Recurrent machines for likelihood-free inference

Arthur Pesah
KTH

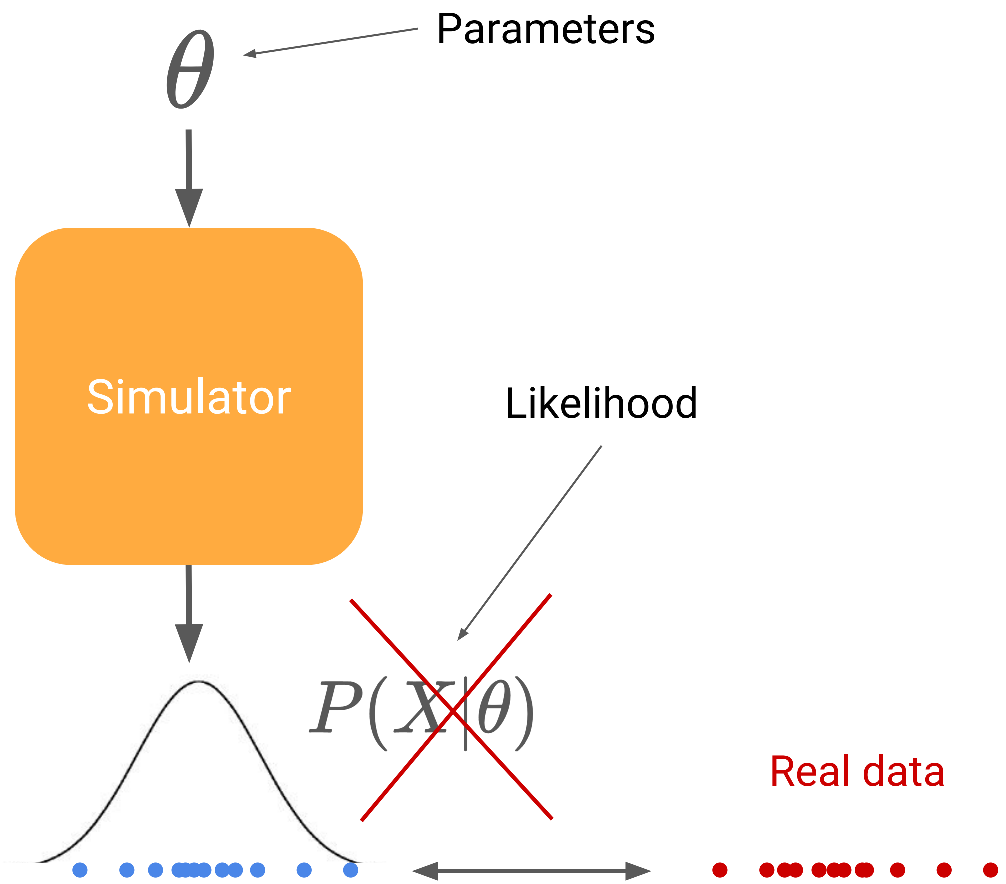
Antoine Wehenkel
ULiège

Gilles Louppe
ULiège



Likelihood-free Inference

Likelihood-free inference: what?



Goal
Finding the parameters corresponding to real data

How?
Maximum likelihood

But...
We don’t have the likelihood!

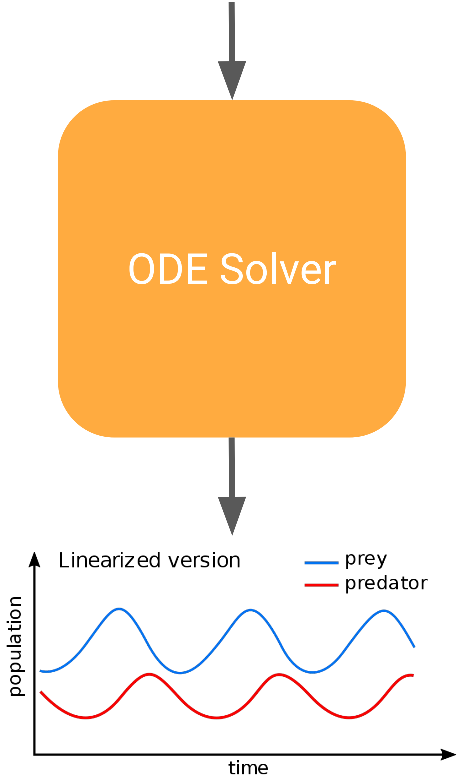
Likelihood-free inference: when?

Likelihood-free inference: when?

Example: Population biology

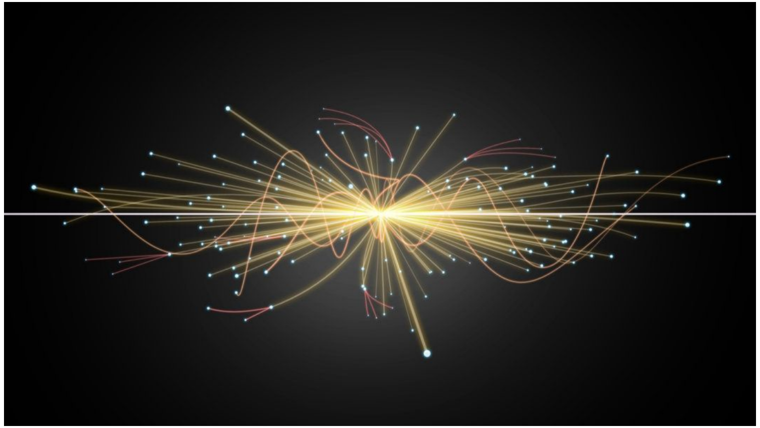
The evolution of a population can be modelled by a differential equation that can be solved by a simulator (numerical solver).

Coefficients of the differential equations.



Likelihood-free inference: when?

Particle Physics



Particle accelerators (like the LHC) produce particle collisions and observe the resulting particles with detectors.

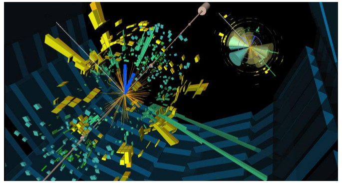
Physical constants (mass of particles, strength of interactions, etc.)

↓

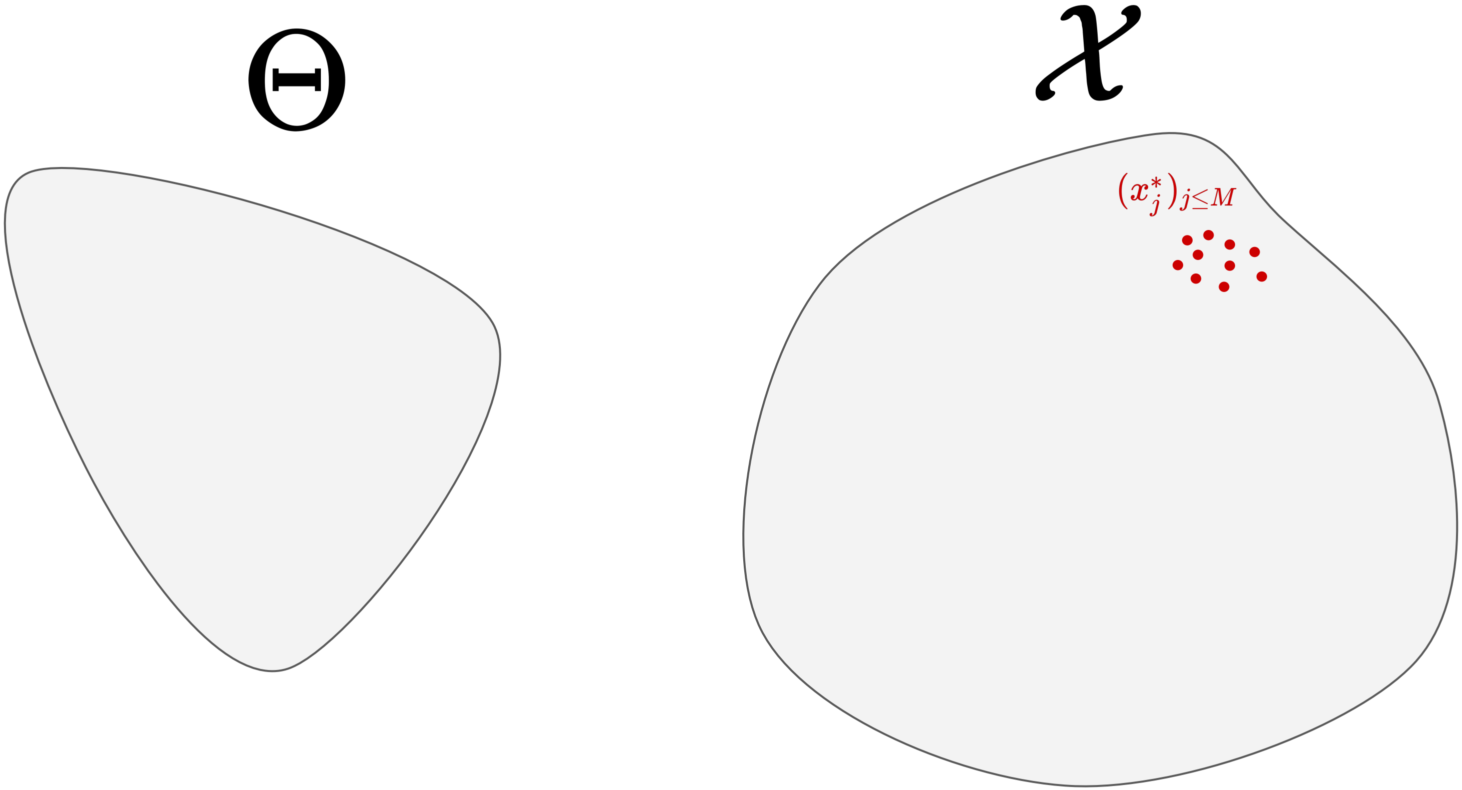
Particle collisions simulator (e.g. Geant4)

↓

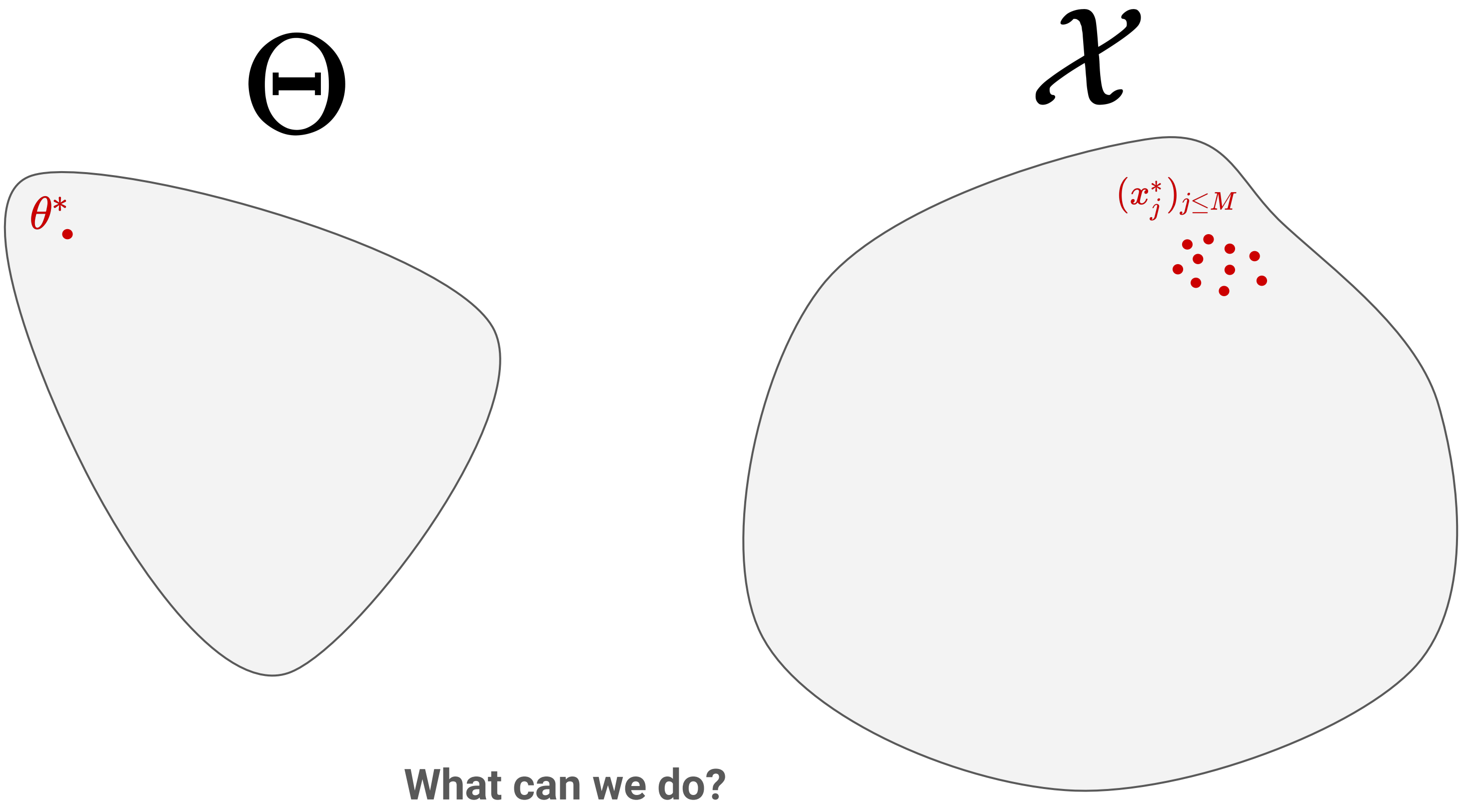
Detectors response after a collision



Likelihood-free inference: how?



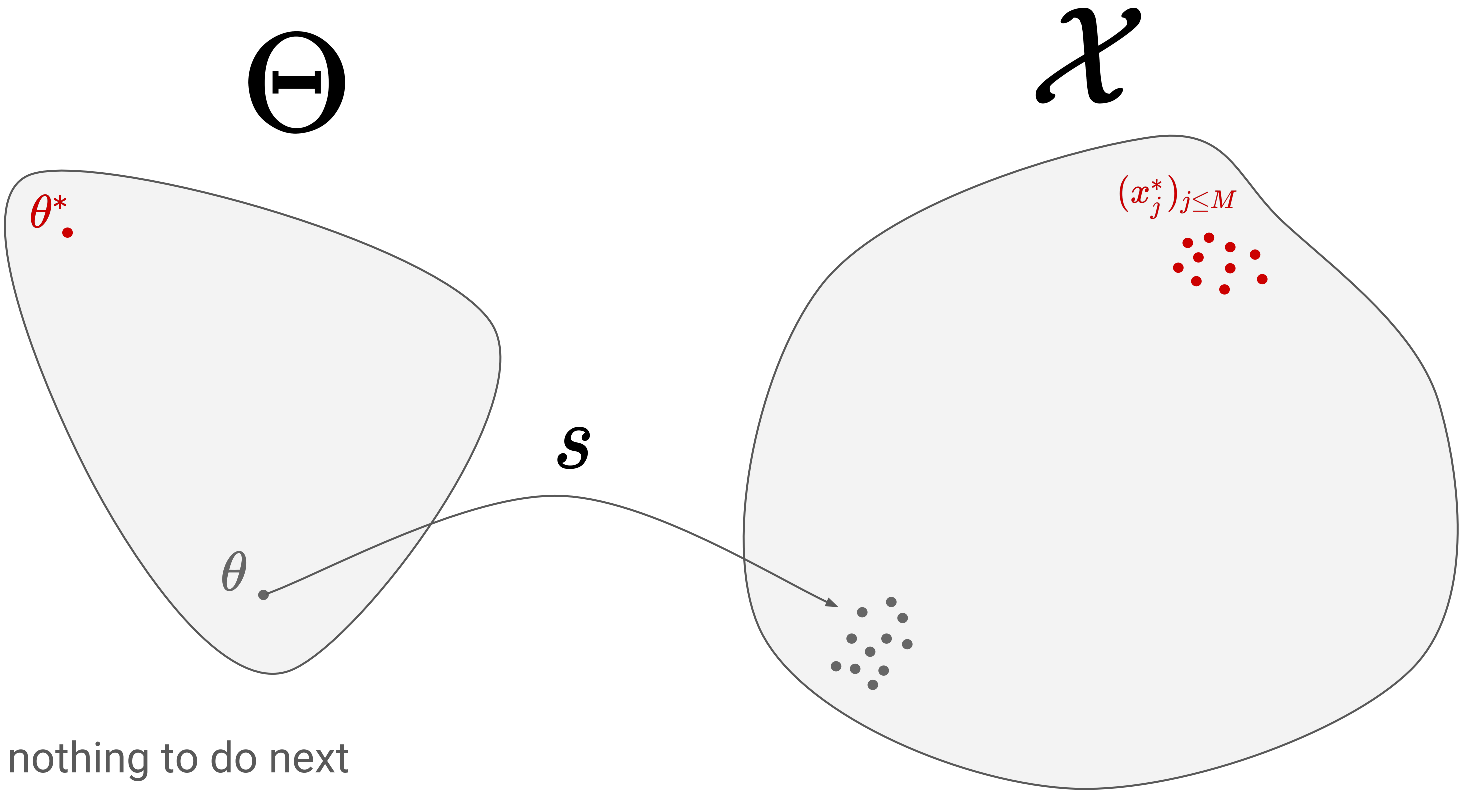
Likelihood-free inference: how?



What can we do?

Likelihood-free inference: how?

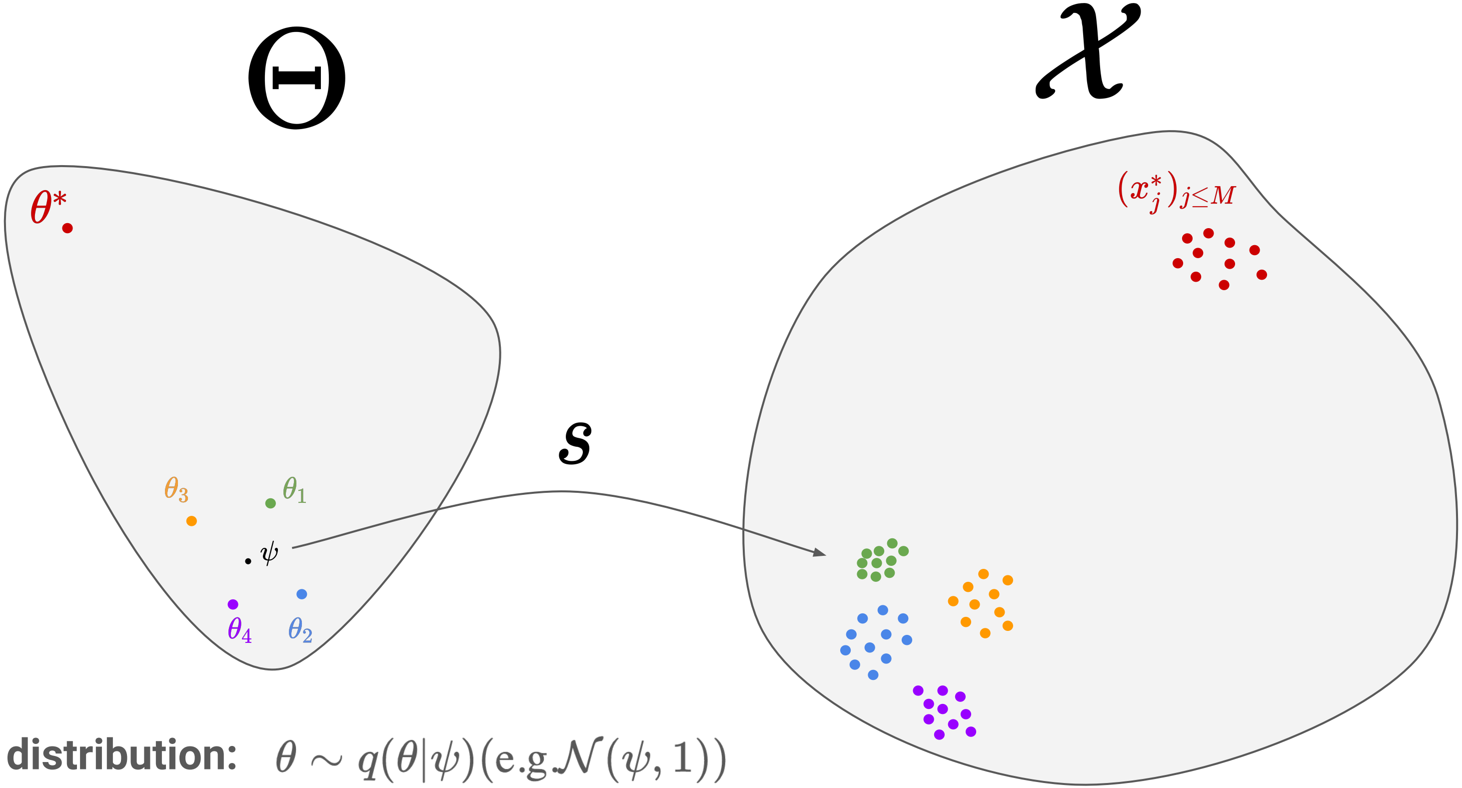
Idea 1: choose a random parameter and simulate it



Problem: nothing to do next

Likelihood-free inference: how?

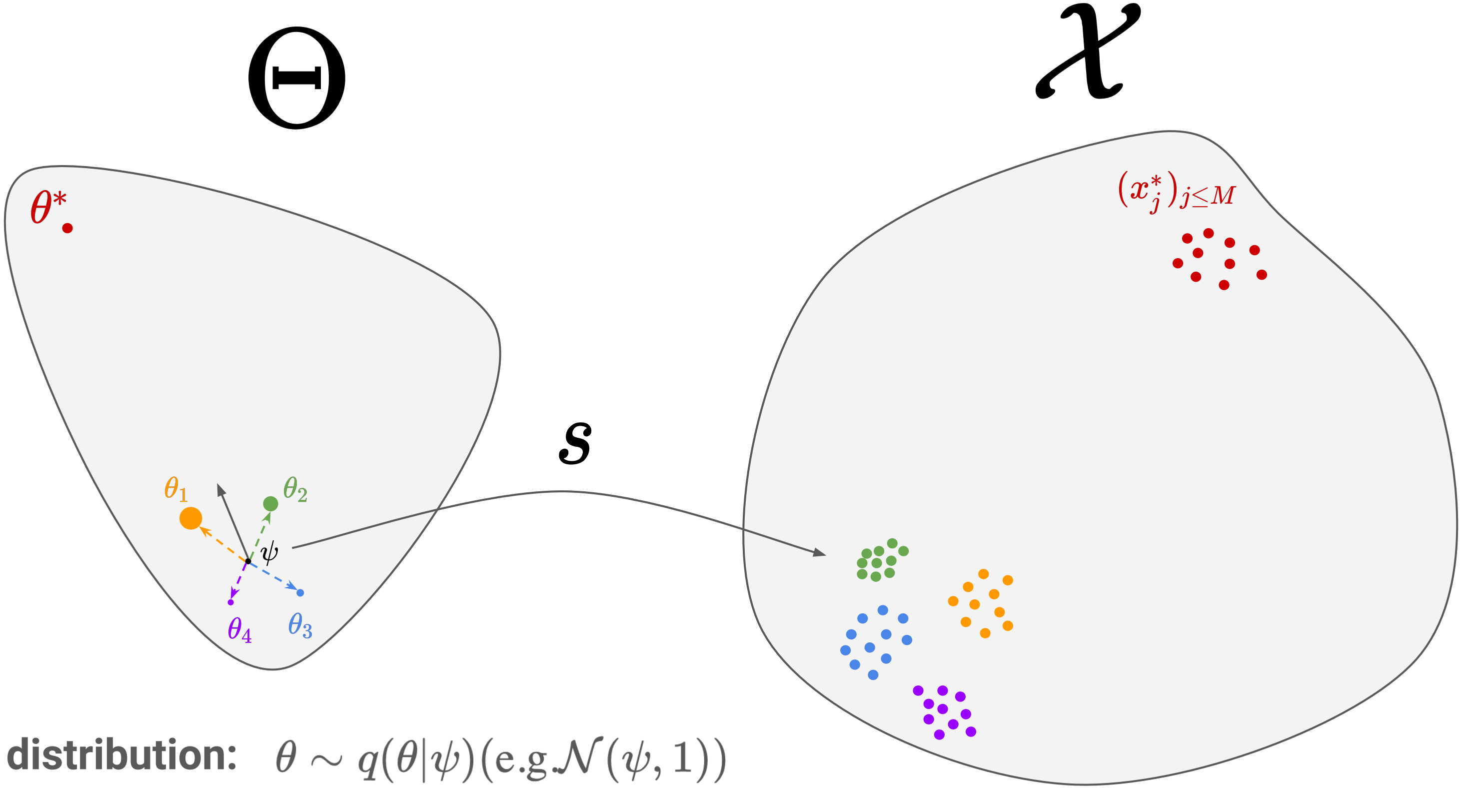
Idea 2: sample several parameters from a **distribution** and simulate them



Proposal distribution: $\theta \sim q(\theta|\psi)(\text{e.g.}\mathcal{N}(\psi, 1))$

Likelihood-free inference: how?

Then: comparing the different simulated data and choosing an appropriate direction



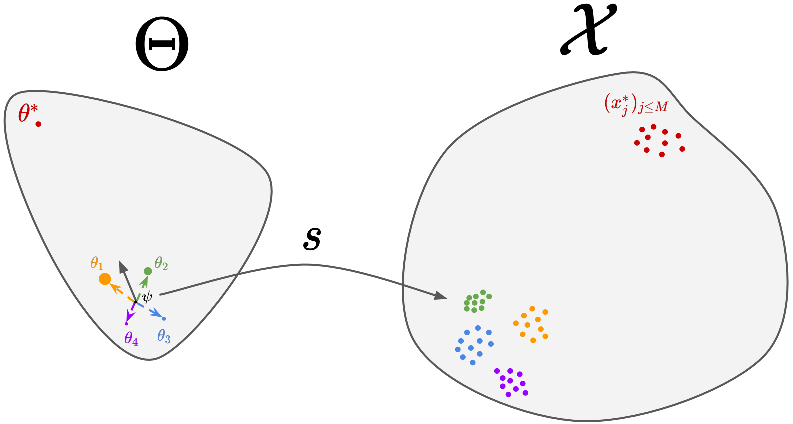
Proposal distribution: $\theta \sim q(\theta|\psi)(\text{e.g.}\mathcal{N}(\psi, 1))$

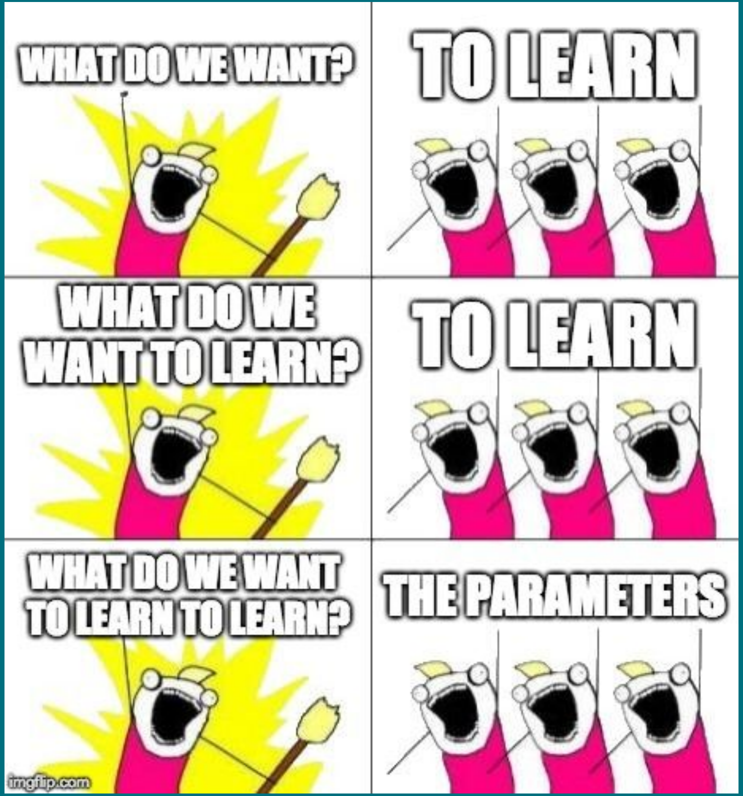
Likelihood-free inference: how?

How to choose the best direction in the parameter space?

- Some algorithms rely on a predefined update rule and a handcrafted similarity measure between the generated and the real samples.

- Why not **learning** this optimization procedure and the similarity measure?





Likelihood-free inference with meta-learning

Likelihood-free inference with meta-learning

Learning to learn by gradient descent by gradient descent

Marcin Andrychowicz[1], **Misha Denil**[1], **Sergio Gómez Colmenarejo**[1], **Matthew W. Hoffman**[1], **David Pfau**[1], **Tom Schaul**[1], **Brendan Shillingford**[1,2], **Nando de Freitas**[1,2,3]

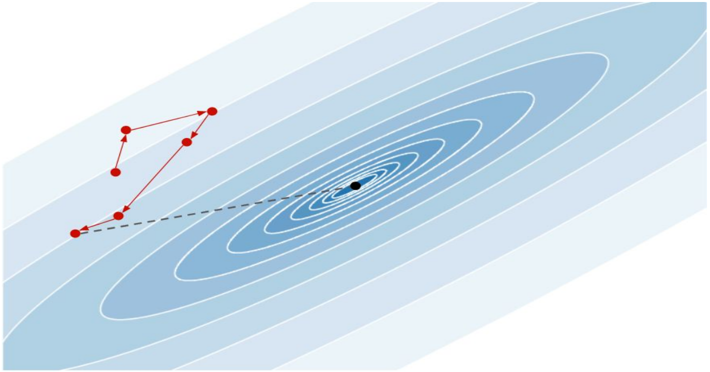
[1]Google DeepMind [2]University of Oxford [3]Canadian Institute for Advanced Research

marcin.andrychowicz@gmail.com
{mdenil,sergomez,mwhoffman,pfau,schaul}@google.com
brendan.shillingford@cs.ox.ac.uk, nandodefreitas@google.com

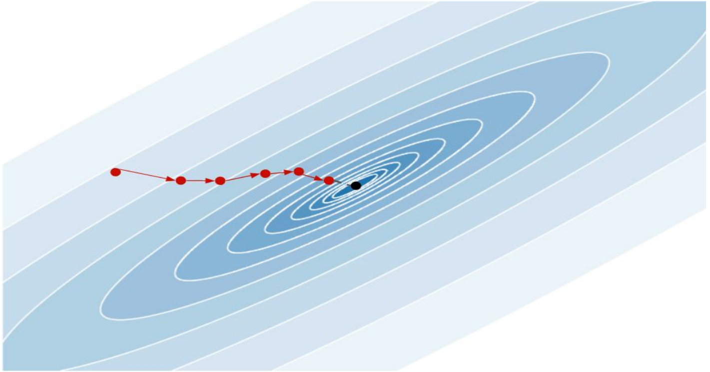
Abstract

The move from hand-designed features to learned features in machine learning has been wildly successful. In spite of this, optimization algorithms are still designed by hand. In this paper we show how the design of an optimization algorithm can be cast as a learning problem, allowing the algorithm to learn to exploit structure in the problems of interest in an automatic way. Our learned algorithms, implemented by LSTMs, outperform generic, hand-designed competitors on the tasks for which they are trained, and also generalize well to new tasks with similar structure. We demonstrate this on a number of tasks, including simple convex problems, training neural networks, and styling images with neural art.

Before meta-training

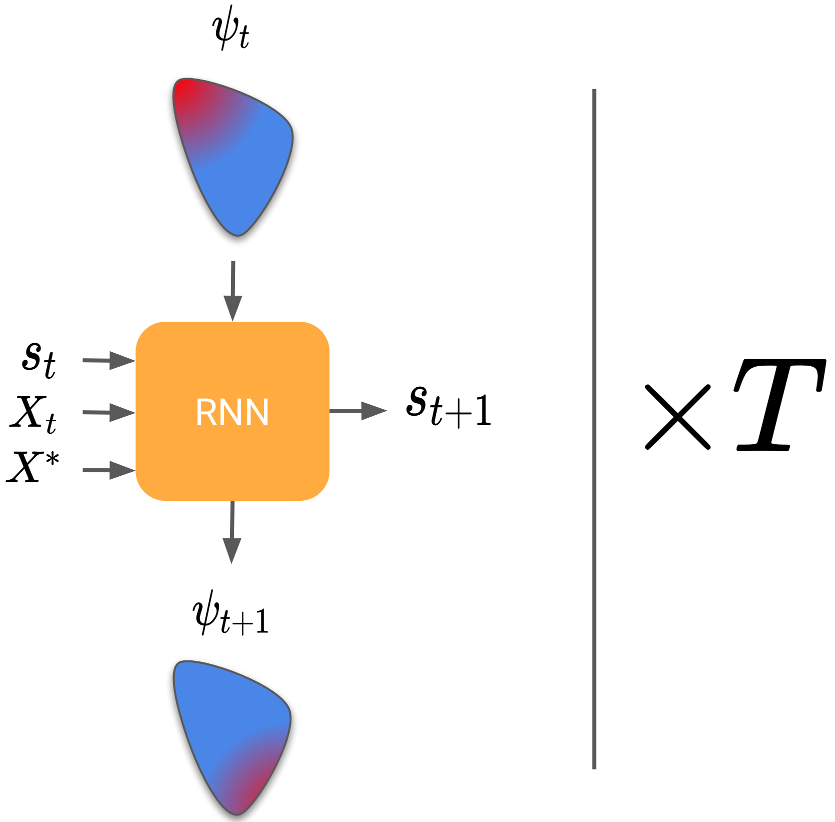


After meta-training

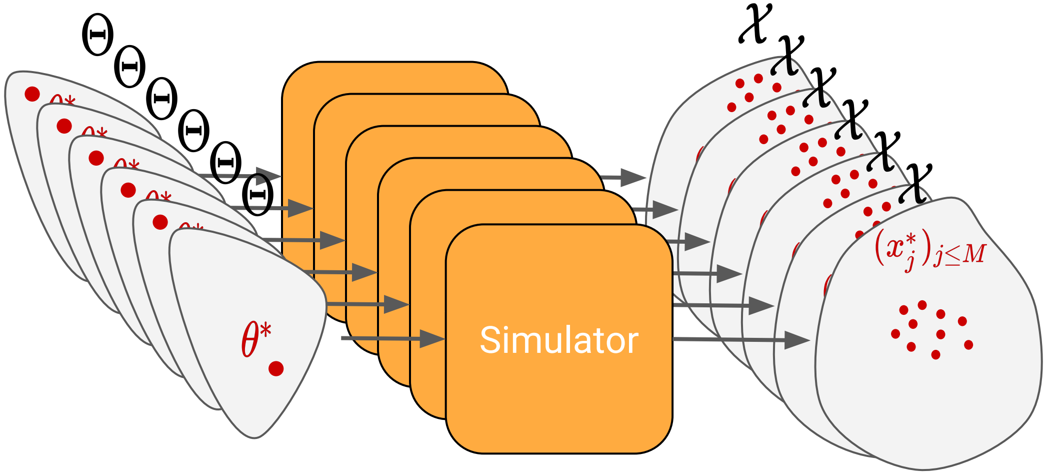


ALFI: Automatic Likelihood-free Inference

Principle: learning a descent using an Recurrent Neural Network

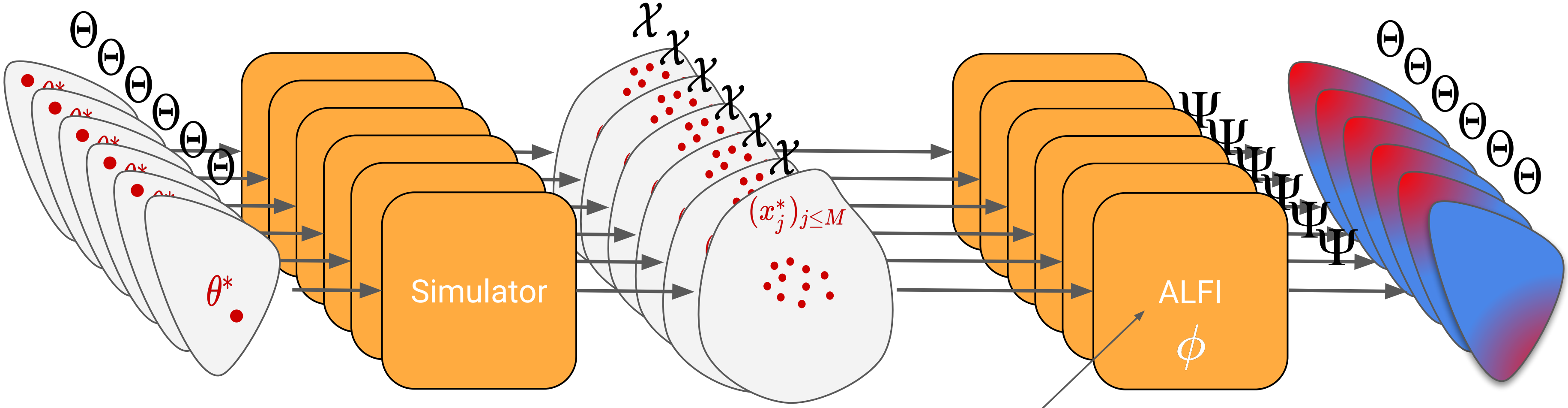


Likelihood-free inference with meta-learning



Generate a meta-dataset

Likelihood-free inference with meta-learning



Generate a meta-dataset

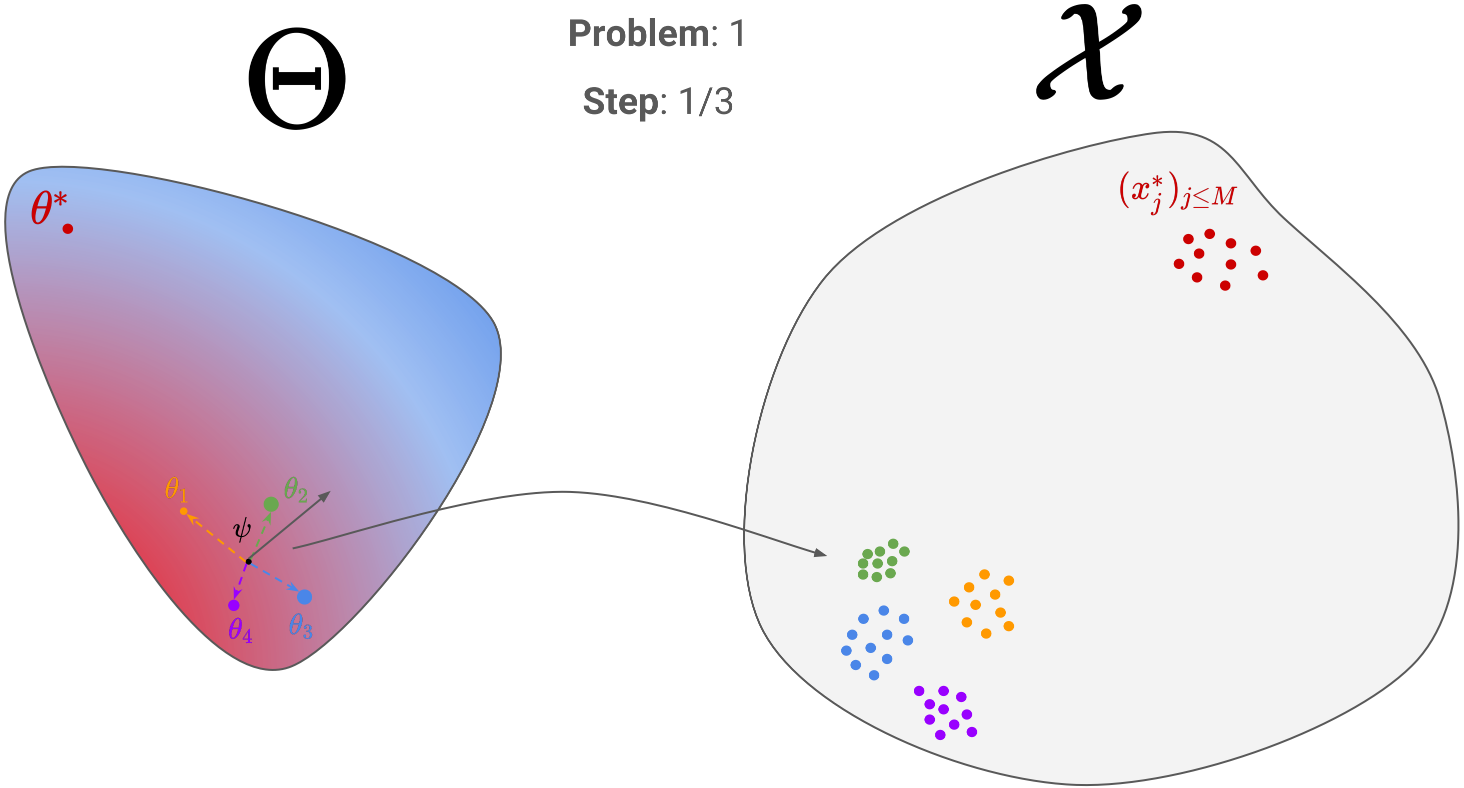
Automatic Likelihood-Free Inference
(the name of our RNN-based machine)

ALFI: Automatic Likelihood-free Inference

Principle: learning a descent using an Recurrent Neural Network

Problem: 1

Step: 1/3

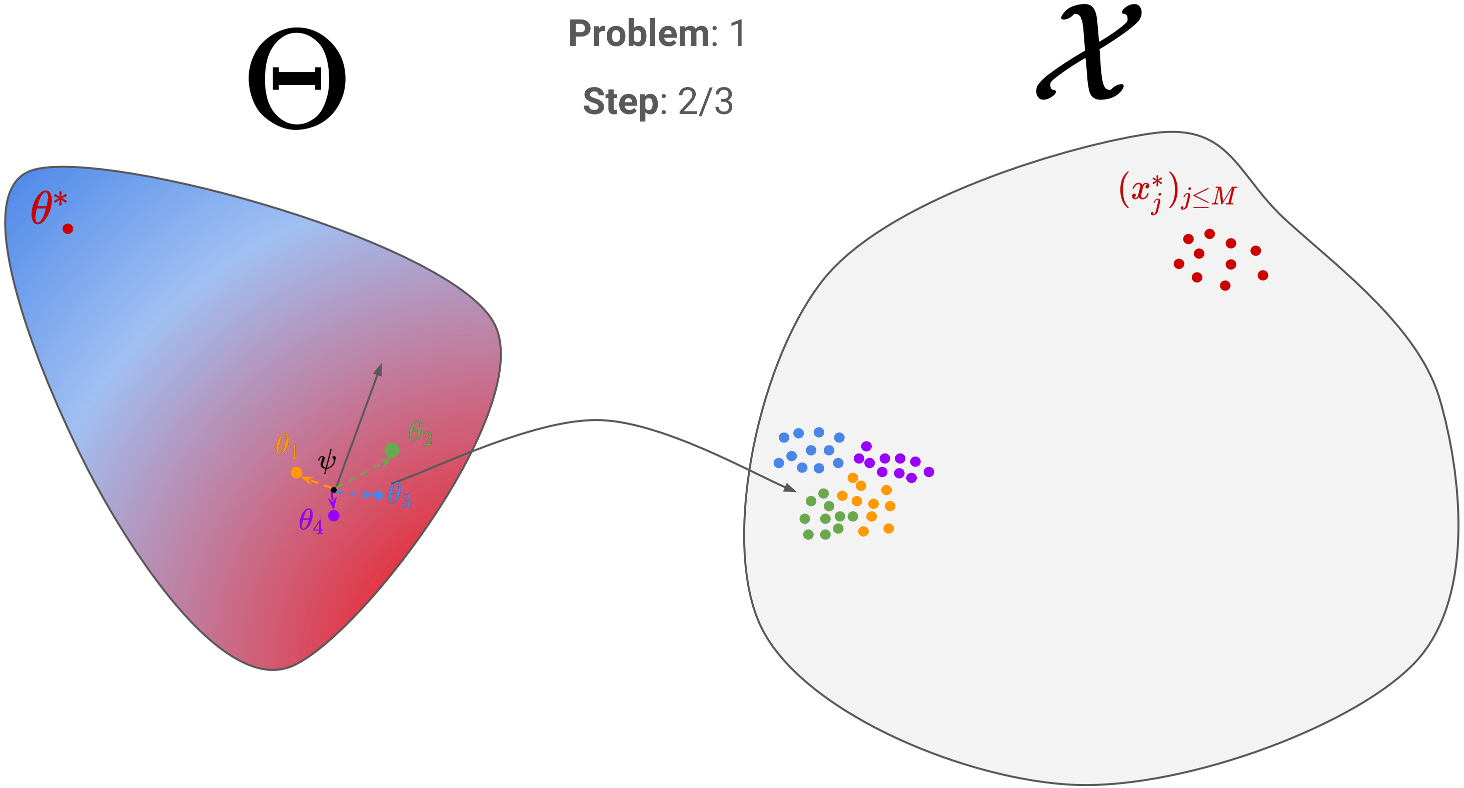


ALFI: Automatic Likelihood-free Inference

Principle: learning a descent using an Recurrent Neural Network

Problem: 1

Step: 2/3

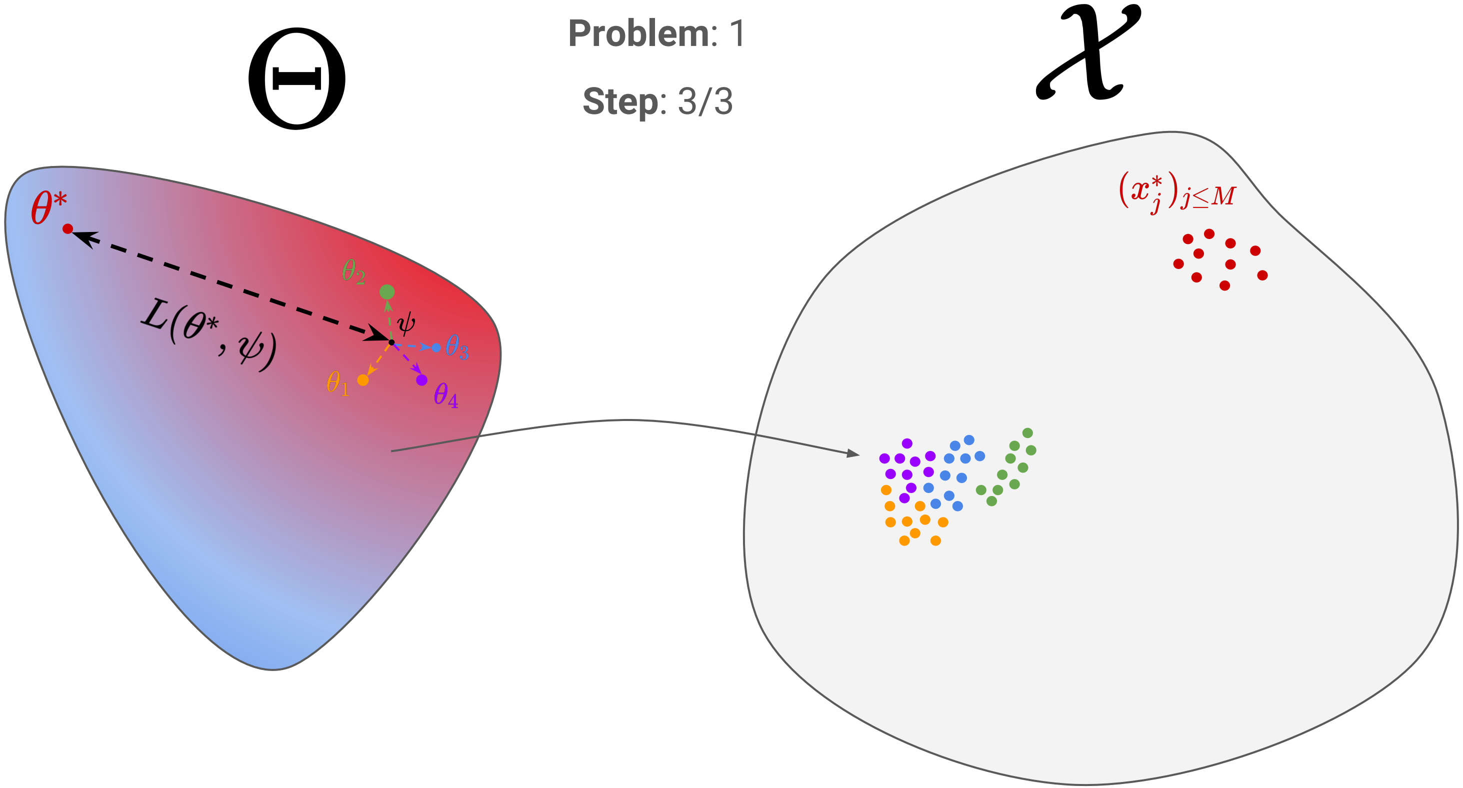


ALFI: Automatic Likelihood-free Inference

Principle: learning a descent using an Recurrent Neural Network

Problem: 1

Step: 3/3

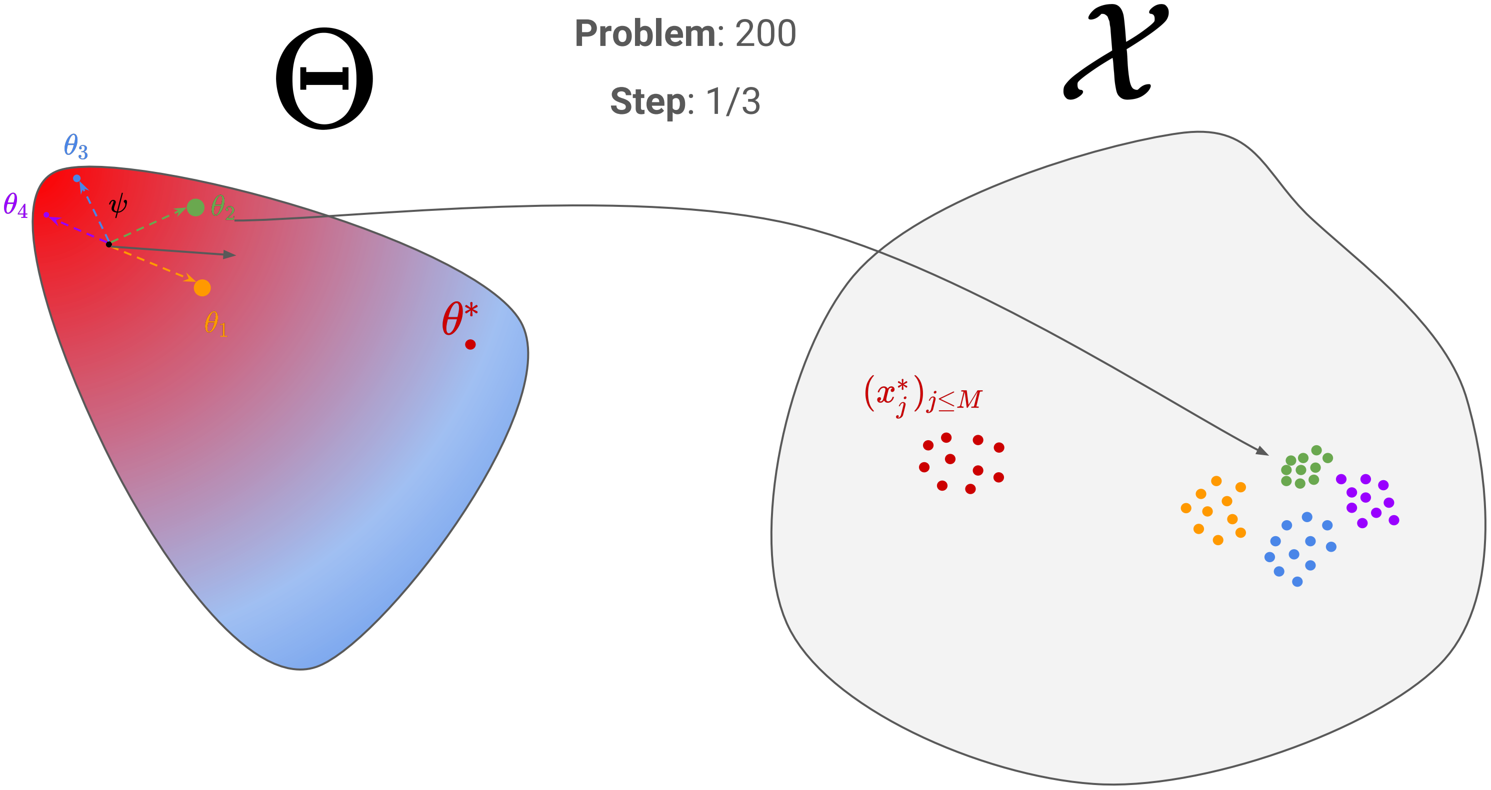


ALFI: Automatic Likelihood-free Inference

Principle: learning a descent using an Recurrent Neural Network

Problem: 200

Step: 1/3

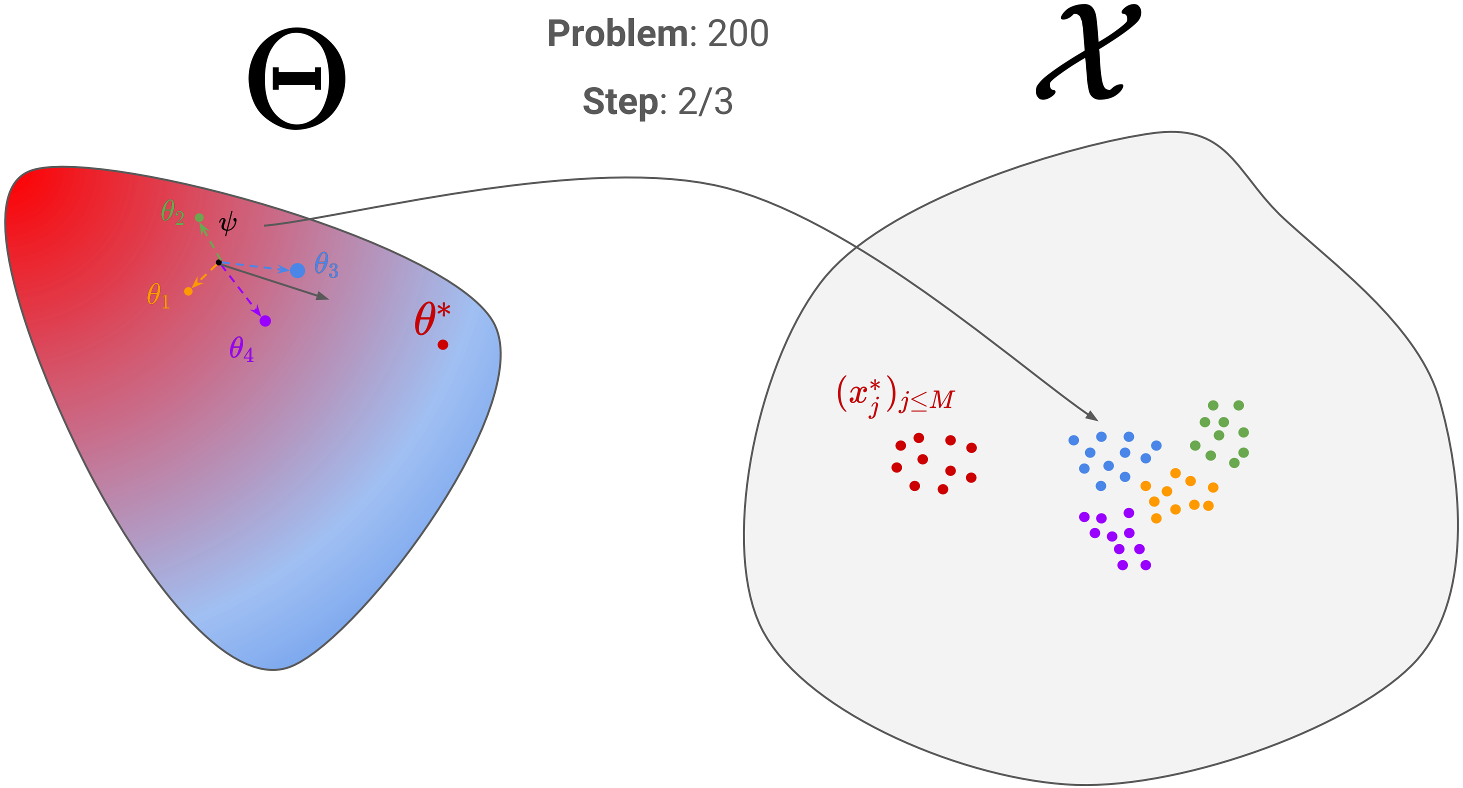


ALFI: Automatic Likelihood-free Inference

Principle: learning a descent using an Recurrent Neural Network

Problem: 200

Step: 2/3

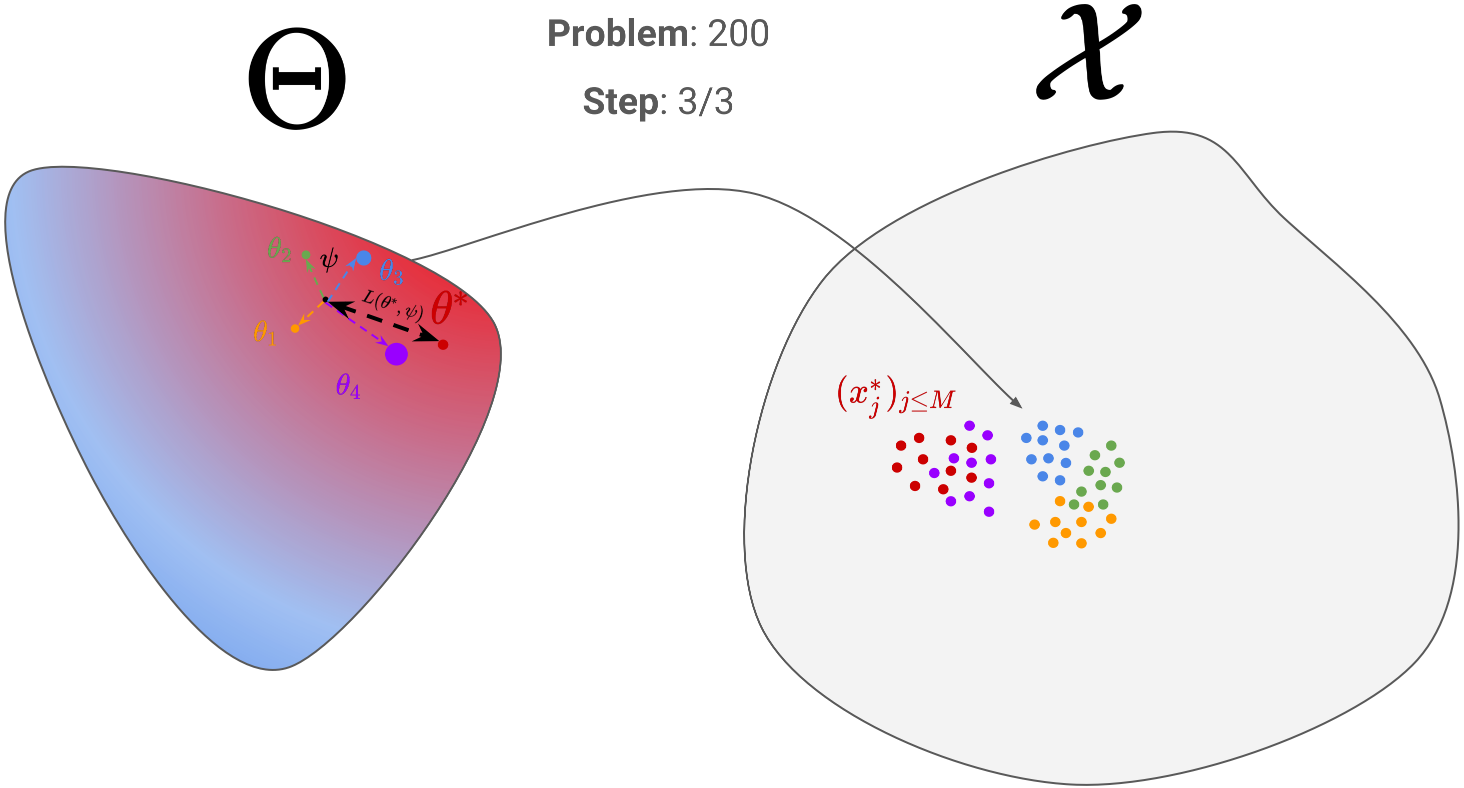


ALFI: Automatic Likelihood-free Inference

Principle: learning a descent using an Recurrent Neural Network

Problem: 200

Step: 3/3



Results

With a known likelihood function:

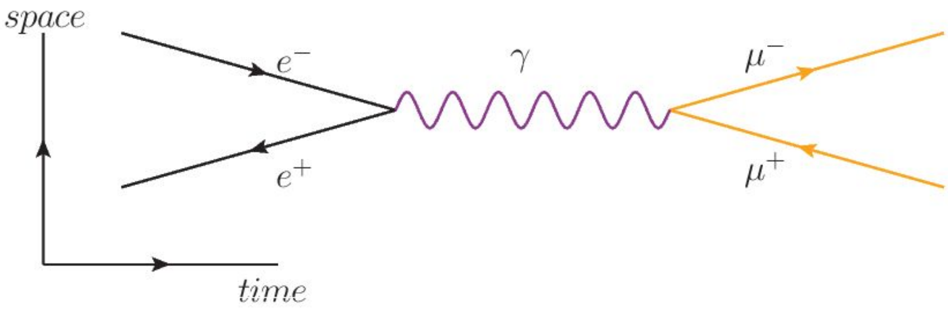
Poisson:

$$\boxed{\mathcal{X}^i \sim \mathcal{P}(\theta_i^*) \quad \text{with} \quad P_{\mathcal{P}(\theta_i^*)}(\mathcal{X} = k) = \frac{e^{-\theta_i^*}\theta_i^{*\,k}}{k!}}$$

With an unknown likelihood function:

Weinberg:

- Electron muon collision.
- Parameters are the beam energy and the fermi constant.
- Observations are the angle between the two muons.



Limitations and future work

Limitations

- Scaling it up on more complex simulators
- Learning intensive in the number of simulator calls

Future work

- Getting a better understanding of the optimization procedure learned by ALFI:
  - Is it comparable to other existing methods?
  - Does it generalize to other simulators?
- Training the model on more complex simulators and compare it to state-of-the-art likelihood-free inference methods

Conclusion

Thanks

ArXiv: 1811.12932 - Recurrent machines for likelihood-free inference

GitHub: github.com/artix41/ALFI-pytorch