The Frontier of Simulation-based Inference

Kyle Cranmer, Johann Brehmer & Gilles Louppe

Motivation

Many scientific domains have developed complex simulators

Examples: protein folding, economics, climate science, astrophysics

High fidelity samples, but likelihood intractable (*implicit models*)

Use summary statistics to compare observed and simulated data

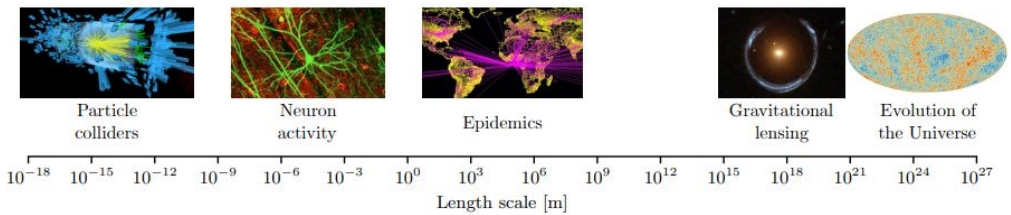


Fig. 1. Examples of phenomena at various length scales described by a diverse set of simulators, each with an intractable likelihood. Contains image material from Refs. (5–9).

Simulators

Probabilistic program (involves sampling)

Takes as input a vector of parameters θ - affects transition probabilities, typically fixed and interpretable (e.g. virulence rate of pathogen)

Samples a series of latent variables $z_i \sim p_i(z_i|\theta, z_{<i})$ - in/directly corresponds to physical state, continuous/discrete, dimensionality not fixed, typically unobserved

Produces a data vector $x \sim p(x|\theta, z)$ as output - observations, low/high-dimensional, un/structured

Inference

Inference of θ can be frequentist/Bayesian, point estimate/distribution

Frequentist $\rightarrow$ confidence set

Bayesian $\rightarrow$ posterior $p(\theta|x) = p(x|\theta)p(\theta) / \int p(x|\theta')p(\theta') \, d\theta'$

For both approaches, intractability of likelihood $p(x|\theta) = \int p(x, z|\theta) \, dz$ - requires all possible traces of the simulator

If observations are i.i.d., can factorise, but may not be (e.g. time series)

For Bayesian approaches, intractability of evidence $p(x)$ handled via MCMC/VI

Approximate Bayesian Computation (ABC)

Approximate likelihood by simulations, comparing with observed data

Rejection ABC:

1. $\theta \sim p(\theta)$
2. $x_{sim} \sim p(\cdot|\theta)$
3. Keep θ as posterior sample if $\rho(x_{sim}, x_{obs}) < \epsilon$, where ρ is distance measure

As $\epsilon \rightarrow 0$, inference with ABC becomes exact, but acceptance probability vanishes

Inference for new observations requires repeating the entire inference algorithm

Density Estimation

Use histograms/KDE, then proceed with standard inference

Amortised - new observations can be evaluated efficiently after upfront cost of density estimation

Challenges

Sample efficiency - particularly curse of dimensionality

Require low-dimensional summary statistics y(x) to scale to high-dimensional x

Quality of inference - loss of information (large ϵ or KDE bandwidth)

Amortization

Advances

ML allows working with high-dimensional observations

Active learning improves sample efficiency

Exposing internals of simulator to inference (no longer a black box)

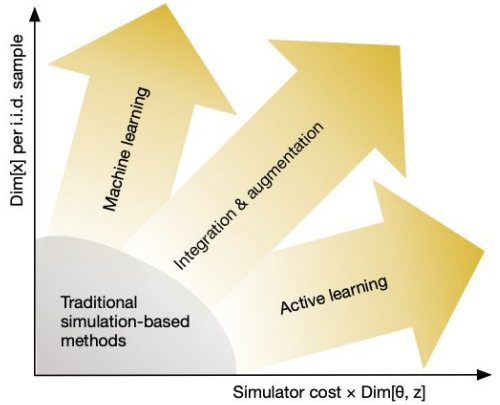


Fig. 2. A schematic illustration of how machine learning, active learning, and integration of automatic differentiation and probabilistic programming into the simulation code are expanding the frontier of traditional approaches to simulation-based inference.

Deep Learning

Deep learning allows scale to high-dimensional data via hierarchy, compositionality, *inductive biases* for data structures - CNNs for images, GNNs for graphs, ...

Normalising flows - transform variables from simple base distribution p(u) with parameterized invertible transformation $x = g_\varphi(u)$ that has a tractable Jacobian; $p_g(x)$ is given by the change-of-variables formula; trained with MLE

Autoregressive models - $p(x) = \prod_i p_i(x_i|x_{<i})$

GANs - intractable density, but very expressive; trained with likelihood ratio trick

Active Learning

Run simulator at θ points that maximise expected information gain

Applied to Bayesian posteriors and frequentist confidence sets

Related to decision making, experimental design and reinforcement learning

Integration and Augmentation

Probabilistic programming - can condition values of variables in the program on observations

Autodiff - automatically calculate derivatives wrt variables

Augment training data with additional information

Additional Quantities

1. $p(x|z, \theta)$ - probability density of the outputs given the latents
2. $t(x, z|\theta) \equiv \nabla_\theta \log p(x, z|\theta)$ - joint score (gradient of the joint log probability density of outputs and latents wrt parameters)
3. $\nabla_z \log p(x, z|\theta)$ - gradient of the joint log probability wrt latents
4. $r(x, z|\theta, \theta') \equiv p(x, z|\theta) / p(x, z|\theta')$ - probability ratio for two parameter points θ and θ'
5. $\nabla_\theta(x, z)$ - derivative of outputs and latents wrt parameters
6. $\nabla_z x$ - gradient of the outputs wrt latents

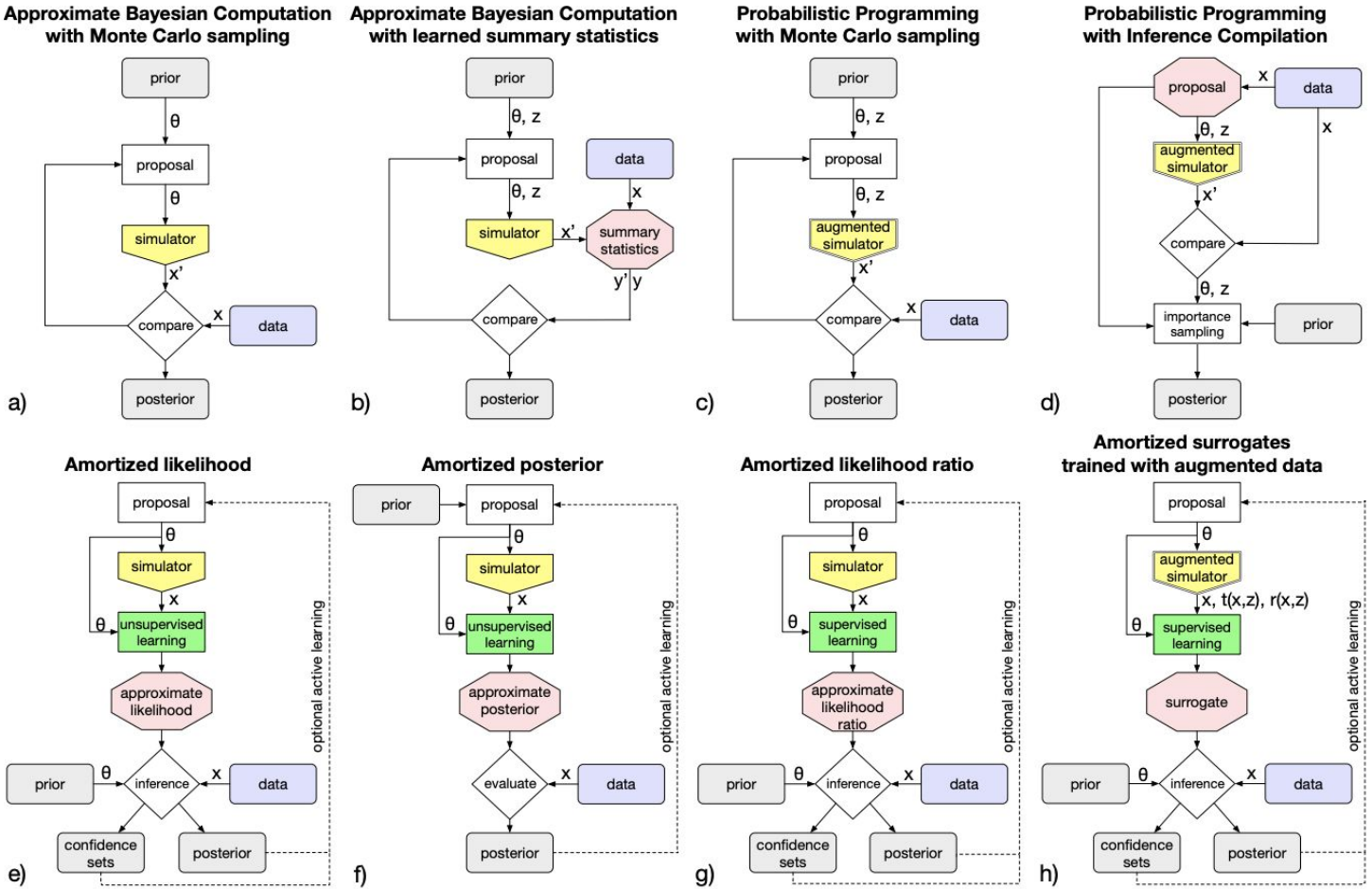


Fig. 3. Overview of different approaches to simulation-based inference.

Using Simulator Directly During Inference

Rejection ABC - expensive and sample inefficient

Use learned classifier to estimate discrepancy between observed and simulated data

Use active learning to sample better parameter points θ

Build simulator using probabilistic programming, requires tractable $p(x|z, \theta)$ or ABC

Inference engine controls execution and can bias samples of z to better match the observed data - improves sample efficiency

Allows more efficient inference for z

ABC with MC sampling

Run simulator with θ ~ proposal

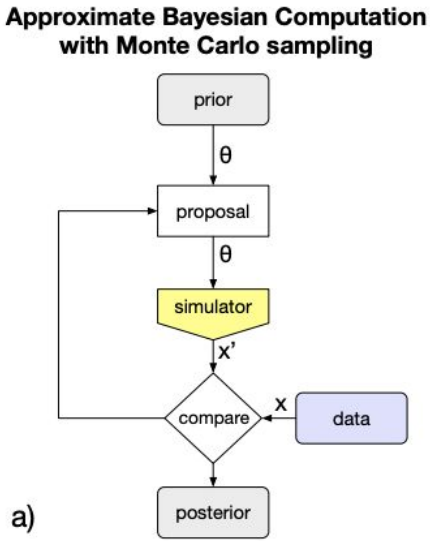
(dependent on prior when Bayesian)

Compare simulated x’ directly with data

x

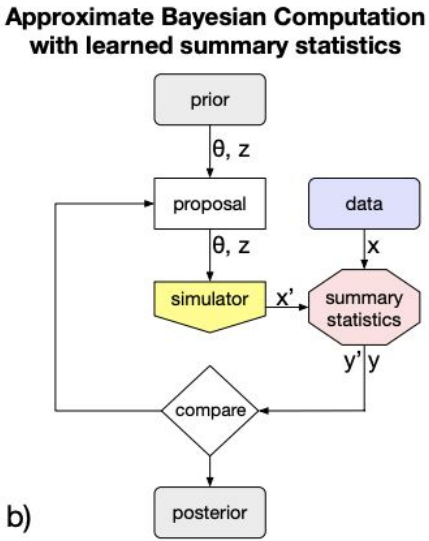
Add θ to posterior based on rejection

sampling



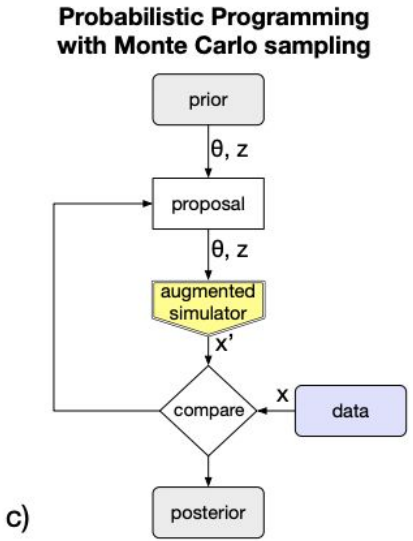
ABC with learned summary statistics

Perform ABC with learned summary statistics y(x)



PP with MC sampling

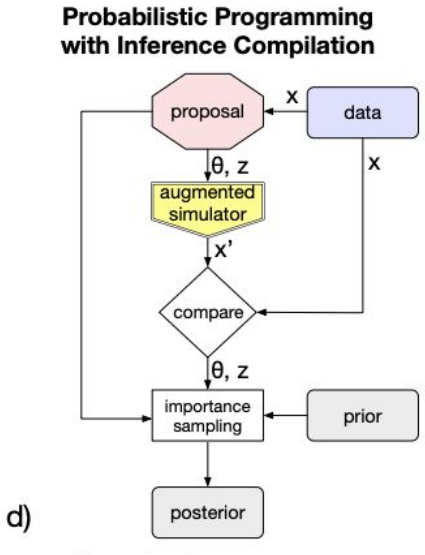
Can draw samples of θ, z from the posterior p(θ, z|x) using MC sampling



PP with Inference Compilation

Use ML to learn proposal model for z, θ

Correct using importance sampling



Surrogate Models

Allows amortised inference

Model implicitly marginalizes over latent variables from simulator

Removes requirement for low-dimensional summary statistics

Can impose useful inductive biases, even reflect causal structure of simulator

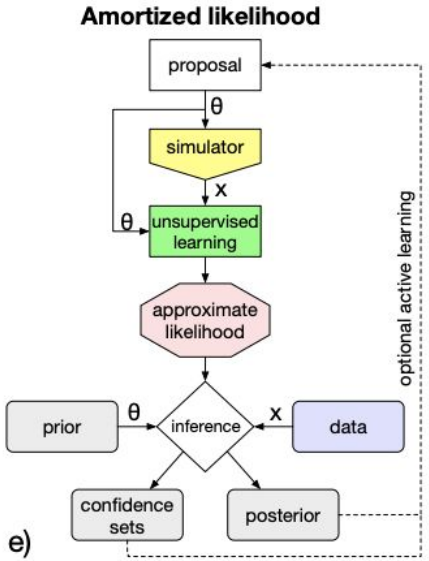
Amortised likelihood

Train conditional density estimator for likelihood p(x|θ)

Can sample new x

Enables frequentist inference; Bayesian inference requires additional MCMC/VI step for sampling from the posterior

Allows changing prior during inference



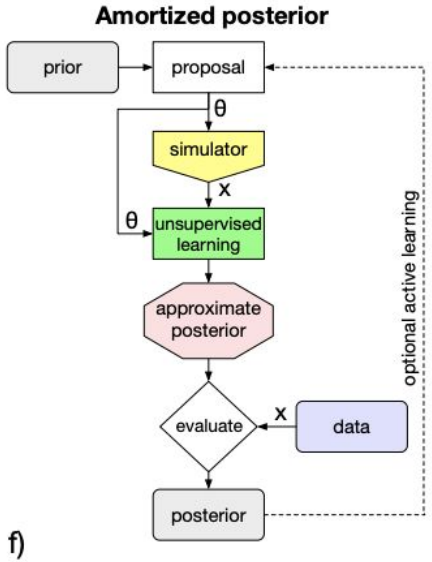
Amortised posterior

Train conditional density estimator for posterior p(θ|x)

Can sample new x

Directly provides posterior for Bayesian inference

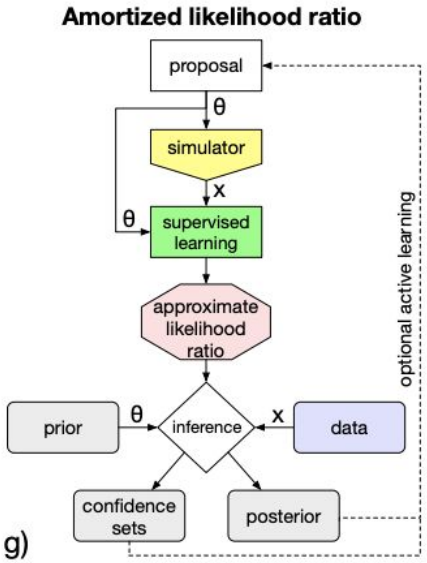
Depends on prior at every step of inference



Amortised likelihood ratio

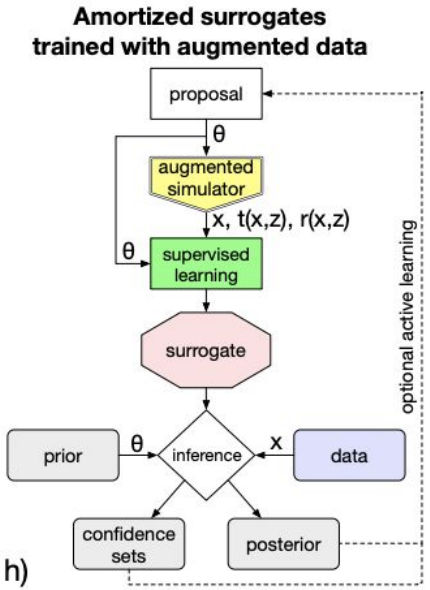
Train function for likelihood ratio
$r(x, z|\theta, \theta')$

Supervised learning problem - usually simpler than density estimation (unsupervised learning)



Amortised surrogates

Additional quantities can be used to augment data to train surrogate models with improved sample efficiency



Postprocessing Checks

Calibration, e.g. using a bootstrap on the parameters

Testing reference parameters

Check known expectation values of likelihood/likelihood ratio/score

Check asymptotic properties

Recommendations

Use active learning

Good low-dimensional summary statistics required for traditional approaches

Use PP if possible - gives better access to additional quantities and makes inference on z easier

Try surrogate models

Learning likelihood ratio can be easier if sampling from surrogate not needed