George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with bayesian conditional density estimation. *Advances in neural information processing systems*, 29, 2016.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

George Papamakarios, David Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 837–848. PMLR, 2019.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Matthew David Parno. *Transport maps for accelerated Bayesian computation*. PhD thesis, Massachusetts Institute of Technology, 2015.

Umberto Picchini. Inference for sde models via approximate bayesian computation. *Journal of Computational and Graphical Statistics*, 23(4):1080–1100, 2014.

Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

Jonathan K Pritchard, Mark T Seielstad, Anna Perez-Lezaun, and Marcus W Feldman. Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798, 1999.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Simo Särkkä. *Bayesian filtering and smoothing*. Number 3. Cambridge university press, 2013.

Scott A Sisson, Yanan Fan, and Mark Beaumont. *Handbook of approximate Bayesian computation*. CRC Press, 2018.

Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

Alvaro Tejero-Cantero, Jan Boelts, Michael Deistler, Jan-Matthis Lueckmann, Conor Durkan, Pedro J. Gonçalves, David S. Greenberg, and Jakob H. Macke. sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505, 2020. doi: 10.21105/joss.02505. URL `https://doi.org/10.21105/joss.02505`.

T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. 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.

Darren J Wilkinson. *Stochastic modelling for systems biology*. CRC press, 2018.

A Derivations of ABC and incremental posteriors of HMM

A.1 Joint distribution for HMM using ABC

NLFI methods are designed to efficiently sample from the marginal distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$. In ABC although the desired outcome often is the marginal distribution, however it is easy to show that for a latent variable model, such as an implicit HMM, ABC does indeed target an approximation of the joint distribution $p(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y})$.

In ABC we rely upon simulation of a pseudo-data $\hat{\boldsymbol{y}}$, when the likelihood $p(\boldsymbol{y}|\boldsymbol{\theta})$ is intractable. The operating principle of any standard ABC algorithm, based on rejection sampling (Pritchard et al.,

1999), MCMC (Marjoram et al., 2003) or SMC (Toni et al., 2009; Del Moral et al., 2012), is to jointly sample the parameters $\boldsymbol{\theta}$ and the pseudo-data $\hat{\boldsymbol{y}}$ from their posterior density (Marin et al., 2012)

$$p_\epsilon(\boldsymbol{\theta}, \hat{\boldsymbol{y}}|\boldsymbol{y}) = \frac{\mathbb{1}_\epsilon \left\{d(s(\hat{\boldsymbol{y}}), s(\boldsymbol{y}) < \epsilon)\right\} p(\hat{\boldsymbol{y}}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int \mathbb{1}_\epsilon \left\{d(s(\hat{\boldsymbol{y}}), s(\boldsymbol{y}) < \epsilon)\right\} p(\hat{\boldsymbol{y}}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}}, \tag{16}$$

where $\mathbb{1}_\epsilon(\cdot)$ is the indicator function, $d(\cdot)$ is a chosen distance metric, $\epsilon > 0$ and we consider the summary $s(\cdot)$ to be sufficient. The desired marginal posterior then follows as

$$p_\epsilon(\boldsymbol{\theta}|\boldsymbol{y}) = \int p_\epsilon(\boldsymbol{\theta}, \hat{\boldsymbol{y}}|\boldsymbol{y}) d\hat{\boldsymbol{y}}. \tag{17}$$

Note that the pseudo-data distribution $p(\hat{\boldsymbol{y}}|\boldsymbol{\theta})$ appearing in equation 16 is not required analytically in any of the ABC algorithms. This distribution is essentially the generative model under consideration.

For the HMM such a pseudo data is sampled from the distribution

$$p(\hat{\boldsymbol{y}}, \boldsymbol{x}|\boldsymbol{\theta}) = \left(\prod_{t=0}^{M-1} g(\hat{\boldsymbol{y}}_t|\boldsymbol{X}_t, \boldsymbol{\theta})\right)\left(\prod_{t=1}^{M-1} f(\boldsymbol{X}_t|\boldsymbol{X}_{t_1}, \boldsymbol{\theta})\right), \tag{18}$$

where $f(\cdot)$, $g(\cdot)$ and thus $p(\hat{\boldsymbol{y}}, \boldsymbol{x}|\boldsymbol{\theta})$ need not be analytically tractable, just a sample $\hat{\boldsymbol{y}}$ of the pseudo-data from this distribution is required. Sampling from this distribution is essentially the process of forward sampling from the generative model of the HMM given by equation 1 (see main text). Considering $\hat{\boldsymbol{y}}$ alone from the pair $(\hat{\boldsymbol{y}}, \boldsymbol{x})$ we have a sample of the pseudo-data drawn from its marginal $p(\hat{\boldsymbol{y}}|\boldsymbol{\theta})$. Thus, when ABC is applied to the HMM in equation 1 the joint density in equation 16 is replaced by a density over the triplet $(\boldsymbol{\theta}, \boldsymbol{x}, \hat{\boldsymbol{y}})$ given by

$$p_\epsilon(\boldsymbol{\theta}, \boldsymbol{x}, \hat{\boldsymbol{y}}|\boldsymbol{y}) = \frac{\mathbb{1}_\epsilon \left\{d(s(\hat{\boldsymbol{y}}), s(\boldsymbol{y}) < \epsilon)\right\} p(\hat{\boldsymbol{y}}, \boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int \mathbb{1}_\epsilon \left\{d(s(\hat{\boldsymbol{y}}), s(\boldsymbol{y}) < \epsilon)\right\} p(\hat{\boldsymbol{y}}, \boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}}, \tag{19}$$

from which samples of the pair $(\boldsymbol{\theta}, \boldsymbol{x})$ is distributed from $p_\epsilon(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y})$. And the corresponding ABC marginal posterior is given by

$$p_\epsilon(\boldsymbol{\theta}|\boldsymbol{y}) = \int p_\epsilon(\boldsymbol{\theta}, \boldsymbol{x}, \hat{\boldsymbol{y}}|\boldsymbol{y}) d\hat{\boldsymbol{y}} d\boldsymbol{x}. \tag{20}$$

From equation 19 it is evident that any ABC algorithm applied to the HMM will target the joint distribution $p_\epsilon(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y})$. However, this distribution will only be an approximation to the true posterior $p(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y})$, since $\epsilon \neq 0$ (considering $s(\cdot)$ to be sufficient). Note that since $\boldsymbol{x}$ is sampled from its prior thus if ϵ is set to zero (or a small value) then a practically infeasible amount of simulations is required to produce an ABC posterior $p(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y})$ that can approximate closely the true posterior.

A.2 Deriving the incremental posterior

Consider the posterior distribution of the sample path and the parameters (including the initial values) conditioned on the observations. We can write this density, upto a normalising constant, as follows

$$\begin{aligned} &p(\boldsymbol{X}_{M-1}, \ldots, \boldsymbol{X}_1, \boldsymbol{\theta}|\boldsymbol{y}) \propto \\ &p(\boldsymbol{\theta})\left(\prod_{t=0}^{M-1} g(\boldsymbol{y}_t|\boldsymbol{X}_t, \boldsymbol{\theta})\right)\left(\prod_{t=1}^{M-1} f(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}, \boldsymbol{\theta})\right). \end{aligned} \tag{21}$$

We can obtain from this the density of $\boldsymbol{X}_{M-1}$ conditioned on all other random variables by only retaining the terms that involve it. So we have this conditional density, upto a normalising constant, given by

$$p(\boldsymbol{X}_{M-1}|\boldsymbol{X}_{M-2}, \ldots, \boldsymbol{X}_1, \boldsymbol{\theta}, \boldsymbol{y}) \propto g(\boldsymbol{y}_{M-1}|\boldsymbol{X}_{M-1}, \boldsymbol{\theta}) f(\boldsymbol{X}_{M-1}|\boldsymbol{X}_{M-2}, \boldsymbol{\theta}) p(\boldsymbol{\theta}), \tag{22}$$

which is simply the density $p(\boldsymbol{X}_{M-1}|\boldsymbol{X}_{M-2}, \boldsymbol{y}_{M-1}, \boldsymbol{\theta})$.

We can also write the conditional distribution of any intermediate sample point $\boldsymbol{X}_t$ as follows:

$$\begin{aligned} &p(\boldsymbol{X}_t|\boldsymbol{X}_{M-1}, \ldots, \boldsymbol{X}_{t+1}, \boldsymbol{X}_{t-1}, \ldots, \boldsymbol{X}_1, \boldsymbol{\theta}, \boldsymbol{y}) \propto \\ &p(\boldsymbol{\theta})\left(\prod_{t=0}^{M-1} g(\boldsymbol{y}_t|\boldsymbol{X}_t, \boldsymbol{\theta})\right)\left(\prod_{i=1}^{M-1} f(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}, \boldsymbol{\theta})\right) \propto \\ &f(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t, \boldsymbol{\theta}) f(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}, \boldsymbol{\theta}) g(\boldsymbol{y}_t|\boldsymbol{X}_t, \boldsymbol{\theta}) p(\boldsymbol{\theta}), \end{aligned} \tag{23}$$

which is simply the density $p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}, \boldsymbol{X}_{t+1}, \boldsymbol{y}_t, \boldsymbol{\theta})$.

B ABC-SMC IMPLEMENTATION DETAILS

We applied the particular version of ABC-SMC algorithm, that was proposed in Toni et al. (2009), using 1000 particles. Furthermore, we used an adaptive tolerance sequence where the tolerance ϵ_τ at the τ-th step of the algorithm is selected as the 0.1-quantile of the distances of the accepted particles in the $\tau - 1$-th step. Moreover, we chose the perturbation kernel of ABC-SMC (see Toni et al. (2009)) as a multivariate Gaussian whose covariance is based on a *k-nearest neighbours* strategy, with $k = 15$, proposed in Filippi et al. (2013). We terminated the ABC-SMC algorithm when a predetermined number of simulations has been carried out. If that number is exceeded within the τ-th step, we then considered the weighted particle system at the $\tau - 1$-th step as the desired ABC posterior.

C MODEL DETAILS

C.1 STOCHASTIC LOTKA-VOLTERRA MODEL

The stochastic Lotka-Volterra model, a stochastic kinetic system, can be defined through the following list of reactions:

$$\begin{aligned} \mathcal{R}_1: &\quad X^{prey} \xrightarrow{c_1} 2X^{prey} \\ \mathcal{R}_2: &\quad X^{prey} + X^{pred} \xrightarrow{c_2} 2X^{pred} \\ \mathcal{R}_3: &\quad X^{pred} \xrightarrow{c_2} \emptyset, \end{aligned} \tag{24}$$

where we denote by X^{prey}, X^{pred} the prey and predator species respectively. We further denote the corresponding numbers of the species as the system state $\boldsymbol{X}_t = (X_t^{prey}, X_t^{pred})$. The hazard vector for this system is $h(\boldsymbol{X}_t, \boldsymbol{c}) = \left(c_1 X_t^{prey}, c_2 X_t^{prey} X_t^{pred}, c_3 X_t^{pred}\right)$. The stoichiometry matrix for this system is given by

$$S = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}. \tag{25}$$

We set the initial values as $\mathcal{X}_0 = (100, 100)$ and consider them known.

A MJP describing a stochastic kinetic system, like the one above or the PKY model, is characterised by the transition probability $p(t_0, \boldsymbol{X}_0, t, \boldsymbol{X}_t) := p(\boldsymbol{X}, t)$ for the process arriving at state $\boldsymbol{X}_t$ at time t conditioned on an initial state $\boldsymbol{X}_0$ at time t_0. This is basically the density $f(\cdot)$ in equation 1 (main text), in continuous time. Now this transition probability is given by the solution of the following differential equation:

$$\frac{\partial p(\boldsymbol{X}, t)}{\partial t} = \sum_{i=1}^{v} = \{h_i(\boldsymbol{X} - S^i, c_i) p(\boldsymbol{X} - S^i, t) - h_i(\boldsymbol{X}, c_i) p(\boldsymbol{X}, t)\}, \tag{26}$$

known as the chemical master equation (Golightly & Gillespie, 2013, and the references therein). The CME only admits an analytical solution for a handful of simple models (not for the ones we have used: LV and PKY). Thus $f(\cdot)$ in equation 1 (main text) cannot be evaluated. However, the seminal work in Gillespie (1977) developed an algorithm, commonly referred to as the *stochastic simulation algorithm*, that can simulate $\boldsymbol{X}$ exactly.

We generated simulated trajectories from this model using the stochastic simulation algorithm and added Gaussian noise corruption, with variance 100, at 50 time points. We used the following generative values of the parameters $\boldsymbol{\theta} = (0.3, 0.0025, 0.5)$ to ensure that the model follows an oscillatory regime. Moreover, following previous studies we considered the initial values to be known and set at $\boldsymbol{X}_{t_0} = (100, 100)$. For running ABC-SMC and all the NLFI methods we downsampled the generated time series by a factor of 5 to create a summary statistic $s(\boldsymbol{y}) \in \mathbb{R}^{20}$ which is used in place of the full data $\boldsymbol{y}$. For further details of the model and simulations see Appendix C. We used the following set of prior distributions: $c_1 \sim \text{Beta}(1, 2)$, $c_2 \times 10^3 \sim \mathcal{U}(15, 50)$ and $c_3 \sim \text{Beta}(2, 1)$.

C.2 PROKARYOTIC AUTOREGULATORY GENE NETWORK

We considered the autoregulatory model used to benchmark the particle MCMC method in Golightly & Wilkinson (2011). This is a simplified model that describes a mechanism for autoregulation in prokaryotes based on a negative feedback mechanism of dimers of a protein coded by a gene

repressing its own transcription. Essentially this is a stochastic kinetic model described by the following set of reactions:

$$\begin{aligned}
&\mathcal{R}_1 : DNA + P2 \rightarrow DNA \cdot P2 \\
&\mathcal{R}_2 : DNA \cdot P2 \rightarrow DNA + P2 \\
&\mathcal{R}_3 : DNA \rightarrow DNA + RNA \\
&\mathcal{R}_4 : RNA \rightarrow RNA + P \\
&\mathcal{R}_5 : 2P \rightarrow P2 \\
&\mathcal{R}_6 : P2 \rightarrow 2P \\
&\mathcal{R}_7 : RNA \rightarrow \emptyset \\
&\mathcal{R}_8 : P \rightarrow \emptyset.
\end{aligned} \tag{27}$$

We order the variables as $\boldsymbol{X} = (RNA, P, P2, DNA, DNA \cdot P2)$ leading to a stoichiometry matrix for the system:

$$S = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -2 & 2 & 0 & -1 & \\ -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{28}$$

and the associated hazard function is given by

$$h(\boldsymbol{X}, \boldsymbol{c}) = (c_1 DNA \times P2, c_2 DNA \cdot P2, c_3 DNA, c_4 RNA, c_5 P(P-1)/2, c_6 P2, c_7 RNA, c_8 P). \tag{29}$$

This model has one conservation law (Golightly & Wilkinson, 2011)

$$DNA \cdot P2 + DNA = k, \tag{30}$$

where k is the number of copies of this gene in the genome. Following Golightly & Wilkinson (2011) we use this relation to to remove $DNA \cdot P2$ from the model, replacing any occurrences of $DNA \cdot P2$ in rate laws with $k - DNA$. This leads to a reduced full-rank model with species $\boldsymbol{X} = (RNA, P, P2, DNA)$, stoichiometry matrix:

$$S = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -2 & 2 & & -1 & \\ -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{31}$$

and associated hazard function

$$h(\boldsymbol{X}, \boldsymbol{c}) = (c_1 DNA \times P2, c_2(k - DNA), c_3 DNA, c_4 RNA, c_5 P(P-1)/2, c_6 P2, c_7 RNA, c_8 P). \tag{32}$$

We consider k to be known and set to 10. Again we generated simulated trajectories from this model using the stochastic simulation algorithm.

Following Golightly & Wilkinson (2011), we considered the observations as a linear combination of the proteins $P, P2$ as follows:

$$y_t = P_t + 2P2_t + \epsilon_t, \tag{33}$$

where ϵ is assumed to be iid Gaussian noise. We generated 100 simulated observations from this model at times $t = [0 : .5 : 50]$ with generative rate constants $\boldsymbol{\theta} = (0.1, 0.7, 0.35, 0.2, 0.1, 0.9, 0.3, 0.1)$ and $\epsilon \sim \mathcal{N}(0, 4)$. In this case also we consider the initial values $\boldsymbol{X}_{t_0}$ to be known and set to $(8, 8, 8, 5)$. We downsampled the simulated data by a factor of five to obtain the summary statistics $s(\boldsymbol{y}) \in \mathbb{R}^{20}$. Furthermore, we placed a $\mathrm{Gamma}(2, 3)$ prior on all the rate constants.

C.3 The SIR Compartmental Model

The stochastic version of the SIR model, for a population of N_{pop} people, with states variables $\boldsymbol{X}_t = (S_t, I_t)$ can be defined using an Itô SDE:

$$d\boldsymbol{X}_t = \boldsymbol{a}(\boldsymbol{X}_t, \boldsymbol{\theta})dt + \sqrt{\boldsymbol{B}(\boldsymbol{X}_t, \boldsymbol{\theta})}d\boldsymbol{W}_t, \tag{34}$$

driven by a two-dimensional Brownian motion $\boldsymbol{W}_t$ with the following drift and diffusion terms (see Fuchs (2013) for derivation):

$$\begin{aligned} \boldsymbol{a}(\boldsymbol{X}_t, \boldsymbol{\theta}) &= \begin{bmatrix} -\beta S_t I_t \\ \beta S_t I_t - \gamma I_t \end{bmatrix}, \\ \boldsymbol{B}(\boldsymbol{X}_t, \boldsymbol{\theta}) &= \frac{1}{N_{pop}} \begin{bmatrix} \beta S_t I_t & -\beta S_t I_t \\ -\beta S_t I_t & \beta S_t I_t + \gamma I_t \end{bmatrix}, \end{aligned} \tag{35}$$

where the β and γ are the unknown rate of infection and recovery. We generated simulated trajectories by numerically solving this SDE using the Euler-Maruyama solver (Kloeden et al., 2012).

We simulated the SDE on a time interval $t = [0 : 0.2, 14]$, with generative values of $\boldsymbol{\theta} = (1.7, 0.6, 1 - 0.05)$, to generate the state trajectory. We consider the observations to be the prevalence of the latent infectious state I_t observed through a Poisson counting process. Thus, we have the observations as $p(y_t|\beta, \gamma, s_0) = \text{Poisson}(I_t)$, and placed the following priors: $\beta \sim \mathcal{U}(0.5, 2)$, $\gamma \sim \mathcal{U}(0.1, 1)$ and $1 - s_0 \sim \mathcal{U}(0.001,)$. We then downsampled the observation time series and retain 14 values as the summary statistics $s(\boldsymbol{y}) \in \mathbb{R}^{14}$.

D PLOTS OF PARAMETER POSTERIORS

In the subsequent plots Figure 4, 5 and 6 we compare the parameter estimates of the three models between NLFI based methods, SNLE/SRE, and ABC-SMC. Here we have considered the estimates for one of the 10 different simulated datasets. Note that the parameter estimates are reasonably close to each other and thus the estimate of the posterior predictive distribution is largely influenced by the estimates of the hidden states.

D.1 LOTKA-VOLTERRA

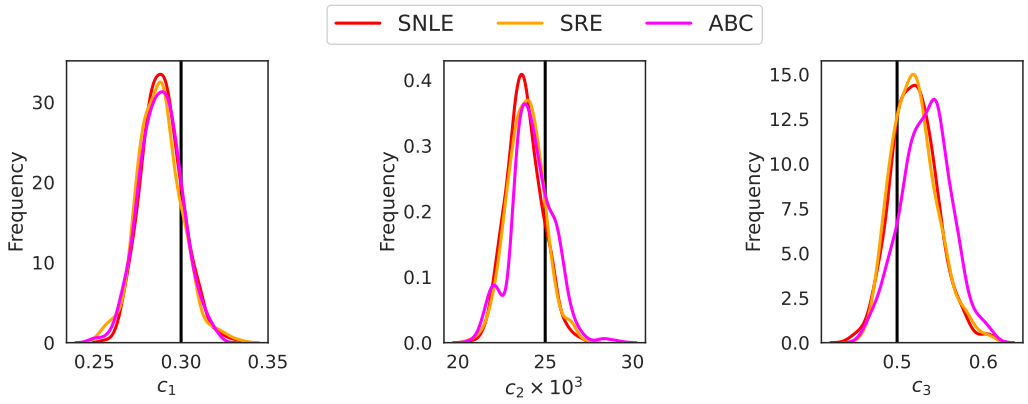


Figure 4: Posterior marginal densities of the parameters of the **Lotka-Volterra** model, inferred from one of the 10 datasets.

D.2 SIR MODEL

D.3 PROKARYOTIC AUTOREGULATORY GENE NETWORK

E EVALUATIONS WITHOUT USING SUMMARY STATISTICS

All our evaluations on the three biological HMMs were based on the use of hand-crafted summary statistics. Here we repeat the analysis for the PKY model without using summary statistics. For ABC-SMC this means calculating a distance between the full observed data (considering all the time points) and the simulated one. Note that the particular ABC-SMC algorithm that we have used

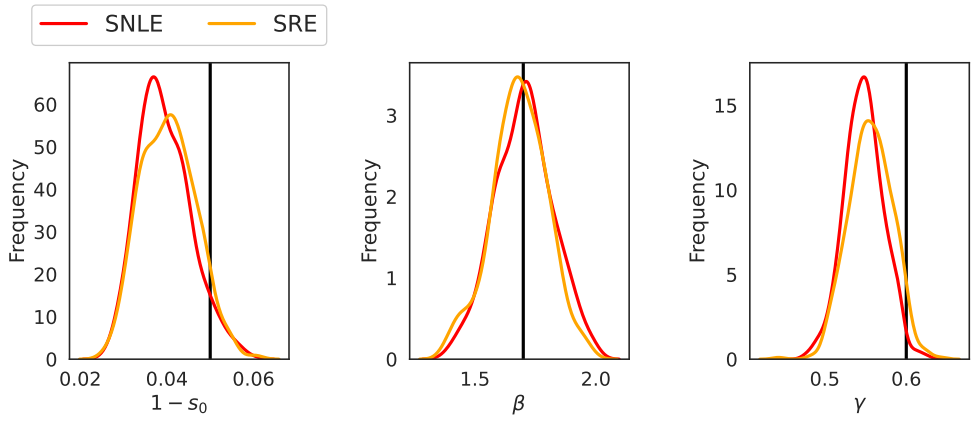


Figure 5: Posterior marginal densities of the parameters of the **SIR** model, inferred from one of the 10 datasets.

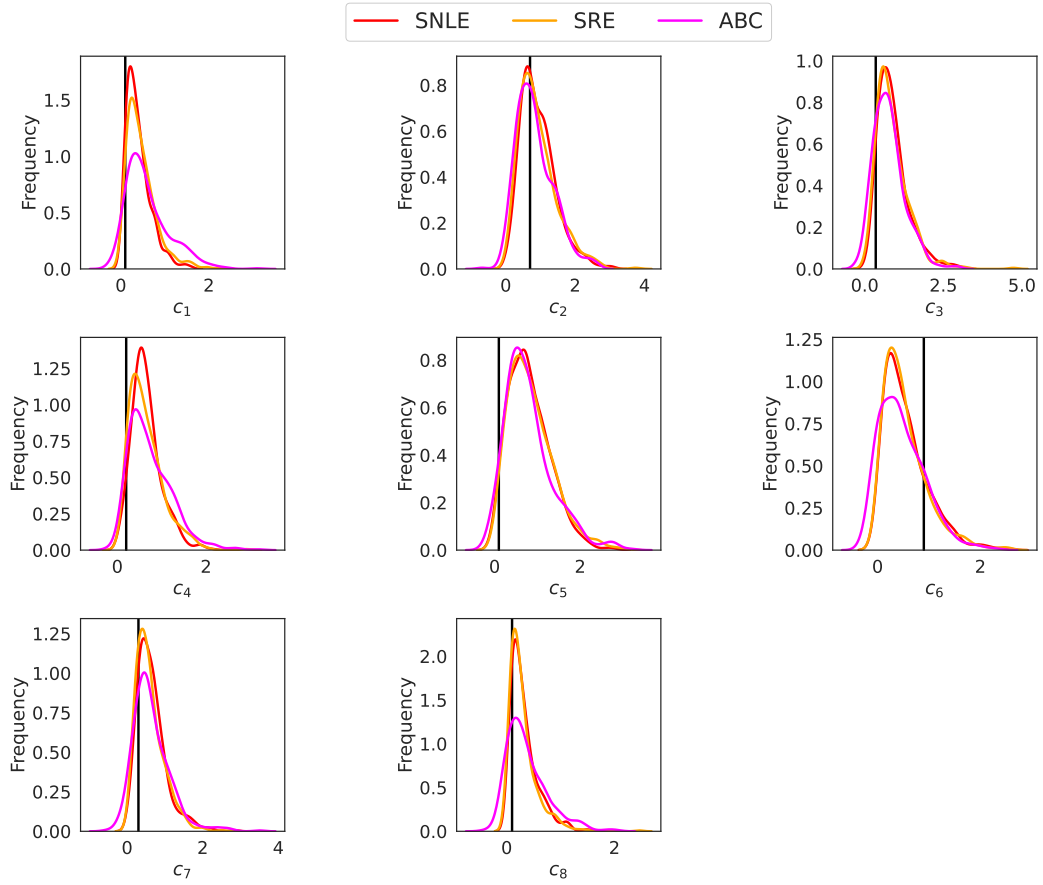


Figure 6: Posterior marginal densities of the parameters of the **Prokaryotic autoregulatory** model, inferred from one of the 10 datasets.

(Toni et al., 2009) was originally designed to work with full data. For obtaining the hidden states and subsequently the posterior predictive distribution using SMC, IDE and PrDyn we have used an estimate of $\boldsymbol{\theta}$ obtained using SRE trained on the full dataset. For this we extended the classifier neural

Table 2: The sum of pathwise MMD (smaller the better, given by equation 15) for the PKY model, when **SRE** and **ABC-SMC** are fitted to full data. Here the These MMDs are summarised by the **mean** $\pm$ **standard deviation** across 10 different simulated datasests.

COMPARISON OF THE ESTIMATE OF $p(\boldsymbol{y}^r|\boldsymbol{y})$ WITH **SMC** (BASELINE)

(WHEN $p(\boldsymbol{\theta}|\boldsymbol{y})$ FOR **SMC**, **IDE** AND **PRDYN** OBTAINED USING **SRE**)

MODEL	**IDE**	**PRDYN**	**ABC-SMC**
PKY	$\mathbf{4.4506 \pm 1.4192}$	17.0275 ± 1.7681	13.9841 ± 1.6488

network with a 2-layer LSTM, trained simultaneously with the classifier, to embed the data into a smaller dimensional summary statistics. We used a LSTM with a 10-dimensional hidden state and fed the hidden state, corresponding to the last time-step, into a fully connected layer consisting 8 hidden units and a ReLU activation function. Thus, we have a 8-dimensional summary statistics that is learnt on the fly.

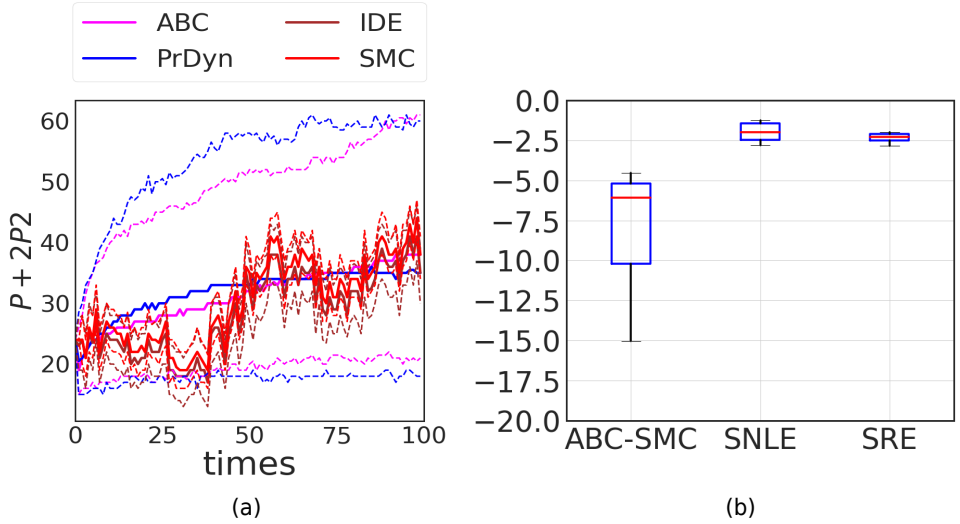


Figure 7: (a) Posterior distributions of the latent sample path $\boldsymbol{x}$ summarised by the mean (solid lines) and 95% credible intervals (broken lines), for the **Prokaryotic autoregulator**. The **ABC-SMC** is using the full dataset. (b) Accuracy of parameter estimates for the **Prokaryotic autoregulator** model, evaluated using the log probability of the true generative parameter vector, summarised across the 10 datasets. **SRE** and **ABC-SMC** is using the full dataset.

In Table 2 we furnish the pathwise MMDs. We again notice that the IDE producing an estimate of the posterior predictive distribution that is closest to the baseline (SMC's estimate). Additionally, we notice a slight improvement of ABC-SMC's performance in estimating the hidden states (see also Figure 7 (a) where we have plotted the estimated hidden states for one dataset), however the accuracy of the parameters estimates (summarised in Figure 7 (b)) does not improve significantly. Note that the accuracy of the parameter estimates did not improve significantly for the SRE as well. Despite having access to the full data the ABC-SMC's proposal mechanism for the hidden states is still too inefficient to significantly improve the accuracy of reconstructing the hidden states within a practically feasible simulation budget.

F EVALUATIONS USING A NONLINEAR GAUSSIAN STATE-SPACE MODEL

Here we want to evaluate how well the IDE can approximate the true incremental posterior (True-IP) $p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}, \boldsymbol{y}_t, \boldsymbol{\theta})$. This density is tractable for Gaussian state-space models. Thus, for this

evaluation we have chosen the following state-space model:

$$\begin{aligned} \boldsymbol{X}_t &\sim \mathcal{N}(\boldsymbol{A}\gamma(\boldsymbol{X}_{t-1}), \sigma_x^2 \mathbb{I}) \quad t \geq 1 \\ \boldsymbol{y}_t &\sim \mathcal{N}(\boldsymbol{B}\boldsymbol{X}_t, \sigma_y^2 \mathbb{I}), \end{aligned} \tag{36}$$

where $\gamma(\boldsymbol{X}) = \sin(\exp(\boldsymbol{X}_{t-1}))$, applied elementwise, $\boldsymbol{A} = \mathbb{I}_{K\times K}$, $B = 2\boldsymbol{A}$ and $\boldsymbol{X}_0 = \boldsymbol{0}$. We consider the dimensionailty of $\boldsymbol{X}_t$ and $\boldsymbol{y}_t$ to be the same, K. Moreover, we consider a relatively high-dimensional state-space by setting $K = 25$. We also consider the parameters $\boldsymbol{\theta} = (\sigma_x, \sigma_y, \boldsymbol{X}_0)$ to be fixed and known. Thus, we only focus on this distribution $p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}, \boldsymbol{y}_t)$. For the model above this incremental posterior distribution is known analytically and happens to be a Gaussian:

$$p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}, \boldsymbol{y}_t) = \mathcal{N}(\boldsymbol{X}_t; \boldsymbol{m}, \boldsymbol{\Sigma}), \tag{37}$$

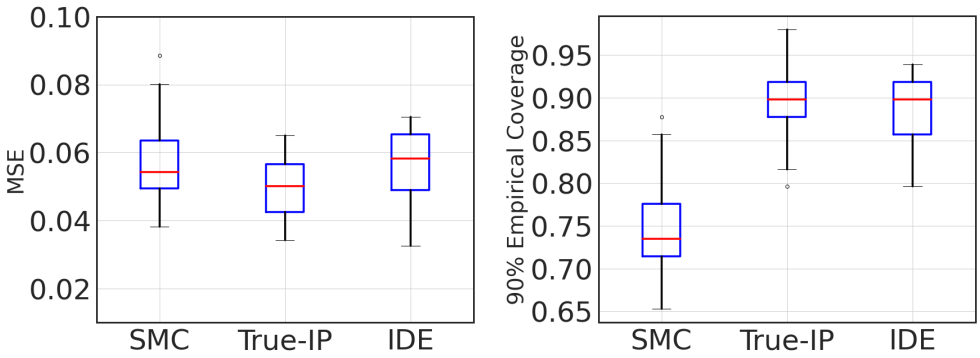
where the mean and the covariance are given by

$$\begin{aligned} \boldsymbol{\Sigma}^{-1} &= \Sigma_x^{-1} + B\Sigma_y^{-1}B \\ \boldsymbol{m} &= \boldsymbol{\Sigma}(\Sigma_x^{-1}\gamma(\boldsymbol{X}_{t-1}) + B\Sigma_y^{-1}\boldsymbol{y}_t), \end{aligned} \tag{38}$$

where $\Sigma_x = \sigma_x^2\mathbb{I}$ and $\Sigma_y = \sigma_y^2\mathbb{I}$.

Our goal is to primarily compare the proposed IDE with the True-IP. However, for completeness we also considered SMC for comparison. For this comparison we generated two sets of data corresponding to $M = 50$ and $M = 500$ time points. We used $\sigma_x = \sigma_y = 0.5$ to generate the simulated data. We then estimated the hidden states $\boldsymbol{X}_t$ using the IDE, the True-IP and SMC (that uses the true-IP as the importance proposal). For the SMC we used 100 particles and correspondingly we generated 100 samples from the True-IP and IDE. For the IDE's MAF we have used $J = 3$ transformations and trained it on $N = 500$ samples generated from the model. We found such a small sample size to be enough for learning a Gaussian density.

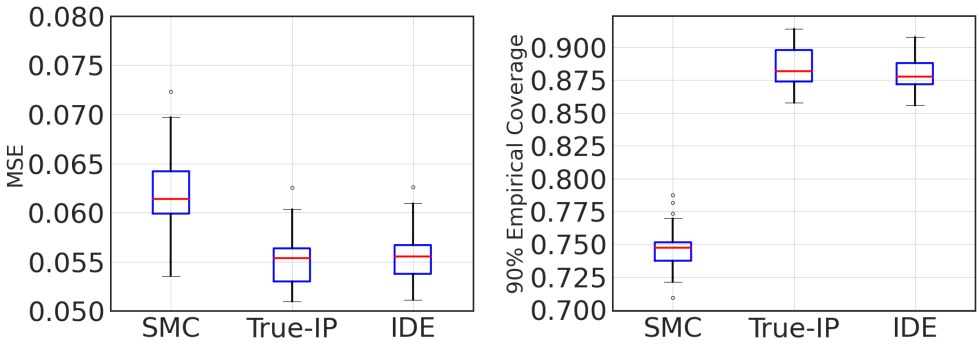
We compared the performance of all the methods using two metrics: (i) *mean squared error* (MSE) and (ii) 90% *empirical coverage* (EC). We computed these metric per dimension and summarised across them. In Figure 8 we summarise the metrics for the dataset consisting of $M = 50$ time points and in Figure 9 we do the same for the dataset consisting of $M = 500$ time points.



(a) (b)

Figure 8: Comparison between **True-IP**, **IDE** and **SMC** in recovering the hidden states of a 25-dimensional nonlinear Gaussian state-space model. Performance assessed in terms of (a) MSE and (b) EC. The above plots show the performance metrics corresponding to a dataset with $M = 50$ time points.

It is evident from both Figure 8 and 9 that the IDE produces a very close approximation of the True-IP, in a high-dimensional model and for a long time series. Interestingly the SMC performs slightly worse in terms of the coverage, which can be partially attributed to the difficulty of scaling importance sampling to higher dimensions. Increasing the number of particles improves both the metric, but usage of more particles require more model simulation in the context of our work. However, there are various strategies that can be adopted to improve the SMC's performance such as using an *auxiliary particle-filter* (Pitt & Shephard, 1999) or using MCMC within the SMC (Gilks & Berzuini, 2001), which we did not explore here.



(a) (b)

Figure 9: Comparison between **True-IP**, **IDE** and **SMC** in recovering the hidden states of a 25-dimensional nonlinear Gaussian state-space model. Performance assessed in terms of (a) MSE and (b) EC. The above plots show the performance metrics corresponding to a dataset with $M = 500$ time points.

G JOINT INFERENCE OF THE SAMPLE PATH AND PARAMETERS USING A MAF

We have argued before (see the last paragraph of section 3) that NLFI methods cannot be used directly for inferring the joint posterior $p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$. Next, we have shown results for an experiment, using the LV model, that supports our argument. Note that due to the unavailability of $p(\boldsymbol{x}, \boldsymbol{\theta})$, the only strategy that can be applied is of using a normalizing-flow to directly emulate the joint posterior $p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) \approx q_\psi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$. We denote this approach as neural posterior estimation (NPE). We used 10^6 simulations from the model to train a MAF representing $q_\psi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$. Note that for the proposed IDE approach we used 35×10^3 (including inference of $\boldsymbol{\theta}$). We retained the same architecture and optimisation settings that we used in other experiments. Once trained, we used one of the simulated dataset for the LV model to carry out inference. This is the same dataset corresponding to the plot shown in Figure 3(a).

In Figure 10 we plot components of the hidden state estimated by SMC, IDE, ABC-SMC and NPE. Note that SMC, IDE are using same samples of $\boldsymbol{\theta}$ estimated using SNLE. All methods use 500 samples from the posteriors of $\boldsymbol{\theta}, \boldsymbol{x}$. In Figure 11 we show the corresponding parameter estimates. Although NPE estimates the hidden state better than ABC-SMC, its estimation quality drops at those time points where the concentration reaches a peak before decreasing again. This drop is much more pronounced near the last peak. The parameter estimates are however significantly different than all the other methods. From which it can be concluded that NPE performs worse than even ABC-SMC to produce the posterior of the parameters when targeting $\boldsymbol{x}, \boldsymbol{\theta}$ jointly.

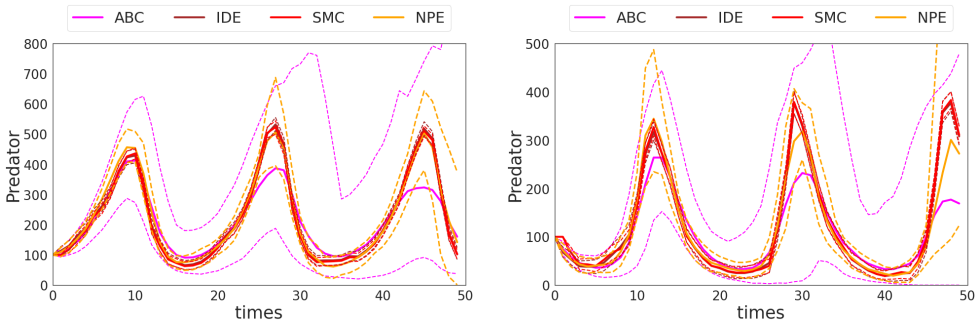


Figure 10: Comparison between methods that estimate jointly the parameters and hidden states of a HMM (in this case the **Lotka-Volterra** model), such as **ABC-SMC** & **NPE**, with those that estimate these quantities separately, such as **SMC** & **IDE**. The plot above shows the posteriors of the hidden states summarised by the mean (solid lines) and 95% credible intervals (broken lines). The proposed method **IDE** reduces the simulation burden by a large factor in comparison to **NPE**. Note that even with a much larger simulation budget **NPE** fails to correctly estimate the hidden states as well as the parameters (see Figure 11).

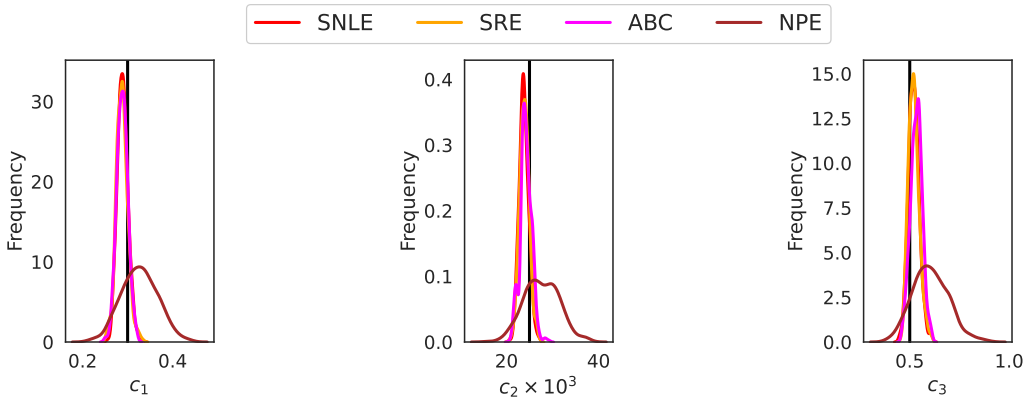


Figure 11: Posterior marginal densities of the parameters of the **Lotka-Volterra** model obtained using SNLE, SRE (both targeting the marginal $p(\boldsymbol{\theta}|\boldsymbol{y})$) with NPE, ABC-SMC (both targeting the joint $p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$). NPE failed to estimate $\boldsymbol{\theta}$ correctly.