# Simulation-based Inference

## Learning a Posterior by Inverting Simulators

Talk by Stefan Wezel

mlcolab @ Tübingen University

May 12, 2021

# Overview

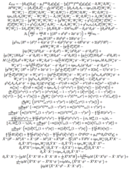
- Problem Setting
- Traditional Approaches and their Issues
- Using Neural Networks to alleviate them
- SBI in the Wild

- Inverting simulators?
- What is a simulator?
  - Forward, generative model with parameters and stochasticity
  - Produces observations
  - Computer program, electrical circuit, …
- Used by scientists to model empirically observed data
- In particle physics, population genetics, epidemiology, …
- Encode knowledge about systems

# What and Why?

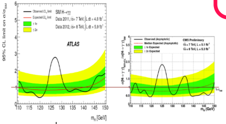
## An Example

Standard model

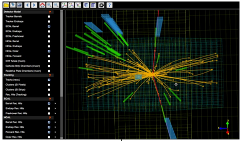


Experiment

Observation




Simulator



What is mass of Higgs boson?

Change mass until observation is reproduced?

SBI!

Figure derived from cern.ch, Achintya Rao and Tom McCauley, sciencealert.com

# What and Why?

## An Example

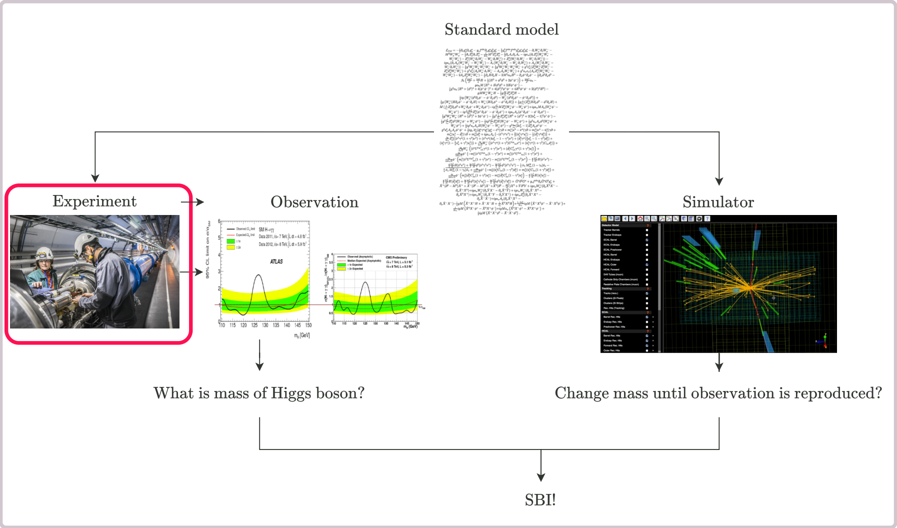



Figure derived from cern.ch, Achintya Rao and Tom McCauley, sciencealert.com

# What and Why?

## An Example

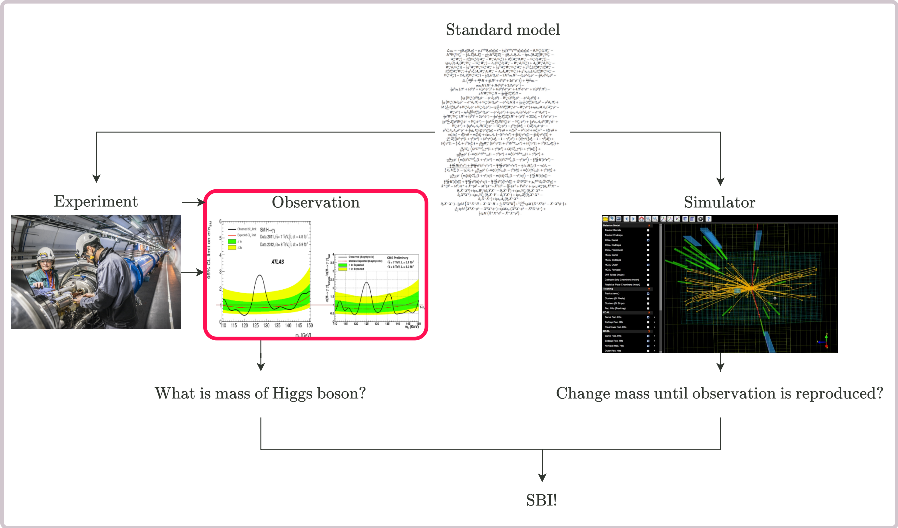
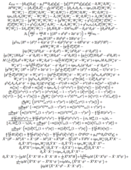
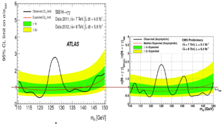



Figure derived from cern.ch, Achintya Rao and Tom McCauley, sciencealert.com

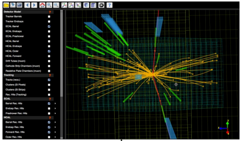
Standard model



Experiment → Observation




Simulator



What is mass of Higgs boson?

Change mass until observation is reproduced?

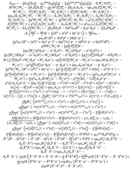
SBI!

Figure derived from cern.ch, Achintya Rao and Tom McCauley, sciencealert.com

# What and Why?

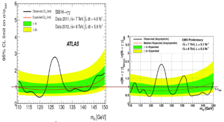
## An Example

Standard model

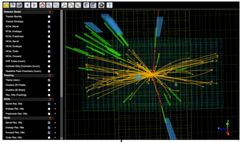


Experiment

Observation




Simulator



What is mass of Higgs boson?

Change mass until observation is reproduced?

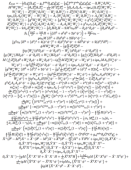
SBI!

Figure derived from cern.ch, Achintya Rao and Tom McCauley, sciencealert.com

# What and Why?

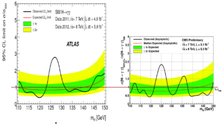
## An Example

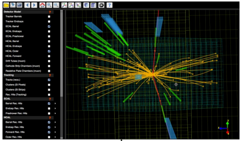
Standard model



Experiment → Observation

Simulator



What is mass of Higgs boson?

Change mass until observation is reproduced?

SBI!

Figure derived from cern.ch, Achintya Rao and Tom McCauley, sciencealert.com

# What and Why?

## An Example

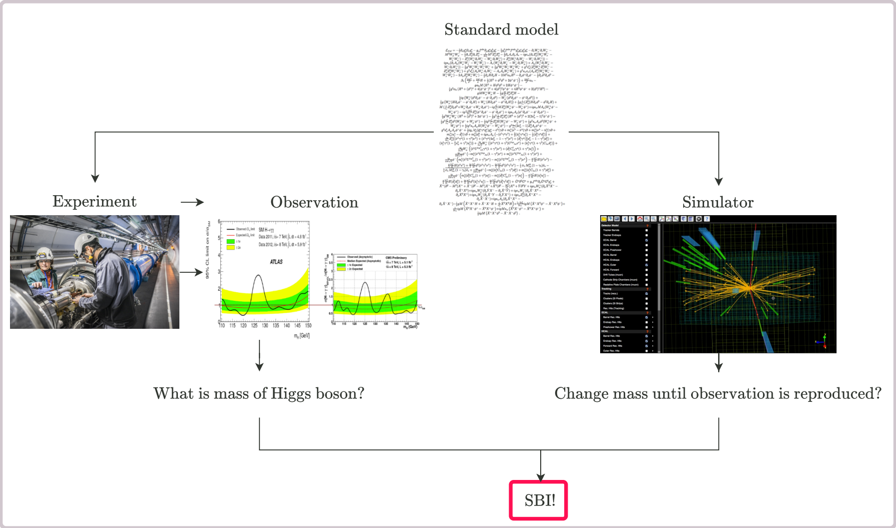



Figure derived from cern.ch, Achintya Rao and Tom McCauley, sciencealert.com

# What and Why?

## Formalizing our Example

- Simulator has parameters $\theta$, nuisance parameter $z$
- Produces data $x$
  - Implicitly defines $p(x, z \mid \theta)$
- Given (real) observation $x_0$, we are interested in parameters $\theta$ that produced it

$$p(\theta \mid x = x_0) = \frac{p(x \mid \theta)p(\theta)}{p(x)} = \frac{\overbrace{\int p(x, z \mid \theta)\, \mathrm{d}z\, p(\theta)}^{\text{intractable}}}{\int p(x \mid \theta)p(\theta)\, \mathrm{d}\theta} \tag{1}$$

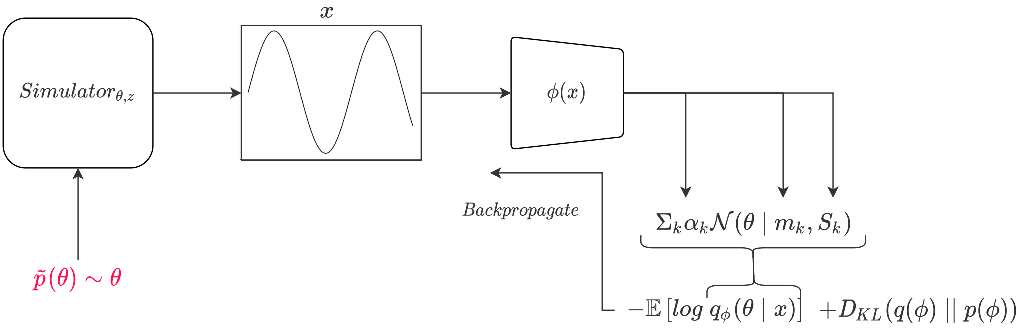
- Typical problem setting of sciences
  - Sophisticated forward model that encapsulates prior knowledge
  - Inference: What are parameters $\theta$ that produce observation $x_0$?
- To find $\theta$, invert simulator

- Broader term: Approximate Bayesian Computation (ABC)
- Rejection ABC
  - Change $\theta$ until something $\epsilon$-close to $x_0$ is produced
  - Inefficient for small $\epsilon$
  - No actual posterior but only point estimates
  - Approximates $p(\theta \mid \|x - x_0\| < \epsilon)$
- Some improvements [6, 1, 2]
  - Reduce parameter space
  - Faster
- Point-estimate to posterior over parameters
- for $\epsilon$-ball around observation

- Papamakarios and Murray [7] learn approximation to actual posterior
- Use simulator to create training set
  - $\{x_j, \theta_j\}_{j \in N}$
- Train Bayesian neural network $\phi$ to predict parameters of posterior over $\theta$s
  - Each weight is defined through mean $m$ and log variance $S$
  - Posterior over network weights
  - $\phi = \mathcal{N}(\phi_m, exp^{\frac{1}{2}\phi_S})$

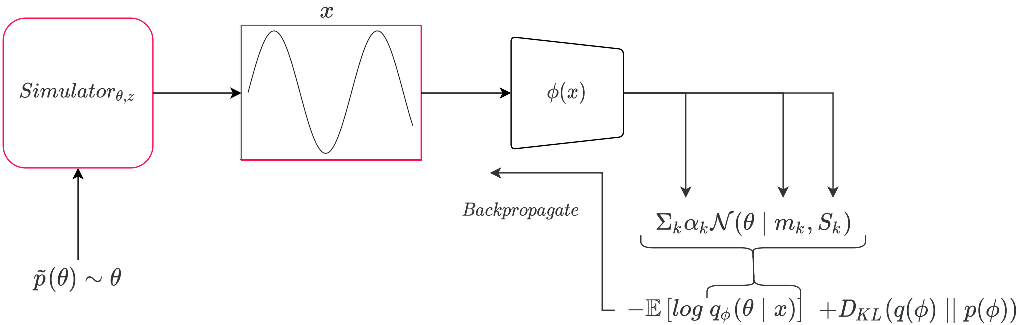
# Learning a Posterior

## from Simulations



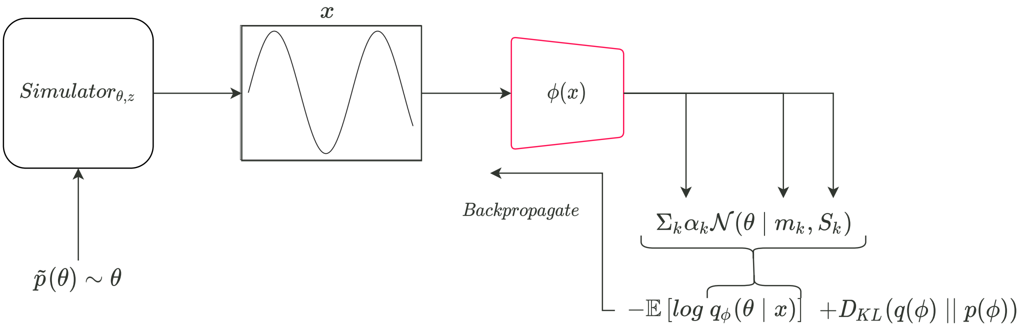
# Learning a Posterior

## from Simulations



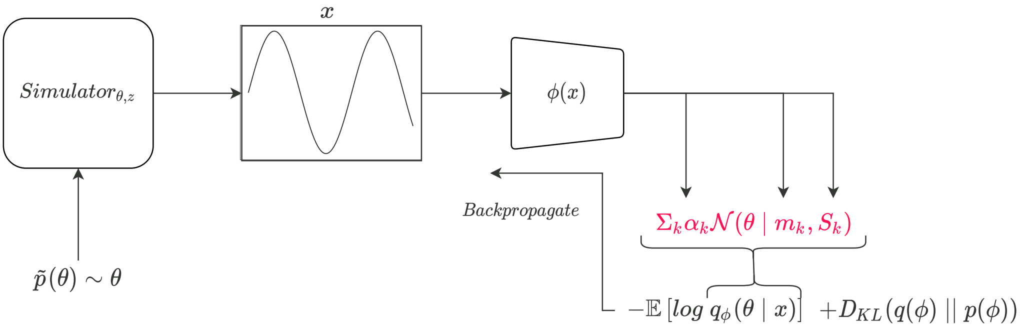
# Learning a Posterior

## from Simulations



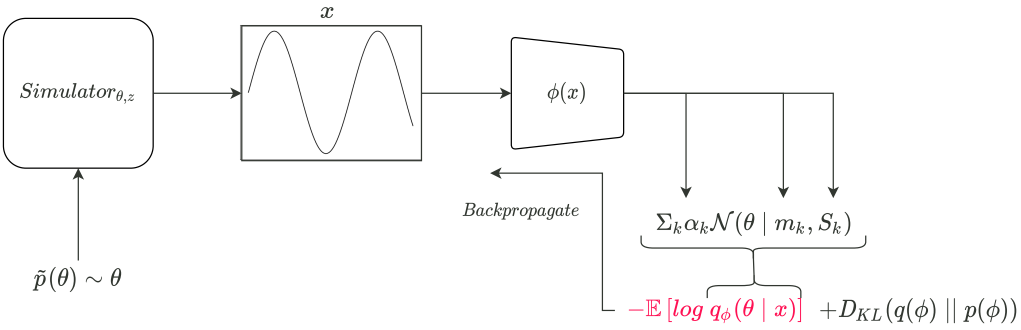
# Learning a Posterior

## from Simulations



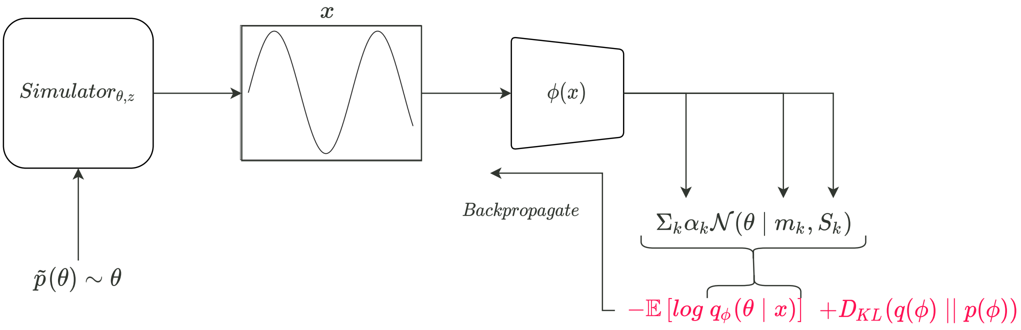
# Learning a Posterior

## from Simulations



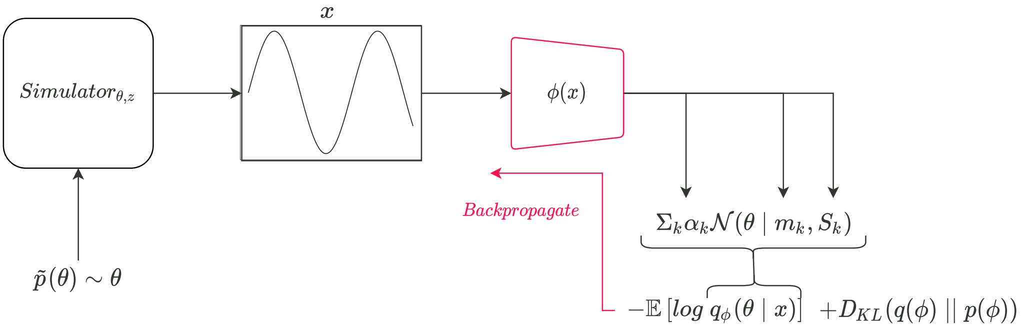
# Learning a Posterior

## from Simulations



# Learning a Posterior

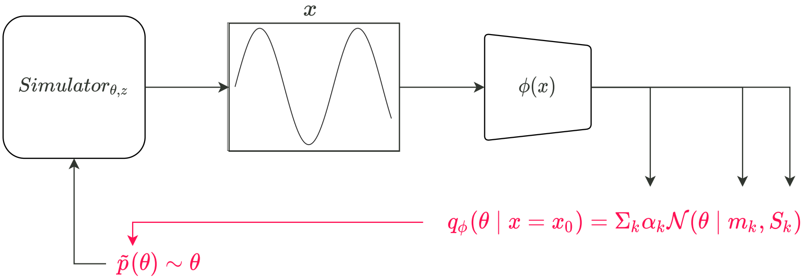
## from Simulations



# Updating the Prior

## With the Posterior

- What if we only care about single observation?
- Iteratively replace prior with posterior
- Faster convergence



# Updating the Prior

## With the Posterior

$\tilde{p}(\theta) \leftarrow p(\theta)$

**repeat**

  **for** $n = 1..N$ **do**

    sample $\theta_n \sim \tilde{p}(\theta)$

    sample $x_n \sim p(x \mid \theta_n)$

  **end**

  train $q_\phi(\theta \mid x)$ on $\{x_n, \theta_n\}$

  $\tilde{p}(\theta) \leftarrow q_\phi(\theta \mid x)$

**until** $\tilde{p}(\theta)$ has converged;

Pseudocode derived from Papamakarios and Murray [7]

# Updating the Prior

## With the Posterior

$\tilde{p}(\theta) \leftarrow p(\theta)$

**repeat**

  **for** $n = 1..N$ **do**

    sample $\theta_n \sim \tilde{p}(\theta)$

    sample $x_n \sim p(x \mid \theta_n)$

  **end**

  train $q_\phi(\theta \mid x)$ on $\{x_n, \theta_n\}$

  $\tilde{p}(\theta) \leftarrow q_\phi(\theta \mid x)$

**until** $\tilde{p}(\theta)$ has converged;

Pseudocode derived from Papamakarios and Murray [7]

# Updating the Prior

## With the Posterior

$\tilde{p}(\theta) \leftarrow p(\theta)$
**repeat**
    **for** $n = 1..N$ **do**
        sample $\theta_n \sim \tilde{p}(\theta)$
        sample $x_n \sim p(x \mid \theta_n)$
    **end**
    train $q_\phi(\theta \mid x)$ on $\{x_n, \theta_n\}$
    $\tilde{p}(\theta) \leftarrow q_\phi(\theta \mid x)$
**until** $\tilde{p}(\theta)$ has converged;

Pseudocode derived from Papamakarios and Murray [7]

# Updating the Prior

## With the Posterior

$\tilde{p}(\theta) \leftarrow p(\theta)$

**repeat**

  **for** $n = 1..N$ **do**

    sample $\theta_n \sim \tilde{p}(\theta)$

    sample $x_n \sim p(x \mid \theta_n)$

  **end**

  train $q_\phi(\theta \mid x)$ on $\{x_n, \theta_n\}$

  $\tilde{p}(\theta) \leftarrow q_\phi(\theta \mid x)$

**until** $\tilde{p}(\theta)$ has converged;

Pseudocode derived from Papamakarios and Murray [7]

- Lueckmann et al. [5] apply SBI to neuroscience setting
  - Nonsensical observations
  - Summary statistics challenging
- They propose
  - Train classifier to predict whether parameters fail
  - Extract features with RNN
  - Posterior over network weights
  - Use distribution over weights from previous rounds as prior in regularization term
    - $\mathcal{L}_t := -\frac{1}{N}\sum_n \mathbb{E}\left[log\ q_\phi(\theta \mid x)\right] + \frac{1}{N}\mathrm{D}_{KL}(q^t(\phi) \mid\mid q^{t-1}(\phi))$
    - Implicitly store data from previous rounds

- Apply to Hodgkin-Huxley model [4]
  - Dynamics of neuron’s membrane
  - Parameters (concentration of sodium, potassium, ...)
  - Produces spikes
  - Numerical simulators available (NEURON software [3])

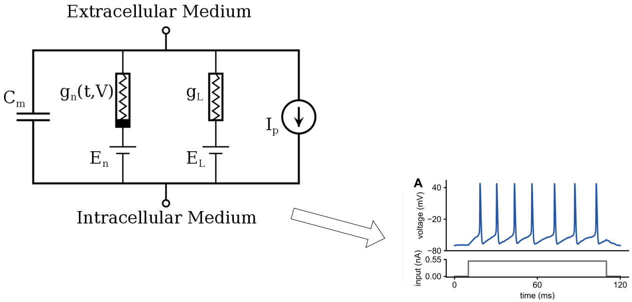



Figure derived from Lueckmann et al. [5] and Krishnavedala

# Applying SBI

## Results

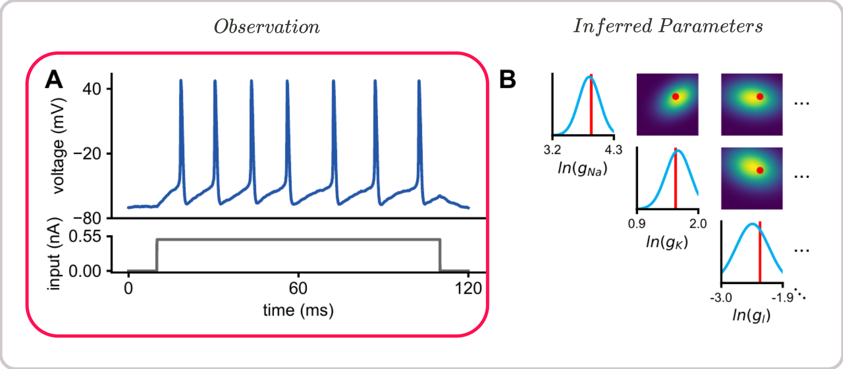



Figure derived from Lueckmann et al. [5]

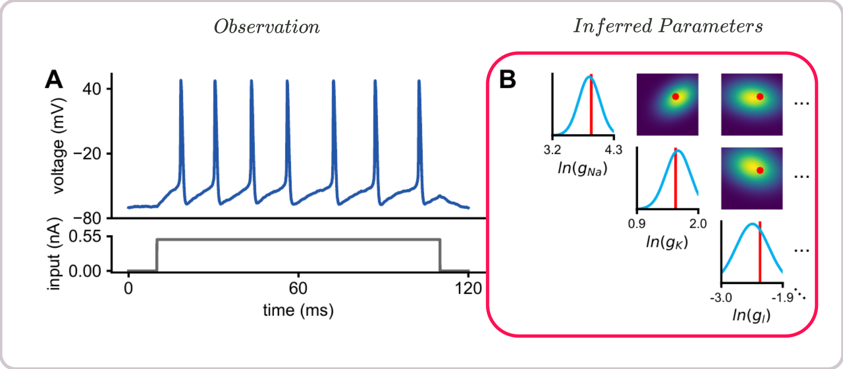



Figure derived from Lueckmann et al. [5]

# How to use SBI?

## Some advertisement

- mlcolab and mackelab is developing sbi python library [8]
- If you are interested: try it out and give us your feedback
- `colab.research.google.com/drive/1L1T45hruoScyu3hN4WkDW6f0De4UemjI?usp=sharing`
- `github.com/mackelab/sbi`
- `pip install sbi` (ignore error message)

# Questions for you

## Discussion

- Do you know a setting where SBI could be applied?
- What are downsides of SBI?
- Can’t we just use a variational autoencoder?

# References


[1] M. A. Beaumont, J.-M. Cornuet, J.-M. Marin, and C. P. Robert. Adaptive approximate bayesian computation. Biometrika, 96(4):983–990, 2009.

[2] F. V. Bonassi, M. West, et al. Sequential monte carlo with adaptive weights for approximate bayesian computation. Bayesian Analysis, 10(1):171–187, 2015.

[3] N. T. Carnevale and M. L. Hines. The NEURON book. Cambridge University Press, 2006.

[4] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of physiology, 117(4):500–544, 1952.

[5] J.-M. Lueckmann, P. J. Goncalves, G. Bassetto, K. Öcal, M. Nonnenmacher, and J. H. Macke. Flexible statistical inference for mechanistic models of neural dynamics. arXiv preprint arXiv:1711.01861, 2017.

[6] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain monte carlo without likelihoods. Proceedings of the National Academy of Sciences, 100(26):15324–15328, 2003.

[7] G. Papamakarios and I. Murray. Fast epsilon-free inference of simulation models with bayesian conditional density estimation. arXiv preprint arXiv:1605.06376, 2016.

[8] A. Tejero-Cantero, J. Boelts, M. Deistler, J.-M. Lueckmann, C. Durkan, P. J. Gonçalves, D. S. Greenberg, and J. H. Macke. Sbi: a toolkit for simulation-based inference. Journal of Open Source Software, 5(52):2505, 2020.

- Example (if there is time)