How machine learning can help us get the most out of high-precision physics models

Johann Brehmer
New York University

Bosch Center for AI

August 25, 2020

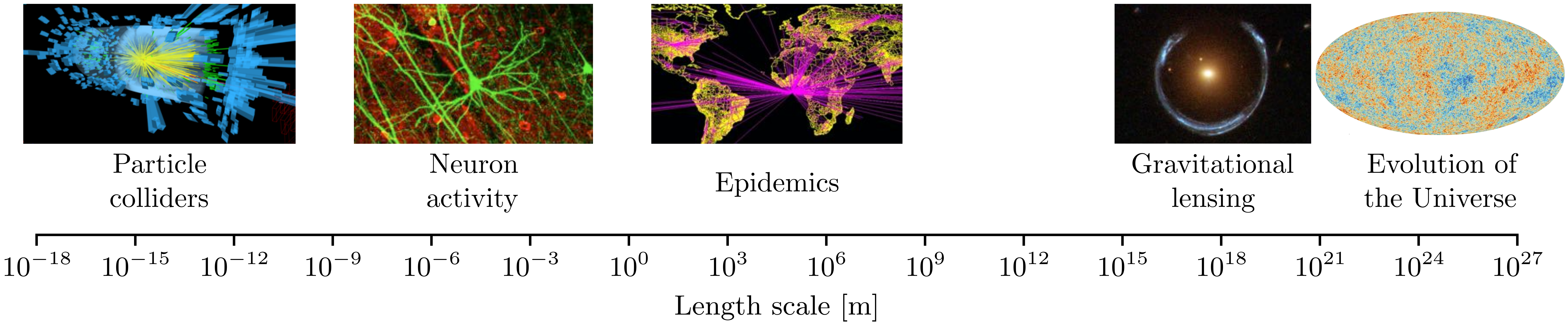

[CMS]

[Hubble Space Telescope]

[Nextstrain, CDC]

Big picture

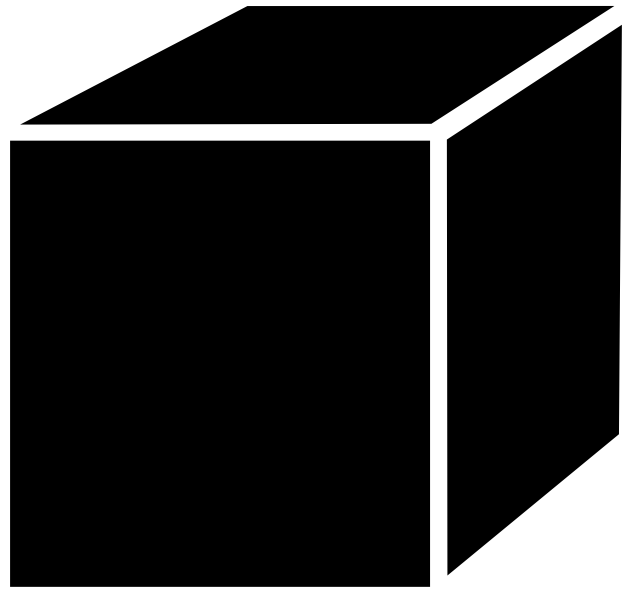
[K. Cranmer, JB, G. Louppe 1911.01429]



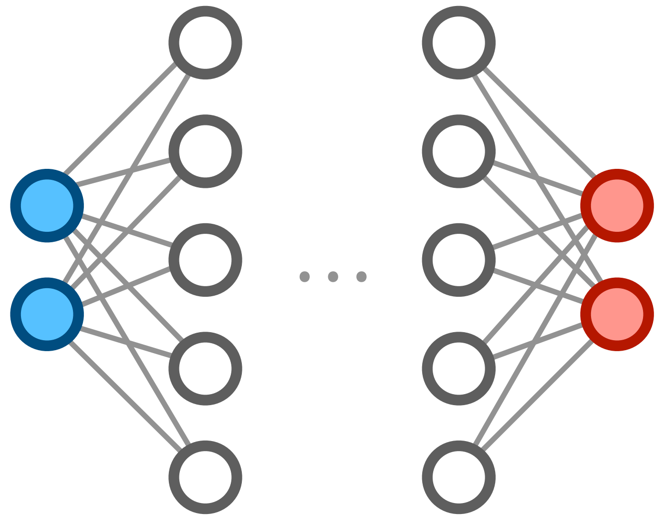
Simulators give high-precision predictions for many phenomena in science and engineering.

Unfortunately, they are poorly suited for inference.

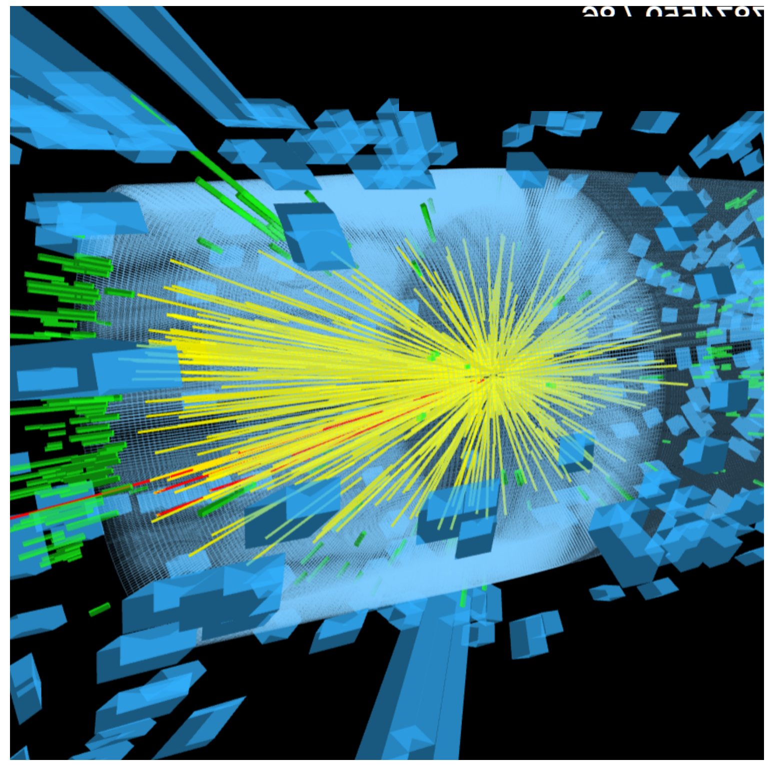
Machine learning can provide efficient models for powerful inference algorithms.



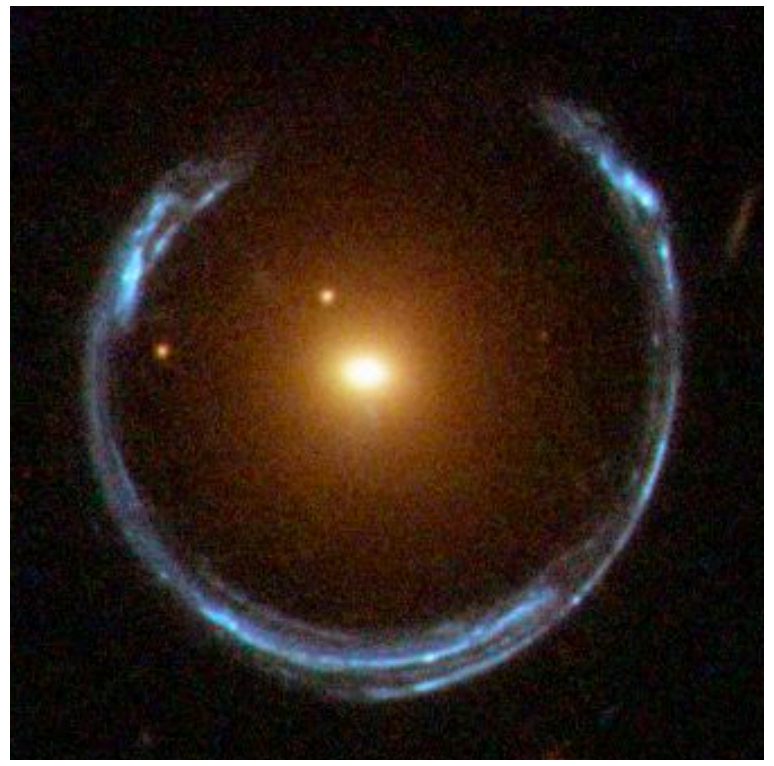
1. The simulation-based inference problem



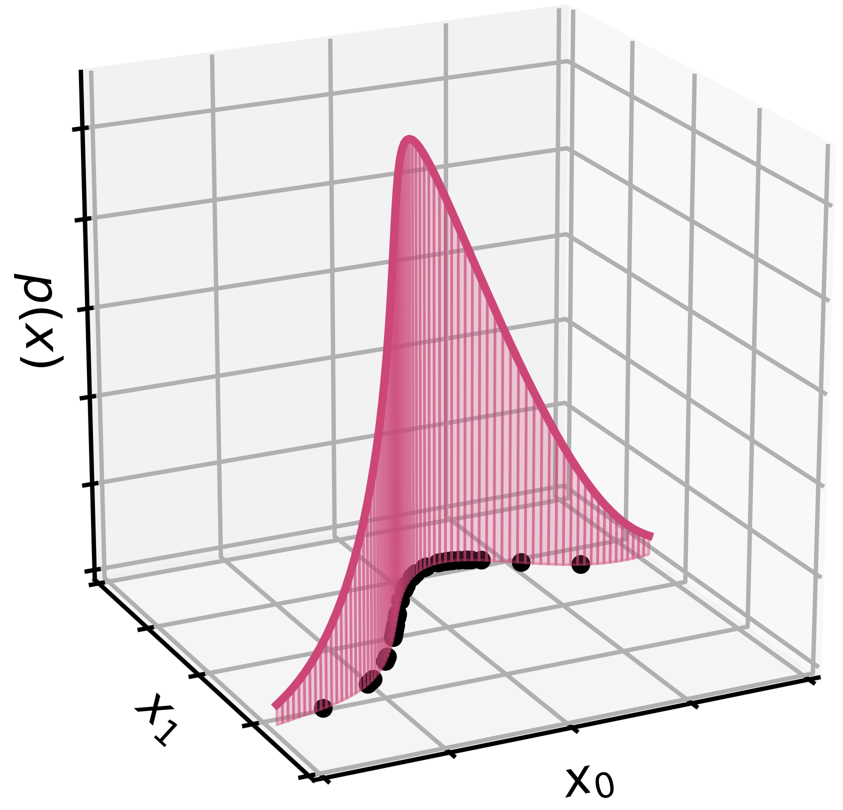
2. Machine learning methods



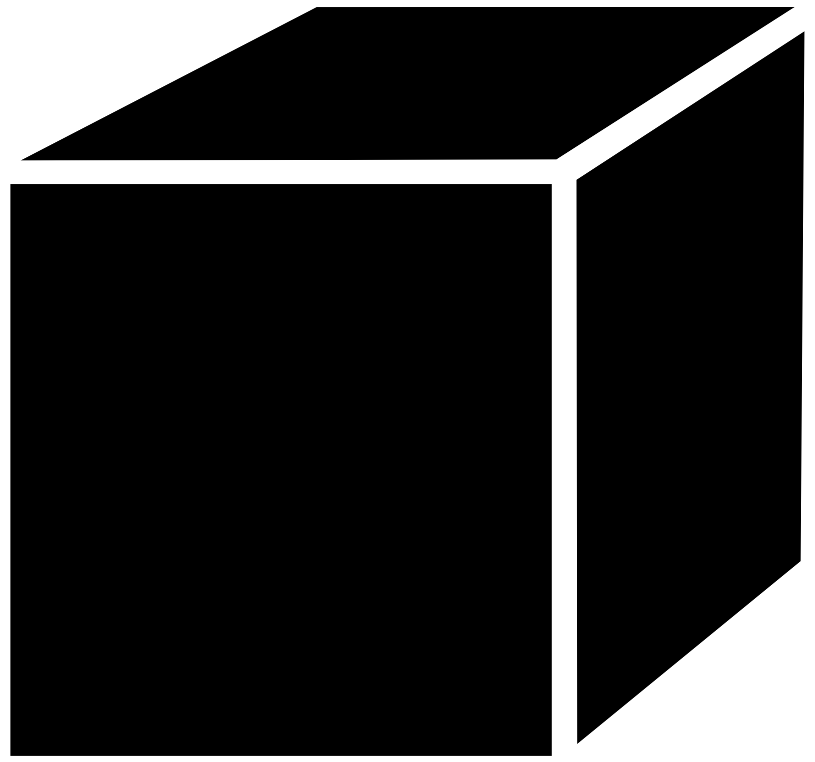
3. Particle physics



4. Astrophysics

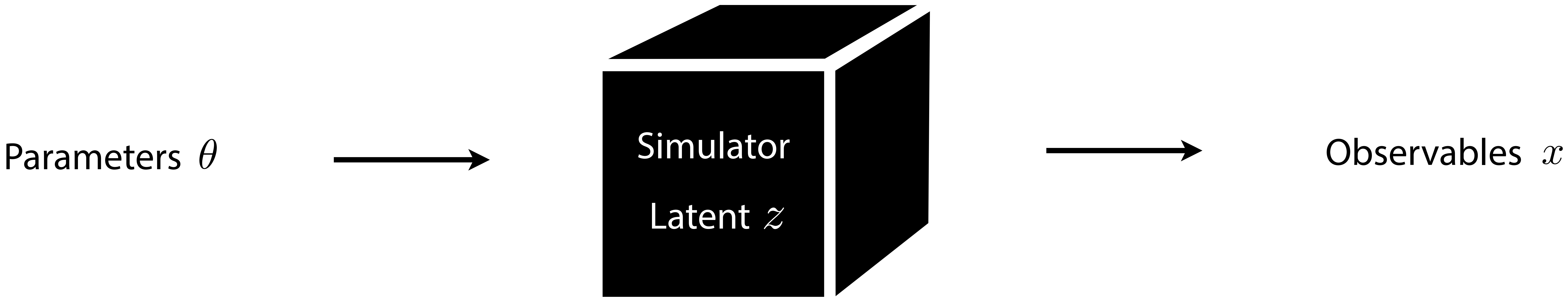


5. Tangents



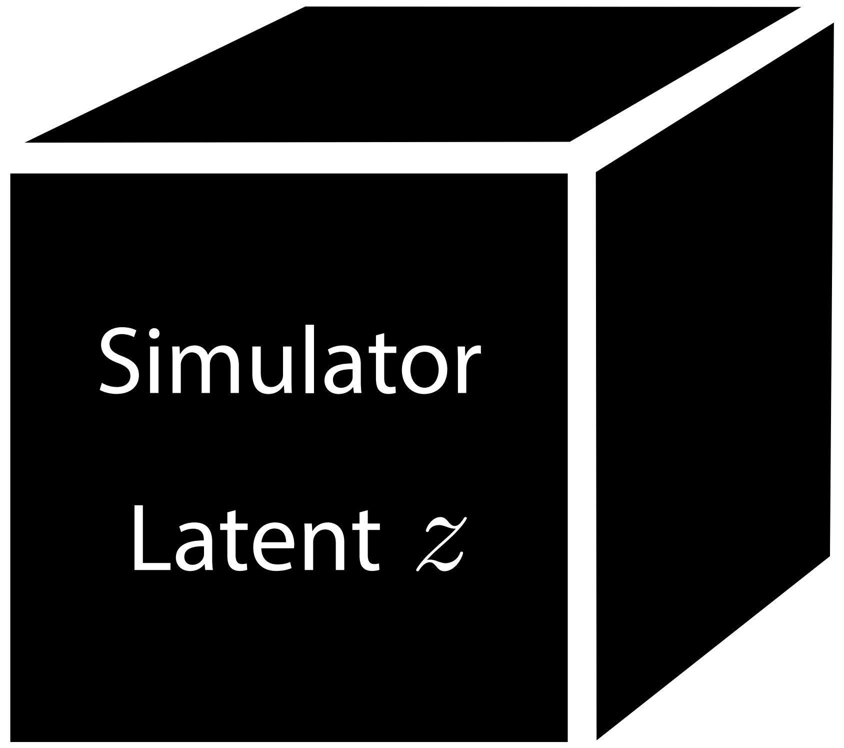
1. The simulation-based inference problem

The problem of simulation-based ("likelihood-free") inference



The problem of simulation-based ("likelihood-free") inference

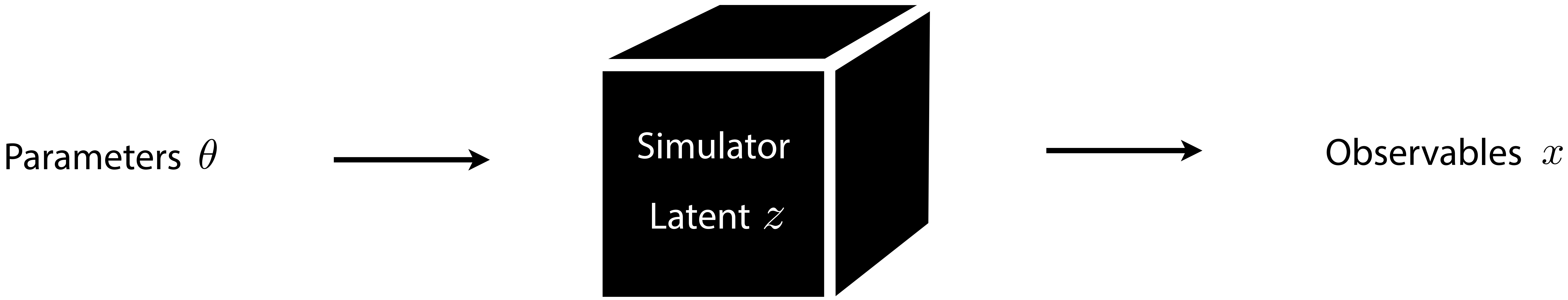
Parameters θ $\longrightarrow$



$\longrightarrow$ Observables x

Prediction:
- Well-motivated mechanistic, causal model
- Simulator can generate samples $x \sim p(x|\theta)$

The problem of simulation-based ("likelihood-free") inference



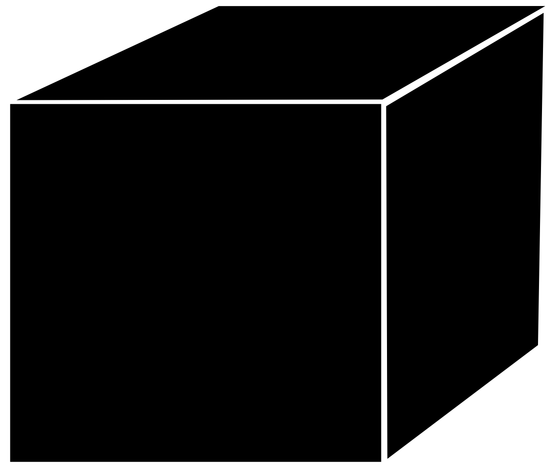
Prediction:
- Well-motivated mechanistic, causal model
- Simulator can generate samples $x \sim p(x|\theta)$

Inference:
- Interactions between low-level components lead to challenging inverse problems
- Likelihood $p(x|\theta) = \int \mathrm{d}z \; p(x, z|\theta)$ is intractable

Problem statement

Given

- a simulator that lets you generate N samples $x_i \sim p(x_i|\theta_i)$ for parameter points θ_i of our choice,



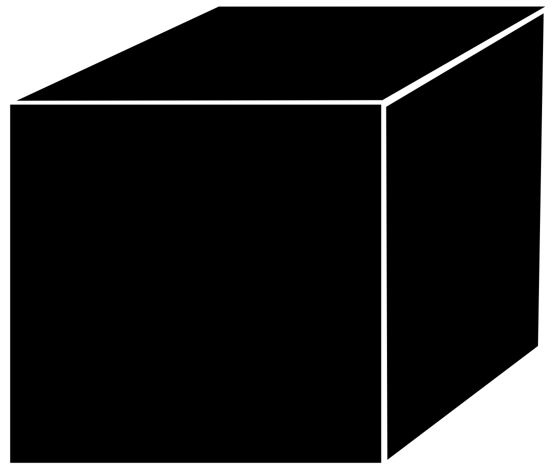
- observed data $x_{\text{obs}} \sim p(x_{\text{obs}}|\theta_{\text{true}})$, and

- some prior belief $p(\theta)$,

Problem statement

Given

- a simulator that lets you generate N samples $x_i \sim p(x_i|\theta_i)$ for parameter points θ_i of our choice,



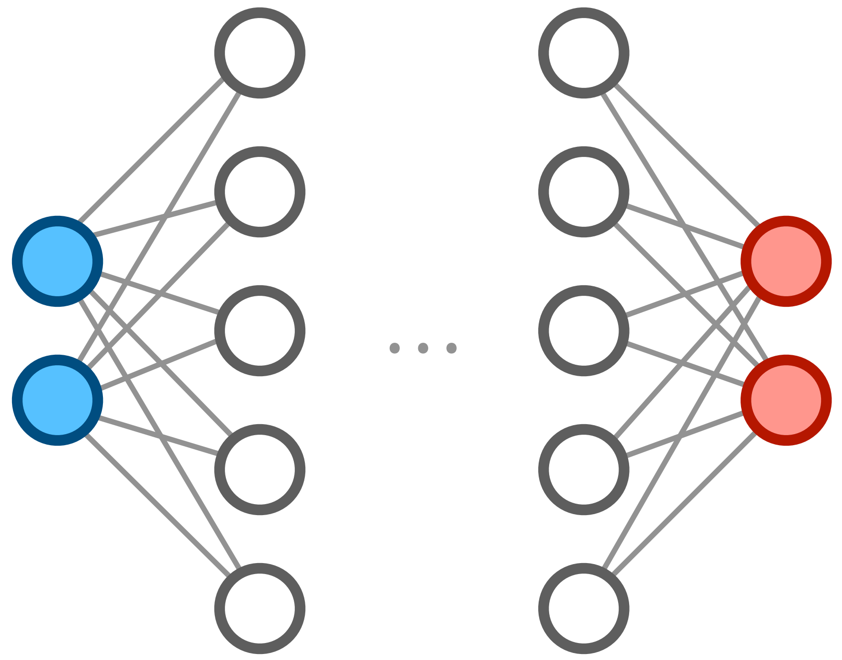
- observed data $x_{\text{obs}} \sim p(x_{\text{obs}}|\theta_{\text{true}})$, and
- some prior belief $p(\theta)$,

can we estimate

- the true parameters $\hat{\theta}_{\text{true}}$,
- confidence sets based on likelihood $\hat{p}(x_{\text{obs}}|\theta)$,
- or the posterior $\hat{p}(\theta|x_{\text{obs}}) = \dfrac{\hat{p}(x_{\text{obs}}|\theta)\, p(\theta)}{\int \mathrm{d}\theta' \, \hat{p}(x_{\text{obs}}|\theta')\, p(\theta')}$

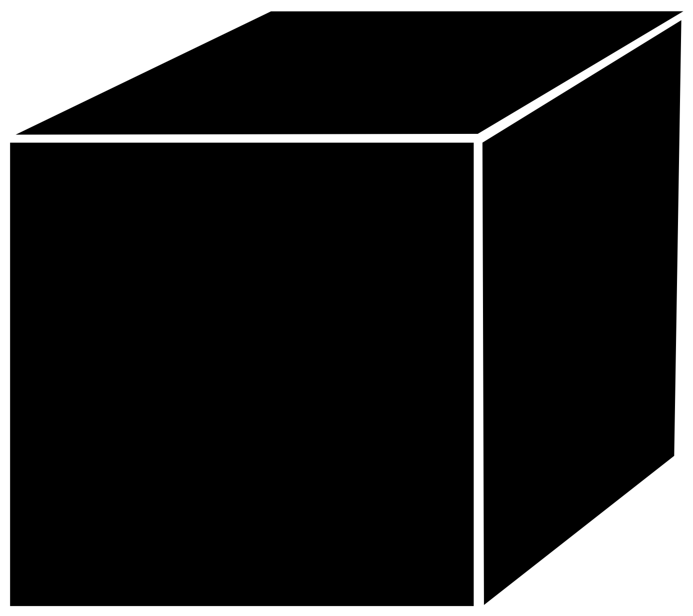
  or samples from posterior $\theta \sim \hat{p}(\theta|x_{\text{obs}})$

(which one depends on problem and field)?



2. Machine learning methods

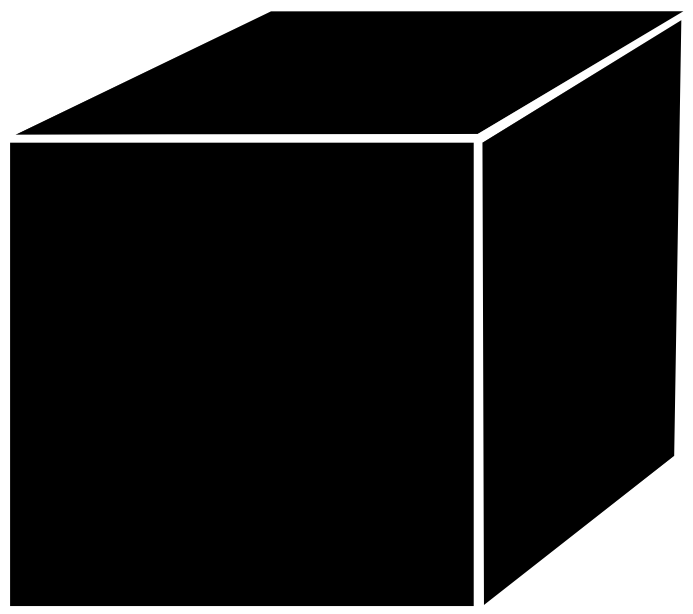
Get the best of two worlds

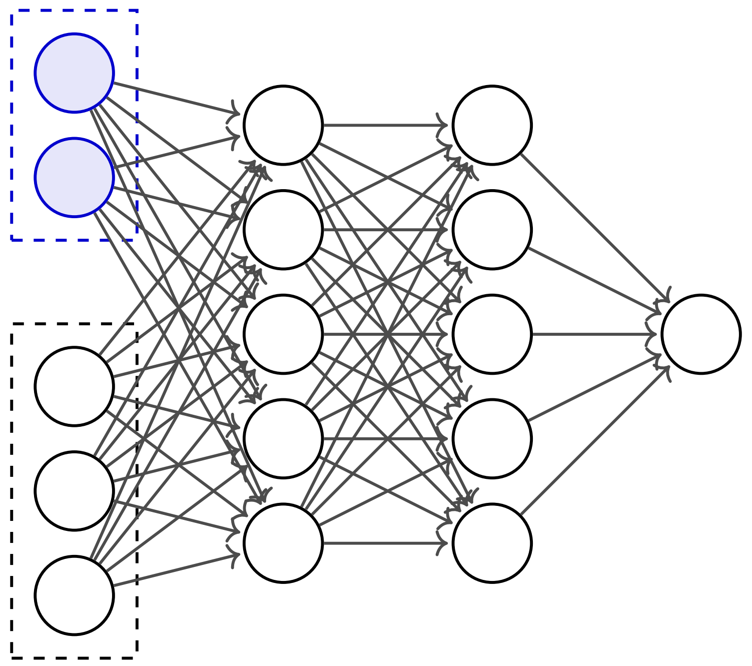


Simulators: focus on understanding

- based on mechanistic, causal model
- interpretable parameters

Get the best of two worlds





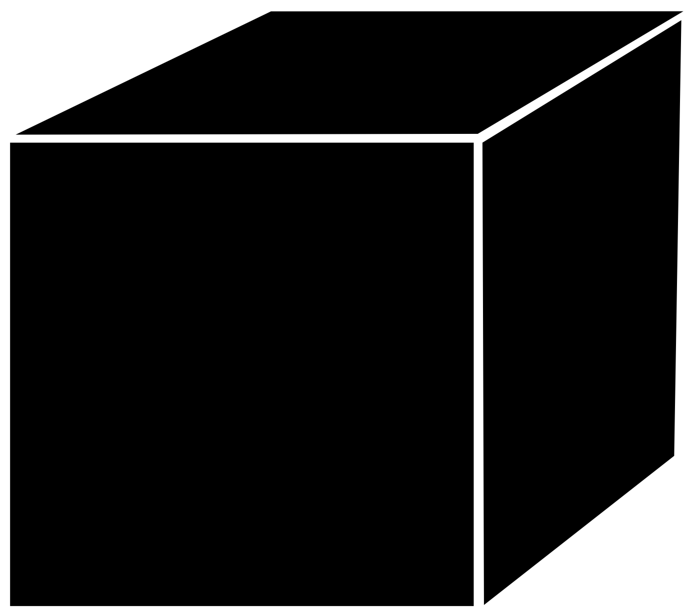
Simulators: focus on understanding

- based on mechanistic, causal model
- interpretable parameters

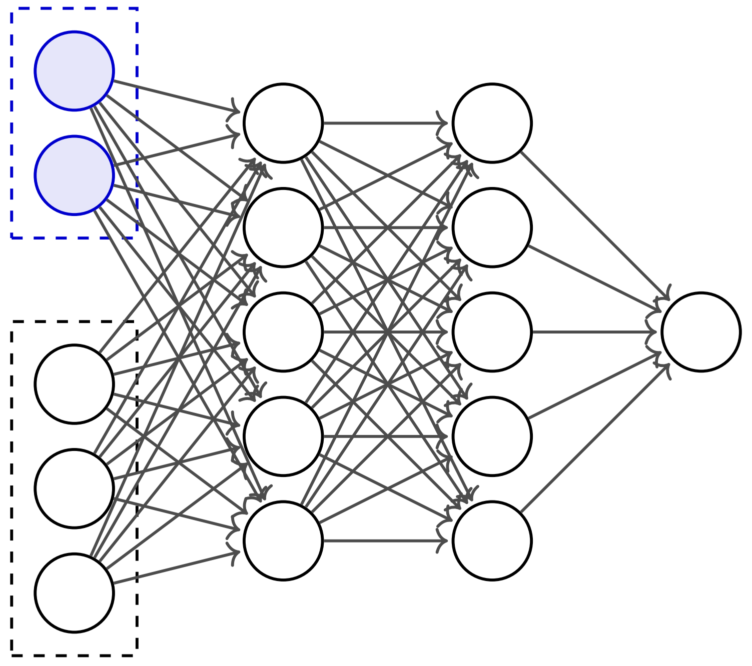
Machine learning models: focus on performance

- good at learning representations from data
- good inductive biases (images, sequences, graphs, symmetries, hierarchical structures…)
- differentiable, often invertible, probabilistic: well-suited for inference / fitting

Get the best of two worlds



Can we use ML models to fit simulators to data?



Simulators: focus on understanding

- based on mechanistic, causal model
- interpretable parameters

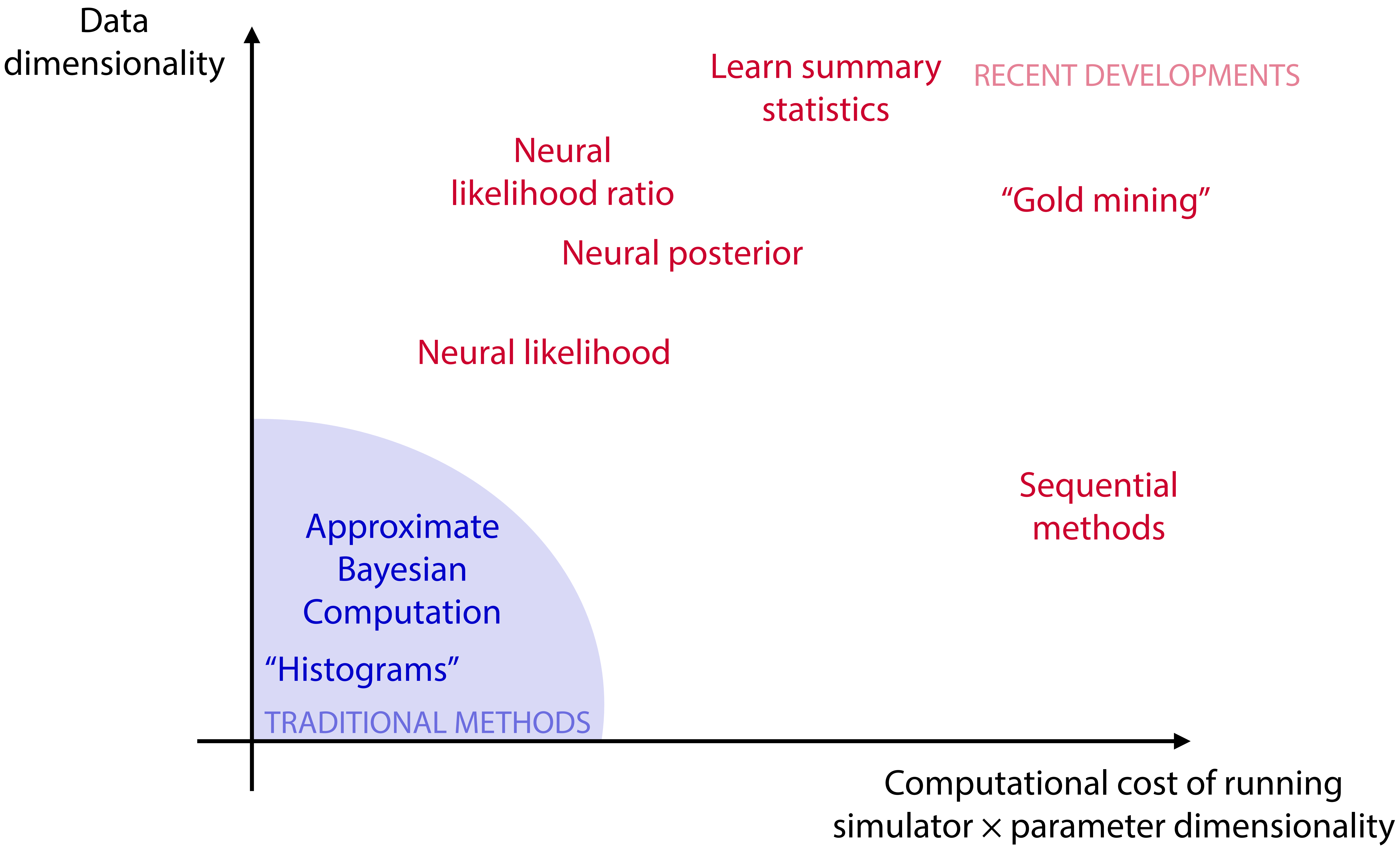
Machine learning models: focus on performance

- good at learning representations from data
- good inductive biases (images, sequences, graphs, symmetries, hierarchical structures…)
- differentiable, often invertible, probabilistic: well-suited for inference / fitting

Can we inject domain knowledge into ML models?

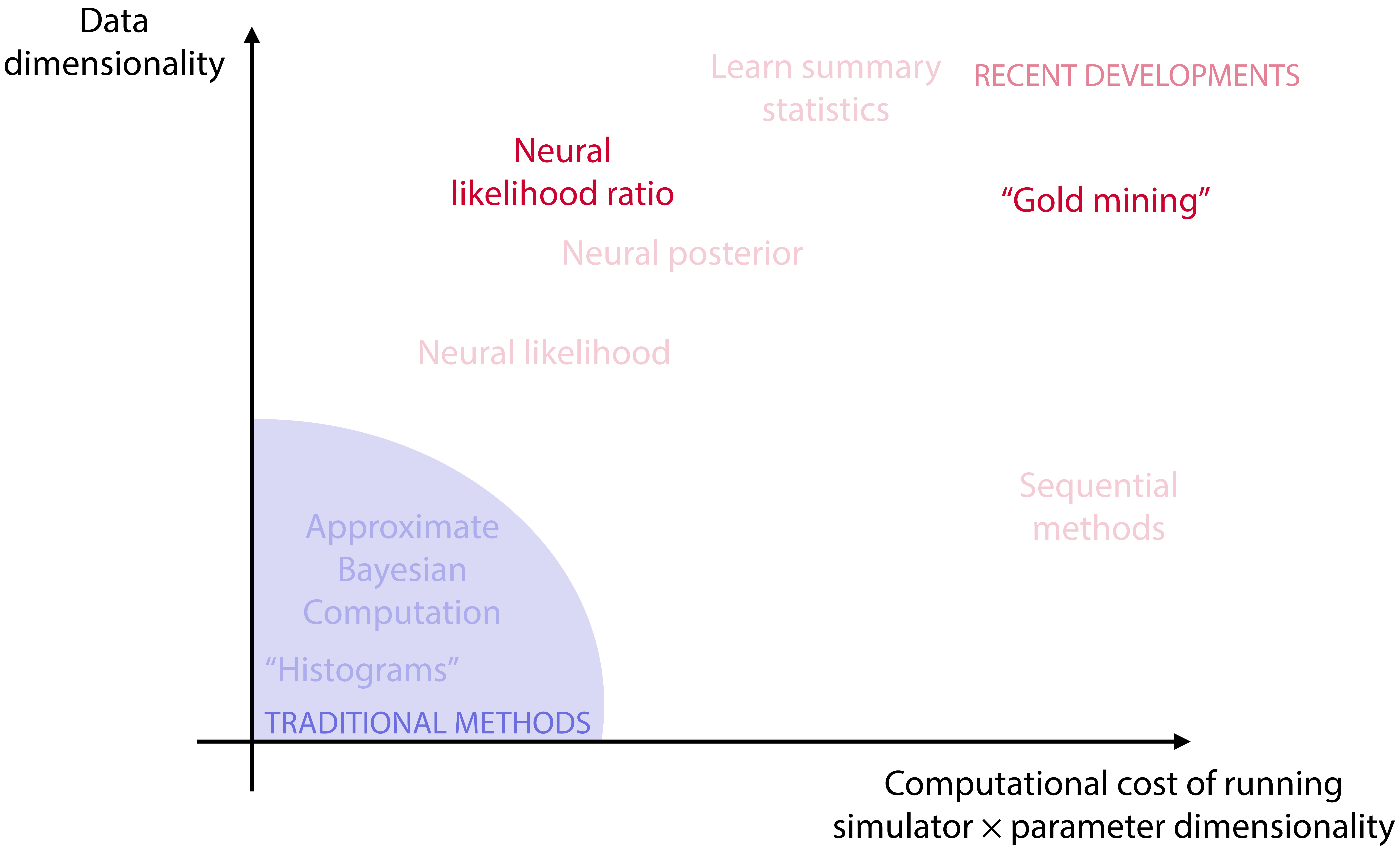
Simulation-based inference methods

[K. Cranmer, JB, G. Louppe 1911.01429]



Simulation-based inference methods

[K. Cranmer, JB, G. Louppe 1911.01429]



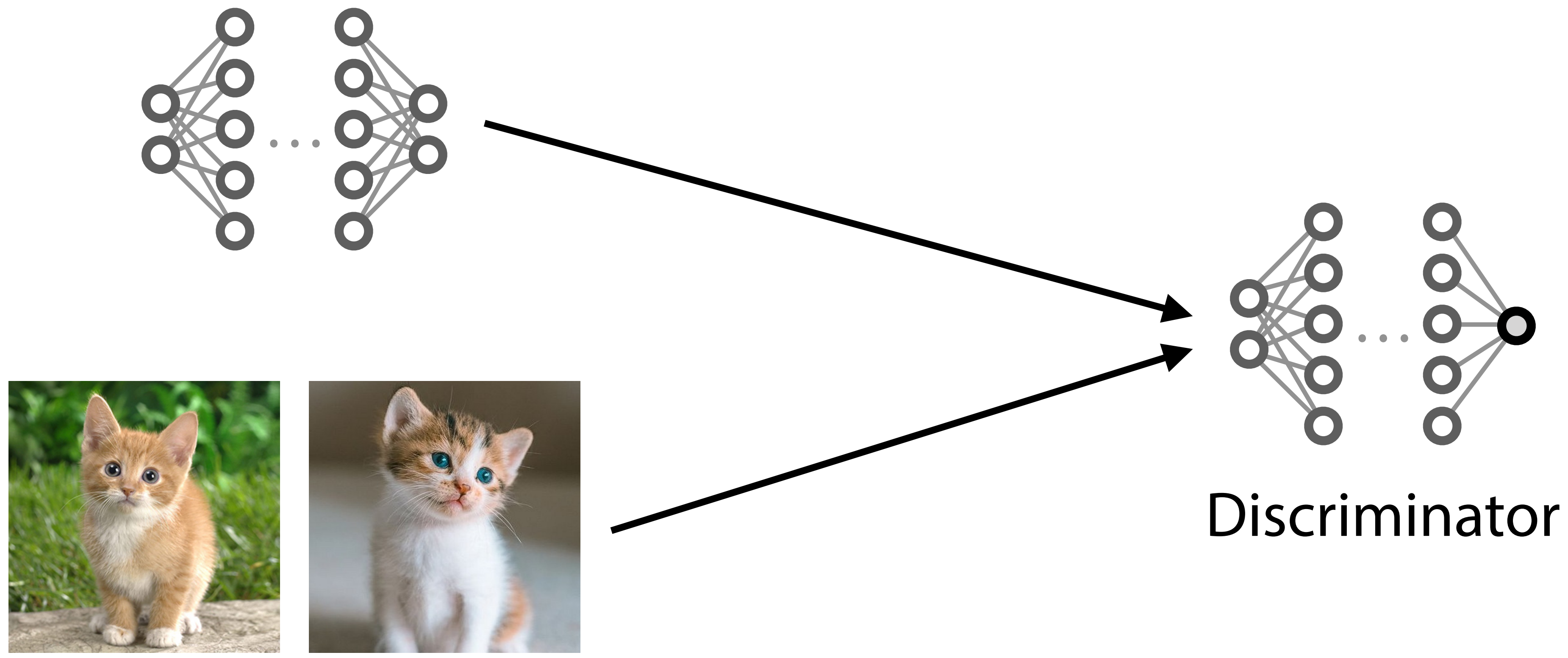
Idea 1: the likelihood ratio trick

- Generative Adversarial Networks (GANs):

[I. Goodfellow et al. 1406.2661]

Generator
(intractable likelihood $p_{\mathrm{gen}}(x)$)

Examples
(intractable likelihood $p_{\mathrm{true}}(x)$)



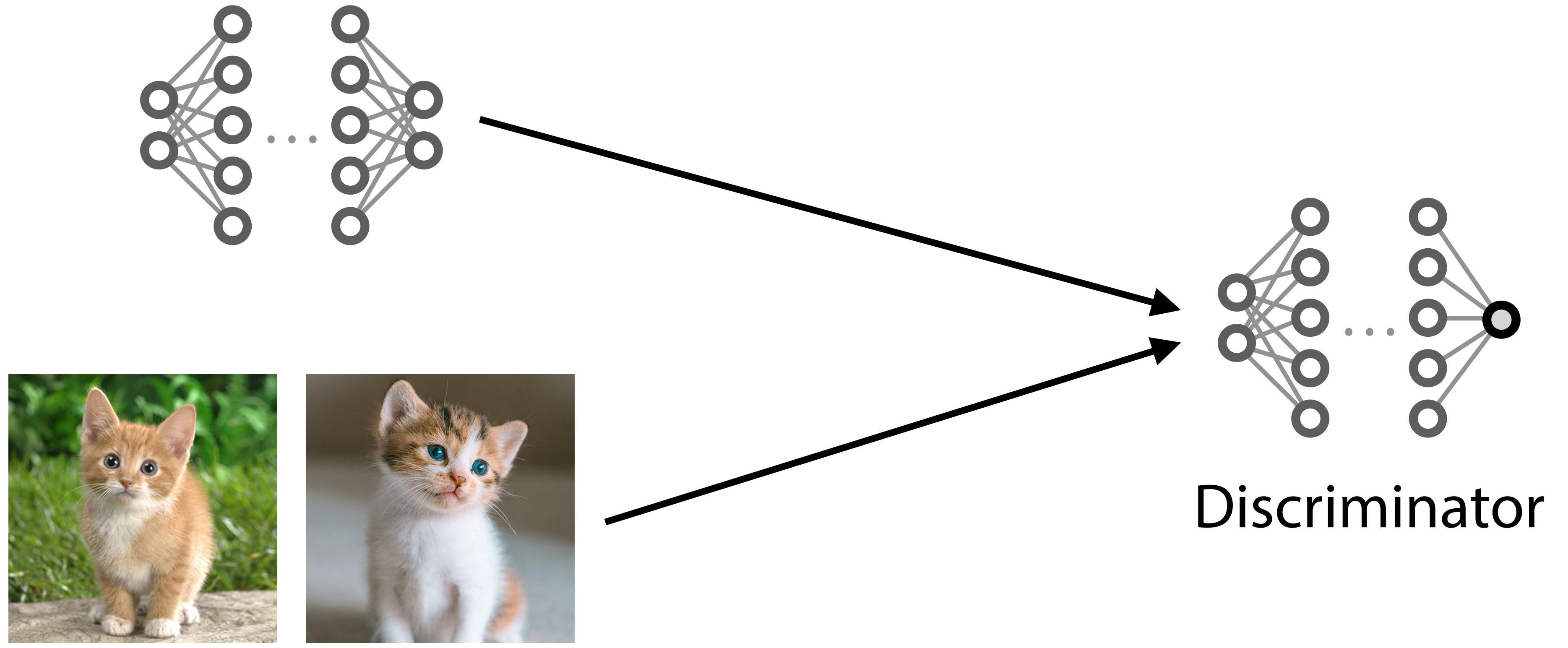
Idea 1: the likelihood ratio trick

- Generative Adversarial Networks (GANs):

[I. Goodfellow et al. 1406.2661]

Generator
(intractable likelihood $p_{\text{gen}}(x)$)

Examples
(intractable likelihood $p_{\text{true}}(x)$)



Discriminator learns decision function

$$s(x) \to \frac{p_{\text{true}}(x)}{p_{\text{gen}}(x) + p_{\text{true}}(x)}$$

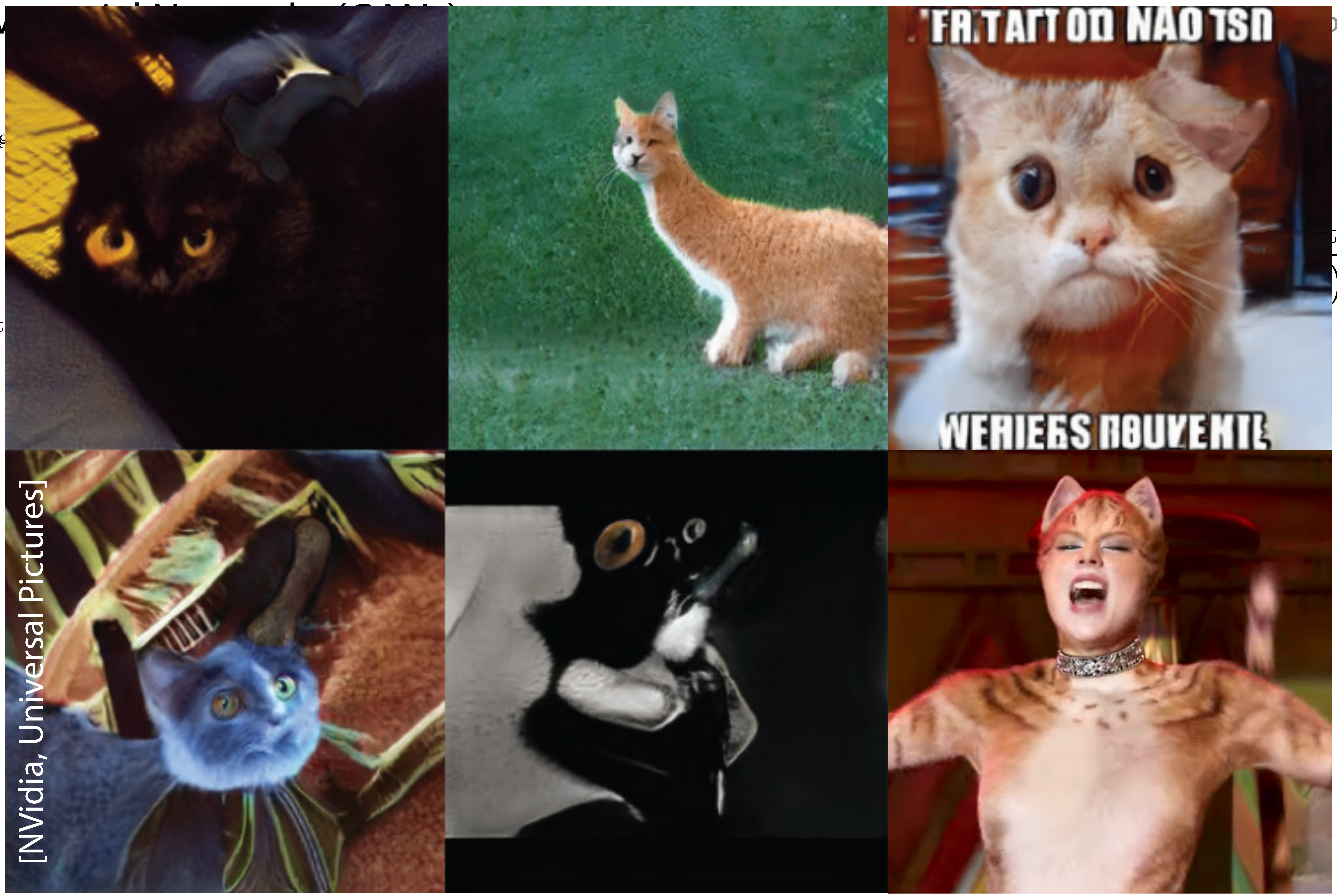
Idea 1: the likelihood ratio trick

- Generative Adv

odfellow et al. 1406.2661]

Generator
(intractable likelihood p_{g}

Examples
(intractable likelihood p_{t}


[NVidia, Universal Pictures]

decision function

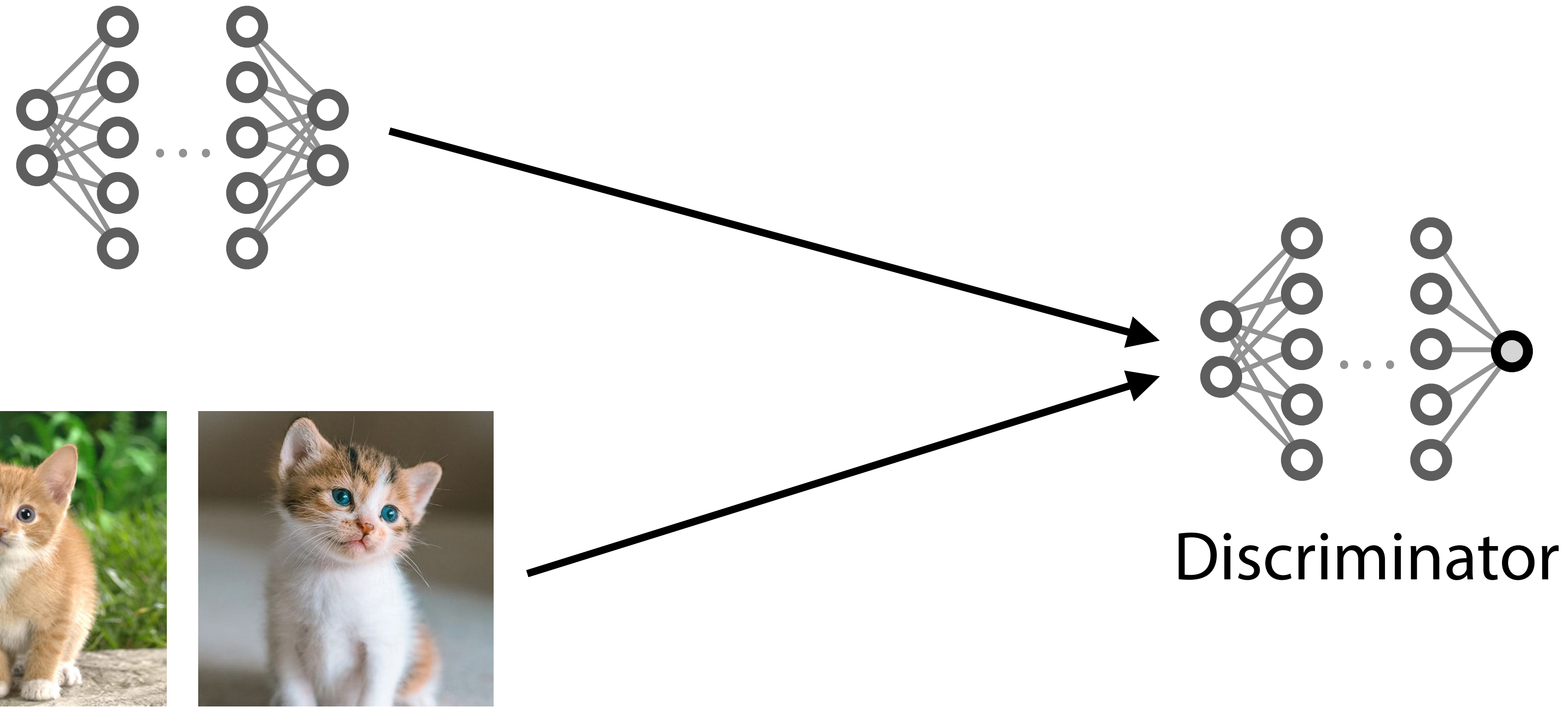
$$\frac{\ldots_{\mathrm{true}}(x)}{\ldots) + p_{\mathrm{true}}(x)}$$

Idea 1: the likelihood ratio trick

- Generative Adversarial Networks (GANs): [I. Goodfellow et al. 1406.2661]

Generator
(intractable likelihood $p_{\rm gen}(x)$)

Examples
(intractable likelihood $p_{\rm true}(x)$)



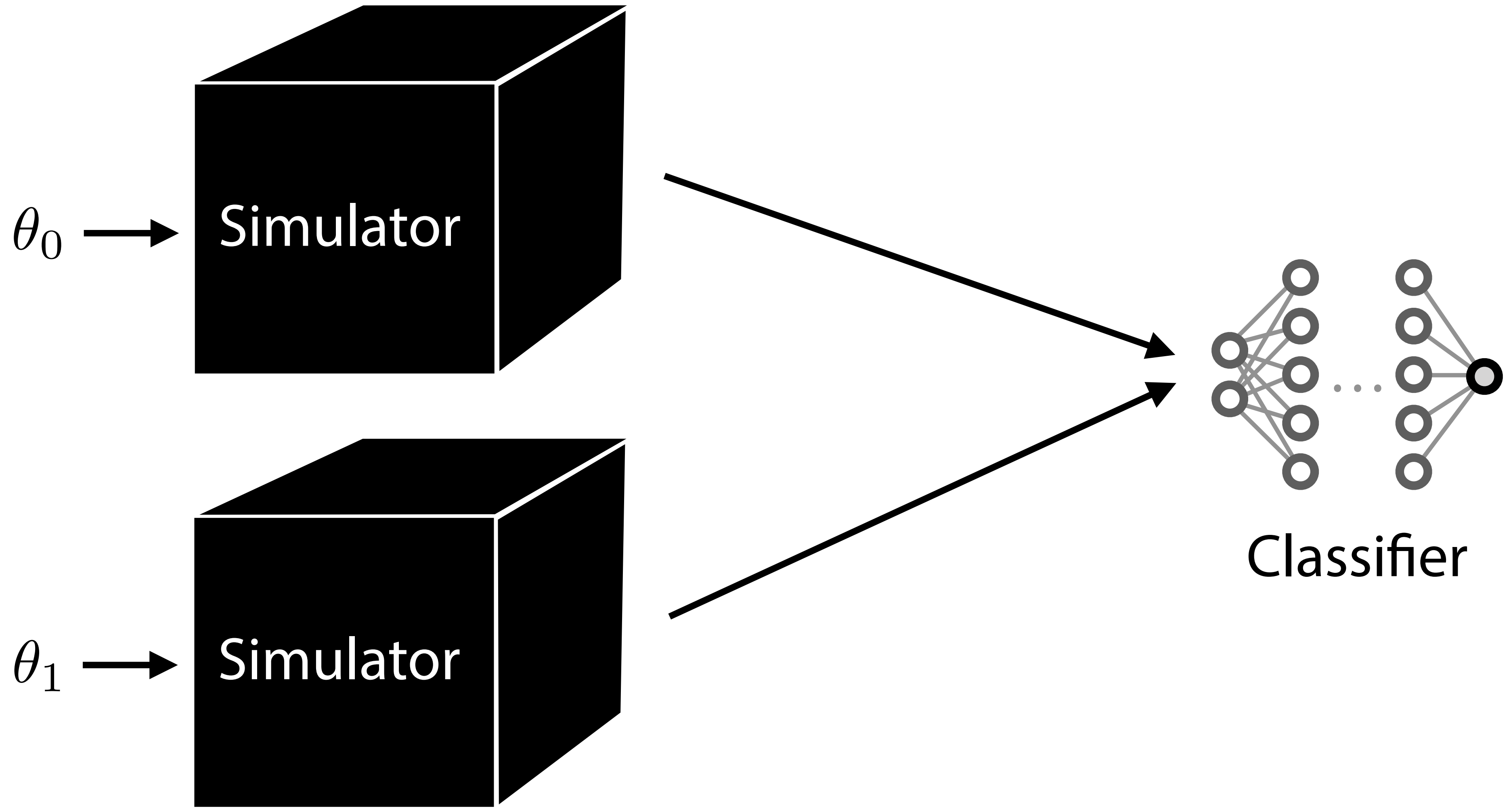
Discriminator learns decision function

$$s(x) \to \frac{p_{\rm true}(x)}{p_{\rm gen}(x) + p_{\rm true}(x)}$$

- Similarly, we can train a classifier between two sets of simulated samples [K. Cranmer, J. Pavez, G. Louppe 1506.02169]

Simulator
(intractable likelihood $p(x|\theta_0)$)

Simulator
(intractable likelihood $p(x|\theta_1)$)

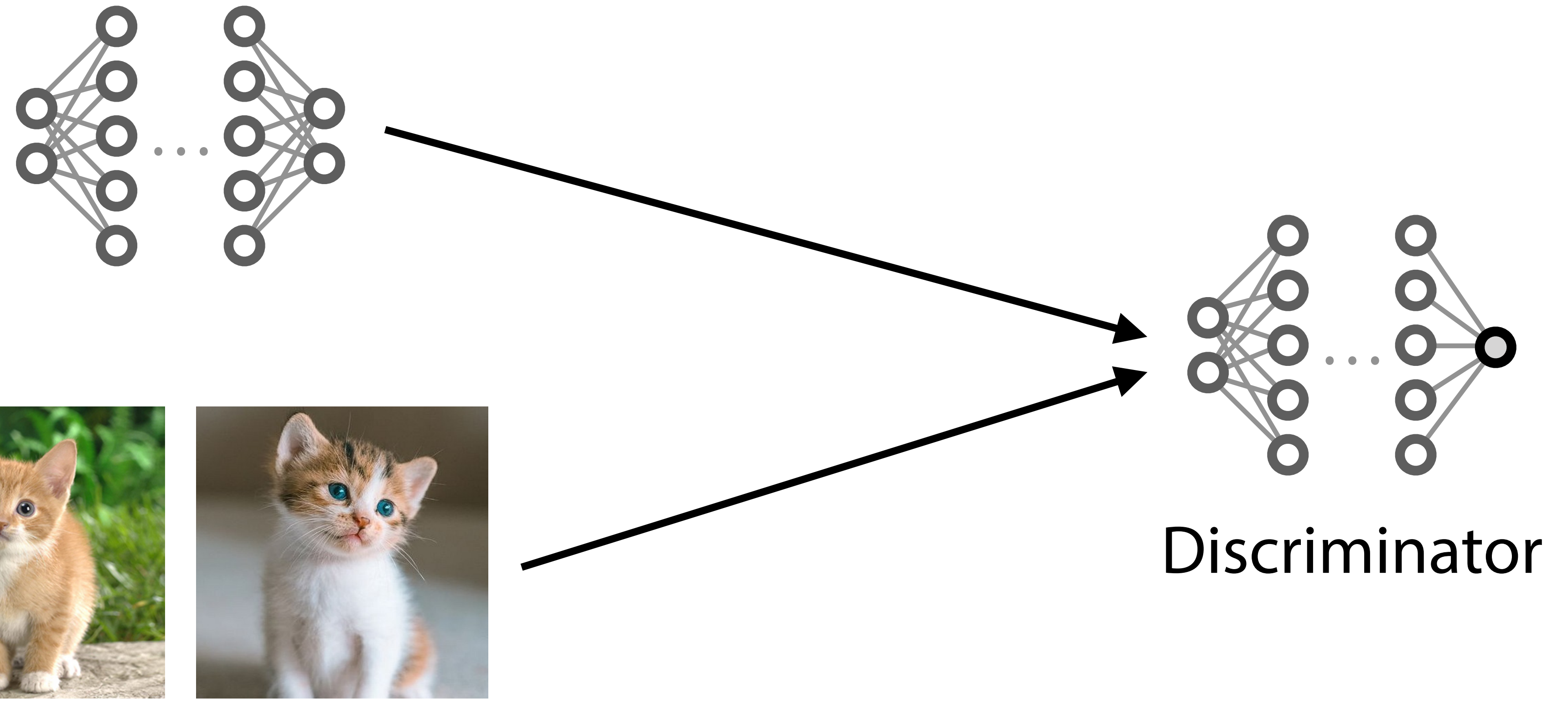


Idea 1: the likelihood ratio trick

- Generative Adversarial Networks (GANs): [I. Goodfellow et al. 1406.2661]

Generator
(intractable likelihood $p_{\text{gen}}(x)$)

Examples
(intractable likelihood $p_{\text{true}}(x)$)



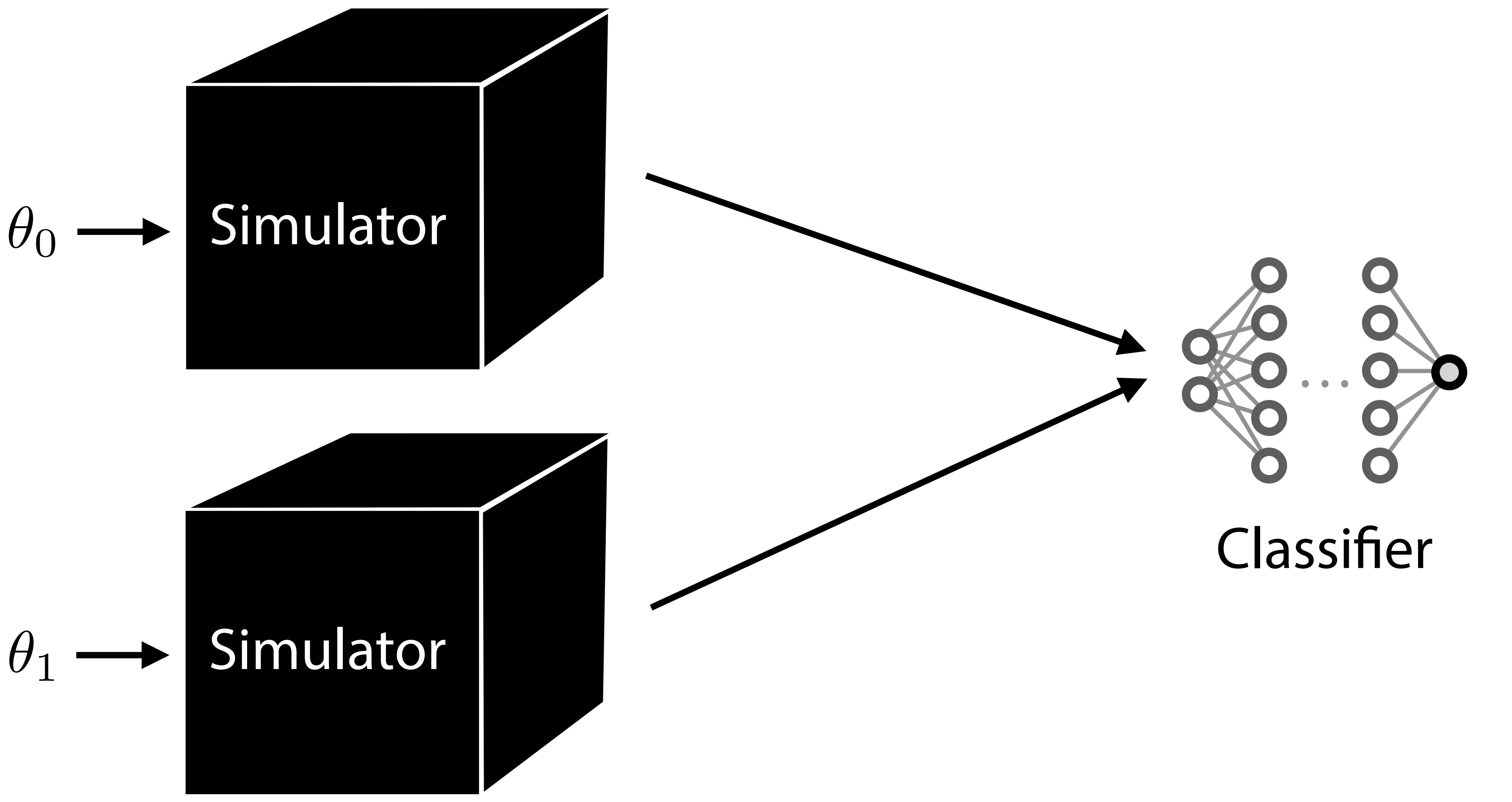
Discriminator learns decision function

$$s(x) \to \frac{p_{\text{true}}(x)}{p_{\text{gen}}(x) + p_{\text{true}}(x)}$$

- Similarly, we can train a classifier between two sets of simulated samples [K. Cranmer, J. Pavez, G. Louppe 1506.02169]

Simulator
(intractable likelihood $p(x|\theta_0)$)

Simulator
(intractable likelihood $p(x|\theta_1)$)



Classifier learns decision function

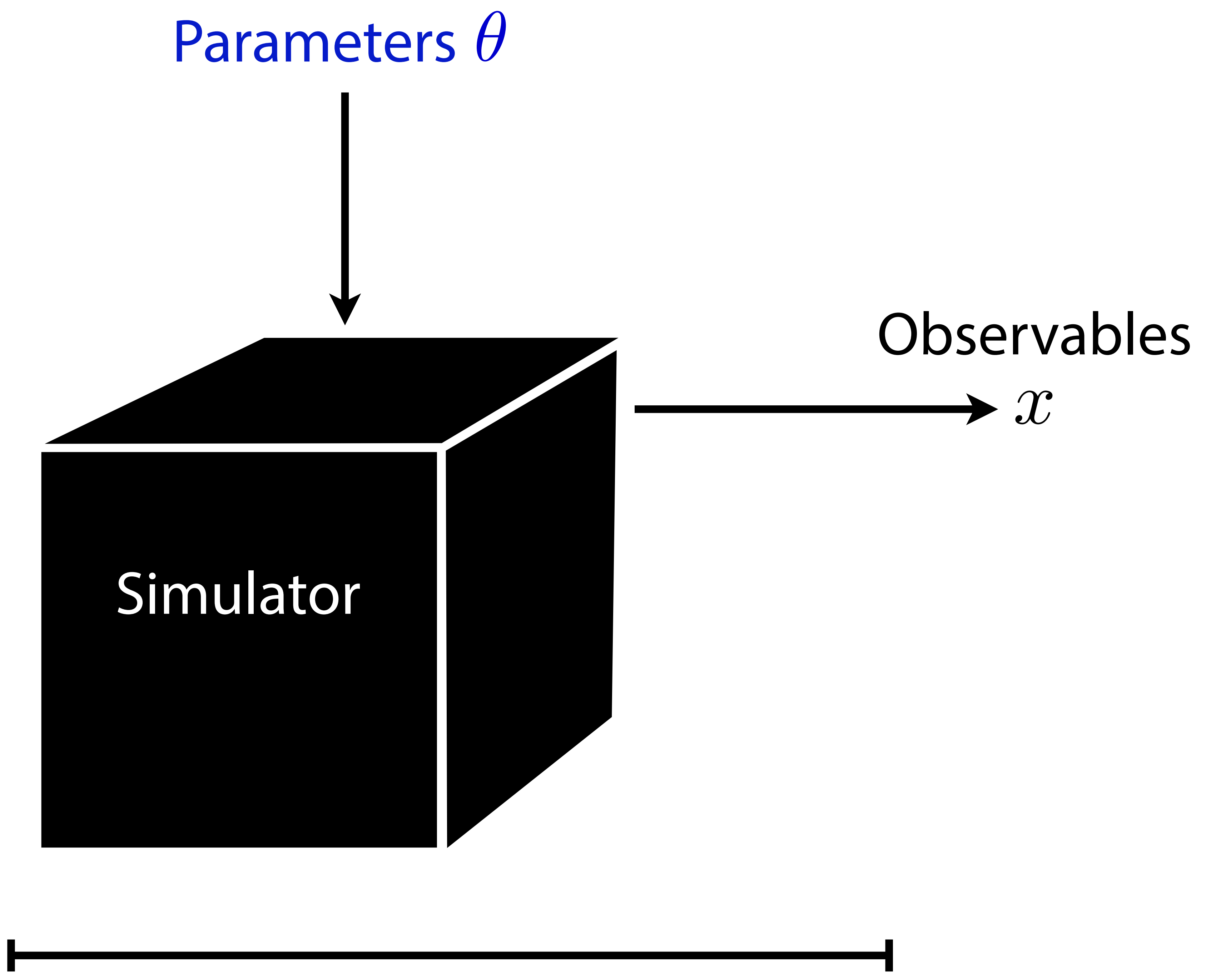
$$s(x) \to \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)}$$

⇒ Estimator for likelihood ratio

$$\hat{r}(x) = \frac{1 - s(x)}{s(x)} \to \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

Inference by likelihood ratio trick

[K. Cranmer, J. Pavez, G. Louppe 1506.02169]

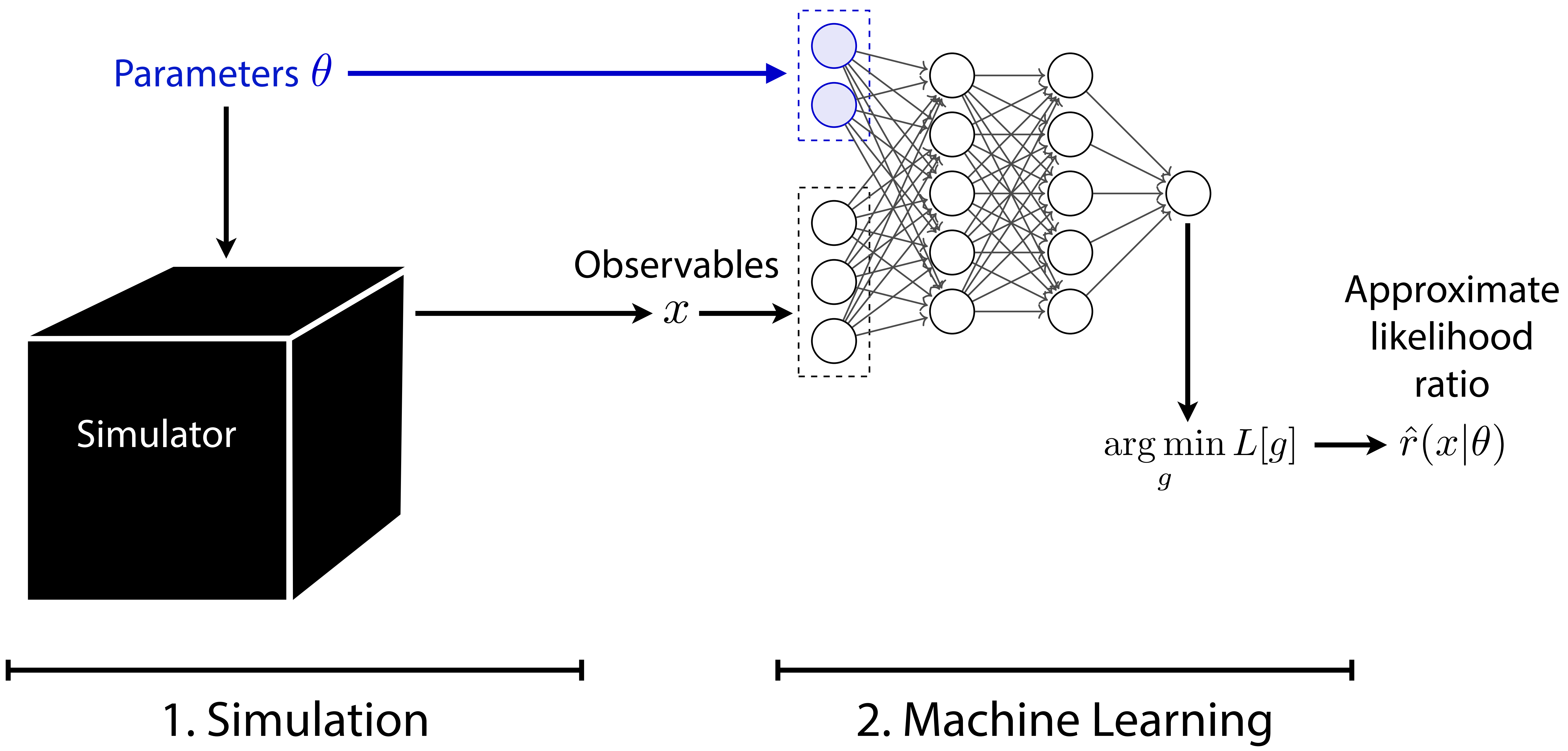


1. Simulation

Run simulator and save data

Inference by likelihood ratio trick

[K. Cranmer, J. Pavez, G. Louppe 1506.02169]

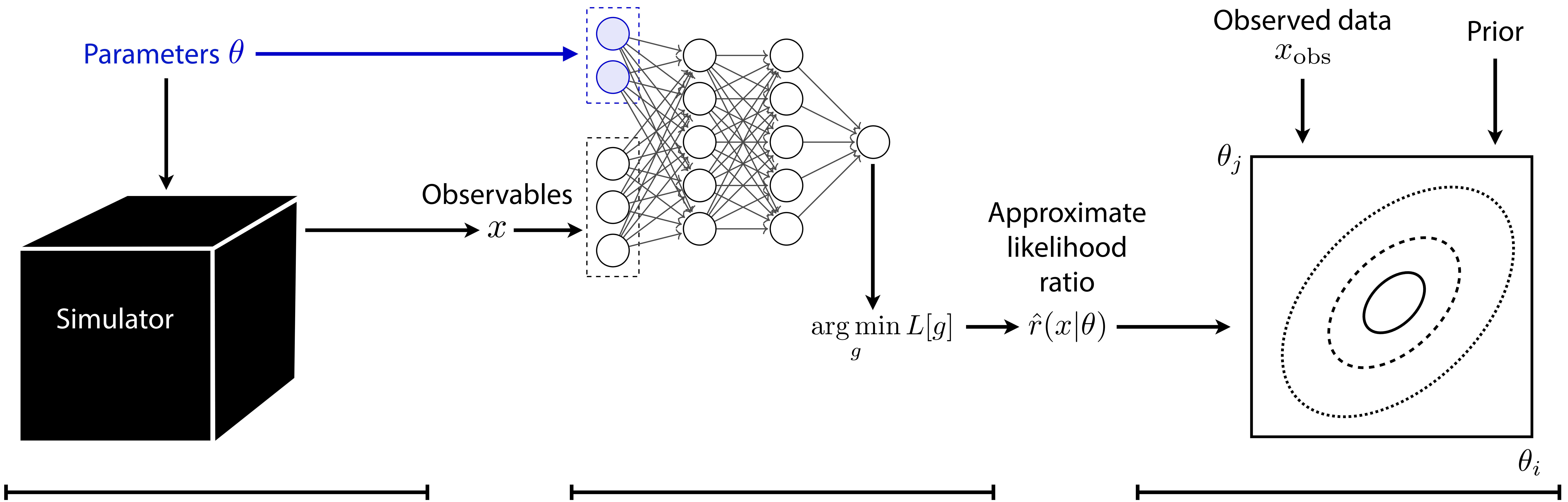


Run simulator and save data

Train NN classifier, interpret as likelihood ratio estimator

Inference by likelihood ratio trick

[K. Cranmer, J. Pavez, G. Louppe 1506.02169]



1. Simulation

Run simulator and save data

2. Machine Learning

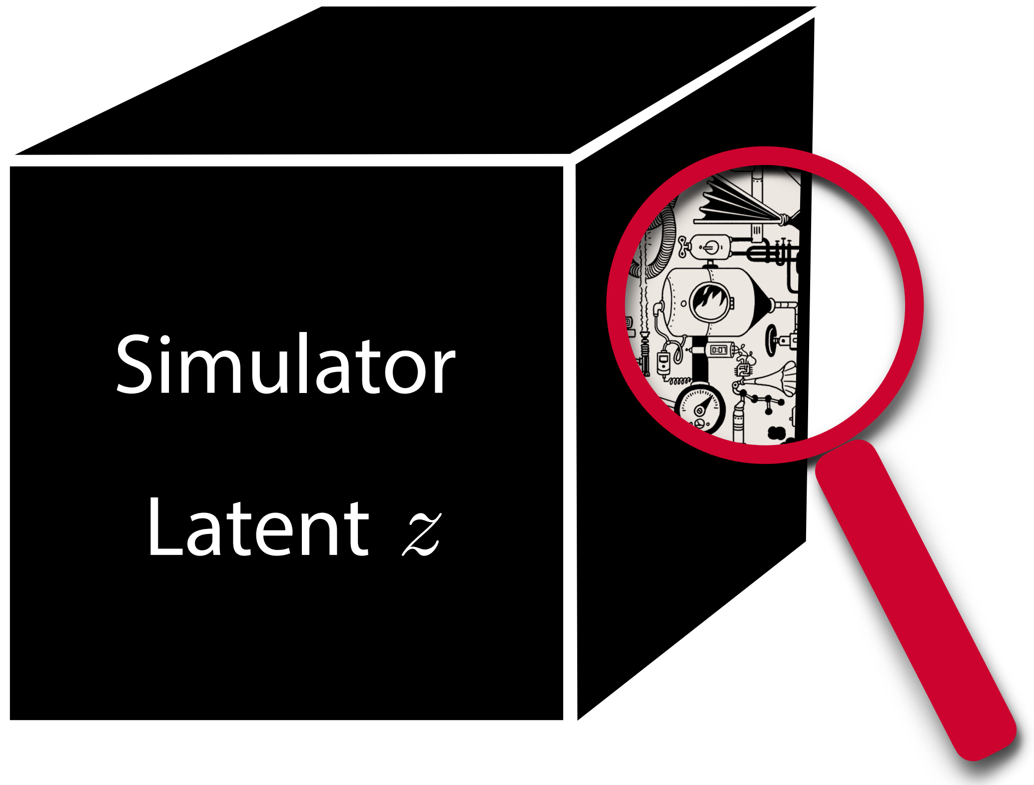
Train NN classifier, interpret as likelihood ratio estimator

3. Inference

Amortized: cheap to repeat for new data

Idea 2: gold mining

[JB, G. Louppe, J. Pavez, K. Cranmer 1805.12244, 1805.00013, 1805.00020]



We cannot compute $p(x|\theta) = \int \mathrm{d}z\, p(x, z|\theta)$,

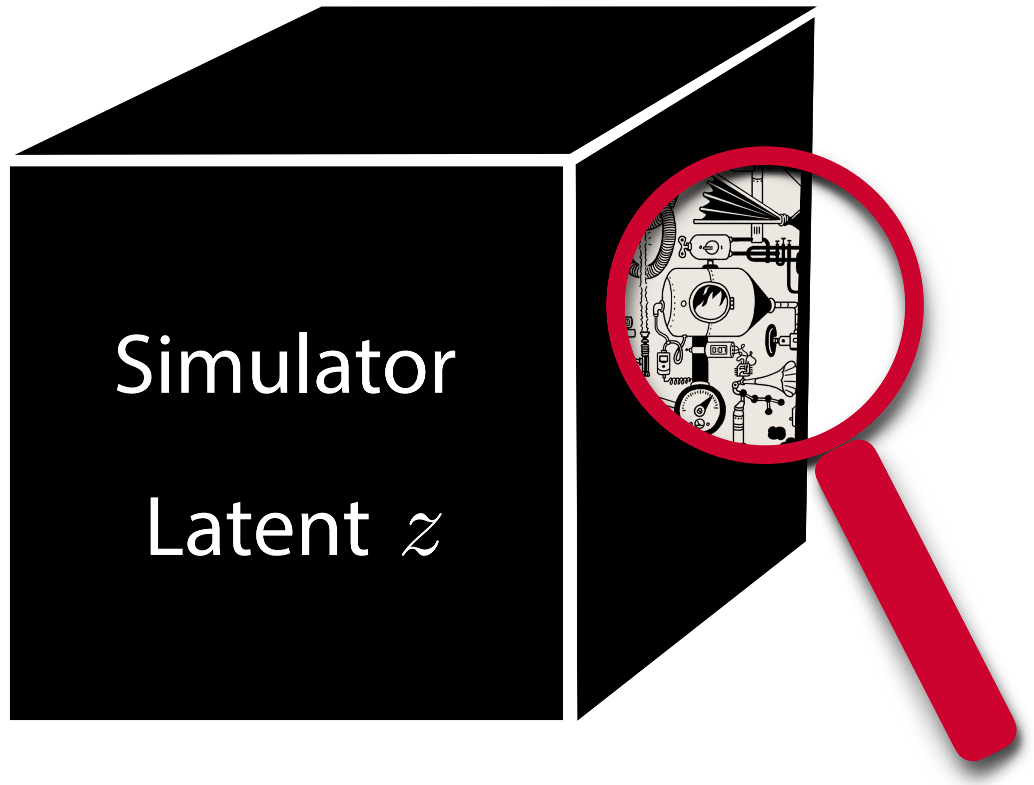
but often we can use domain knowledge (or autodiff) to compute

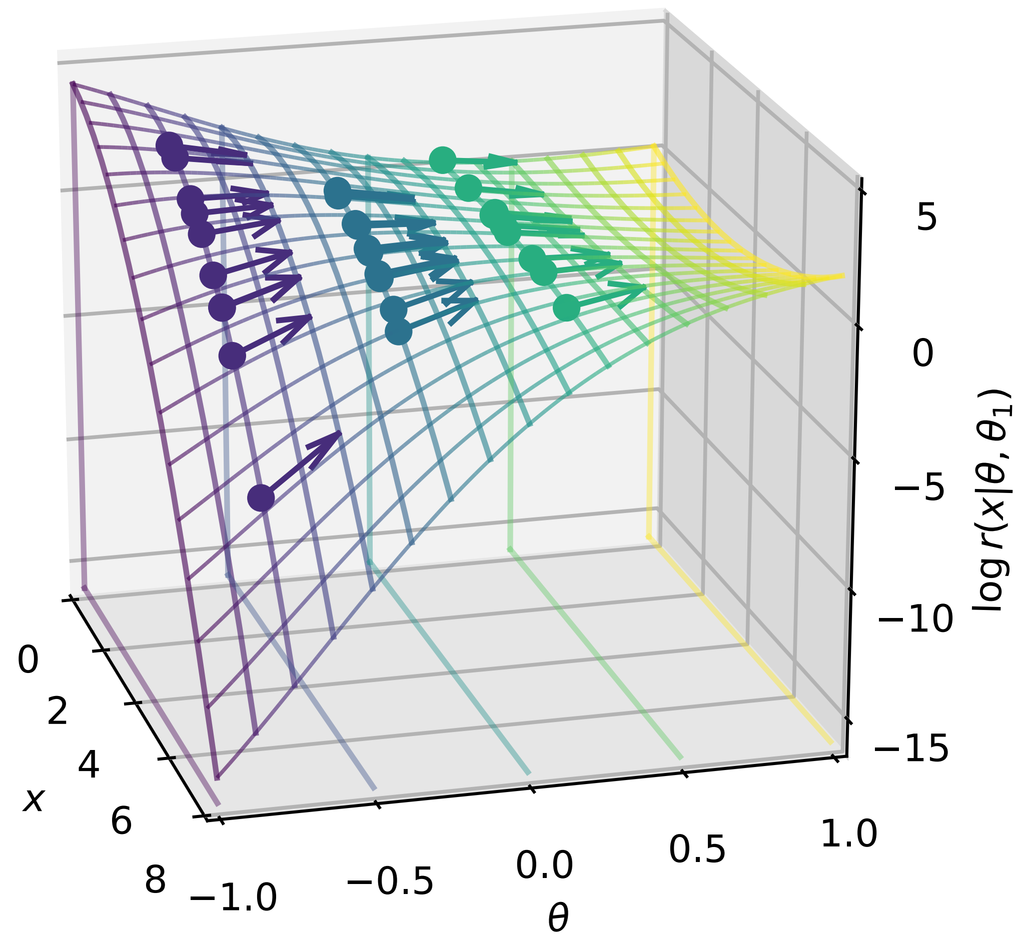
- the joint likelihood ratio $r(x, z|\theta) = \dfrac{p(x, z|\theta)}{p_{\mathrm{ref}}(x, z)}$
- the joint score $t(x, z|\theta) = \nabla_\theta \log p(x, z|\theta)$

(Both depend on the simulator latent variables z)

Idea 2: gold mining

[JB, G. Louppe, J. Pavez, K. Cranmer 1805.12244, 1805.00013, 1805.00020]





We cannot compute $p(x|\theta) = \int \mathrm{d}z\, p(x, z|\theta)$,

but often we can use domain knowledge (or autodiff) to compute

- the joint likelihood ratio $r(x, z|\theta) = \dfrac{p(x, z|\theta)}{p_{\text{ref}}(x, z)}$
- the joint score $t(x, z|\theta) = \nabla_\theta \log p(x, z|\theta)$

(Both depend on the simulator latent variables z)

Pleasant surprises: we have shown that

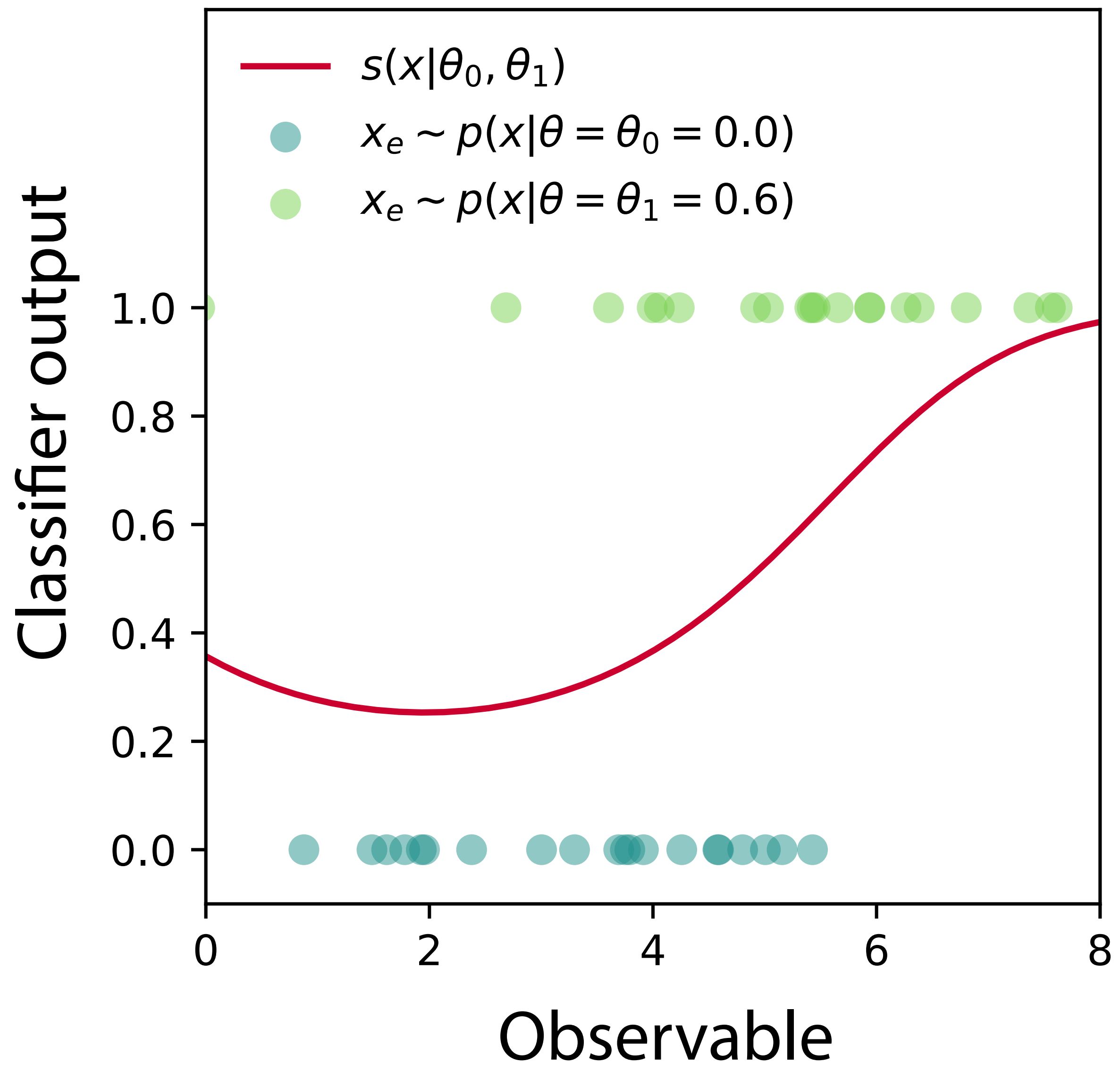
- the joint likelihood ratio is an unbiased estimator of the likelihood ratio
- the joint score provides unbiased gradient information

$\Rightarrow$ use them as labels in supervised NN training!

Mining gold adds information

[JB, G. Louppe, J. Pavez, K. Cranmer
1805.12244, 1805.00013, 1805.00020]

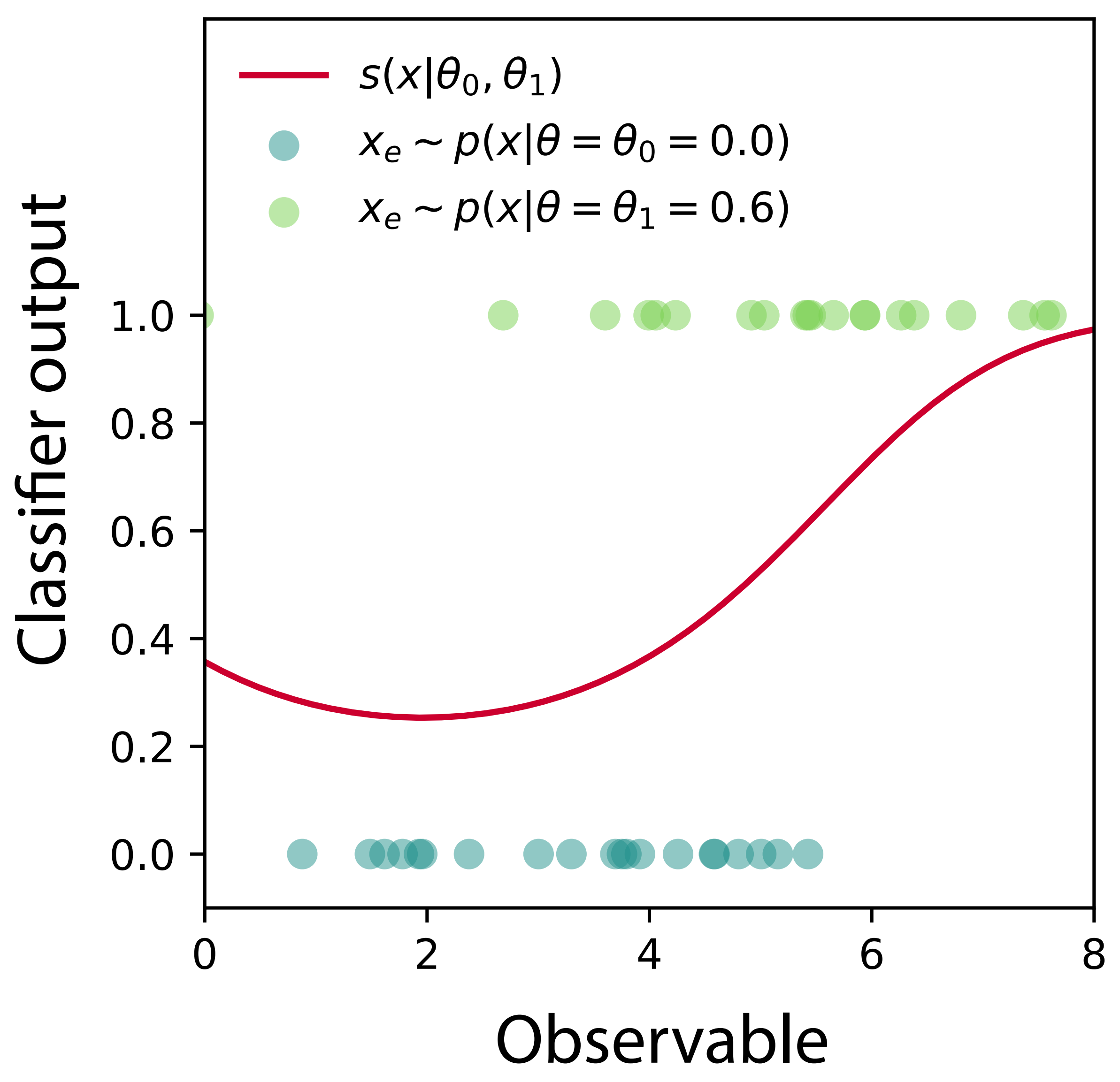
Likelihood ratio trick



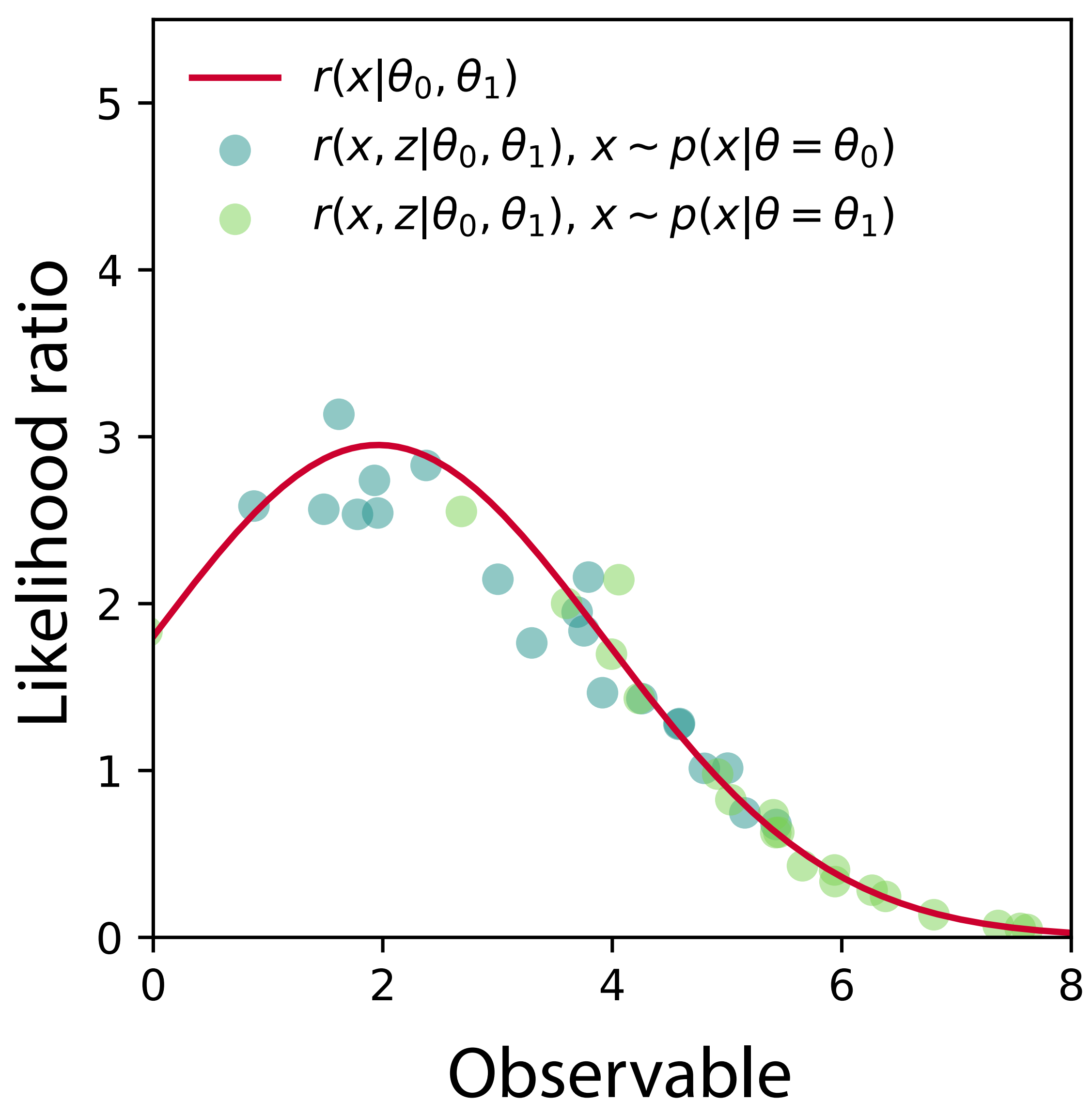
Mining gold adds information

[JB, G. Louppe, J. Pavez, K. Cranmer
1805.12244, 1805.00013, 1805.00020]

Likelihood ratio trick



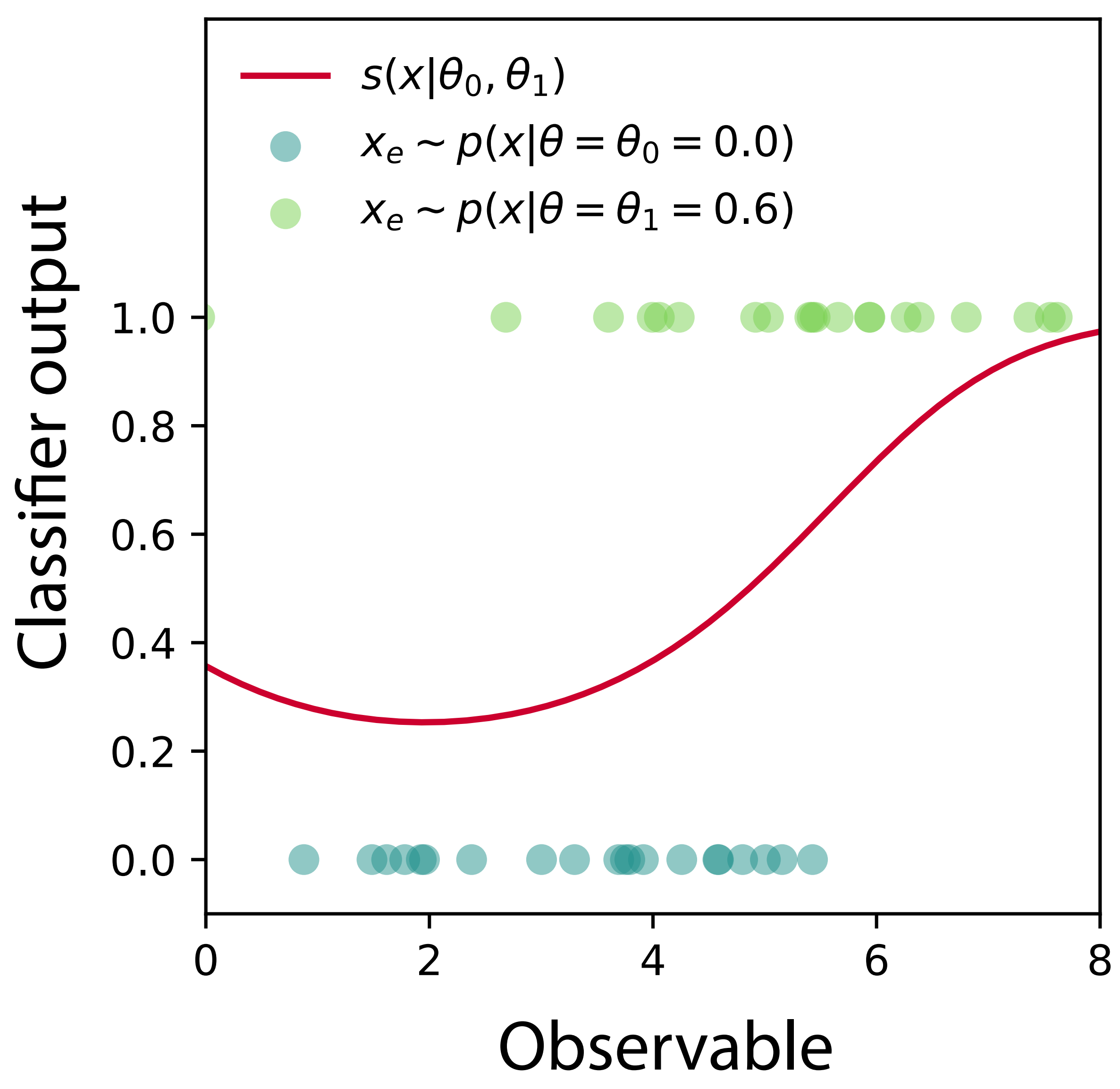
+ joint likelihood ratio



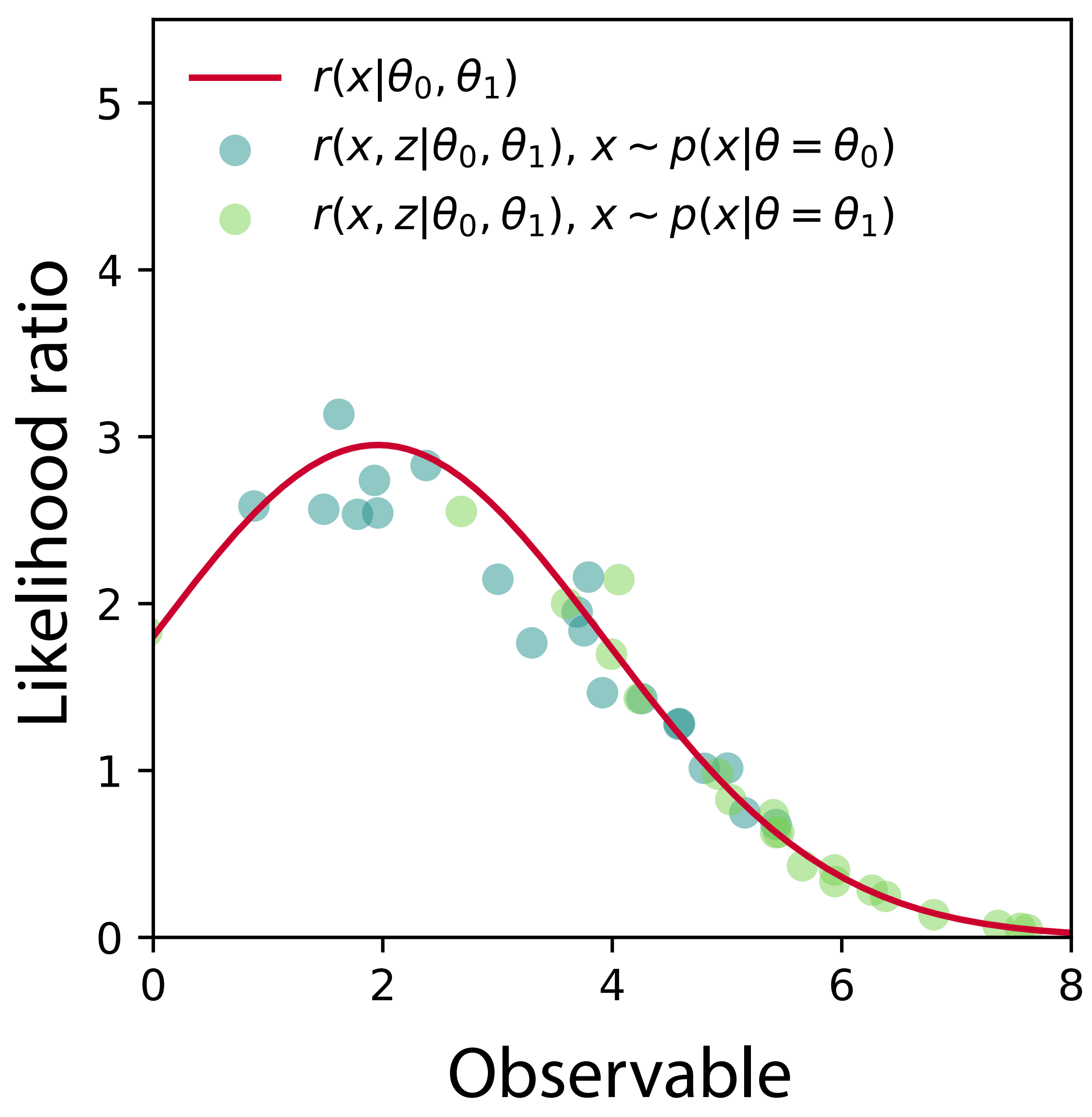
Mining gold adds information

[JB, G. Louppe, J. Pavez, K. Cranmer
1805.12244, 1805.00013, 1805.00020]

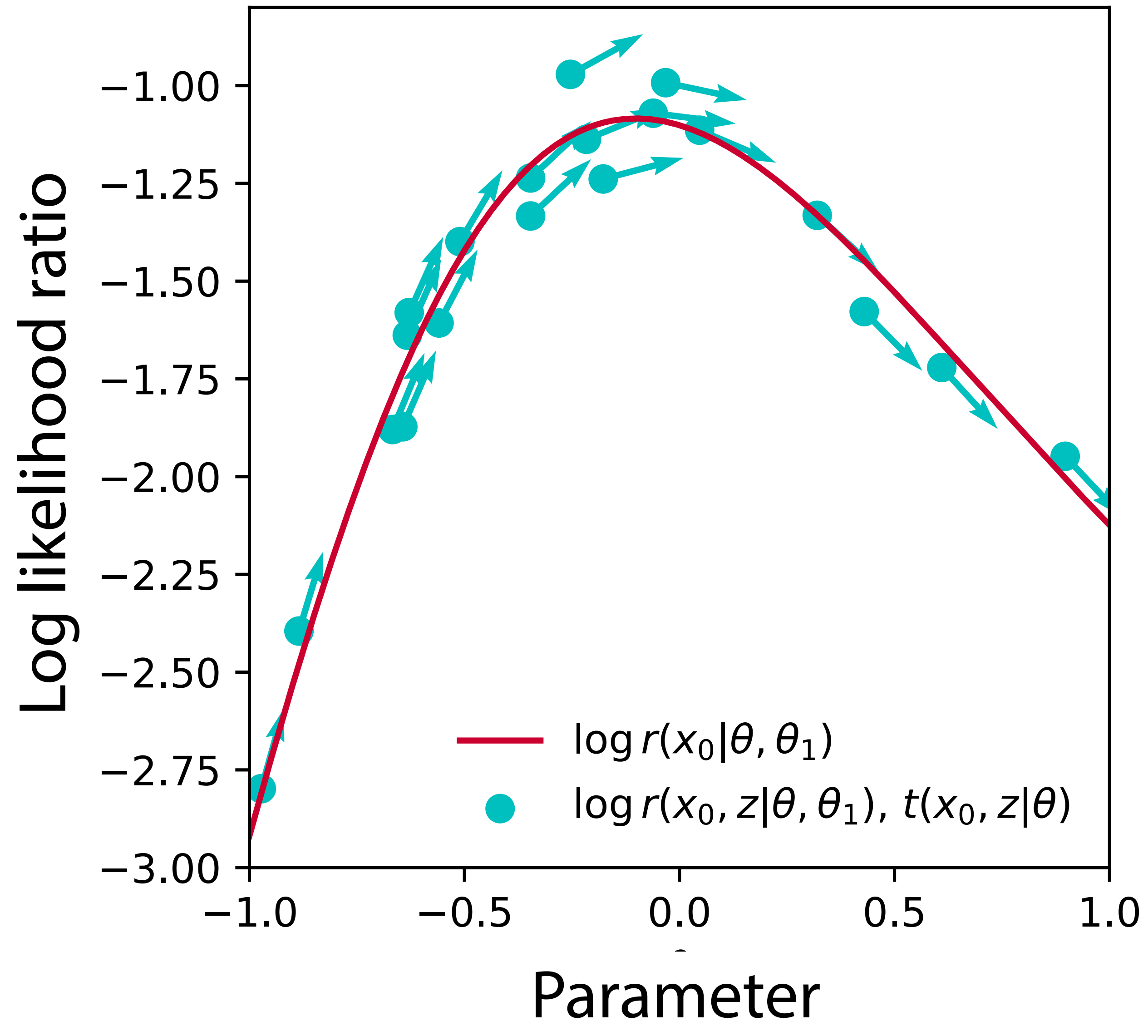
Likelihood ratio trick



+ joint likelihood ratio



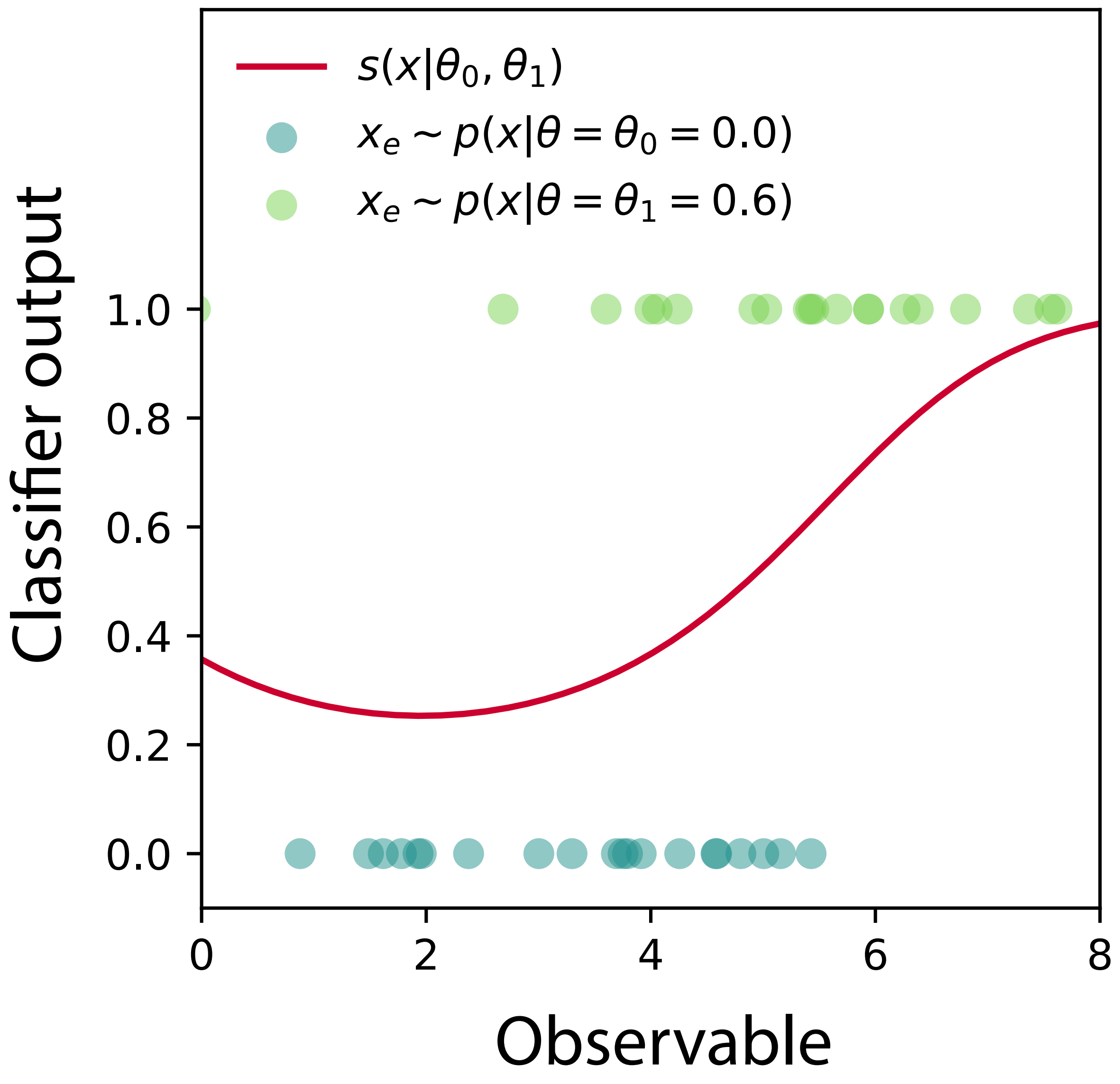
+ joint score



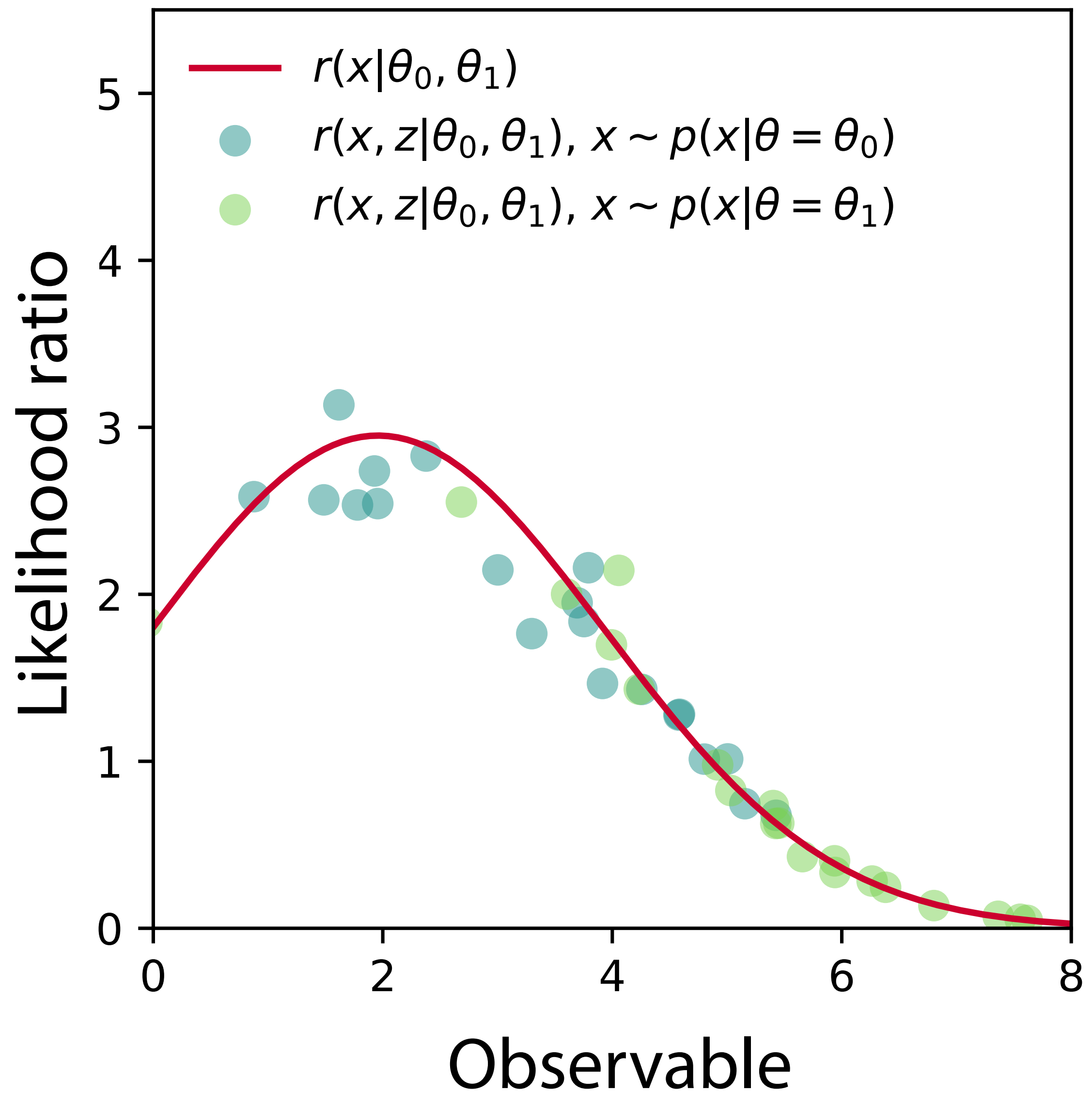
Mining gold adds information

[JB, G. Louppe, J. Pavez, K. Cranmer
1805.12244, 1805.00013, 1805.00020]

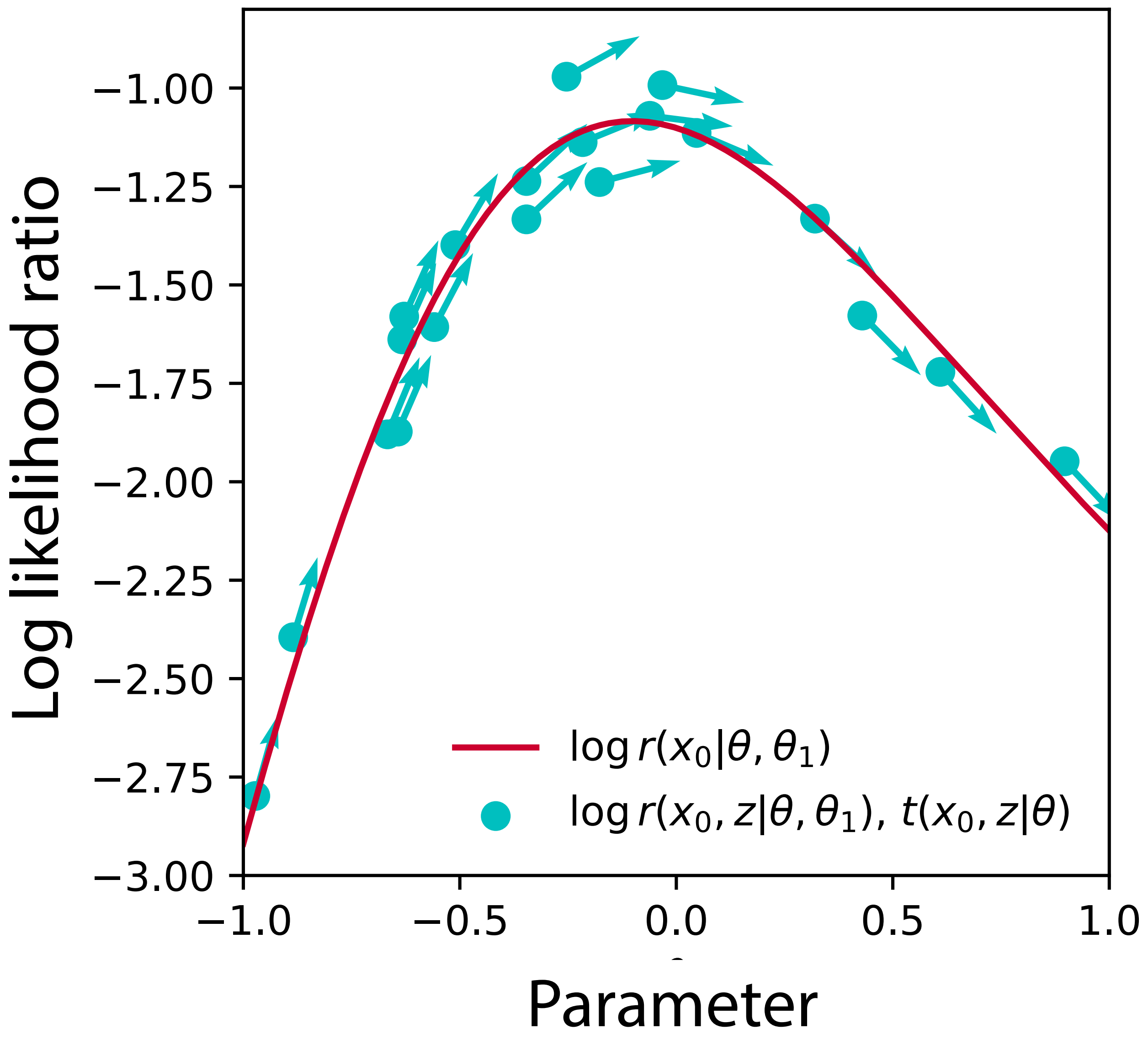
Likelihood ratio trick



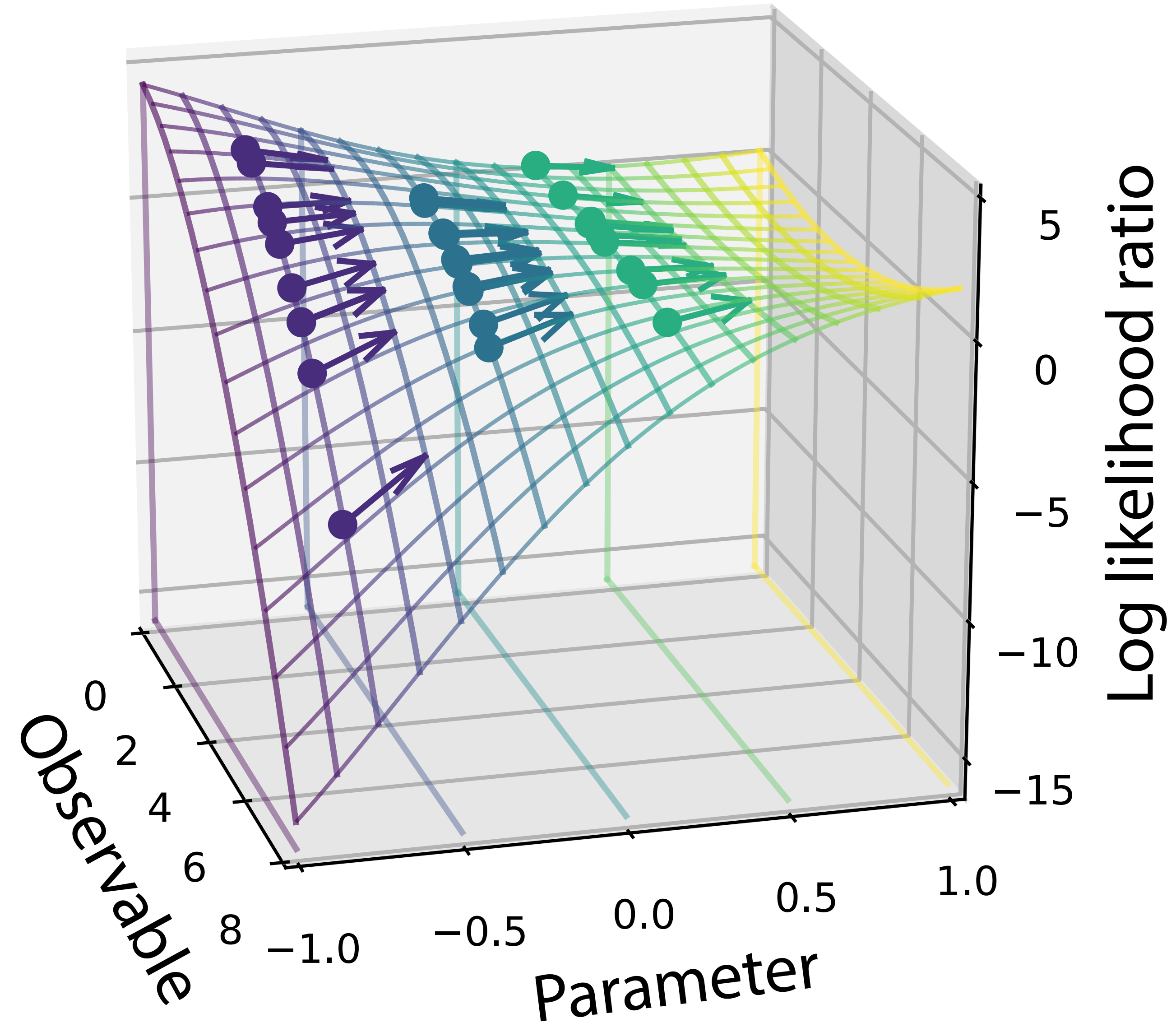
+ joint likelihood ratio



+ joint score



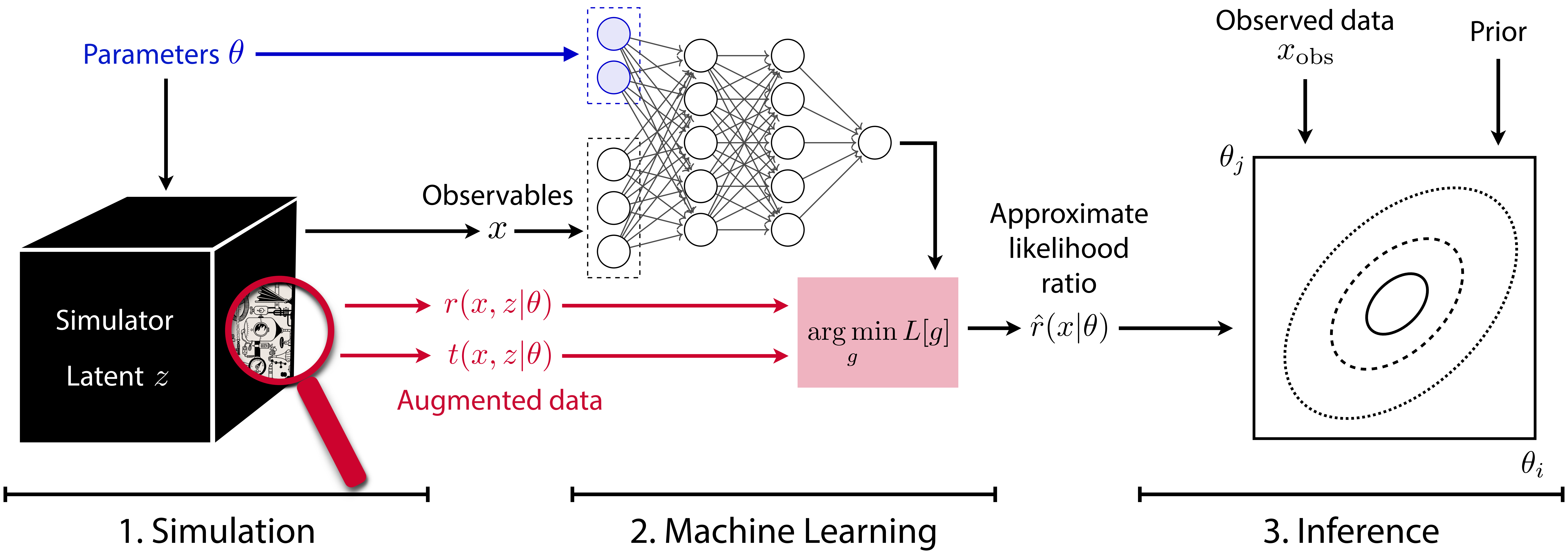
= RASCAL



Using more information = more sample-efficient inference

RASCAL: Likelihood ratio trick + gold mining

[JB, G. Louppe, J. Pavez, K. Cranmer
1805.12244, 1805.00013, 1805.00020]

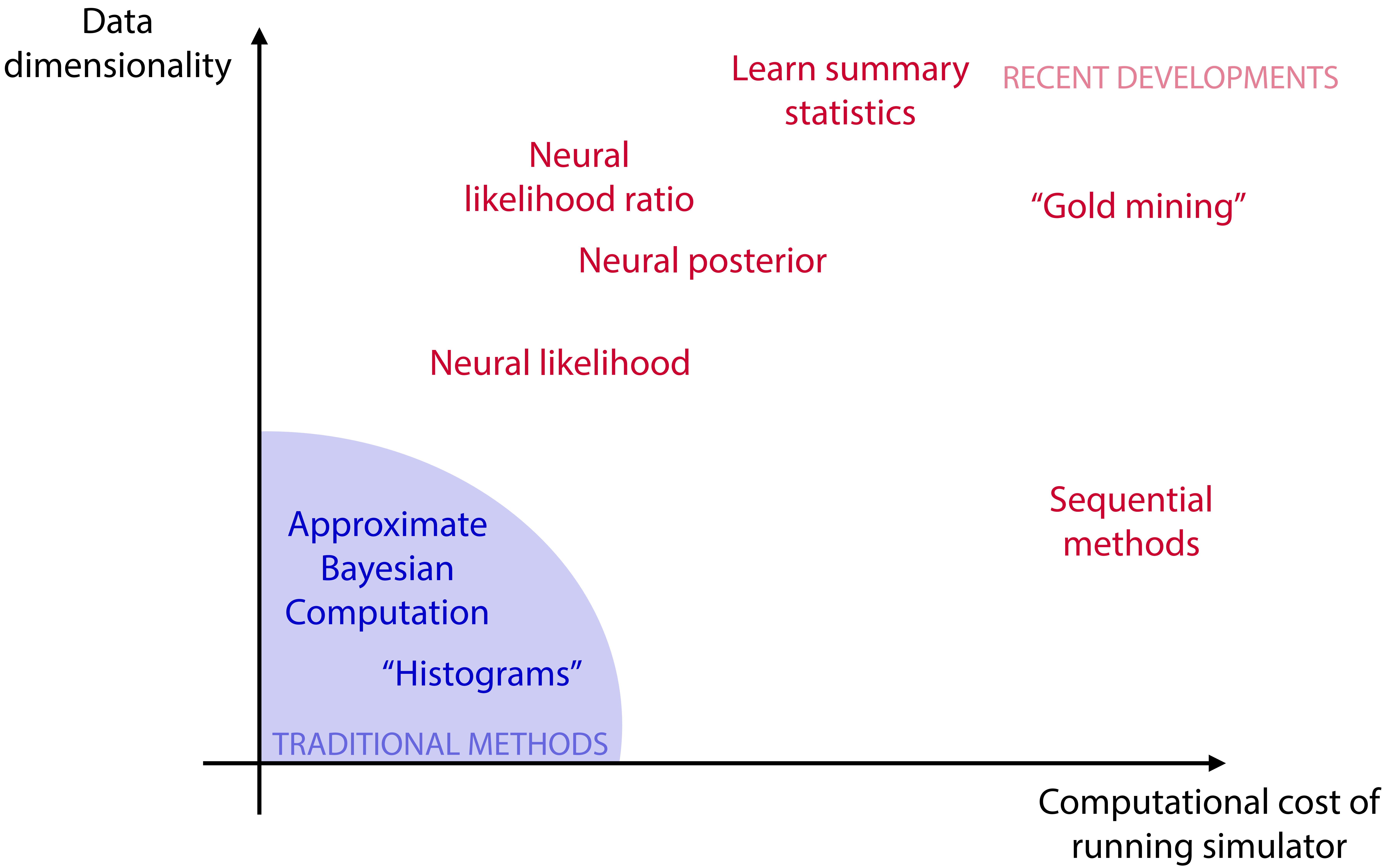


Extract joint likelihood ratio
and joint score from simulator

Augment training data &
use as labels in new loss functions
⇒ improve training efficiency

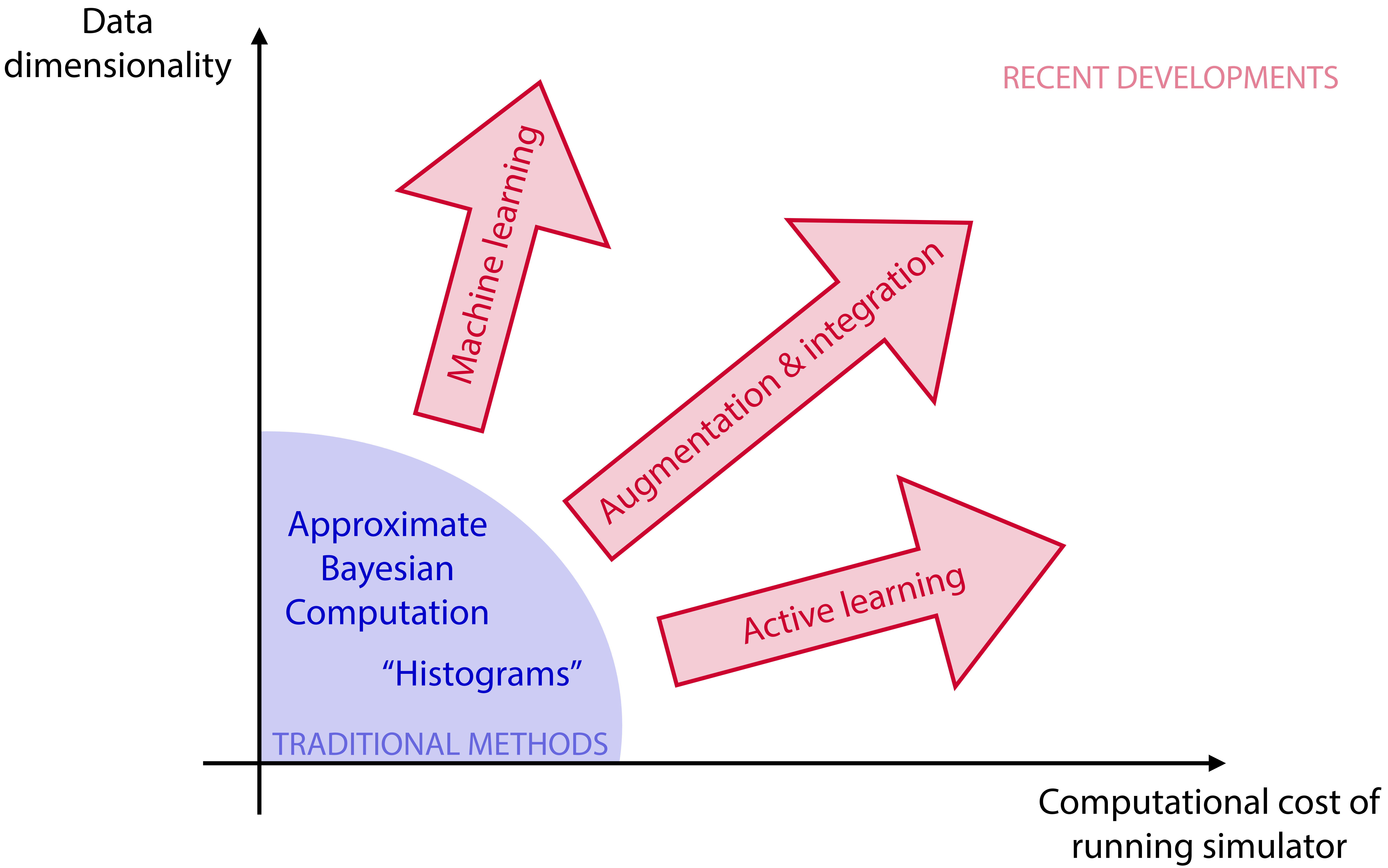
The frontier of simulation-based inference

[K. Cranmer, JB, G. Louppe 1911.01429]



The frontier of simulation-based inference

[K. Cranmer, JB, G. Louppe 1911.01429]



The

[K. Cranmer, JB, G. Louppe 1911.01429]

Check for updates

COLLOQUIUM PAPER

The frontier of simulation-based inference

Kyle Cranmer[a,b,1], Johann Brehmer[a,b], and Gilles Louppe[c]

[a]Center for Cosmology and Particle Physics, New York University, New York, NY 10003; [b]Center for Data Science, New York University, New York, NY 10011; and [c]Montefiore Institute, University of Liège, B-4000 Liège, Belgium

Edited by Jitendra Malik, University of California, Berkeley, CA, and approved April 10, 2020 (received for review November 4, 2019)

Many domains of science have developed complex simulations to describe phenomena of interest. While these simulations provide high-fidelity models, they are poorly suited for inference and lead to challenging inverse problems. We review the rapidly developing field of simulation-based inference and identify the forces giving additional momentum to the field. Finally, we describe how the frontier is expanding so that a broad audience can appreciate the profound influence these developments may have on science.

statistical inference | implicit models | likelihood-free inference | approximate Bayesian computation | neural density estimation

Mechanistic models can be used to predict how systems will behave in a variety of circumstances. These run the gamut of distance scales, with notable examples including particle physics, molecular dynamics, protein folding, population genetics, neuroscience, epidemiology, economics, ecology, climate science, astrophysics, and cosmology. The expressiveness of programming languages facilitates the development of complex, high-fidelity simulations and the power of modern computing provides the ability to generate synthetic data from them. Unfortunately, these simulators are poorly suited for statistical inference. The source of the challenge is that the probability density (or likelihood) for a given observation—an essential ingredient for both frequentist and Bayesian inference methods—is typically intractable. Such models are often referred to as implicit models and contrasted against prescribed models where the likelihood for an observation can be explicitly calculated (1). The problem setting of statistical inference under intractable likelihoods has been dubbed likelihood-free inference—although it is a bit of a misnomer as typically one attempts to estimate the intractable likelihood, so we feel the term simulation-based inference is more apt.

The intractability of the likelihood is an obstruction for scientific progress as statistical inference is a key component of the scientific method. In areas where this obstruction has appeared, scientists have developed various ad hoc or field-specific methods to overcome it. In particular, two common traditional approaches rely on scientists to use their insight into the system to construct powerful summary statistics and then compare the observed data to the simulated data. In the first one, density estimation methods are used to approximate the distribution of

the simulator—is being recognized as a key idea to improve the sample efficiency of various inference methods. A third direction of research has stopped treating the simulator as a black box and focused on integrations that allow the inference engine to tap into the internal details of the simulator directly.

Amidst this ongoing revolution, the landscape of simulation-based inference is changing rapidly. In this review we aim to provide the reader with a high-level overview of the basic ideas behind both old and new inference techniques. Rather than discussing the algorithms in technical detail, we focus on the current frontiers of research and comment on some ongoing developments that we deem particularly exciting.

Simulation-Based Inference

Simulators. Statistical inference is performed within the context of a statistical model, and in simulation-based inference the simulator itself defines the statistical model. For the purpose of this paper, a simulator is a computer program that takes as input a vector of parameters θ, samples a series of internal states or latent variables $z_i \sim p_i(z_i|\theta, z_{<i})$, and finally produces a data vector $x \sim p(x|\theta, z)$ as output. Programs that involve random samplings and are interpreted as statistical models are known as probabilistic programs, and simulators are an example. Within this general formulation, real-life simulators can vary substantially:

- The parameters θ describe the underlying mechanistic model and thus affect the transition probabilities $p_i(z_i|\theta, z_{<i})$. Typically the mechanistic model is interpretable by a domain scientist and θ has relatively few components and a fixed dimensionality. Examples include coefficients found in the Hamiltonian of a physical system, the virulence and incubation rate of a pathogen, or fundamental constants of Nature.
- The latent variables z that appear in the data-generating process may directly or indirectly correspond to a physically meaningful state of a system, but typically this state is unobservable in practice. The structure of the latent space varies substantially between simulators. The latent variables may be continuous or discrete and the dimensionality of the latent space may be fixed or may vary, depending on the control flow of the simulator. The simulation can freely combine deterministic and stochastic steps. The deterministic components of the simulator may be differentiable or may involve discontinuous control

STATISTICS

NT DEVELOPMENTS

mputational cost of running simulator

The

[K. Cranmer, JB, G. Louppe 1911.01429]

The frontier of simulation-based inference

Kyle Cranmer[a,b,1], Johann Brehmer[a,b], and Gilles Louppe[c]

[a]Center for Cosmology and Particle Physics, New York University, New York, NY 10003; [b]Center for Data Science, New Yo
and [c]Montefiore Institute, University of Liège, B-4000 Liège, Belgium

Edited by Jitendra Malik, University of California, Berkeley, CA, and approved April 10, 2020 (received for review Novem

Many domains of science have developed complex simulations to describe phenomena of interest. While these simulations provide high-fidelity models, they are poorly suited for inference and lead to challenging inverse problems. We review the rapidly developing field of simulation-based inference and identify the forces giving additional momentum to the field. Finally, we describe how the frontier is expanding so that a broad audience can appreciate the profound influence these developments may have on science.

statistical inference | implicit models | likelihood-free inference | approximate Bayesian computation | neural density estimation

Mechanistic models can be used to predict how systems will behave in a variety of circumstances. These run the gamut of distance scales, with notable examples including particle physics, molecular dynamics, protein folding, population genetics, neuroscience, epidemiology, economics, ecology, climate science, astrophysics, and cosmology. The expressiveness of programming languages facilitates the development of complex, high-fidelity simulations and the power of modern computing provides the ability to generate synthetic data from them. Unfortunately, these simulators are poorly suited for statistical inference. The source of the challenge is that the probability density (or likelihood) for a given observation—an essential ingredient for both frequentist and Bayesian inference methods—is typically intractable. Such models are often referred to as implicit models and contrasted against prescribed models where the likelihood for an observation can be explicitly calculated (1). The problem setting of statistical inference under intractable likelihoods has been dubbed likelihood-free inference—although it is a bit of a misnomer as typically one attempts to estimate the intractable likelihood, so we feel the term simulation-based inference is more apt.

The intractability of the likelihood is an obstruction for scientific progress as statistical inference is a key component of the scientific method. In areas where this obstruction has appeared, scientists have developed various ad hoc or field-specific methods to overcome it. In particular, two common traditional approaches rely on scientists to use their insight into the system to construct powerful summary statistics and then compare the observed data to the simulated data. In the first one, density estimation methods are used to approximate the distribution of

the simulator—is being recognize
sample efficiency of various infere
of research has stopped treating th
focused on integrations that allo
into the internal details of the sim

Amidst this ongoing revolution
based inference is changing rapi
provide the reader with a high-le
behind both old and new infere
discussing the algorithms in tech
current frontiers of research and
developments that we deem partic

Simulation-Based Inference

Simulators. Statistical inference is
of a statistical model, and in si
simulator itself defines the statis
of this paper, a simulator is a c
as input a vector of parameters θ
states or latent variables $z_i \sim p_i($
a data vector $x \sim p(x|\theta, z)$ as o
random samplings and are interp
known as probabilistic programs,
ple. Within this general formulatio
substantially:

- The parameters θ describe the
and thus affect the transition p
ically the mechanistic model i
scientist and θ has relatively
dimensionality. Examples inclu
Hamiltonian of a physical system
rate of a pathogen, or fundamen
- The latent variables z that appe
cess may directly or indirectly co
ingful state of a system, but typi
in practice. The structure of the l
between simulators. The latent
or discrete and the dimensional
fixed or may vary, depending o
ulator. The simulation can free
stochastic steps. The determinis
tor may be differentiable or may

Likelihood-Free Inference Workshop

18-22 March 2019 @ Flatiron Institute, NYC

Home Schedule Hackathon Logistics Participants Registration

Rationale

The goal of this interdisciplinary meeting is to gather developers and users of Likelihood-Free Inference methods to share latest techniques, use cases and applications across different fields, and discuss open challenges.

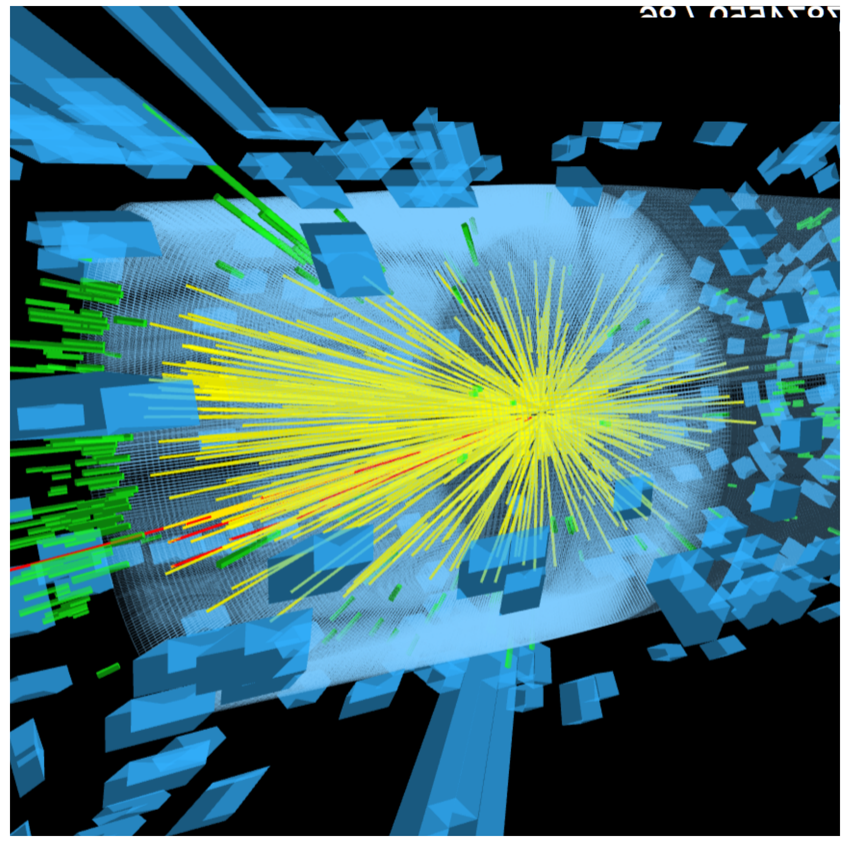
The first two days of the workshop will be focused on talks and discussions, while the remaining of the week will be dedicated to a hackathon with the goal of seeding the development of a common probabilistic programming framework for Likelihood-Free Inference as well as collaboratively working on LFI-related hack projects.

News

- *March 4th, 2019* : Preliminary schedule available, new Gitter channel chat on gitter, new Hackathon page
- *February 19th, 2019* : Main registration is closed, contact organizers for late registration
- *February 19th, 2019* : Travel funding application deadline
- *February 6th, 2019* : Opening registration

Organizing Committee

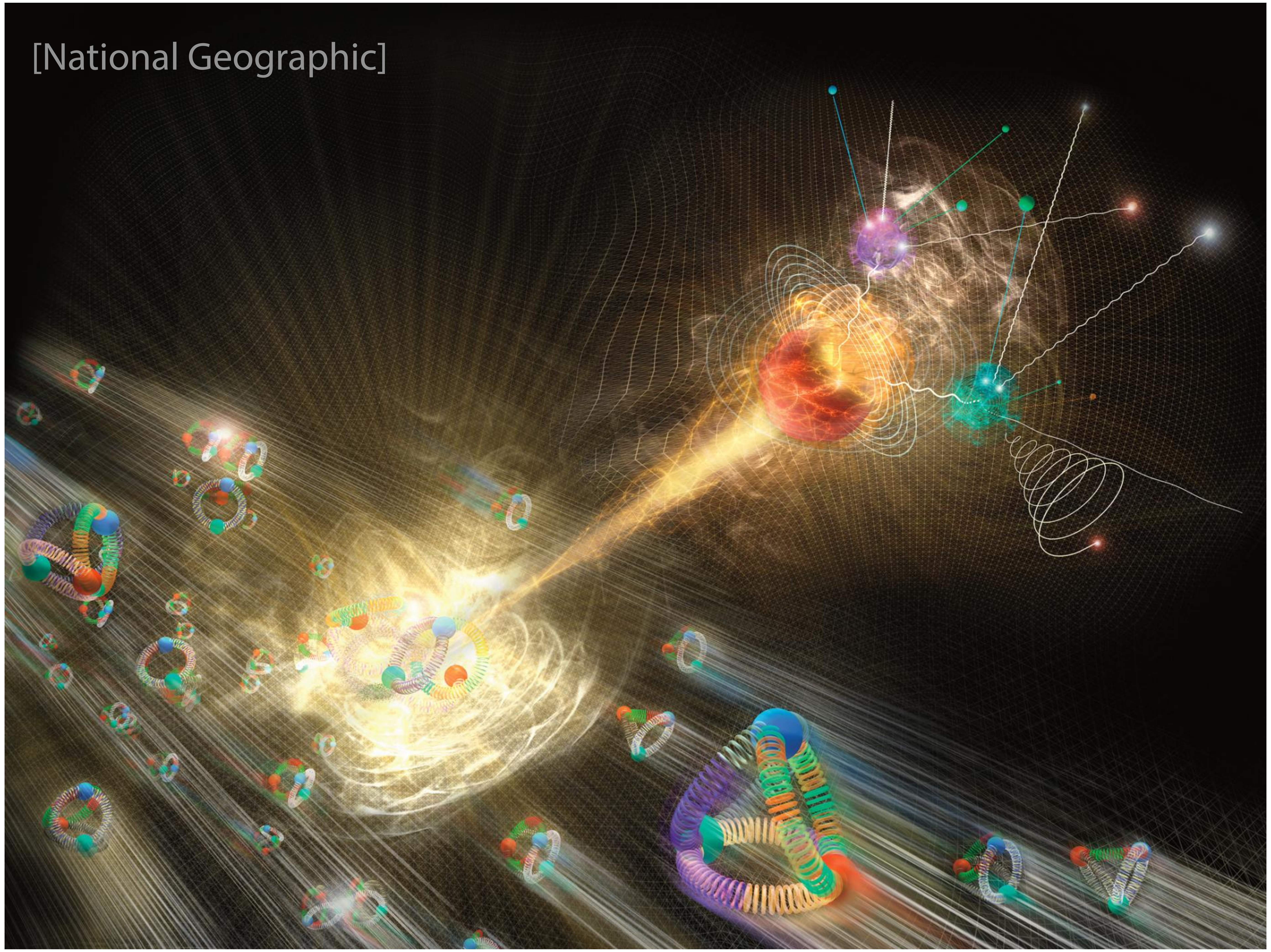
- Justin Alsing, Oskar Klein Center, Stockholm University
- Johann Brehmer, Center for Data Science, New York University
- Stephen Feeney, Center for Computational Astrophysics, Flatiron Institute



3. Particle physics

[Maximilien Brice, CERN]

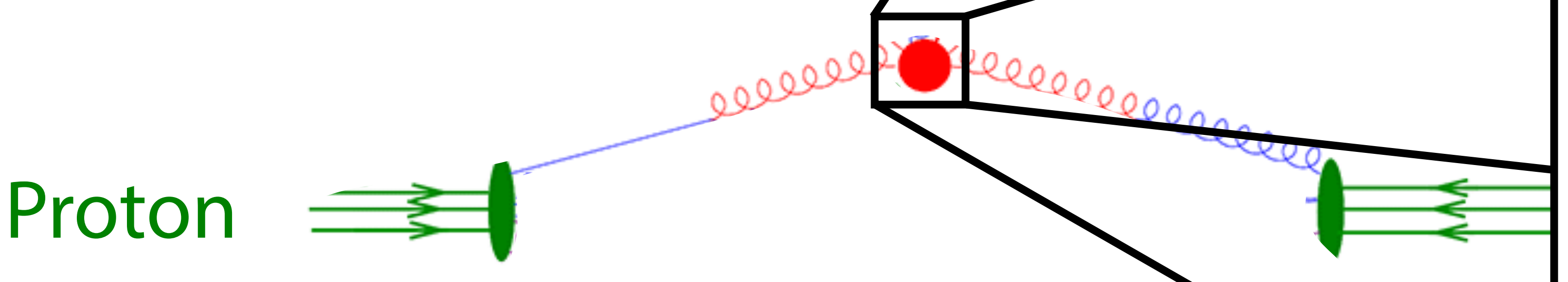
Proton collisions



Proton collisions

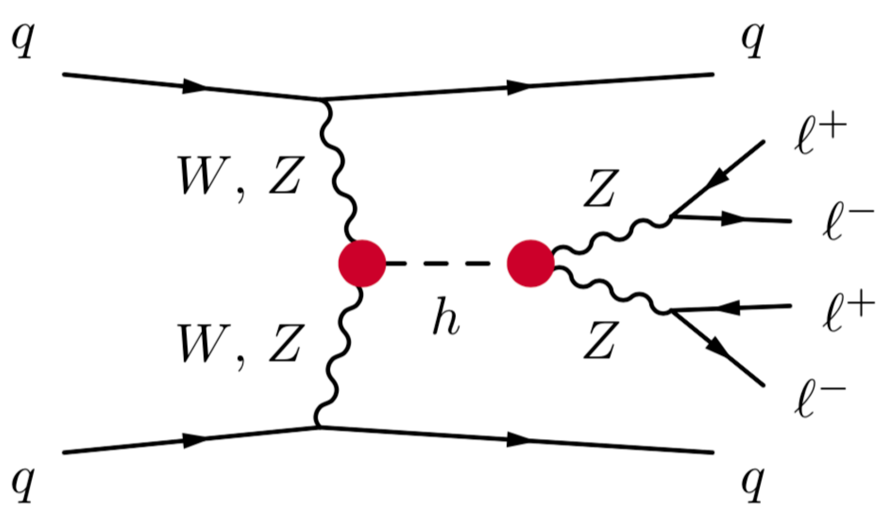
[F. Krauss]

Proton collisions



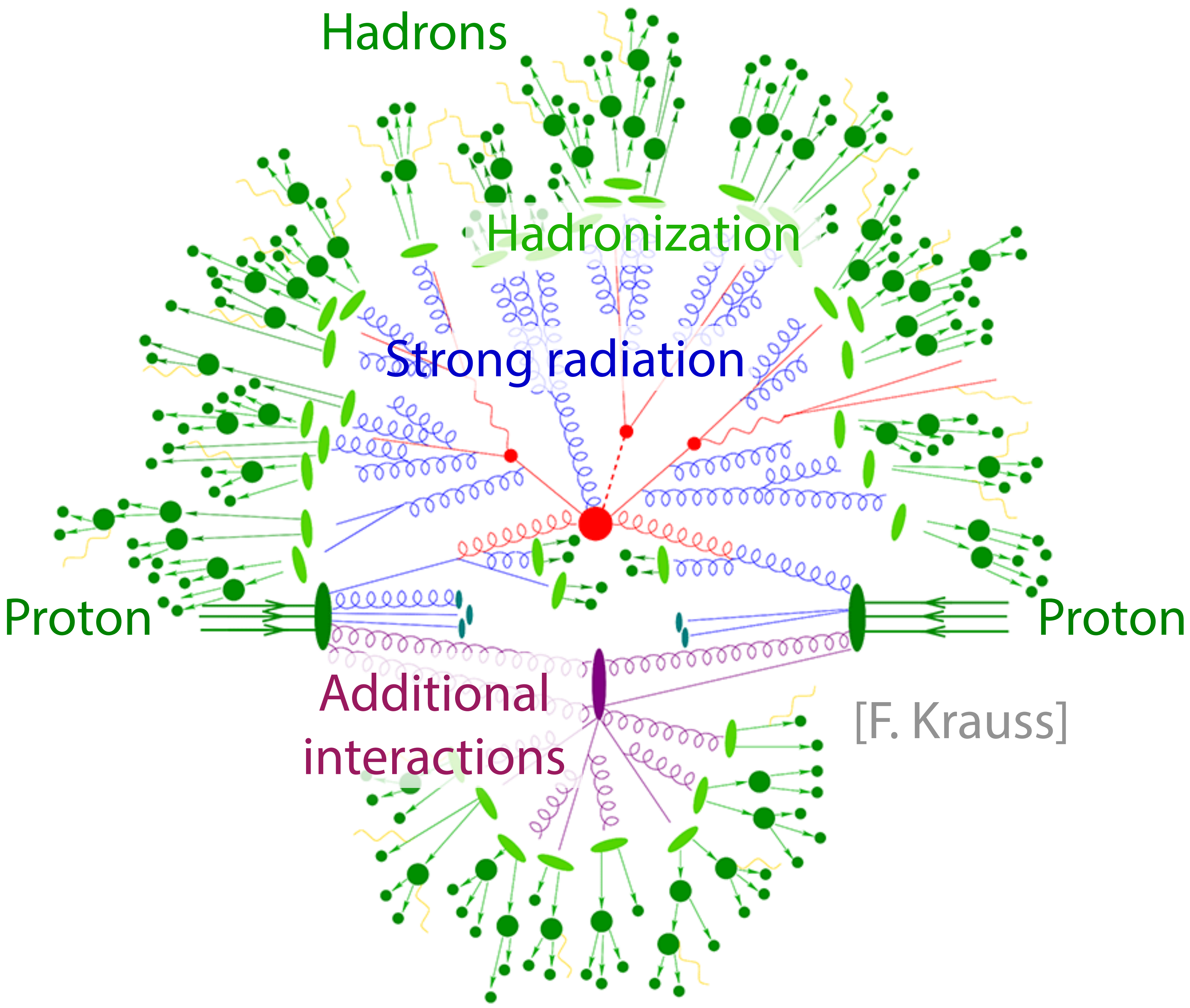
High-energy interactions:

- Rigorously described by quantum field theories
- Sensitive to new physics



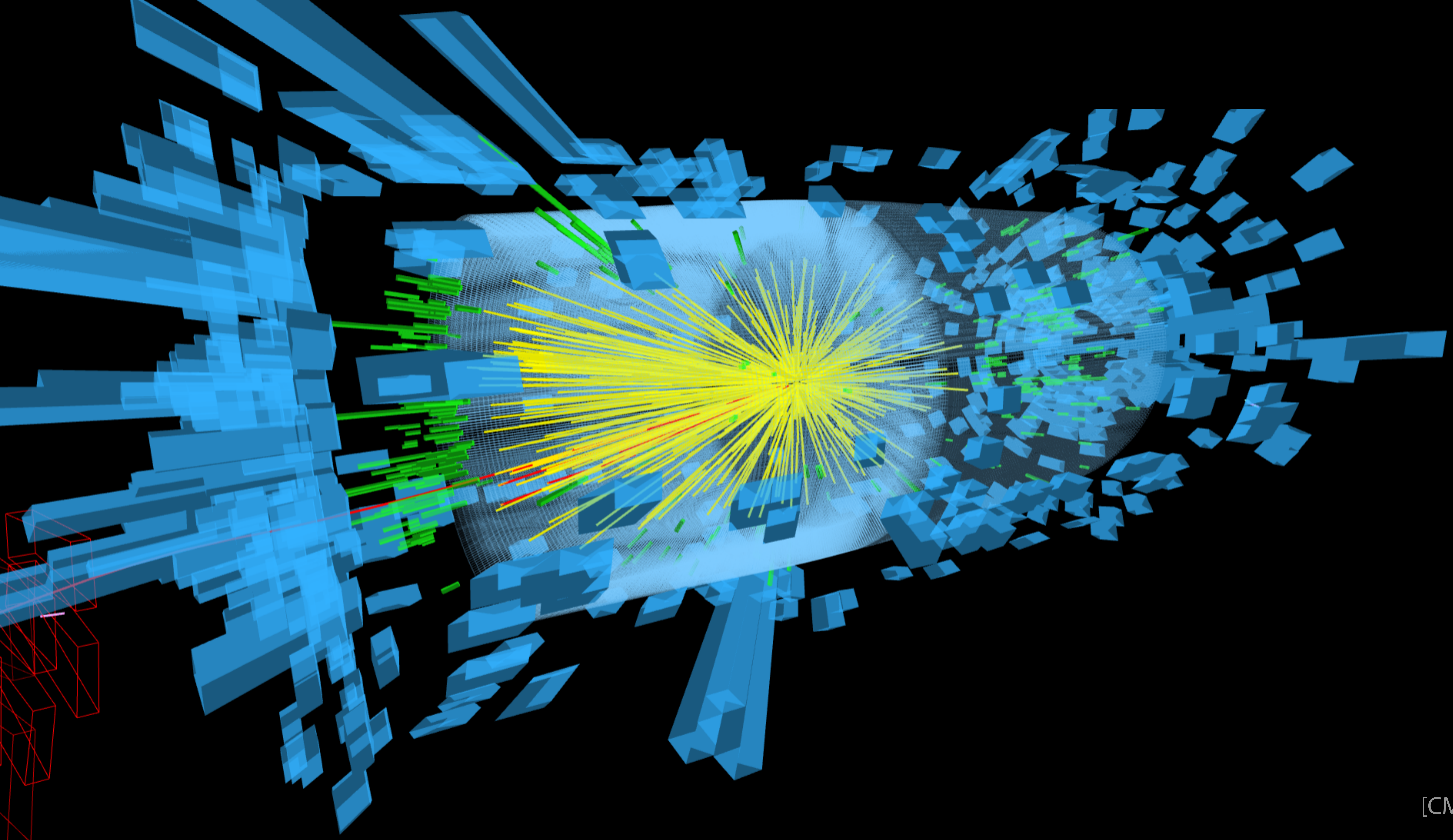
$$
\begin{aligned}
S = \int \mathrm{d}^4x \Bigg[&\mathcal{L}_{\mathrm{SM}} + \frac{f_{\phi,2}}{\Lambda^2}\,\frac{1}{2}\,\partial_\mu(\phi^\dagger\phi)\,\partial^\mu(\phi^\dagger\phi) + \frac{f_{\phi,3}}{\Lambda^2}\,\frac{1}{3}(\phi^\dagger\phi)^3 \\
&+ \frac{f_{GG}}{\Lambda^2}\,(\phi^\dagger\phi)\,G^a_{\mu\nu}\,G^{\mu\nu\,a} - \frac{f_{BB}}{\Lambda^2}\,\frac{g'^2}{4}\,(\phi^\dagger\phi)\,B_{\mu\nu}\,B^{\mu\nu} - \frac{f_{WW}}{\Lambda^2}\,\frac{g^2}{4}\,(\phi^\dagger\phi)\,W^a_{\mu\nu}\,W^{\mu\nu\,a} \\
&+ \frac{f_B}{\Lambda^2}\,\frac{\mathrm{i}g'}{2}\,(D^\mu\phi)^\dagger\,D^\nu\phi\,B_{\mu\nu} + \frac{f_W}{\Lambda^2}\,\frac{\mathrm{i}g}{2}\,(D^\mu\phi)^\dagger\,\sigma^a\,D^\nu\phi\,W^a_{\mu\nu} \\
&+ \frac{f_\ell}{\Lambda^2}\,(\phi^\dagger\phi)\,\bar{L}_L\,\phi\,\ell_R + \frac{f_u}{\Lambda^2}\,(\phi^\dagger\phi)\,\bar{Q}_L\,\tilde{\phi}\,u_R + \frac{f_d}{\Lambda^2}\,(\phi^\dagger\phi)\,\bar{Q}_L\,\phi\,d_R \\
&+ \frac{f_{G\widetilde{G}}}{\Lambda^2}\,(\phi^\dagger\phi)\,G^a_{\mu\nu}\,\widetilde{G}^{\mu\nu\,a} - \frac{f_{B\widetilde{B}}}{\Lambda^2}\,\frac{g'^2}{4}(\phi^\dagger\phi)\,B_{\mu\nu}\,\widetilde{B}^{\mu\nu} - \frac{f_{W\widetilde{W}}}{\Lambda^2}\,\frac{g^2}{4}(\phi^\dagger\phi)\,W^a_{\mu\nu}\,\widetilde{W}^{\mu\nu\,a} \Bigg]
\end{aligned}
$$

Proton collisions



[ATLAS]


[ATLAS]
ATLAS



[CMS]

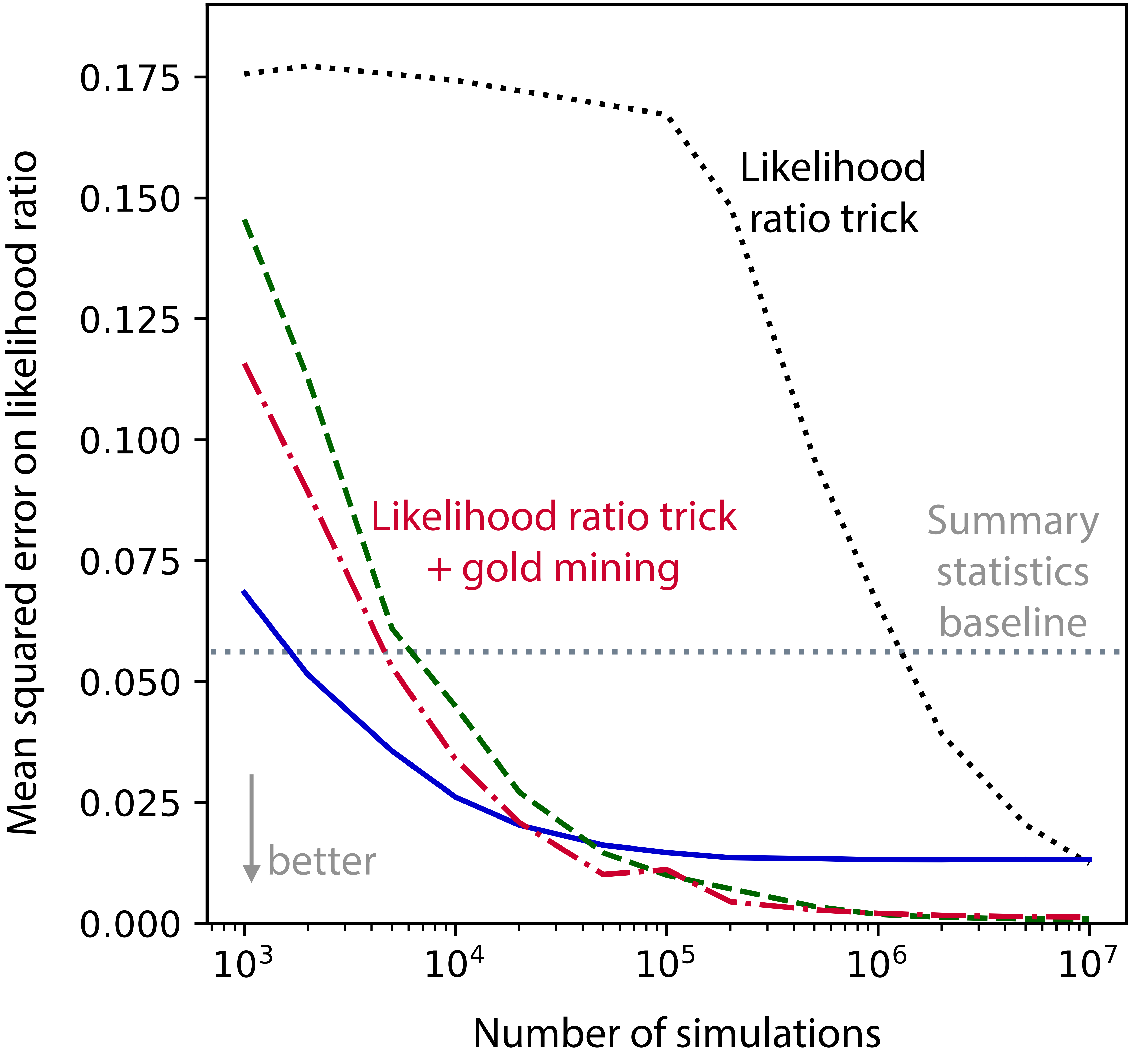
Understanding the Higgs boson

[CMS; D. Balibouse; nobelprize.org]

Now, precision measurements of Higgs properties and interactions may help us answer big questions:

- Why is gravity is so weak?
- Why are some particles so light and others so heavy?
- Why is there more matter than antimatter?
- Is the vacuum stable?

Improving sample efficiency

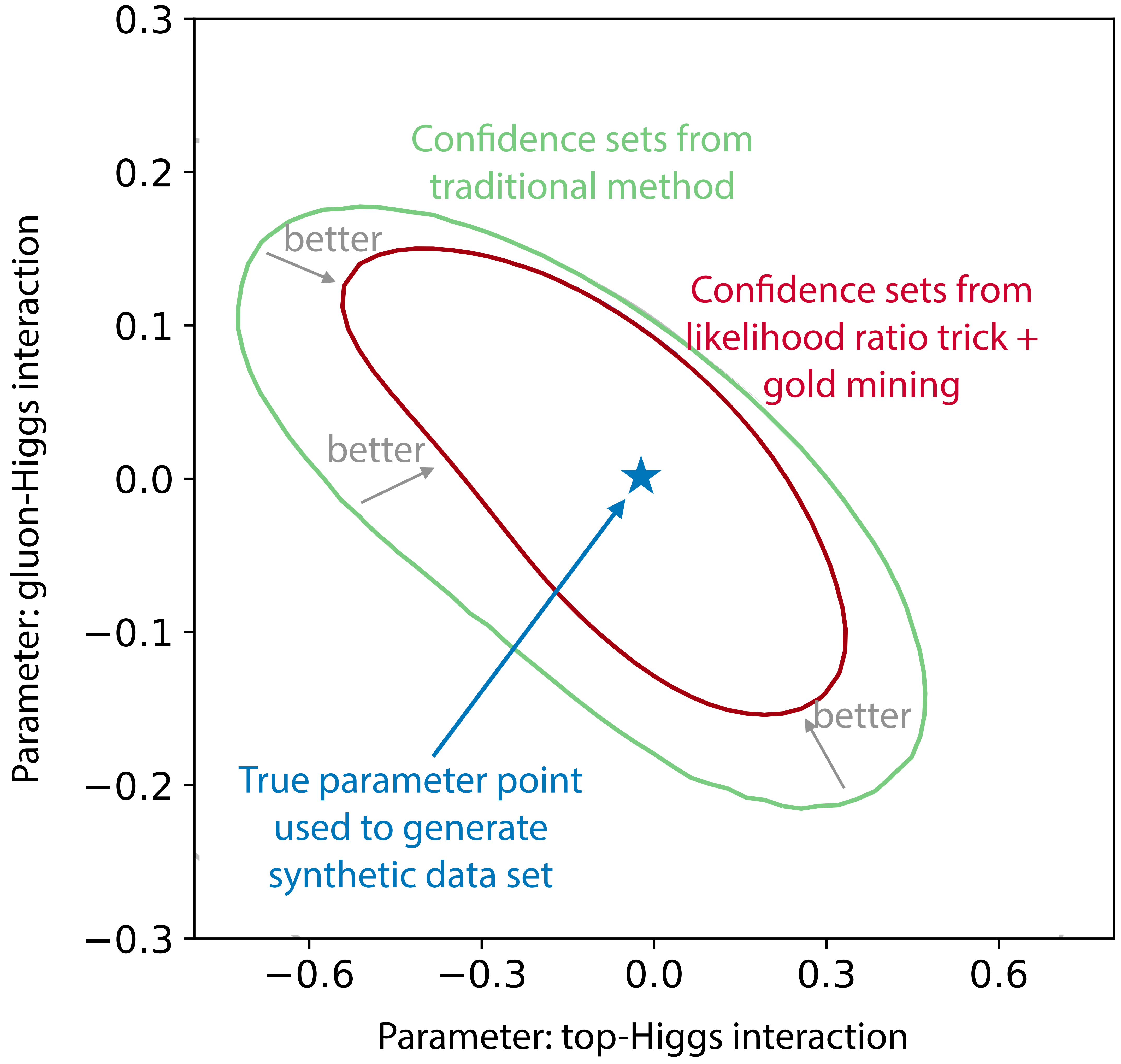


With enough training data, the ML algorithms get the likelihood function right.

Using more information from the simulator improves sample efficiency substantially.

[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013; 1805.00020; M. Stoye, JB, K. Cranmer, G. Louppe, J. Pavez 1808.00973]

Improving quality of inference



In some processes, the ML-based inference techniques improve the precision as much as taking 90% more data would!

[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013; 1805.00020; JB, F. Kling, I. Espejo, K. Cranmer 1907.10621]

Automation

[JB, F. Kling, I. Espejo, K. Cranmer 1907.10621]

Our open-source Python package MadMiner makes it straightforward to apply these ML-based inference techniques

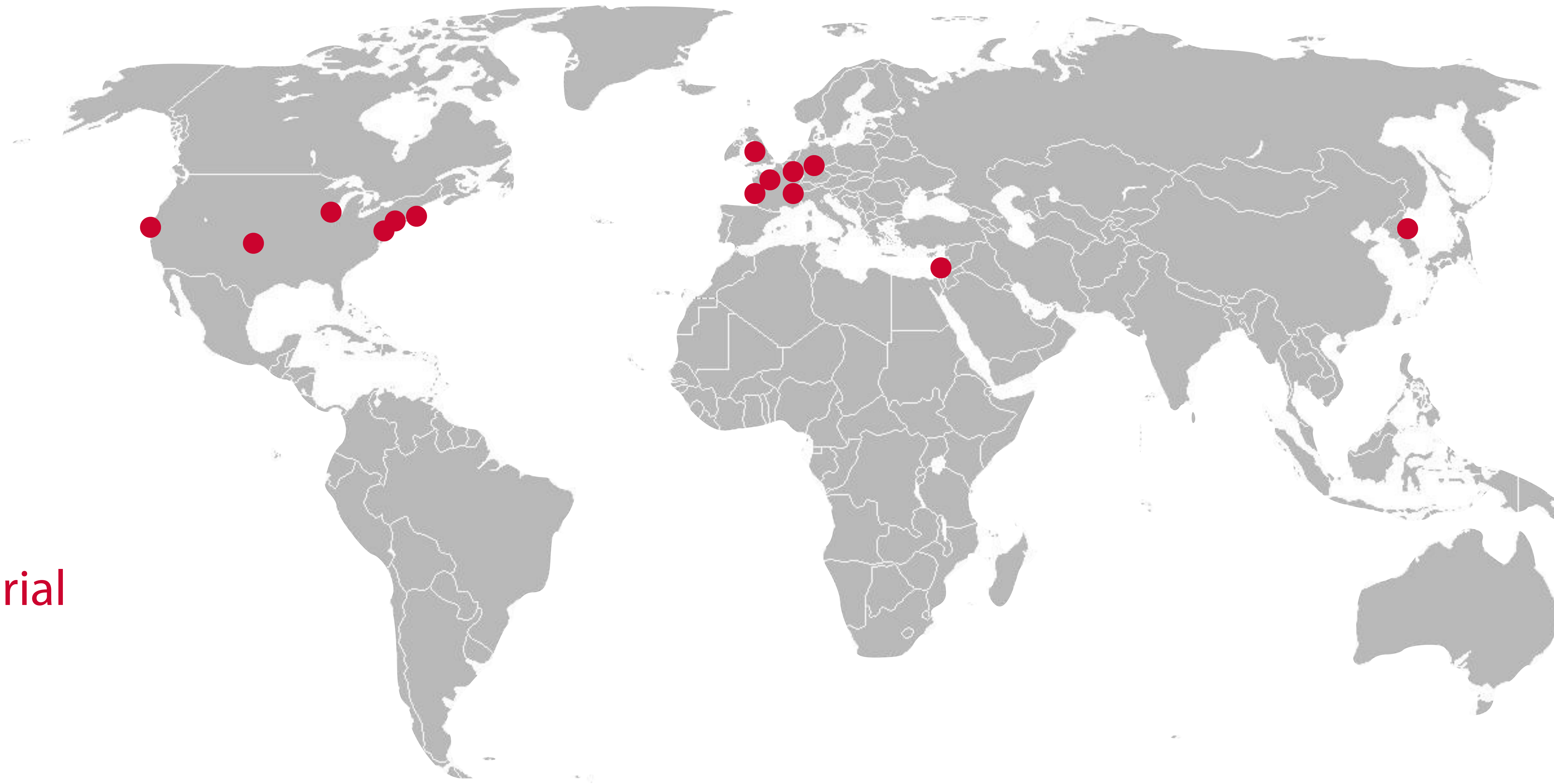
github.com/diana-hep/madminer

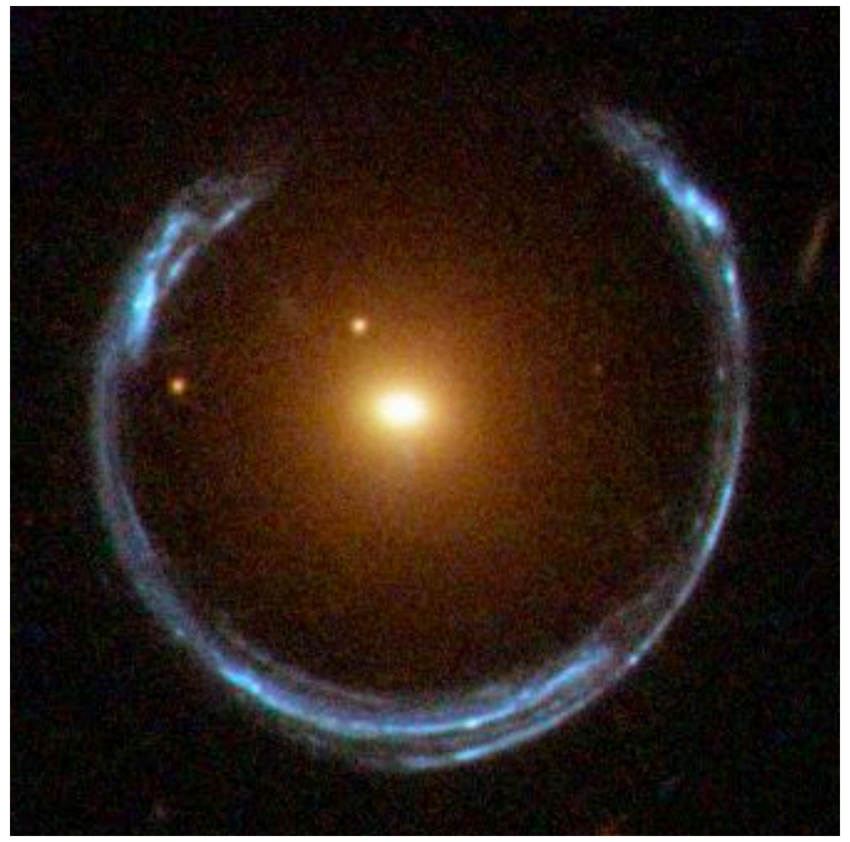
`pip install madminer`

hub.docker.com/u/madminertool

cranmer.github.io/madminer-tutorial

madminer.readthedocs.io





4. Astrophysics

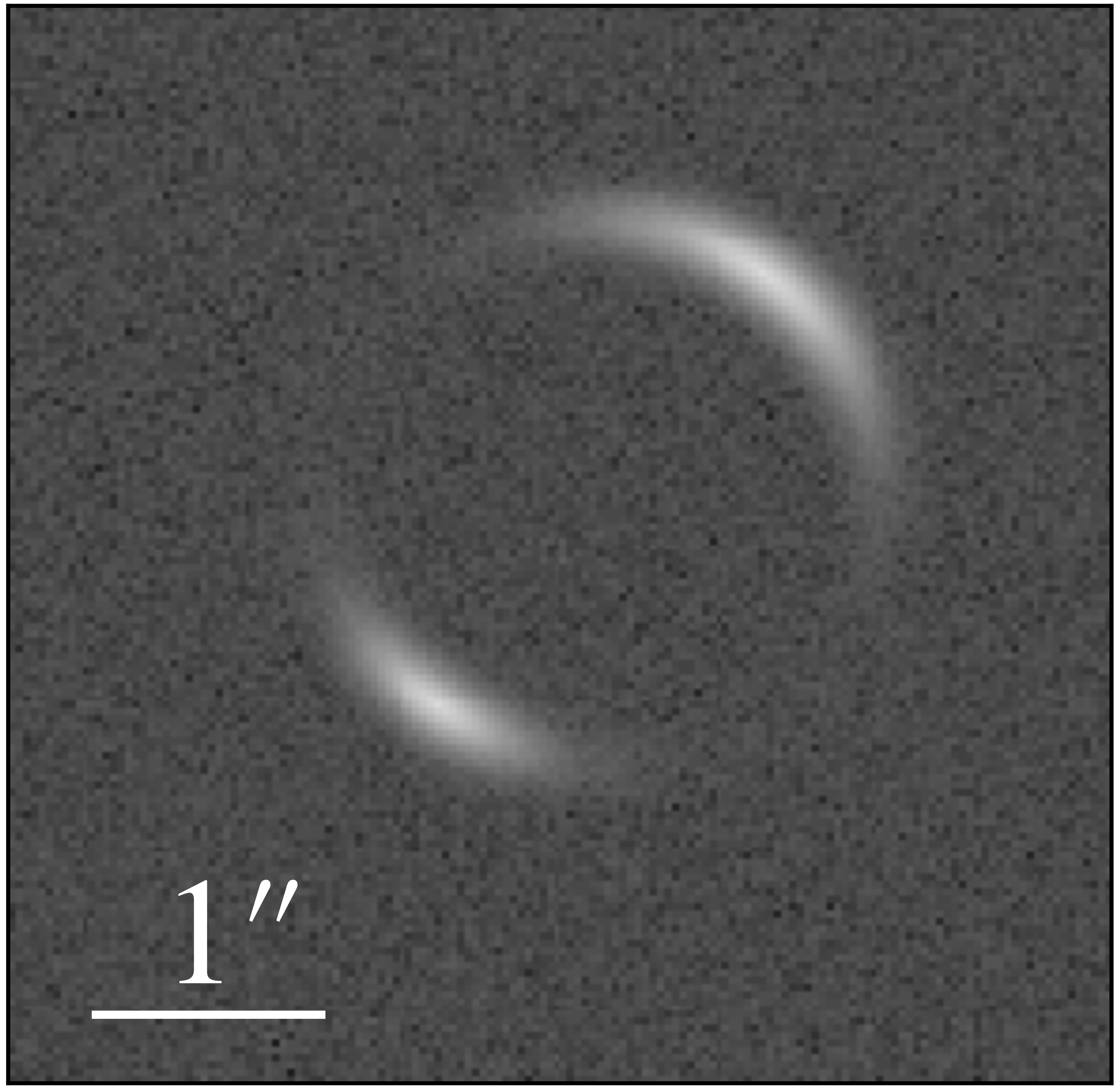

[T. Brown, J.Tumlinson]

[NASA/ESA; Abe Books]

[Hubble Space Telescope]

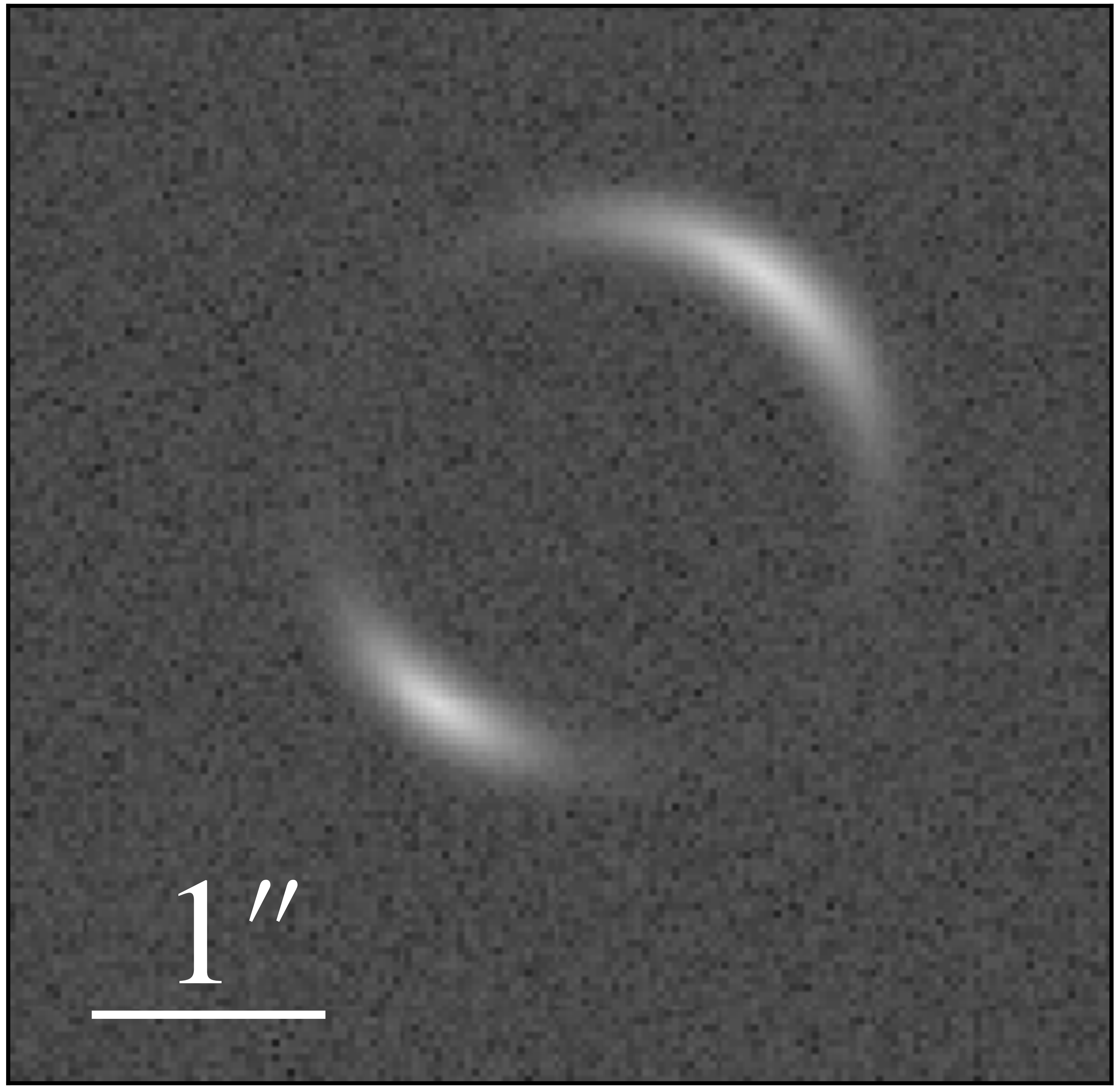
Subhalos affect strong lensing

Smooth halo only

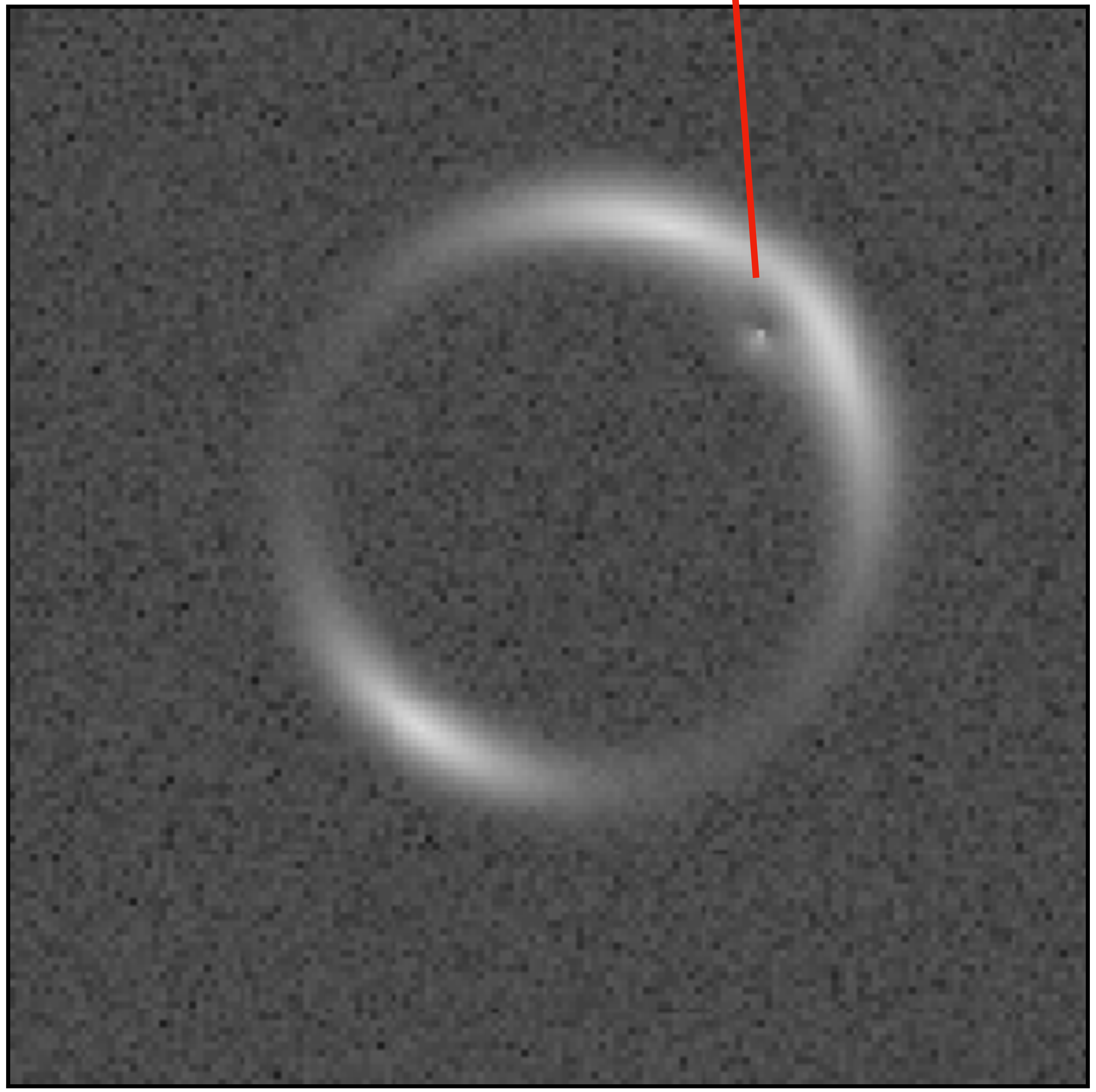


Subhalos affect strong lensing

Smooth halo only

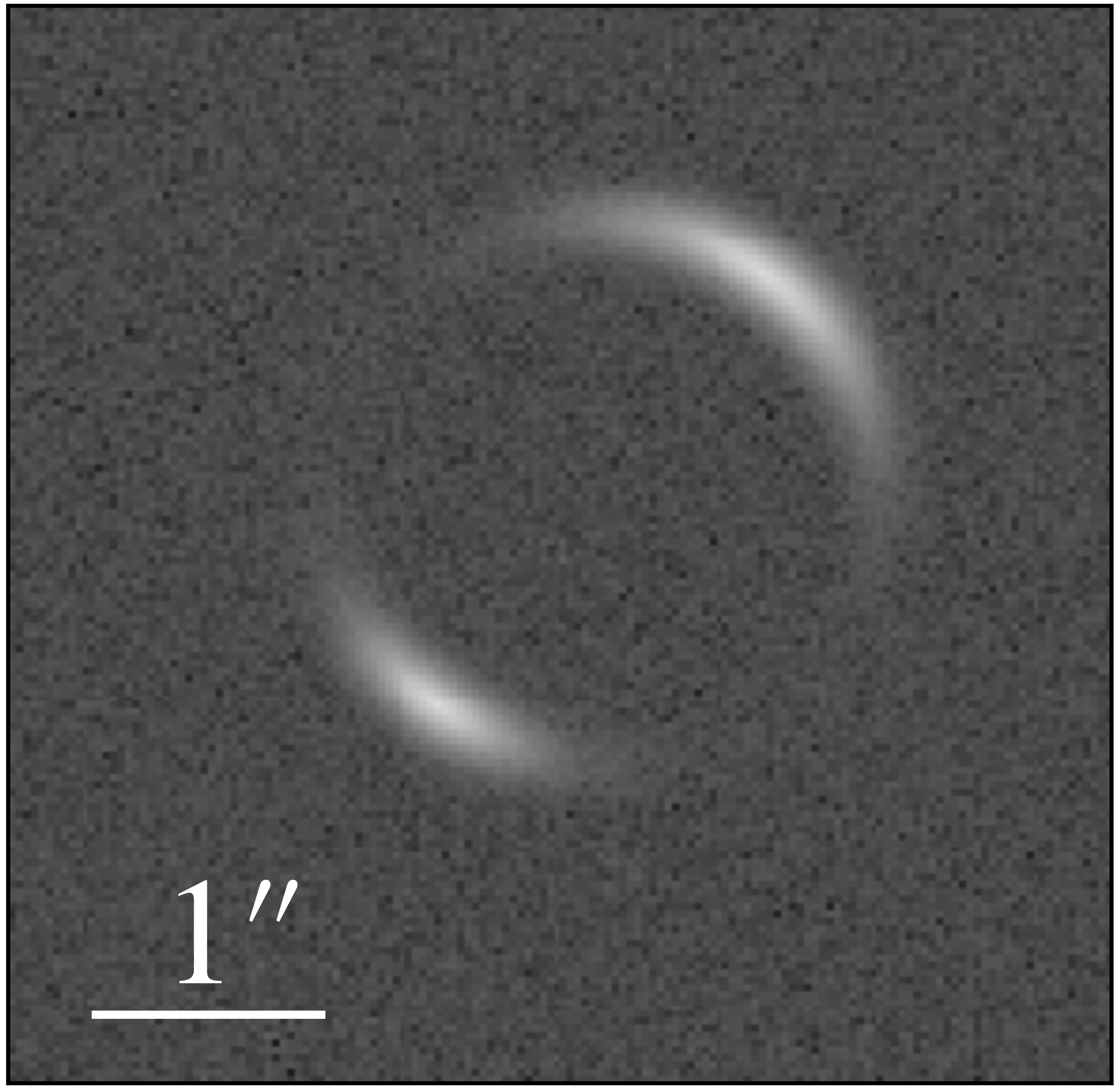


Smooth halo + subhalo

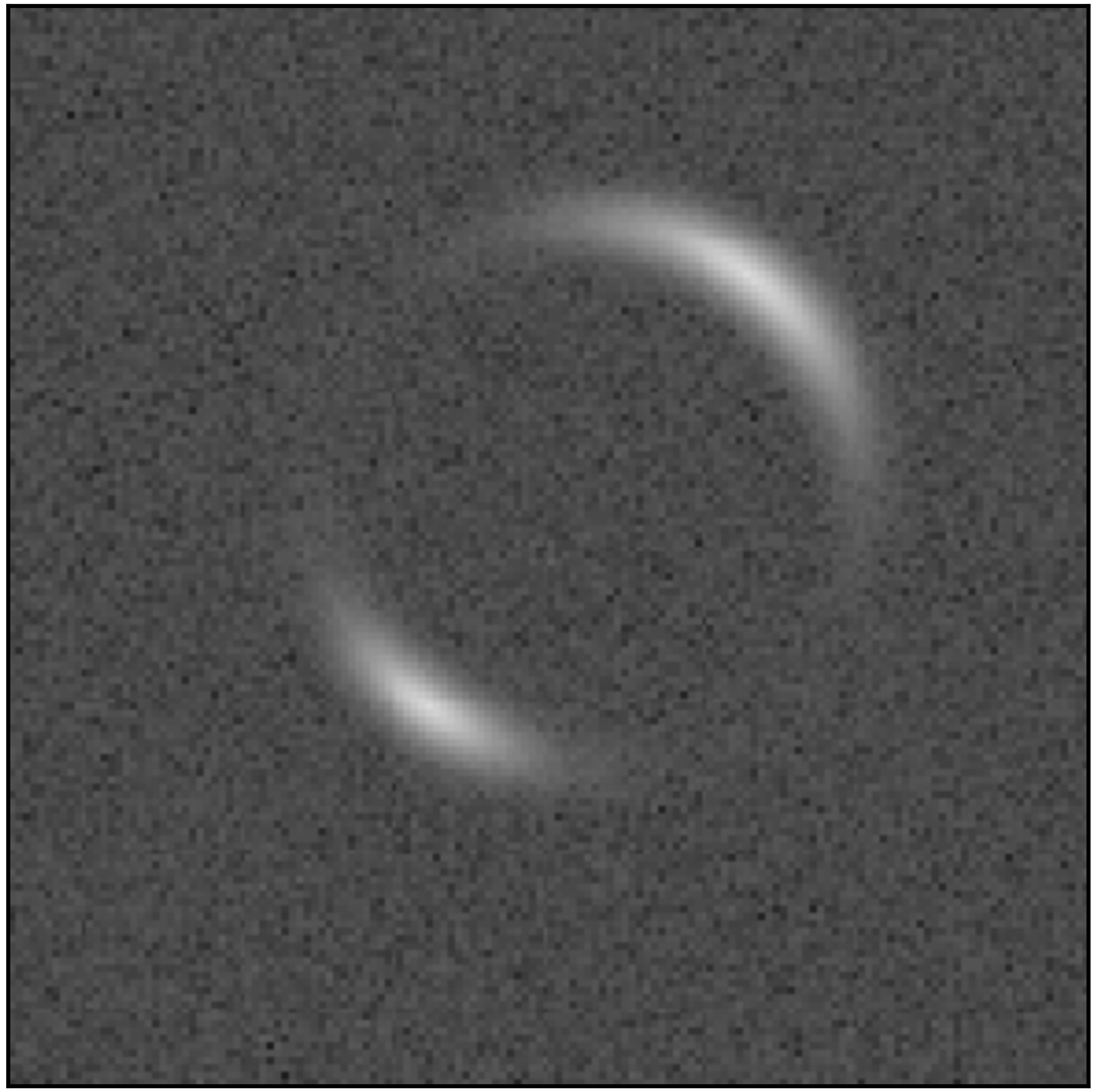


Subhalos affect strong lensing… realistically

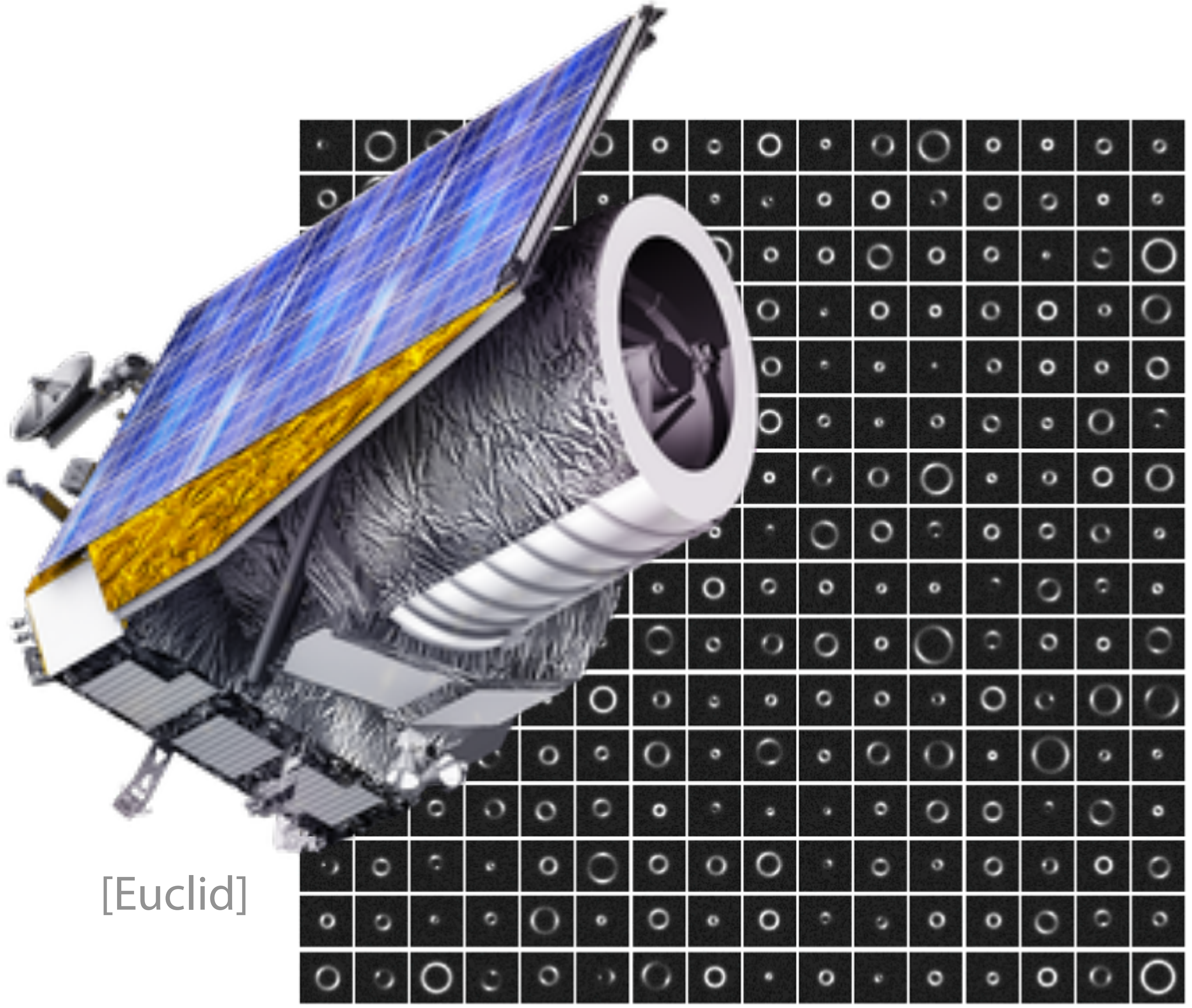
Smooth halo only

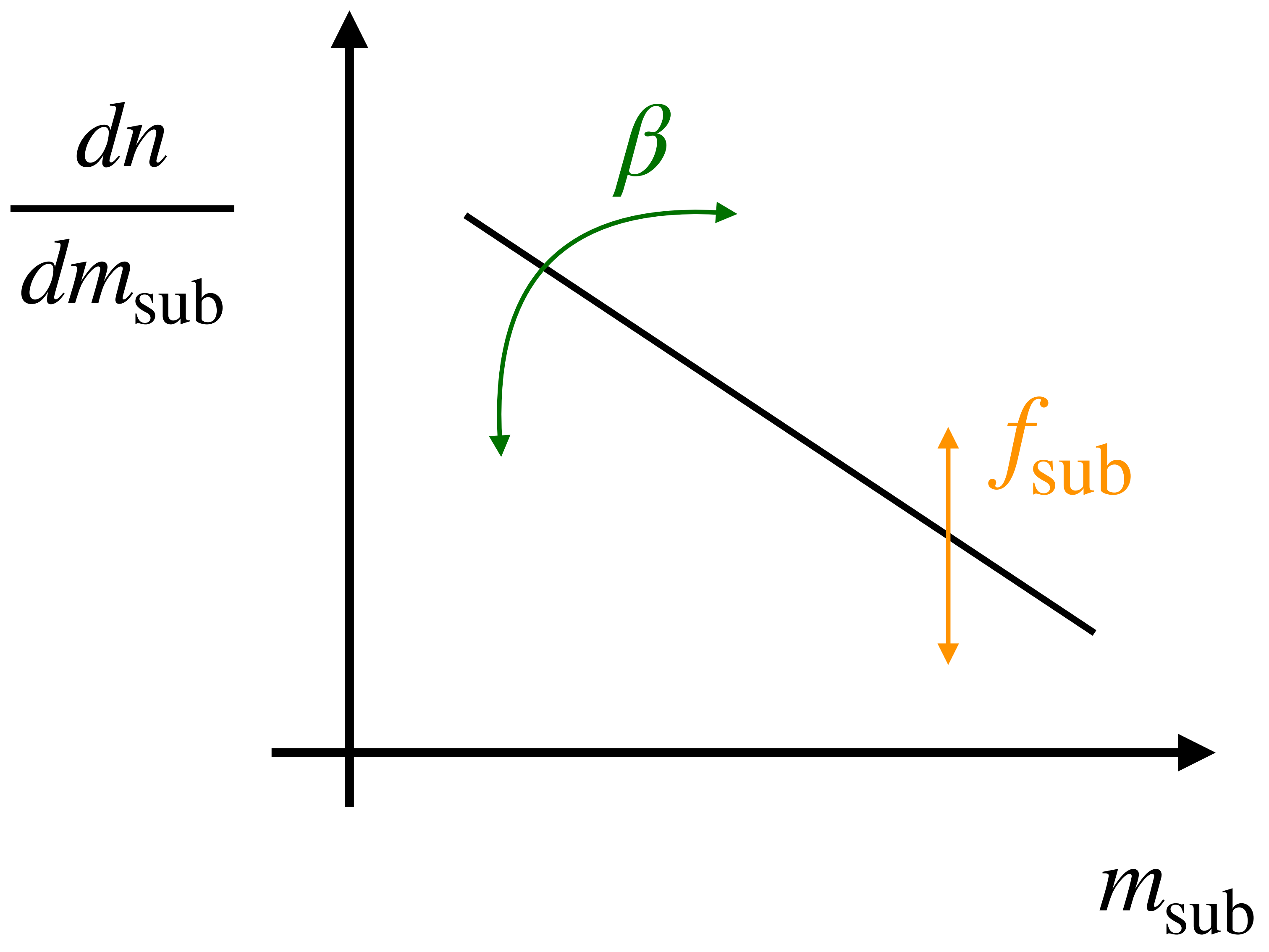


Smooth halo + subhalos



Scalable inference for small subhalos



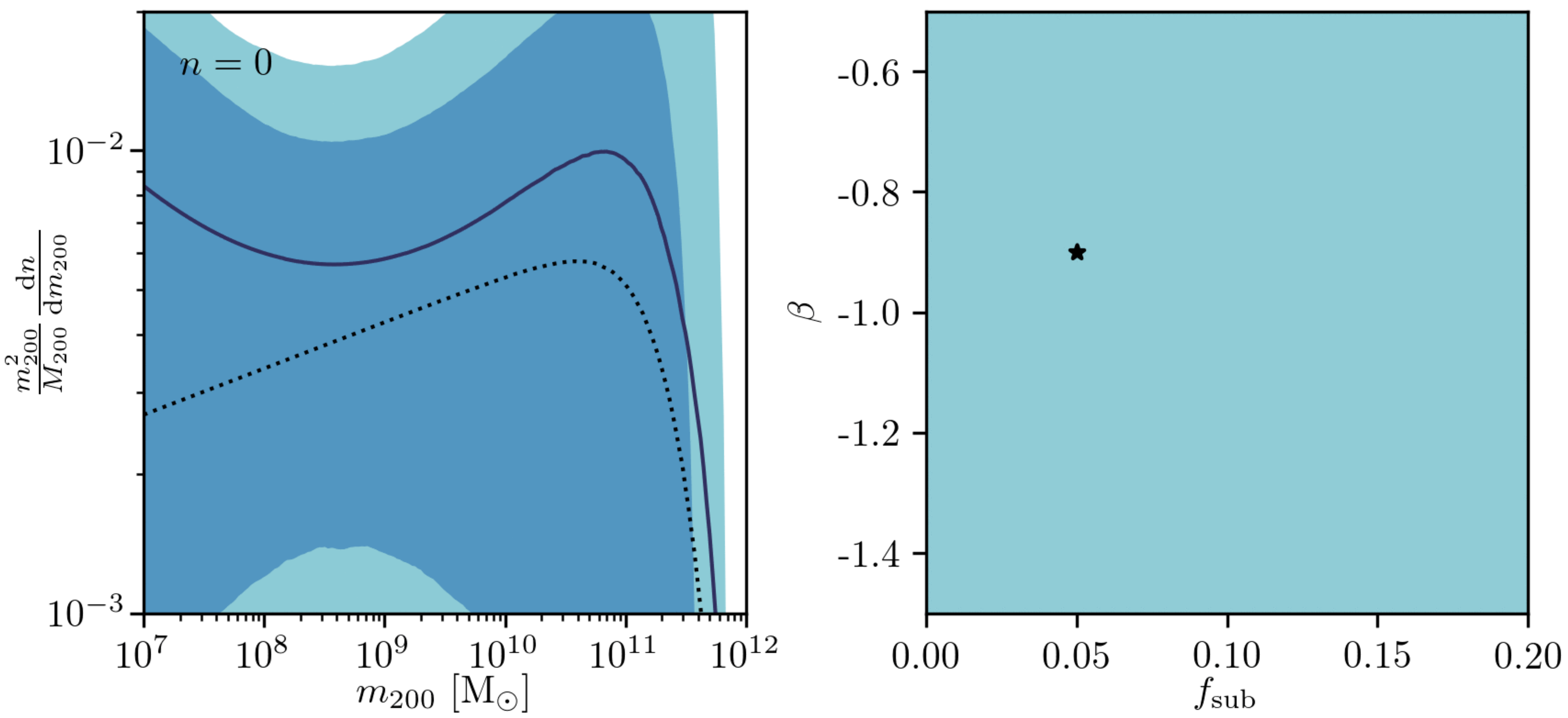


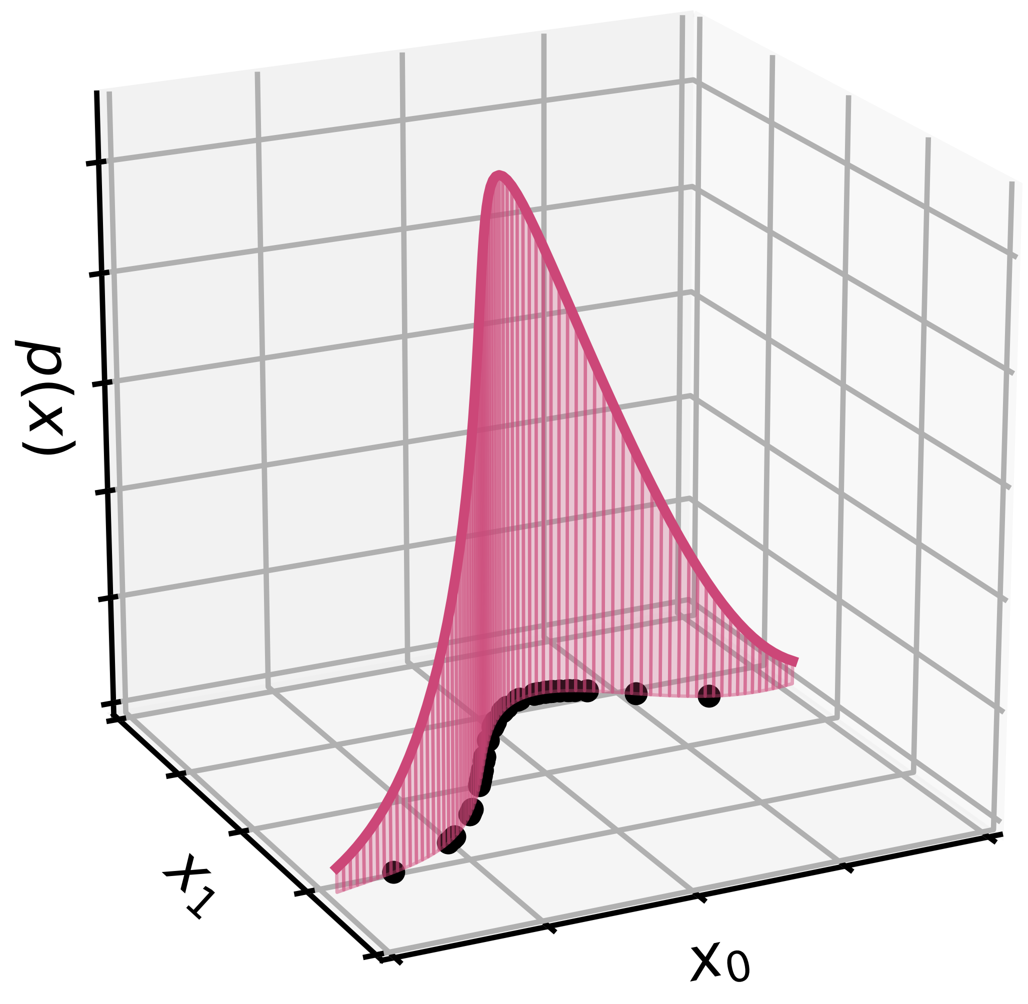
Near-future telescopes and satellites will collect hundreds of lensing images [Collett et al 1507.02657]

Goal: infer DM properties from all images and all clumps at once

ML-based Bayesian inference

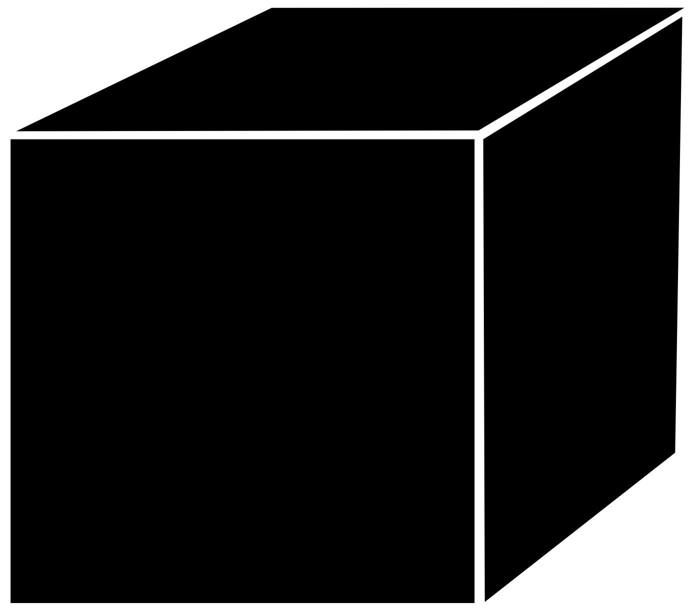
[JB, S. Mishra-Sharma, J. Hermans, G. Louppe, K. Cranmer 1909.02005]





5. Tangents

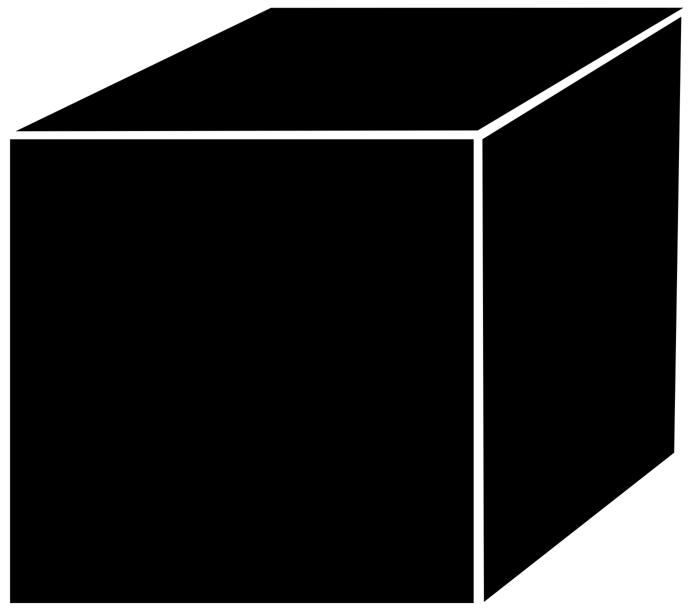
The Achilles heel: model misspecification

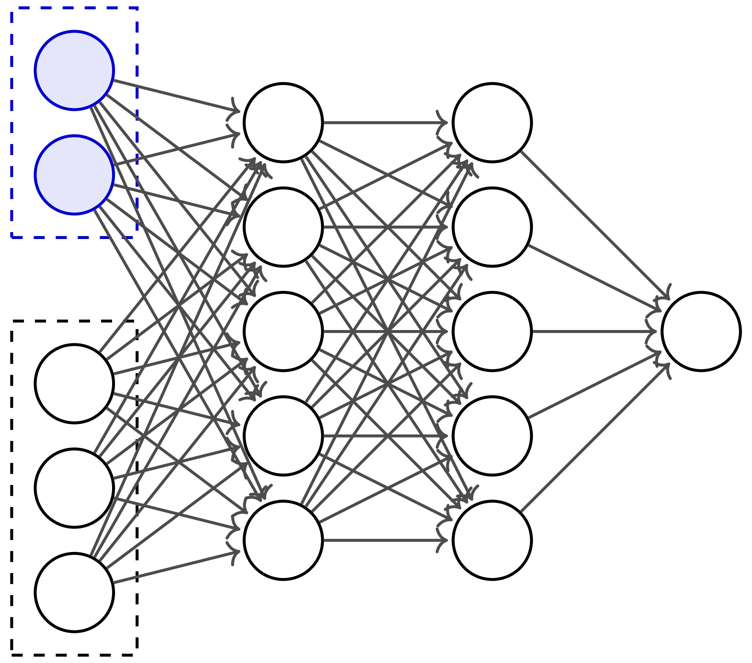


Simulator is wrong?

- Model uncertainties explicitly:
  nuisance parameters + profiling / marginalization
- Make analysis robust:
  ideas from domain adaptation, algorithmic fairness
  [G. Louppe, M. Kagan, K. Cranmer 1611.01046; J. Alsing, B. Wandelt 1903.01473; P. de Castro, T. Dorigo 1806.04743]

The Achilles heel: model misspecification





Simulator is wrong?

- Model uncertainties explicitly:
  nuisance parameters + profiling / marginalization
- Make analysis robust:
  ideas from domain adaptation, algorithmic fairness
  [G. Louppe, M. Kagan, K. Cranmer 1611.01046; J. Alsing, B. Wandelt 1903.01473; P. de Castro, T. Dorigo 1806.04743]

NN didn't converge to minimum?

- Sanity checks: expectation values, "critic" tests
- Calibration / Neyman construction with toys
  (badly trained network can lead to suboptimal limits, but not to wrong limits)
  [JB, G. Louppe, J. Pavez, K. Cranmer 1805.00020]

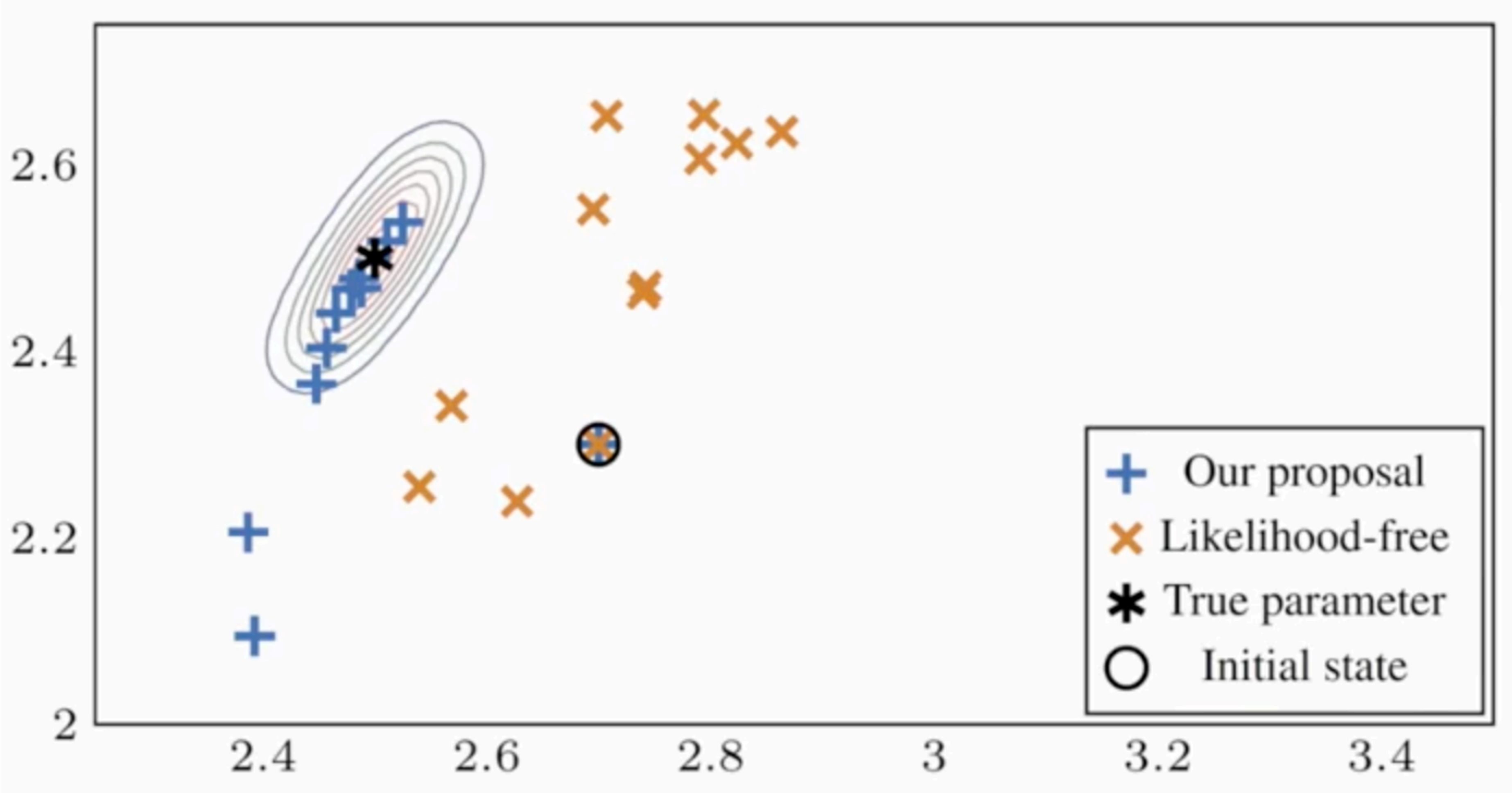
More black boxes to open

Solving ODE inverse problems:

[H. Kersting, N. Krämer, M. Schiegg, C. Daniel, M. Tiemann, P. Hennig 2002.09301]

- Typically treated as black-box simulation-based inference problem
- But the dynamical system is known! With probabilistic numerics, one can often compute the likelihood (gradients) approximately

⇒ More efficient optimization and sampling

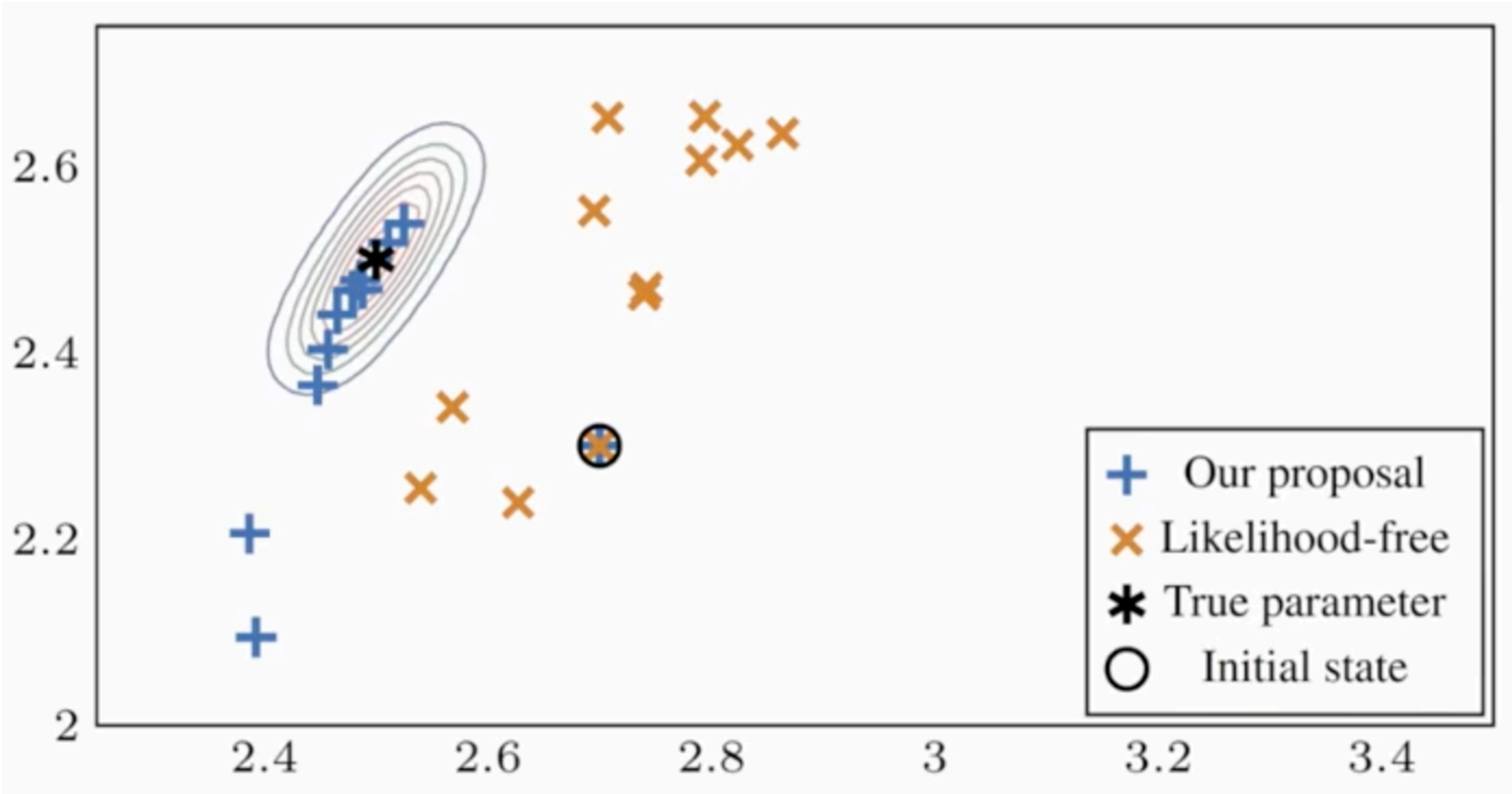


More black boxes to open

Solving ODE inverse problems:
[H. Kersting, N. Krämer, M. Schiegg, C. Daniel, M. Tiemann, P. Hennig 2002.09301]

- Typically treated as black-box simulation-based inference problem
- But the dynamical system is known! With probabilistic numerics, one can often compute the likelihood (gradients) approximately

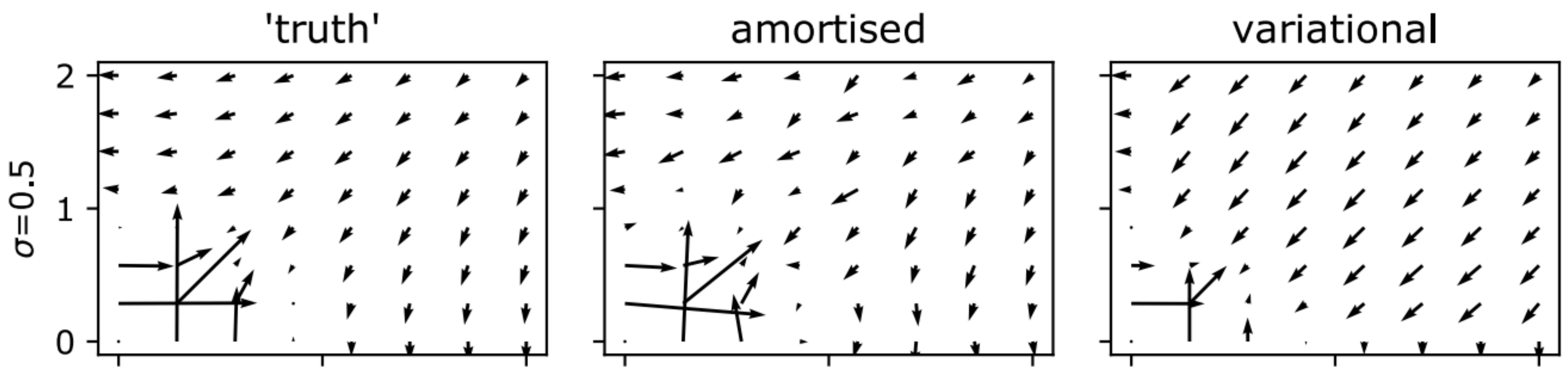
⇒ More efficient optimization and sampling



Training latent-variable models:
[L. Wenliang, T. Moskovitz, H. Kanagawa, M. Sahani 2002.09737]

- Generative latent-variable models (e.g. VAEs) are hard to train because their likelihood is intractable
- Use the "gold-mining" idea to get unbiased estimators of likelihood gradients wrt the NN parameters

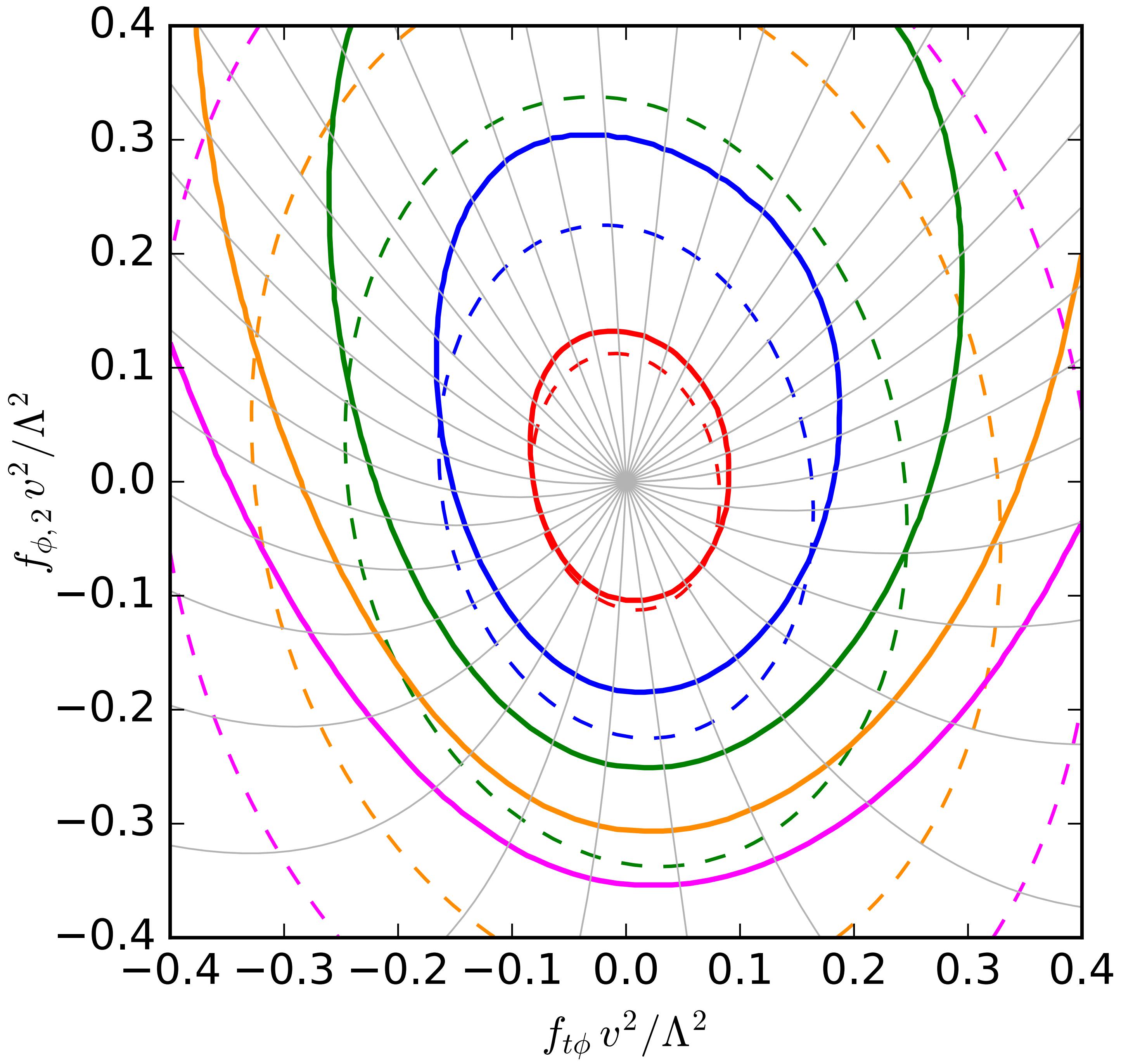
⇒ More efficient training



Information geometry

[JB, K. Cranmer, F. Kling, T. Plehn 1612.05261;
JB, F. Kling, T. Plehn, T. Tait 1712.02350]

Study manifold of probability distributions geometrically

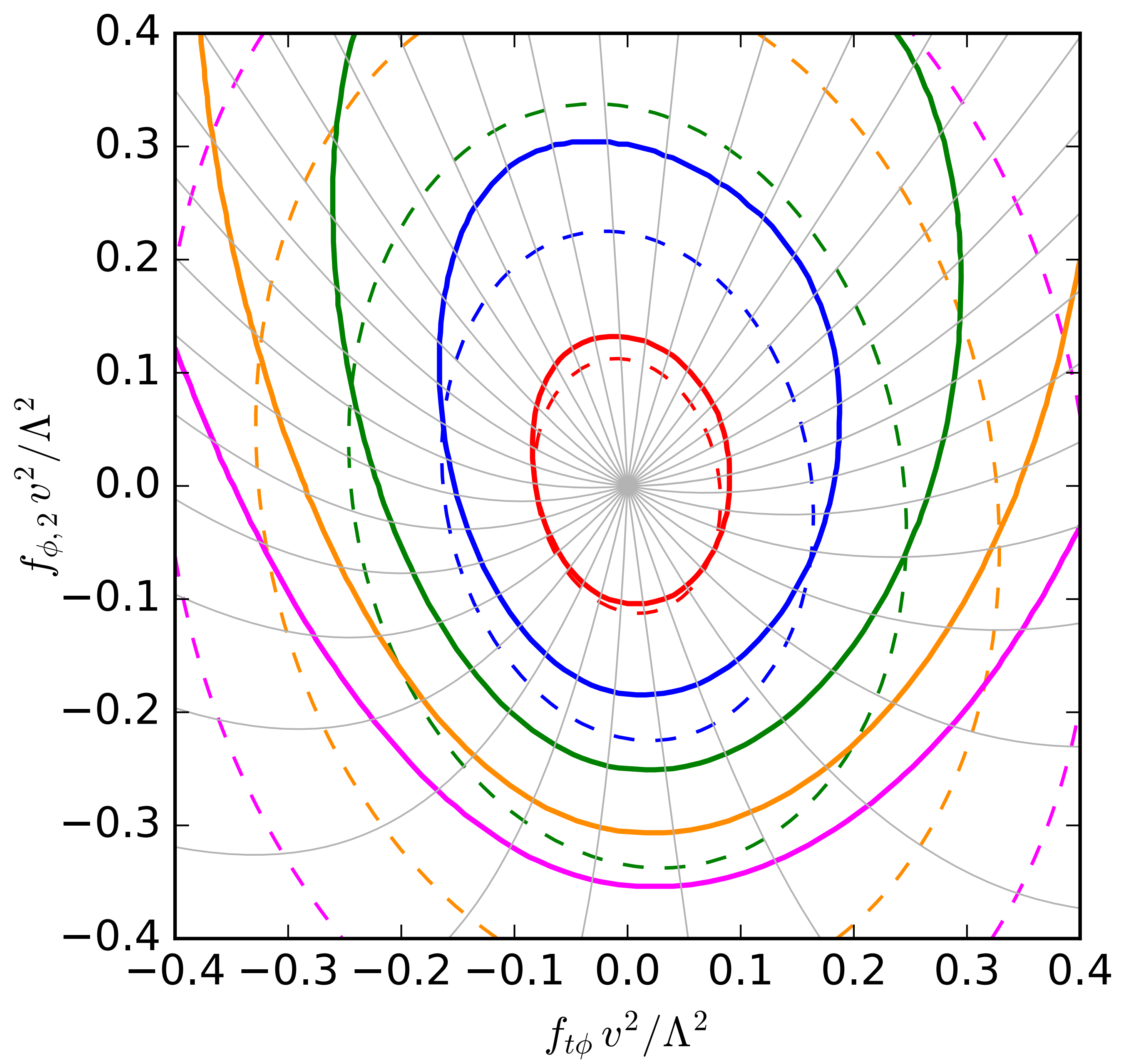


Distances ↔ sensitivity with which an experiment can distinguish parameter points

Information geometry

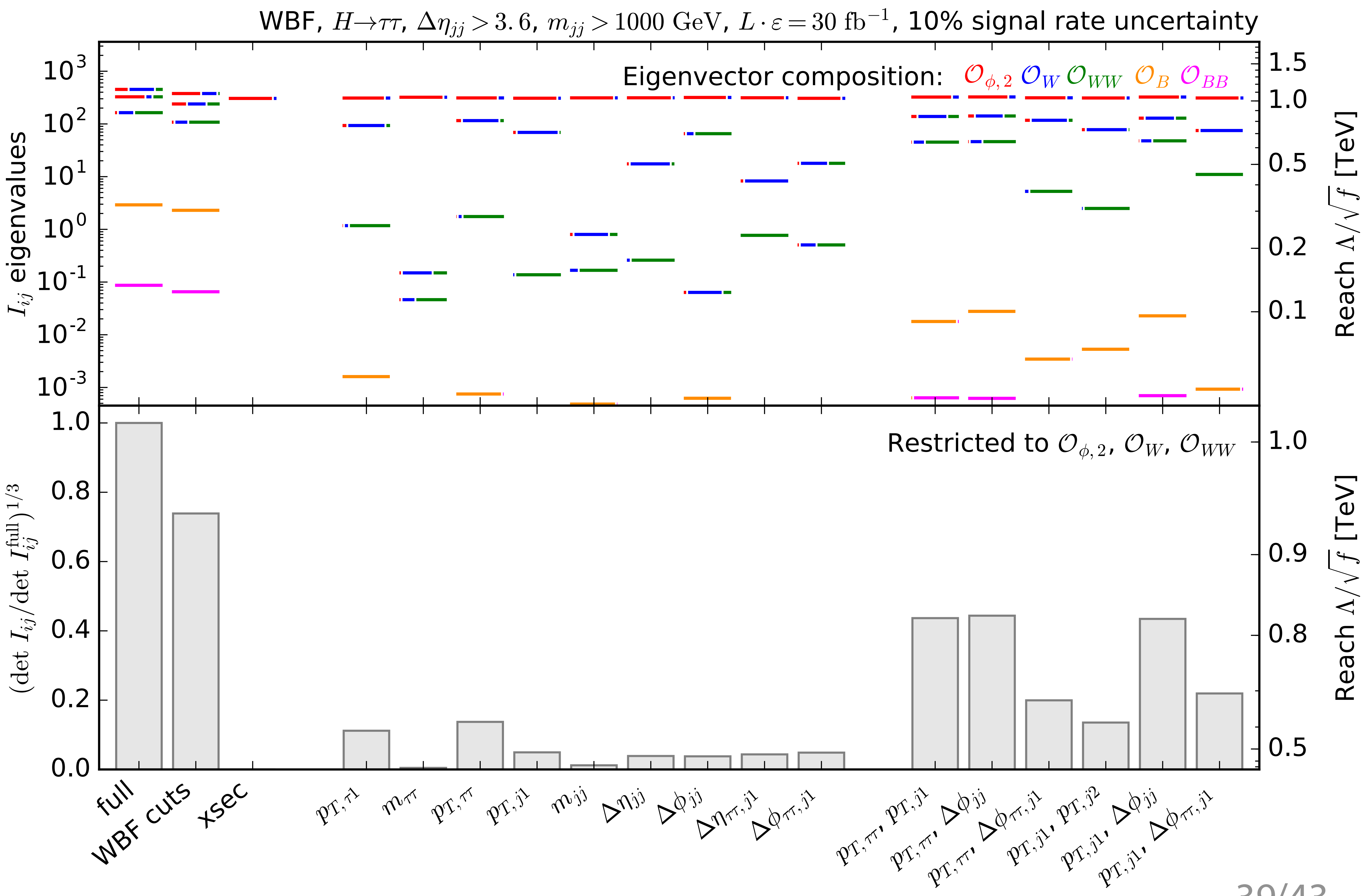
[JB, K. Cranmer, F. Kling, T. Plehn 1612.05261;
JB, F. Kling, T. Plehn, T. Tait 1712.02350]

Study manifold of probability distributions geometrically



Distances ↔ sensitivity with which an experiment can distinguish parameter points

In practice, we can use this for sensitivity forecasting, feature selection, experimental design



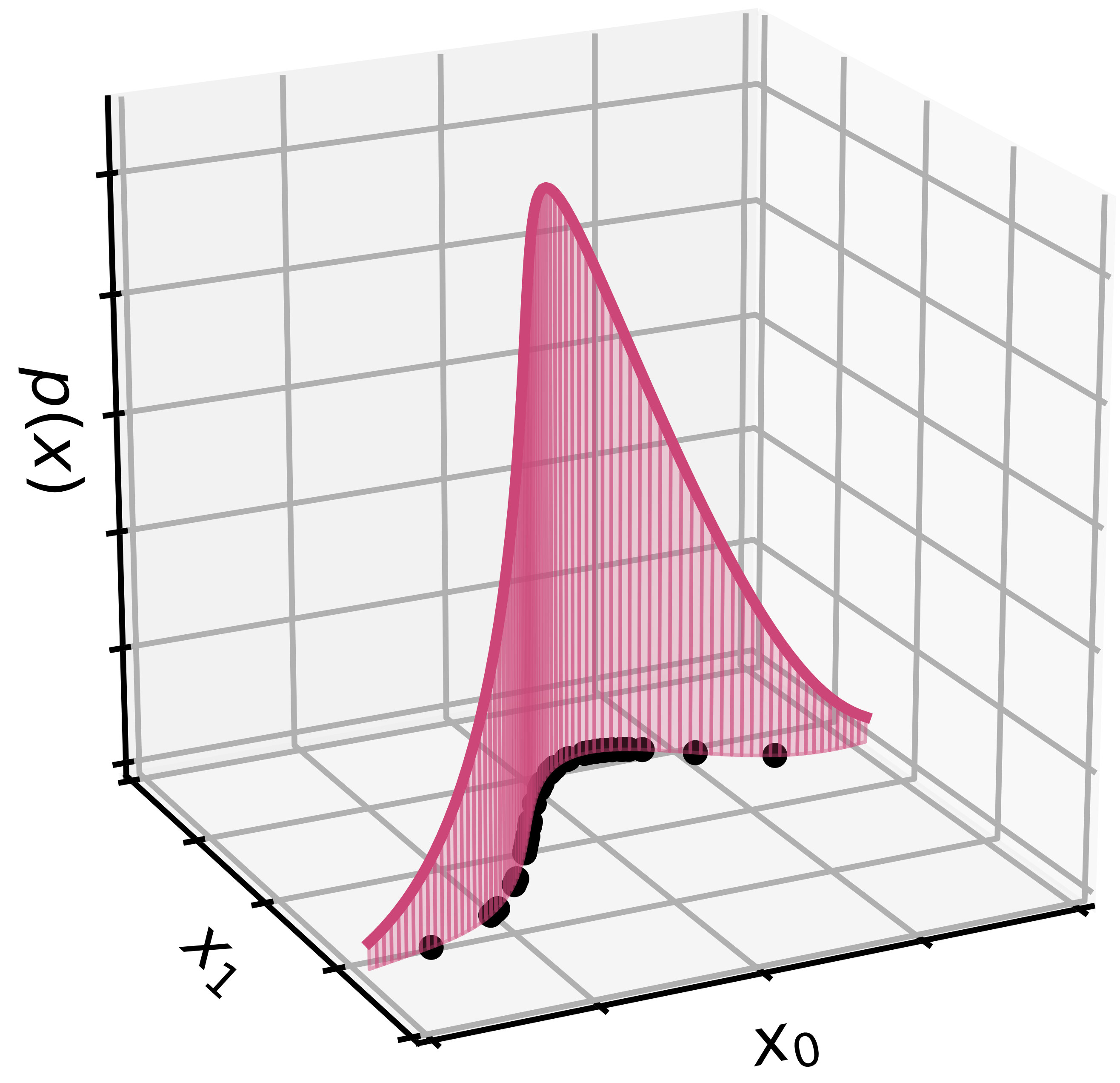
$\mathcal{M}$-flows

[JB, K. Cranmer 2003.13913]

Often data is restricted to a lower-dimensional manifold embedded in the data space

$\mathcal{M}$-flows are a new probabilistic / generative model that

- describe data as a tractable probability density on a lower-dimensional manifold
- learn manifold and density from data



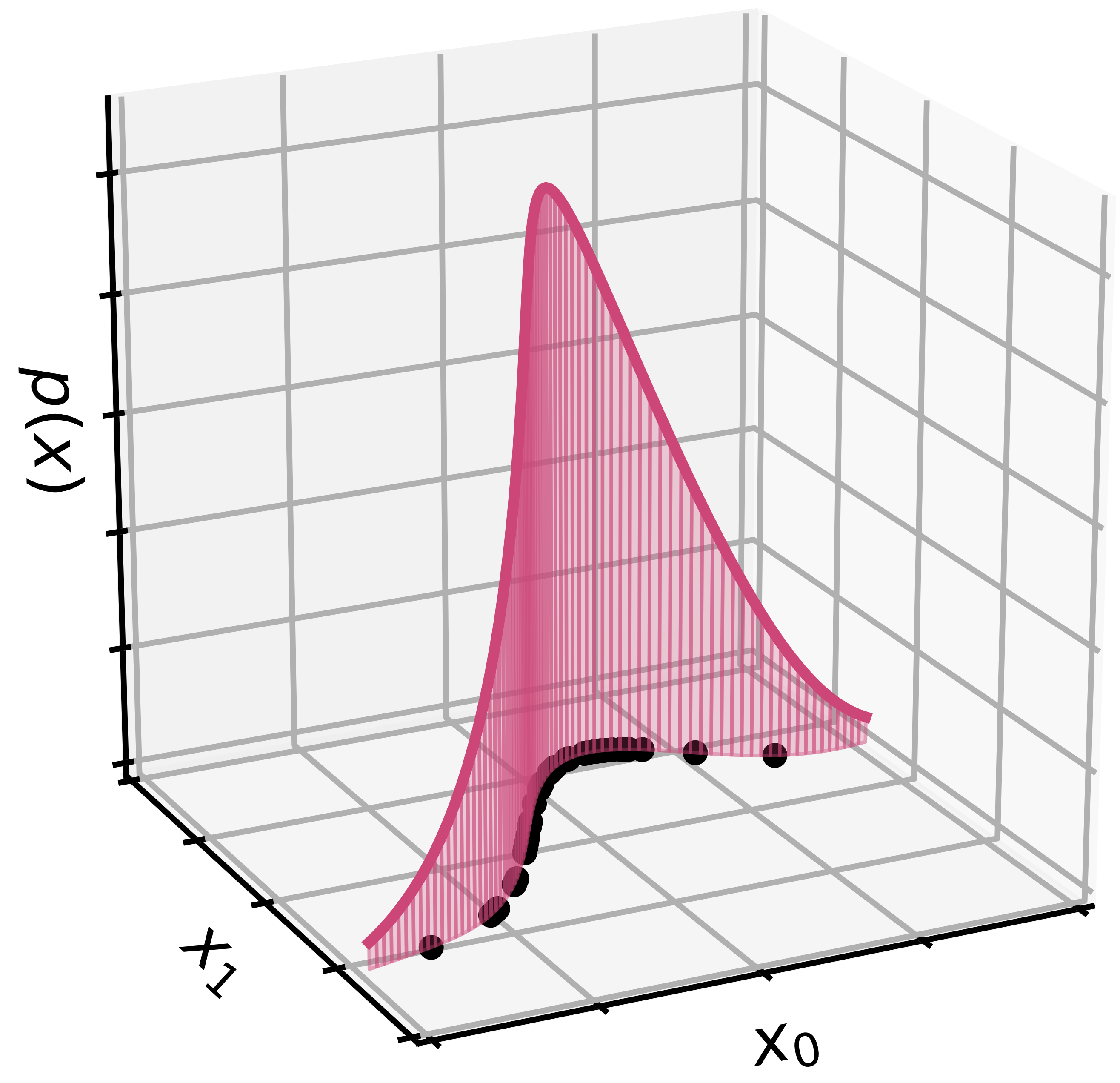
$\mathcal{M}$-flows

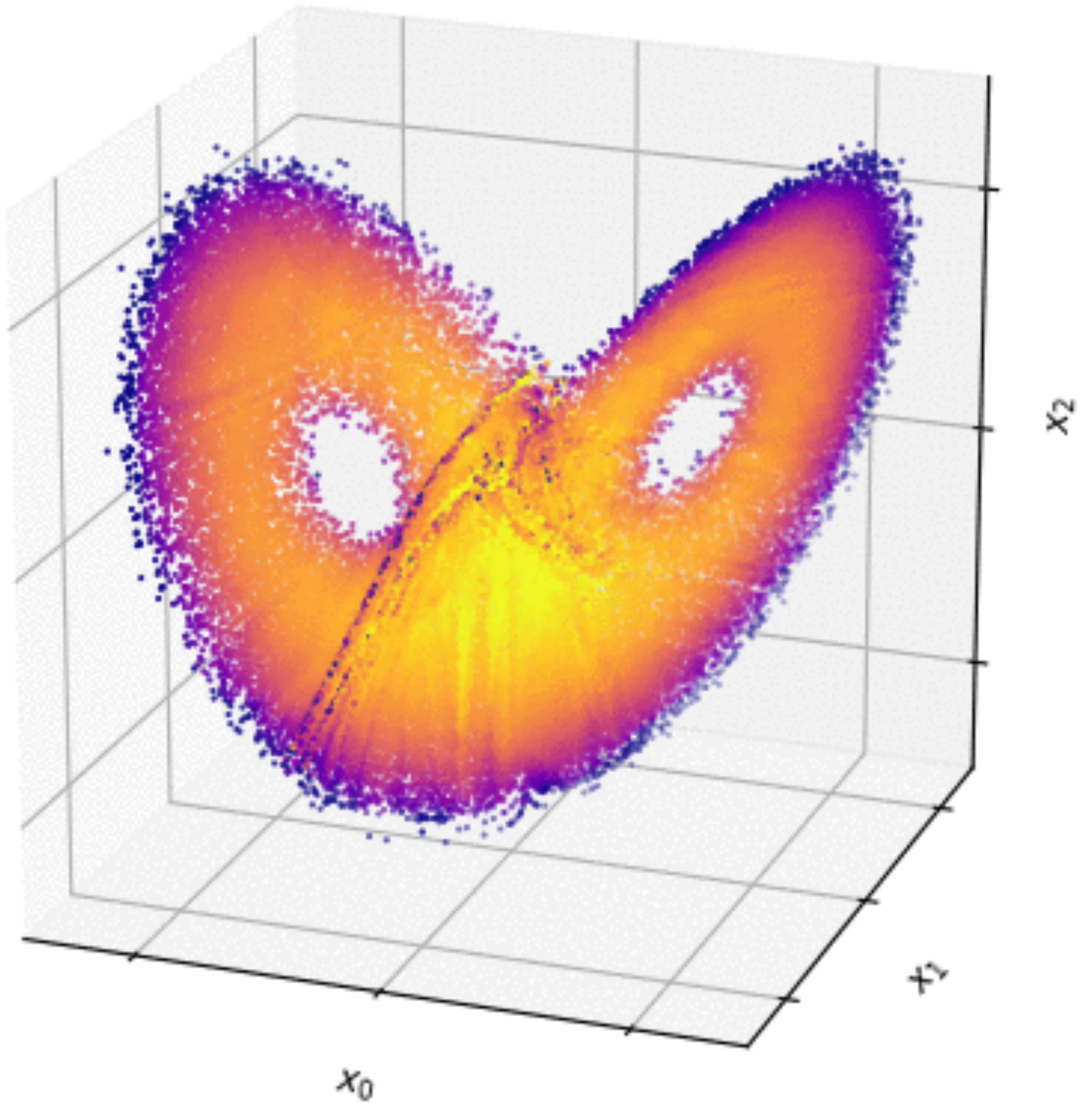
[JB, K. Cranmer 2003.13913]

Often data is restricted to a lower-dimensional manifold embedded in the data space

$\mathcal{M}$-flows are a new probabilistic / generative model that

- describe data as a tractable probability density on a lower-dimensional manifold
- learn manifold and density from data





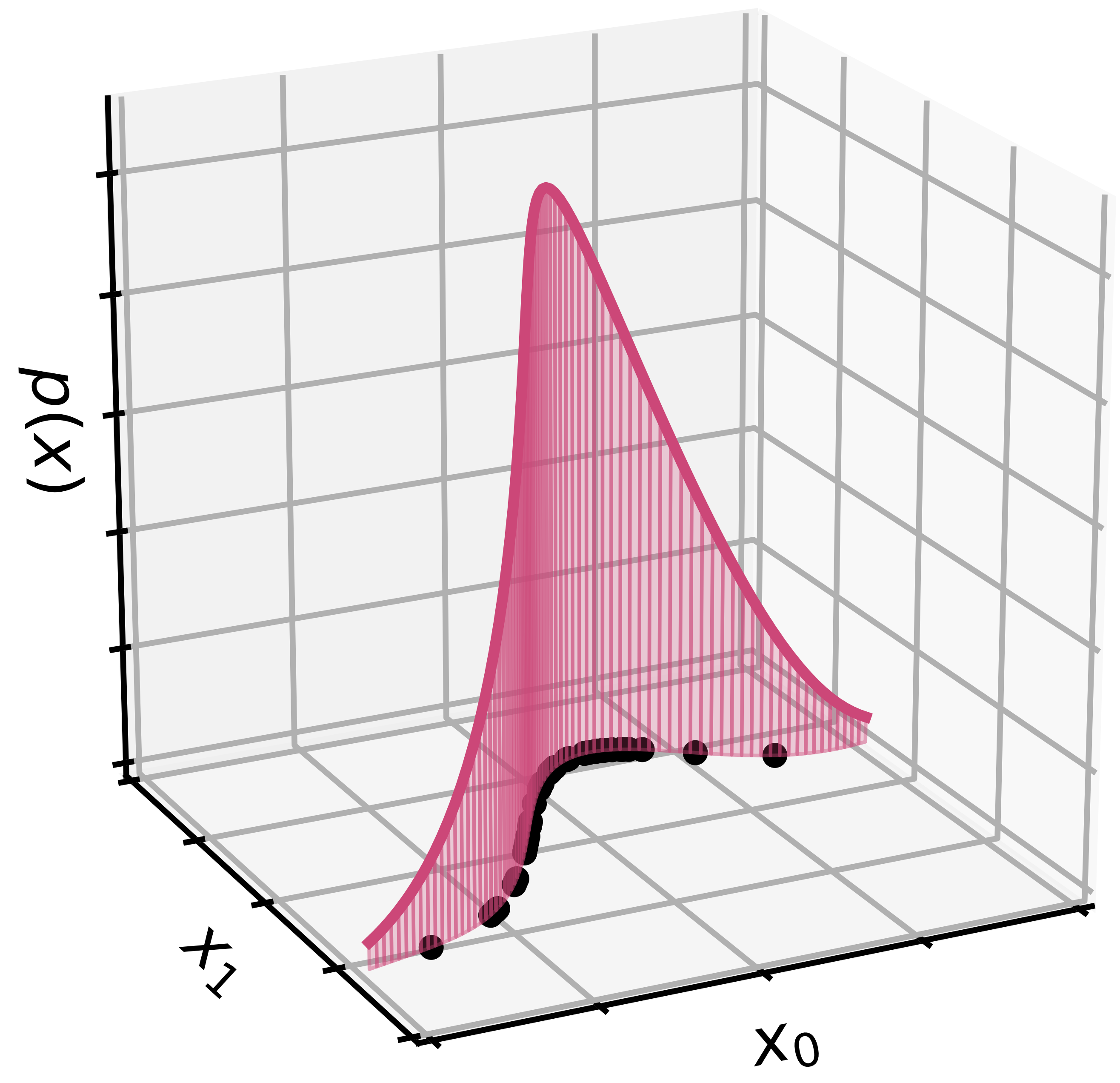
$\mathcal{M}$-flows

[JB, K. Cranmer 2003.13913]

Often data is restricted to a lower-dimensional manifold embedded in the data space

$\mathcal{M}$-flows are a new probabilistic / generative model that

- describe data as a tractable probability density on a lower-dimensional manifold
- learn manifold and density from data




Kyle Cranmer


Gilles Louppe


Juan Pavez

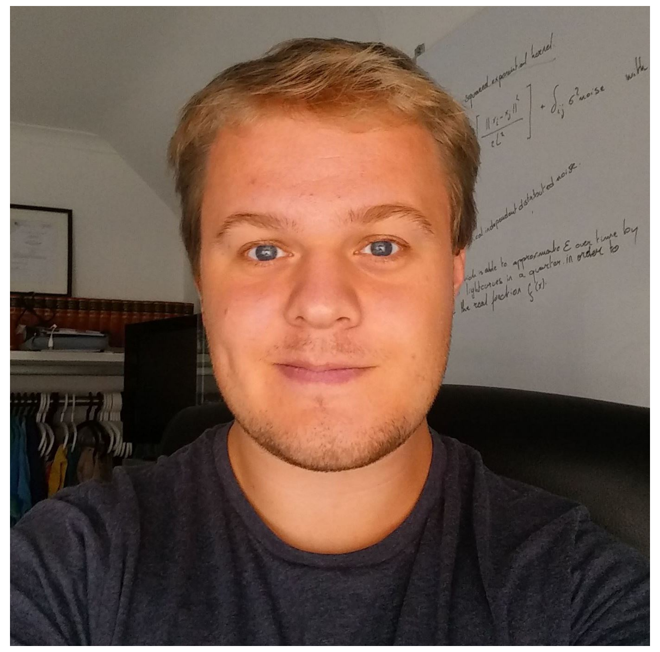

Markus Stoye


Felix Kling

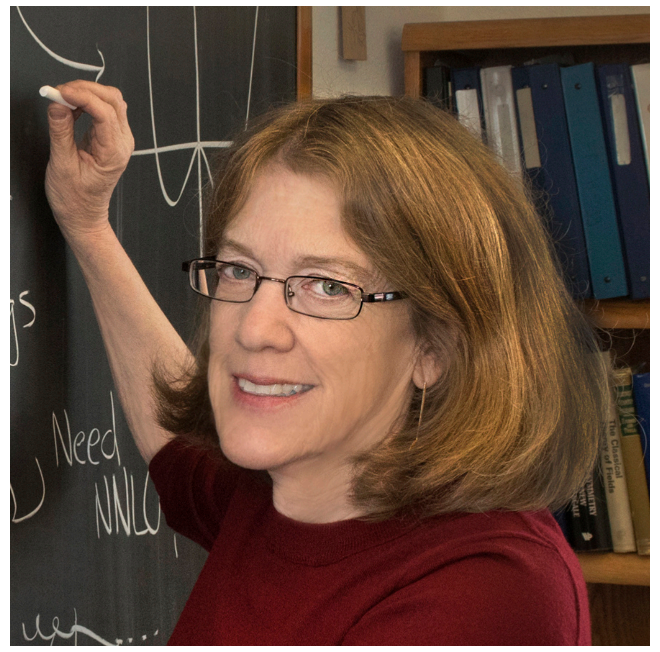

Irina Espejo


Sinclert Perez


Sid Mishra-Sharma


Joeri Hermans


Tilman Plehn


Sally Dawson


Sam Homiller


Zubair Bhatti

Parts of this talk were inspired by great presentations by Kyle Cranmer, Gilles Louppe, Sid Mishra-Sharma, and Jakob Macke

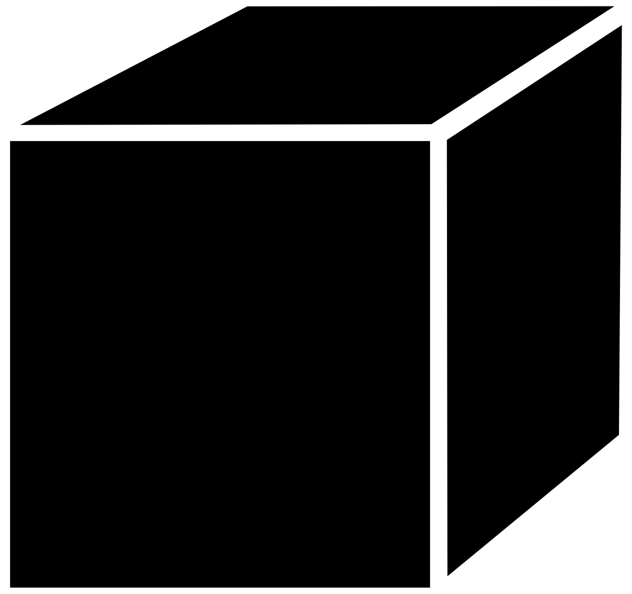
NYU

MSDSE
Moore - Sloan
Data Science Environments

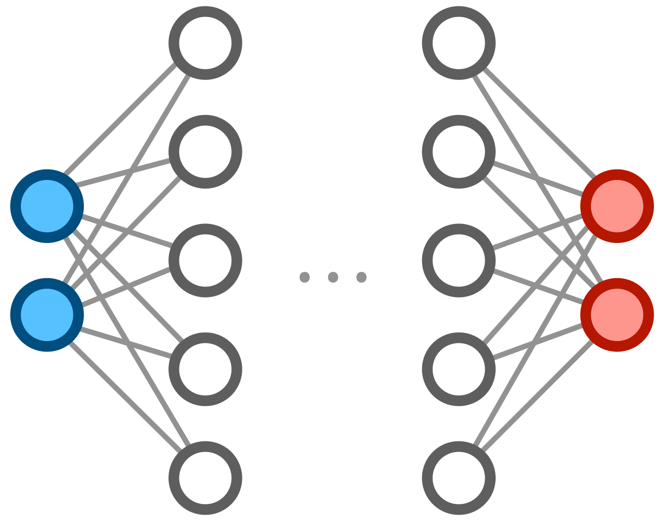
dianahep

iris hep
Institute for Research and Innovation
in Software for High Energy Physics

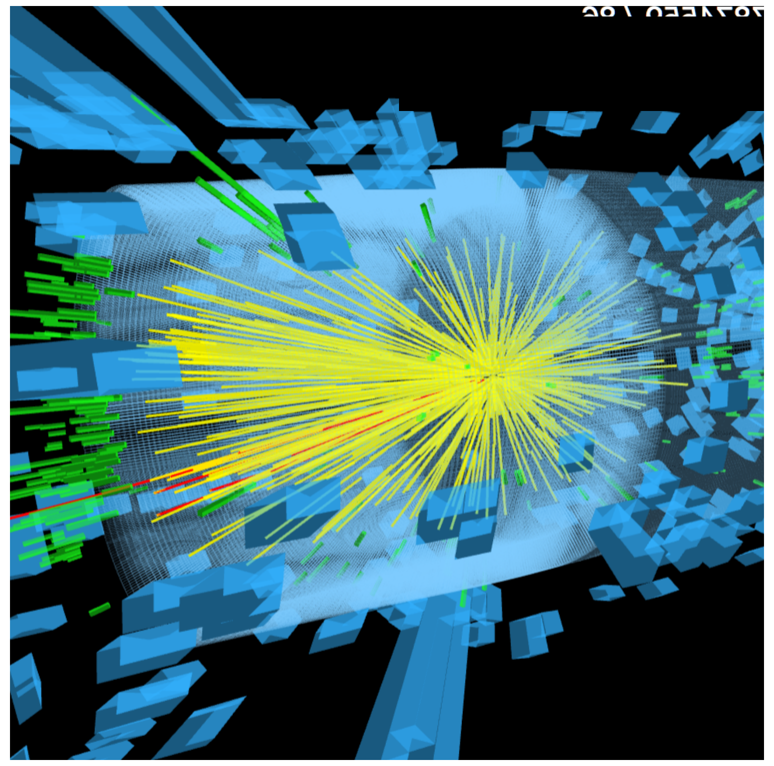
The SCAILFIN Project
scailfin.github.io



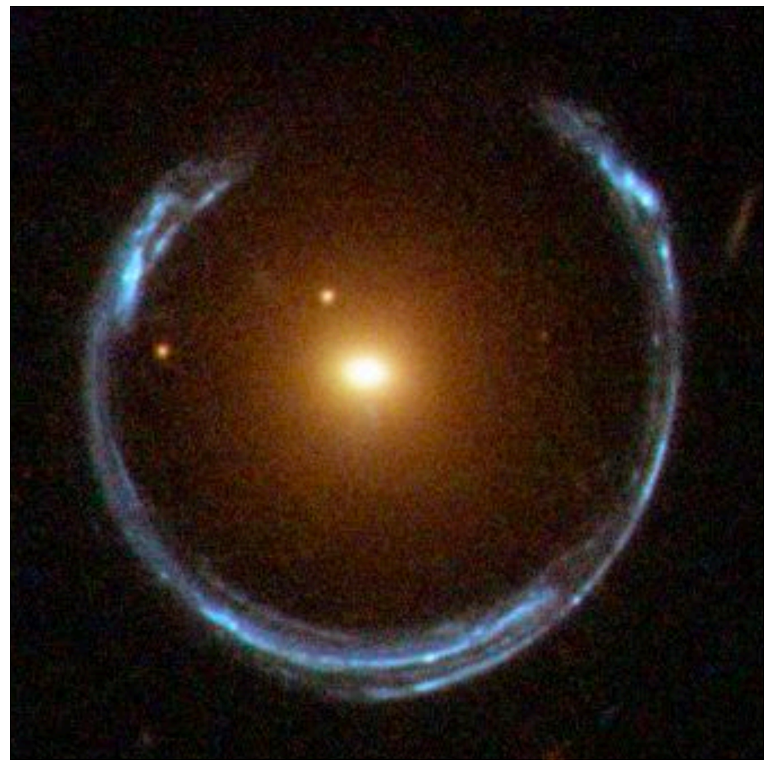
Simulators make precise predictions, but inference is challenging.



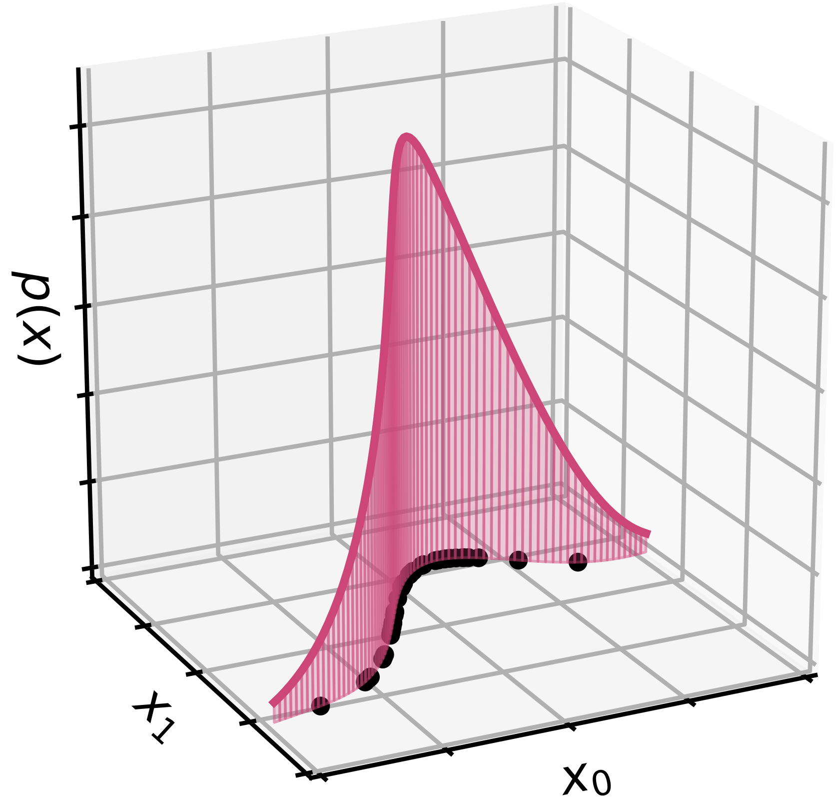
Machine learning enables powerful inference methods, especially when we inject more information.



They work in problems from the smallest…



… to the largest scales.



Further ML advances will translate into scientific progress.

Selected references

Review of simulation-based inference

K. Cranmer, **J. Brehmer**, G. Louppe:
"The frontier of simulation-based inference"
PNAS, 1911.01429

Machine learning–based inference methods

J. Brehmer, G. Louppe, J. Pavez, K. Cranmer:
"Mining gold from implicit models to improve likelihood-free inference"
PNAS, 1805.12244

M. Stoye, **J. Brehmer**, K. Cranmer, G. Louppe, J. Pavez:
"Likelihood-free inference with an improved cross-entropy estimator"
NeurIPS workshop, 1808.00973

Probabilistic models

J. Brehmer and K. Cranmer:
"Flows for simultaneous manifold learning and density estimation"
ICML workshop, 2003.13913

Particle physics

J. Brehmer, K. Cranmer, G. Louppe, J. Pavez:
"Constraining Effective Field Theories with machine learning"
PRL, 1805.00013

J. Brehmer, K. Cranmer, G. Louppe, J. Pavez:
"A guide to constraining Effective Field Theories with machine learning"
PRD, 1805.00020

J. Brehmer, F. Kling, I. Espejo, K. Cranmer:
"MadMiner: Machine learning—based inference for particle physics"
CSBS, 1907.10621, https://github.com/diana-hep/madminer

J. Brehmer, K. Cranmer, F. Kling, and T. Plehn:
"Better Higgs Measurements Through Information Geometry"
PRD, 1612.05261

Astrophysics

J. Brehmer, S. Mishra-Sharma, J. Hermans, G. Louppe, K. Cranmer
"Mining for Dark Matter Substructure: Inferring subhalo population properties from strong lenses with machine learning"
ApJ, 1909.02005