Simulation-based inference of dynamical systems with model uncertainty

Nick Galioto and Alex Gorodetsky

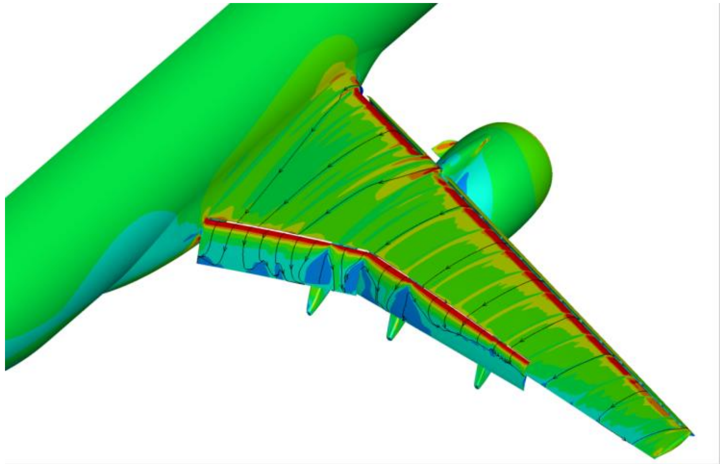
Department of Aerospace Engineering, University of Michigan

World Congress on Computational Mechanics (WCCM) in Vancouver, Canada

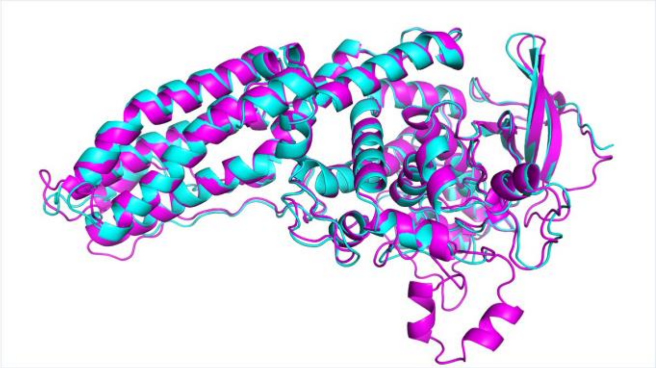
July 24, 2024

Motivation

- Dynamical systems arise in various fields and applications
  - Fluid flow
  - Financial markets
  - Biological systems
- Dynamics models are important tools for studying such systems
  - Forecasting, control, etc…
- When estimating dynamics models, accounting for model uncertainty can lead to several benefits
  - Improved generalization and accuracy
- Unfortunately, including model uncertainty also adds complexity
  - Posterior is costly to evaluate
  - Posterior is challenging to sample from



https://hiliftpw.larc.nasa.gov/index.html



https://www.technologyreview.com/2020/11/30/1012712/deepmind-protein-folding-ai-solved-biology-science-drugs-disease/

Simulation-based inference

- Simulation-based inference are a group of methods that draw samples from a *surrogate* distribution using simulations
- A simulator is a computer model that generates a stochastic realization of the model outputs

- **Question:** Can these methods be used to improve sampling in terms of speed and efficiency?

- **Objective:** Compare a filtering-based approach to two simulation-based approaches for likelihood estimation on an example problem

Outline

1. Filtering-based approach
2. Simulation-based inference
3. Results
4. Conclusions and future work

ngalioto@umich.edu

Introduction

Filtering-based approach

Simulation-based inference

Results

Conclusions and future work

Probabilistic formulation

The inclusion of process noise accounts for model uncertainty

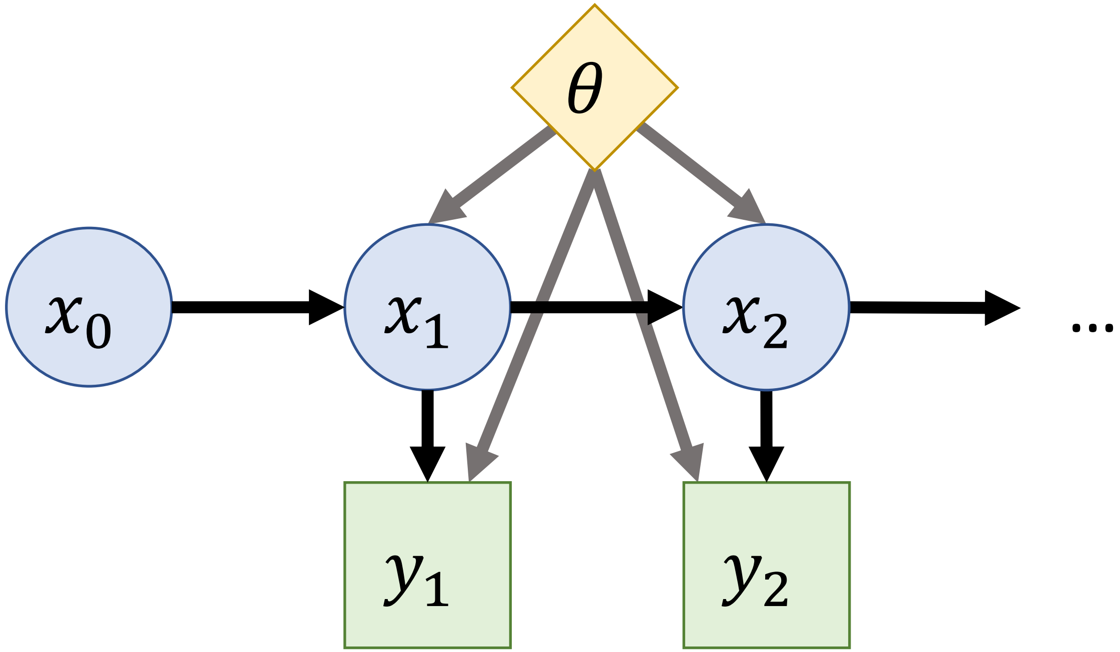
$$X_k \in \mathbb{R}^{d_x}, \qquad Y_k \in \mathbb{R}^{d_y}, \qquad \theta = (\theta_\Psi, \theta_h, \theta_\Sigma, \theta_\Gamma) \in \mathbb{R}^{d_\theta}$$

$$X_k = \Psi(X_{k-1}, \theta_\Psi) + \xi_{k-1}; \quad \xi_{k-1} \sim \mathcal{N}\big(0, \Sigma(\theta_\Sigma)\big)$$

$$Y_k = h(X_k, \theta_h) + \eta_k; \qquad \eta_k \sim \mathcal{N}\big(0, \Gamma(\theta_\Gamma)\big)$$

The process noise term ξ_k accounts for model error

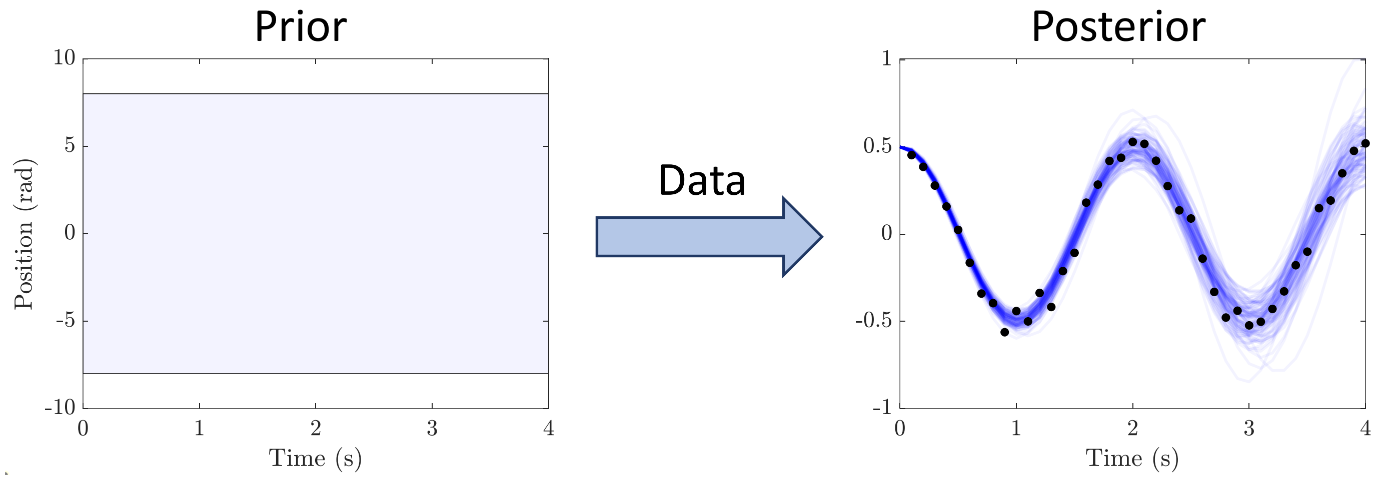
- Parameter error
- Integration error
- Insufficient model expressiveness



1. Parameter Uncertainty
2. Model Uncertainty
3. Measurement Uncertainty

Bayesian inference

- Goal: compute $p(\theta|\mathcal{Y}_T)$ where $\mathcal{Y}_T = (y_1, y_2, \dots, y_T)$
- Bayes' rule: $p(\theta|\mathcal{Y}_T) = \dfrac{\mathcal{L}(\theta; \mathcal{Y}_T)p(\theta)}{p(\mathcal{Y}_T)}$



- Due to uncertainty in the states, we can only access the joint likelihood: $\mathcal{L}(\theta;\ \mathcal{X}_T, \mathcal{Y}_T)$
- To get the marginal likelihood, we must evaluate the integral

$$\mathcal{L}(\theta;\ \mathcal{Y}_T) = \int \mathcal{L}(\theta;\ \mathcal{X}_T, \mathcal{Y}_T)\, d\mathcal{X}_T$$

ngalioto@umich.edu

Bayesian system identification algorithm

MCMC

for $i = 1, \dots, M$

Propose sample θ

<u>Evaluate posterior:</u> $\pi(\theta|\mathcal{Y}_T) = \pi(\theta)\prod_{k=1}^{T}\mathcal{L}_k(\theta;\mathcal{Y}_k)$

Bayesian filtering

for $k = 1, \dots, T$

Predict: $\pi(\mathbf{x}_{k+1}|\mathcal{Y}_k,\theta) = \int \pi(\mathbf{x}_{k+1}|\mathbf{x}_k,\theta)\pi(\mathbf{x}_k|\mathcal{Y}_k,\theta)\mathrm{d}\mathbf{x}_k$

Marginalize: $\mathcal{L}_{k+1}(\theta;\mathcal{Y}_{k+1}) = \int \pi(\mathbf{y}_{k+1}|\mathbf{x}_{k+1},\theta)\pi(\mathbf{x}_{k+1}|\mathcal{Y}_k,\theta)\mathrm{d}\mathbf{x}_{k+1}$

Update: $\pi(\mathbf{x}_{k+1}|\mathcal{Y}_{k+1},\theta) = \dfrac{\pi(\mathbf{y}_{k+1}|\mathbf{x}_{k+1},\theta)\pi(\mathbf{x}_{k+1}|\mathcal{Y}_k,\theta)}{\pi(\mathbf{y}_{k+1}|\mathcal{Y}_k,\theta)}$

end for

Accept θ with Metropolis-Hastings probability; otherwise reject

end for

Särkkä, S. (2013). *Bayesian filtering and smoothing* (No. 3). Cambridge University Press.

The computational expense of filtering can quickly become limiting

The computational cost is on the order

$$\mathcal{O}\left(T\left(d_x^3 + d_y^3 + d_xF\right)\right)$$

where F is the cost of one timestep of forward model/simulator

Goal: Replace the filter with a simulation-based surrogate likelihood to significantly reduce the computational cost

Introduction
Filtering-based approach

Simulation-based inference

Results
Conclusions and future work

Bayesian synthetic likelihood (BSL)

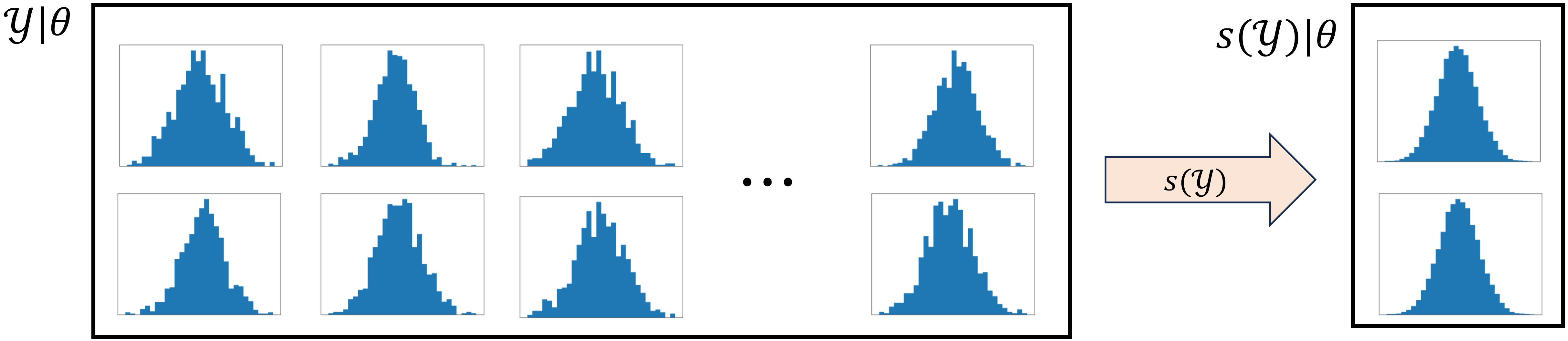
Wood, Simon N. "Statistical inference for noisy nonlinear ecological dynamic systems." *Nature* 466.7310 (2010): 1102-1104.

- The likelihood is non-Gaussian

$$p(\theta|\mathcal{Y}) \propto p(\mathcal{Y}|\theta)p(\theta)$$

- Assume we can map the data to a collection of summary statistics that are approximately Gaussian
  - E.g., coefficients of polynomial regression
- Replace the non-Gaussian likelihood with the Gaussian *synthetic likelihood* $p(s(\mathcal{Y})|\theta)$

$$p(\theta|\mathcal{Y}) \approx p(\theta|s(\mathcal{Y})) \propto p(s(\mathcal{Y})|\theta)p(\theta)$$



Choice of summary statistics

Maraia, Ramona, et al. "Bayesian synthetic likelihood for stochastic models with applications in mathematical finance." *Frontiers in Applied Mathematics and Statistics* 9 (2023): 1187878.

Donsker's theorem:

Guarantees asymptotic Gaussianity of the empirical cumulative distribution function (eCDF) vectors

Select a set of features $\mathcal{Y}_T = (y_1, \dots, y_T)$, $\mathrm{d}\mathcal{Y}_T = (y_2 - y_1, \dots, y_T - y_{T-1})$

Select a collection of d_{bins} bins $\mathcal{B}$

Construct eCDF vectors $s(\mathcal{Y}_T) \in \mathbb{R}^{d_{bins}}$

Bayesian synthetic likelihood (BSL)

Input:

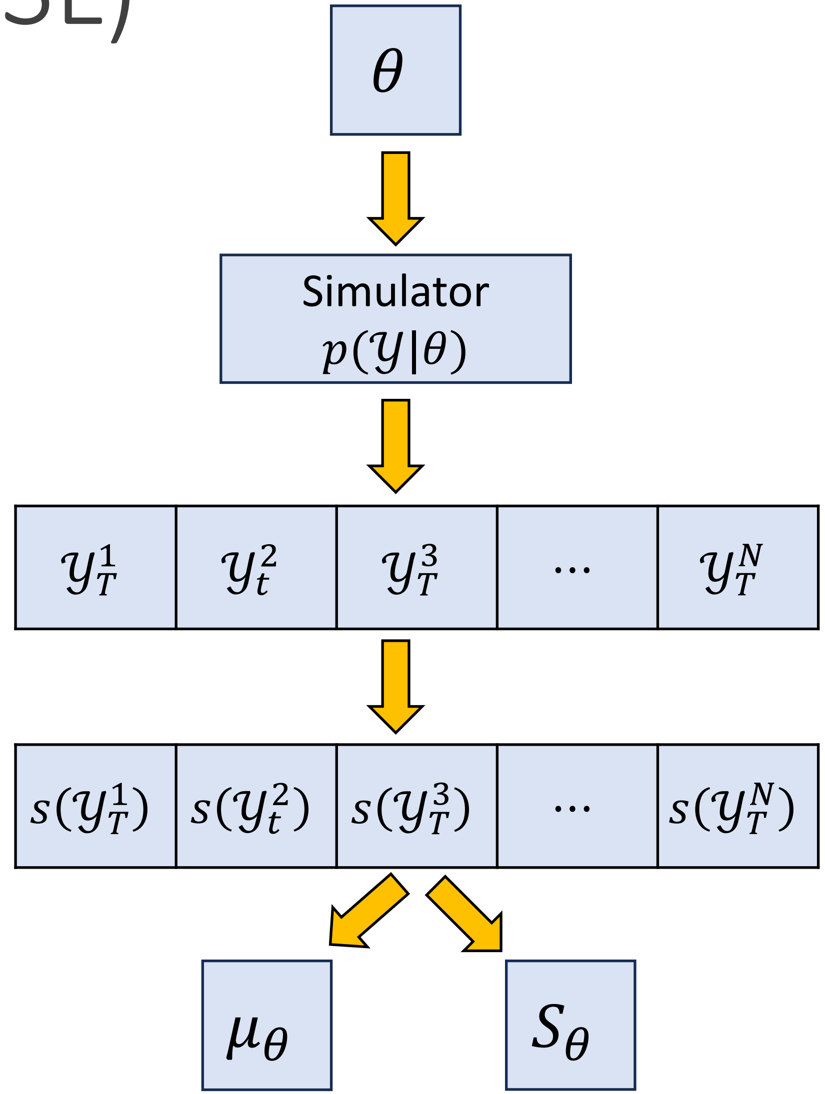
Parameter value θ

Number of simulations N

Output:

Evaluation of synthetic likelihood

1. Draw $\mathcal{Y}_T$ samples $\mathcal{Y}_T \sim p(\mathcal{Y}_T|\theta)$
2. Compute summary statistics $s(\mathcal{Y}_T)$
3. Compute empirical mean μ_θ and covariance S_θ
4. Evaluate $\mathcal{N}(\bar{\mathcal{Y}}_T; \mu_\theta, S_\theta)$



Sequential neural likelihood (SNL)

- The Gaussian assumption can be limiting in many situations
- The neural likelihood uses a neural network, e.g., a normalizing flow, to model the likelihood density

- Requires a proposal distribution to draw samples to train the network
- To achieve convergence, these samples should come from high density regions of the posterior
- SNL iteratively improves its proposal to target high density regions

Sequential neural likelihood (SNL)

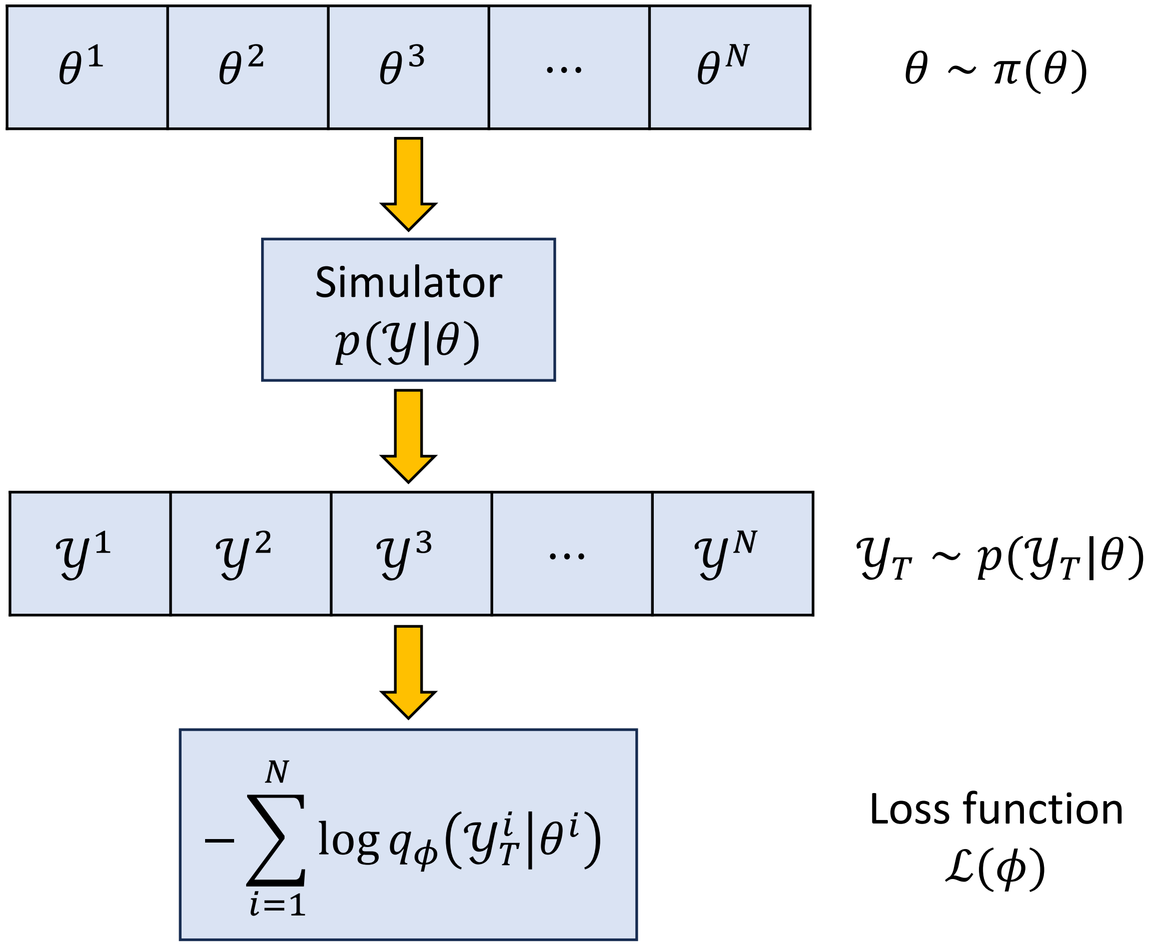
Input:

Initial proposal $\pi_0(\theta)$
Density model q_ϕ
Number of simulations N
Number of training rounds R

Output:

Density $q_\phi(\mathcal{Y}_T|\theta) \approx p(\mathcal{Y}_T|\theta)$

1. Set proposal $\pi(\theta) = \pi_0(\theta)$
   For $round = 1, \dots, R$:
2. Draw θ samples $\theta \sim \pi(\theta)$
3. Draw $\mathcal{Y}_T$ samples $\mathcal{Y}_T \sim p(\mathcal{Y}_T|\theta)$
4. Train $q_\phi(\mathcal{Y}_T|\theta) \approx p(\mathcal{Y}_T|\theta)$
   If $round < R$:
5. Set proposal $\pi(\theta) \propto q_\phi(\mathcal{Y}_T|\theta)p(\theta)$



Schematic of training round (lines 2—4)

Introduction
Filtering-based approach
Simulation-based inference

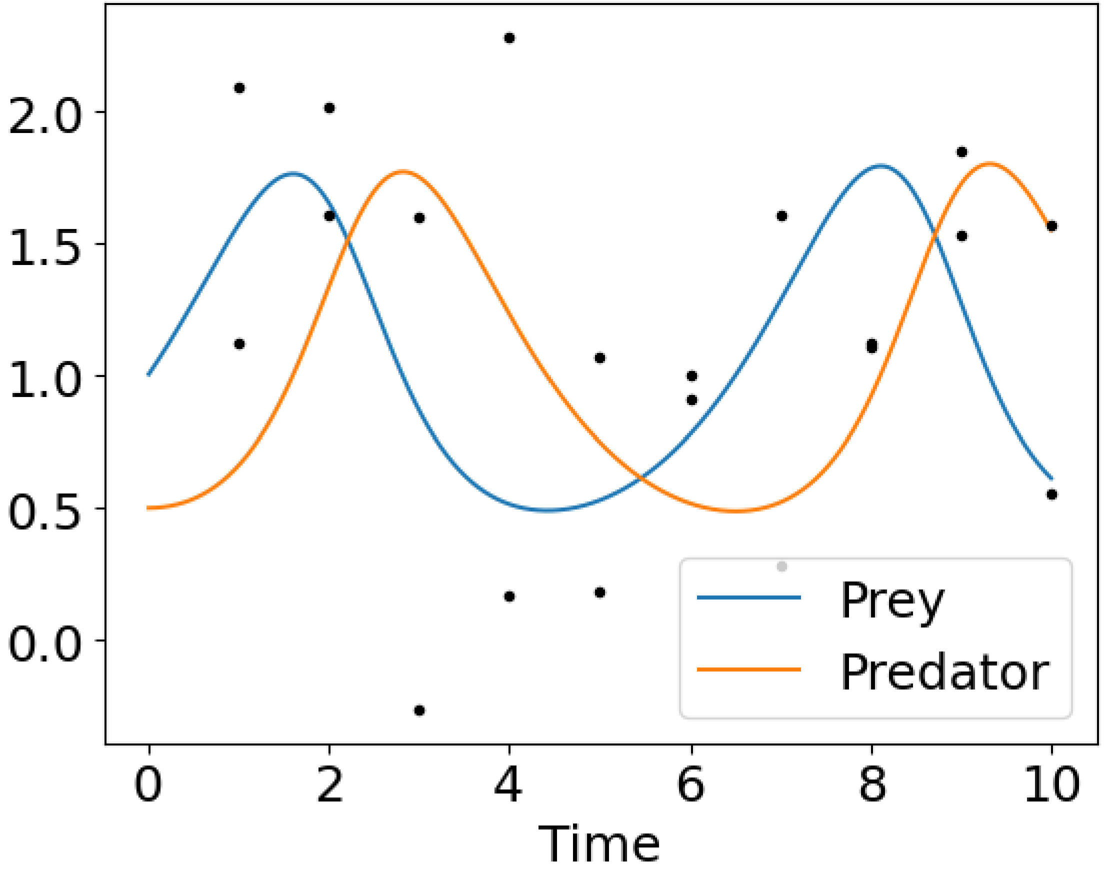
Results

Conclusions and future work

Lotka-Volterra

$$\dot{x} = \alpha x - xy$$

$$\dot{y} = \beta xy - y$$



Model setup:

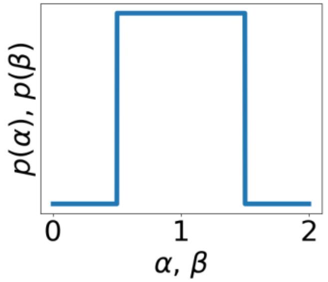
Euler-Maruyama with $\Delta t = 0.01$
Data at times $t = 1s, 2s, \ldots, 10s$
Measurement noise: $\mathcal{N}(0, 0.5^2)$

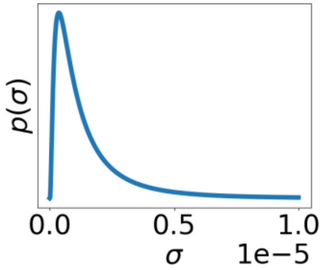
Learnable parameters:

$\theta = [\alpha, \beta, \sigma]$,
where $\Sigma = \sigma^2 I_2$

Priors:

$\alpha, \beta \sim \mathcal{U}[0.5, 1.5]$,
$\sigma \sim lognormal(10^{-6}, 1)$

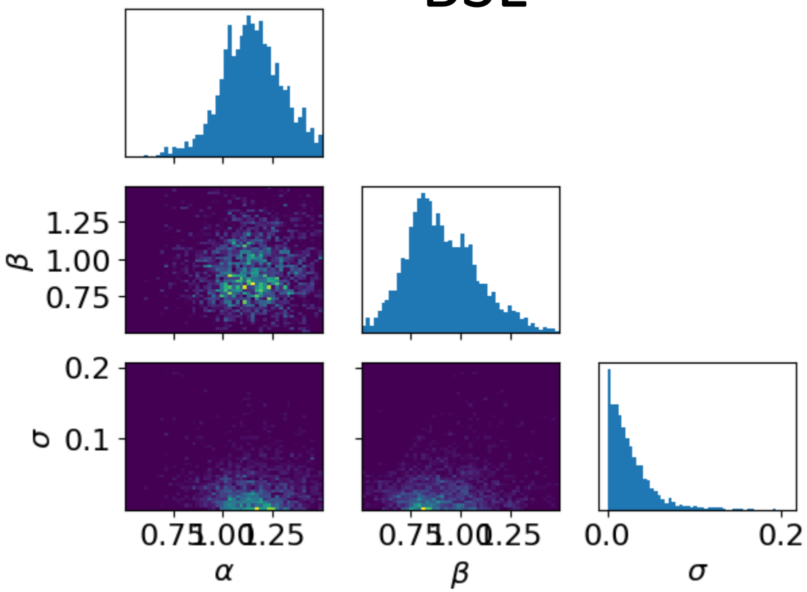




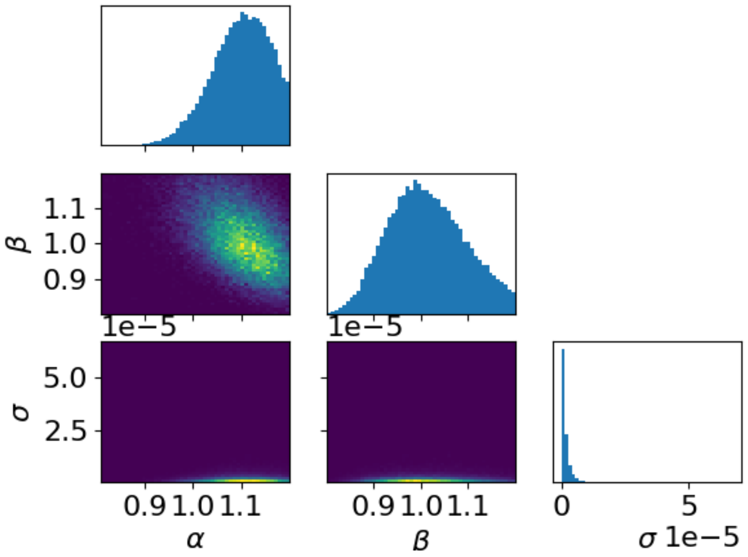
Distributions

SNL visually matches filter with better sampling

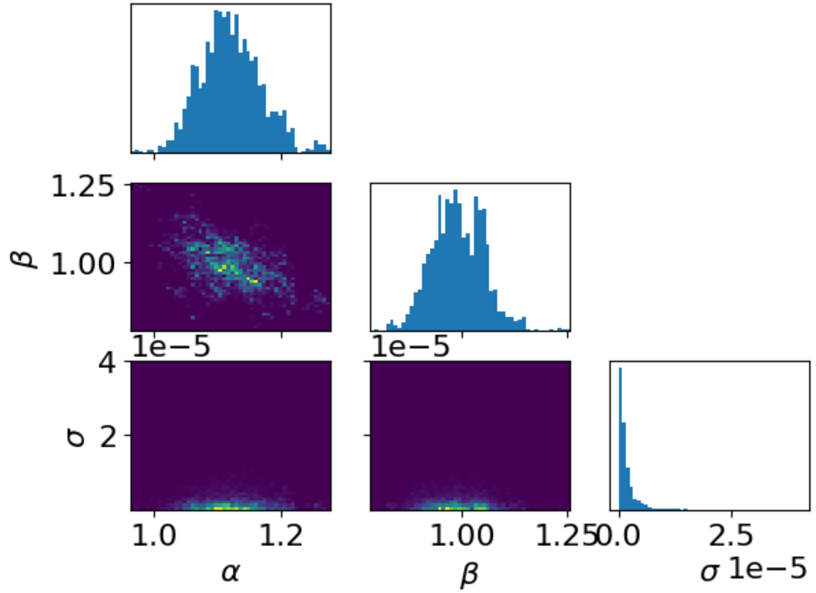
BSL



SNL



Filter

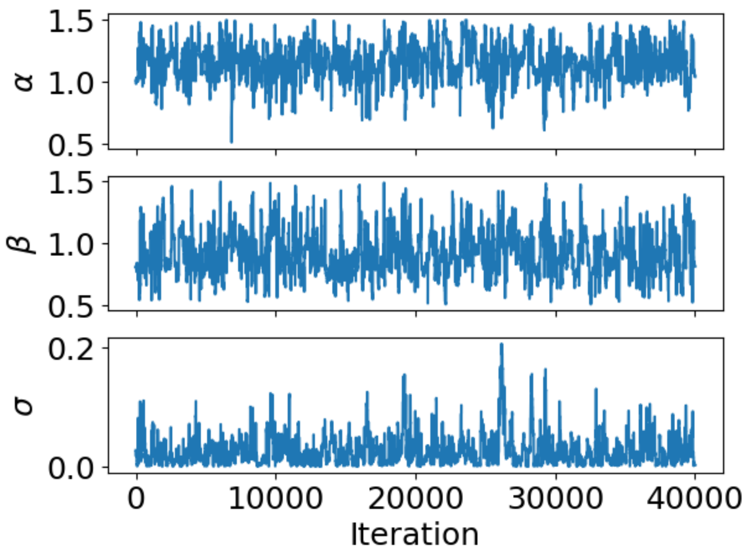


Chains and time comparison

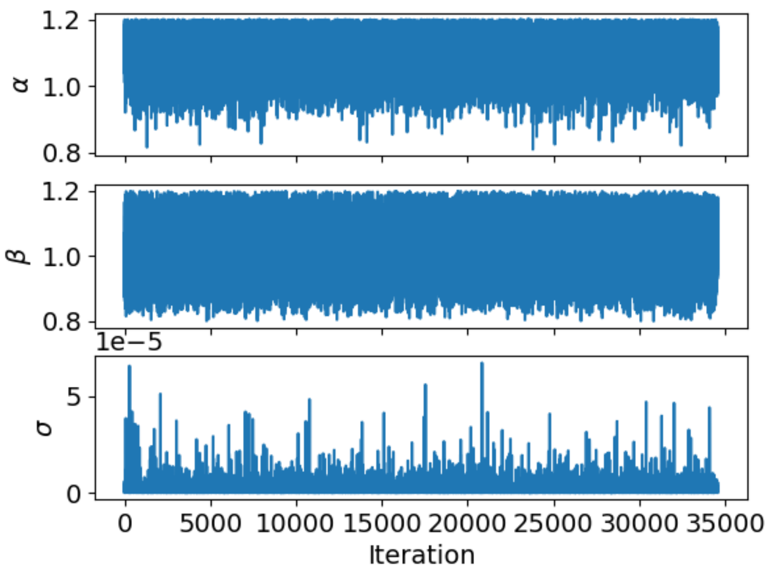
The SNL is fastest by a wide margin

Algorithm	Total time (h:mm:ss)
BSL	4:47:36
SNL	0:17:20
Filter	2:05:42

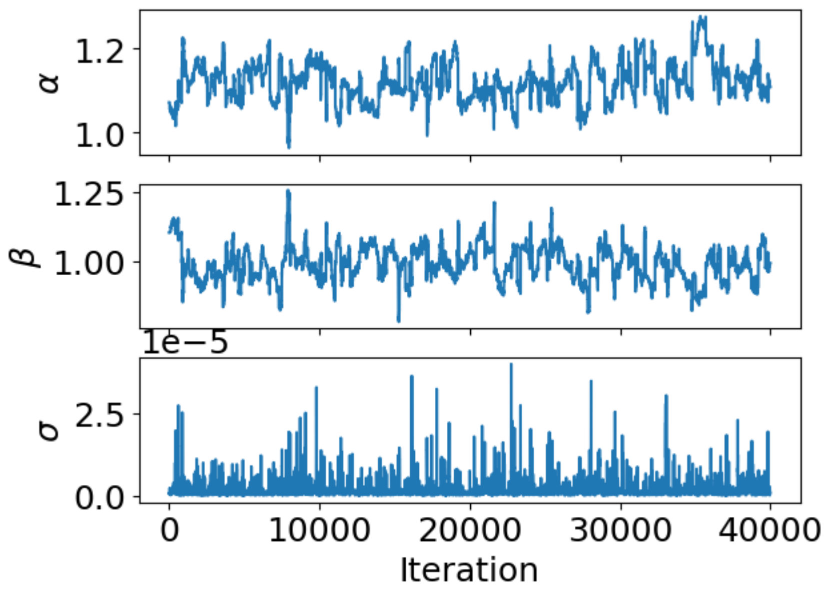
BSL



SNL

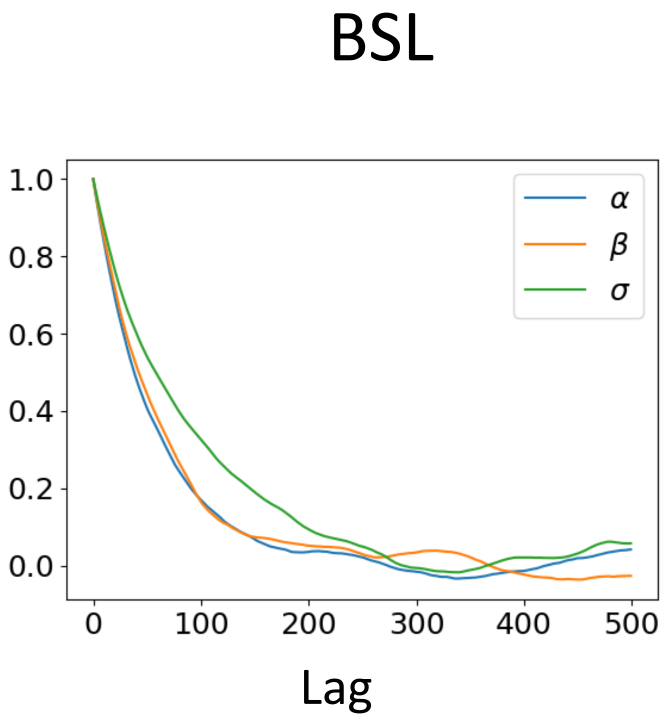


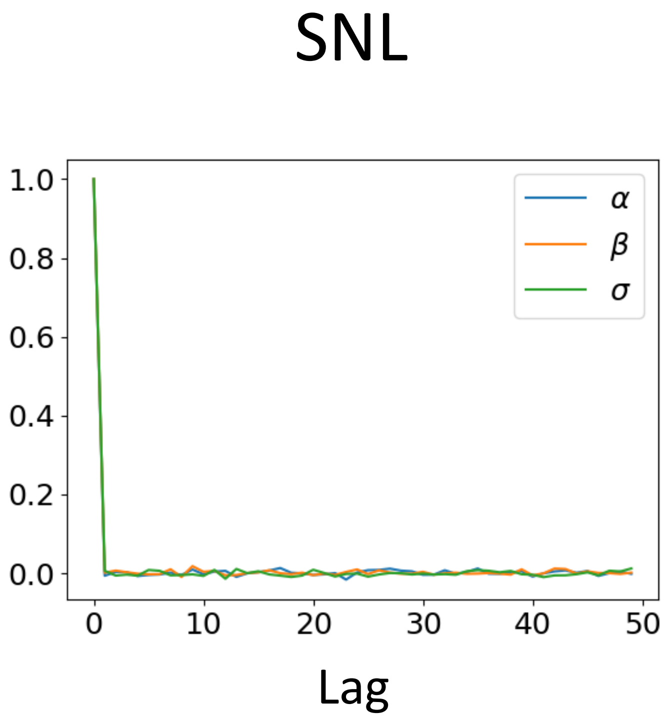
Filter

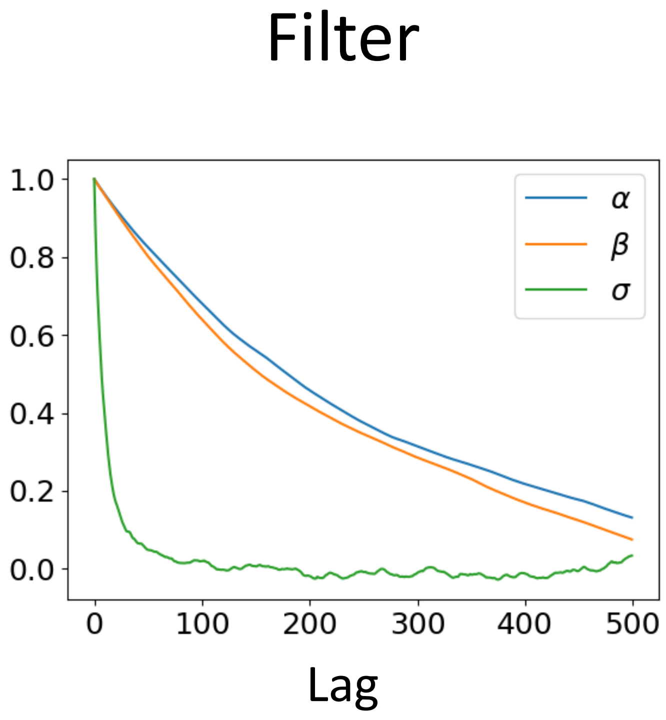


Autocorrelation plots

The SNL shows the best sample efficiency







Introduction
Filtering-based approach
Simulation-based inference
Results

Conclusions and future work

Conclusions

- Simulation-based methods show potential for accelerating posterior sampling in estimating dynamics models
- The SNL showed the best performance on the example problem in terms of speed and sampling efficiency
- The BSL method is potentially better-suited for high dimensions

Future work

- Investigate discrepancies in sampling efficiency between the methods
- Analyze the differences in performance of simulation-based methods when process noise is and is not included

References

Galioto, Nicholas, and Alex Arkady Gorodetsky. "Bayesian system ID: optimal management of parameter, model, and measurement uncertainty." *Nonlinear Dynamics* 102.1 (2020): 241-267.

Maraia, Ramona, et al. "Bayesian synthetic likelihood for stochastic models with applications in mathematical finance." *Frontiers in Applied Mathematics and Statistics* 9 (2023): 1187878.

Papamakarios, George, David Sterratt, and Iain Murray. "Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows." *The 22nd international conference on artificial intelligence and statistics*. PMLR, 2019.

Tejero-Cantero, Alvaro, et al. "SBI--A toolkit for simulation-based inference." *arXiv preprint arXiv:2007.09114* (2020).

Funding

AFOSR Program in Computational Mathematics FA9550-19-1-0013

ngalioto@umich.edu