economic theory*. Economic issues, problems and perspectives. Nova Publishers, New York, 2016. ISBN 978-1-63484-937-1.

G. M. Rao and C. Satyanarayana. Visual object target tracking using particle filter: a survey. *International Journal of Image, Graphics and Signal Processing*, 5(6):61–70, 2013. doi: 10.5815/ijigsp.2013.06.08.

R. Ross. An application of the theory of probabilities to the study of a priori pathometry.—part i. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 92(638):204–230, February 1916. doi: 10.1098/rspa.1916.0007. URL https://doi.org/10.1098/rspa.1916.0007.

R. Ross. An application of the theory of probabilities to the study of a priori pathometry.—part ii. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 93(650):212–225, May 1917. doi: 10.1098/rspa.1917.0014. URL https://doi.org/10.1098/rspa.1917.0014.

R. Ross and H. Hudson. An application of the theory of probabilities to the study of a priori pathometry.—part iii. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 89(621):225–240, 1917. doi: 10.1098/rspa.1917.0015. URL https://doi.org/10.1098/rspa.1917.0015.

Donald B. Rubin. Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, 12(4):1151–1172, 1984. doi: 10.1214/aos/1176346785.

B. Sartorius, A. Lawson, and R. Pullan. Modelling and predicting the spatio-temporal spread of covid-19, associated deaths and impact of key risk factors in england. *Scientific Reports*, 11(1):5378, 2021.

R. Schlickeiser and M. Kröger. Analytical solution of the sir-model for the temporal evolution of epidemics. part b: Semi-time case. *Journal of Physics A*, 54(17):175601, 2021. doi: 10.1088/1751-8121/abd66a. URL https://doi.org/10.1088/1751-8121/abd66a.

I. Schuster, H. Strathmann, and B. Paige. Kernel sequential monte carlo. In *Machine Learning and Knowledge Discovery in Databases*, pages 293–309. Springer, 2017. URL https://arxiv.org/pdf/1510.03105.

F. Septier and G. W. Peters. An overview of recent advances in monte carlo methods for bayesian filtering in high-dimensional spaces. In *Theoretical Aspects of Spatial-Temporal Modeling*, pages 15–32. Springer, 2015. URL https://link.springer.com/chapter/10.1007/978-4-431-55336-6_2.

D. Siegmund. Importance sampling in the monte carlo study of sequential tests. *The Annals of Statistics*, 4(4):673–684, 1976. URL https://www.jstor.org/stable/2958179.

D. Silk, S. Filippi, and M. P. H. Stumpf. Optimizing threshold-schedules for approximate bayesian computation sequential monte carlo samplers: applications to molecular systems. *arXiv preprint*, 2012. URL https://arxiv.org/abs/1210.3296.

S. A. Sisson, Y. Fan, and M. M. Tanaka. Likelihood-free inference in complex models. *Statistical Science*, 23(3):268–294, 2007a. doi: 10.1214/07-STS242.

S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007b. doi: 10.1073/pnas.0607208104.

Scott A. Sisson, Yanan Fan, and Mark A. Beaumont. *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC, 2018.

M. Sunnåker, A. G. Busetto, E. Numminen, J. Corander, M. Foll, and C. Dessimoz. A practical introduction to approximate bayesian computation. *PLoS Computational Biology*, 9(1):e1002872, 2013. doi: 10.1371/journal.pcbi.1002872.

He Hua Tang, Wan and Xin Tu. *Applied Categorical and Count Data Analysis*. Chapman & Hall/CRC Texts in Statistical Science (Taylor & Francis), 2012.

M. A. Tanner. *Tools for statistical inference*. Springer, 1993. URL https://link.springer.com/content/pdf/10.1007/978-1-4684-0192-9.pdf.

T. Tian, J. Tan, W. Luo, Y. Jiang, M. Chen, S. Yang, C. Wen, W. Pan, and X. Wang. The effects of stringent and mild interventions for coronavirus pandemic. *Journal of the American Statistical Association*, 116(534):481–491, 2021.

F. Tobar, P. M. Djurić, and D. P. Mandic. Unsupervised state-space modeling using reproducing kernels. *IEEE Transactions on Signal Processing*, 63(24):6640–6653, 2015. URL https://ieeexplore.ieee.org/ielaam/78/7217861/7130658-aam.pdf.

T. Toni. Abc smc for parameter estimation and model selection with applications in systems biology. *Nature Precedings*, 2011. URL https://www.nature.com/articles/npre.2011.5964.1.

Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael P. H. Stumpf. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, February 2009. doi: 10.1098/rsif.2008.0172.

G. Toscani and M. Zanella. Kinetic compartmental models driven by opinion dynamics: vaccine hesitancy and social influence. *arXiv preprint arXiv:2310.19601*, 2023. URL https://arxiv.org/abs/2310.19601.

N.M. Tsoukias and S.C. George. A two-compartment model of pulmonary nitric oxide exchange dynamics. *Journal of Applied Physiology*, 85(2):653–666, 1998. doi: 10.1152/jappl.1998.85.2.653. URL https://journals.physiology.org/doi/abs/10.1152/jappl.1998.85.2.653.

Richard D. Wilkinson. Accelerating abc methods using gaussian processes. *Bayesian Analysis*, 9(3):673–694, 2014. doi: 10.1214/14-BA870.

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

Yong Zeng and Shiqi Wu. *State-space models: Applications in economics and finance*. Springer, 2013. doi: 10.1007/978-1-4614-7789-1. URL https://link.springer.com/content/pdf/10.1007/978-1-4614-7789-1.pdf.