Probabilistic and Differentiable Programming for Scientific Discovery

Atılım Güneş Baydin
gunes@robots.ox.ac.uk

ML Nosh seminar
10 Mar 2025

Key collaborators:

Oxford *AI for Science* Lab

The Oxford AI for Science lab is a part of the **Department of Computer Science** at the **University of Oxford**, and is led by **Atılım Güneş Baydin**.

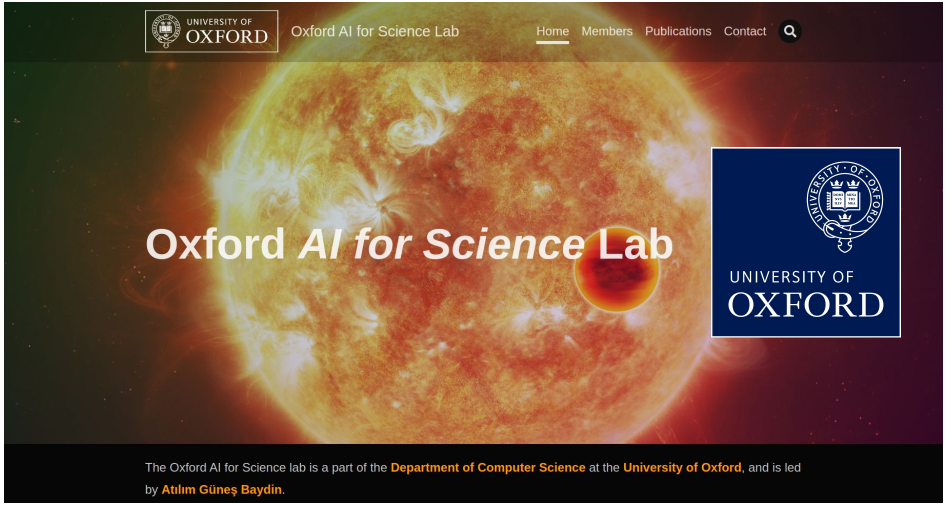
Atılım Güneş Baydin

Lecturer in Department of Computer Science and Jesus College

Oxford *AI for Science* Lab

https://oxai4science.github.io

- Specializing in **probabilistic machine learning and scientific discovery**
- Working with experts in high-energy physics, heliophysics, astrobiology, Earth science, space safety and other disciplines
- Solve challenging problems through application and development of AI methods



Funding:

NASA UK SPACE AGENCY BERKELEY LAB Microsoft Google NVIDIA

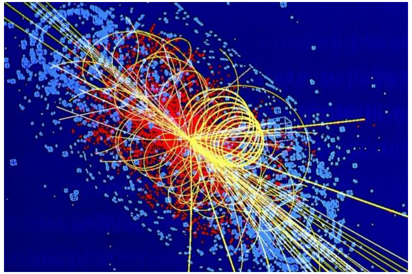
Outline

- **Probabilistic programming** and (scientific) simulators
  - Etalumis: existing simulators as probabilistic programs
  - Surrogates: replacing the simulator entirely
- **Differentiable programming** and simulators
  - When autodiff is not feasible
  - When autodiff is feasible
- Events and community

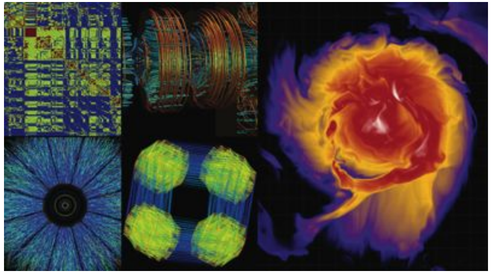
Probabilistic programming and scientific simulators

Simulation and physical sciences

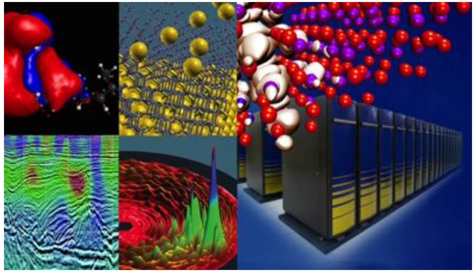
Computational models and simulation are key to scientific advance at all scales



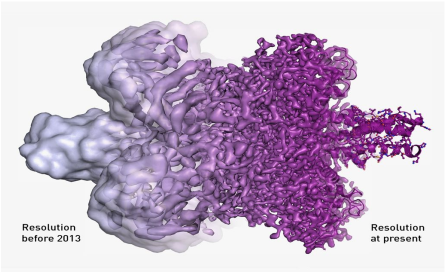
Particle physics



Nuclear physics

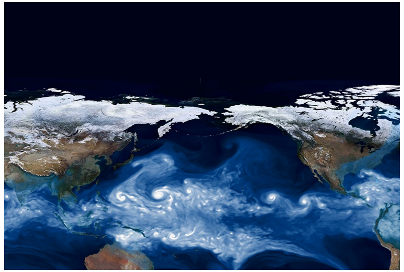


Material design

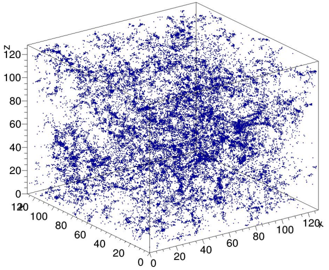


Drug discovery

Weather

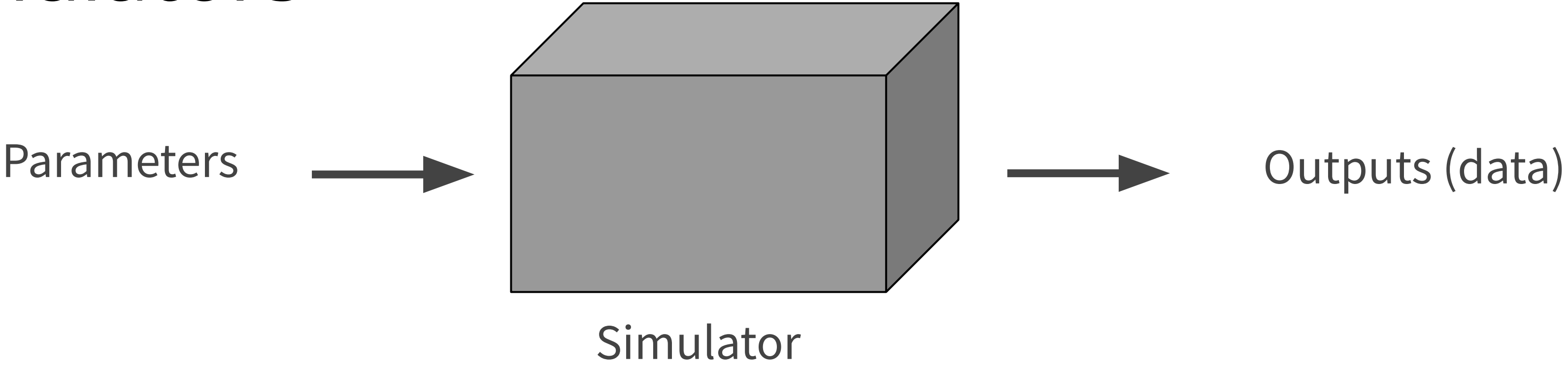


Climate science



Cosmology

Simulators



Prediction:

- Simulate forward evolution of the system
- Generate samples of output

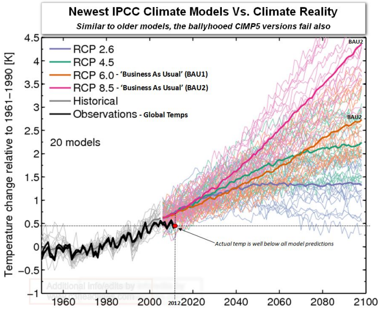
Simulators

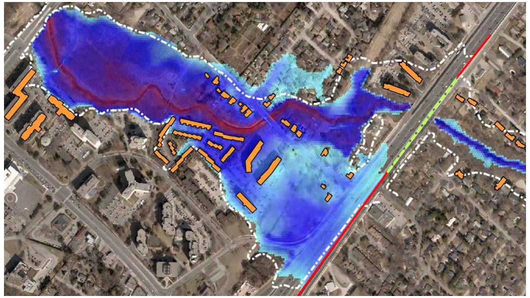
Parameters →  → Outputs (data)

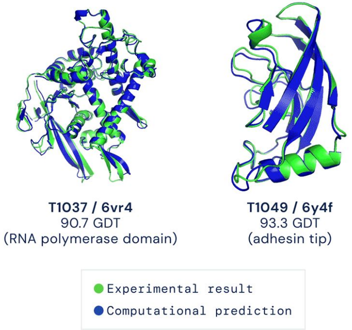
Simulator

Prediction:

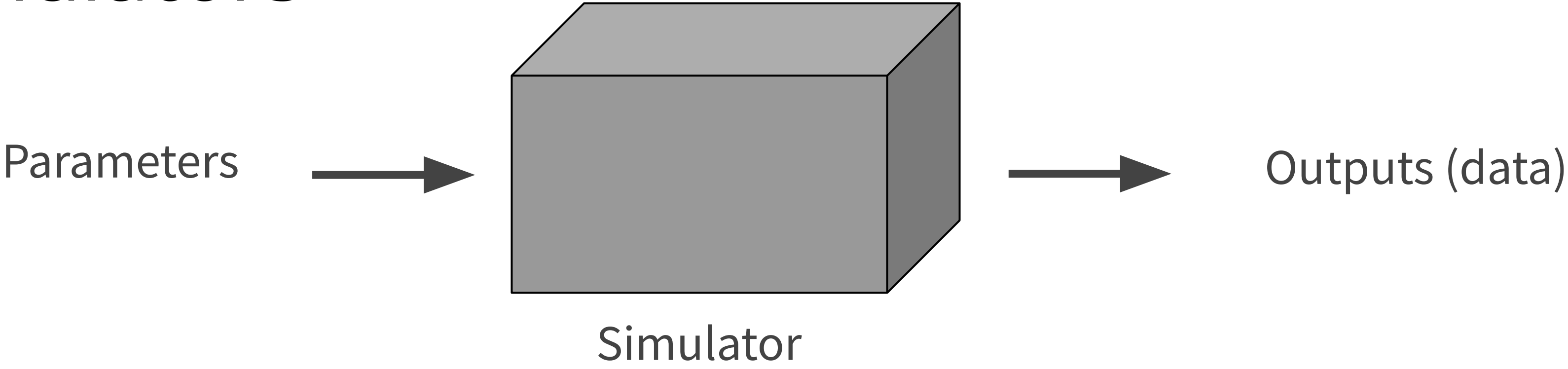
- Simulate forward evolution of the system
- Generate samples of output







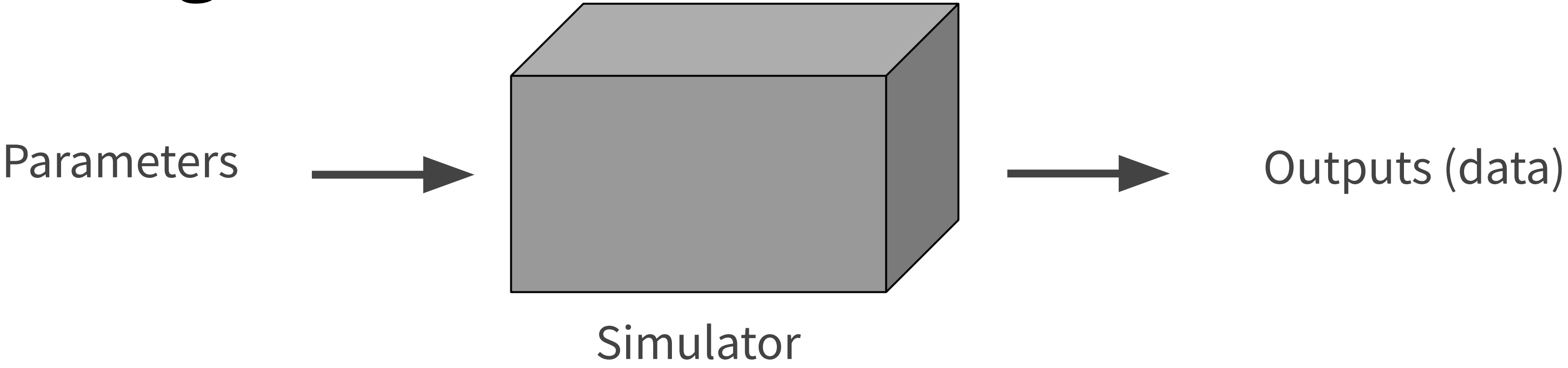
Simulators



Prediction:
- Simulate forward evolution of the system
- Generate samples of output

WE NEED THE INVERSE!

Inverting simulators



Prediction:

- Simulate forward evolution of the system
- Generate samples of output

Inference:

- Find parameters that can produce (explain) observed data
- Inverse problem
- Often a manual process

Inverting simulators

Parameters

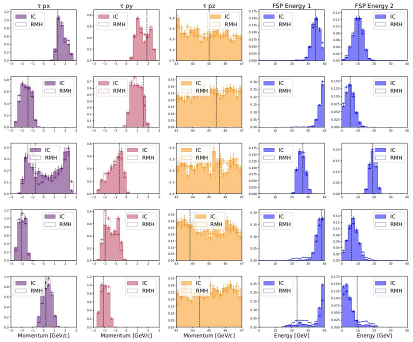
Outputs (data)

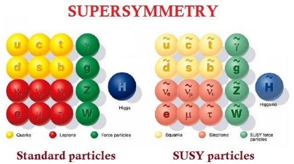
Simulator

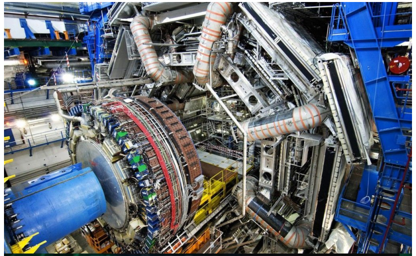
Inferred parameters

Observed data

Event analyses & new particle discoveries

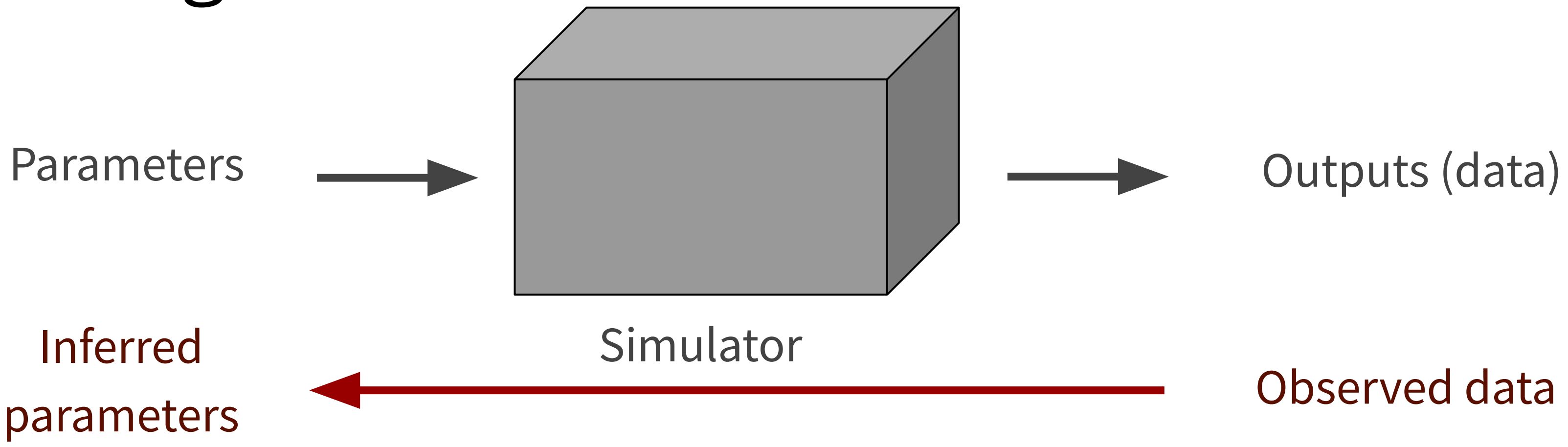






Particle detector readings

Inverting simulators




```
bool c1, c2;
c1 = Bernoulli(0.5);
c2 = Bernoulli(0.5);
observe(c1 || c2);
return(c1, c2);
```

Probabilistic programming is the perfect tool for this setting

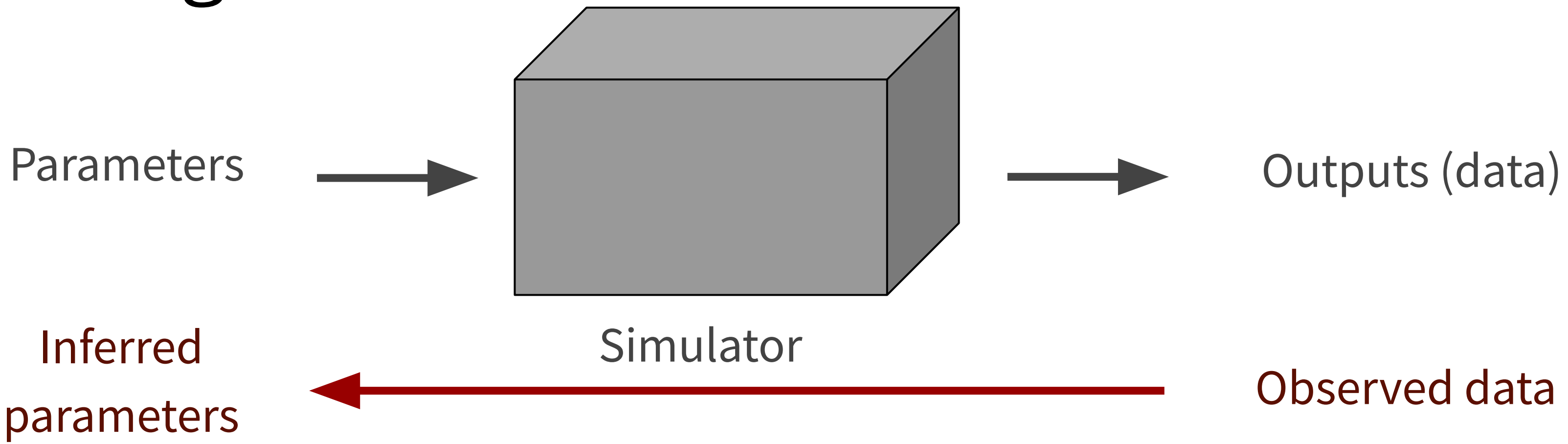
- define **generative model p(x, z) = p(x|z)p(z)**
- run automated Bayesian **inference of latent variables z conditioned on observed data x**

likelihood prior

$$\underbrace{p(z|x)}_{\text{posterior}} = \frac{p(x|z)\,p(z)}{\underbrace{p(x)}_{\text{evidence}}}$$

Pyro Edward Stan PyMC

Inverting simulators




```
bool c1, c2;
c1 = Bernoulli(0.5);
c2 = Bernoulli(0.5);
e(c1 || c2);
(c1, c2);
```

Probabilistic programming is the perfect tool for this setting

- define **ge**
- run auto **conditio**

- Somewhat limited to **small-scale problems**
- Normally requires one to **implement a probabilistic model from scratch** in the chosen language/system

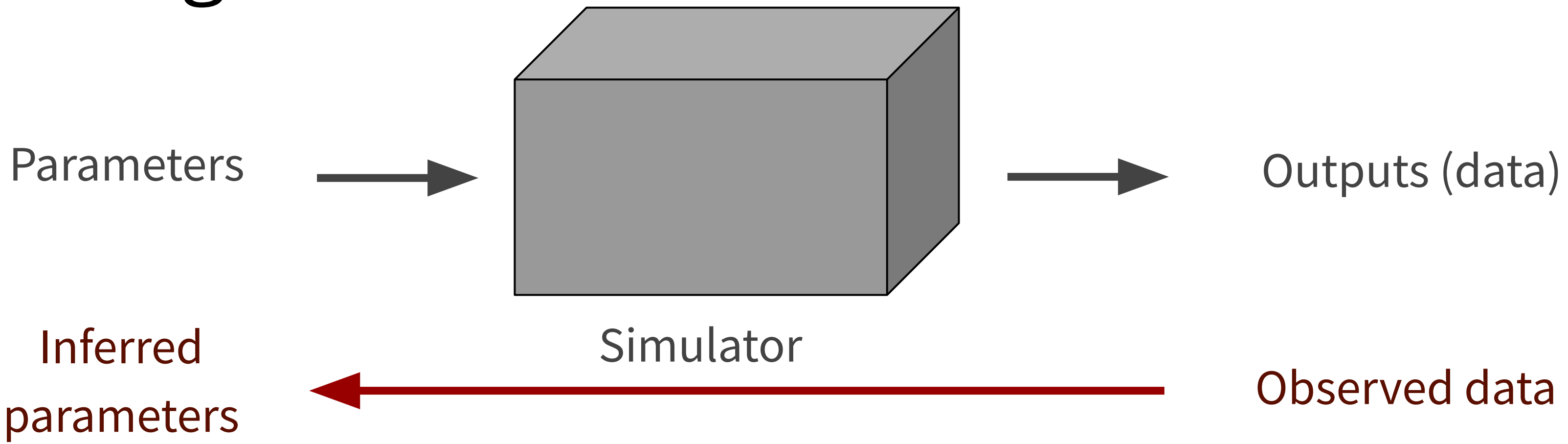
prior

$$\frac{p(x|z)\,p(z)}{p(x)}$$

posterior evidence

Pyro Edward Stan PyMC

Inverting simulators

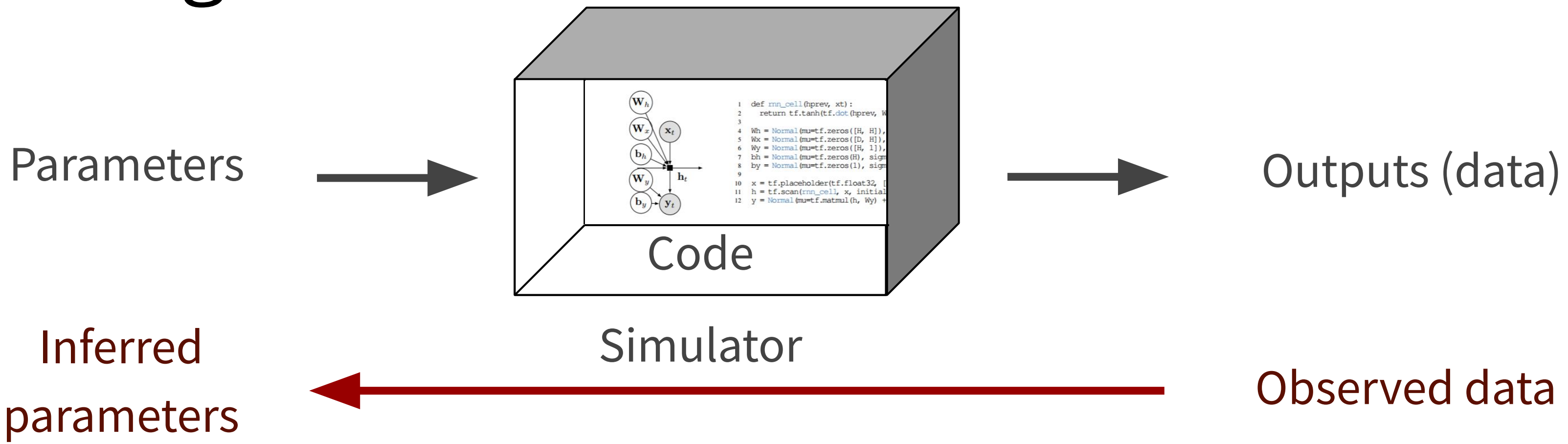


Key idea:

Many simulators are stochastic and they define probabilistic models by sampling random numbers

Baydin et al. “Efficient Probabilistic Inference in the Quest for Physics Beyond the Standard Model” **NeurIPS 2019**

Inverting simulators

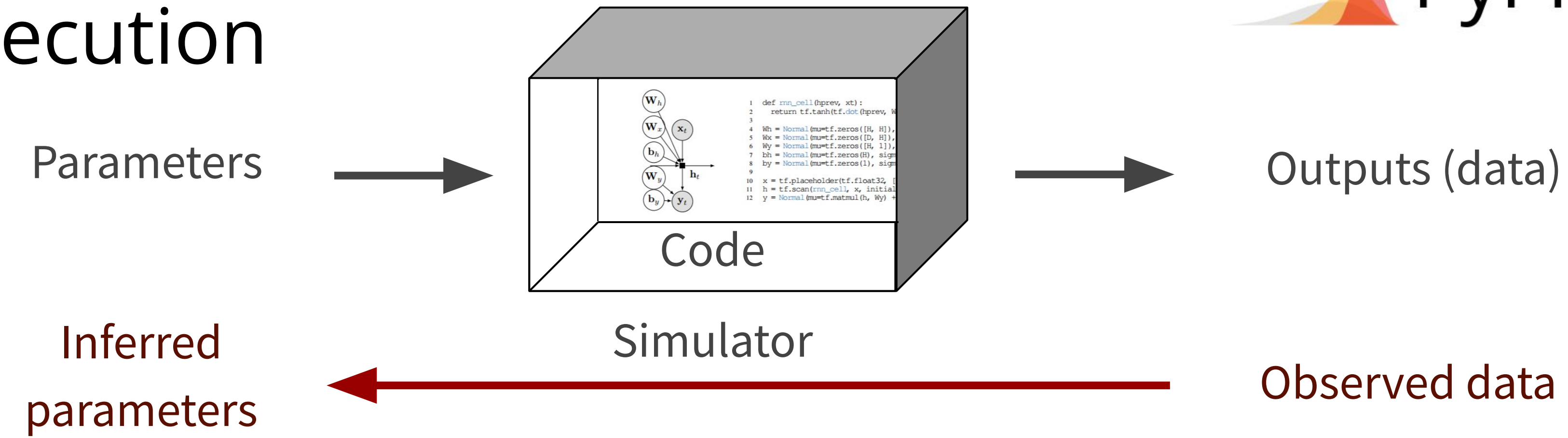


Key idea:
Many simulators are stochastic and they define probabilistic models by sampling random numbers

Simulators *are* probabilistic programs!
If we develop necessary techniques to execute them probabilistically

Probabilistic execution

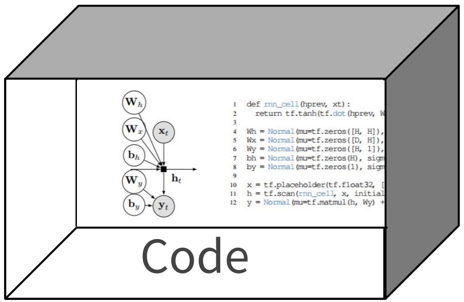
PyProb



Probabilistic execution

PyProb

Parameters



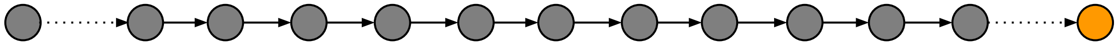
Outputs (data)

Simulator

Inferred parameters

Observed data

- **Run forward & catch all random choices** (“hijack” all calls to RNG)
- Record an **execution trace**: a record of all parameters, random choices, outputs



PPX

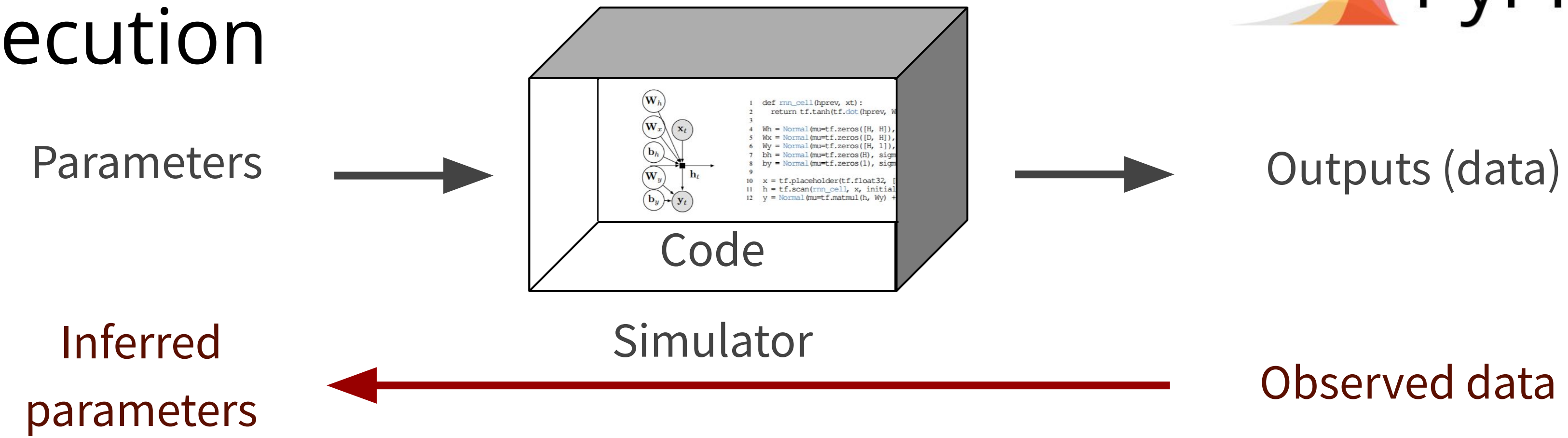
Probabilistic **P**rogramming e**X**ecution protocol

C++, C#, Dart, Go, Java, JavaScript, Lua, Python, Rust and others

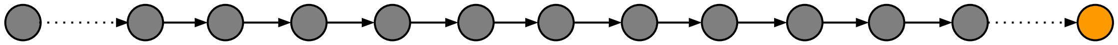
Inspired by the Open Neural Network Exchange ONNX

Probabilistic execution

PyProb



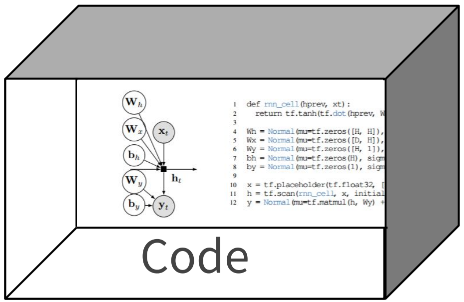
- **Run forward & catch all random choices** (“hijack” all calls to RNG)
- Record an **execution trace**: a record of all parameters, random choices, outputs



  - Conditioning: compare ***simulated output*** and ***observed data***
- **Approximate the distribution of parameters** that can produce (explain) observed data, using inference engines like Markov-chain Monte Carlo (MCMC)

Probabilistic execution

PyProb

Parameters → 

→ Outputs (data)

Simulator

Inferred parameters ← Observed data

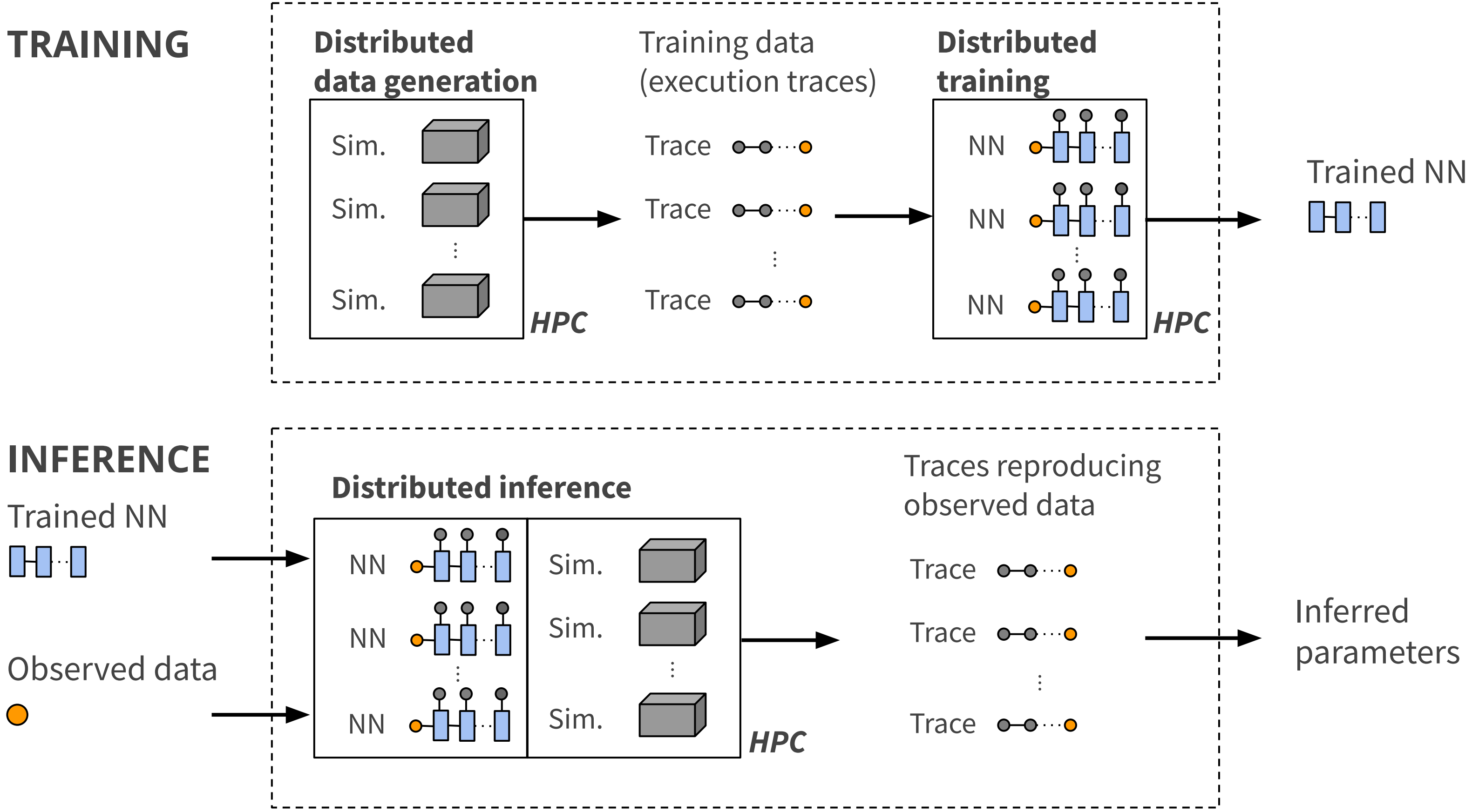
- Ru
- Re ... outputs
  - ○
- Ap ... n)
  ob ... MCMC)

Simulators = giant probabilistic models so inference is hard and computationally costly

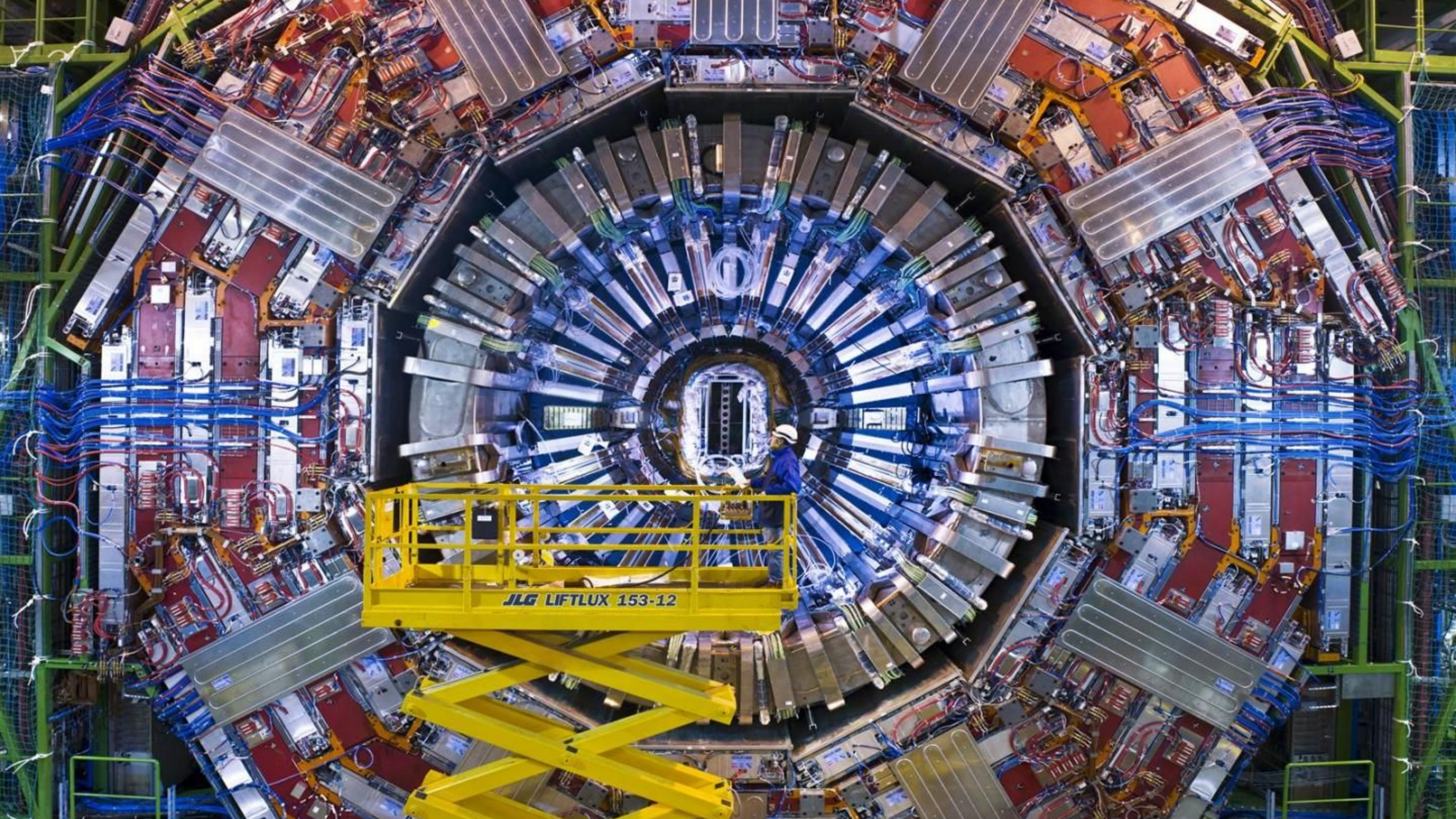
- Need to run simulator up to millions of times
- Simulator execution and MCMC inference are sequential
- MCMC has "burn-in period" and autocorrelation

But we can amortize the cost of inference using deep learning

Amortized inference



Baydin et al. "Efficient Probabilistic Inference in the Quest for Physics Beyond the Standard Model" **NeurIPS 2019**


JLG LIFTLUX 153-12

“etalumis”
simulate

- Developed techniques that led to the **largest scale inference** in Turing-complete probabilistic programing, approx. 25,000 latents in Sherpa, 1M lines of C++ code, 32,768 CPU cores
- Largest-scale PyTorch MPI (128k minibatch size)
- **First tractable Bayesian inference for LHC physics**

Cori supercomputer, Lawrence Berkeley Lab
2,388 Haswell nodes (32 cores per node)
9,688 KNL nodes (68 cores per node)

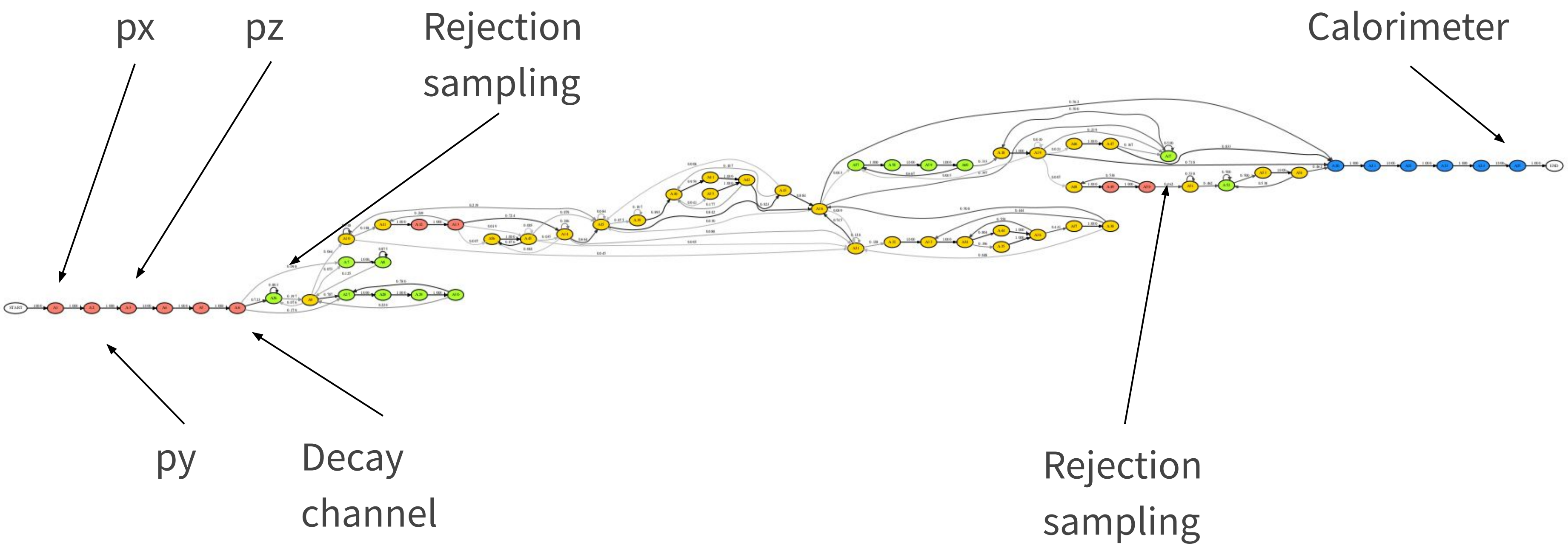
Best Paper Finalist at SC19, top international supercomputing venue

Funding: DOE/Lawrence Berkeley Lab

Baydin et al. “Etalumis: Bringing Probabilistic Programming to Scientific Simulators at Scale” **SC19**

Interpretability

Develop techniques to inspect probabilistic structure of simulators



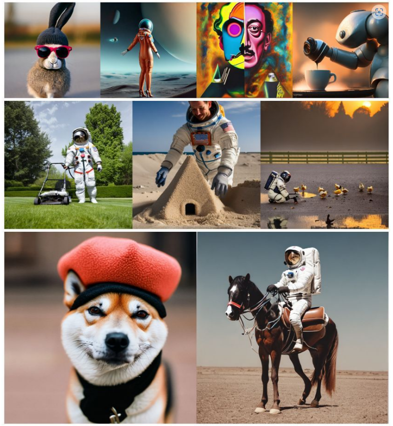
Latent probabilistic structure (**250** most frequent traces) of the Standard Model in Sherpa

Differentiable programming and simulators

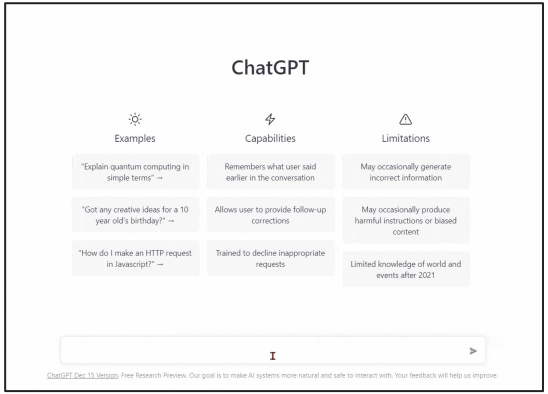
What is differentiable programming?

Deep learning (neural networks) has been at the core of recent advances in machine learning and artificial intelligence

Tesla Autopilot (2020)



Stable Diffusion 2 (2022)



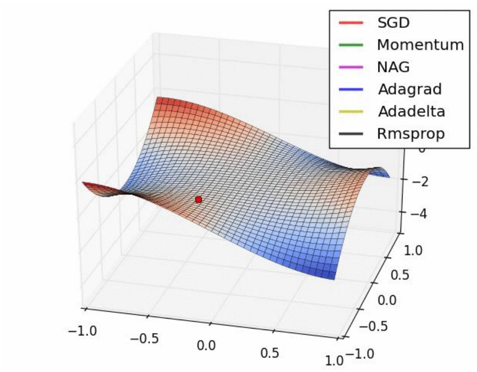
OpenAI ChatGPT (2022)

Baydin, Pearlmutter, Radul, Siskind, 2018. Automatic differentiation in machine learning: a survey. ***Journal of Machine Learning Research (JMLR)*** http://jmlr.org/papers/v18/17-468.html

What is differentiable programming?

A generalization of deep learning to arbitrary programs

- Neural networks = nonlinear **differentiable functions (programs)** whose parameters we tune by **gradient-based optimization**
- We get derivatives by running the code via **automatic differentiation** (mainly backpropagation / reverse mode)



Ruder (2017)

$$f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$$

automatic differentiation

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right)$$

theano

torch

PyTorch

TensorFlow

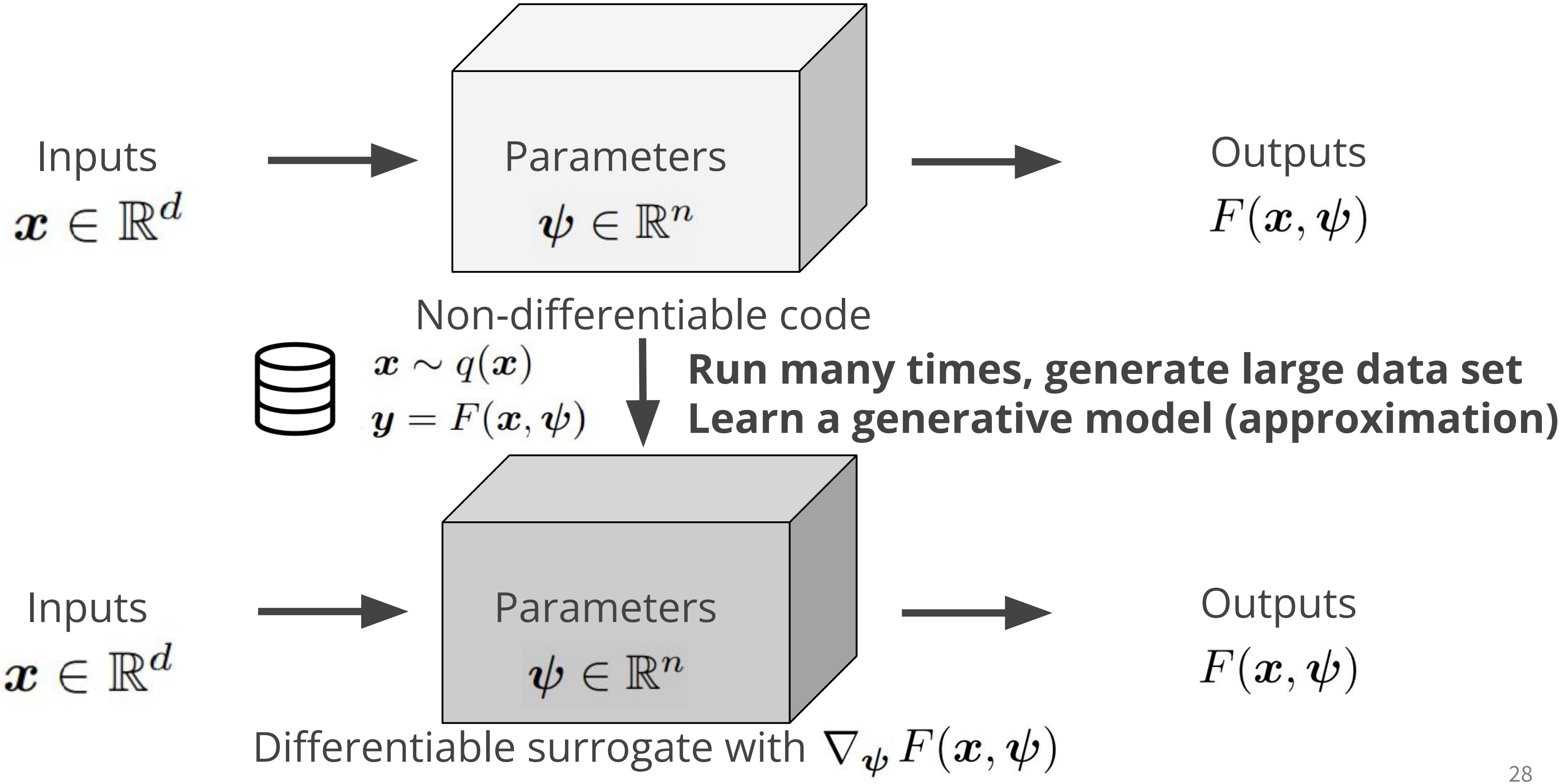
Baydin, Pearlmutter, Radul, Siskind, 2018. Automatic differentiation in machine learning: a survey. ***Journal of Machine Learning Research (JMLR)*** http://jmlr.org/papers/v18/17-468.html

Simulators and differentiability

- Simulator code is **not differentiable**
  □ Use surrogates (differentiable approximation learned from data)

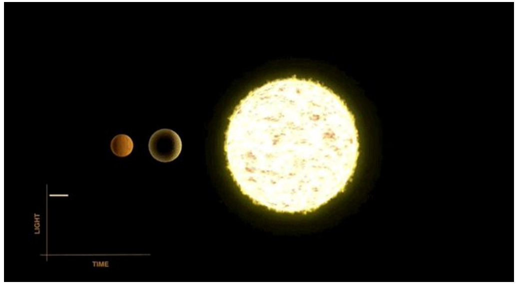
- Simulator code is **differentiable**
  (but has not been used in a differentiable way so far)
  □ Use automatic differentiation if feasible

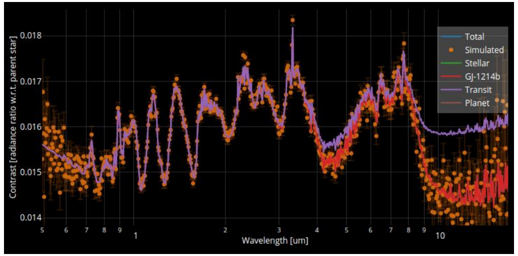
Non-differentiable simulator

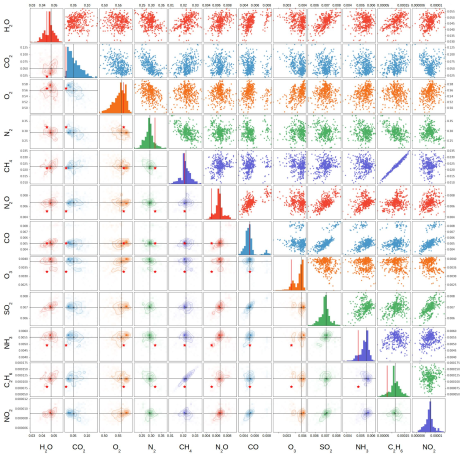


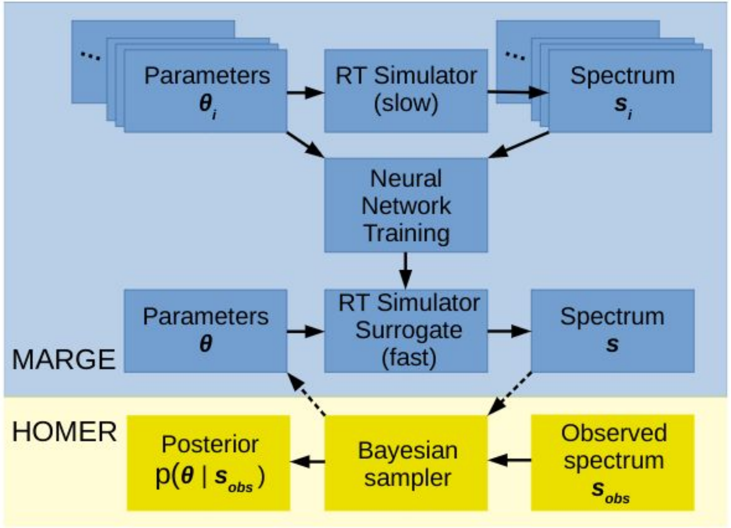
Example: Exoplanet radiative transfer

- **Posterior distributions of gas concentrations** in exoplanet atmospheres, conditioned on observed spectra, using radiative transfer simulators
- Surrogates allow **up to 180× faster** inference





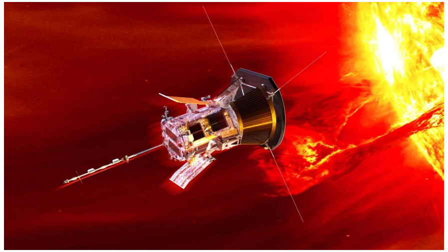




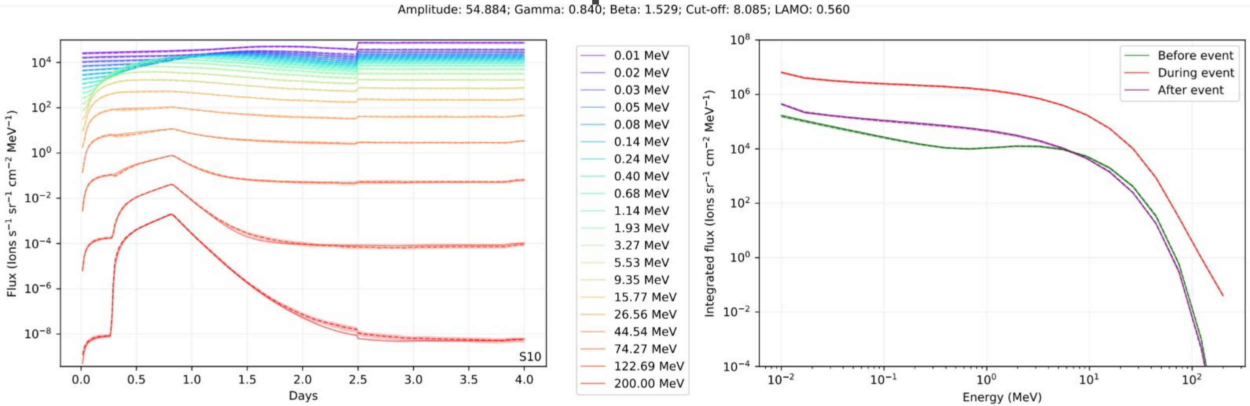
*Himes, Harrington, Cobb, **Baydin**, Soboczenski, O'Beirne, Zorzan, Wright, Scheffer, Domagal-Goldman, Arney. 2022. "**Accurate Machine-Learning Atmospheric Retrieval via a Neural-Network Surrogate Model for Radiative Transfer**" The **Planetary Science Journal** 3 (4). American Astronomical Society: 236–250. doi:10.3847/PSJ/abe3fd.*

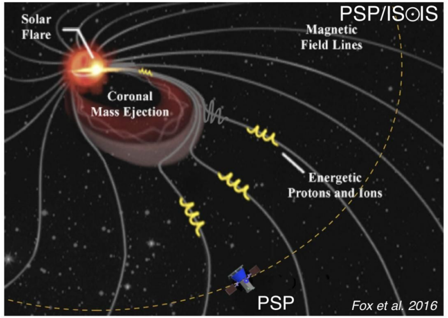
Example: Solar energetic particles

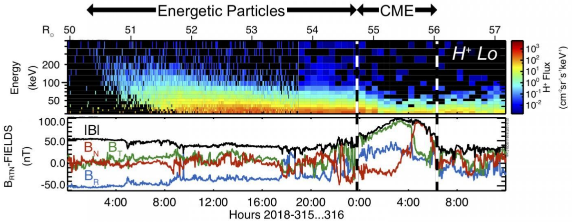
- Solar energetic particles (SEPs) pose threats to
  - Humans in deep space exploration
  - Scientific instruments onboard spacecraft
- Developed an EPREM **simulator surrogate** (**10^9× faster**) enabling **posterior inference conditioned on real space weather events**



Parker Solar Probe (NASA)







*Poduval, **Baydin**, Schwadron. 2021. “Studying Solar Energetic Particles and Their Seed Population Using Surrogate Models” In Machine Learning for Space Sciences Workshop, 43rd Committee on Space Research (COSPAR) Scientific Assembly, Sydney, Australia.*

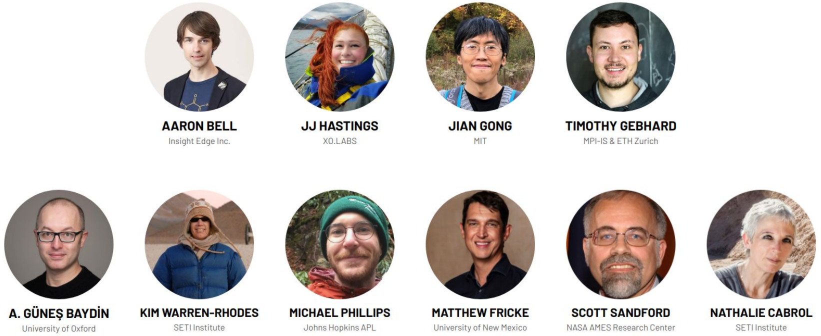
***Baydin**, Poduval, Schwadron. 2023. “A Surrogate Model For Studying Solar Energetic Particle Transport and the Seed Population.” Space Weather 21 (12). American Geophysical Union*

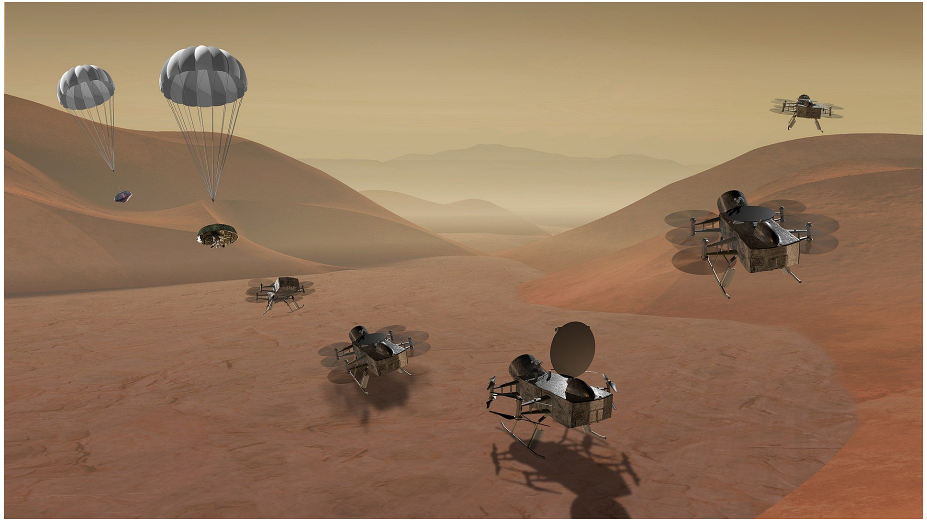
Example: Astrobiology

Dragonfly: NASA mission to Saturn/Titan (launch planned June 2027)

Robotic rotorcraft (VTOL) with Dragonfly Mass Spectrometer (DraMS) and other instruments

Machine learning methods for detection of "life" using molecular complexity as a biosignature





https://www.nasa.gov/dragonfly

Example: Astrobiology

Google NASA

Created a dataset of 400,000+ molecules
(17,000+ with mass spectra)

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY U.S. DEPARTMENT OF COMMERCE · PubChem · ChEMBL · NIH NATIONAL CANCER INSTITUTE · Reaxys®

Machine learning methods for complexity prediction:

“Molecule → complexity” prediction
extreme speed-up ~ 1.04B times faster than the classic algorithm (Go)

“Mass spectrum → complexity” prediction
cheap enough to be deployed onboard

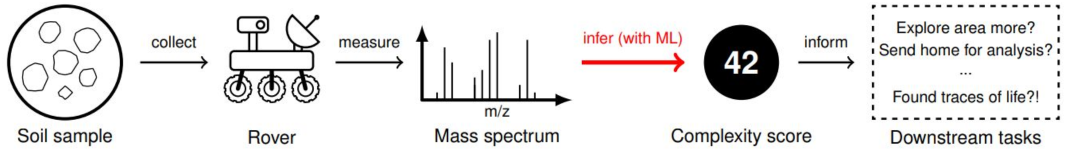
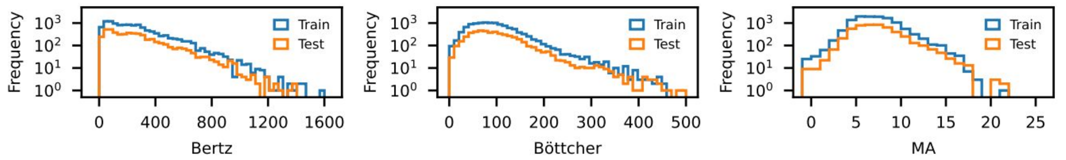


Figure 1: A schematic workflow that illustrates where our work on inferring MC from mass spectral data fits into the “bigger picture” of space exploration and the search for traces of extraterrestrial life.

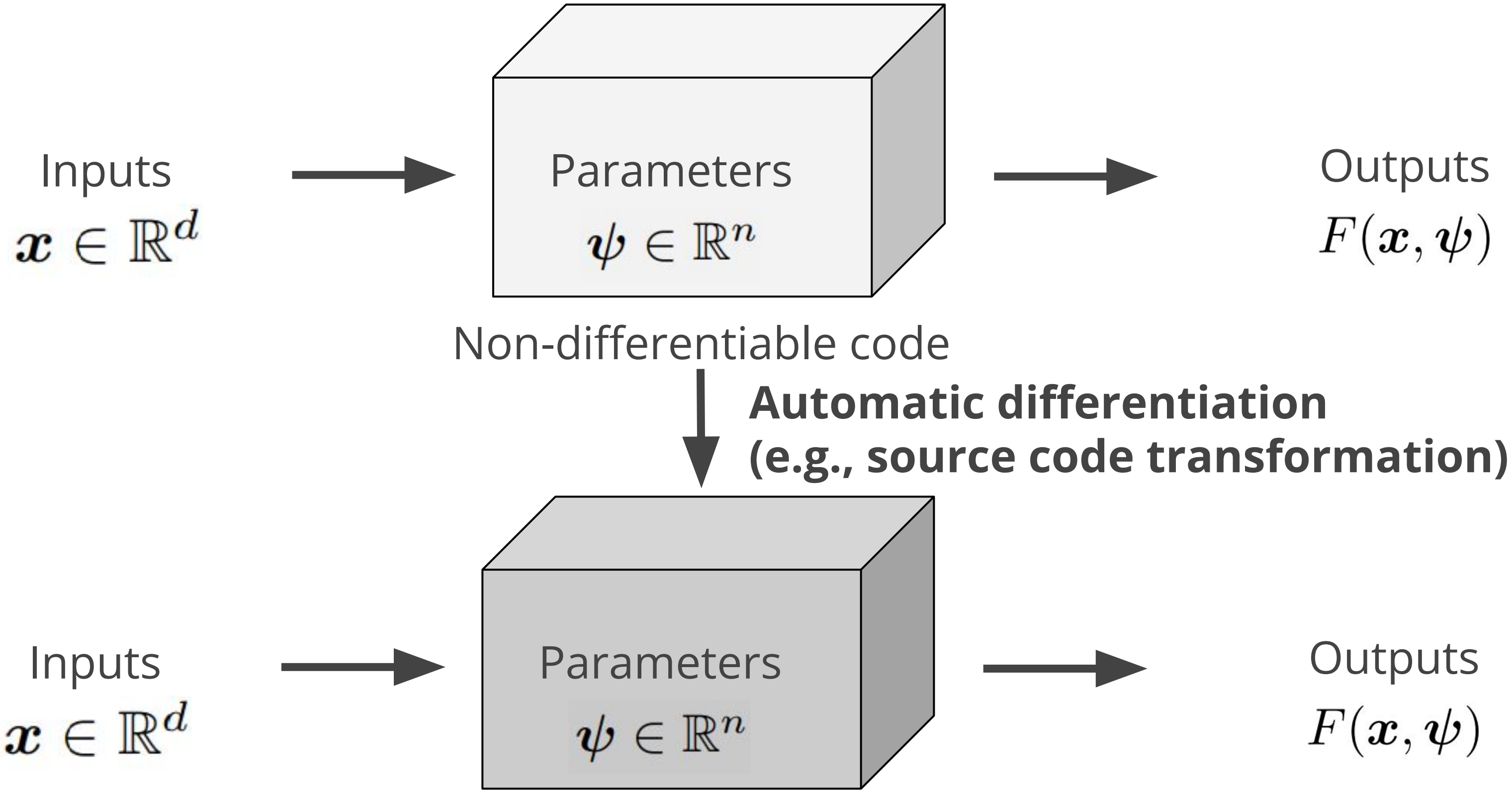


(a) Bertz (b) Böttcher (c) MA

Figure 2: Distributions of the three complexity scores in our training and evaluation (test) dataset.

Gebhard, Bell, Gong, Hastings, Fricke, Cabrol, Sandford, Phillips, Warren-Rhodes, **Baydin**. 2022. “**Inferring Molecular Complexity from Mass Spectrometry Data Using Machine Learning.**” In *Machine Learning and the Physical Sciences Workshop, NeurIPS 2022*.

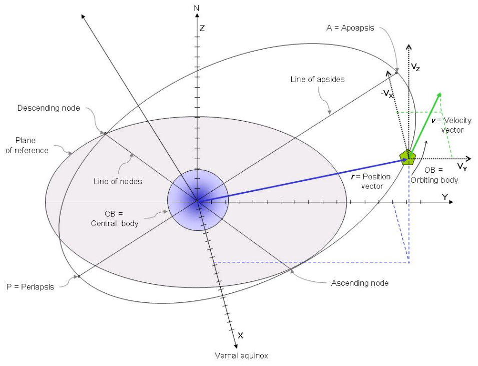
Differentiable simulator



Same simulator, but now **differentiable**! $\nabla_{\boldsymbol{\psi}} F(\boldsymbol{x}, \boldsymbol{\psi})$

Example: Differentiable Orbit Propagation

- **SGP4** propagator computes **orbital states of satellites and space debris** around the Earth
- Predicting the effect of perturbations caused by the Earth's shape, drag, radiation, gravitation effects of the Sun and the Moon
- Uses two-line elements produced by NORAD and NASA

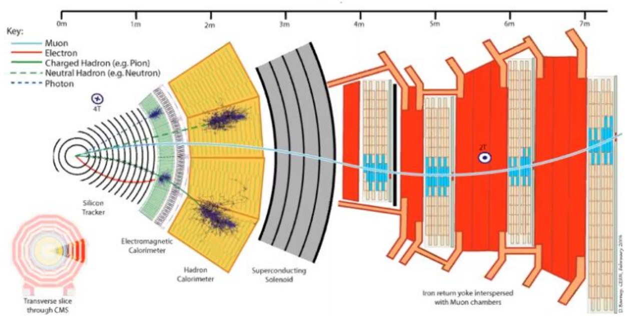


- **dSGP4** based on PyTorch
- Collaboration with CelesTrak, based on data from Space Track (US Space Force)

CELESTRAK SINCE 1985

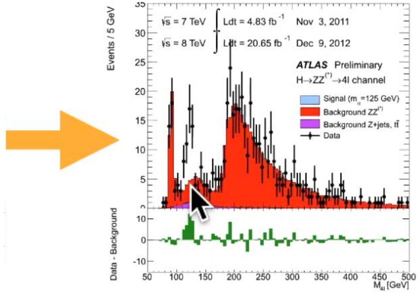
G Acciarini, AG Baydin, D Izzo. Closing the Gap Between SPG4 and High-Precision Propagation via Differentiable Programming.
https://arxiv.org/abs/2402.04830

Differentiable programming in particle physics



- **Differentiable analysis pipelines**
  Unify analysis pipelines, simultaneously optimize free parameters of analysis w.r.t. desired physics objective

- **Gradient-based inference (probabilistic programming)**
  Enable efficient simulation-based inference, reduce number of events needed by orders of magnitude



https://mode-collaboration.github.io/

AG Baydin, K Cranmer, P de Castro Manzano, C Delaere, D Derkach, J Donini, T Dorigo, A Giammanco, J Kieseler, L Layer, G Louppe, F Ratnikov, G Strong, M Tosi, A Ustyuzhanin, P Vischia, H Yarar. 2021. **"Toward Machine Learning Optimization of Experimental Design."** Nuclear Physics News 31 (1). Taylor & Francis: 25–28. doi:10.1080/10619127.2021.1881364

AG Baydin, K Cranmer, M Feickert, L Gray, L Heinrich, A Held, A Melo, M Neubauer, J Pearkes, N Simpson, N Smith, G Stark, S Thais, V Vassilev, G Watts. 2020. **"Differentiable Programming in High-Energy Physics."** In Snowmass 2021 Letters of Interest (LOI), Division of Particles and Fields (DPF), American Physical Society. https://snowmass21.org/loi

Community

Machine-learning Optimized Design of Experiments (MODE)

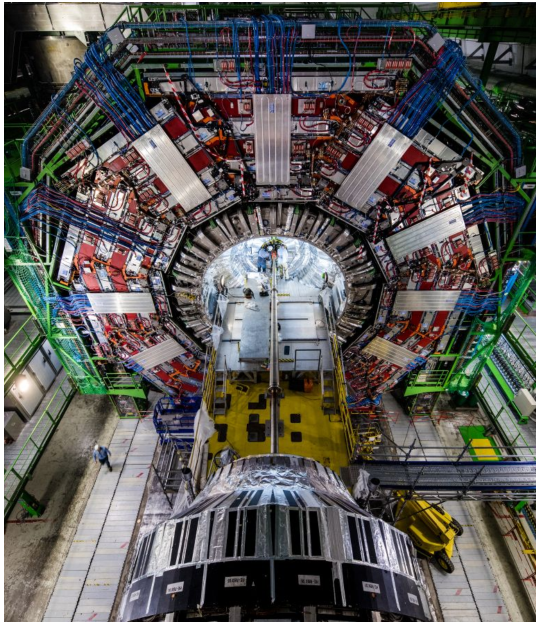
Probabilistic and differentiable programming in design optimization of **next-generation (large-scale) instruments** for particle physics and industry (CERN, Padova, UC Louvain, Oxford, NYU, Rutgers, Uppsala, TU Munich, Durham)

https://mode-collaboration.github.io/

Workshop series on **Differentiable Programming for Experiment Design**

- 8–13 Jun 2025: Crete, Greece
- 23–25 Sep 2024: Valencia, Spain
- 24–26 Jul 2023: Princeton University, US
- 12–16 Sep 2022: Orthodox Academy of Crete, Greece
- 6–8 Sep 2021: Université catholique de Louvain, Belgium

https://indico.cern.ch/event/1481852/



AG Baydin, K Cranmer, P de Castro Manzano, C Delaere, D Derkach, J Donini, T Dorigo, A Giammanco, J Kieseler, L Layer, G Louppe, F Ratnikov, G Strong, M Tosi, A Ustyuzhanin, P Vischia, H Yarar. 2021. **"Toward Machine Learning Optimization of Experimental Design."** Nuclear Physics News 31 (1). Taylor & Francis: 25–28. doi:10.1080/10619127.2021.1881364

AG Baydin, K Cranmer, M Feickert, L Gray, L Heinrich, A Held, A Melo, M Neubauer, J Pearkes, N Simpson, N Smith, G Stark, S Thais, V Vassilev, G Watts. 2020. **"Differentiable Programming in High-Energy Physics."** In Snowmass 2021 Letters of Interest (LOI), Division of Particles and Fields (DPF), American Physical Society. https://snowmass21.org/loi

T Dorigo, A Giammanco, P Vischia, M Aehle, M Bawaj, A Boldyrev, P de Castro Manzano, D Derkach, J Donini, A Edelen, F Fanzago, NR Gauger, C Glaser, **AG Baydin**, L Heinrich, R Keidel, J Kieseler, C Krause, M Lagrange, M Lamparth, L Layer, G Maier, F Nardi, HES Pettersen, A Ramos, F Ratnikov, D Röhrich, R Ruiz de Austri, P Martínez Ruiz del Árbol, O Savchenko, N Simpson, GC Strong, A Taliercio, M Tosi, A Ustyuzhanin, H Zaraket. 2022. "**Toward the End-to-End Optimization of Particle Physics Instruments with Differentiable Programming: a White Paper.**" https://arxiv.org/abs/2203.13818

MIAPbP program

Munich Institute for Astro-, Particle and BioPhysics,
Technical University of Munich

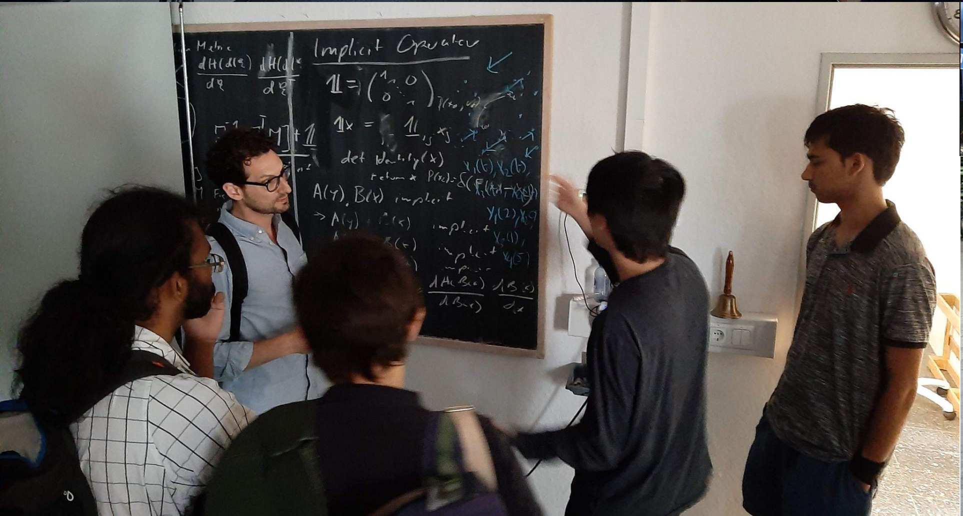
Max Planck Institute for Extraterrestrial Physics

Month-long program in **Differentiable and Probabilistic Programming for Fundamental Physics**

- Organizers: Lukas Heinrich, Torsten Enßlin, Michael Kagan, Atılım Güneş Baydin, Vassil Vassilev
- Bringing together **probabilistic programming** and **fundamental physics** communities
- Hosted in Munich during 5–30 Jun 2023

https://www.munich-iapbp.de/probabilistic-programming


MIAPbP Differentiable &
Probabilistic Programming
June 2023


Implicit Operator


HOFBRÄU MÜNCHEN

Frontier Development Lab

https://fdl.ai

- A research accelerator for state-of-the-art ML and space sciences
- Two main versions
  - NASA Ames & SETI Institute (FDL US)
  - ESA & University of Oxford (FDL Europe)
- Access to compute provided by industry (Google, Intel, Nvidia and others)
- Teams of
  - PhD students / postdocs (two machine learning, two domain science)
  - supervising faculty


NASA
Welcome to
NASA in Silicon Valley
Ames Research Center


esa
Buildings 1 - 2 - 3 - 4 - 5 - 12 - 14
Buildings 6 - 7 (Magazzino - Store)
Building 8
Buildings 9 - 10 - 11
Reception
Parking
Building 15 Child Care - Sport Facilities


Heliophysics System Observatory
Challenges:
telemetry/data storage
mission cost
unutilized data


Tests and Experiments
FDL

ML and the Physical Sciences

Machine Learning and the Physical Sciences workshop

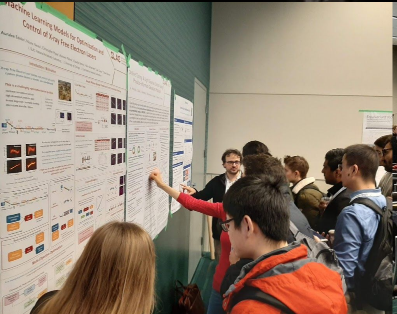
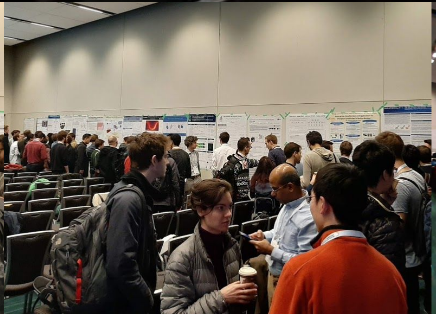
Conference on Neural Information Processing Systems (NeurIPS)
2017, 2019, 2020, 2021, 2022, 2023, 2024

- One of the largest NeurIPS workshops
- > 200 papers and 250 reviewers in 2024
- Cutting-edge research on ML and physical sciences

https://ml4physicalsciences.github.io/

Please consider submitting your work!

Funding: DeepMind, Nvidia, Intel, Cray, Moore Foundation, Vector Institute



Emine Kucukbenli
Harvard University / Boston University


Atılım Güneş Baydin
University of Oxford


Adji Bousso Dieng
Princeton University


Gilles Louppe
University of Liège


Savannah Thais
Princeton University / IRIS-HEP


Brian Nord
Fermilab


Benjamin Nachman
Lawrence Berkeley National Laboratory


Siddharth Mishra-Sharma
IAIFI / MIT / Harvard


Rianne van den Berg
Microsoft Research, Amsterdam


Kyle Cranmer
University of Wisconsin / Meta AI


Anima Anandkumar
Caltech / NVIDIA


Lenka Zdeborová
EPFL

Thank you for listening

Questions?

Extra slides ahead!