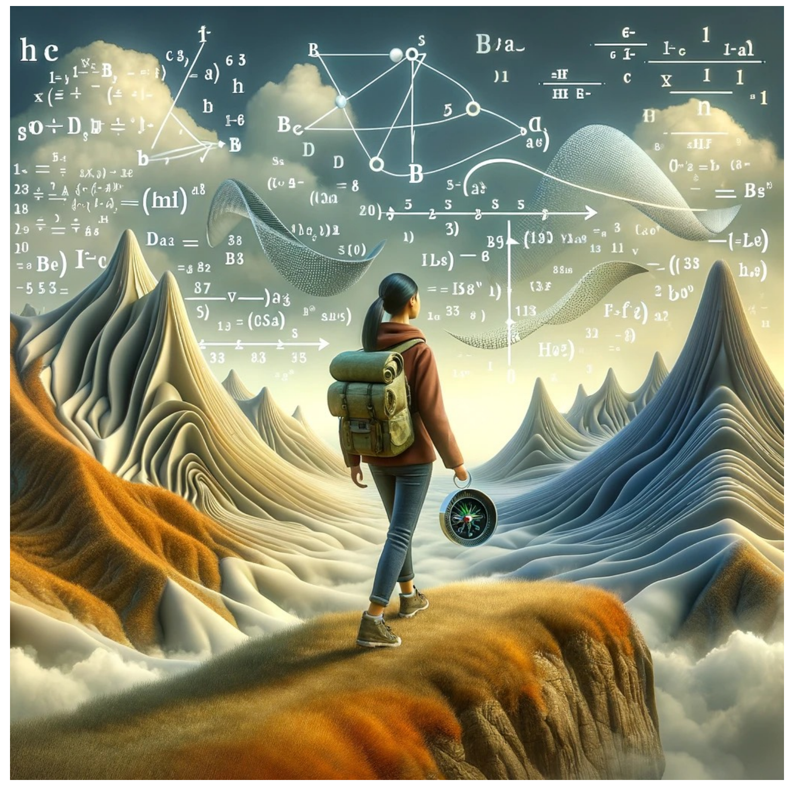
Simulation-based inference



Lecture 1: Introduction

April 2024

Pedro Gonçalves, Anastasia Krouglova
goncalveslab.sites.vib.be/en

Guy Moss
mackelab.org

Acknowledgments

Jakob Macke

Cornelius Schröder

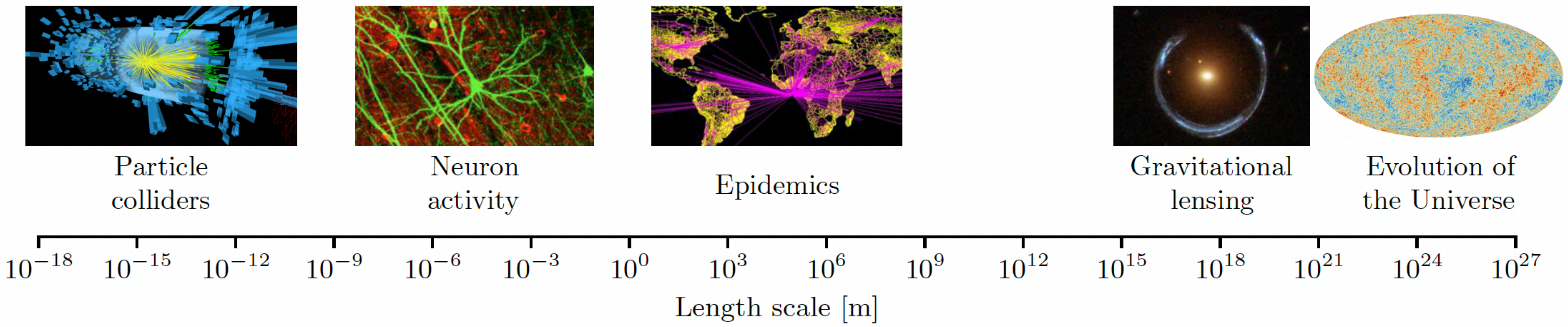
Michael Deistler

Manuel Glöckler

- Parts of the lectures are adapted from AIMS SBI January 2024, JH Macke, C Schröder, PJ Goncalves
- Some slides by Álvaro Tejero-Cantero

1.1 Simulators

Models defined as simulators are everywhere



- Astrophysics: simulate formation of galaxies, stars, or planets
- Neuroscience: simulate neural network dynamics
- Epidemiology: simulate spread of an infectious disease
- Different communities use different names for simulator-based models: generative models, implicit models, stochastic simulation models, probabilistic programs…

But what is a simulator? Why do we need it?

- Making models/simulators is part of the scientific method:
    1. models reproduce (only) some aspects of reality, they are similar to something that exists;
    2. when mathematically formalized, they enable quantitative, testable hypotheses.

- Model functionalities:
    1. prediction — to support decisions;
    2. understanding — to select interventions.

- The structure that does not change is the model:
    1. the malleable part are parameters;
    2. parameters are 'tuned' based on observations.

Simulators come in all sorts of flavors

- Simulator as numerical solver for an explicit model (e.g., Ordinary differential Equations) — based on discretization.

- Simulator as defined by code, i.e., an implicit model built from individual interaction rules (e.g., Cellular Automata).

- Anything in-between.

- Continuous vs. discrete, dynamic vs. steady-state, deterministic vs. stochastic...

We are particularly interested in stochastic simulators

- Stochastic modelling is more general than deterministic modelling, although typically harder to simulate and analyse.

- Often, stochasticity is used for modelling processes that lack mechanistic hypotheses.
- The process itself can be stochastic, e.g., decay in a nucleus.

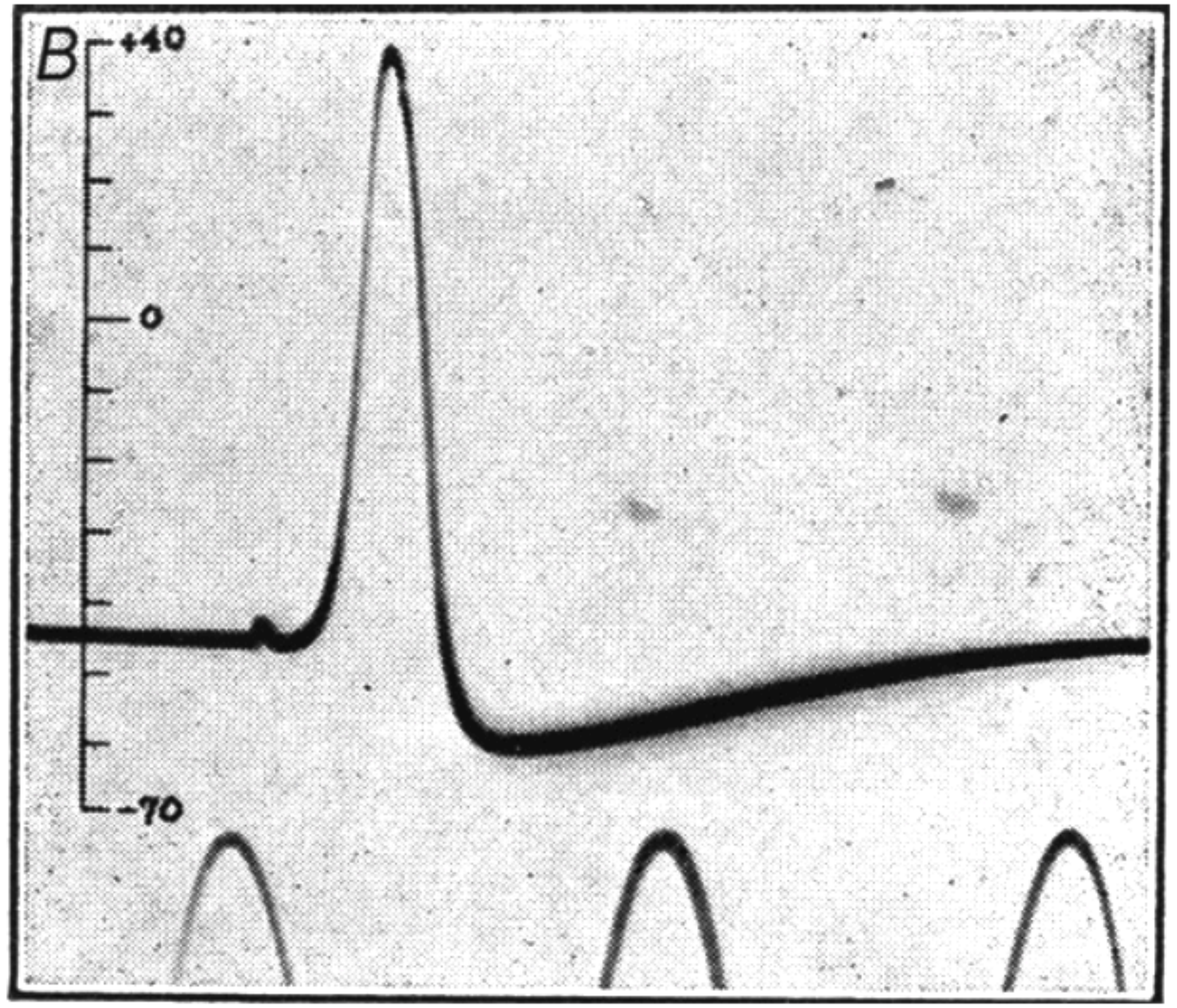
- Some sources of stochasticity: unobserved latent variables, instrument noise, numerical approximations.

- Outputs must be stochastic themselves - random variables: probabilities!

1.2 Three example simulators

1.2.1 A famous neuroscience model: The Hodgkin-Huxley equations

How do neurons generate action potentials?

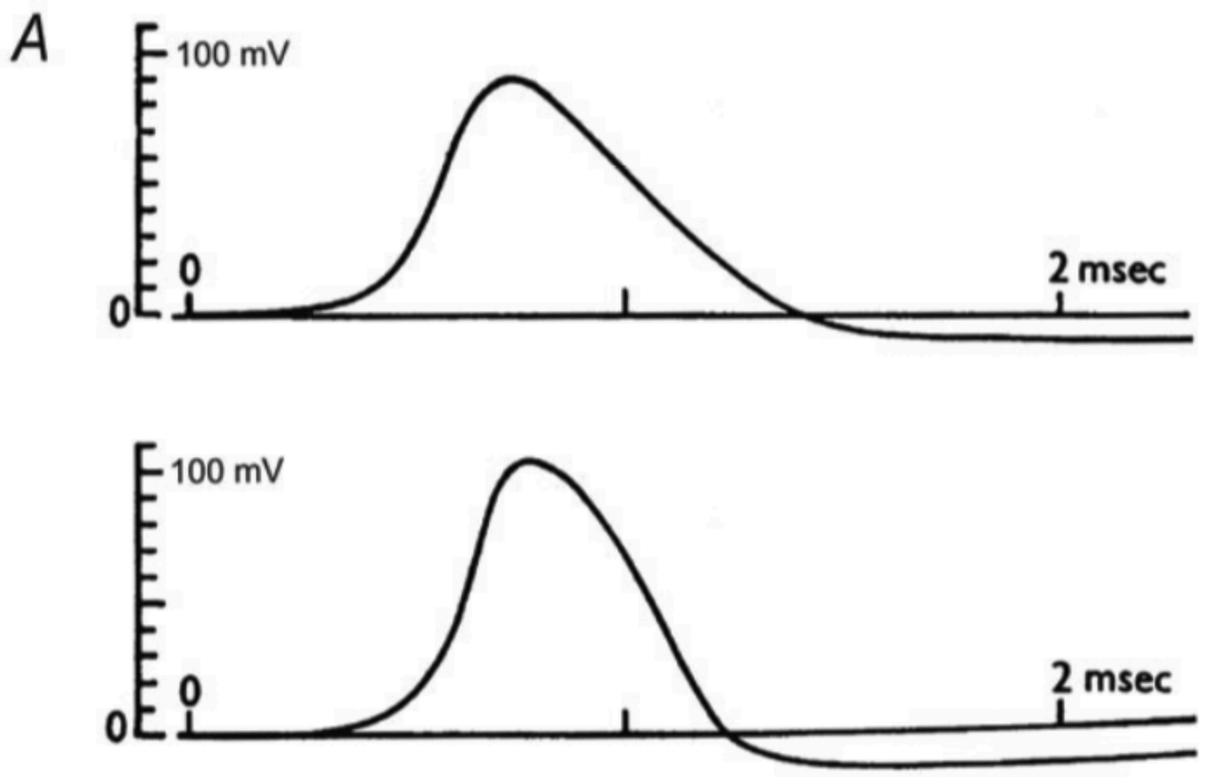


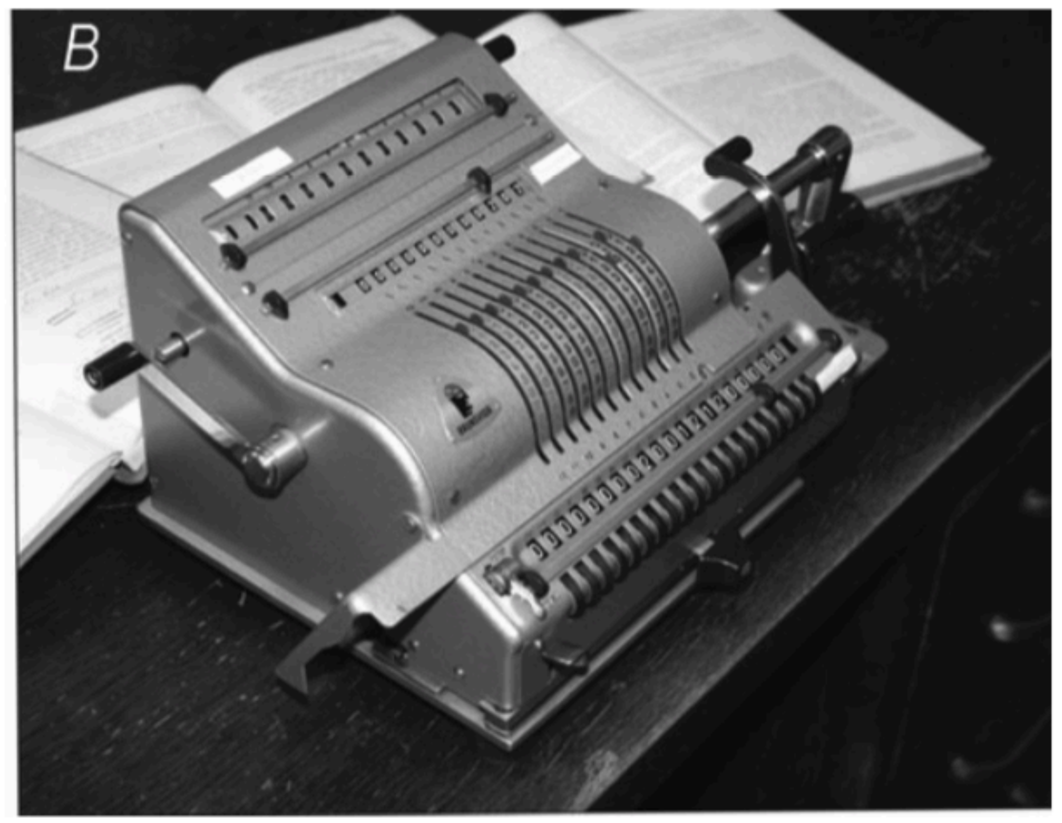
Nobel Prize 1963, images Schwiening 2010

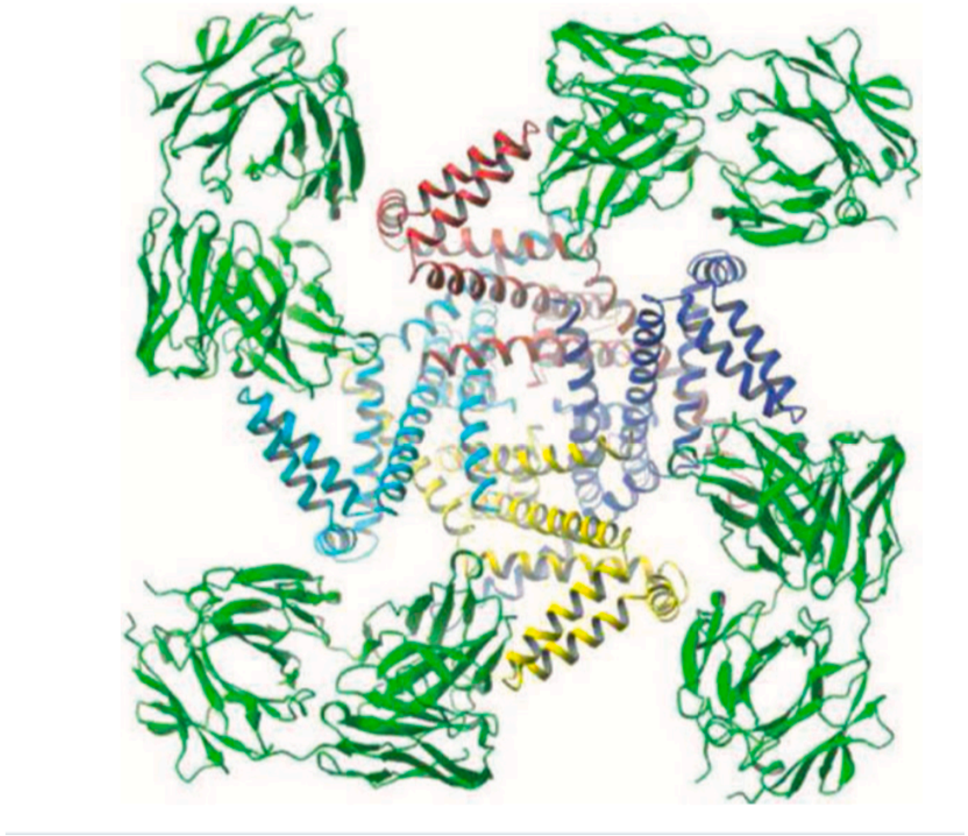
1.2.1 A famous neuroscience model: The Hodgkin-Huxley equations

Differential equations relating voltage to underlying ion-channel kinetics

$$C_m \frac{dV}{dt} = g_{\mathrm{leak}}(E_{\mathrm{leak}} - V) + \bar{g}_{\mathrm{Na}} m^3 h (E_{\mathrm{Na}} - V) + \bar{g}_{\mathrm{K}} \boxed{n^4} (E_{\mathrm{K}} - V) + \bar{g}_M p (E_{\mathrm{K}} - V) + I_{\mathrm{inj}}$$







- Equations (derived from squid) are applicable to (e.g.) humans
- Hypotheses about ion-channels

  MacKinnon Nobel Prize Chemistry 2003

1.2.2 Lotka-Volterra prey-predator model

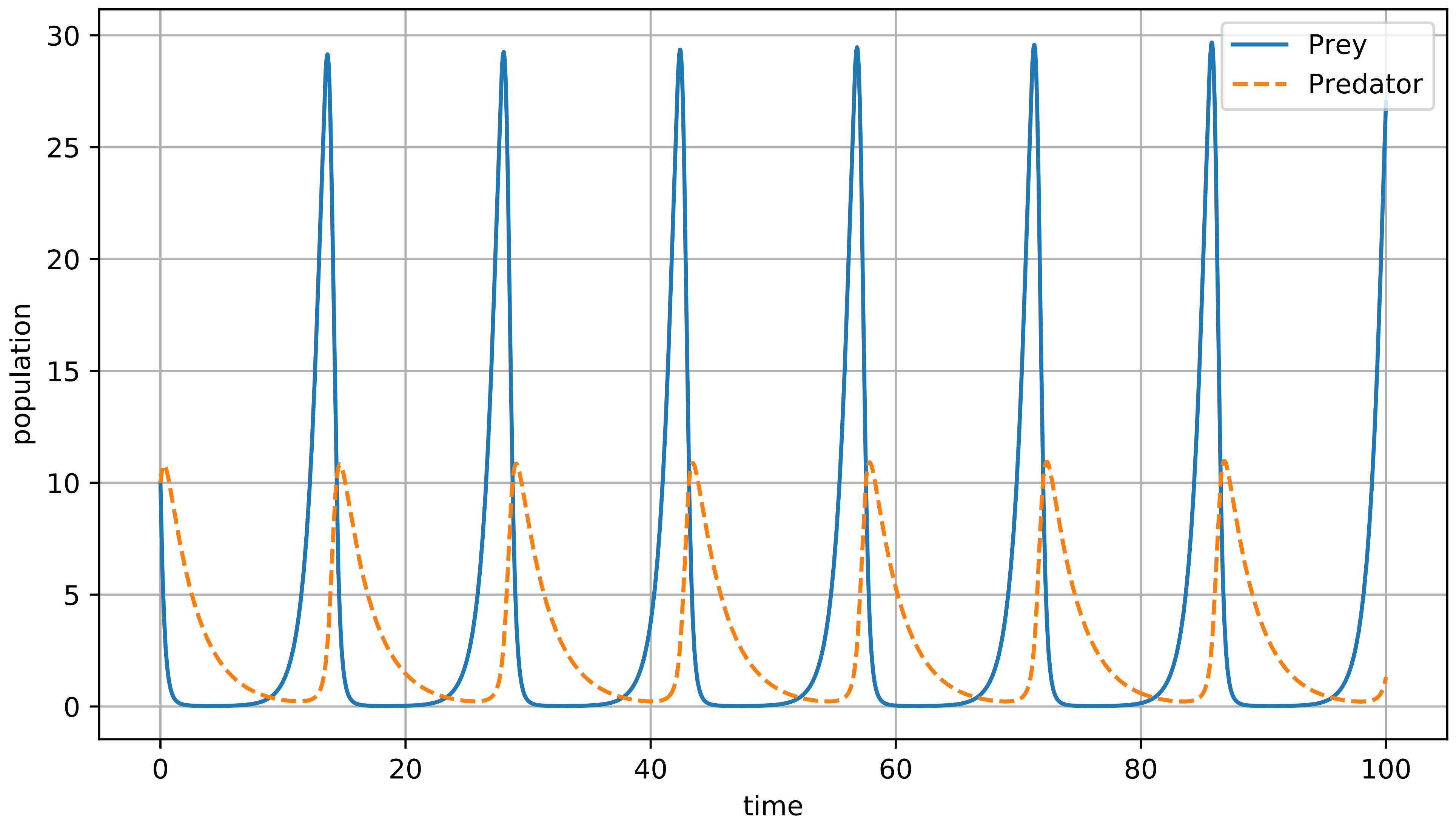
- Model of interaction between a prey and a predator (e.g., rabbit and fox):

$$\frac{dx}{dt} = \alpha x - \beta xy$$

$$\frac{dy}{dt} = \delta xy - \gamma y$$

x and y are the population densities of prey and predator, respectively.

- Applications in ecology, economics…



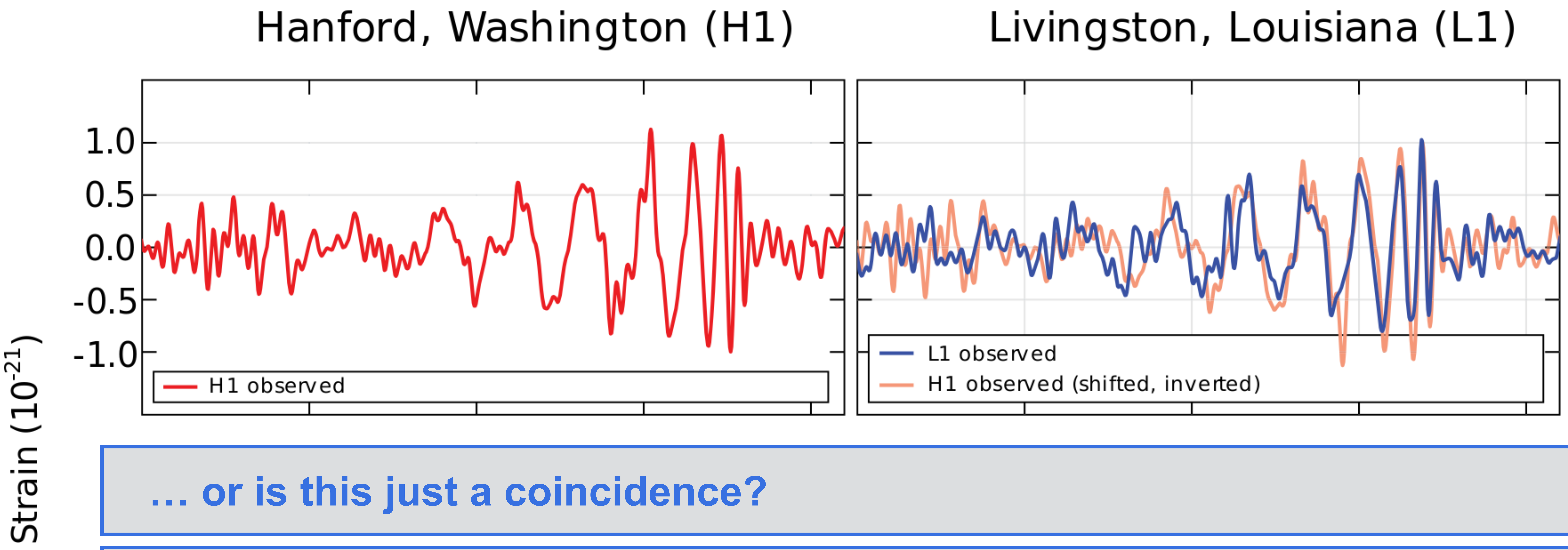
en.wikipedia.org/wiki/Lotka–Volterra_equations

1.2.3 Collision of two black holes

Image Credit: SXS project

LIGO Hanford

1.3.3 Did LIGO detect a merger of two black holes?

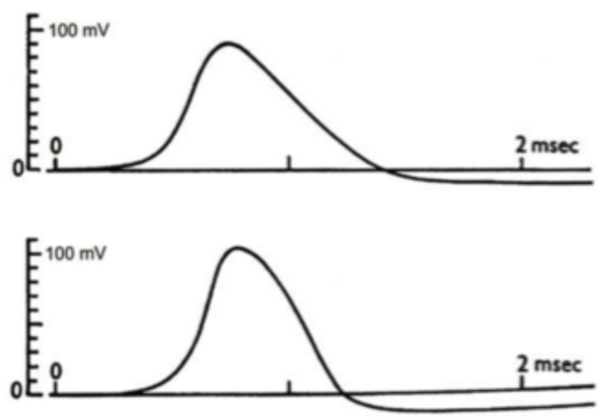


… or is this just a coincidence?

Lesson: (Prior) scientific knowledge can be extremely useful for data-interpretation

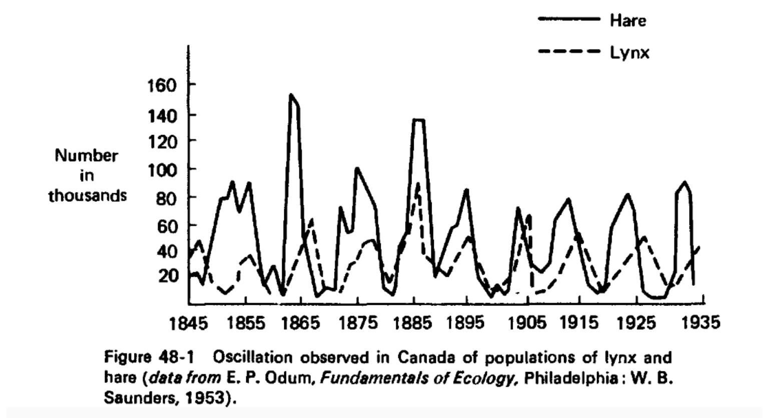
Mechanistic models and simulations play a central role across natural sciences

Neurophysiology



fig; Schwiening 2003

Population Dynamics



fig; Saunders 1953

Psychophysics of Decision Making

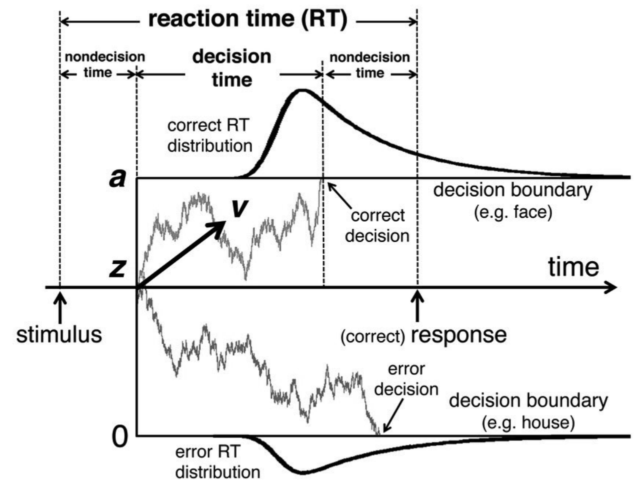


fig: Murata 2014

Particle Physics

$$
\begin{aligned}
\mathcal{L}_{SM} = \; & \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^a_{\mu\nu}G^{\mu\nu}_a}_{\text{kinetic energies and self-interactions of the gauge bosons}} \\
& + \underbrace{\bar{L}\gamma^\mu\left(i\partial_\mu - \frac{1}{2}g\tau\cdot\mathbf{W}_\mu - \frac{1}{2}g'YB_\mu\right)L + \bar{R}\gamma^\mu\left(i\partial_\mu - \frac{1}{2}g'YB_\mu\right)R}_{\text{kinetic energies and electroweak interactions of fermions}} \\
& + \underbrace{\frac{1}{2}\left|\left(i\partial_\mu - \frac{1}{2}g\tau\cdot\mathbf{W}_\mu - \frac{1}{2}g'YB_\mu\right)\phi\right|^2 - V(\phi)}_{W^\pm, Z, \gamma \text{ and Higgs masses and couplings}} \\
& + \underbrace{g''(\bar{q}\gamma^\mu T_a q)G^a_\mu}_{\text{interactions between quarks and gluons}} + \underbrace{(G_1\bar{L}\phi R + G_2\bar{L}\phi_c R + h.c.)}_{\text{fermion masses and couplings to Higgs}}
\end{aligned}
$$

fig: Cranmer 2017

Systems Biology

$$\dot{S} = \alpha - \gamma SI - dS, \qquad (3.10a)$$

$$\dot{I} = \gamma SI - vI - dI, \qquad (3.10b)$$

$$\dot{R} = vI - dR, \qquad (3.10c)$$

fig; Toni 20018

Cosmology

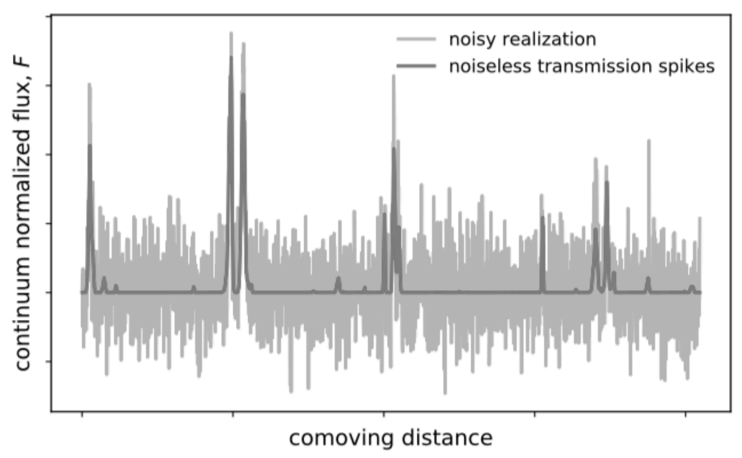


fig: Walting 2018

Molecular Dynamics

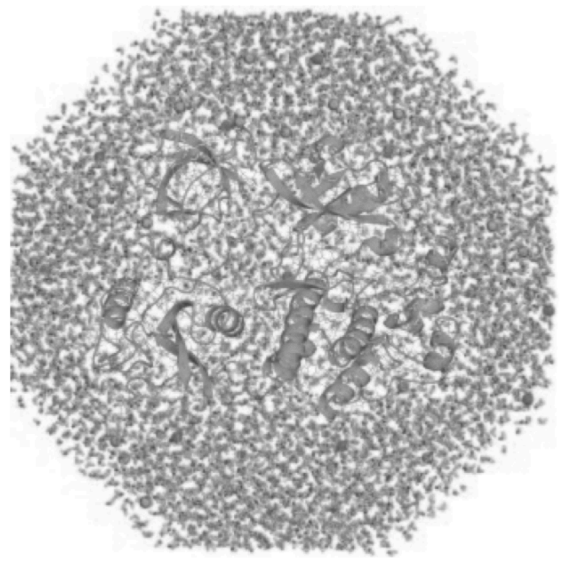
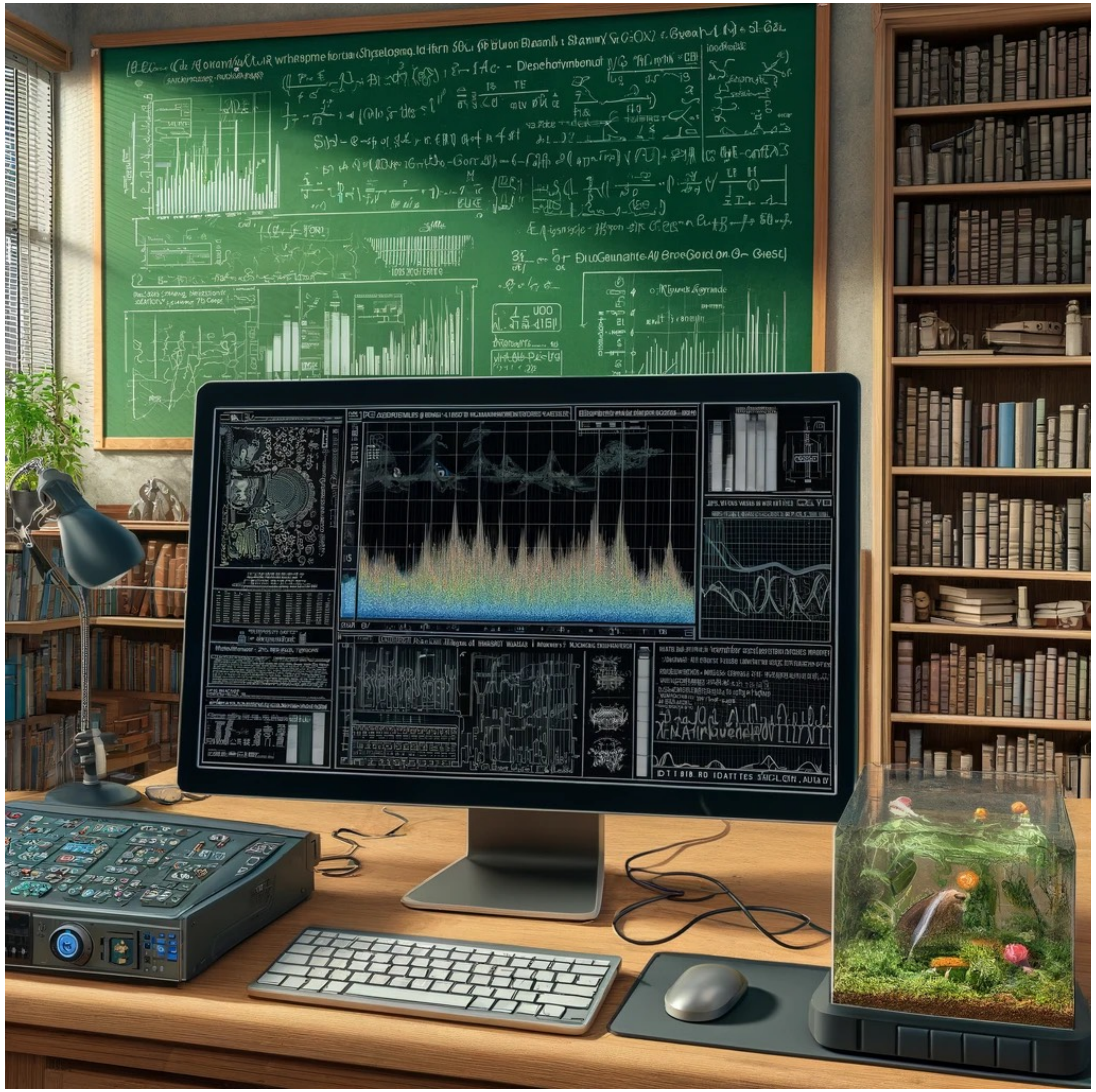


fig: Karplus 2005

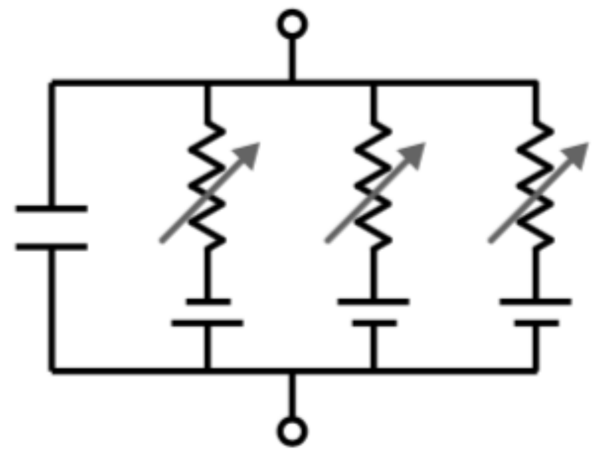
...

1.3 Science and the role of simulators



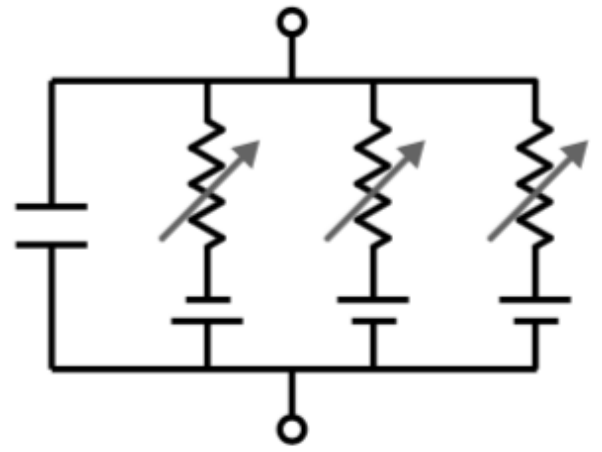
Mechanistic model

$$C_m \frac{dV}{dt} = \sum_c \bar{g}_c g_c [E_c - V] + I$$

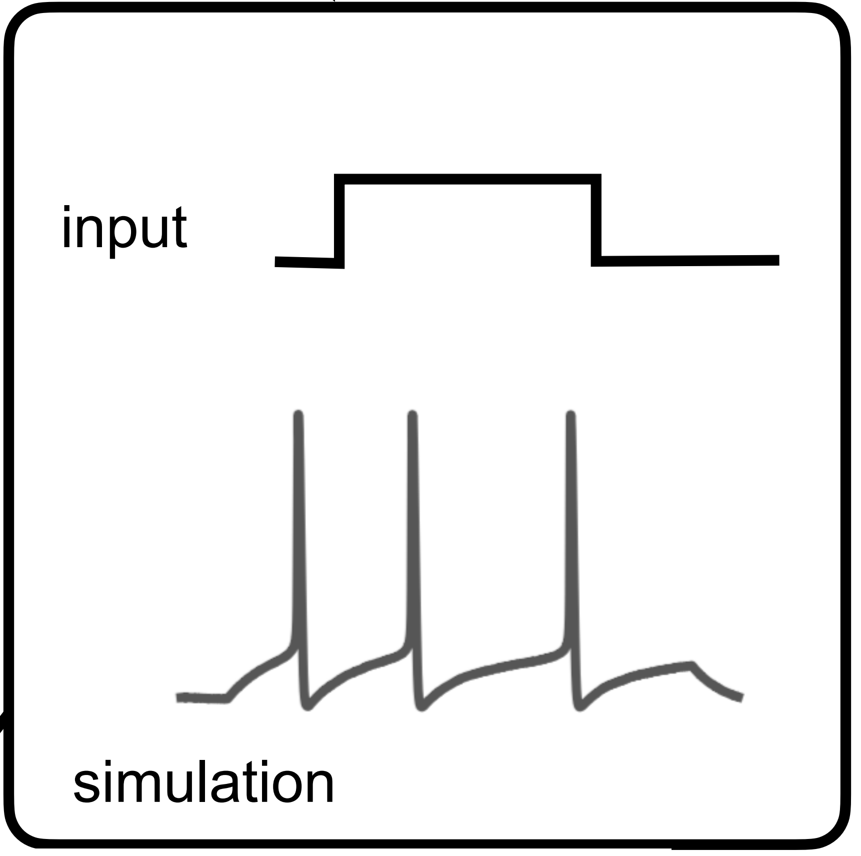


Mechanistic model

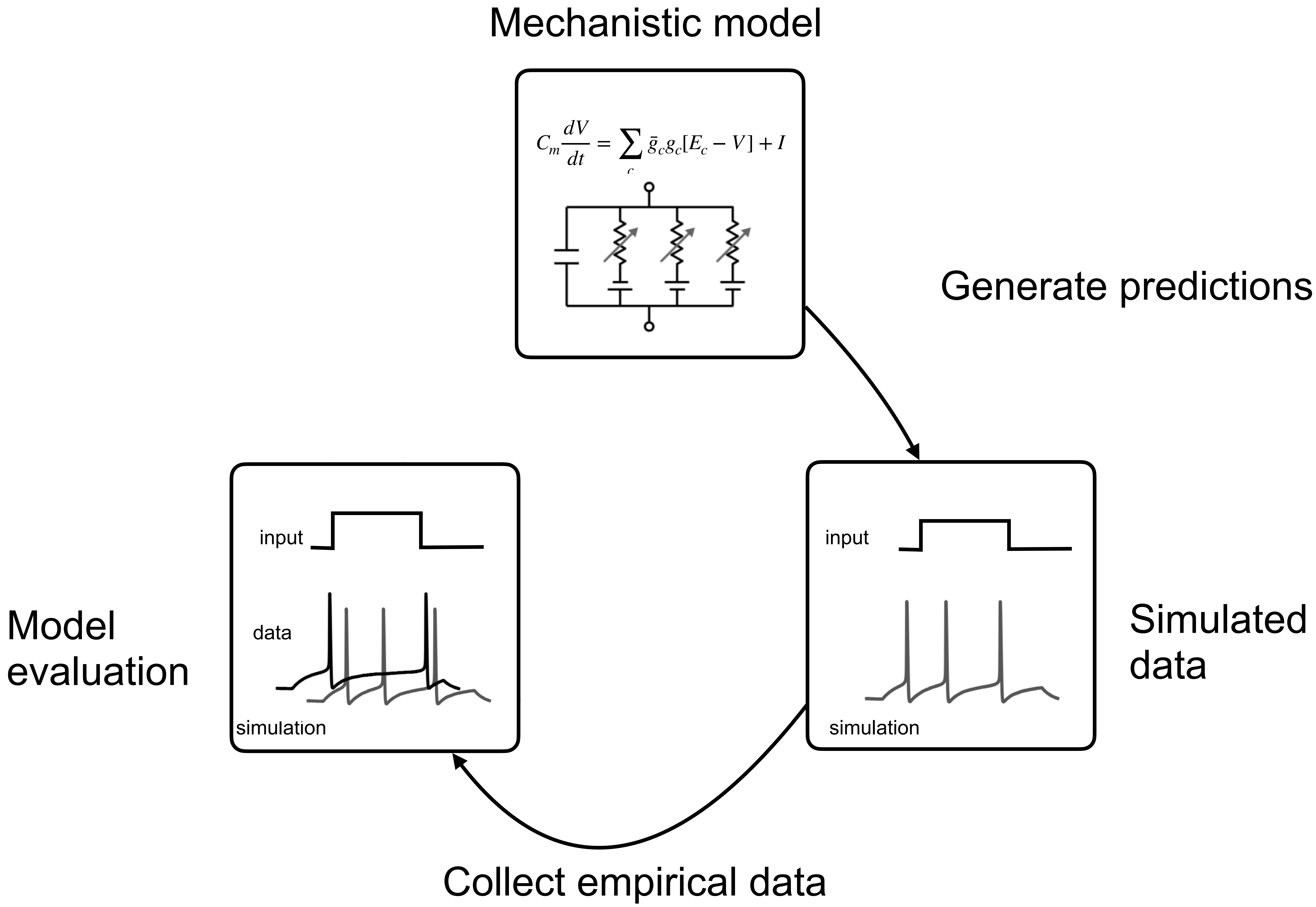
$$C_m \frac{dV}{dt} = \sum_c \bar{g}_c g_c [E_c - V] + I$$



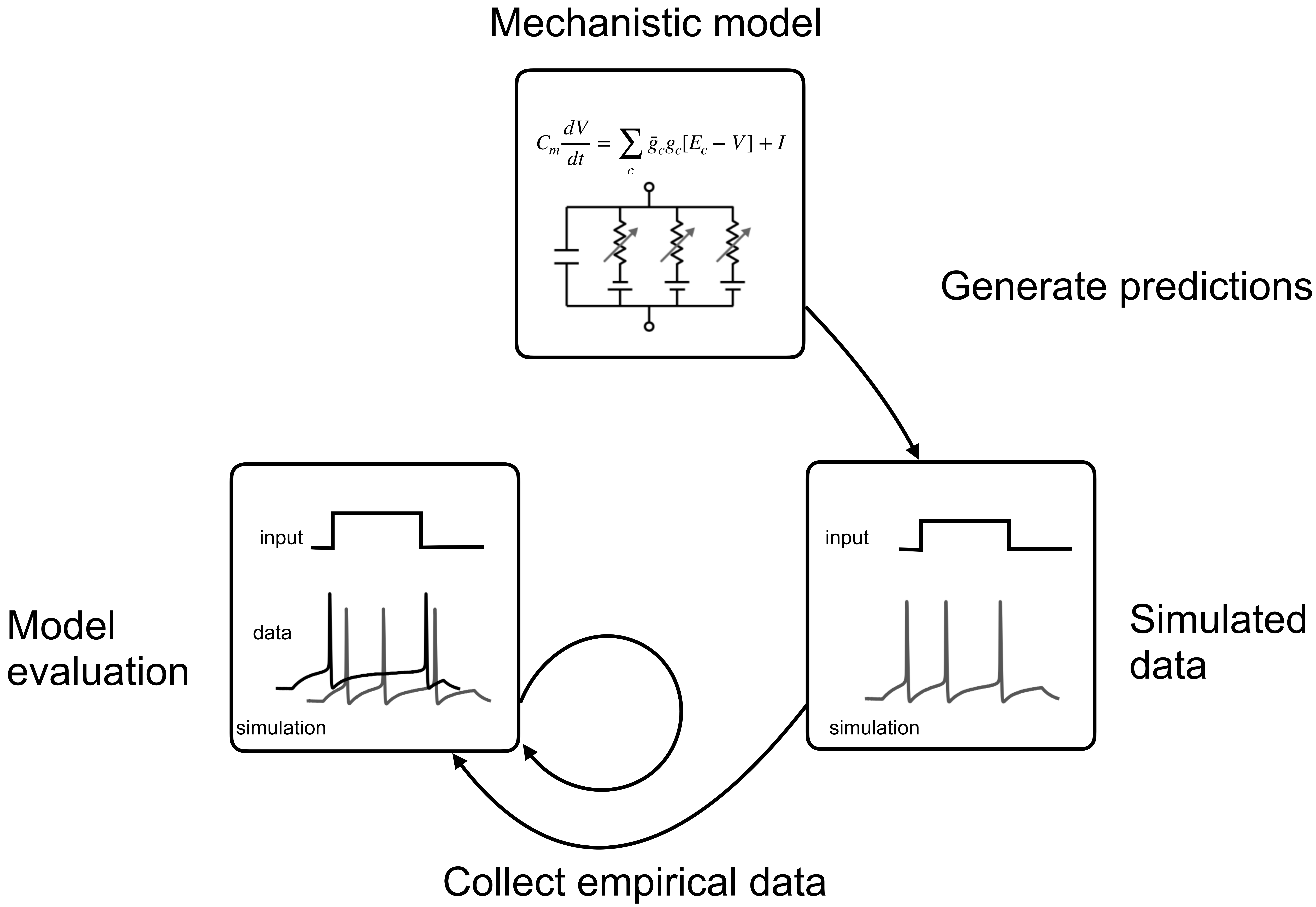
Generate predictions

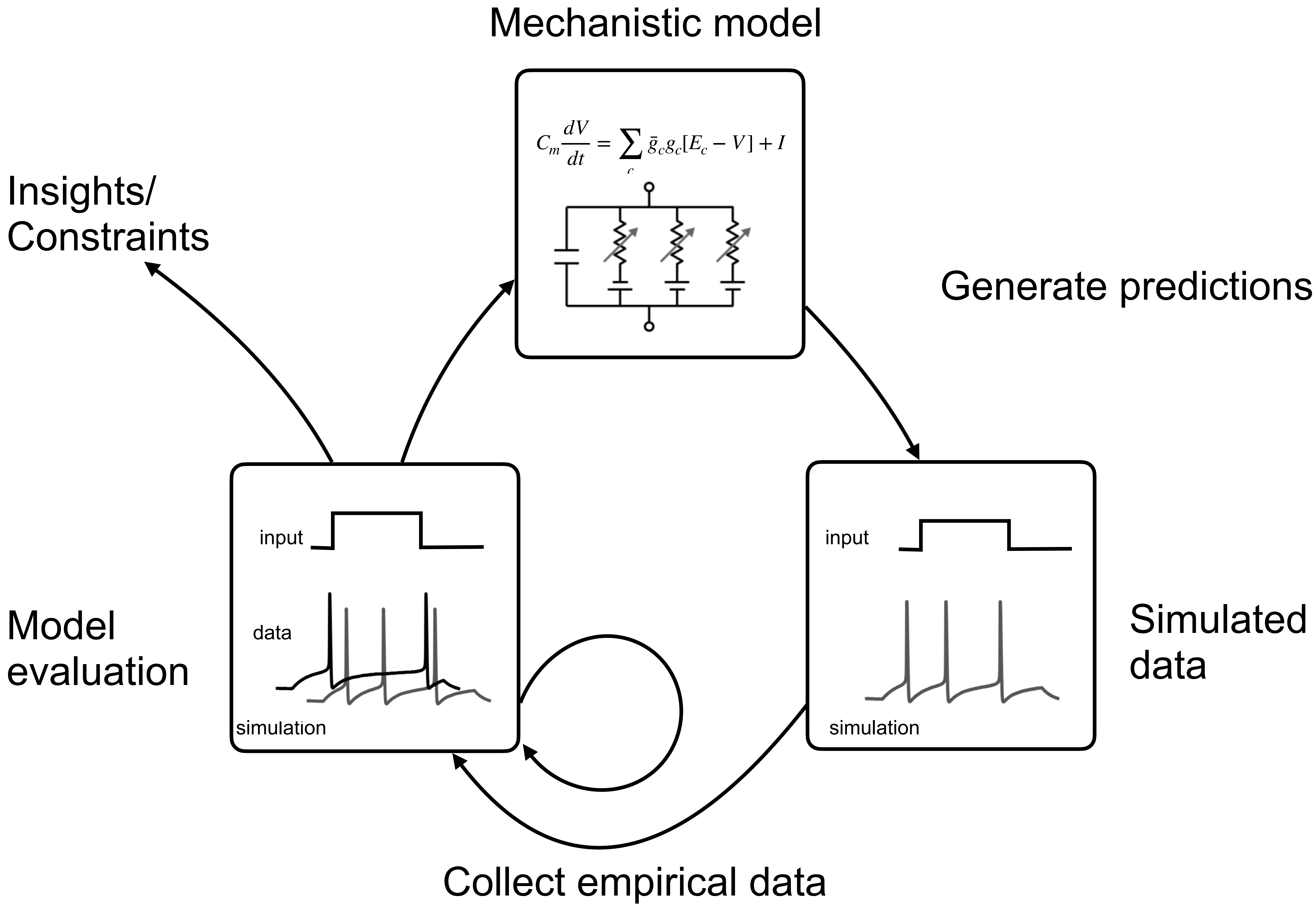


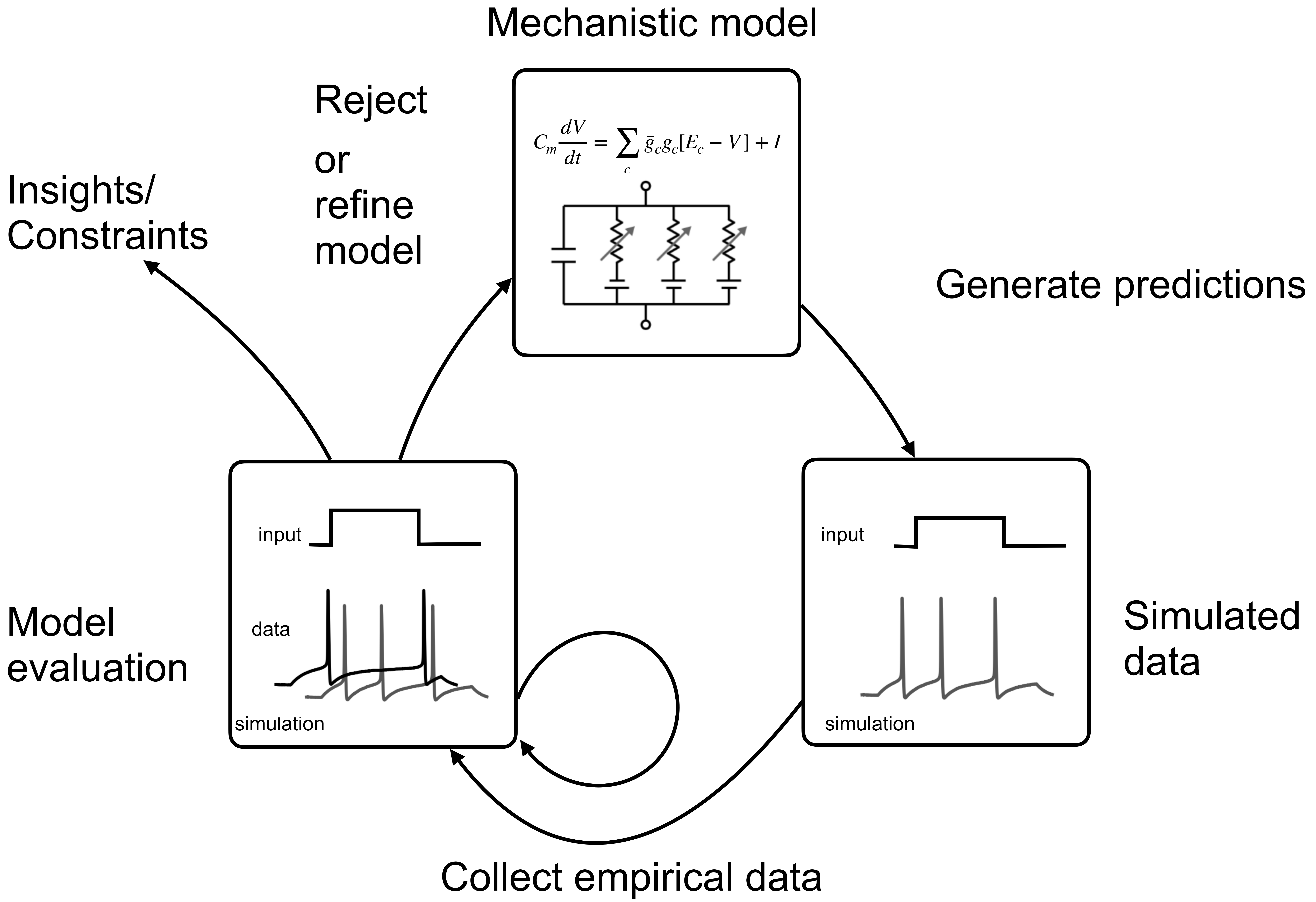
Simulated data



Mechanistic model
$C_m \frac{dV}{dt} = \sum_c \bar{g}_c g_c [E_c - V] + I$
Generate predictions
input
simulation
Simulated data
Collect empirical data
input
data
simulation
Model evaluation

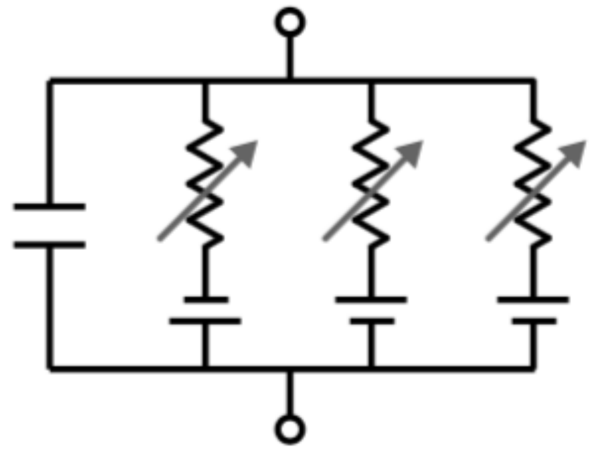

Mechanistic model
$C_m \frac{dV}{dt} = \sum_c \bar{g}_c g_c [E_c - V] + I$
Generate predictions
input
simulation
Simulated data
Collect empirical data
input
data
simulation
Model evaluation


Mechanistic model
$C_m \frac{dV}{dt} = \sum_c \bar{g}_c g_c [E_c - V] + I$
Insights/
Constraints
Generate predictions
input
data
simulation
Model
evaluation
input
simulation
Simulated
data
Collect empirical data


Mechanistic model
$C_m \frac{dV}{dt} = \sum_c \bar{g}_c g_c [E_c - V] + I$
Reject
or
refine
model
Insights/
Constraints
Generate predictions
input
data
simulation
Model
evaluation
input
simulation
Simulated
data
Collect empirical data

Mechanistic model

$$C_m \frac{dV}{dt} = \sum_c \bar{g}_c g_c [E_c - V] + I$$



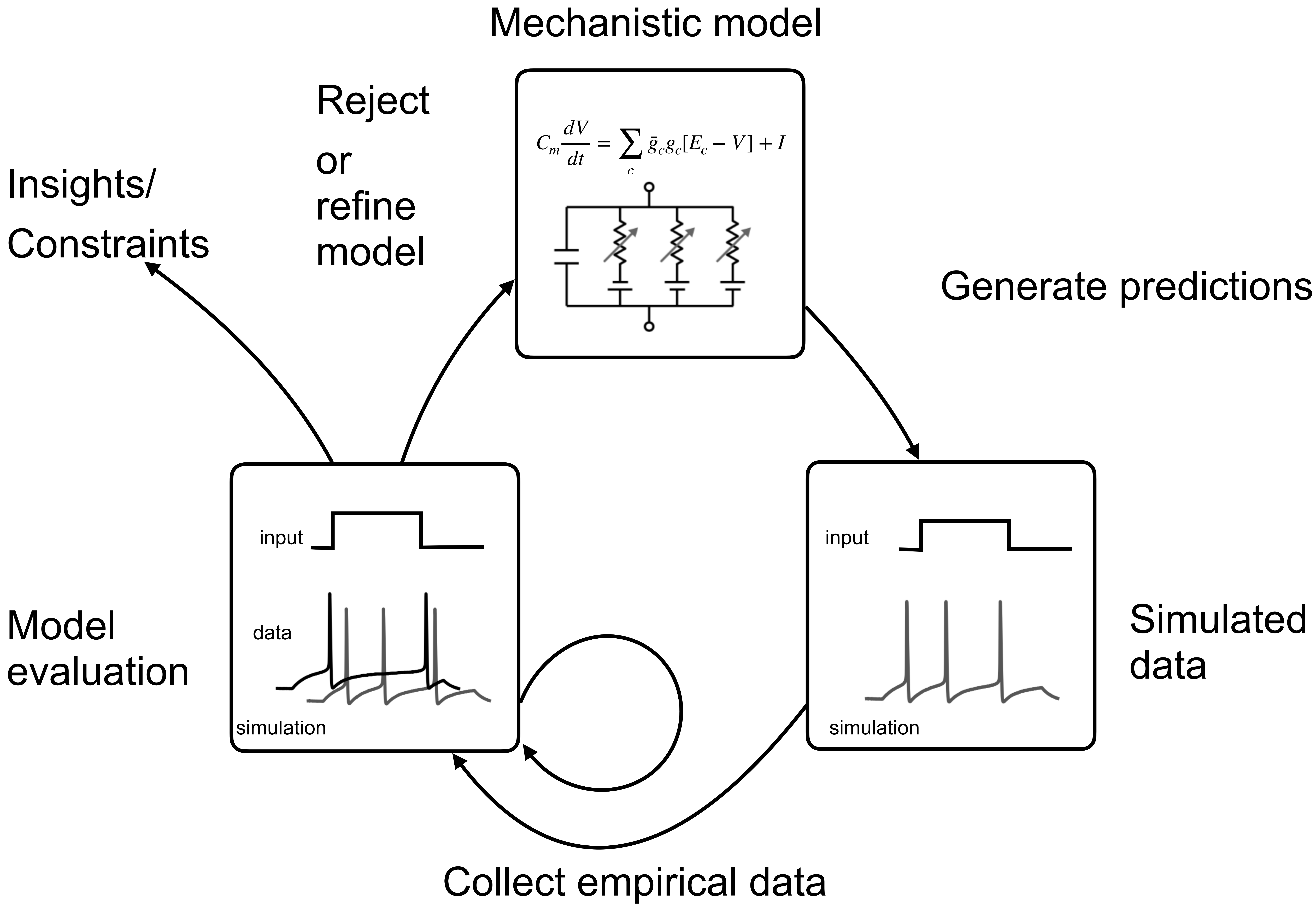
Mechanistic Models

- Goal: Understanding
- built from assumptions about mechanisms
- knowledge of (e.g.) dynamics
- interpretable parameters
- often hard to fit to data

Machine Learning

- Goal: Performance
- built with computation and generalization in mind
- data + inductive bias
- often no direct interpretation
- designed to fit data

Goal: Combine strengths of both approaches to build tools for data-driven science.

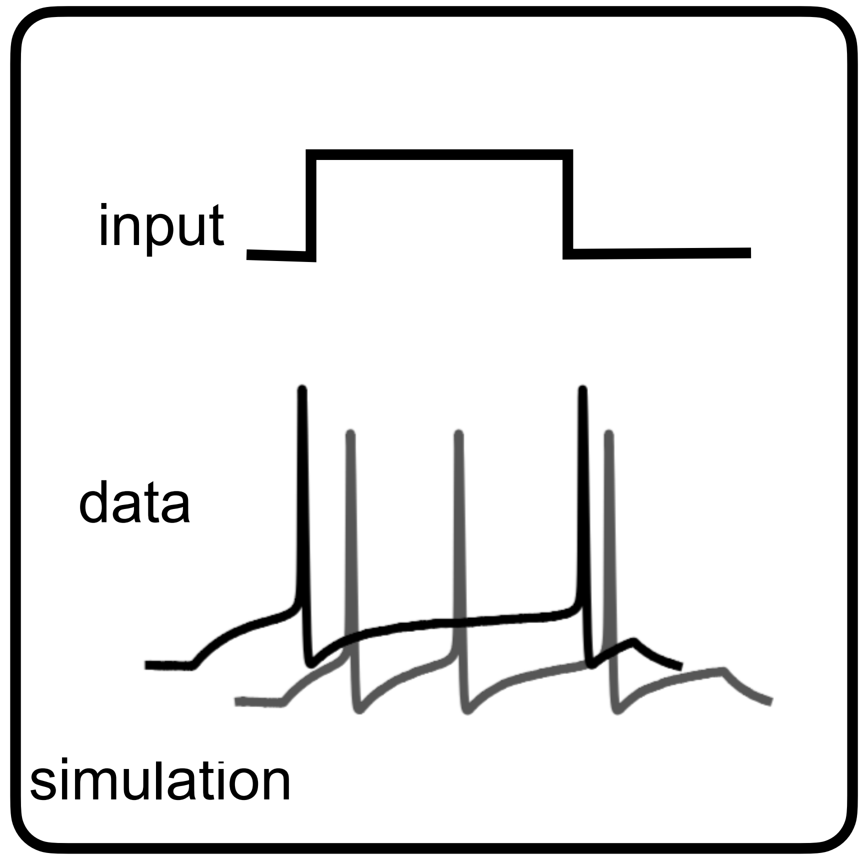


Mechanistic model
$C_m \frac{dV}{dt} = \sum_c \bar{g}_c g_c [E_c - V] + I$
Reject
or
refine
model
Insights/
Constraints
Generate predictions
input
data
simulation
Model
evaluation
input
simulation
Simulated
data
Collect empirical data

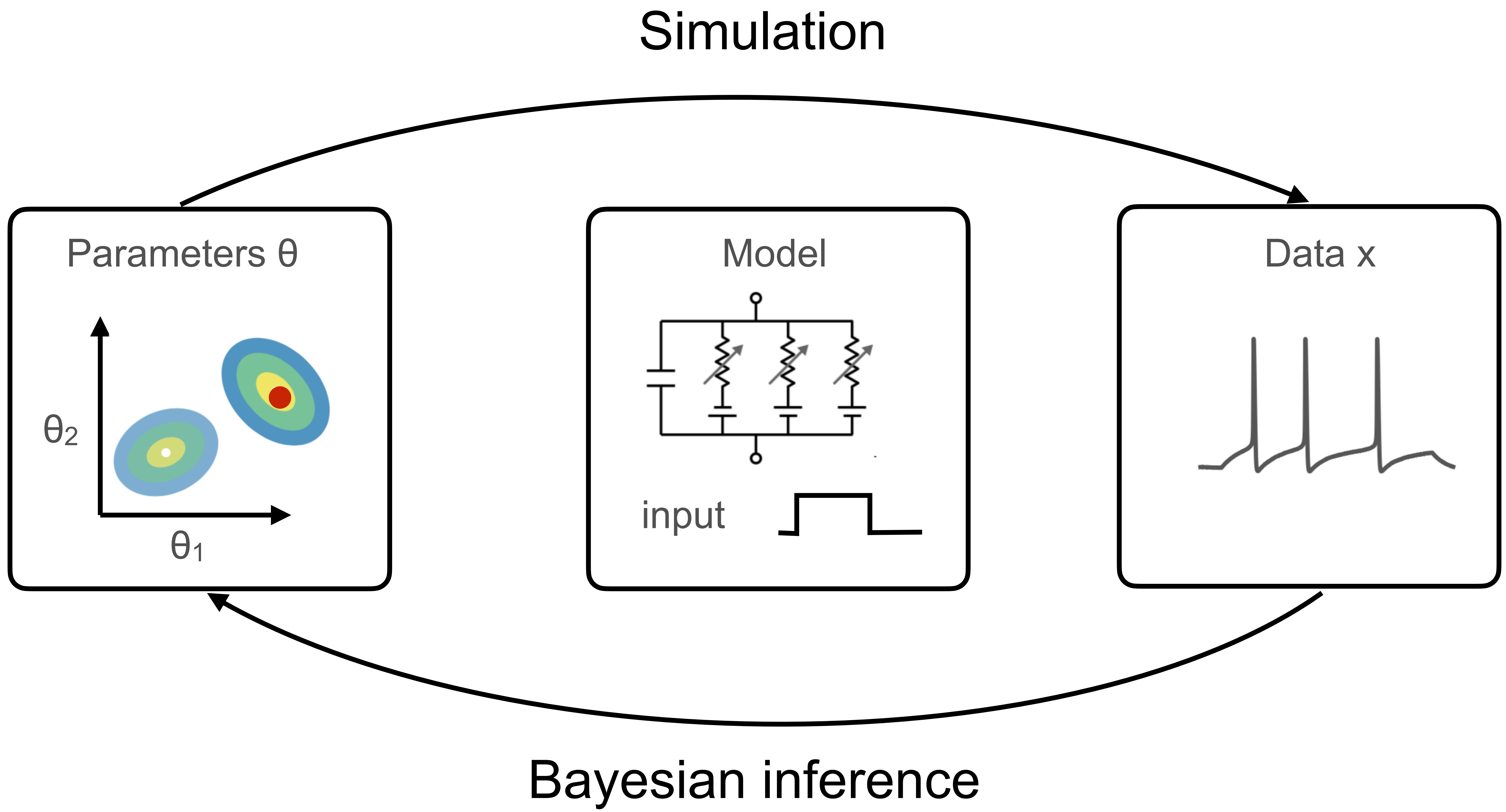
Key question: Which parameters of a mechanistic model are compatible with the data?

Answer: Bayesian inference!

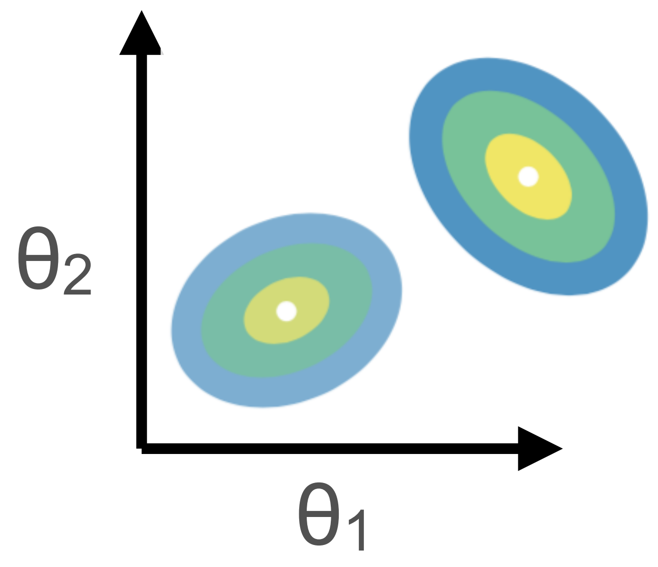
Model
evaluation



Bayesian inference finds model-parameters which are consistent with data and prior knowledge



$$p(\theta\,|\,x) \propto p(x\,|\,\theta)\pi(\theta)$$



$$p(\theta \mid x) \propto p(x \mid \theta)\pi(\theta)$$

posterior: hard to compute

likelihood: often not tractable

For many mechanistic models, we can **simulate x**, but we cannot (easily) evaluate the likelihood **p(x|θ)**.

Models often defined through **black-box** simulators.

➡ A solution: simulation-based inference!

1.4 An intuitive first approach: Approximate Bayesian Computation

based on tutorial by Wilkinson 2016

Approximate Bayesian Computation

- ABC algorithms are a collection of methods used for performing inference on simulators:
  1. They do not require explicit knowledge of the likelihood function
  2. Inference is done using simulations from the model (they are ‘likelihood-free’)

- ABC methods are popular in biological disciplines, particularly genetics:
  1. Simple to implement
  2. Intuitive
  3. Embarrassingly parallelizable

Rejection Approximate Bayesian Computation. Rejection ABC

Rejection Algorithm
for $n = 1 \ldots N$ **do** sample $\boldsymbol{\theta}_n \sim p(\boldsymbol{\theta})$ accept $\boldsymbol{\theta}_n$ with probability $p(\mathbf{x}_o \mid \boldsymbol{\theta}_n)$

Accepted θ are draws from the posterior distribution $p(\theta \mid x_o)$.

If the likelihood $p(x_o \mid \theta)$ is unknown:

'Mechanical' Rejection Algorithm
for $n = 1 \ldots N$ **do** sample $\boldsymbol{\theta}_n \sim p(\boldsymbol{\theta})$ simulate $\mathbf{x}_n$ from model (equivalently, sample $\mathbf{x}_n \sim p(\mathbf{x} \mid \boldsymbol{\theta}_n)$) accept $\boldsymbol{\theta}_n$ if $\mathbf{x}_n = \mathbf{x}_o$, i.e., if model output equals observation

Rejection Approximate Bayesian Computation. Rejection ABC

If $p(x_o)$ is small (or x_o continuous), we will rarely accept any θ. Instead, there is an approximate version:

Uniform Rejection Algorithm

---

for $n = 1 \ldots N$ **do**
  sample $\boldsymbol{\theta}_n \sim p(\boldsymbol{\theta})$
  sample $\mathbf{x}_n \sim p(\mathbf{x}|\boldsymbol{\theta}_n)$
  accept $\boldsymbol{\theta}_n$ if $\rho(\mathbf{x}_n, \mathbf{x}_o) \leq \epsilon$

ϵ reflects the tension between computability and accuracy:

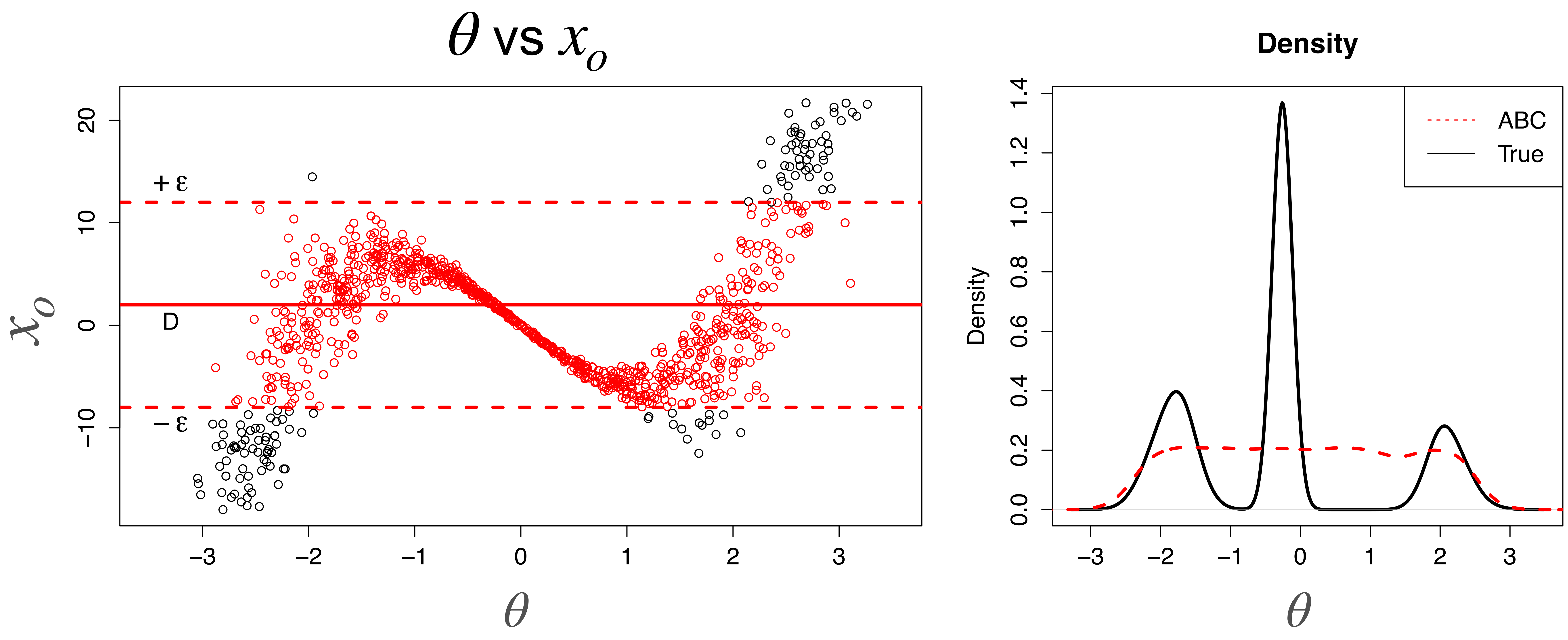
- As $\epsilon \to \infty$, we get observations from the prior $p(\theta)$.
- If $\epsilon = 0$, we generate observations from $p(\theta \mid x_o)$.

An example

$$\theta \sim \mathcal{U}[-10,10], x \sim \mathcal{N}(2(\theta+1)\theta(\theta-2), 0.1+\theta^2)$$

$$\rho(x, x_o) = |x - x_o|, x_o = 2$$

Rejection ABC. $\epsilon = 10$



$$\theta \sim \mathcal{U}[-10,10], x \sim \mathcal{N}(2(\theta+1)\theta(\theta-2), 0.1+\theta^2)$$

$$\rho(x, x_o) = |x - x_o|, x_o = 2$$

image from Wilkinson 2016 tutorial

Rejection ABC. $\epsilon = 7.5$

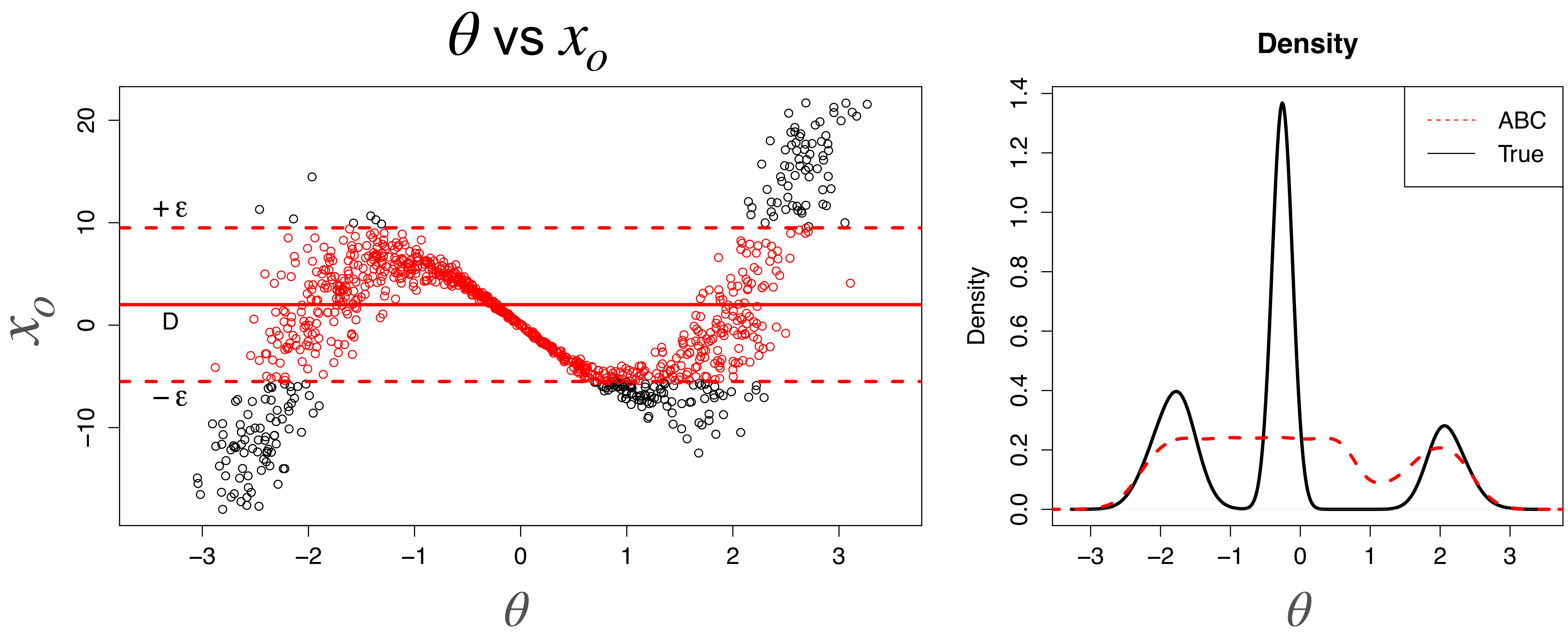
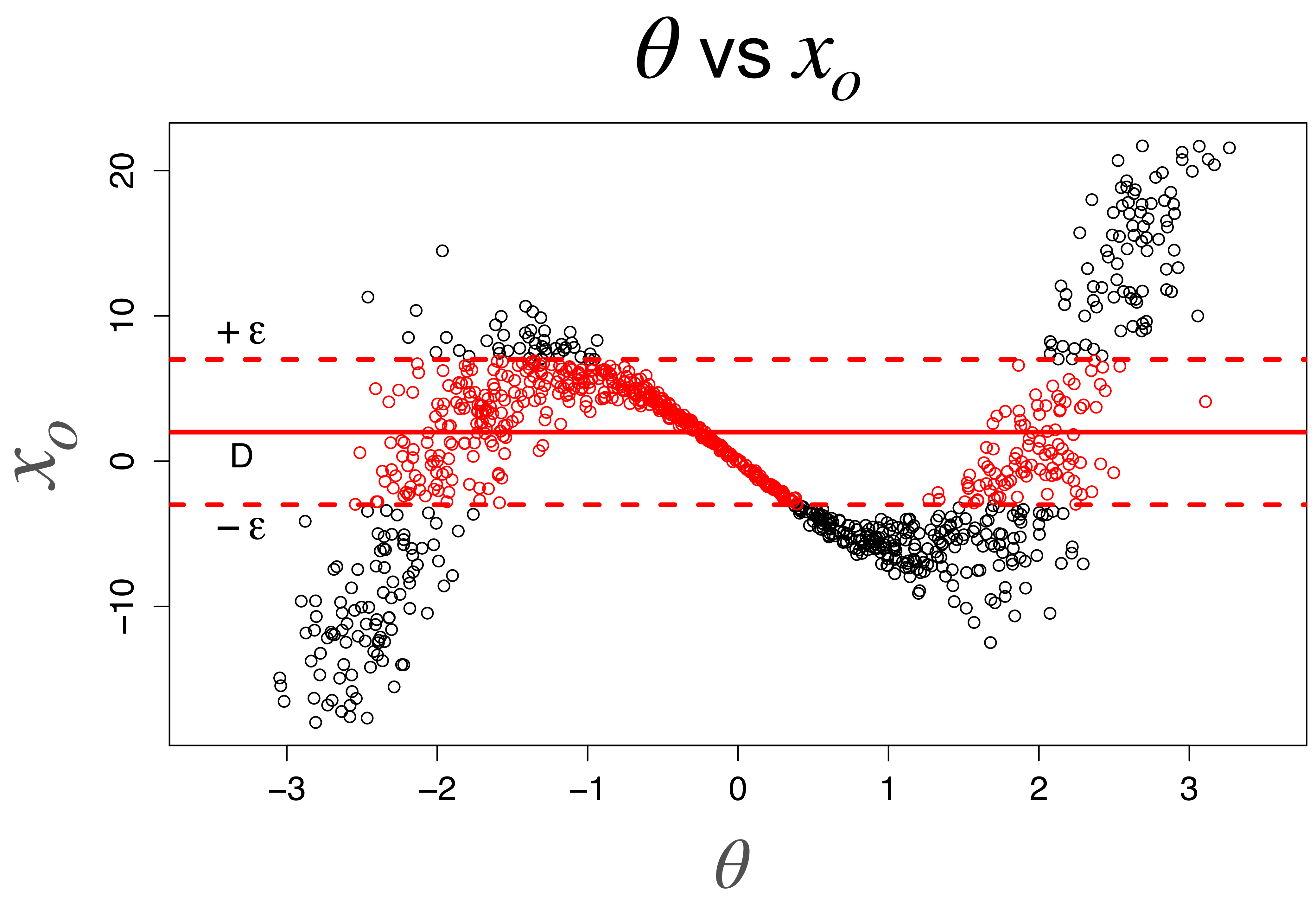


image from Wilkinson 2016 tutorial

Rejection ABC. $\epsilon = 5$



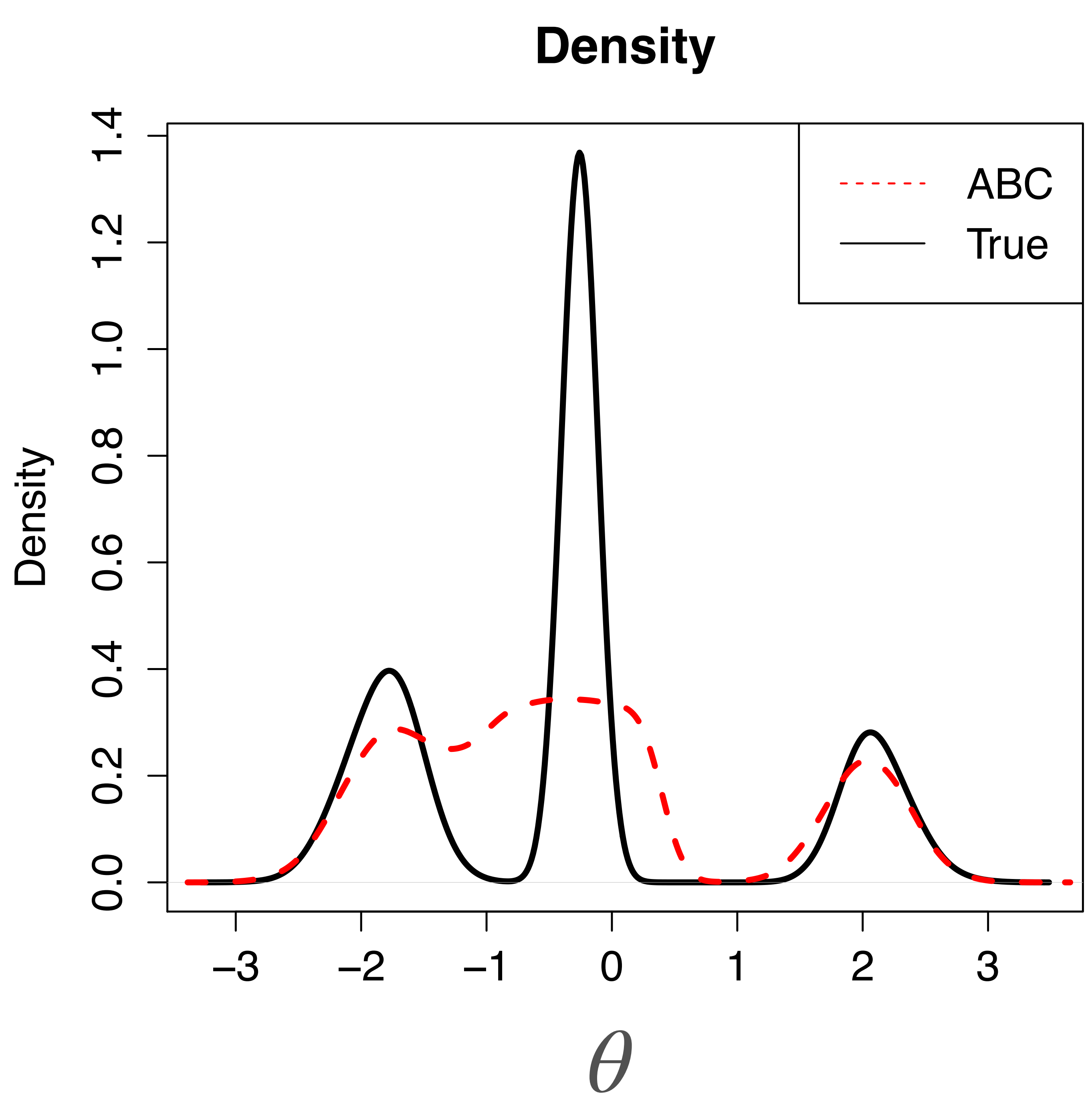


image from Wilkinson 2016 tutorial

Rejection ABC. $\epsilon = 2.5$

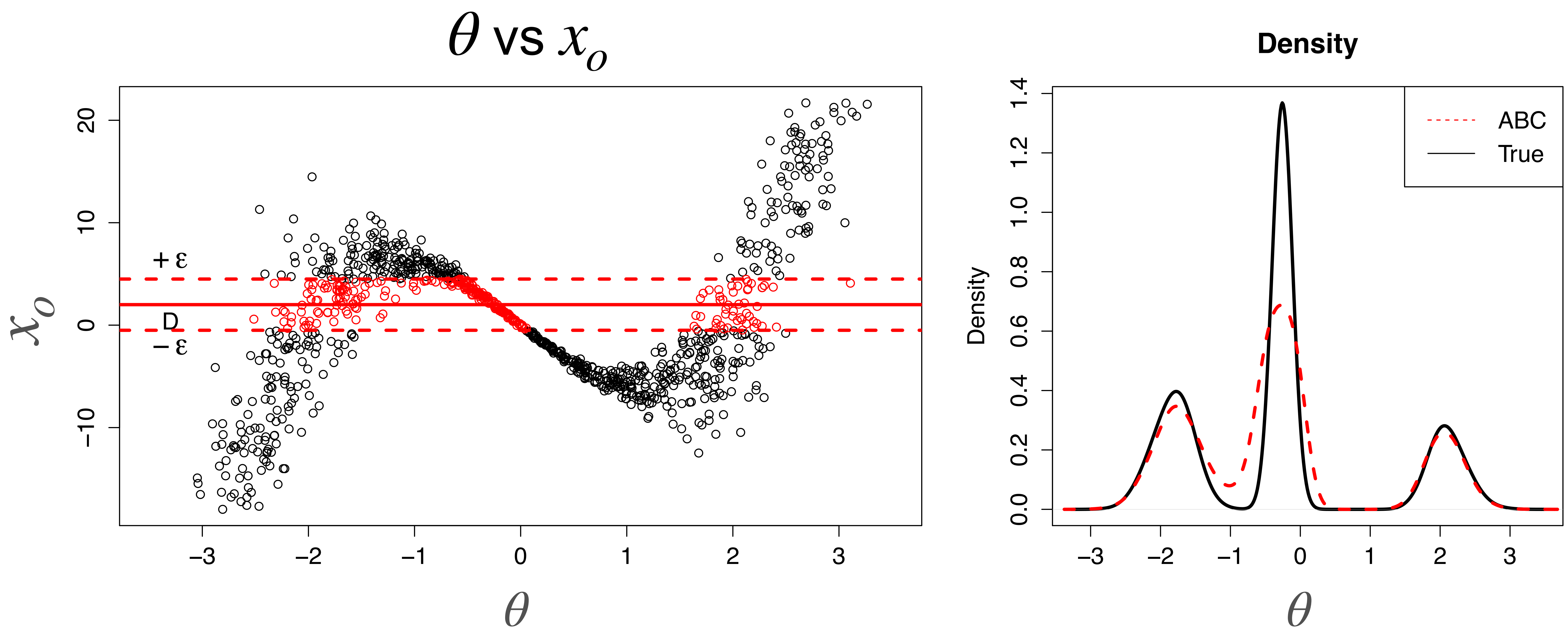


image from Wilkinson 2016 tutorial

Rejection ABC. $\epsilon = 1$

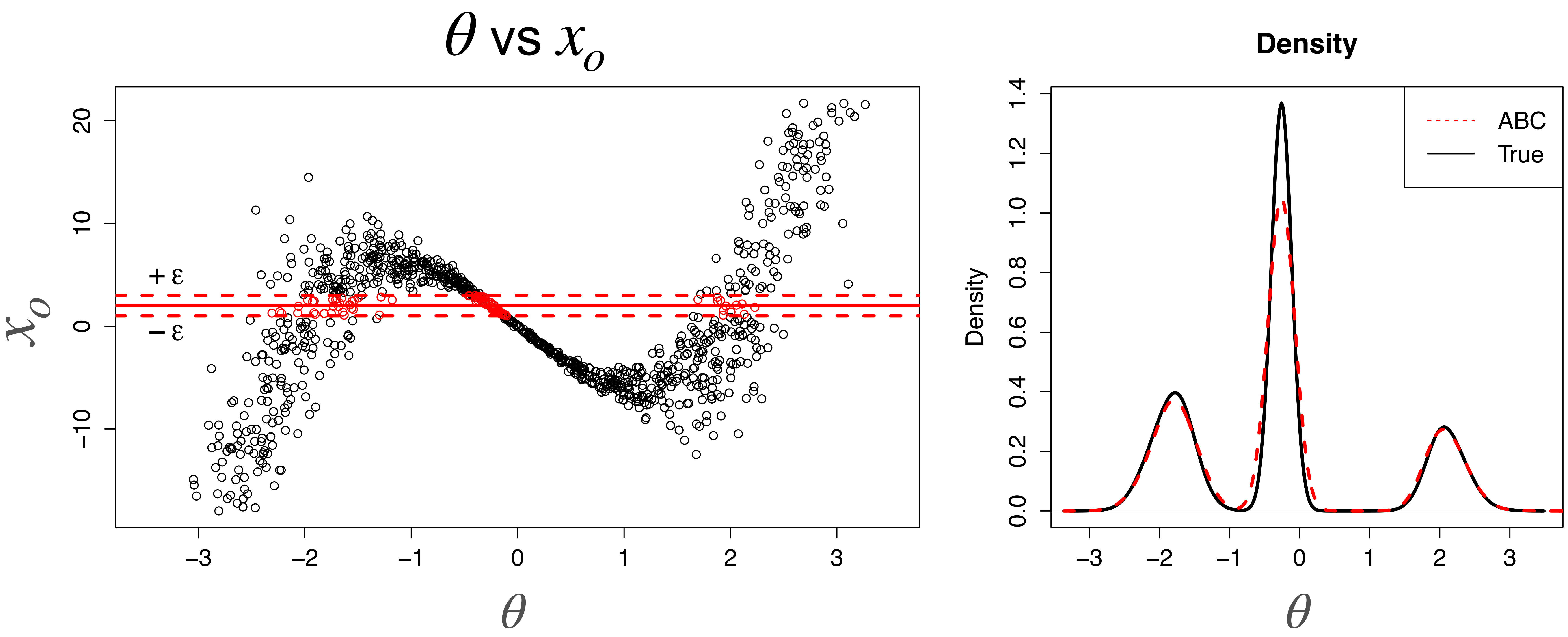


image from Wilkinson 2016 tutorial

Rejection ABC

If the data are too high dimensional, we never observe simulations that are 'close' to the measured data - curse of dimensionality.

Reduce the dimensionality using summary statistics $S(x_o)$.

Approximate Rejection Algorithm With Summaries

for $n = 1 \dots N$ **do**
- sample $\boldsymbol{\theta}_n \sim p(\boldsymbol{\theta})$
- sample $\mathbf{x}_n \sim p(\mathbf{x}|\boldsymbol{\theta}_n)$
- accept $\boldsymbol{\theta}_n$ if $\rho(S(\mathbf{x}_n), S(\mathbf{x}_o)) \leq \epsilon$

Key challenges for ABC

Accuracy in ABC is determined by:

1. Tolerance ϵ, which controls the 'ABC error'
   - there are more efficient algorithms that allow us to use small ϵ
   - constrained by how much computation we can do. Rules out expensive simulators

2. Summary statistic $S(x_o)$, which controls 'information loss'
   - inference is based on $p(\theta \mid S(x_o))$ rather than $p(\theta \mid x_o)$
   - expert-knowledge and machine-learning tools can be used to find informative summaries

Some last comments about ABC

- Can we get more accurate and efficient algorithms (scaling to more parameters and summary statistics)?

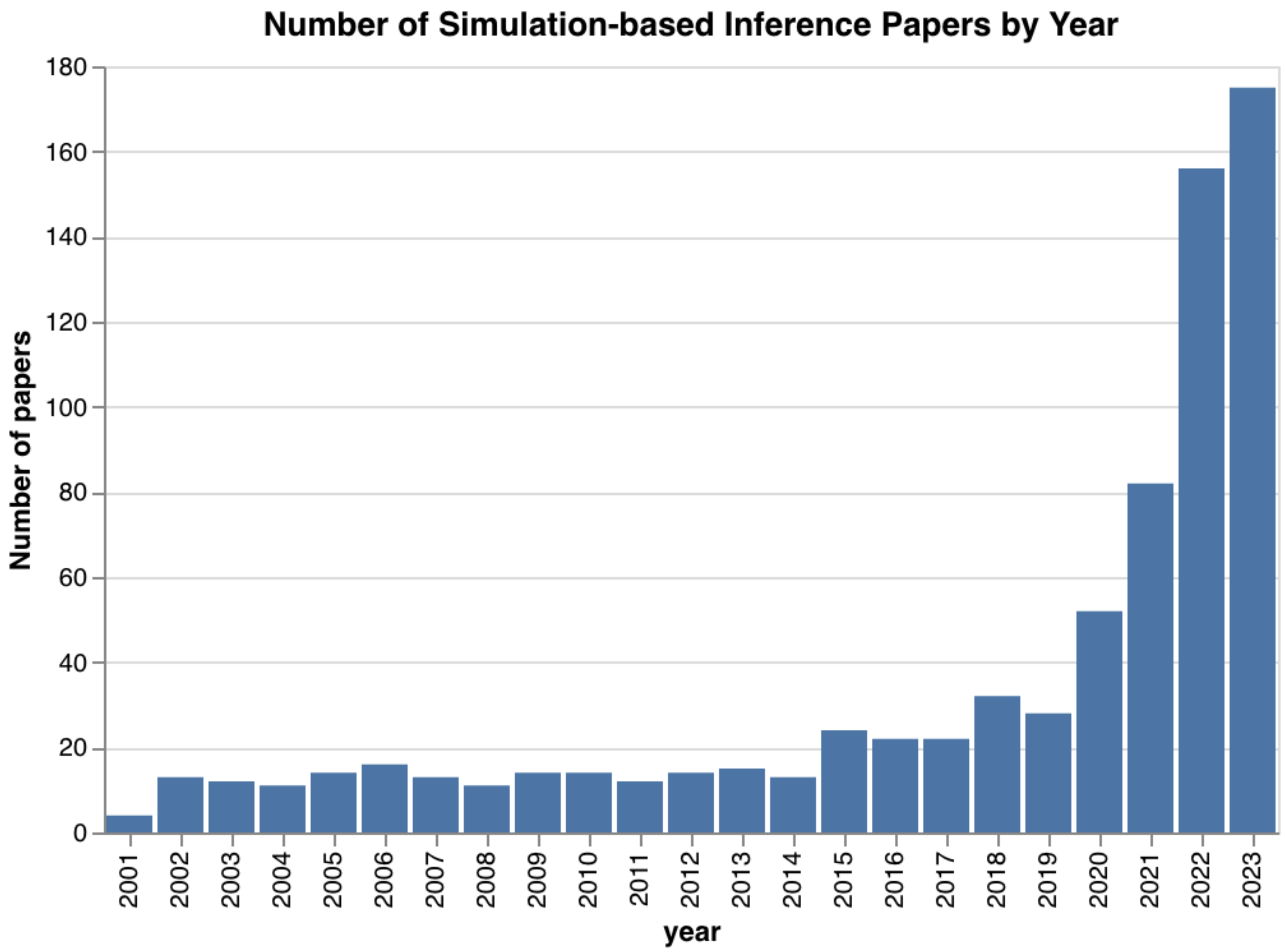
- How to choose summary statistics?

- How to deal with expensive simulators?

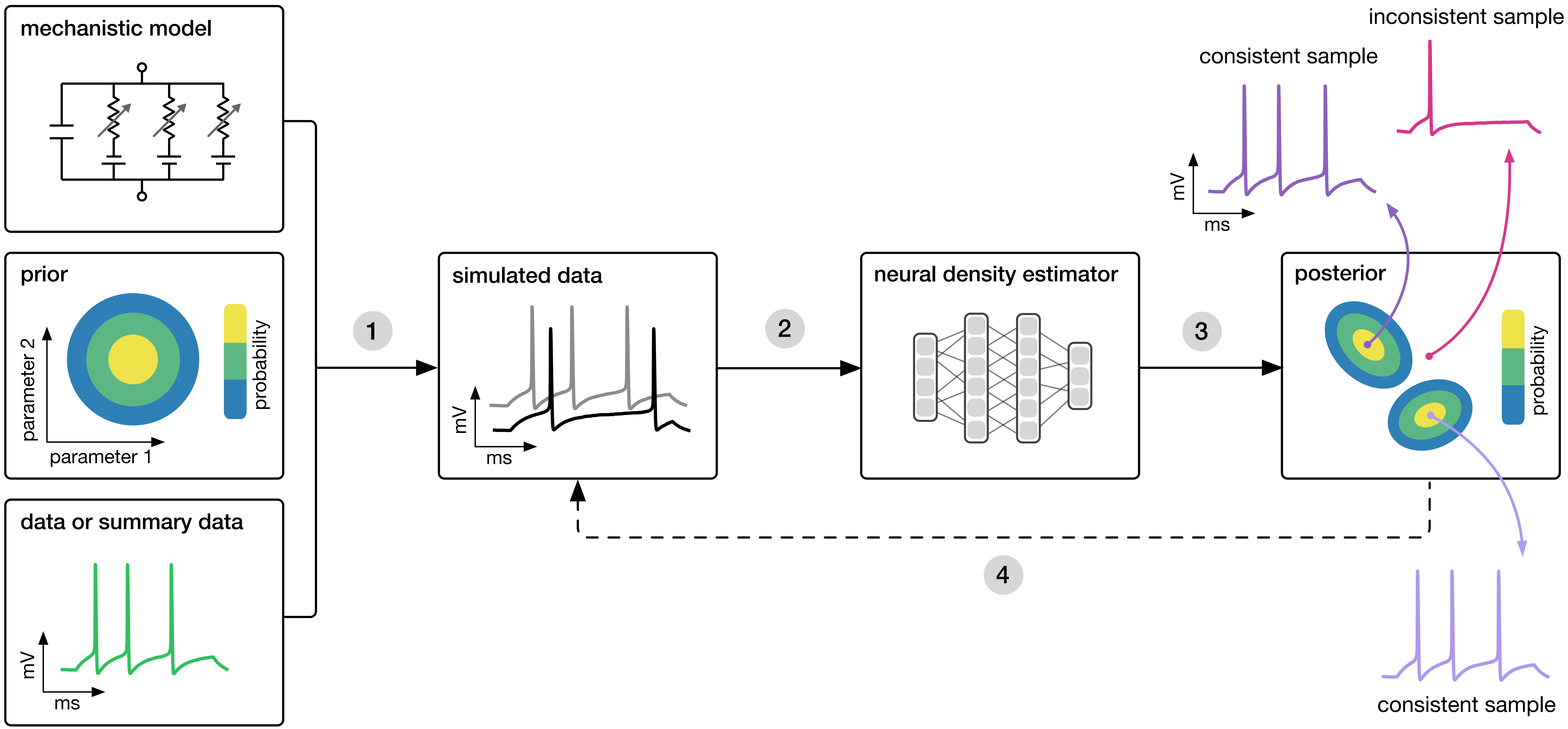
- Many, many ABC algorithms (SMC-ABC, Regression-ABC, GP-ABC, Hamiltonian-ABC…)

1.4 Can we do better? Simulation-based inference with neural networks

Simulation-based inference is rapidly expanding thanks to recent developments in (probabilistic) deep learning



Teaser: train neural networks to identify data-compatible models



Some general aims of the workshop

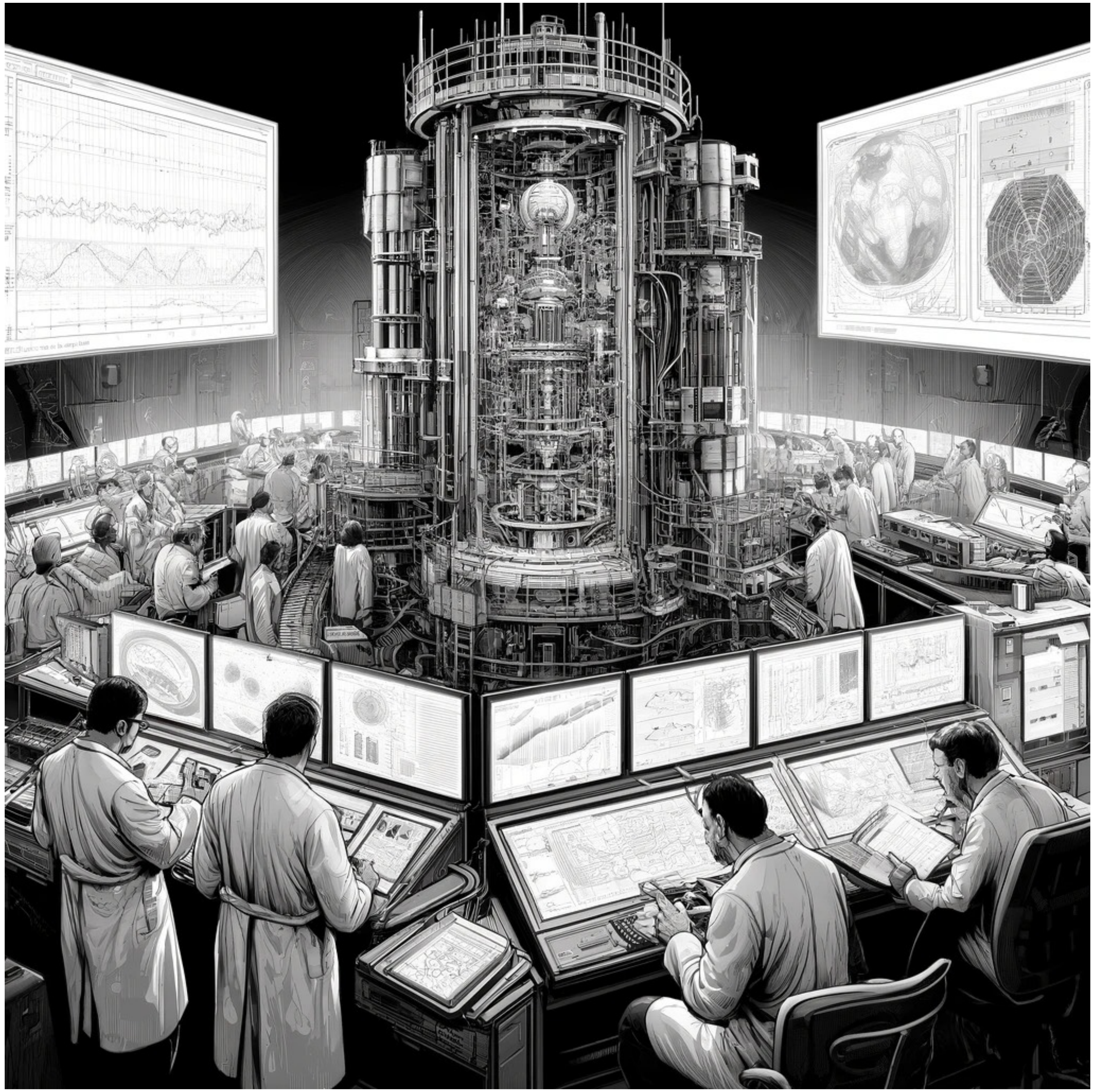
- An appreciation of how simulators and data can be combined to generate insights.
- Probability theory as the mathematical language for performing inference.
- An overview of how the latest advances in machine learning (in particular, neural networks) can be used for Bayesian inference.

Workshop Outline

- Lecture 1: Introduction to simulation-based inference, ABC ✓
- Lecture 2: Neural density estimation, Normalising flows, NPE
- Lecture 3: Simulation-based inference hands-on tutorial
- Lectures 4-5: Advanced topics and applications

Found a cool scientific question involving a simulator?

Let’s discuss it in the group!



Simulation-based inference: How to go from simulator and data to insight?

Lecture 1: Introduction

- For mechanistic insights, we need to combine mechanistic models and data.
- Probability theory as a framework for reasoning with data, and quantifying our uncertainty about it.
- How can we make *causal* statements from data?

- **Simulation-based inference: a toolkit for making sense of the real world with simulations**

Further reading on ABC

- Turner, Brandon M., and Trisha Van Zandt. "A tutorial on approximate Bayesian computation." Journal of Mathematical Psychology 56.2 (2012): 69-85.

- Lintusaari, Jarno, et al. "Fundamentals and recent developments in approximate Bayesian computation." Systematic biology 66.1 (2017): e66-e82.

- ABC Tutorial (video): https://cds.cern.ch/record/2067048