TransferLab

appliedAI institute for europe

Simulated data is all you need: Bayesian parameter inference for scientific simulators with SBI

Jan Boelts (Teusen) | EuroSciPy2024 | August 30th, 2024 | Szczecin

A joint initiative

UNTERNEHMER TUM | IPAI

About me

- AI Researcher at appliedAI

About me

- AI Researcher at appliedAI

- Background
  - Cognitive Science
  - Computational Neuroscience
  - “Machine Learning for Science”

About me

- AI Researcher at appliedAI

- Background
  - Cognitive Science
  - Computational Neuroscience
  - “Machine Learning for Science”
- Maintainer of **sbi**

About me

- AI Researcher at appliedAI

- Background
  - Cognitive Science
  - Computational Neuroscience
  - “Machine Learning for Science”
- Maintainer of **sbi**
- **X, Mastodon**: @janfiete

About me

- AI Researcher at appliedAI

- Background
  - Cognitive Science
  - Computational Neuroscience
  - “Machine Learning for Science”
- Maintainer of **sbi**
- **X, Mastodon**: @janfiete
- **GitHub**: @janfb

TransferLab

About me

- AI Researcher at appliedAI

- Background
  - Cognitive Science
  - Computational Neuroscience
  - “Machine Learning for Science”
- Maintainer of **sbi**
- **X, Mastodon**: @janfiete
- **GitHub**: @janfb

About

About me

- AI Researcher at appliedAI
- Background
  - Cognitive Science
  - Computational Neuroscience
  - “Machine Learning for Science”
- Maintainer of **sbi**
- **X, Mastodon**: @janfiete
- **GitHub**: @janfb

About

- Non-profit (for-impact) organization

About me

- AI Researcher at appliedAI
- Background
  - Cognitive Science
  - Computational Neuroscience
  - “Machine Learning for Science”
- Maintainer of **sbi**
- **X, Mastodon**: @janfiete
- **GitHub**: @janfb

About

- Non-profit (for-impact) organization
- **transferlab.ai**

About me

- AI Researcher at appliedAI
- Background
  - Cognitive Science
  - Computational Neuroscience
  - “Machine Learning for Science”
- Maintainer of **sbi**
- **X, Mastodon**: @janfiete
- **GitHub**: @janfb

About

- Non-profit (for-impact) organization
- **transferlab.ai**
  - research

About me

- AI Researcher at appliedAI
- Background
  - Cognitive Science
  - Computational Neuroscience
  - “Machine Learning for Science”
- Maintainer of **sbi**
- **X, Mastodon**: @janfiete
- **GitHub**: @janfb

About

- Non-profit (for-impact) organization
- **transferlab.ai**
  - research
  - education

About me

- AI Researcher at appliedAI
- Background
  - Cognitive Science
  - Computational Neuroscience
  - “Machine Learning for Science”
- Maintainer of **sbi**
- **X, Mastodon**: @janfiete
- **GitHub**: @janfb

About

- Non-profit (for-impact) organization
- **transferlab.ai**
  - research
  - education
  - open source tools

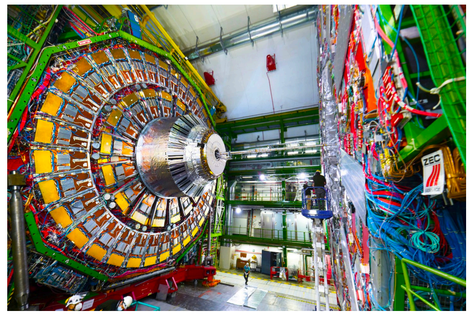
Outline

1. **Why** simulators?
2. **Why** Bayesian parameter inference?
3. **How** to do Bayesian inference for simulators?
4. **Demo:** How can you apply SBI to your simulator?

Outline

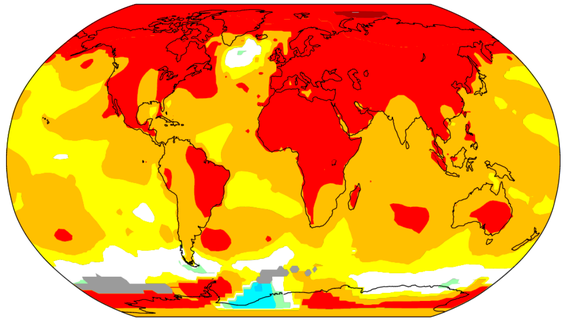
1. **Why** simulators?
2. **Why** Bayesian parameter inference?
3. **How** to do Bayesian inference for simulators?
4. **Demo:** How can you apply SBI to your simulator?

Simulators are used all across science



particle physics

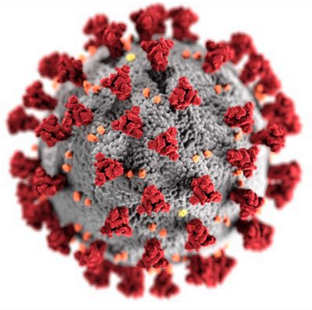
neuroscience



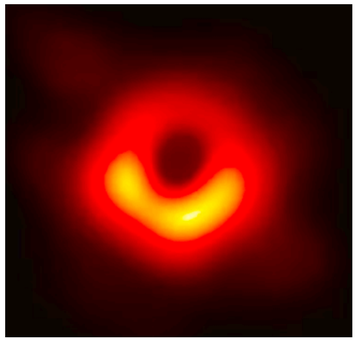
climate science

genomics

epidemiology



astrophysics



CERN; getty images; Wikipedia

What is a simulator?

Simulator: A computer program that takes input parameters θ and returns simulated data x

What is a simulator?

Simulator: A computer program that takes input parameters θ and returns simulated data x

Examples:

What is a simulator?

Simulator: A computer program that takes input parameters θ and returns simulated data x

Examples:

- Explicit **mechanistic** models, e.g., ODEs, PDEs: “simulator as numerical solver”

What is a simulator?

Simulator: A computer program that takes input parameters θ and returns simulated data x

Examples:

- Explicit **mechanistic** models, e.g., ODEs, PDEs: “simulator as numerical solver”
- Implicit models: “simulator as black-box computer simulation”

What is a simulator?

Simulator: A computer program that takes input parameters θ and returns simulated data x

Examples:

- Explicit **mechanistic** models, e.g., ODEs, PDEs: “simulator as numerical solver”
- Implicit models: “simulator as black-box computer simulation”
- Anything in-between: input parameter θ, output simulated data x

What is a simulator?

Simulator: A computer program that takes input parameters θ and returns simulated data x

Examples:

- Explicit **mechanistic** models, e.g., ODEs, PDEs: “simulator as numerical solver”
- Implicit models: “simulator as black-box computer simulation”
- Anything in-between: input parameter θ, output simulated data x

$$x = \text{simulator}(\theta)$$

What is a simulator?

Simulator: A computer program that takes input parameters θ and returns simulated data x

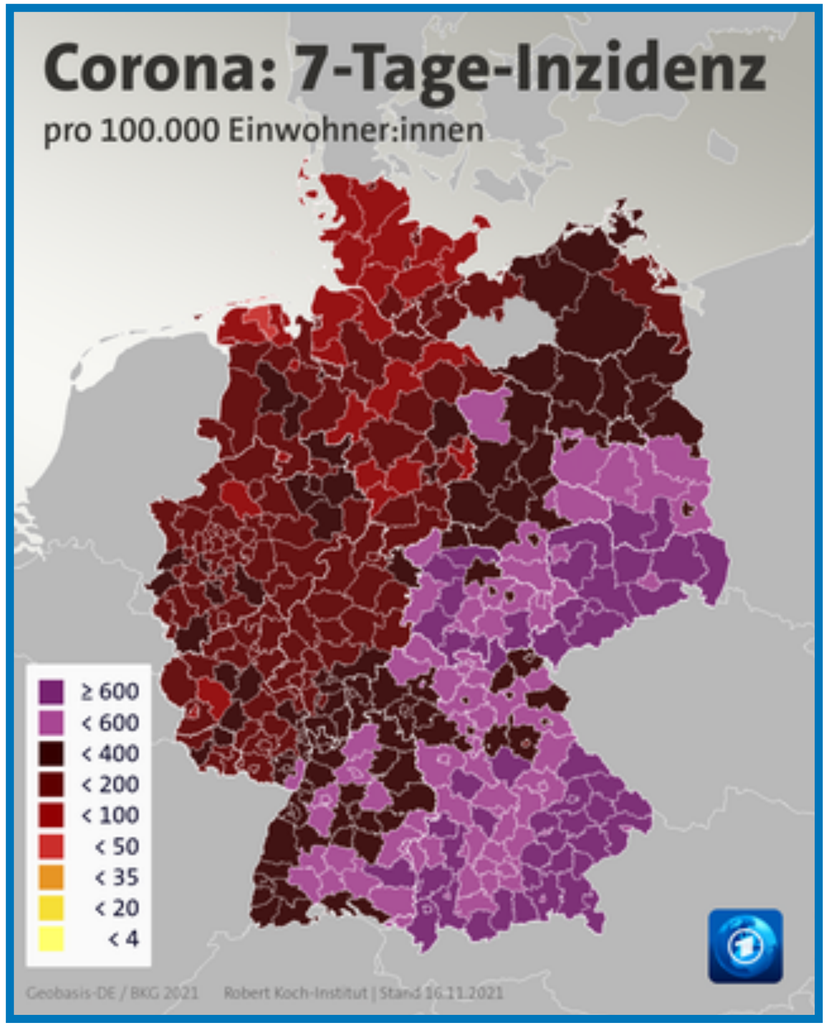
Examples:

- Explicit **mechanistic** models, e.g., ODEs, PDEs: “simulator as numerical solver”
- Implicit models: “simulator as black-box computer simulation”
- Anything in-between: input parameter θ, output simulated data x

$$x = \text{simulator}(\theta)$$

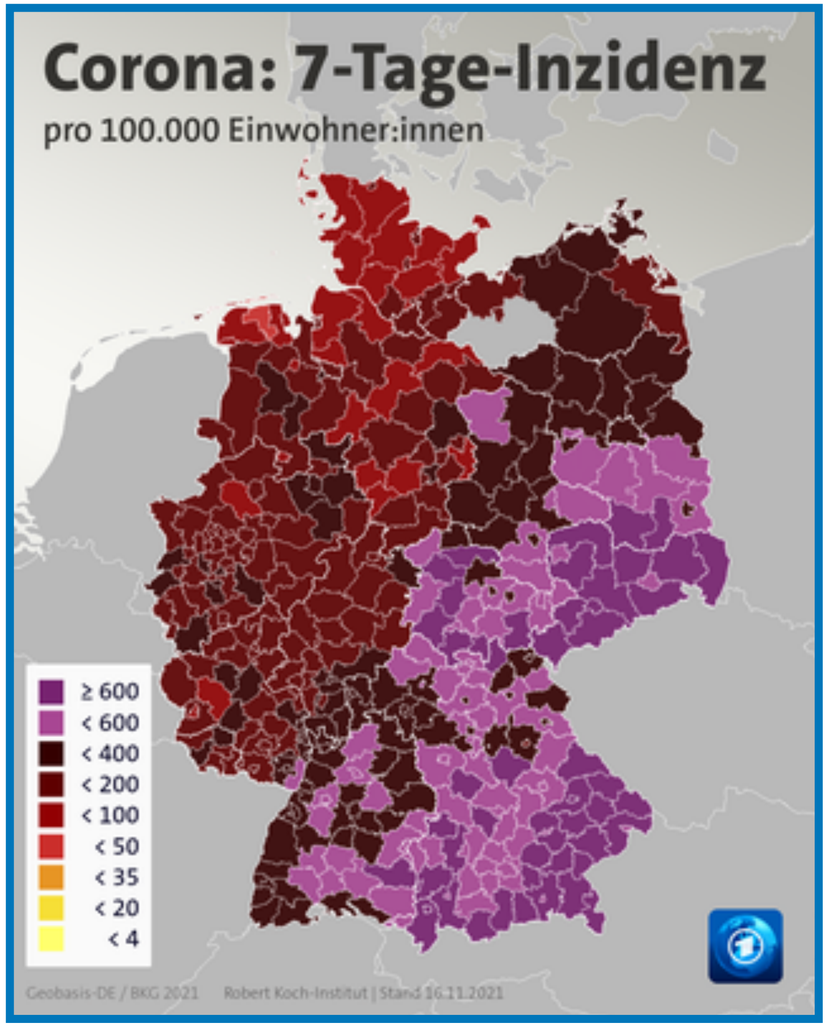
- **no access** to the inner workings (no likelihoods, no gradients)

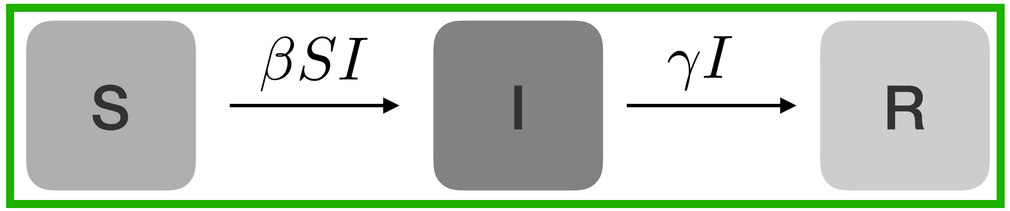
Example: The SIR Simulator for COVID-19



tagesschau.de, 2021; Dehning et al. 2020, Science

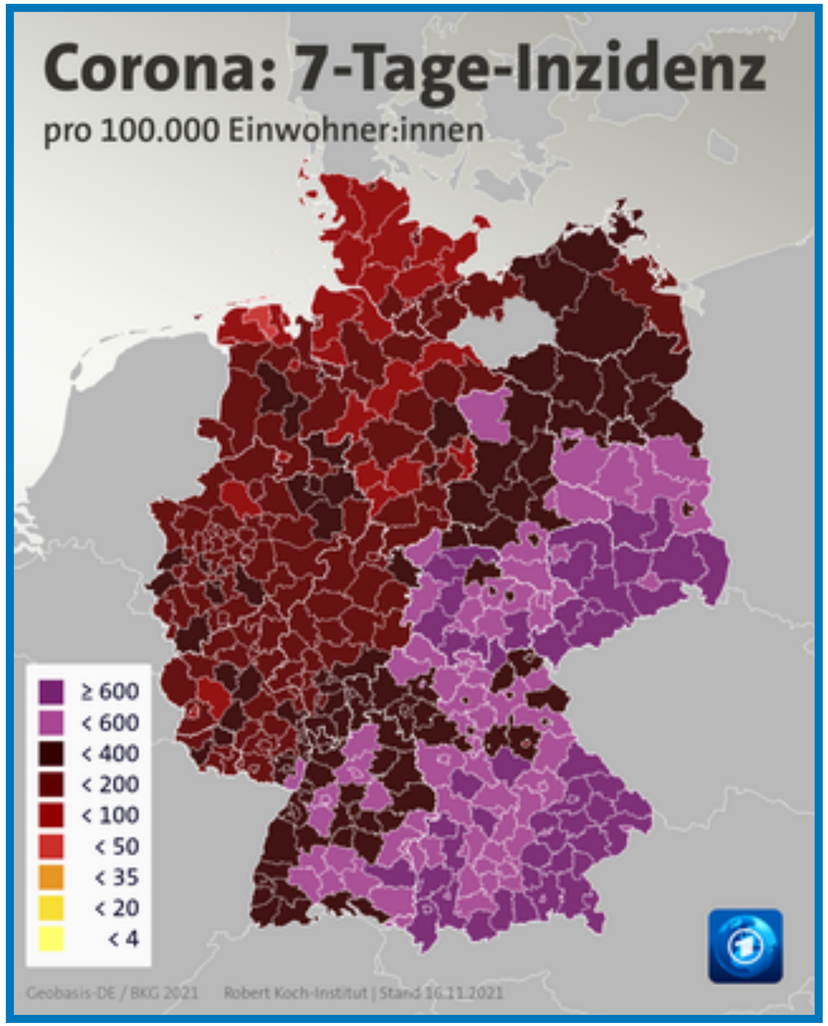
Example: The SIR Simulator for COVID-19

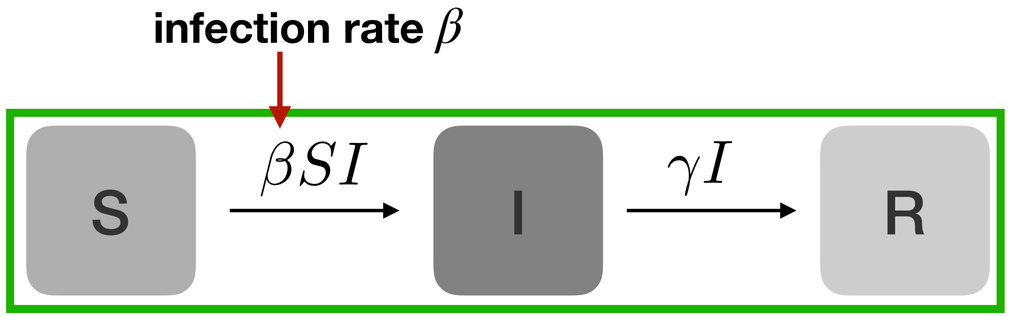




tagesschau.de, 2021; Dehning et al. 2020, Science

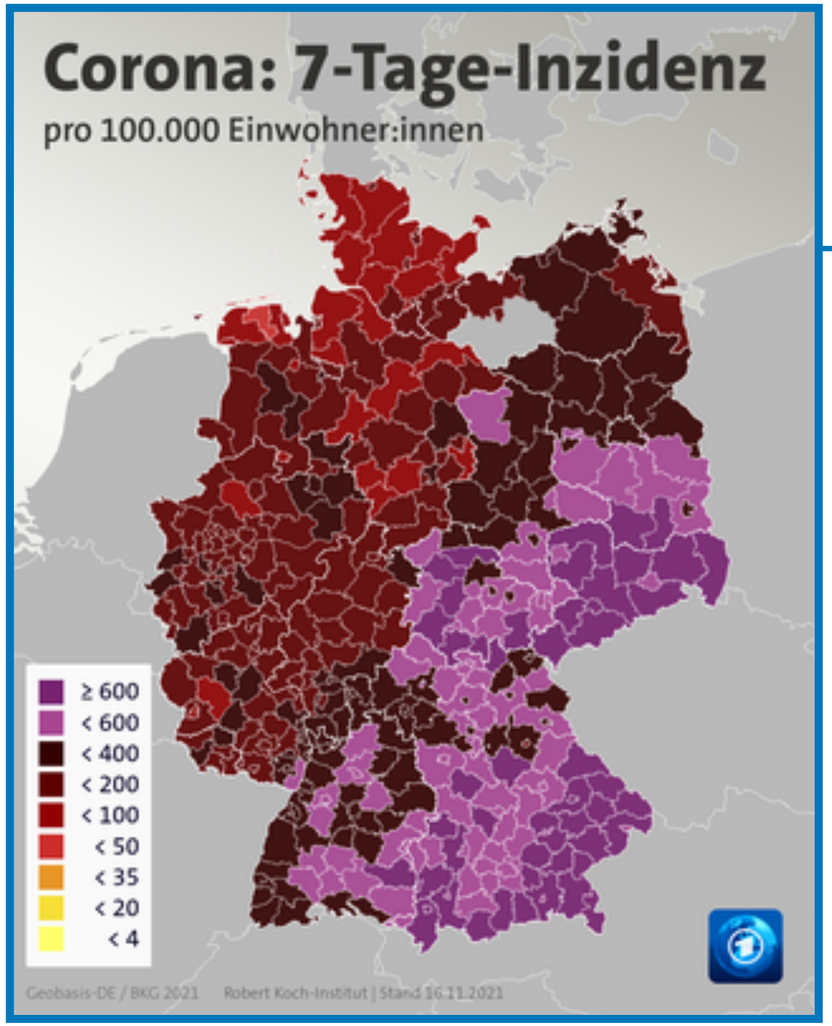
Example: The SIR Simulator for COVID-19

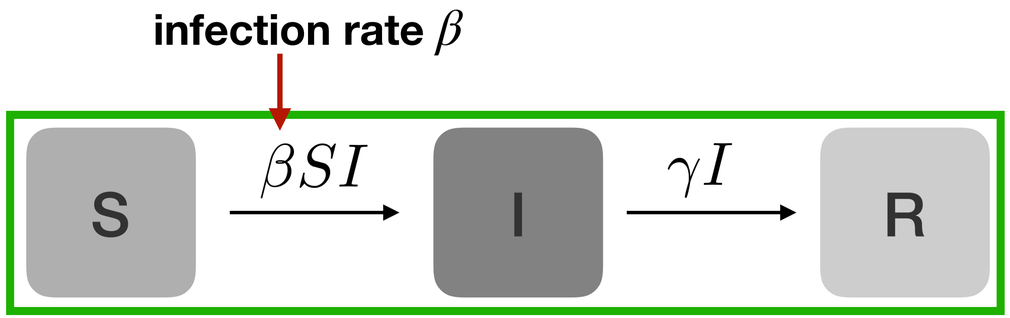


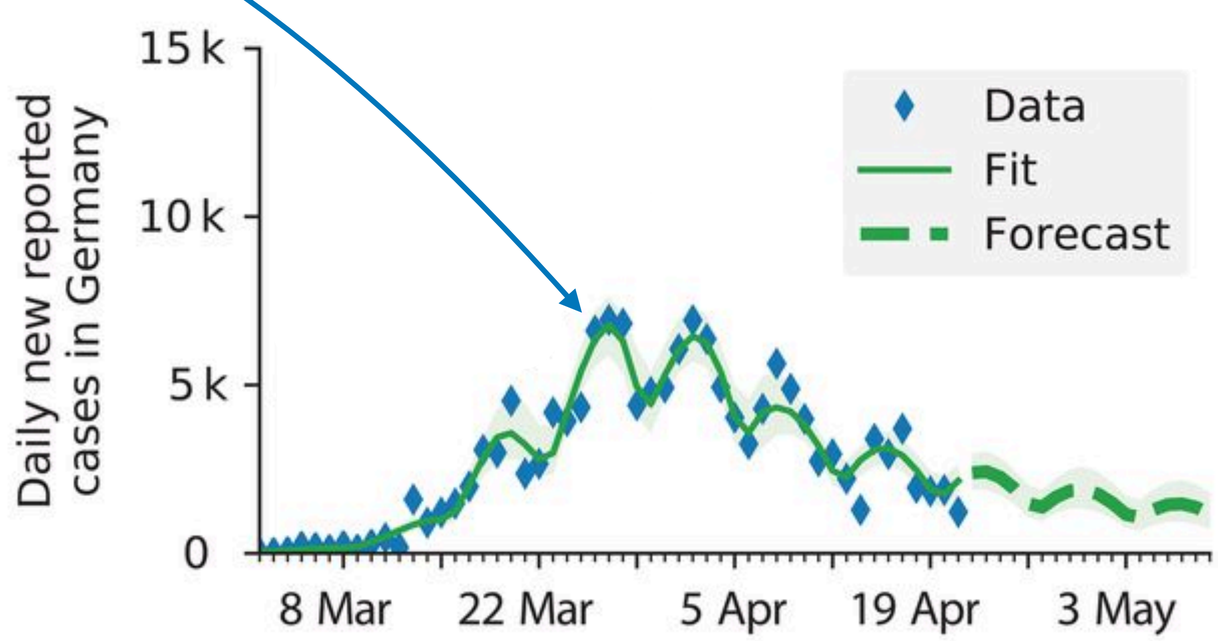


tagesschau.de, 2021; Dehning et al. 2020, Science

Example: The SIR Simulator for COVID-19

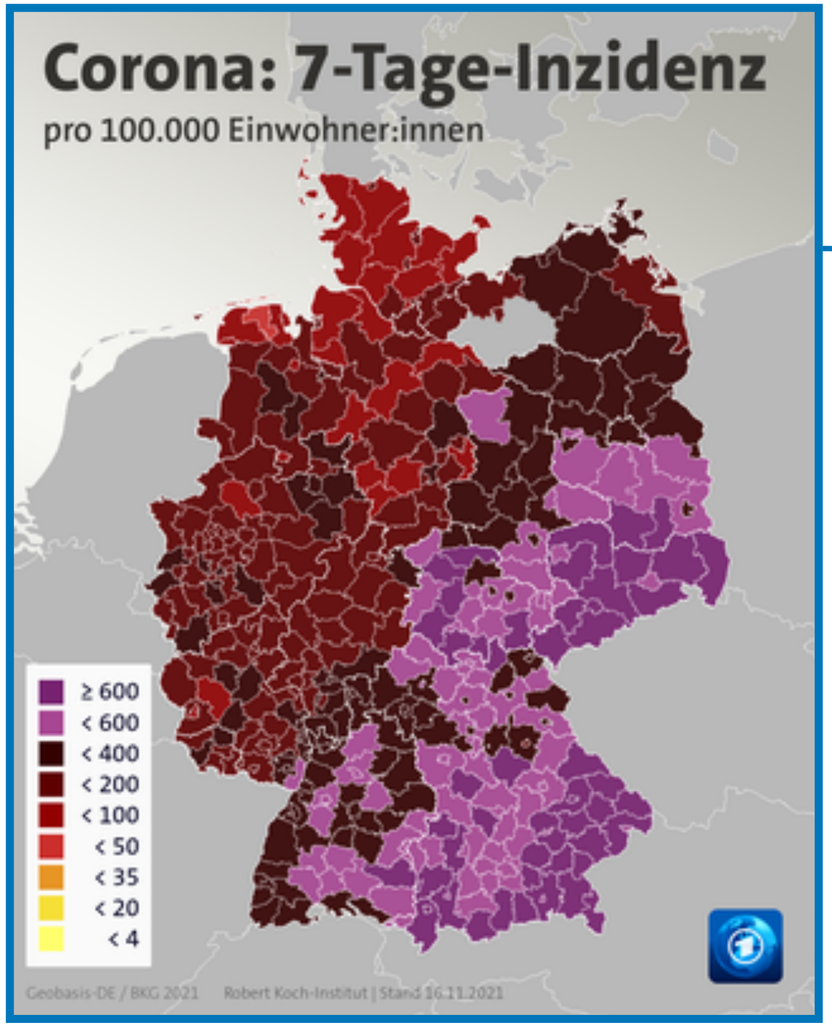


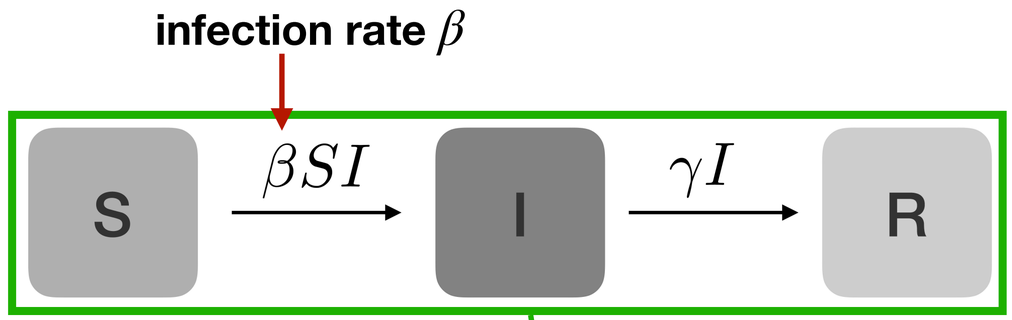


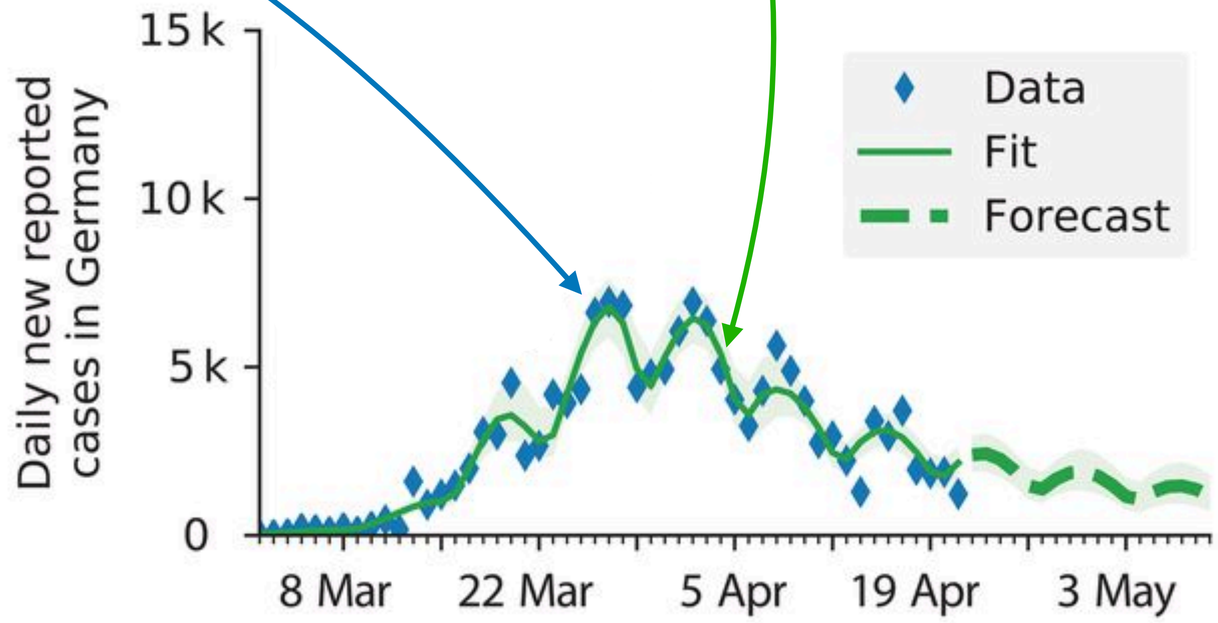


tagesschau.de, 2021; Dehning et al. 2020, Science

Example: The SIR Simulator for COVID-19

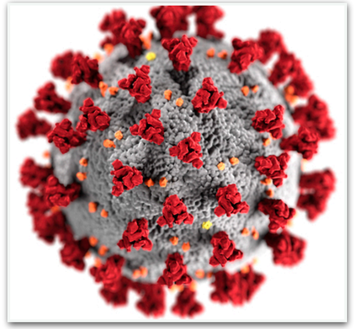






tagesschau.de, 2021; Dehning et al. 2020, Science

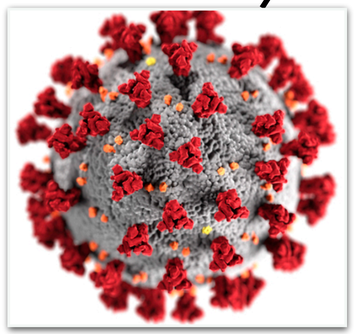
The simulation-based modeling approach

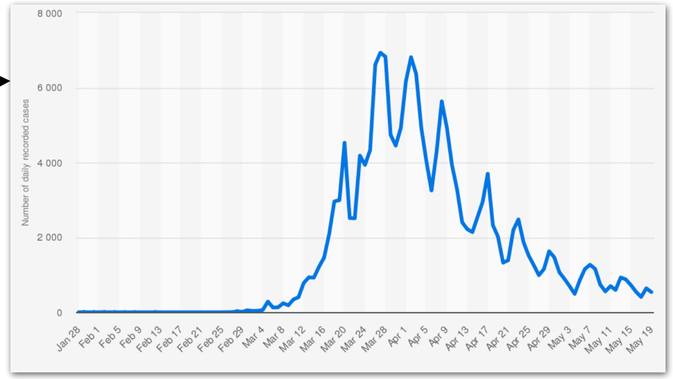


Paul Ehrlich Institut; WHO; Dehning et al. 2020

The simulation-based modeling approach

observe data

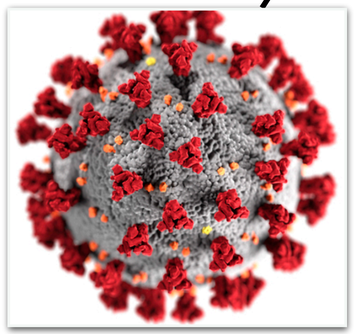


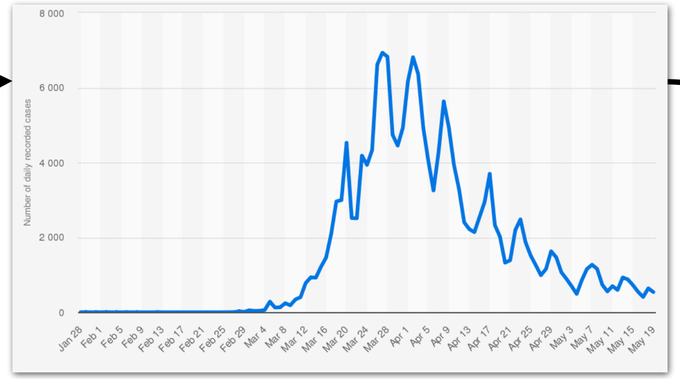


Paul Ehrlich Institut; WHO; Dehning et al. 2020

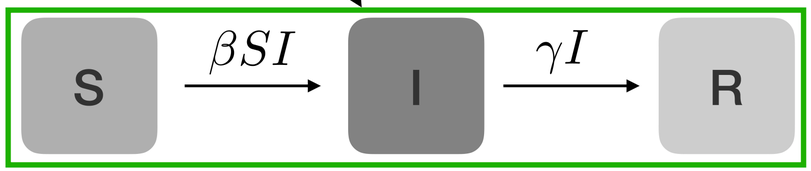
The simulation-based modeling approach

observe data





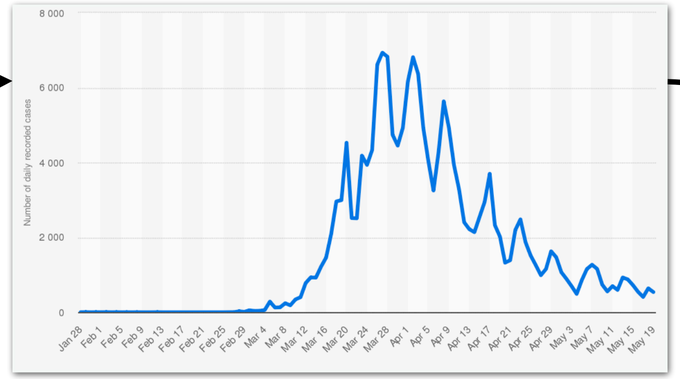
build a model



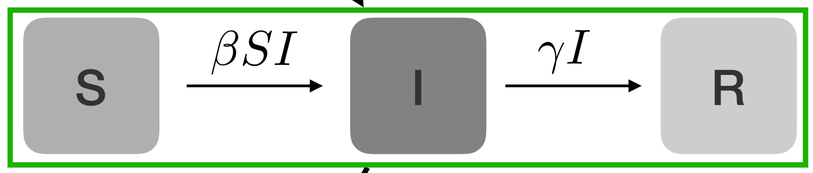
Paul Ehrlich Institut; WHO; Dehning et al. 2020

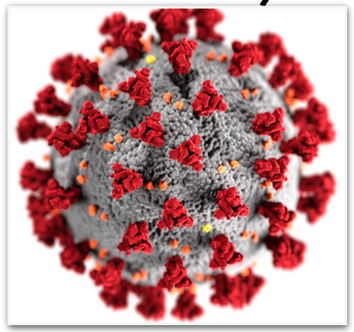
The simulation-based modeling approach

observe data

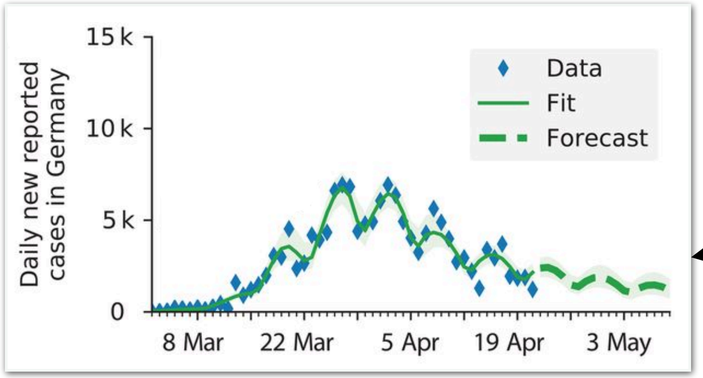


build a model





fit model to data

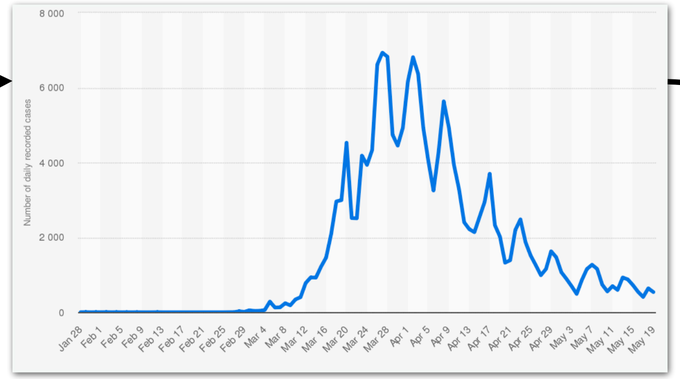


TransferLab

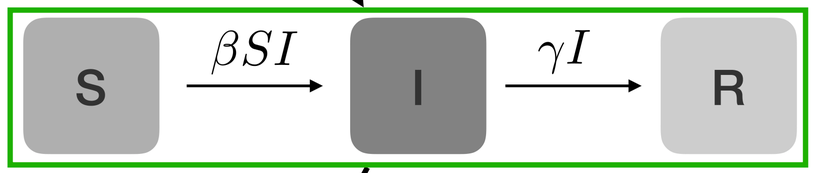
Paul Ehrlich Institut; WHO; Dehning et al. 2020

The simulation-based modeling approach

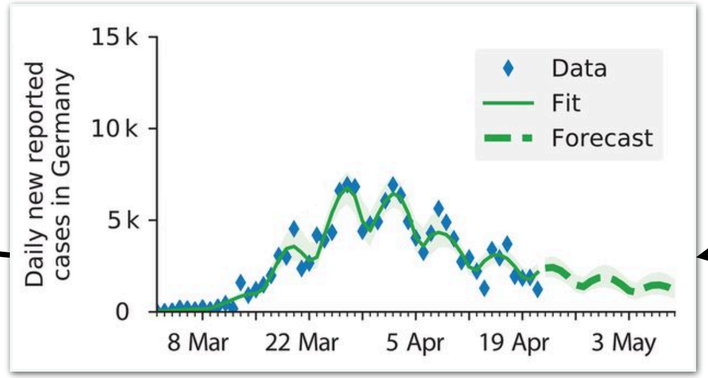
observe data



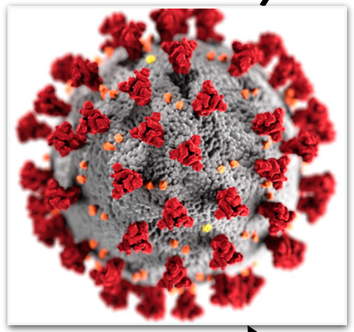
build a model



fit model to data



draw conclusions, make predictions

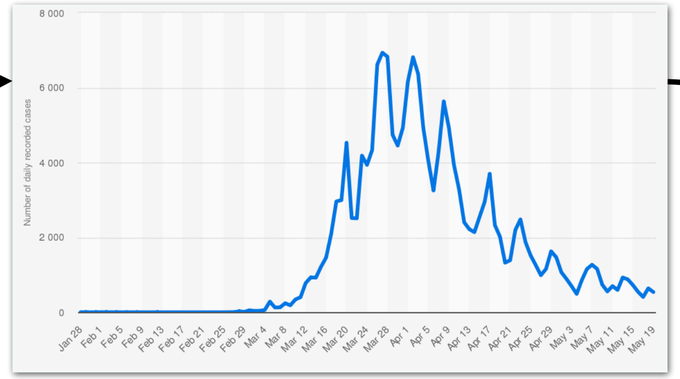


TransferLab

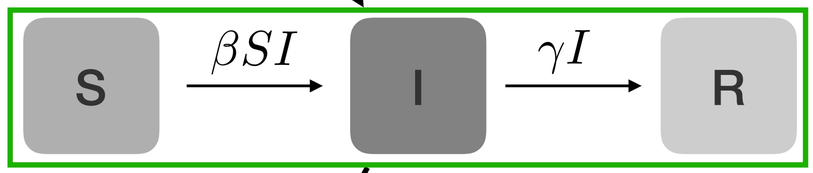
Paul Ehrlich Institut; WHO; Dehning et al. 2020

The simulation-based modeling approach

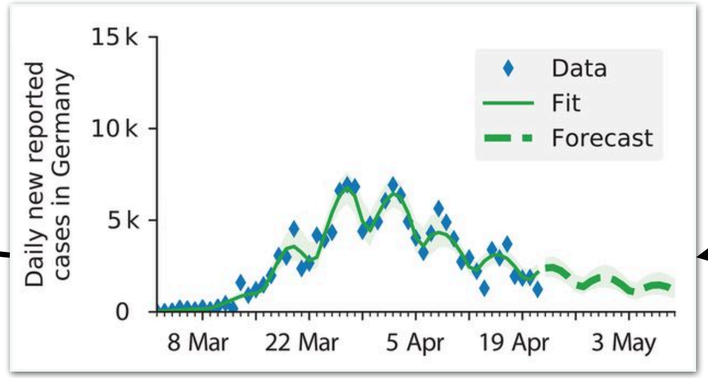
observe data



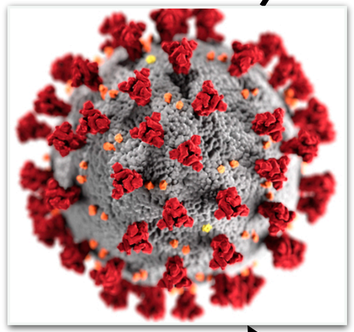
build a model



fit model to data



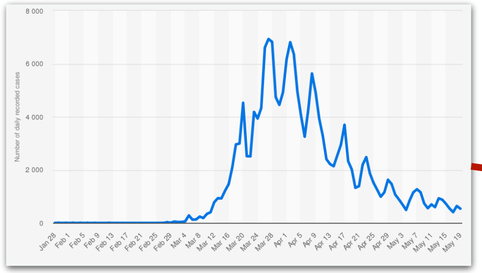
draw conclusions,
make predictions

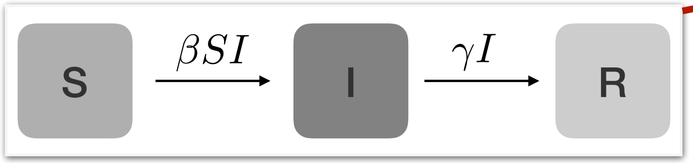


TransferLab

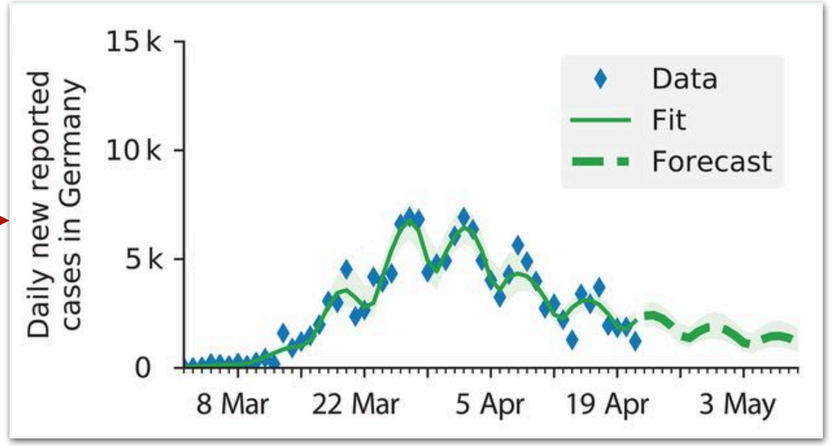
Paul Ehrlich Institut; WHO; Dehning et al. 2020

Central challenge: finding model parameters

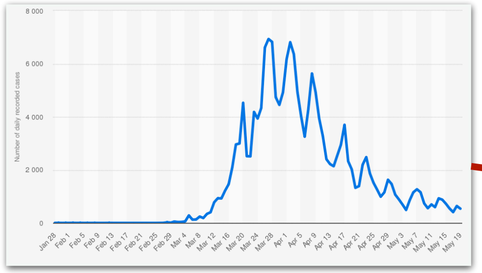


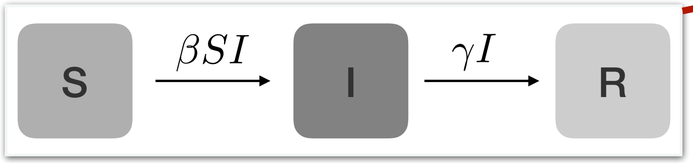


infection rate β
?
recovery rate γ

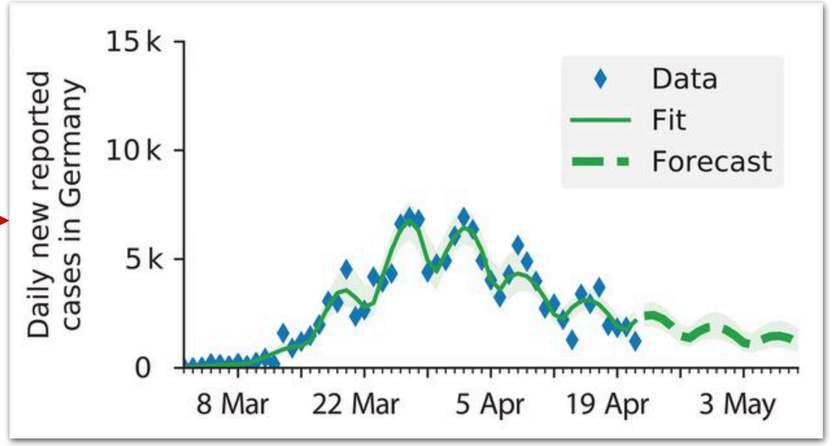


Central challenge: finding model parameters



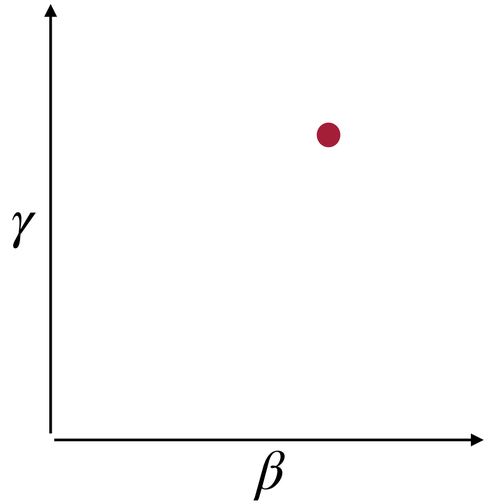


infection rate β
?
recovery rate γ



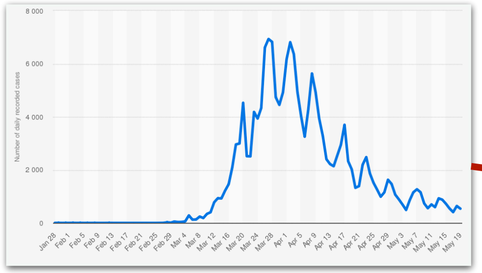
Naive approach

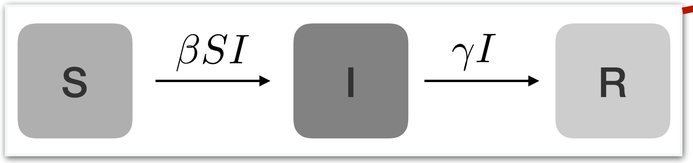
- find single best-fitting parameters
  - hand tuning, grid search
  - optimization, genetic algorithms



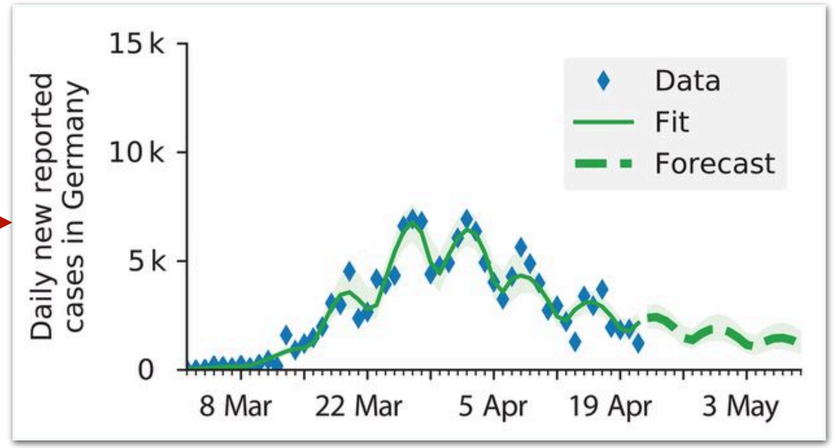
TransferLab

Central challenge: finding model parameters



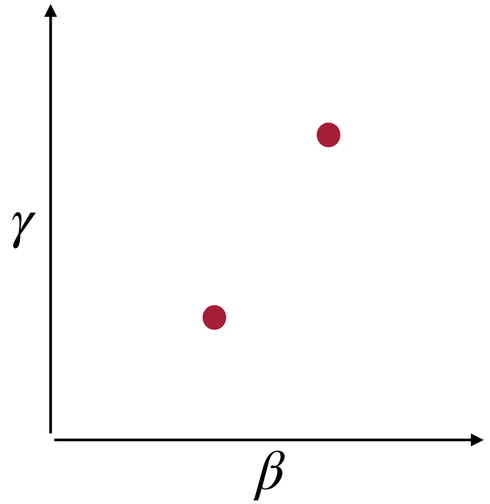


infection rate β
?
recovery rate γ



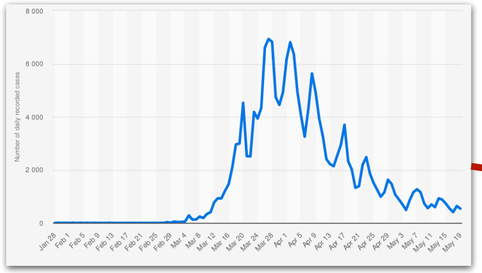
Naive approach

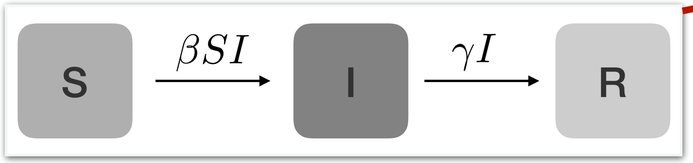
- find single best-fitting parameters
  - hand tuning, grid search
  - optimization, genetic algorithms



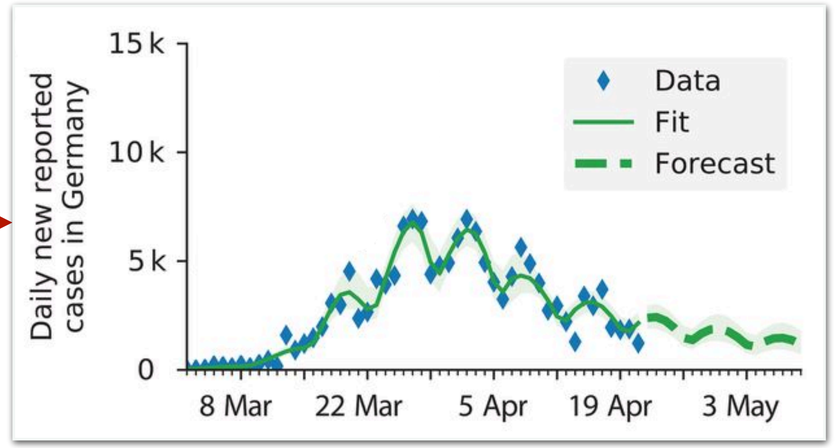
TransferLab

Central challenge: finding model parameters



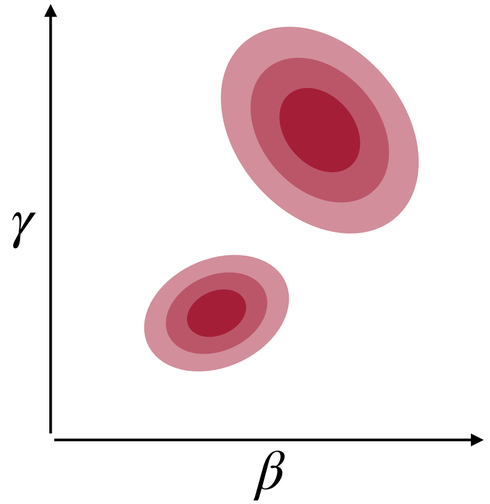


infection rate β
?
recovery rate γ



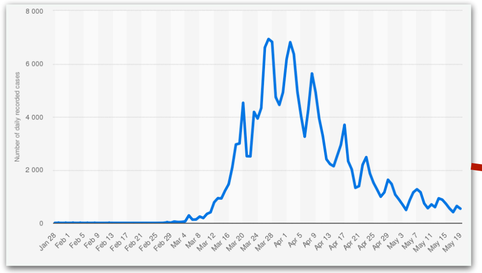
Naive approach

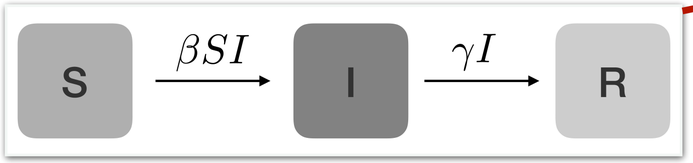
- find single best-fitting parameters
  - hand tuning, grid search
  - optimization, genetic algorithms



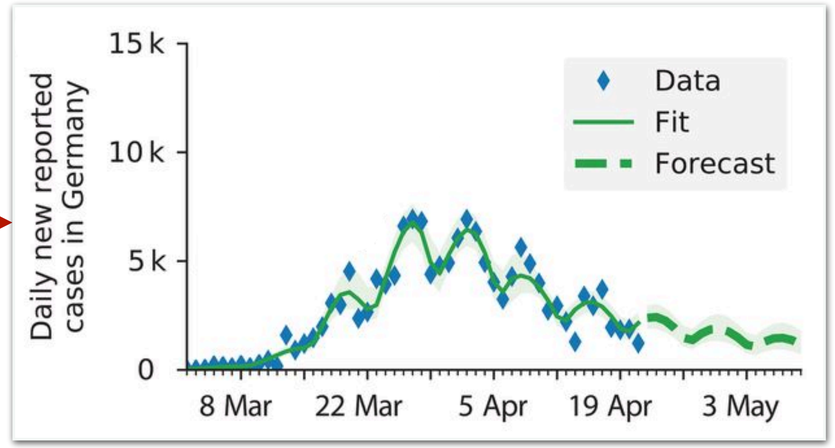
TransferLab

Central challenge: finding model parameters



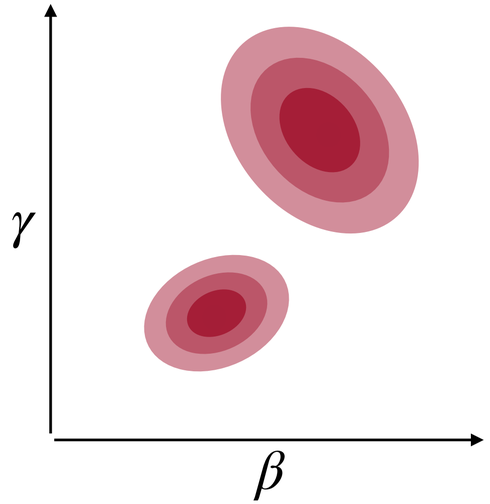


infection rate β
?
recovery rate γ



Naive approach

- find single best-fitting parameters
  - hand tuning, grid search
  - optimization, genetic algorithms



Ideal approach

- find all parameters
- quantify uncertainty
- quantify relations between parameters

TransferLab

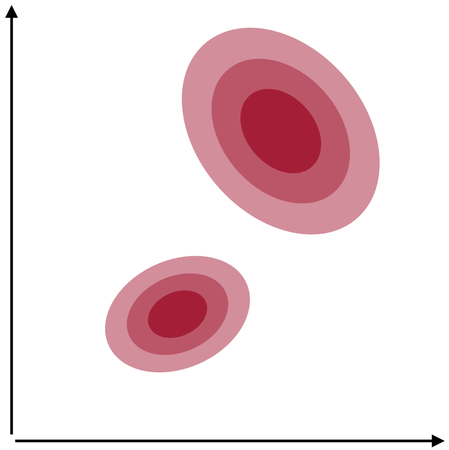
Outline

1. **Why** simulators?
2. **Why** Bayesian parameter inference?
3. **How** to do Bayesian inference for simulators?
4. **How** can you apply SBI to your simulator?

Bayesian parameter inference

Ideal approach

- find all parameters
- quantify uncertainty
- quantify relations between parameters

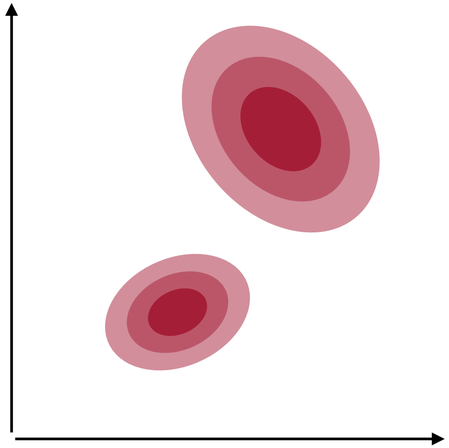


Laplace, Bayes, 1764

Bayesian parameter inference

Ideal approach

- find all parameters
- quantify uncertainty
- quantify relations between parameters



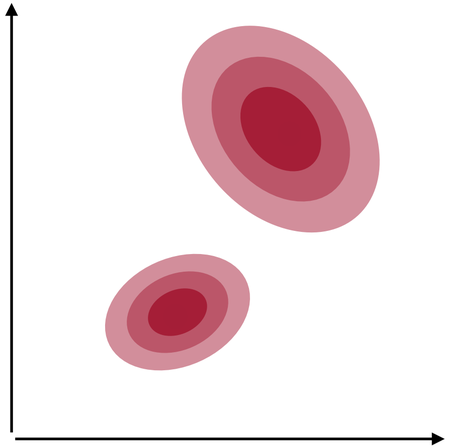
- data x and parameters θ are *random variables*

Laplace, Bayes, 1764

Bayesian parameter inference

Ideal approach

- find all parameters
- quantify uncertainty
- quantify relations between parameters



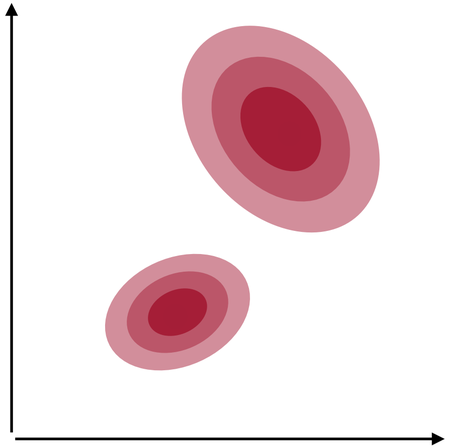
- data x and parameters θ are *random variables*
- **goal**: infer a distribution over parameters $p(\theta \mid x)$

Laplace, Bayes, 1764

Bayesian parameter inference

Ideal approach

- find all parameters
- quantify uncertainty
- quantify relations between parameters



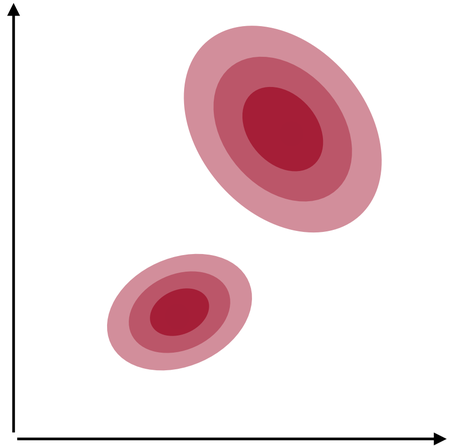
- data x and parameters θ are *random variables*
- **goal**: infer a distribution over parameters $p(\theta \mid x)$
  - characterizes **all** suitable parameters

Laplace, Bayes, 1764

Bayesian parameter inference

Ideal approach

- find all parameters
- quantify uncertainty
- quantify relations between parameters



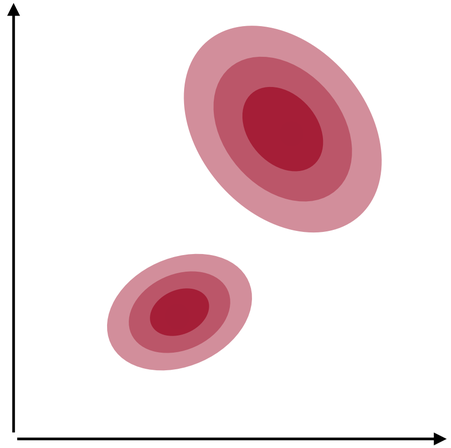
- data x and parameters θ are *random variables*
- **goal**: infer a distribution over parameters $p(\theta \mid x)$
  - characterizes **all** suitable parameters
  - quantifies uncertainty

Laplace, Bayes, 1764

Bayesian parameter inference

Ideal approach

- find all parameters
- quantify uncertainty
- quantify relations between parameters



- data x and parameters θ are *random variables*
- **goal**: infer a distribution over parameters $p(\theta \mid x)$
  - characterizes **all** suitable parameters
  - quantifies uncertainty

Bayes' Theorem

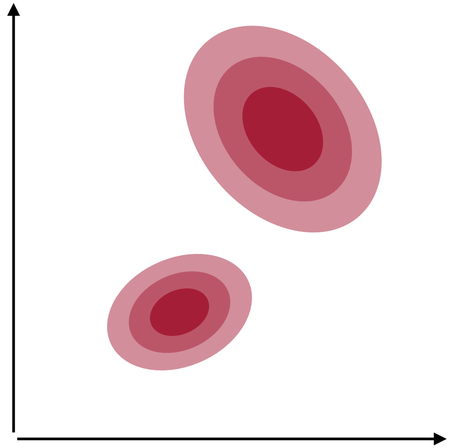
$$p(\theta \mid x) \propto p(x \mid \theta)\ p(\theta)$$

Laplace, Bayes, 1764

Bayesian parameter inference

Ideal approach

- find all parameters
- quantify uncertainty
- quantify relations between parameters



- data x and parameters θ are *random variables*
- **goal**: infer a distribution over parameters $p(\theta \mid x)$
  - characterizes **all** suitable parameters
  - quantifies uncertainty

Bayes' Theorem

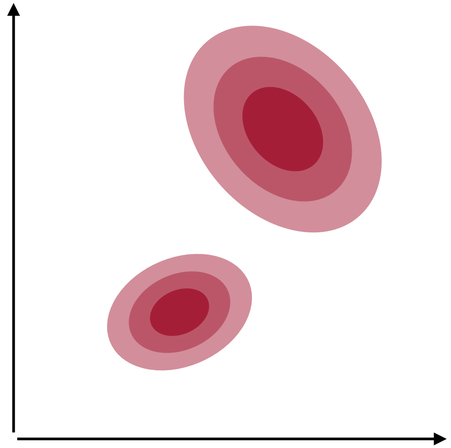
"posterior" → $p(\theta \mid x) \propto p(x \mid \theta)\ p(\theta)$

Laplace, Bayes, 1764

Bayesian parameter inference

Ideal approach

- find all parameters
- quantify uncertainty
- quantify relations between parameters



- data x and parameters θ are *random variables*
- **goal**: infer a distribution over parameters $p(\theta \mid x)$
  - characterizes **all** suitable parameters
  - quantifies uncertainty

Bayes' Theorem

"prior"

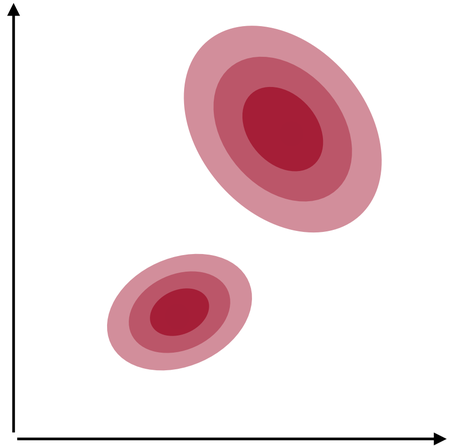
"posterior" → $p(\theta \mid x) \propto p(x \mid \theta)\ p(\theta)$

Laplace, Bayes, 1764

Bayesian parameter inference

Ideal approach

- find all parameters
- quantify uncertainty
- quantify relations between parameters



- data x and parameters θ are *random variables*
- **goal**: infer a distribution over parameters $p(\theta \mid x)$
  - characterizes **all** suitable parameters
  - quantifies uncertainty

Bayes' Theorem

"prior"

"posterior" $\rightarrow$ $p(\theta \mid x) \propto p(x \mid \theta)\ p(\theta)$

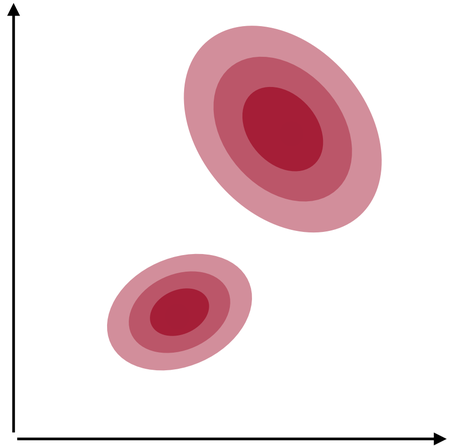
"likelihood"

Laplace, Bayes, 1764

Bayesian parameter inference

Ideal approach

- find all parameters
- quantify uncertainty
- quantify relations between parameters



- data x and parameters θ are *random variables*
- **goal**: infer a distribution over parameters $p(\theta \mid x)$
  - characterizes **all** suitable parameters
  - quantifies uncertainty

Bayes' Theorem

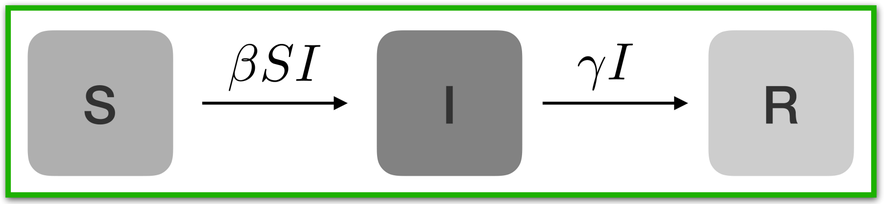
"prior"

"posterior" → $p(\theta \mid x) \propto p(x \mid \theta)\ p(\theta)$

"likelihood" = simulator

Laplace, Bayes, 1764

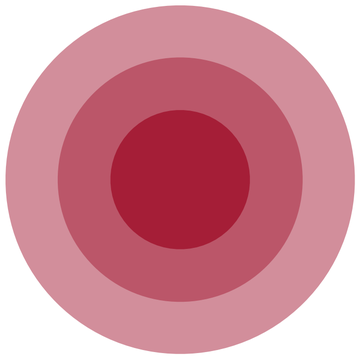
Bayesian parameter inference for simulators



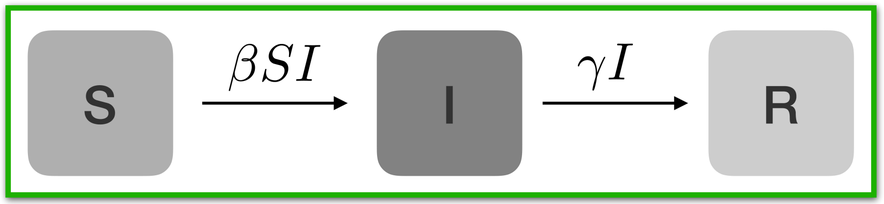
$$p(\theta \mid x) \propto p(x \mid \theta)\, p(\theta)$$

Bayesian parameter inference for simulators

parameters θ



$\theta \sim p(\theta)$



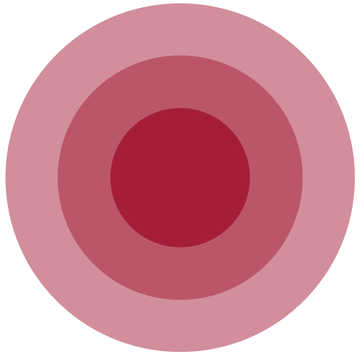
$$p(\theta \mid x) \propto p(x \mid \theta)\, p(\theta)$$

TransferLab

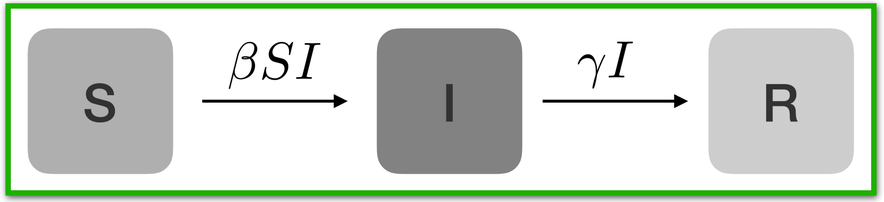
Bayesian parameter inference for simulators

simulation (forward model)

parameters θ

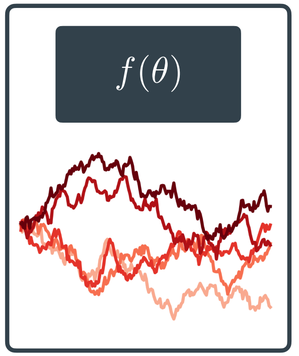


$\theta \sim p(\theta)$



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

data x

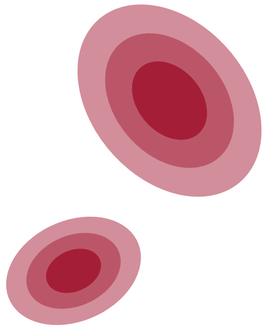


$x \sim p(x \mid \theta)$

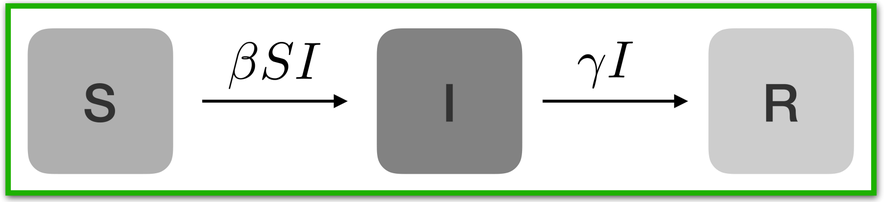
Bayesian parameter inference for simulators

simulation (forward model)

parameters θ

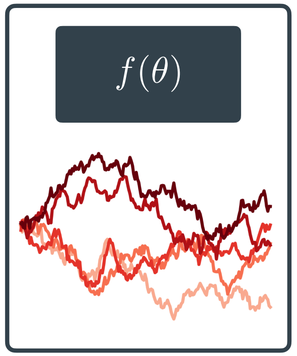


$p(\theta \mid x)$



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

data x

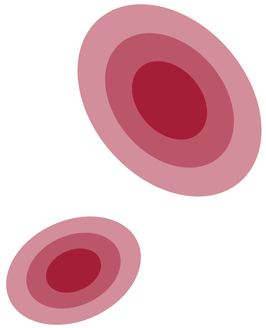


$x \sim p(x \mid \theta)$

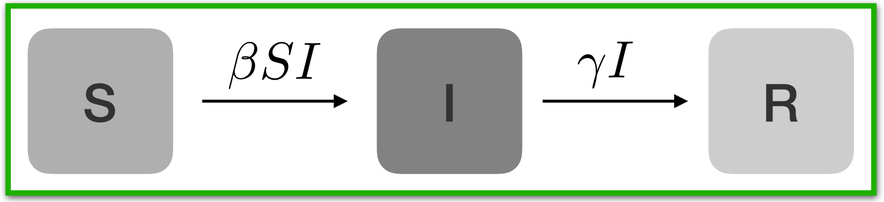
inference (inverse model)

Simulation-based Bayesian inference (SBI)

parameters θ

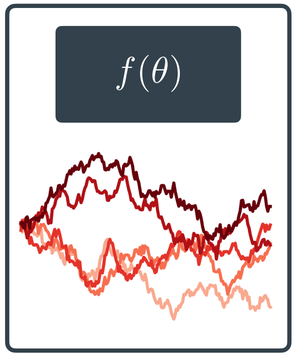


$p(\theta \mid x)$



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

data x

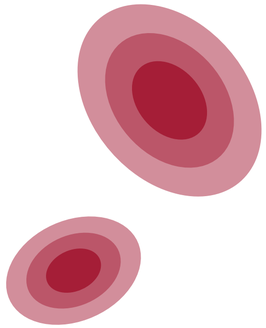


$x \sim p(x \mid \theta)$

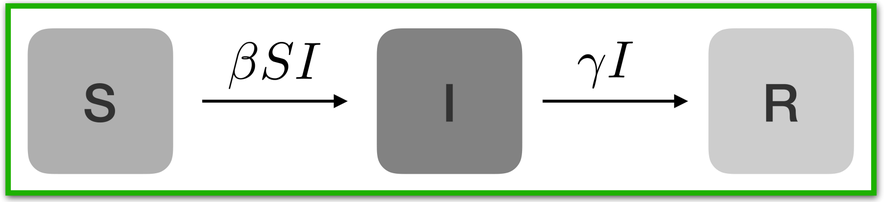
TransferLab

Simulation-based Bayesian inference (SBI)

parameters θ

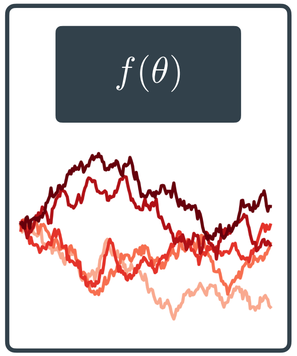


$p(\theta \mid x)$



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

data x

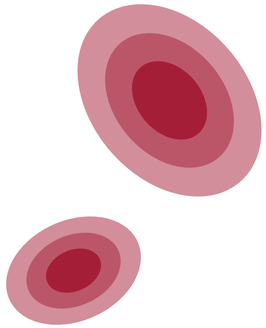


$x \sim p(x \mid \theta)$

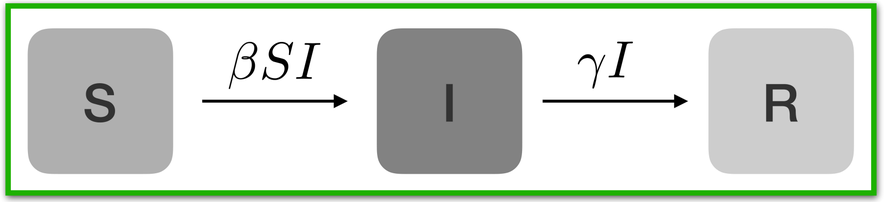
TransferLab

Simulation-based Bayesian inference (SBI)

parameters θ

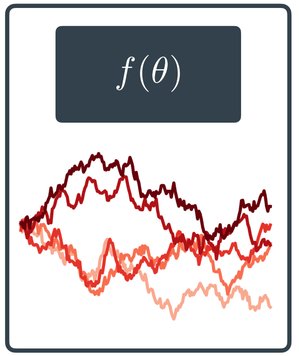


$p(\theta \mid x)$



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

data x



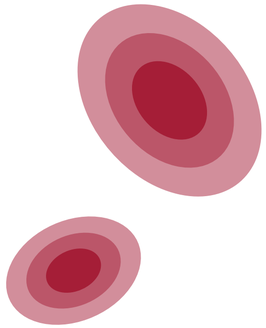
$x \sim p(x \mid \theta)$

TransferLab

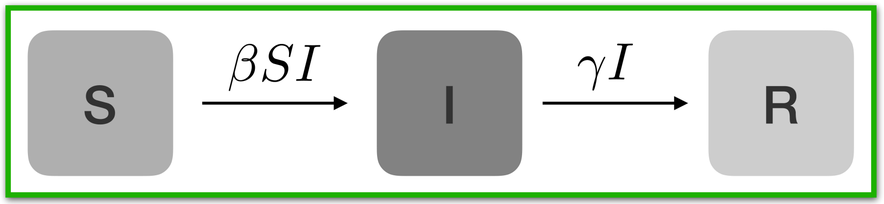
Simulation-based Bayesian inference (SBI)

parameters θ

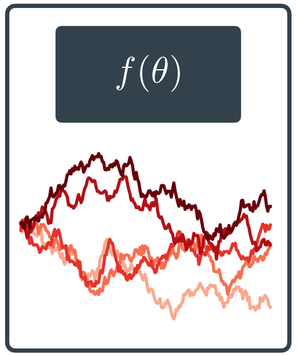
data x



$p(\theta \mid x)$



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



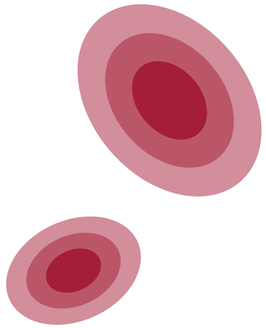
$x \sim p(x \mid \theta)$

No Markov Chain Monte Carlo (MCMC)
No variational inference
No Bayesian optimization

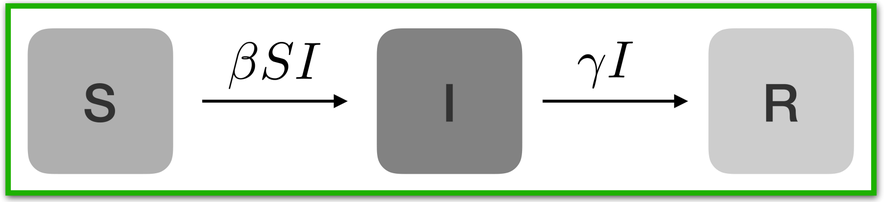
TransferLab

Simulation-based Bayesian inference (SBI)

parameters θ



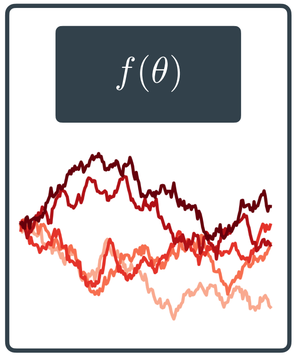
$p(\theta \mid x)$



$$\boxed{p(\theta \mid x)} \propto \boxed{p(x \mid \theta)} p(\theta)$$

use machine learning to perform inference

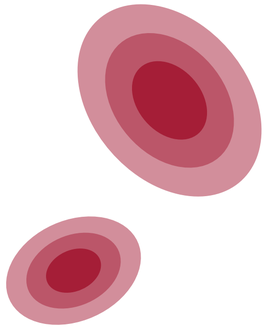
data x



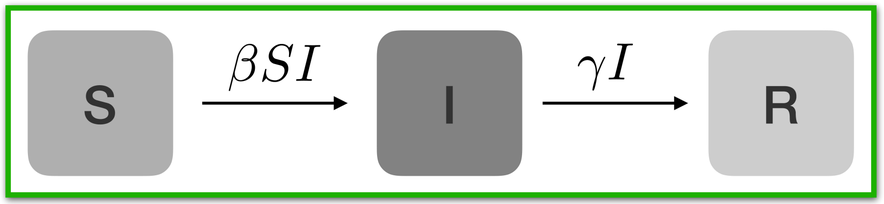
$x \sim p(x \mid \theta)$

Simulation-based Bayesian inference (SBI)

parameters θ

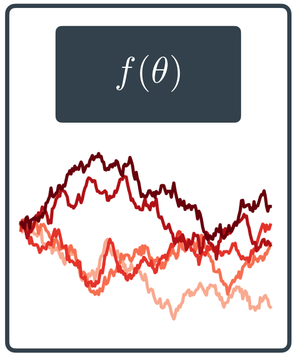


$p(\theta \mid x)$



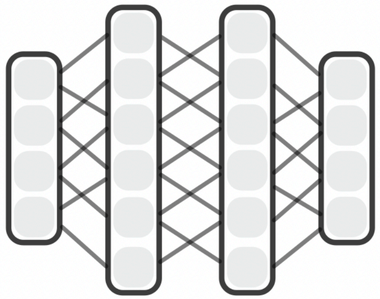
$$p(\theta \mid x) \propto p(x \mid \theta)\, p(\theta)$$

data x



$x \sim p(x \mid \theta)$

use machine learning to perform inference



training data

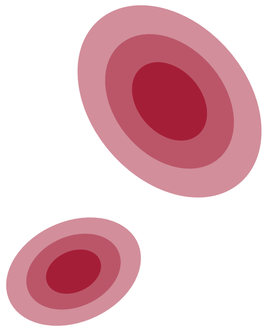
$\{(\theta_1, x_1), (\theta_2, x_2), \ldots\}$

TransferLab

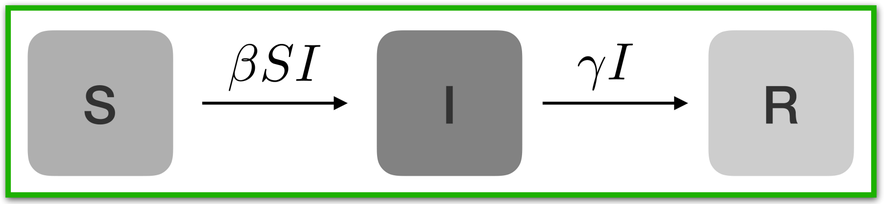
Simulation-based Bayesian inference (SBI)

simulation (forward model)

parameters θ

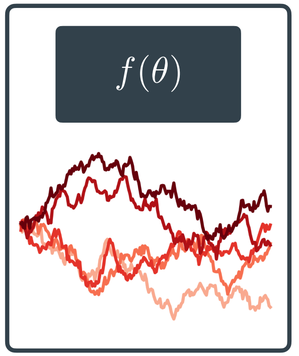


$p(\theta \mid x)$



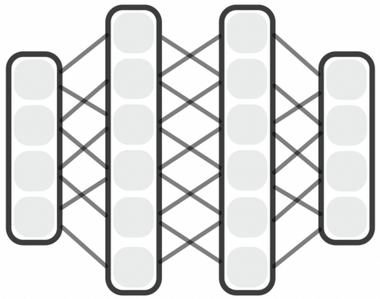
$$p(\theta \mid x) \propto p(x \mid \theta)\, p(\theta)$$

data x



$x \sim p(x \mid \theta)$

use machine learning to perform inference



training data

$\{(\theta_1, x_1), (\theta_2, x_2), \ldots\}$

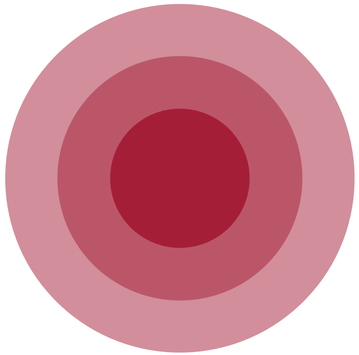
TransferLab

Outline

1. **Why** simulators?
2. **Why** Bayesian parameter inference?
3. **How** to do Bayesian inference for simulators?
4. **How** can you apply SBI to your simulator?

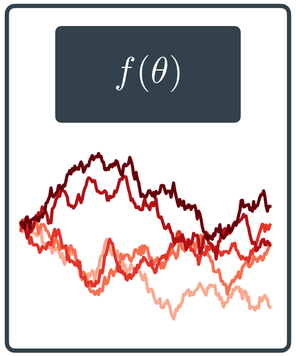
Neural Simulation-Based Inference

prior



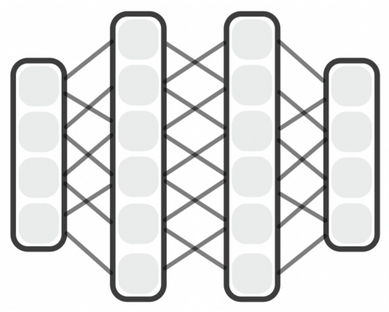
$\theta_i \sim p(\theta)$

simulated data



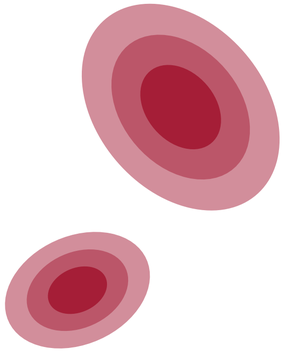
$x_i \sim p(x|\theta_i)$

density estimator



$\underset{\phi}{\operatorname{argmin}}\ \mathcal{L}(\phi)$

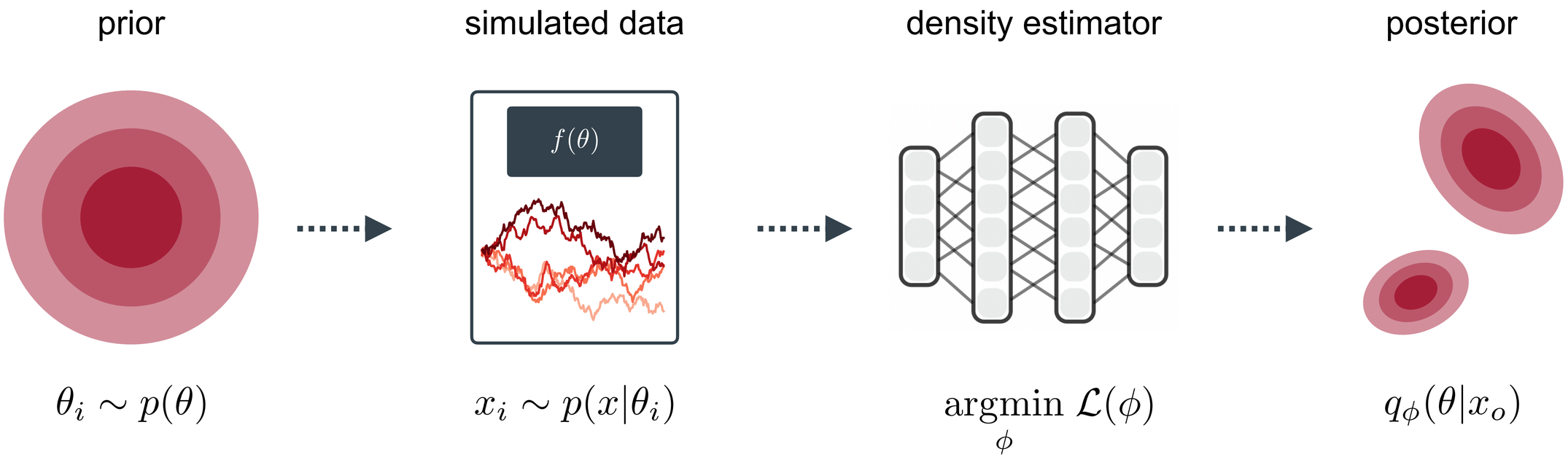
posterior



$q_\phi(\theta|x_o)$

Wood, 2010; Blum et al. 2010; Papamakarios et al. 2016, 2018

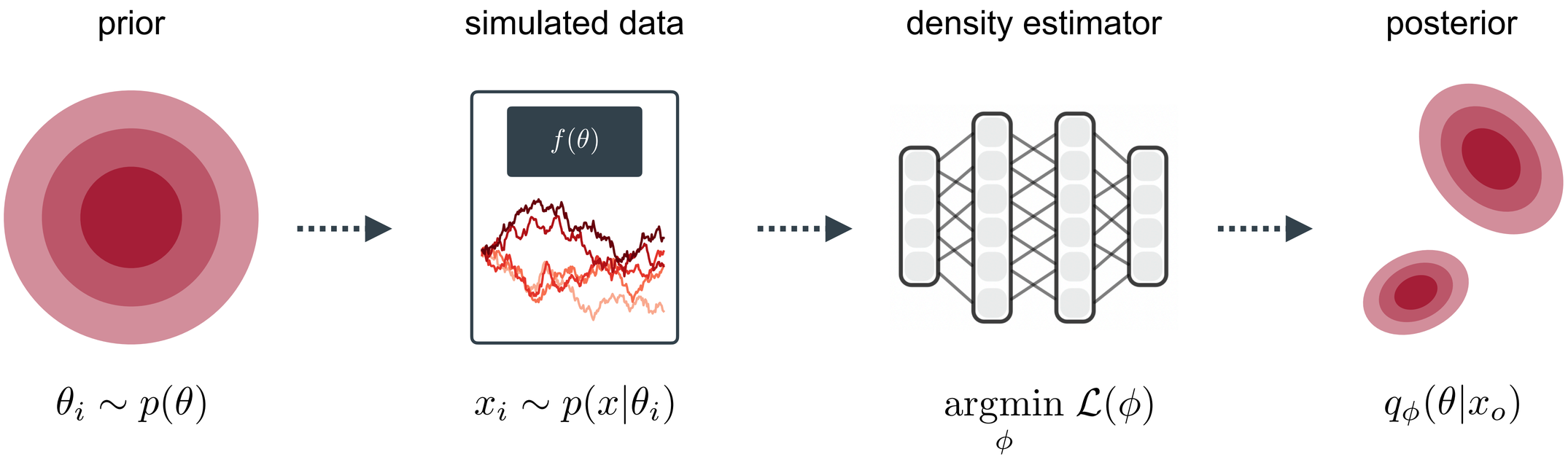
Neural Simulation-Based Inference



1. **Training**: train neural network (NN) with simulated data to learn $q_\phi(\theta \,|\, x)$

Wood, 2010; Blum et al. 2010; Papamakarios et al. 2016, 2018

Neural Simulation-Based Inference



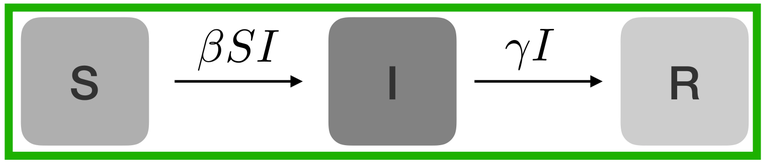
1. **Training**: train neural network (NN) with simulated data to learn $q_\phi(\theta \,|\, x)$

2. **Inference**: plug x_o into NN to obtain $q_\phi(\theta \,|\, x_o)$

Wood, 2010; Blum et al. 2010; Papamakarios et al. 2016, 2018

Neural Simulation-Based Inference

Simulator



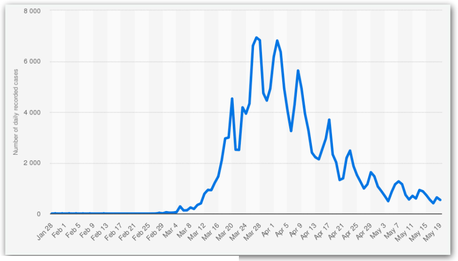
Prior

$$p(\theta) = p(\beta, \gamma)$$

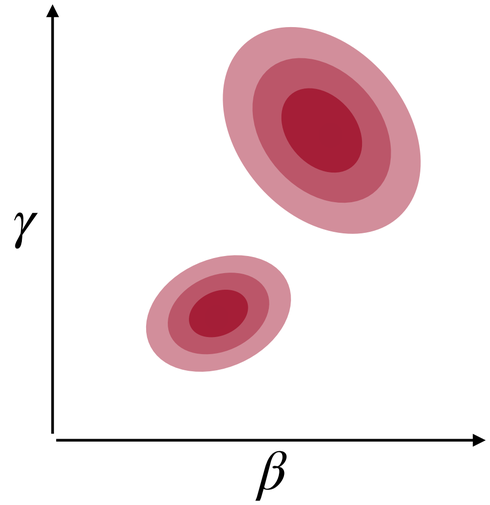
Simulated data

$$\theta_i \sim p(\theta)$$
$$x_i = \textbf{SIR}(\theta_i)$$

Observed data x_o



Conditional Density Estimation

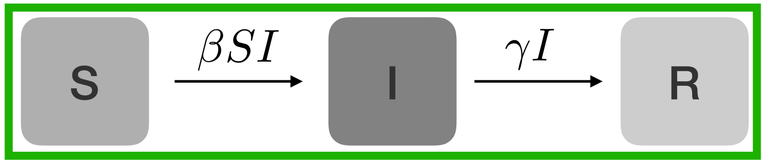


TransferLab

Wood, 2010; Blum et al. 2010; Papamakarios et al. 2016, 2018

Neural Simulation-Based Inference

Simulator



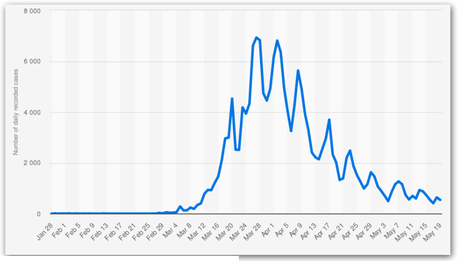
Prior

$$p(\theta) = p(\beta, \gamma)$$

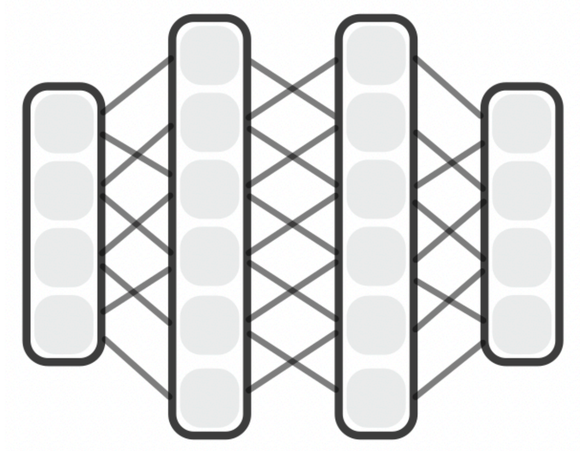
Simulated data

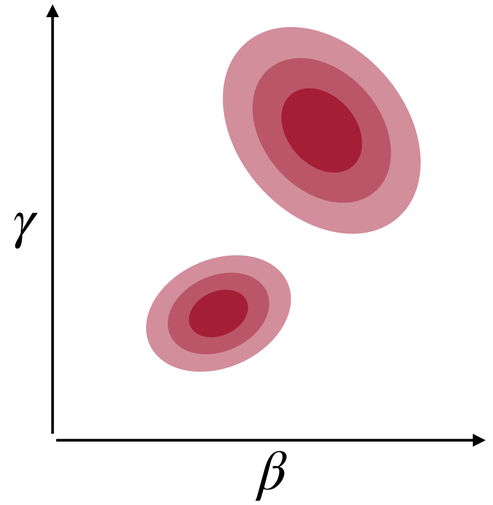
$$\theta_i \sim p(\theta)$$
$$x_i = \textbf{SIR}(\theta_i)$$

Observed data x_o



Conditional Density Estimation



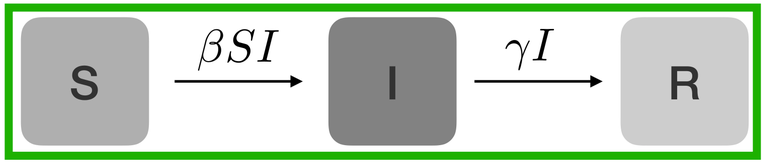


TransferLab

Wood, 2010; Blum et al. 2010; Papamakarios et al. 2016, 2018

Neural Simulation-Based Inference

Simulator



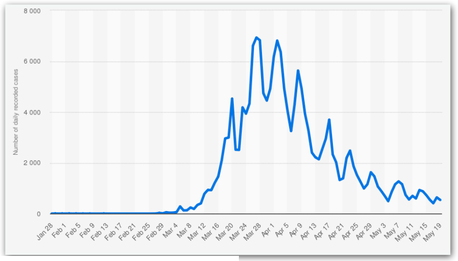
Prior

$$p(\theta) = p(\beta, \gamma)$$

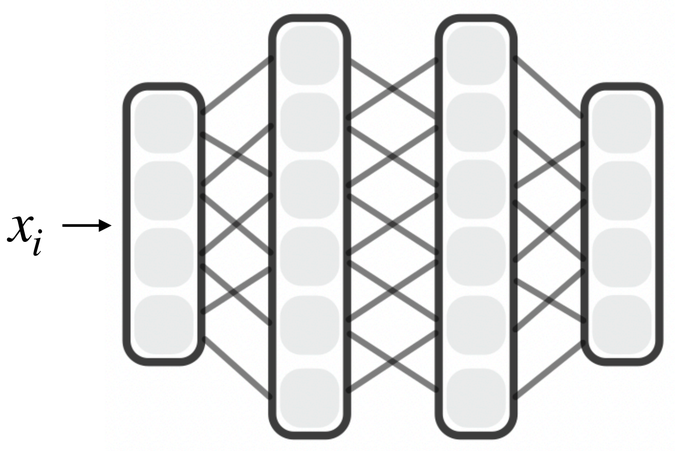
Simulated data

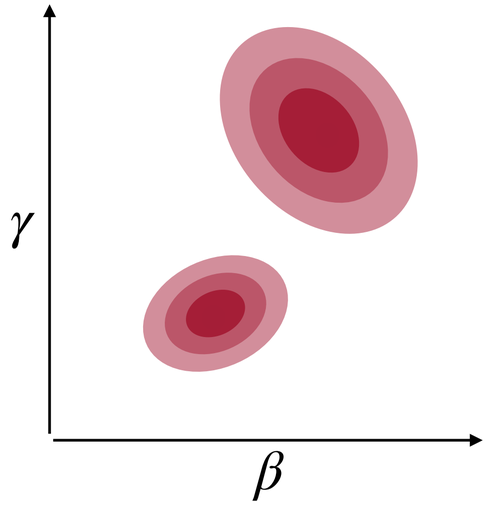
$$\theta_i \sim p(\theta)$$
$$x_i = \textbf{SIR}(\theta_i)$$

Observed data x_o



Conditional Density Estimation



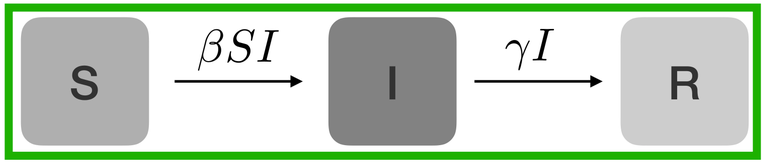


TransferLab

Wood, 2010; Blum et al. 2010; Papamakarios et al. 2016, 2018

Neural Simulation-Based Inference

Simulator



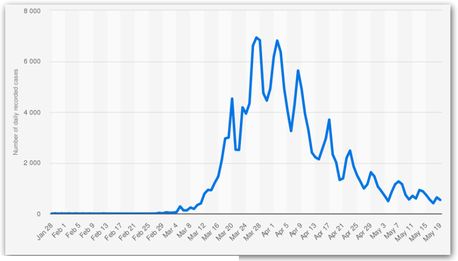
Prior

$$p(\theta) = p(\beta, \gamma)$$

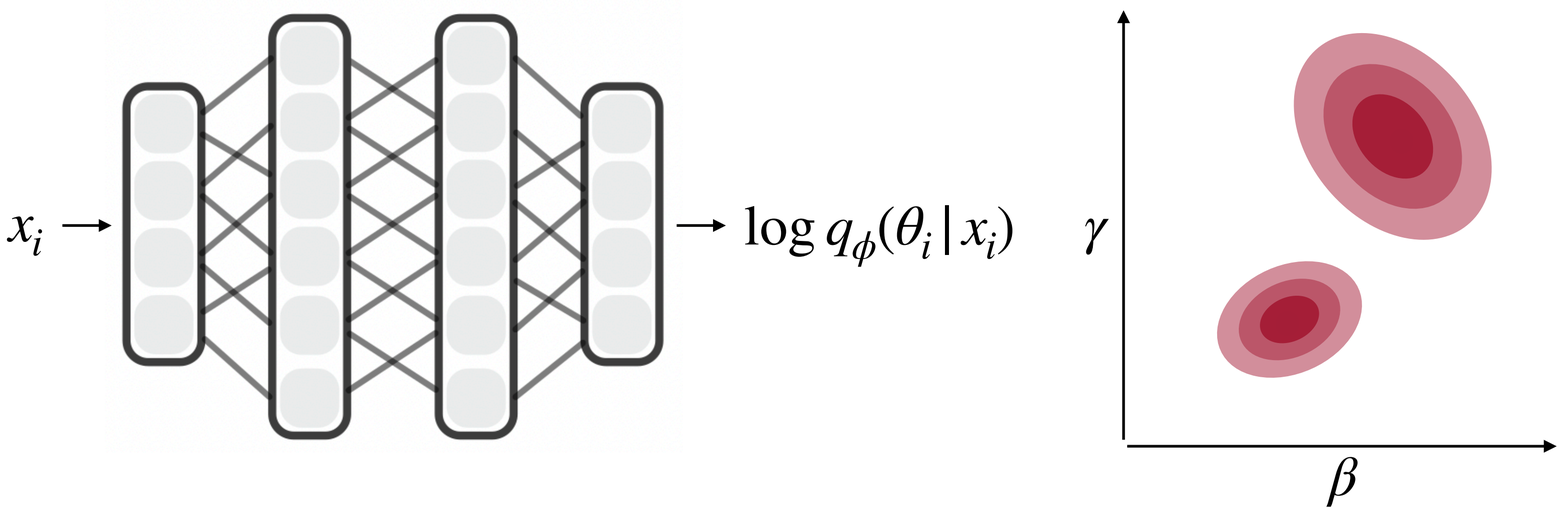
Simulated data

$$\theta_i \sim p(\theta)$$
$$x_i = \textbf{SIR}(\theta_i)$$

Observed data x_o



Conditional Density Estimation

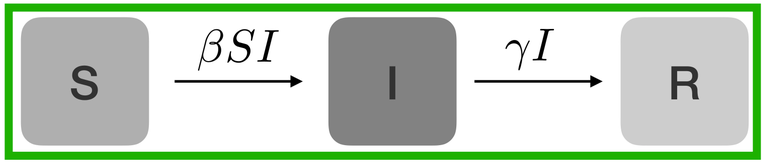


TransferLab

Wood, 2010; Blum et al. 2010; Papamakarios et al. 2016, 2018

Neural Simulation-Based Inference

Simulator



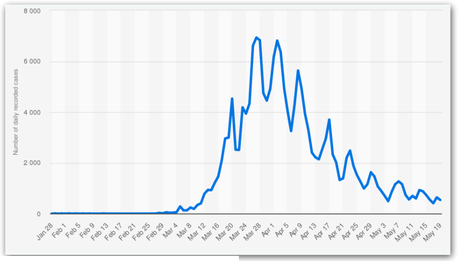
Prior

$$p(\theta) = p(\beta, \gamma)$$

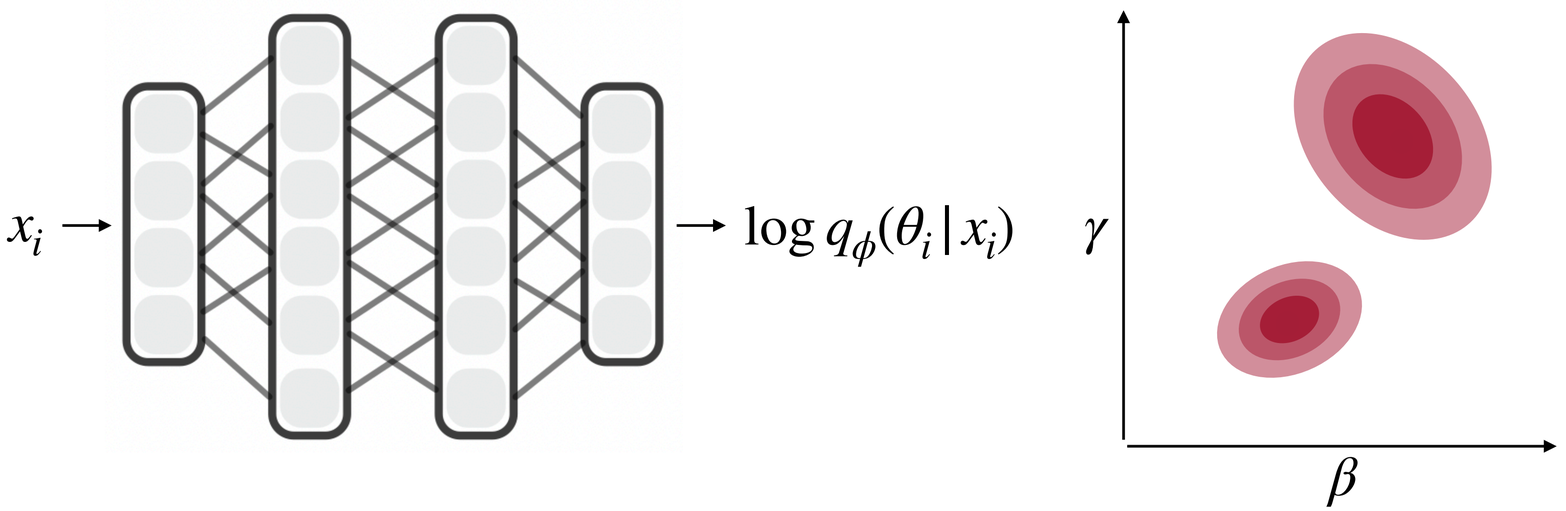
Simulated data

$$\theta_i \sim p(\theta)$$
$$x_i = \textbf{SIR}(\theta_i)$$

Observed data x_o



Conditional Density Estimation

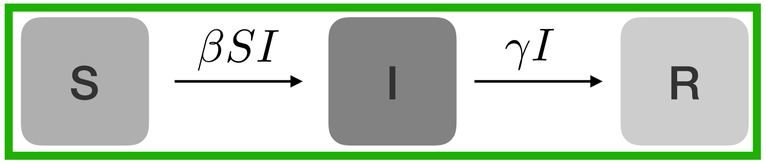


1. Train with simulated data

Wood, 2010; Blum et al. 2010; Papamakarios et al. 2016, 2018

Neural Simulation-Based Inference

Simulator



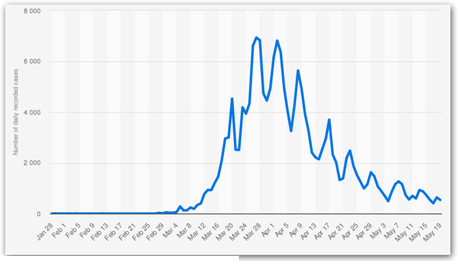
Prior

$$p(\theta) = p(\beta, \gamma)$$

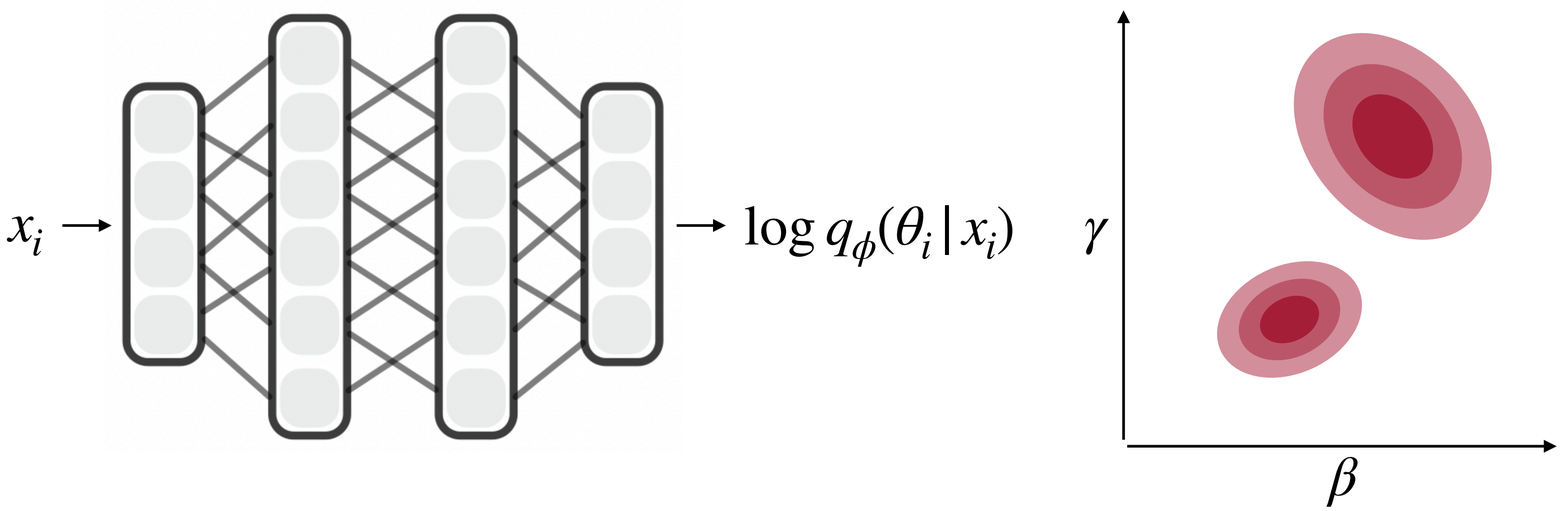
Simulated data

$$\theta_i \sim p(\theta)$$
$$x_i = \textbf{SIR}(\theta_i)$$

Observed data x_o



Conditional Density Estimation



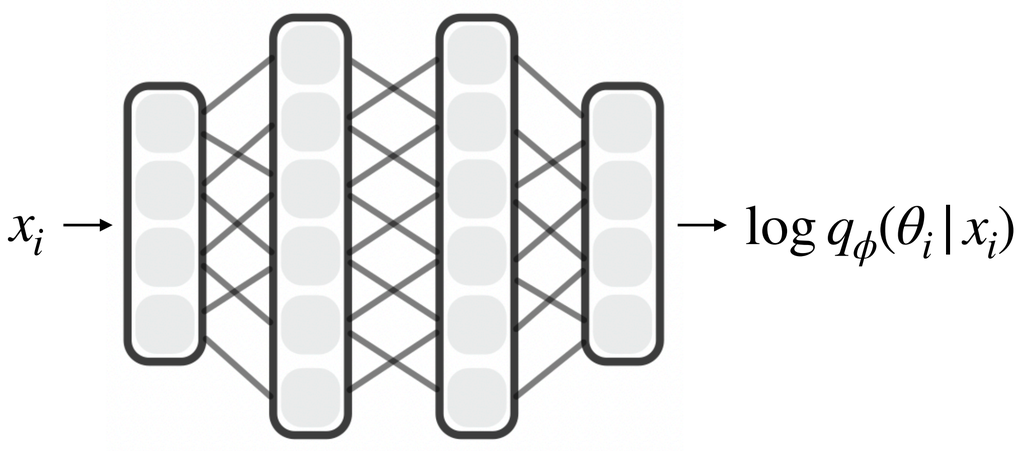
1. Train with simulated data
2. Infer given observed data

TransferLab

Wood, 2010; Blum et al. 2010; Papamakarios et al. 2016, 2018

Conditional density estimation in a nutshell

Conditional Density Estimation

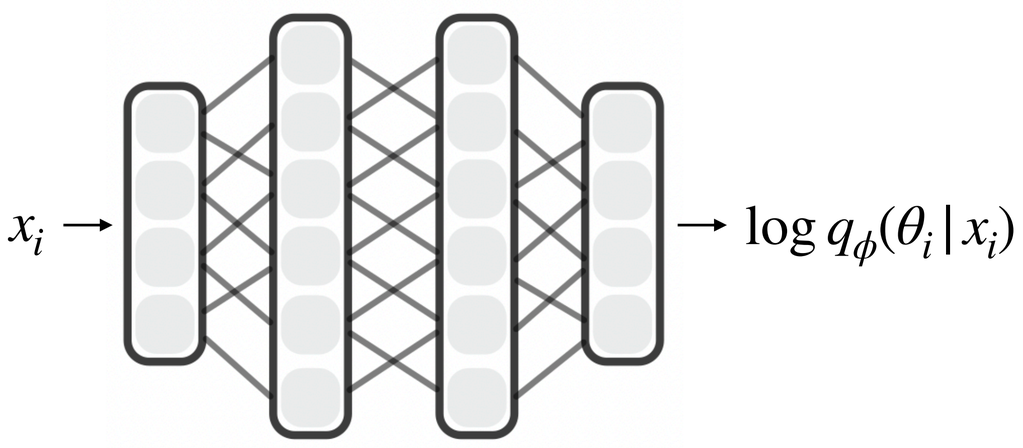


Papamakarios et al. 2016, Greenberg et al. 2019

Conditional density estimation in a nutshell

Conditional Density Estimation

- Assume a parametric model for the posterior, e.g., a Gaussian

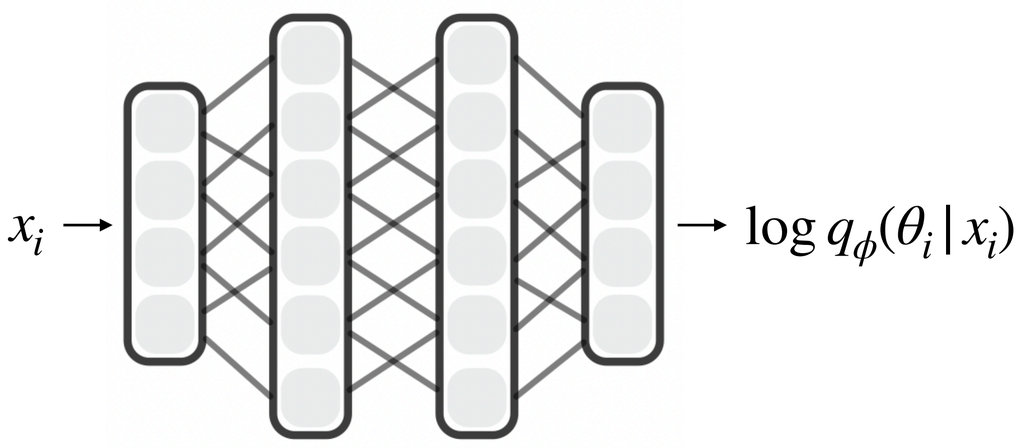


Papamakarios et al. 2016, Greenberg et al. 2019

Conditional density estimation in a nutshell

Conditional Density Estimation

- Assume a parametric model for the posterior, e.g., a Gaussian
- NN predicts Gaussian mean and variance

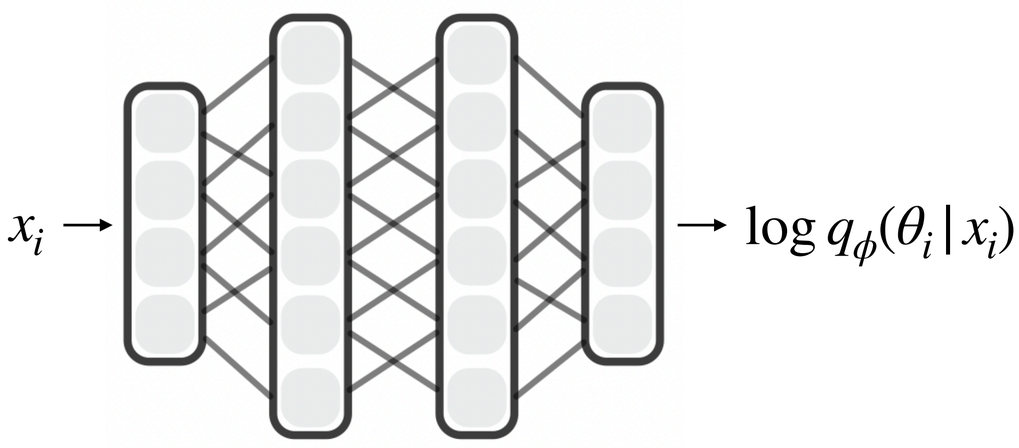


Papamakarios et al. 2016, Greenberg et al. 2019

Conditional density estimation in a nutshell

Conditional Density Estimation

- Assume a parametric model for the posterior, e.g., a Gaussian
- NN predicts Gaussian mean and variance
- Loss signal: **model probability**

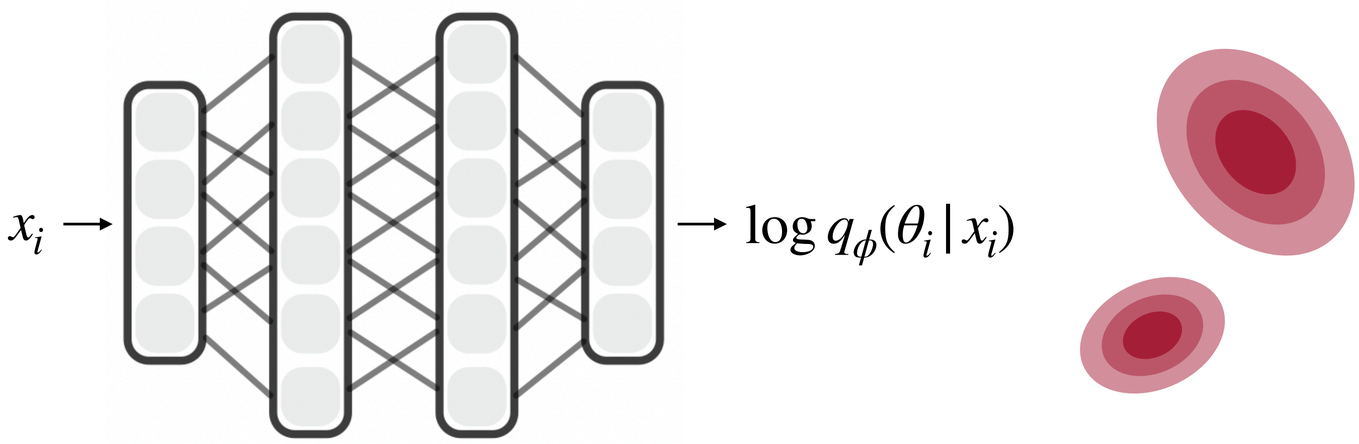


Papamakarios et al. 2016, Greenberg et al. 2019

Conditional density estimation in a nutshell

Conditional Density Estimation

- Assume a parametric model for the posterior, e.g., a Gaussian
- NN predicts Gaussian mean and variance
- Loss signal: **model probability**

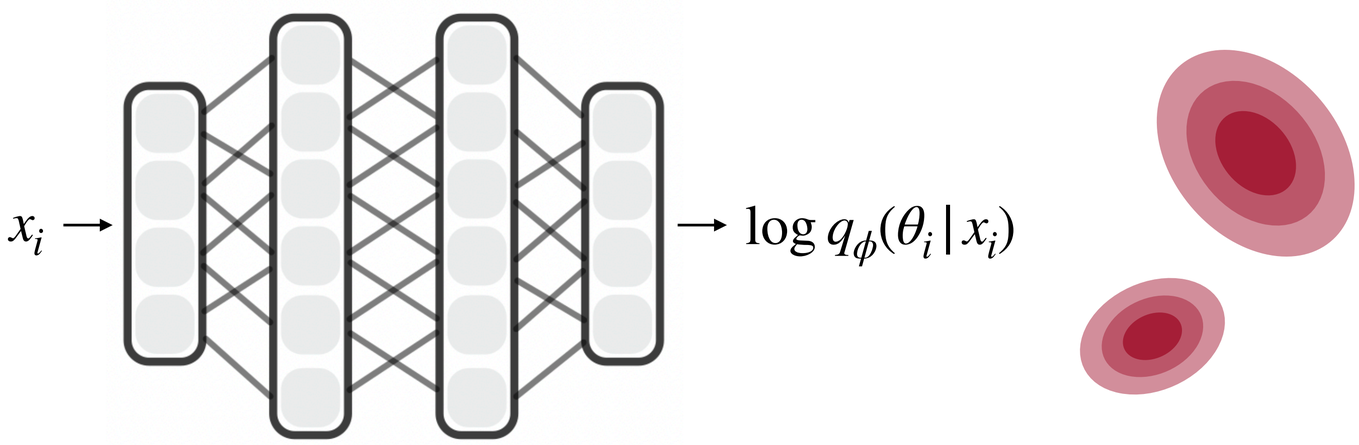


Papamakarios et al. 2016, Greenberg et al. 2019

Conditional density estimation in a nutshell

Conditional Density Estimation

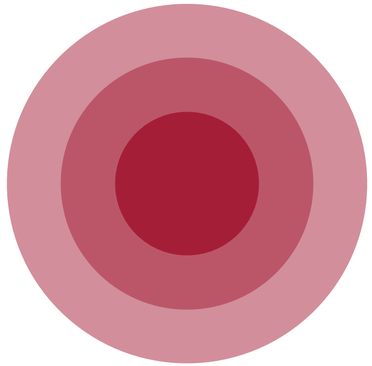
- Assume a parametric model for the posterior, e.g., a Gaussian
- NN predicts Gaussian mean and variance
- Loss signal: **model probability**
- Used in practice:
  - Mixture of Gaussians
  - Normalizing Flows
  - (Flow Matching, Score Matching)



Papamakarios et al. 2016, Greenberg et al. 2019

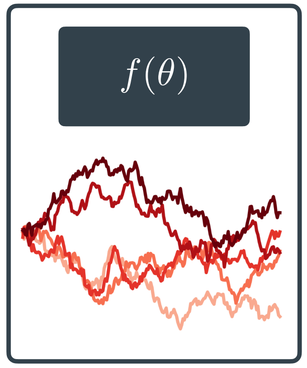
Three main neural SBI approaches

prior



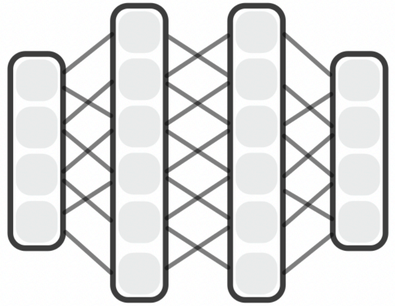
$\theta_i \sim p(\theta)$

simulated data



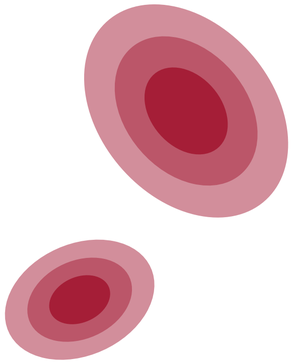
$x_i \sim p(x|\theta_i)$

density estimator



$\underset{\phi}{\text{argmin}}\ \mathcal{L}(\phi)$

posterior

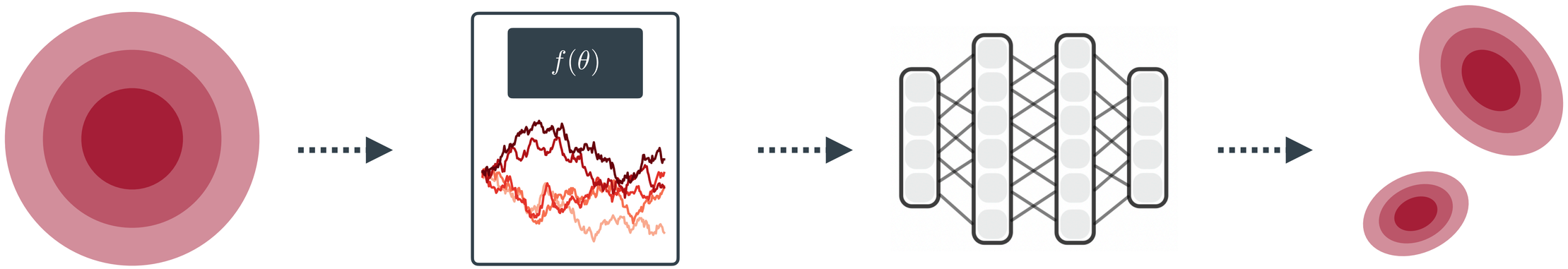


$q_\phi(\theta|x_o)$

Papamakarios et al. 2016, 2018; Hermans et al. 2020

Three main neural SBI approaches

prior | simulated data | density estimator | posterior



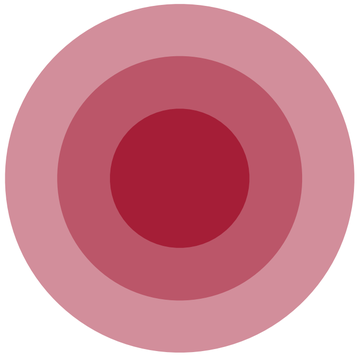
$\theta_i \sim p(\theta)$ $\quad x_i \sim p(x|\theta_i)$ $\quad \underset{\phi}{\text{argmin}}\ \mathcal{L}(\phi)$ $\quad q_\phi(\theta|x_o)$

1. **Neural Posterior Estimation** (NPE): directly learn the posterior $q_\phi(\theta \,|\, x)$

Papamakarios et al. 2016, 2018; Hermans et al. 2020

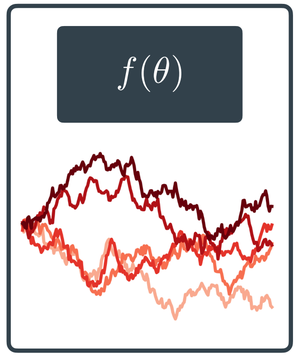
Three main neural SBI approaches

prior



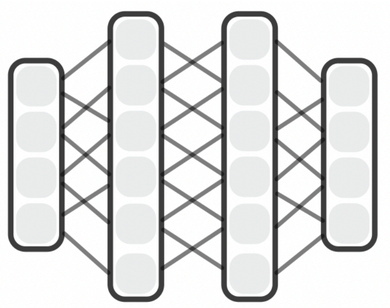
$\theta_i \sim p(\theta)$

simulated data



$\{\theta_i, x_i\}_{i=1}^{N}$

density estimator



$\underset{\phi}{\operatorname{argmin}}\, \mathcal{L}(\phi)$

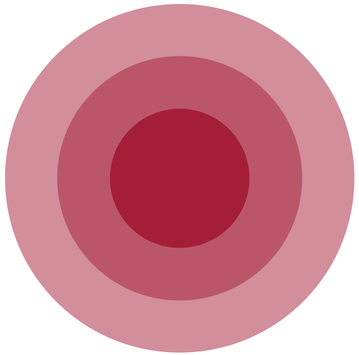
1. **Neural Posterior Estimation** (NPE): directly learn the posterior $q_\phi(\theta \,|\, x)$

2. **Neural Likelihood Estimation** (NLE): learn the likelihood $q_\phi(x \,|\, \theta)$, run MCMC to sample from $p(\theta \,|\, x_o)$

Papamakarios et al. 2016, 2018; Hermans et al. 2020

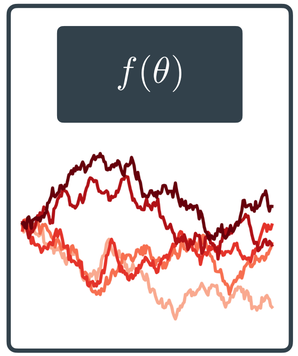
Three main neural SBI approaches

prior



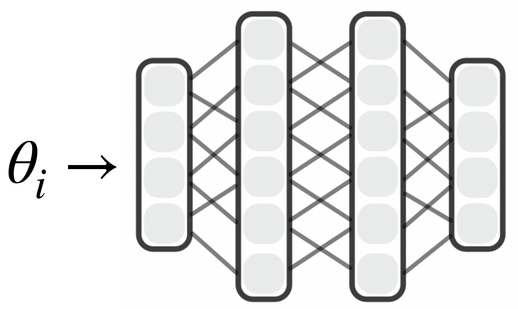
$\theta_i \sim p(\theta)$

simulated data



$\{\theta_i, x_i\}_{i=1}^{N}$

density estimator



$\underset{\phi}{\operatorname{argmin}}\, \mathcal{L}(\phi)$

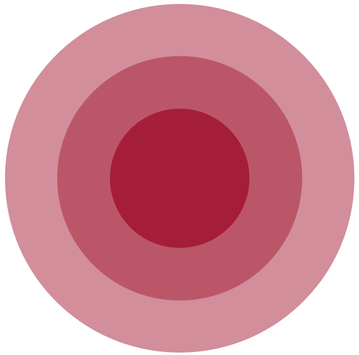
1. **Neural Posterior Estimation** (NPE): directly learn the posterior $q_\phi(\theta \mid x)$

2. **Neural Likelihood Estimation** (NLE): learn the likelihood $q_\phi(x \mid \theta)$, run MCMC to sample from $p(\theta \mid x_o)$

Papamakarios et al. 2016, 2018; Hermans et al. 2020

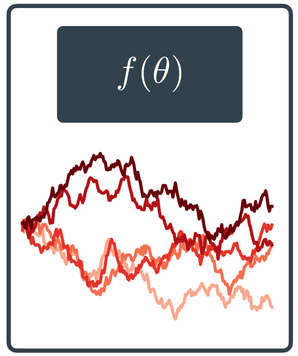
Three main neural SBI approaches

prior



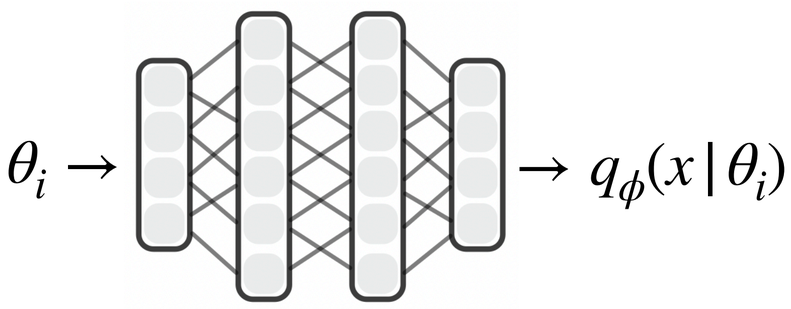
$\theta_i \sim p(\theta)$

simulated data



$\{\theta_i, x_i\}_{i=1}^{N}$

density estimator



$\underset{\phi}{\text{argmin}}\, \mathcal{L}(\phi)$

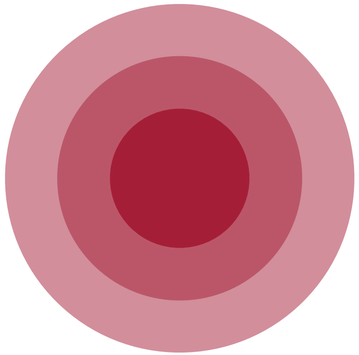
1. **Neural Posterior Estimation** (NPE): directly learn the posterior $q_\phi(\theta \mid x)$

2. **Neural Likelihood Estimation** (NLE): learn the likelihood $q_\phi(x \mid \theta)$, run MCMC to sample from $p(\theta \mid x_o)$

Papamakarios et al. 2016, 2018; Hermans et al. 2020

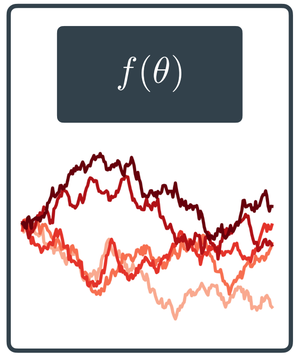
Three main neural SBI approaches

prior



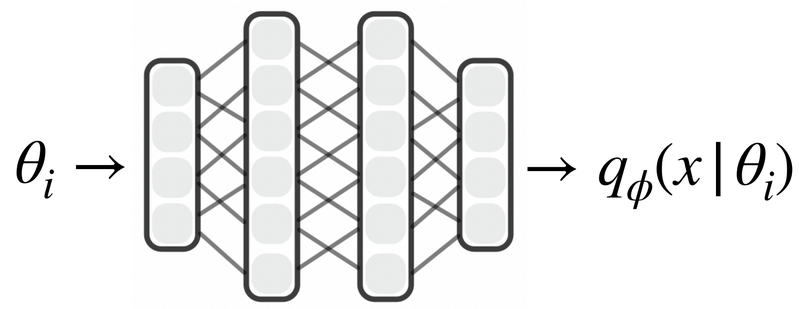
$\theta_i \sim p(\theta)$

simulated data



$\{\theta_i, x_i\}_{i=1}^{N}$

density estimator



$\underset{\phi}{\operatorname{argmin}} \mathcal{L}(\phi)$

$p(\theta \mid x_o) \propto q(x_o \mid \theta)\, p(\theta)$

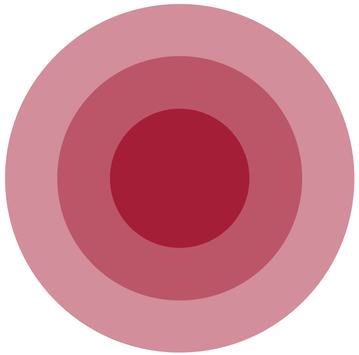
1. **Neural Posterior Estimation** (NPE): directly learn the posterior $q_\phi(\theta \mid x)$

2. **Neural Likelihood Estimation** (NLE): learn the likelihood $q_\phi(x \mid \theta)$, run MCMC to sample from $p(\theta \mid x_o)$

Papamakarios et al. 2016, 2018; Hermans et al. 2020

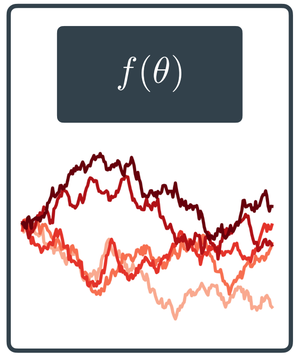
Three main neural SBI approaches

prior



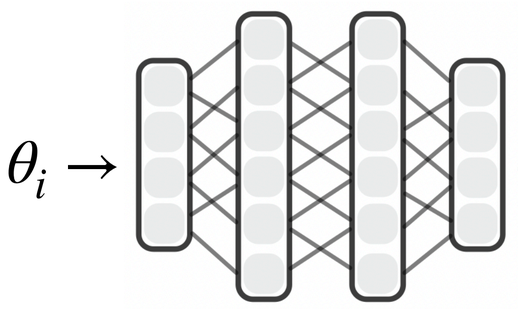
$\theta_i \sim p(\theta)$

simulated data



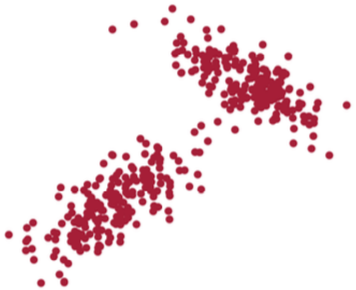
$\{\theta_i, x_i\}_{i=1}^{N}$

density estimator



$\underset{\phi}{\operatorname{argmin}} \mathcal{L}(\phi)$

posterior



$p(\theta \mid x_o) \propto q(x_o \mid \theta)\, p(\theta)$

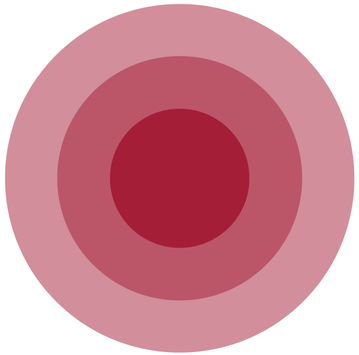
1. **Neural Posterior Estimation** (NPE): directly learn the posterior $q_\phi(\theta \mid x)$

2. **Neural Likelihood Estimation** (NLE): learn the likelihood $q_\phi(x \mid \theta)$, run MCMC to sample from $p(\theta \mid x_o)$

Papamakarios et al. 2016, 2018; Hermans et al. 2020

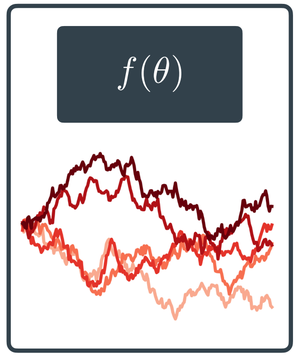
Three main neural SBI approaches

prior



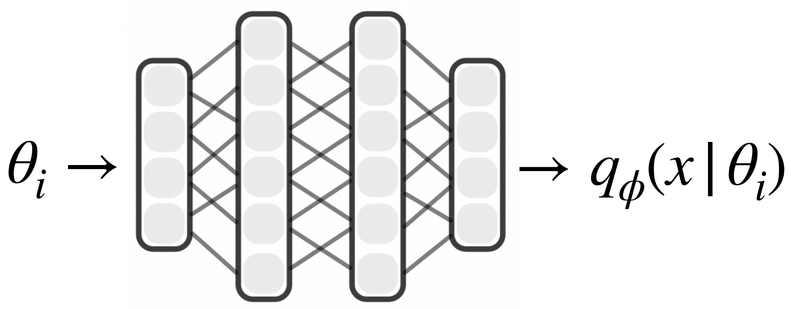
$\theta_i \sim p(\theta)$

simulated data



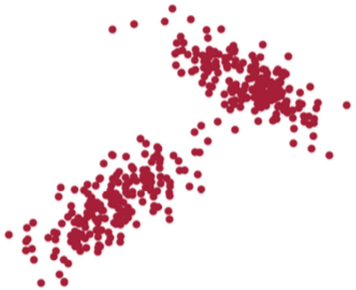
$\{\theta_i, x_i\}_{i=1}^{N}$

density estimator



$\underset{\phi}{\text{argmin}}\ \mathcal{L}(\phi)$

posterior



$p(\theta \mid x_o) \propto q(x_o \mid \theta)\, p(\theta)$

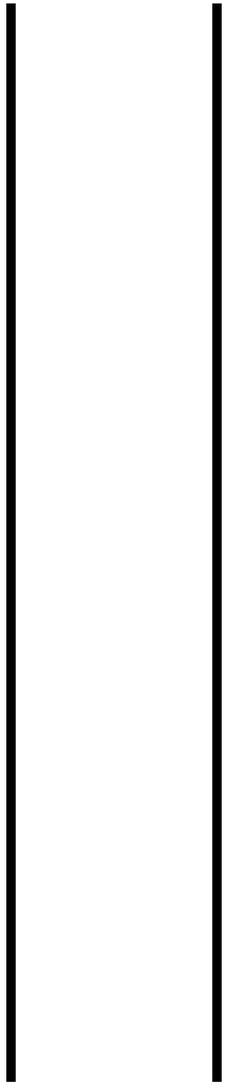
1. **Neural Posterior Estimation** (NPE): directly learn the posterior $q_\phi(\theta \mid x)$
2. **Neural Likelihood Estimation** (NLE): learn the likelihood $q_\phi(x \mid \theta)$, run MCMC to sample from $p(\theta \mid x_o)$
3. **Neural Ratio Estimation** (NRE): learn the likelihood ratio $r(x \mid \theta)$, run MCMC to sample from $p(\theta \mid x_o)$

Papamakarios et al. 2016, 2018; Hermans et al. 2020

Outline

1. **Why** simulators?
2. **Why** Bayesian parameter inference?
3. **How** to do Bayesian inference for simulators?
4. **How** can you apply SBI to your simulator?

Mind the Gap: Methods and Applicability of SBI



Tejero-Cantero, Boelts*, Deistler*, Lueckmann*, Durkan* et al. 2020*

Mind the Gap: Methods and Applicability of SBI

NPE, NLE, NRE

SNLE, SNPE, SNRE

diffusion models

flow matching

Tejero-Cantero, Boelts*, Deistler*, Lueckmann*, Durkan* et al. 2020*

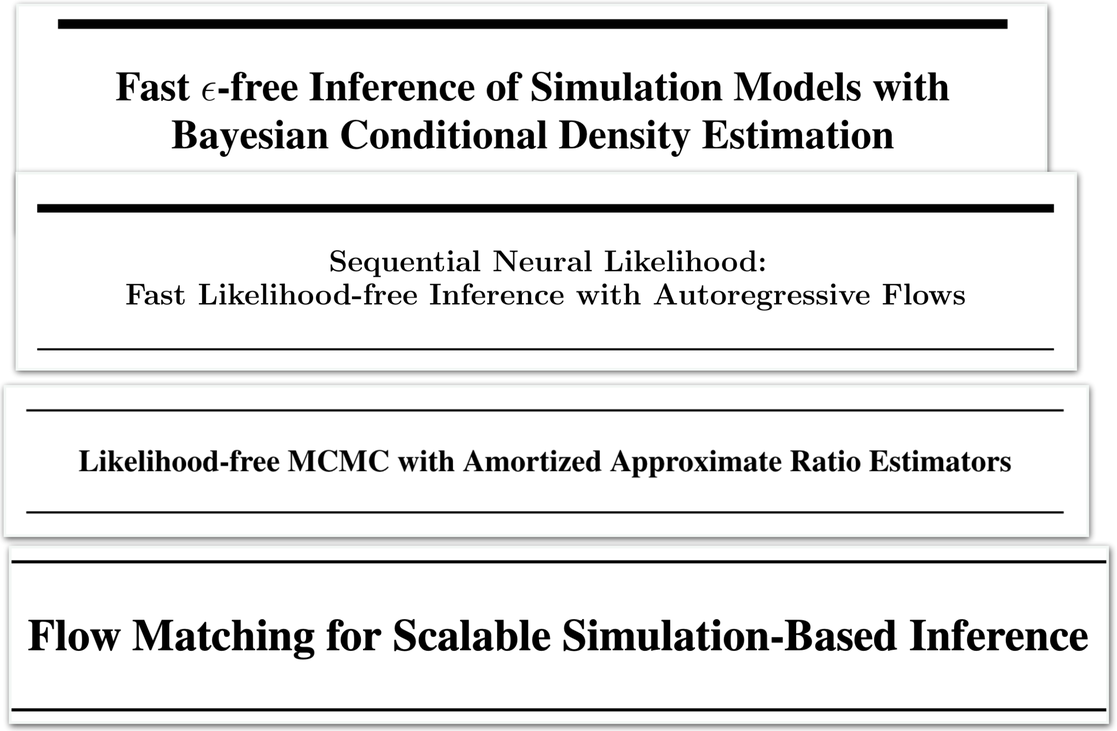
Mind the Gap: Methods and Applicability of SBI

NPE, NLE, NRE

SNLE, SNPE, SNRE

diffusion models

flow matching



Tejero-Cantero, Boelts*, Deistler*, Lueckmann*, Durkan* et al. 2020*

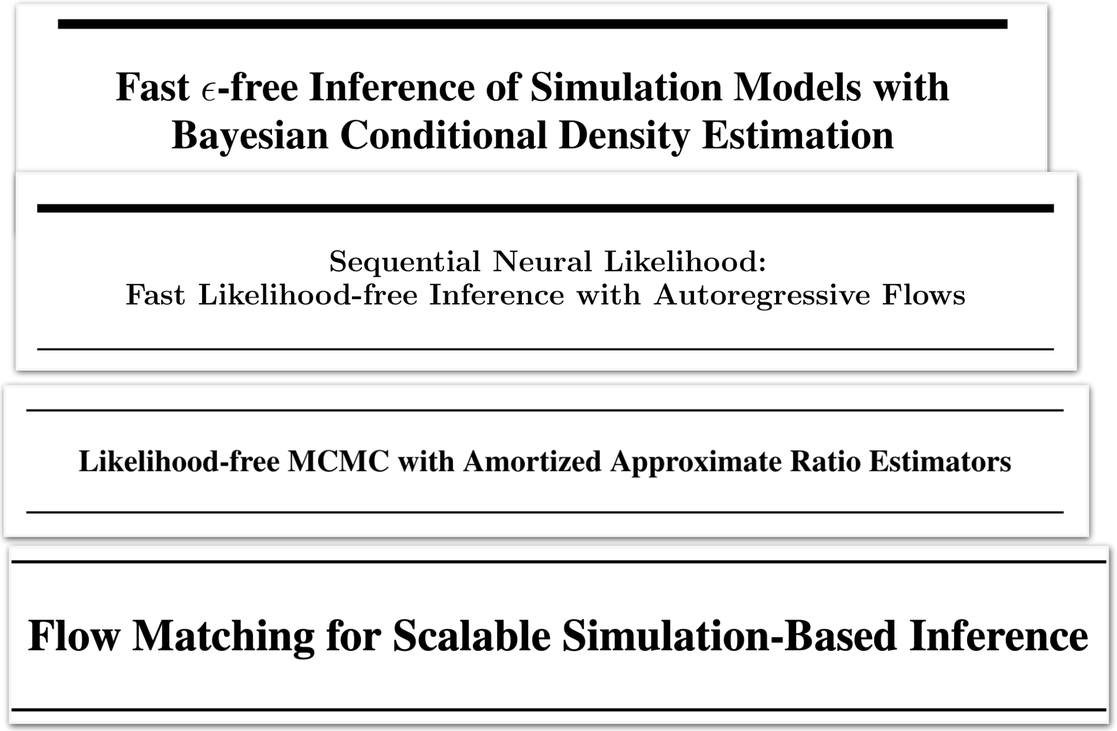
Mind the Gap: Methods and Applicability of SBI

NPE, NLE, NRE

SNLE, SNPE, SNRE

diffusion models

flow matching



- Which method to use?
- How to use it?
- Is there usable code?
- How does it compare to other methods
- …

Tejero-Cantero, Boelts*, Deistler*, Lueckmann*, Durkan* et al. 2020*

Mind the Gap: Methods and Applicability of SBI

NPE, NLE, NRE

SNLE, SNPE, SNRE

diffusion models

flow matching

sbi

Fast ϵ-free Inference of Simulation Models with Bayesian Conditional Density Estimation

Sequential Neural Likelihood:
Fast Likelihood-free Inference with Autoregressive Flows

Likelihood-free MCMC with Amortized Approximate Ratio Estimators

Flow Matching for Scalable Simulation-Based Inference

Tejero-Cantero, Boelts*, Deistler*, Lueckmann*, Durkan* et al. 2020*

Mind the Gap: Methods and Applicability of SBI

NPE, NLE, NRE
SNLE, SNPE, SNRE
diffusion models
flow matching

Fast ϵ-free Inference of Simulation Models with Bayesian Conditional Density Estimation

Sequential Neural Likelihood:
Fast Likelihood-free Inference with Autoregressive Flows

Likelihood-free MCMC with Amortized Approximate Ratio Estimators

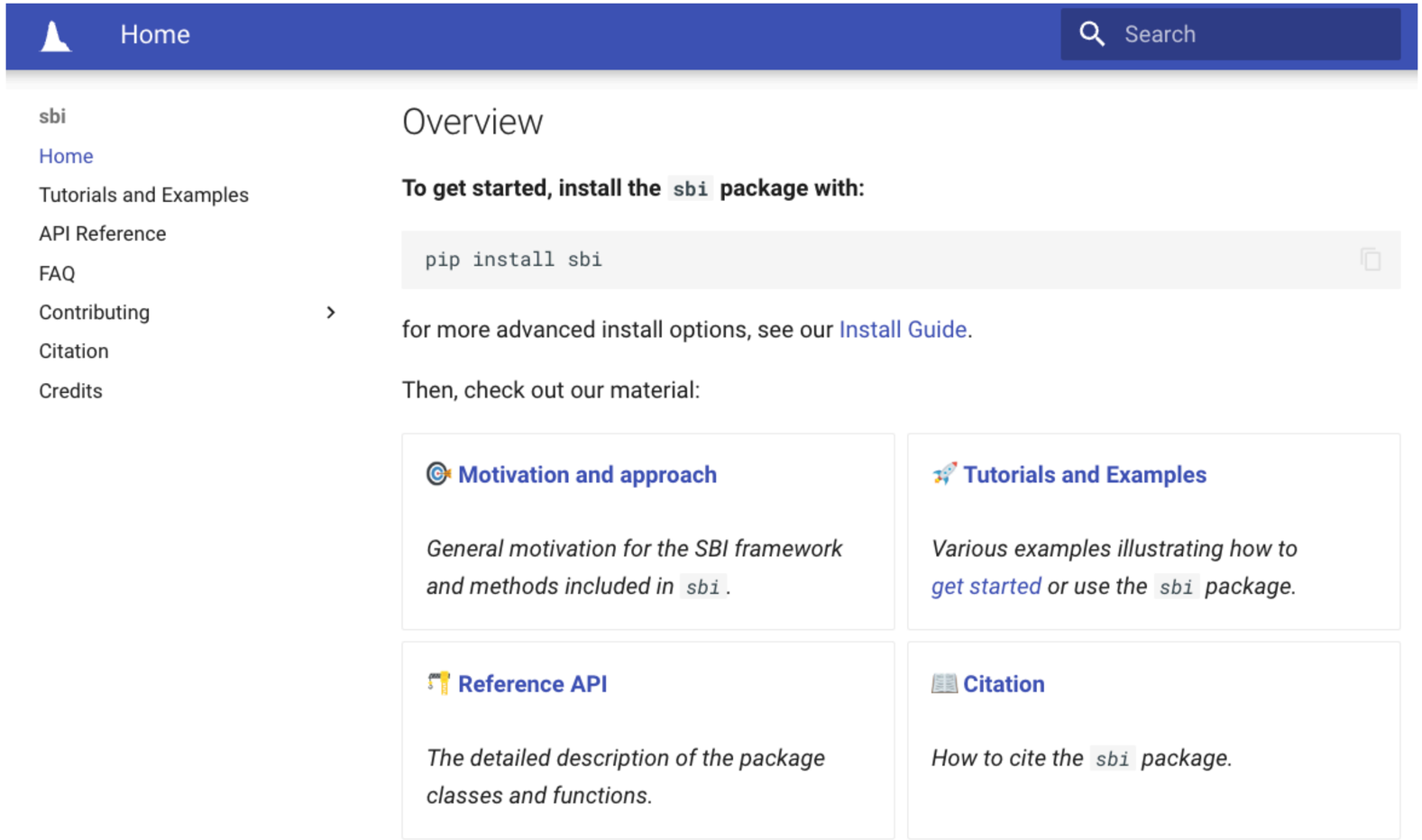
Flow Matching for Scalable Simulation-Based Inference

```python
import torch
from sbi.inference import NPE

# define shifted Gaussian simulator.
def simulator(θ):
  return θ + torch.randn_like(θ)

# draw parameters from Gaussian prior.
θ = torch.randn(1000, 2)
# simulate data
x = simulator(θ)

# choose sbi method and train
inference = NPE()
inference.append_simulations(θ, x).train()

# do inference given observed data
x_o = torch.ones(2)
posterior = inference.build_posterior()
samples = posterior.sample((1000,), x=x_o)
```

TransferLab

Tejero-Cantero, Boelts*, Deistler*, Lueckmann*, Durkan* et al. 2020*

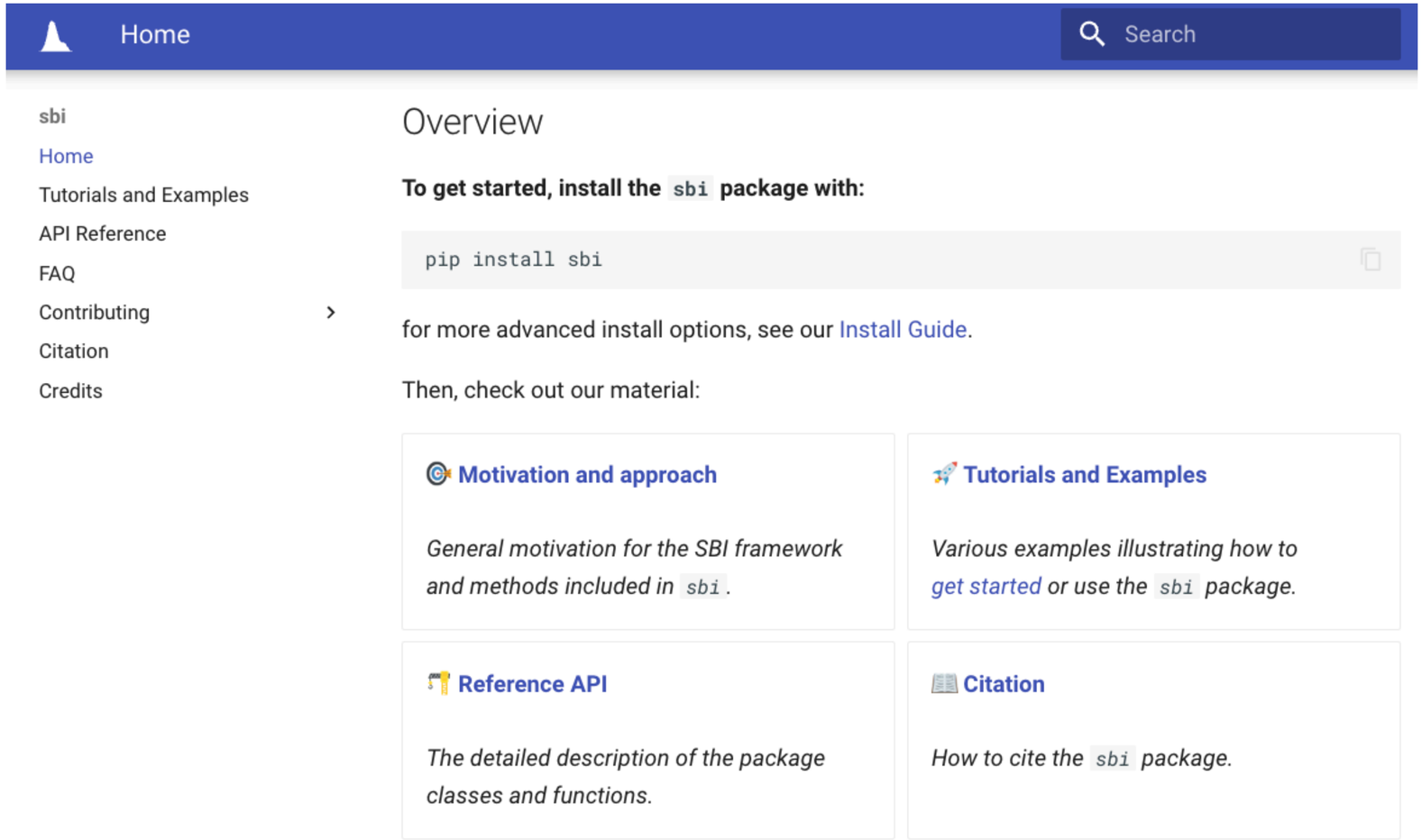
An accessible toolkit for applying SBI



Tejero-Cantero, Boelts*, Deistler*, Lueckmann*, Durkan* et al. 2020*

An accessible toolkit for applying SBI

- Off-the-shelf SBI methods
- Embedding networks
- Validation techniques
- Plotting tools

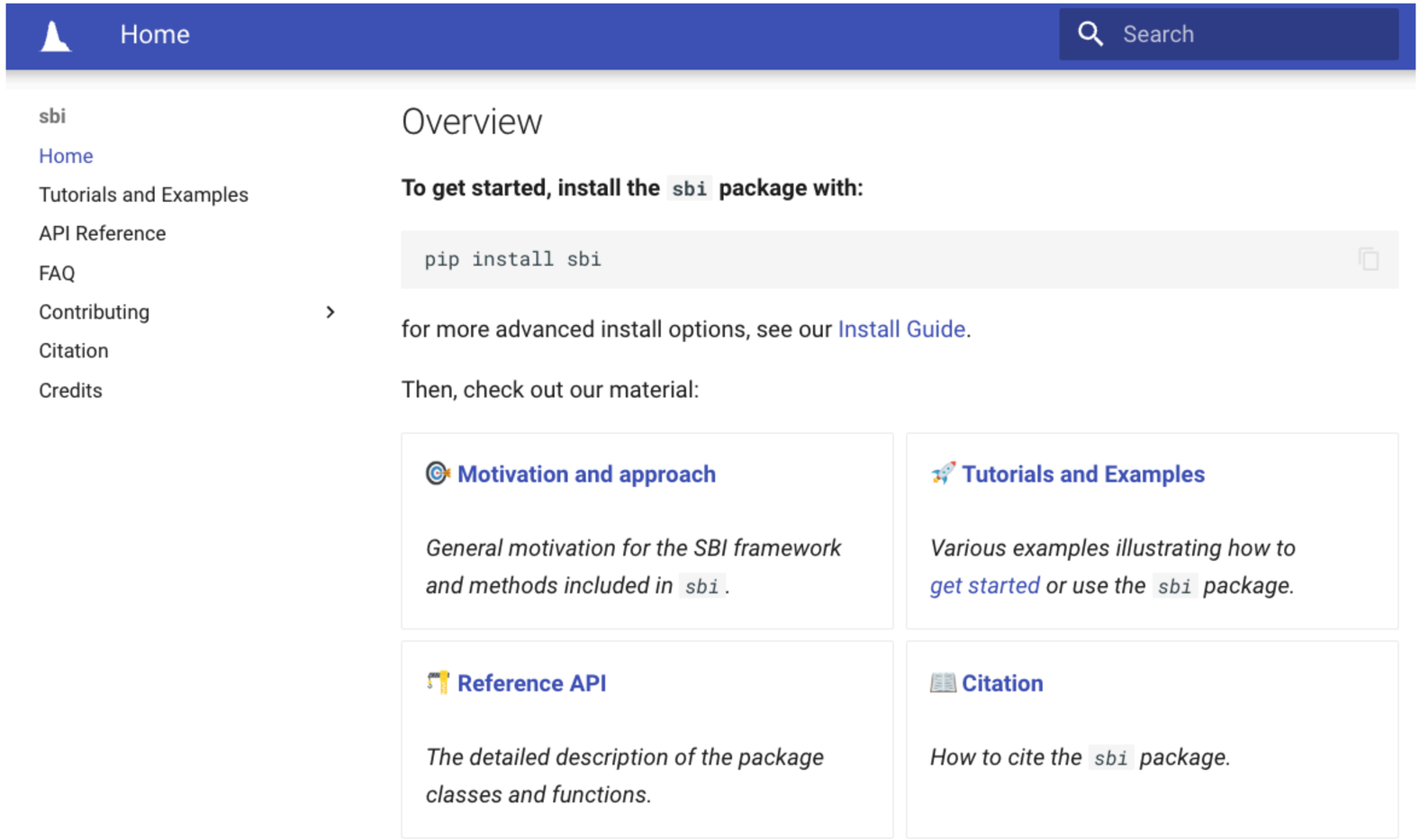


Tejero-Cantero, Boelts*, Deistler*, Lueckmann*, Durkan* et al. 2020*

An accessible toolkit for applying SBI

- Off-the-shelf SBI methods
- Embedding networks
- Validation techniques
- Plotting tools

- Thorough documentation
- Tutorials and examples
- Issues, discussions



Tejero-Cantero, Boelts*, Deistler*, Lueckmann*, Durkan* et al. 2020*

Demo: SBI on the SIR model

SBI toolkit has been adopted by the community

SBI toolkit has been adopted by the community

Article

A biophysical account of multiplication by a single neuron

PNAS RESEARCH ARTICLE | NEUROSCIENCE

Energy-efficient network activity from disparate circuit parameters

Michael Deistler[a], Jakob H. Macke[a,b,1,2], and Pedro J. Gonçalves[a,c,1,2]

ELSEVIER

Neural Networks

Volume 163, June 2023, Pages 178-194

Amortized Bayesian inference on generative dynamical network models of epilepsy using deep neural density estimators

Journal of Neural Engineering

PAPER

Fast inference of spinal neuromodulation for motor control using amortized neural networks

Using simulation-based inference to determine the parameters of an integrated hydrologic model: a case study from the upper Colorado River basin

Calibrating Agent-based Models to Microdata with Graph Neural Networks

Farmer[1 2 4] Sebastian M. Schmon[3 5]

THE ASTROPHYSICAL JOURNAL

OPEN ACCESS

Calibrating Cosmological Simulations with Implicit Likelihood Inference Using Galaxy Growth Observables

Bayesian Object Models for Robotic Interaction with Differentiable Probabilistic Programming

Krishna Murthy Jatavallabhula, Miles

Proceedings of The 6th Conference on

Neural simulation-based inference approach for characterizing the Galactic Center γ-ray excess

Siddharth Mishra-Sharma and Kyle Cranmer

Phys. Rev. D **105**, 063017 – Published 23 March 2022

TransferLab

SBI toolkit has been adopted by the community

Article

A biophysical account of multiplication by a single neuron

PNAS RESEARCH ARTICLE | NEUROSCIENCE

Energy-efficient network activity from … circuit parameters

Michael Deistler[a], Jakob H. Macke[a,b,1,2], and Pedro J. Gonçalves[a,c,1,2]

ELSEVIER

Neural Networks

Volume 163, June 2023, Pages 178-194

Amortized Bayesian inference on generative dynamical network models of epilepsy using deep neural density estimators

Journal of Neural Engineering

… Viktor Sip [a] …, …irsa [a] …

… of spinal neuromodulation for motor control …ed neural networks

>50 contributors

>500 GitHub stars

Using simulation-based inference to determine the … of an integrated hydrologic model: a case study from the upper Colorado River basin

Robert Hull … Elena Leonarduzzi, Luis De La Fuente, Hoang Viet Tran, Andrew Bennett, Peter Melchior, Reed … and …

…gent-based Models to Microdata with Graph Neural Networks

… Farmer [1 2 4] Sebastian M. Schmon [3 5]

THE ASTROPHYSICAL JOURNAL

OPEN ACCESS

Calibrating Cosmological Simulations with Implicit Likelihood Inference Using Galaxy Growth Observables

Bayesian Object Models for Robotic Interaction with Differentiable Probabilistic Programming

Krishna Murthy Jatavallabhula, Miles …

Proceedings of The 6th Conference on …

Neural simulation-based inference approach for characterizing the Galactic Center γ-ray excess

Siddharth Mishra-Sharma and Kyle Cranmer

Phys. Rev. D **105**, 063017 – Published 23 March 2022

TransferLab

New contributors welcome!

NUMFOCUS
OPEN CODE = BETTER SCIENCE

TransferLab

New contributors welcome!

- **A lot of potential** in scientific and industrial applications

New contributors welcome!

- **A lot of potential** in scientific and industrial applications
- **A lot to improve**: new methods, API design, documentation

New contributors welcome!

- **A lot of potential** in scientific and industrial applications
- **A lot to improve**: new methods, API design, documentation
- Actively maintained

New contributors welcome!

NUMFOCUS
OPEN CODE = BETTER SCIENCE

- **A lot of potential** in scientific and industrial applications
- **A lot to improve**: new methods, API design, documentation
- Actively maintained

- Upcoming events:

TransferLab

New contributors welcome!

- **A lot of potential** in scientific and industrial applications
- **A lot to improve**: new methods, API design, documentation
- Actively maintained

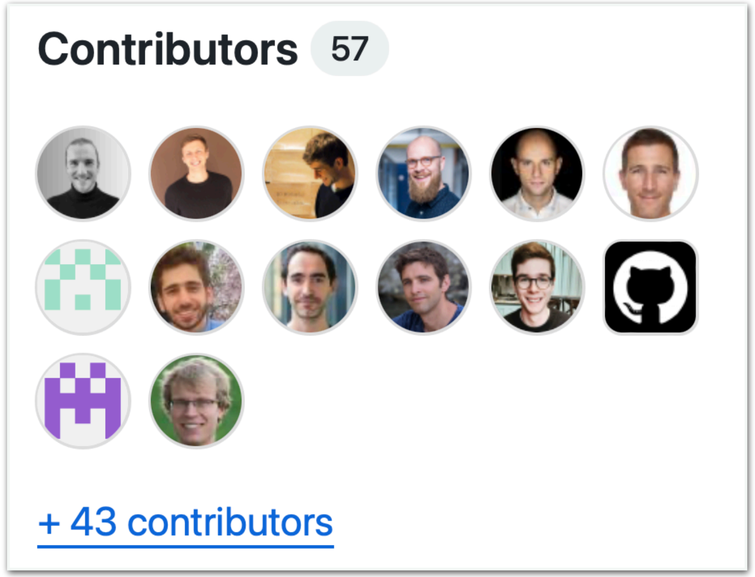
- Upcoming events:
  - **Tomorrow** 9am-11am “New Contributor Sprint”

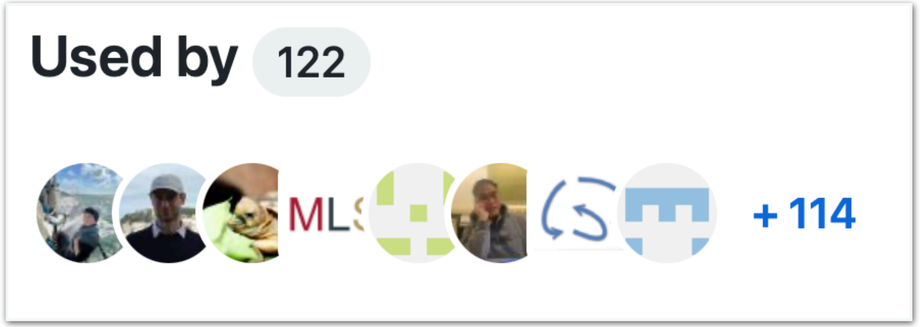
New contributors welcome!

- **A lot of potential** in scientific and industrial applications
- **A lot to improve**: new methods, API design, documentation
- Actively maintained

- Upcoming events:
  - **Tomorrow** 9am-11am “New Contributor Sprint”
  - Spring 2025: SBI Hackathon & Workshop

Acknowledgments





MLS

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Tübingen AI Center

TransferLab

appliedAI
institute
for europe

TransferLab

appliedAI institute for europe

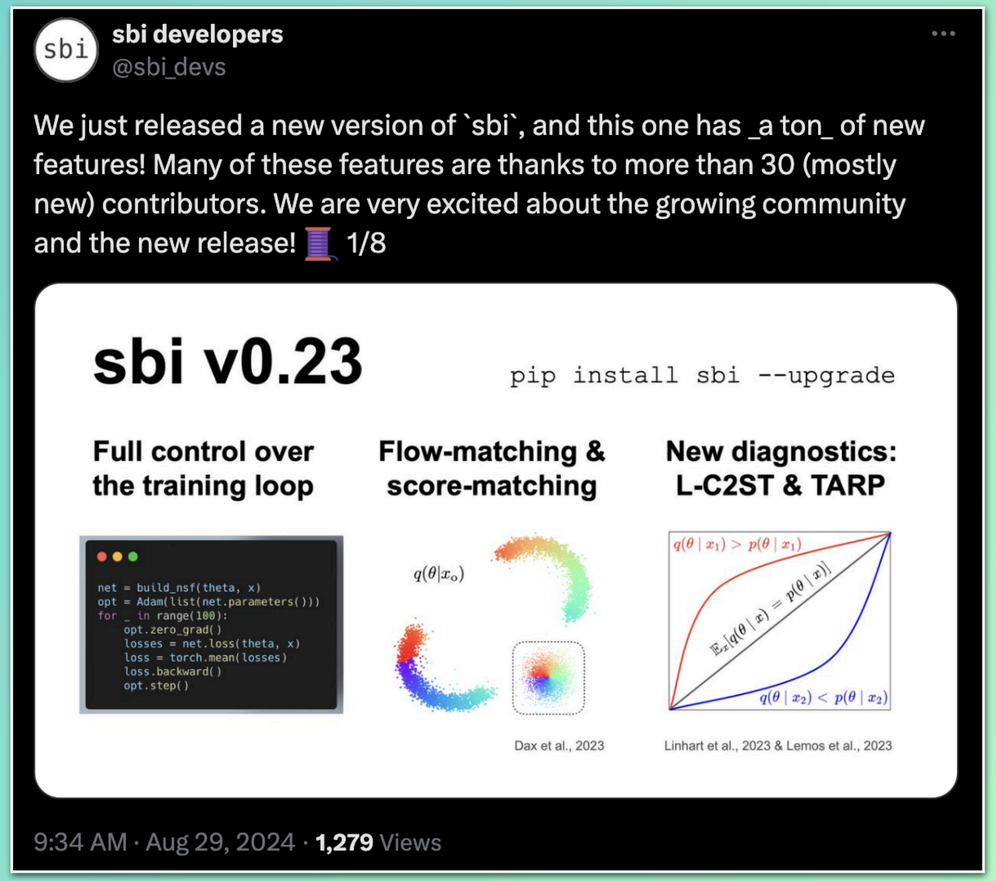
Thank you!

Questions?

Try it out: **pip install sbi**
GitHub: github.com/sbi-dev/sbi

X, Mastodon: @sbi_devs

TransferLab website: transferlab.ai



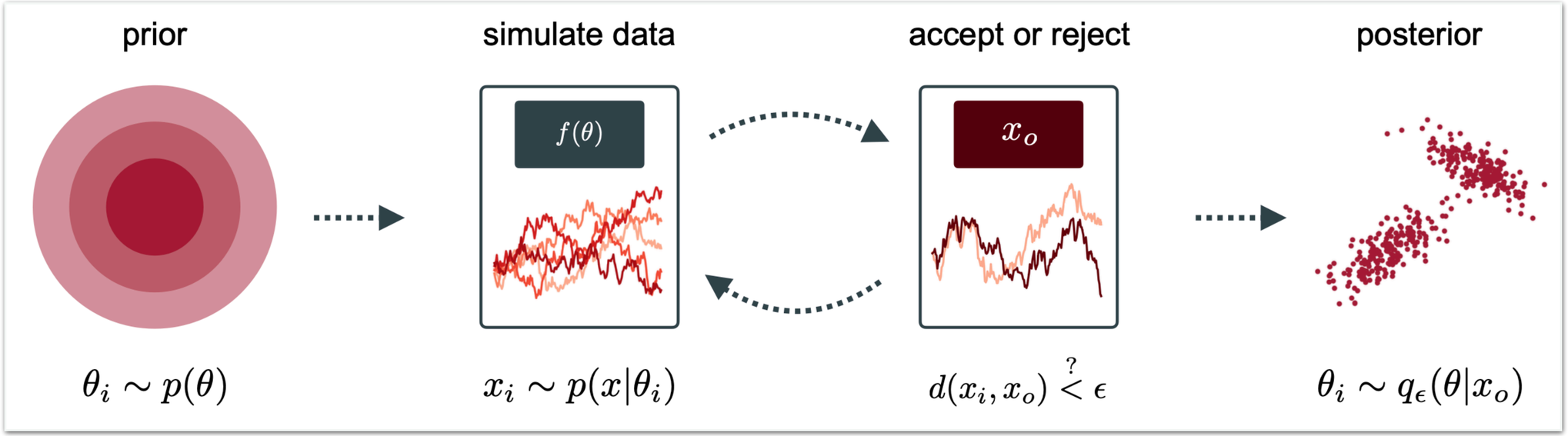
Supplementary slides

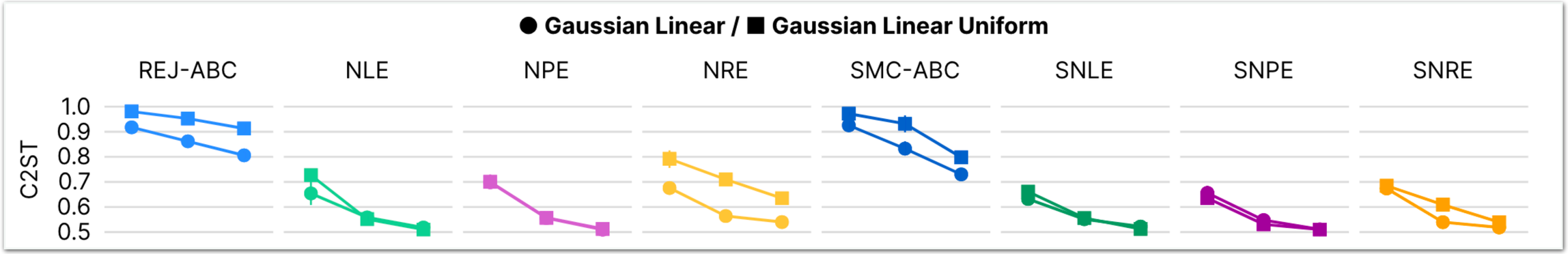
Is SBI for you?

Some heuristics

- Has been applied to simulators with ~10 parameters, ~100 in some cases
- Depends on how much data you can generate: runtime ~sec
- High-dimensional data can be embedded: CNNs, RNNs, transformers etc.
- Different methods for different settings: i.i.d. data, hierarchical models, amortized inference, sequential inference

Approximate Bayesian Computation (ABC)





Diggle et al. 1984; Beaumont et al. 2002; Sisson et al. 2008; Lueckmann et al. 2021

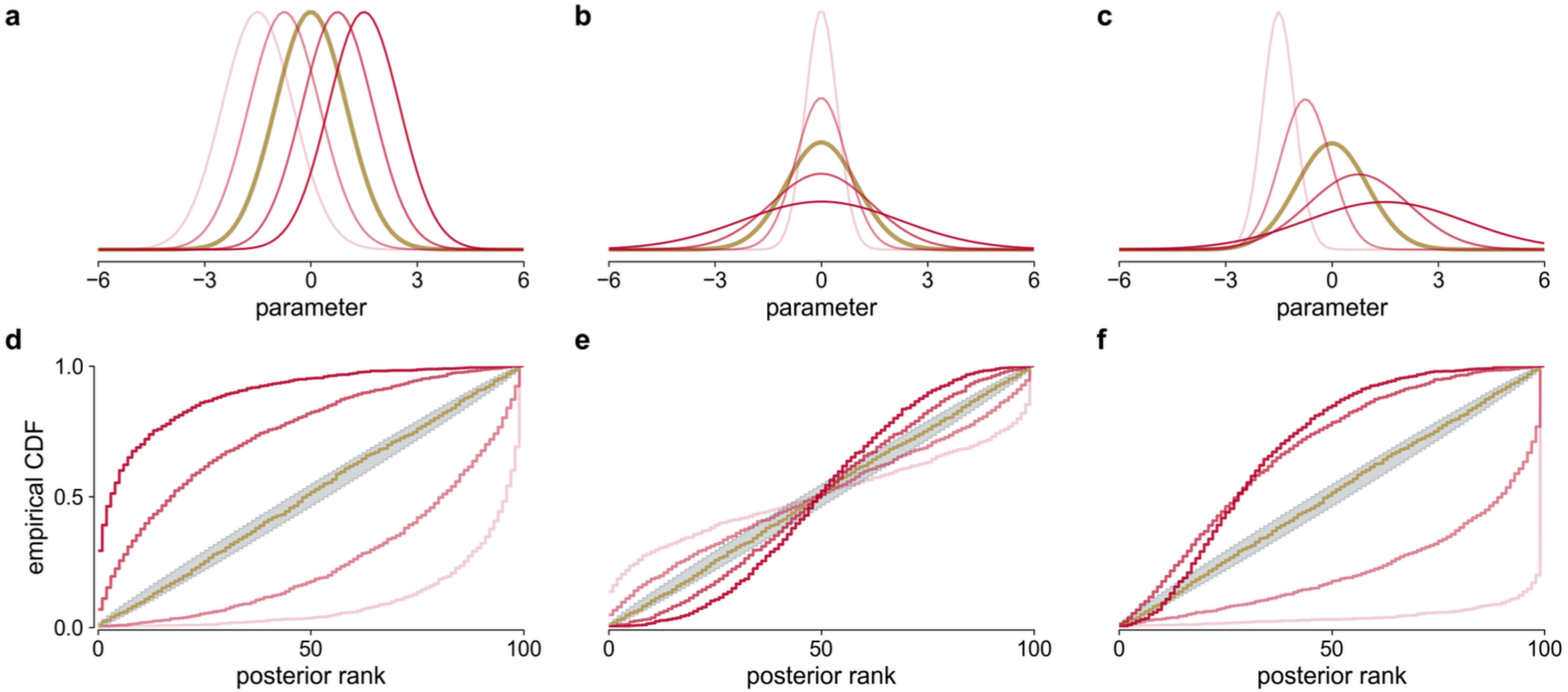
NLE / NPE loss functions

Algorithm 5: Single round Neural Posterior Estimation as in Papamakarios and Murray (2016)

for $j = 1 : N$ **do**
  Sample $\boldsymbol{\theta}_j \sim p(\boldsymbol{\theta})$
  Simulate $\mathbf{x}_j \sim p(\mathbf{x}|\boldsymbol{\theta}_j)$
end
$\boldsymbol{\phi} \leftarrow \arg\min \sum_j^N - \log q_{F(\mathbf{x}_j, \boldsymbol{\phi})}(\boldsymbol{\theta}_j)$
Set $\hat{p}(\boldsymbol{\theta}|\mathbf{x}_o) = q_{F(\mathbf{x}_o, \boldsymbol{\phi})}(\boldsymbol{\theta})$
return Samples from $\hat{p}(\boldsymbol{\theta}|\mathbf{x}_o)$; $q_{F(\mathbf{x}, \boldsymbol{\phi})}(\boldsymbol{\theta})$

Papamakarios & Murray 2016

Simulation-based calibration



$$\theta^* \sim p(\theta)$$
$$x^* \sim p(x|\theta^*)$$

$$\{\theta_1, \ldots \theta_L\} \sim p_i(\theta|x^*)$$

$$r_i = \sum_{j=1}^{L} \mathbb{I}[f(\theta_j) < f(\theta_i^*)] \in [0, L],$$

Talts et al. 2018; figure: Boelts, 2023

Model misspecification in SBI

- **Notation**: $x \sim p(x \mid \theta)$ simulated data; $y \sim p^*(y)$ observed data
- **Problem**: simulator is misspecified
- **Approach**: assume an error model $\tilde{x} \sim p(x \mid y)$, e.g., “spike and slab”

Algorithm 1: Robust neural posterior estimation (RNPE)

for i *in* $1 : N$ **do**
1 Sample $\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta})$
2 Simulate $\boldsymbol{x}_i \sim p(\boldsymbol{x} \mid \boldsymbol{\theta})$
end
3 Train NPE $q(\boldsymbol{\theta} \mid \boldsymbol{x})$ on $\{(\boldsymbol{\theta}_i, \boldsymbol{x}_i)\}_{i=1}^N$
4 Train $q(\boldsymbol{x})$ on $\{\boldsymbol{x}_i\}_{i=1}^N$
5 Sample $\tilde{\boldsymbol{x}}_m \sim \hat{p}(\boldsymbol{x} \mid \boldsymbol{y}_o) \propto p(\boldsymbol{y}_o \mid \boldsymbol{x}) q(\boldsymbol{x}),\ m = 1, \ldots, M$ using MCMC
6 Sample $\tilde{\boldsymbol{\theta}}_m \sim q(\boldsymbol{\theta} \mid \tilde{\boldsymbol{x}}_m),\ m = 1, \ldots, M$
return $\{(\tilde{\boldsymbol{\theta}}_m, \tilde{\boldsymbol{x}}_m)\}_{m=1}^M$, samples drawn approximately from $p(\boldsymbol{\theta}, \boldsymbol{x} \mid \boldsymbol{y}_o)$

Ward et al. 2022