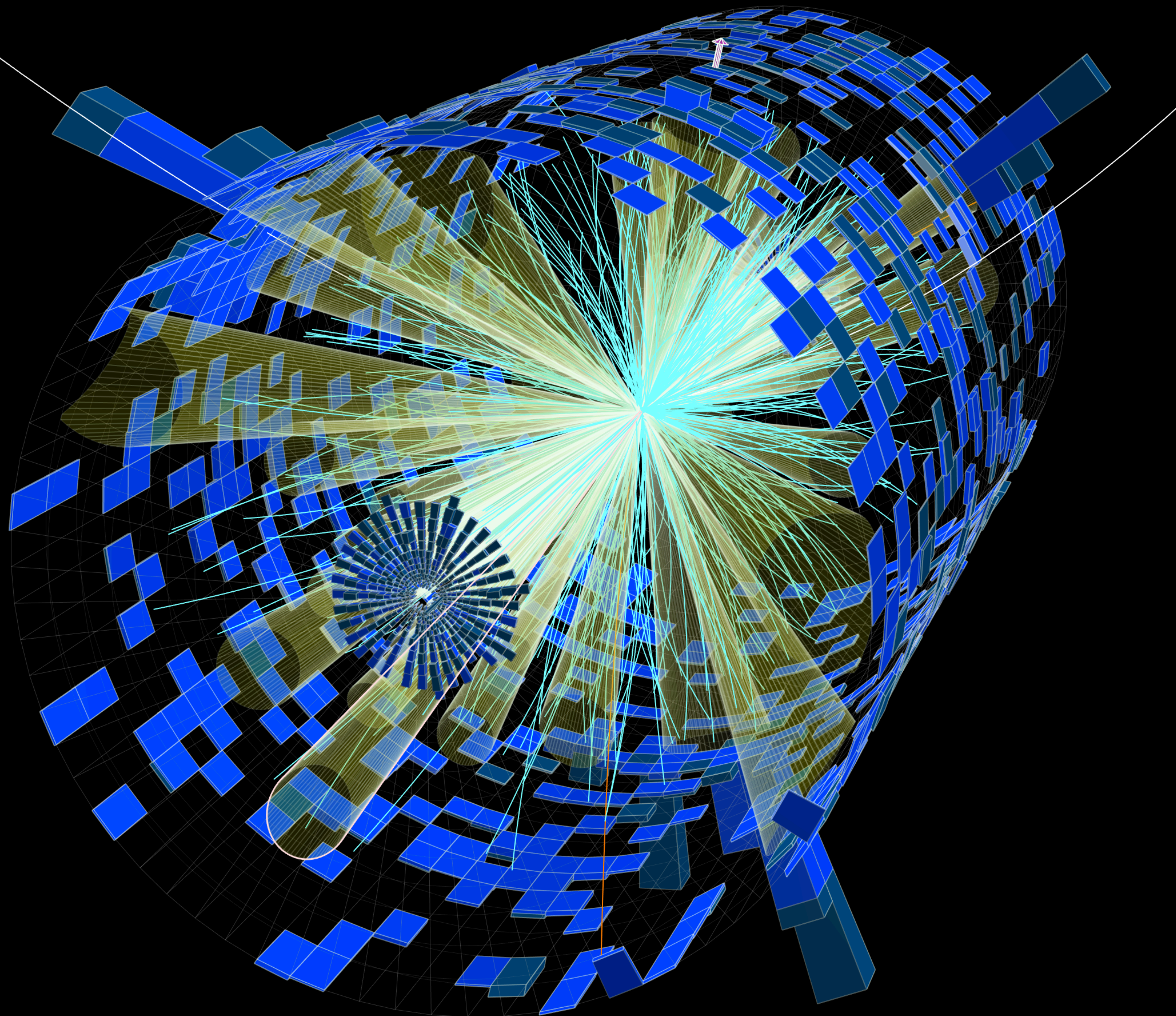
NYU Center for Data Science

Center for Cosmology and Particle Physics

# CONSTRAINING EFTs & DM

## WITH SIMULATION-BASED INFERENCE



**@KyleCranmer**
New York University
Department of Physics
Center for Data Science
CILVR Lab

# Acknowledgements



Johann Brehmer


Gilles Louppe


Juan Pavez

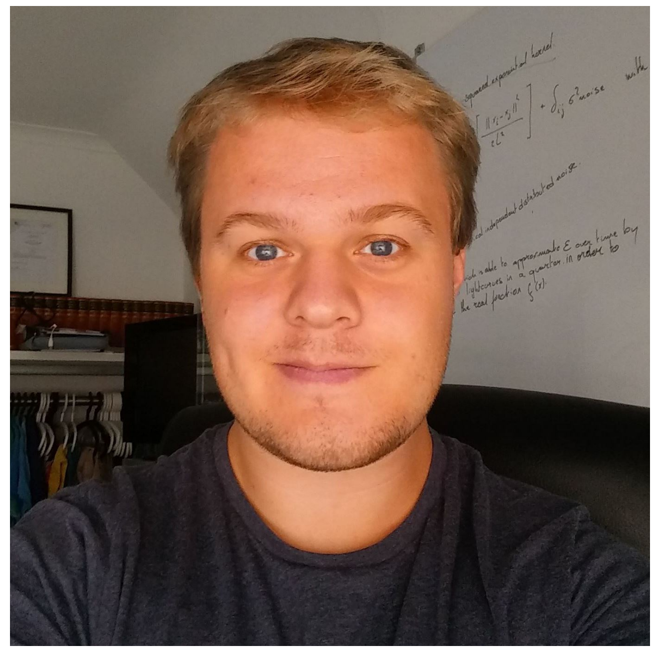

Markus Stoye


Felix Kling

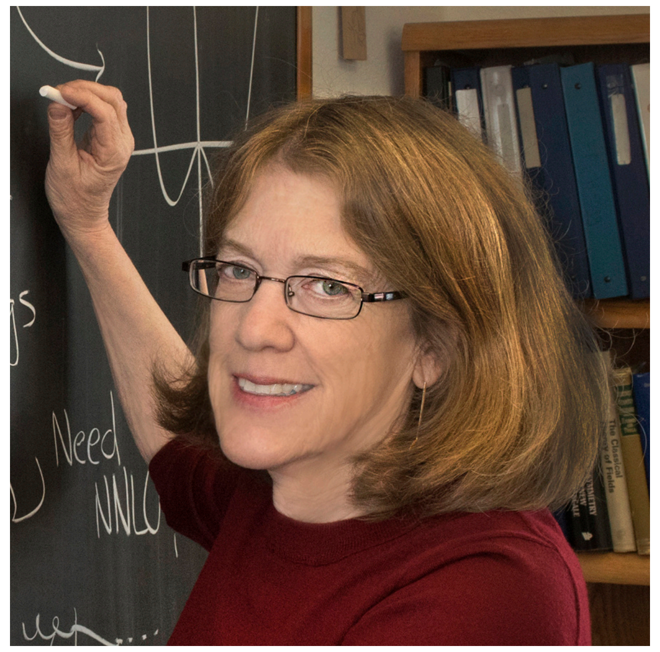

Irina Espejo


Sinclert Perez


Sid Mishra-Sharma


Joeri Hermans


Tilman Plehn


Sally Dawson


Sam Homiller


Zubair Bhatti

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

“I don’t want your pitty, I want change”

- Letetra Widman (Jacob Blake’s sister)

Many of these slides are borrowed from this talk by Johann Brehmer

(who in turn has adapted many of my slides… so it's really hybrid 😃 )

# Constraining effective field theories with machine learning


Johann Brehmer

New York University


ACAT 2019, Saas-Fee

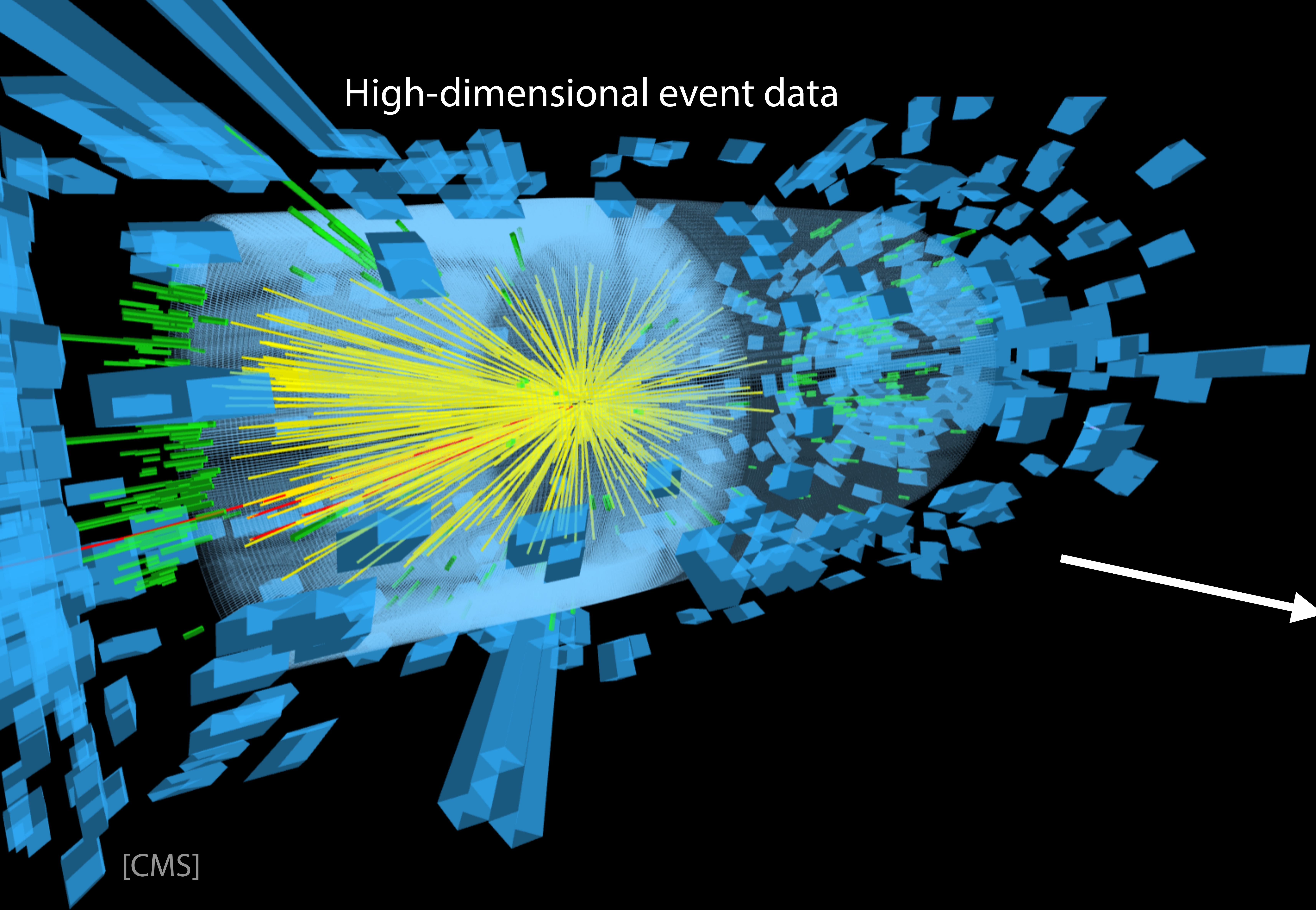
# Motivation

Obviously we would like to directly produce new particles at the LHC, and a robust search program is underway

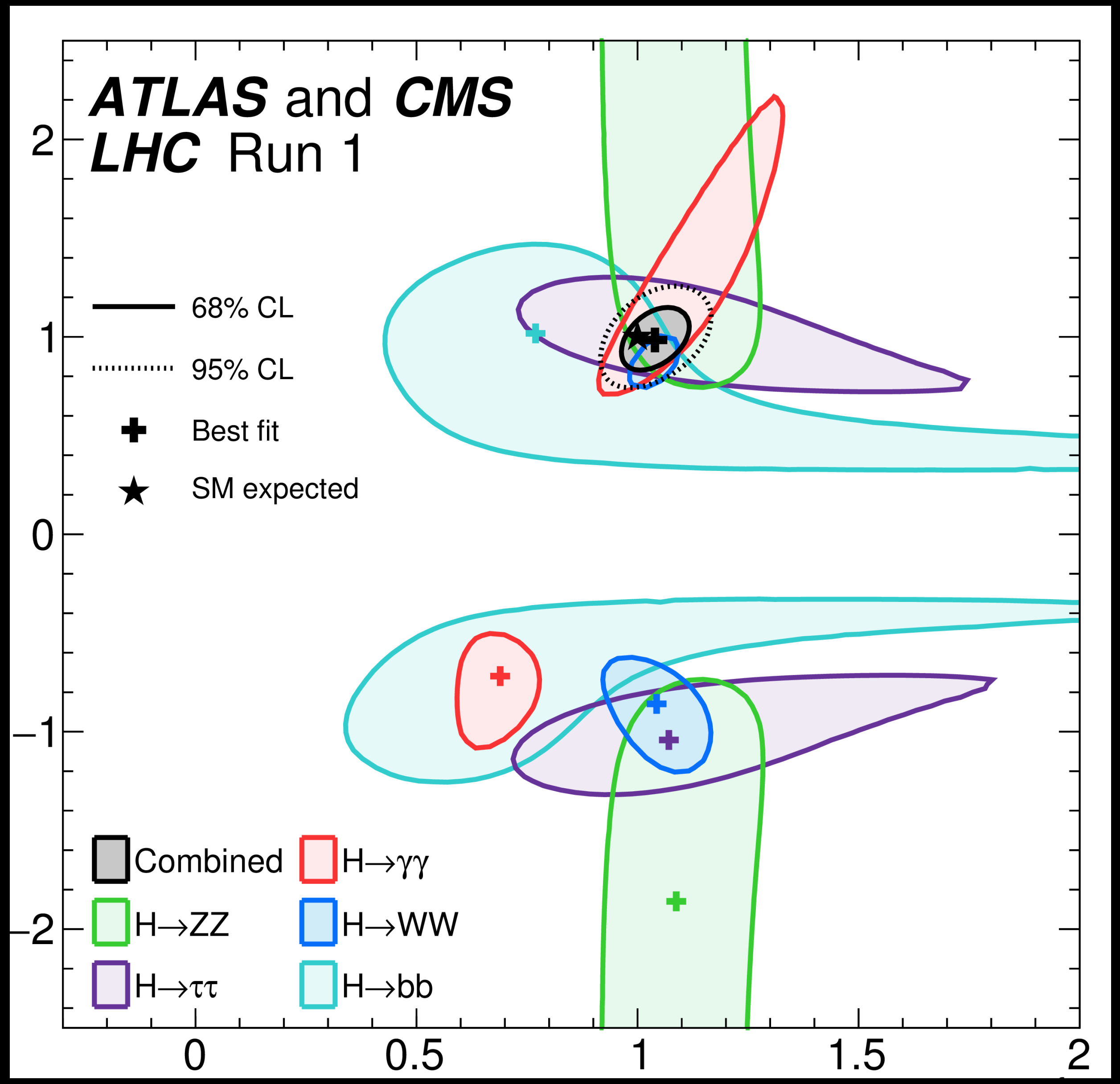
But we can also study new physics indirectly by probing the subtle effects it would have on Standard Model processes.

- These may well be the legacy measurements of the LHC
- Effective Field Theory is a systematic way of parametrizing those effects
- Effect size may be small, so we want to get the most out of the LHC data

High-dimensional event data



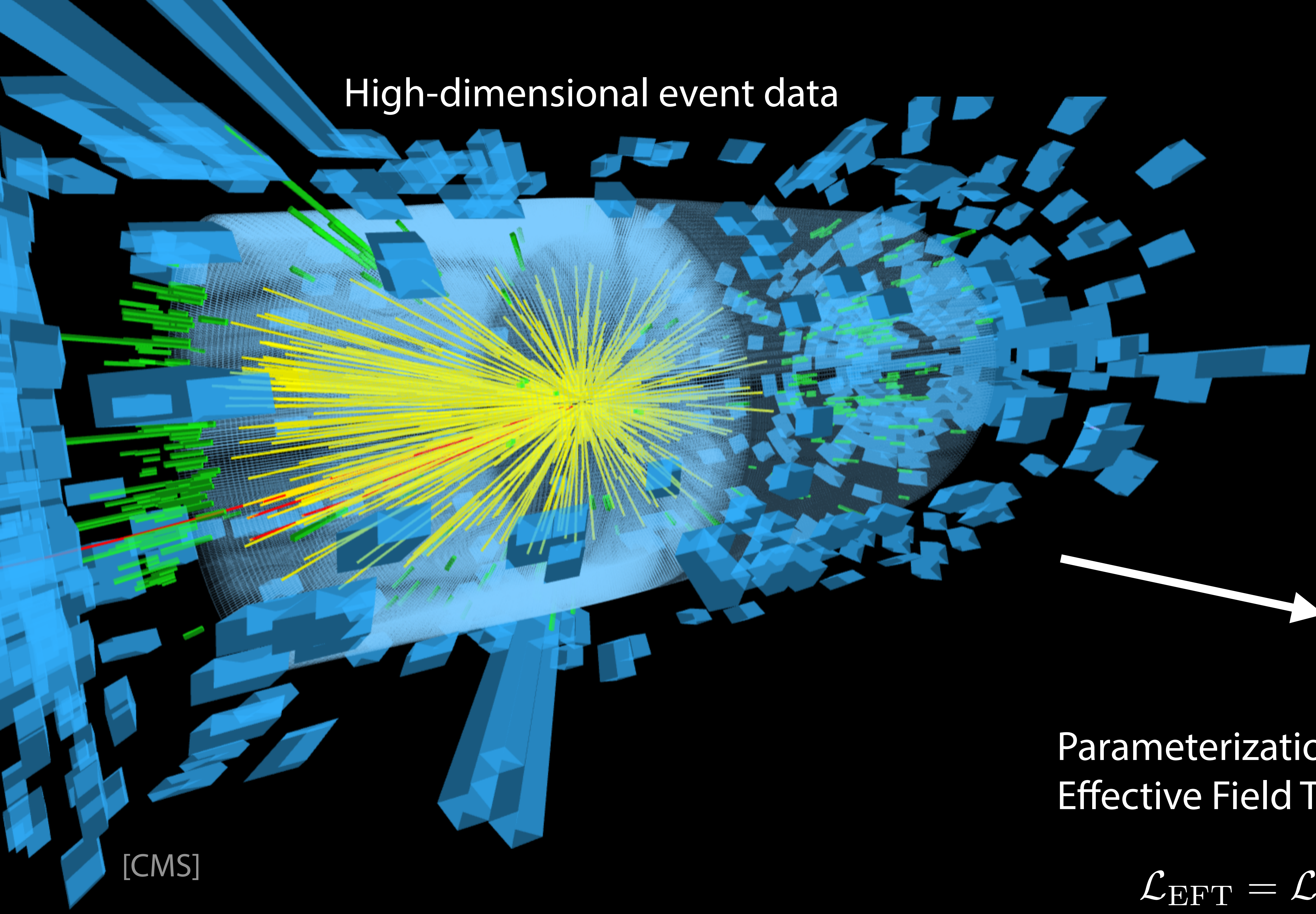
[CMS]



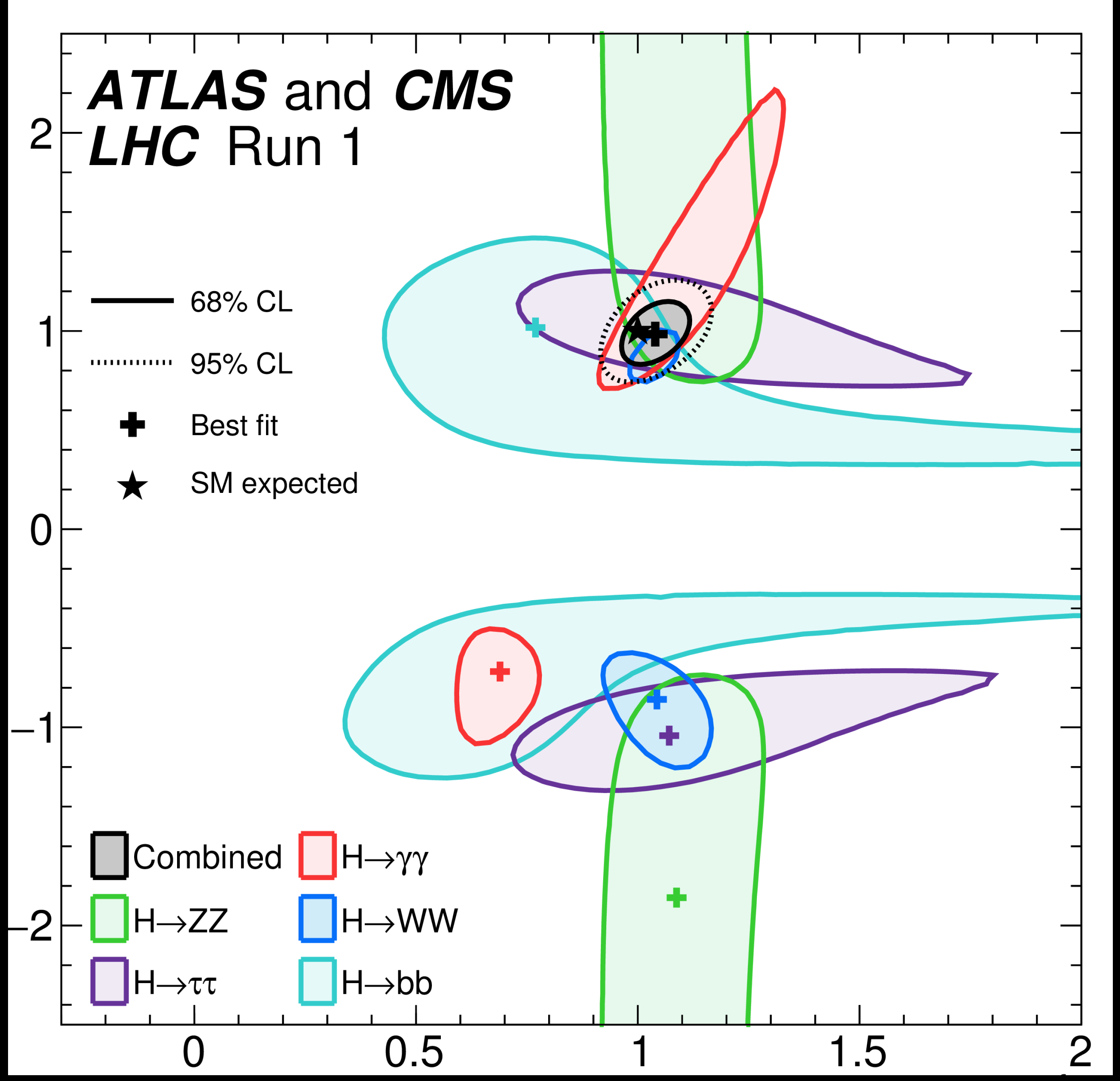

[ATLAS, CMS 1606.02266]

Precision constraints on new physics

High-dimensional event data



[CMS]




[ATLAS, CMS 1606.02266]

Precision constraints on new physics

Parameterization e.g. in Effective Field Theory:

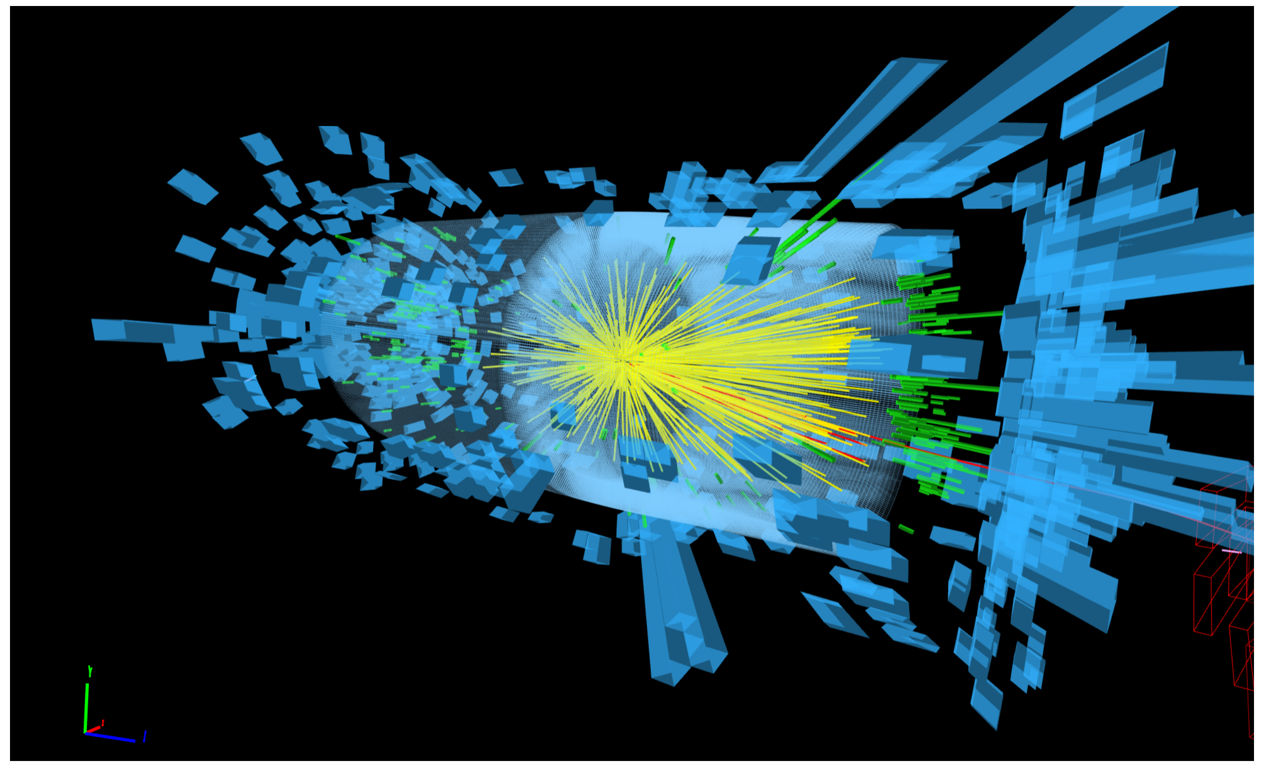
systematic expansion of new physics around Standard Model

$$\mathcal{L}_{\mathrm{EFT}} = \mathcal{L}_{\mathrm{SM}} + \sum_i \frac{f_i}{\Lambda^2} \mathcal{O}_i + \ldots$$

10s to 100s "universal" parameters to measure

# The likelihood is a key object

Let $\theta$ denote the coefficients of higher dimensional operators in the Lagrangian, $x$ be high-dimensional data associated to an event, and $p(x \mid \theta) = \frac{1}{\sigma(\theta)} \frac{d\sigma}{d\theta}$ be the distribution for the data

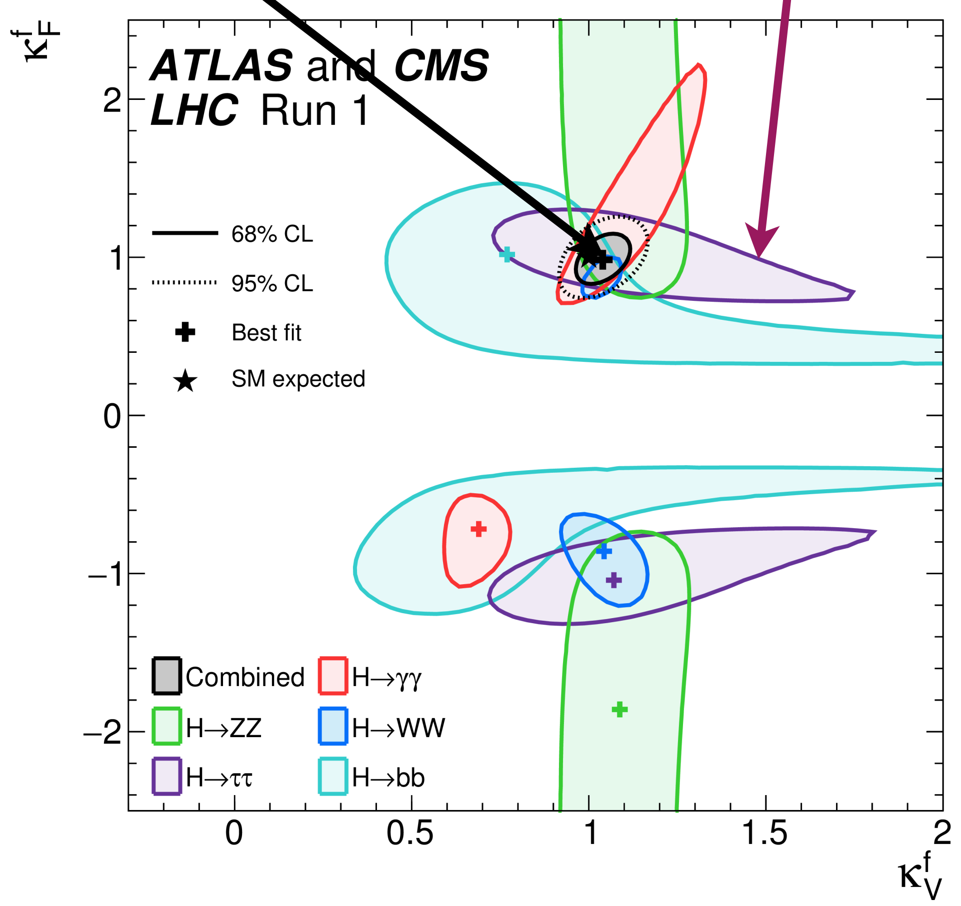


High-dimensional event data $x$

Likelihood function $p(x|\theta)$

Maximum-likelihood estimator

Confidence limits based on likelihood ratio tests




Constraints on parameters $\theta$

# Recap on Likelihood Ratios

$$\frac{P(x|H_1)}{P(x|H_0)} > k_\alpha$$

For signal vs. background searches:

- **Neyman-Pearson Lemma**: optimal hypothesis test given by **likelihood ratio** (basis of Higgs search)
- Likelihood ratio $\frac{p(x|\theta_0)}{p(x|\theta_1)}$ also used for exclusion contours

For estimates of parameters $\hat{\theta}$

- **Cramér-Rao bound** states $\mathrm{cov}[\hat{\theta}|\theta_0]_{ij} \geq I^{-1}_{ij}(\theta_0)$ where $I_{\mathrm{ij}}$ is the **Fisher-information matrix** (Hessian of log-likelihood)
- Motivates **Information Geometry** as a phenomenological tool
- Maximum-likelihood (asymptotically) saturates the bound

Note: $\nabla_\theta \log p(x|\theta)$ acts like a likelihood ratio locally

# Cramér-Rao Bound

The minimum variance bound on an unbiased estimator is given by the Cramér-Rao bound:

$$\mathrm{cov}[\hat{\theta}|\theta_0]_{ij} \geq I^{-1}_{ij}(\theta_0)$$

Expected error of best-fit parameter

Inverse of Fisher information

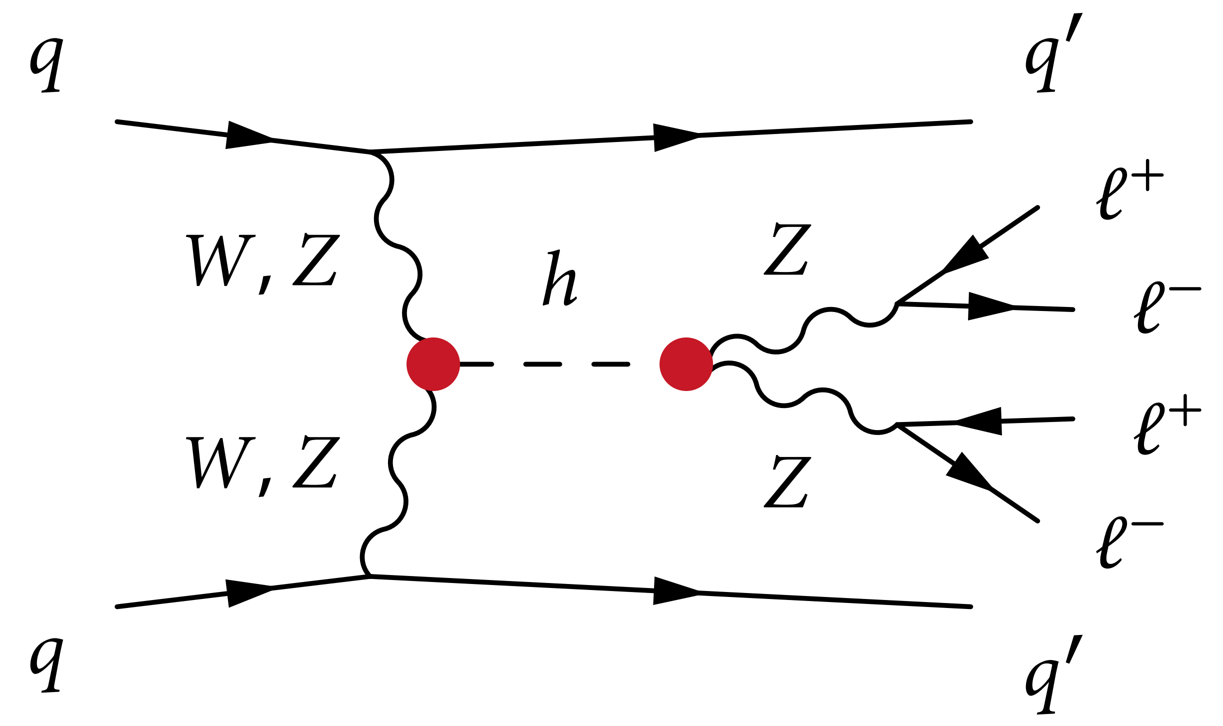
Fisher information matrix (is also a Riemannian metric! )

$$I_{ij}[\theta] = -\,\mathbb{E}\left[\frac{\partial^2 \log p(x|\theta)}{\partial\theta_i\,\partial\theta_j}\middle|\theta\right]$$

Maximum Likelihood Estimators *asymptotically* reach this bound

# Challenge for EFT

Let $\theta$ denote the coefficients of higher dimensional operators in the Lagrangian, $x$ be high-dimensional data associated to an event, and $p(x \mid \theta) = \frac{1}{\sigma(\theta)} \frac{d\sigma}{d\theta}$ be the distribution for the data



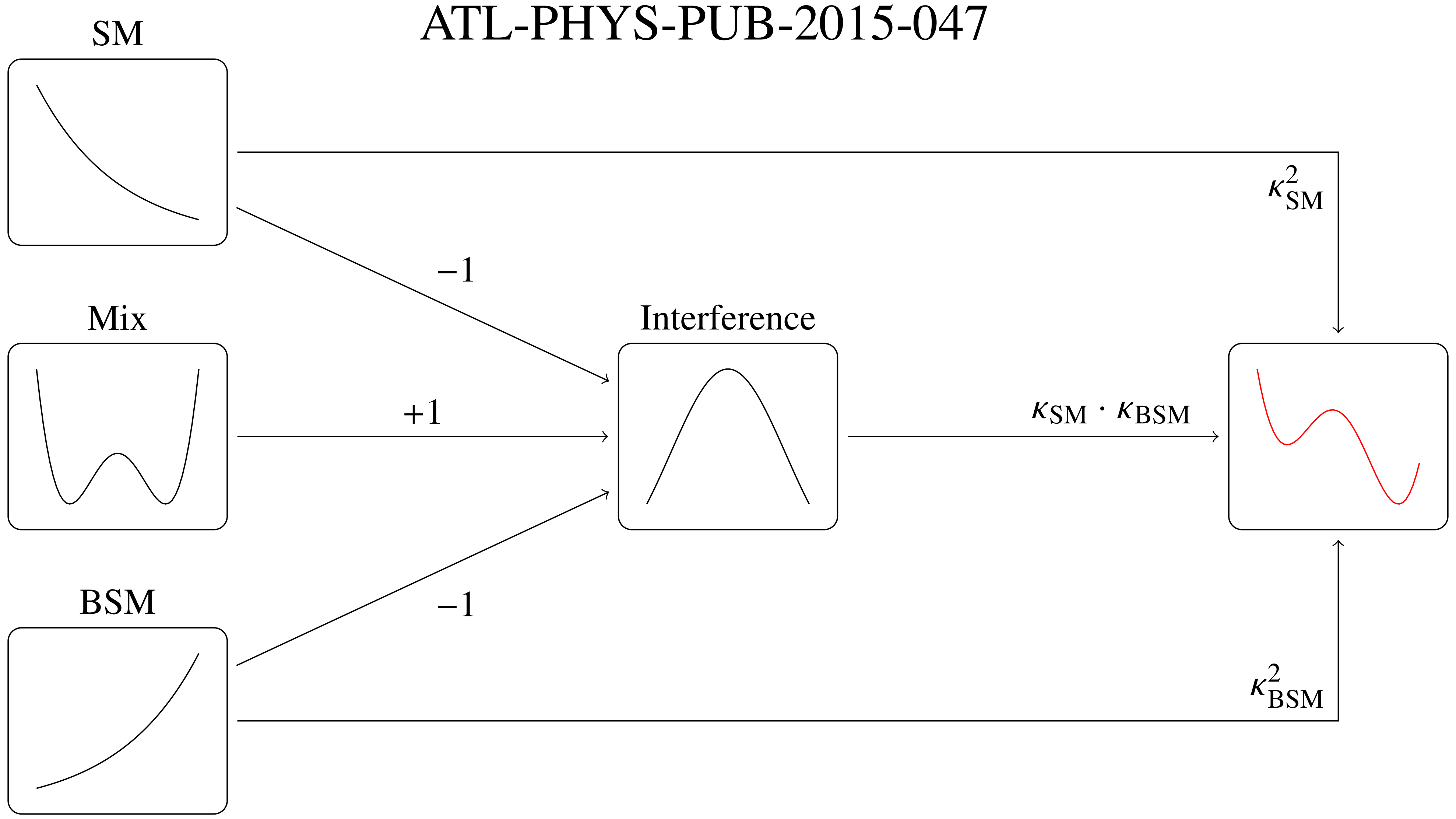

- we want to compare any two points in EFT parameter space
- evaluate the **likelihood ratio** $r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$

Difficulty is that one changes the parameters of the EFT, the distributions p(x|θ) change due to interference.

- It would be very computationally expensive (infeasible) to generate samples for every value of θ and estimate $p(x \mid \theta)$ with histograms. Small changes mean we need a lot of MC events!
- Ideally we could directly estimate the **score** $t(x|\theta_0) \equiv \nabla_\theta \log p(x|\theta)\Big|_{\theta_0}$

# EFT Embedded in a vector space

ATL-PHYS-PUB-2015-047




Difficulty is that one changes the parameters of the EFT, the distributions p(x|θ) change due to interference.

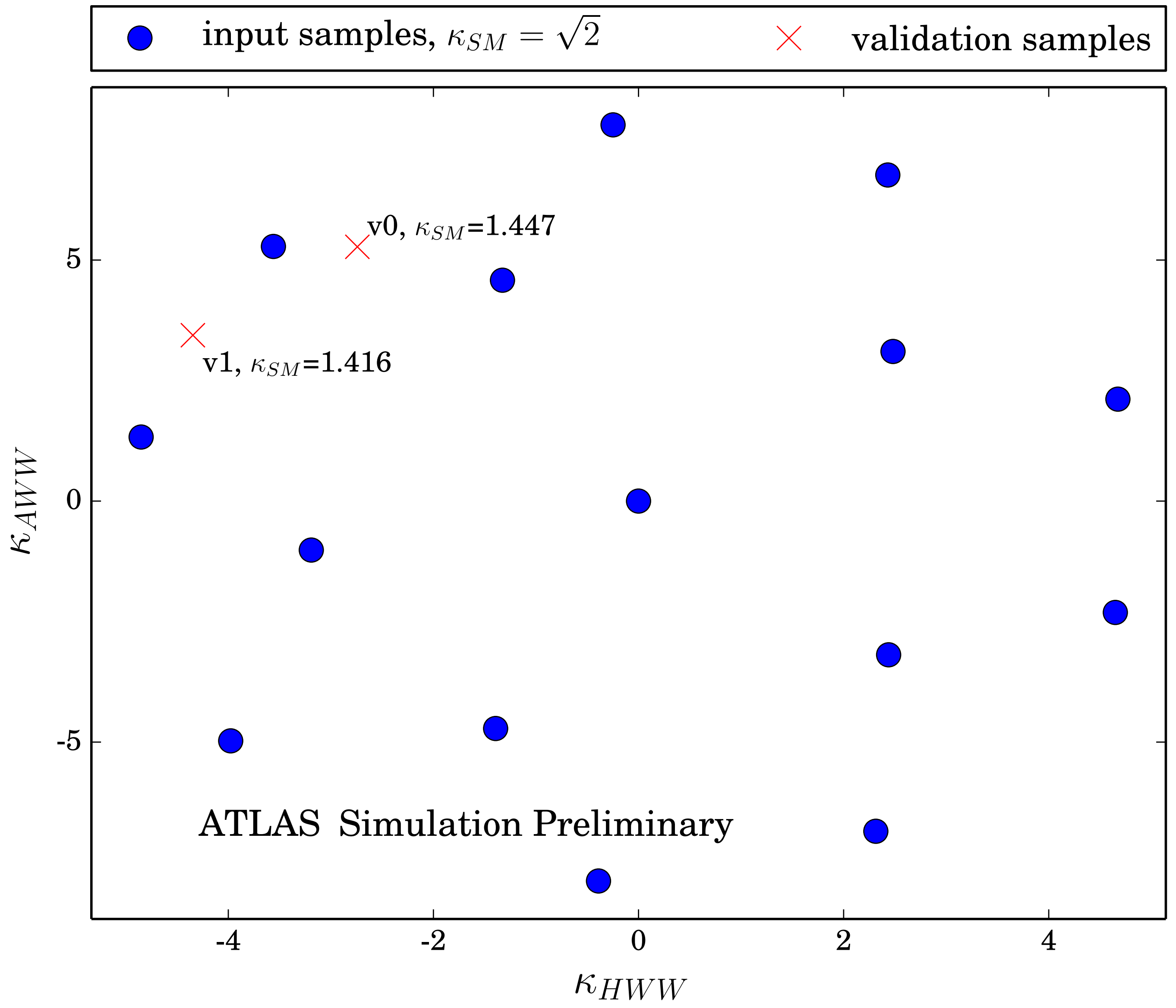
**But there is a trick:**

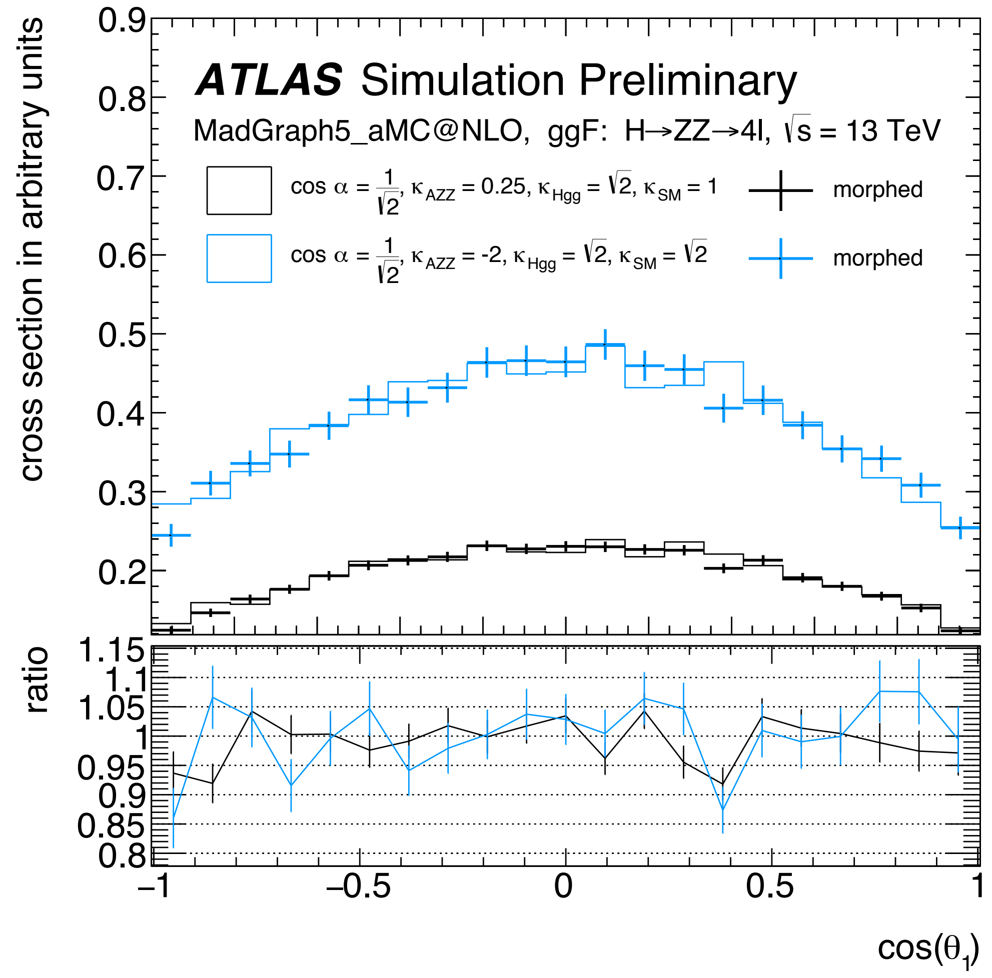
Simple example:

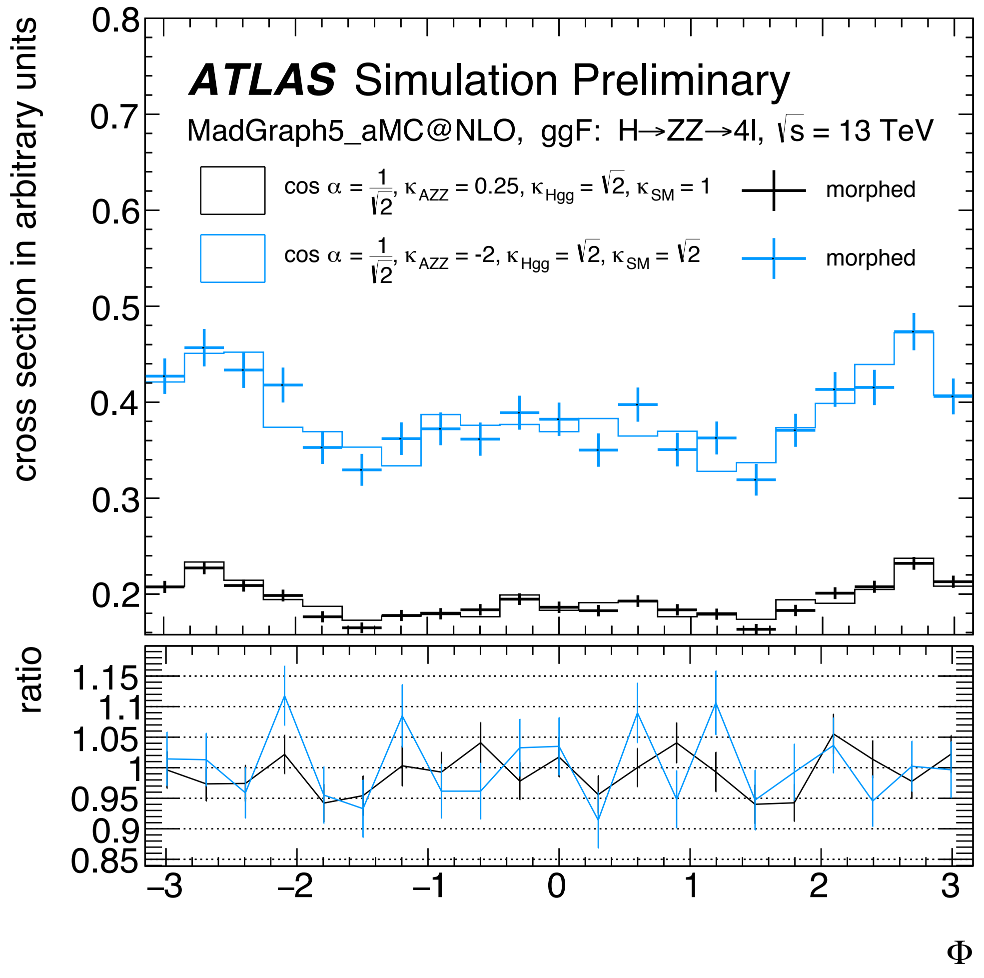
$$|g_1 M_{SM} + g_2 M_{BSM}|^2 = g_1^2 |M_{SM}|^2 + 2 g_1 g_2 Re\left[M_{SM}^* M_{BSM}\right] + g_2^2 |M_{BSM}|^2$$

3-d vector space, distribution for any point in this space is linear mixture of distribution for 3 basis samples!

(real examples need more basis samples)



# EFT Decomposition

$$d\sigma \propto \left| \underbrace{\left( \mathcal{M}^p_{\mathrm{SM}} + \sum_i \frac{f_i}{\Lambda^2} \mathcal{M}^p_i \right)}_{\text{production}} \underbrace{\left( \mathcal{M}^d_{\mathrm{SM}} + \sum_j \frac{f_j}{\Lambda^2} \mathcal{M}^d_j \right)}_{\text{decay}} \right|^2$$

Express EFT as a mixture:

$$p(x|\theta) = \sum_c w_c(\theta) p_c(x)$$

$w_c(\theta)$ are polynomials

$\nabla_\theta \log p(x|\theta)$ is now possible!

| Process | Number of components for $n$ operators | | | | | |
|---|---|---|---|---|---|---|
| | $\mathcal{O}(\Lambda^0)$ | $\mathcal{O}(\Lambda^{-2})$ | $\mathcal{O}(\Lambda^{-4})$ | $\mathcal{O}(\Lambda^{-6})$ | $\mathcal{O}(\Lambda^{-8})$ | $\sum$ |
| $hV$ / WBF production | 1 | $n$ | $\frac{n(n+1)}{2}$ | | | $\frac{(n+1)(n+2)}{2}$ |
| $h \to VV$ decay | 1 | $n$ | $\frac{n(n+1)}{2}$ | | | $\frac{(n+1)(n+2)}{2}$ |
| Production + decay | 1 | $n$ | $\frac{n(n+1)}{2}$ | $\binom{n+2}{3}$ | $\binom{n+3}{4}$ | $\binom{n+4}{4}$ |

Table 1: Number of components $c$ as given in Eq. (6) for different processes, sorted by their suppression by the EFT cutoff scale $\Lambda$.

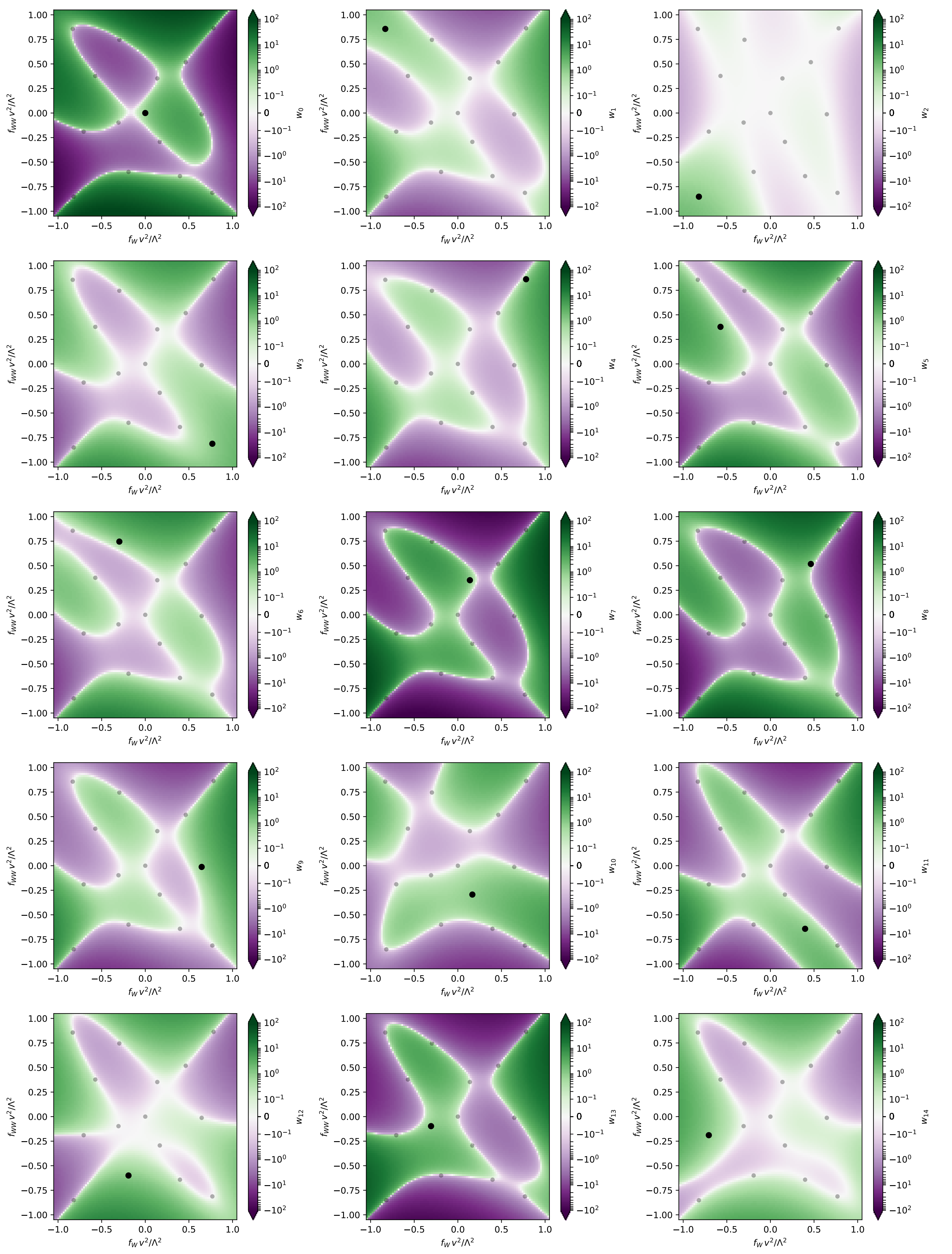


Figure 13: Morphing weights $w_i(\theta)$ for basis points distributed over the full relevant parameter space.

For 2 BSM operators affecting VBF Higgs production and decay, we need a 15-D vector space

For 5 BSM operators we need 126-D vector space

# Information Geometry

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

- Theory language: dimension-6 operators of SM EFT, $\mathcal{L} \supset \sum_i \frac{f_i}{\Lambda^2} \mathcal{O}_i$

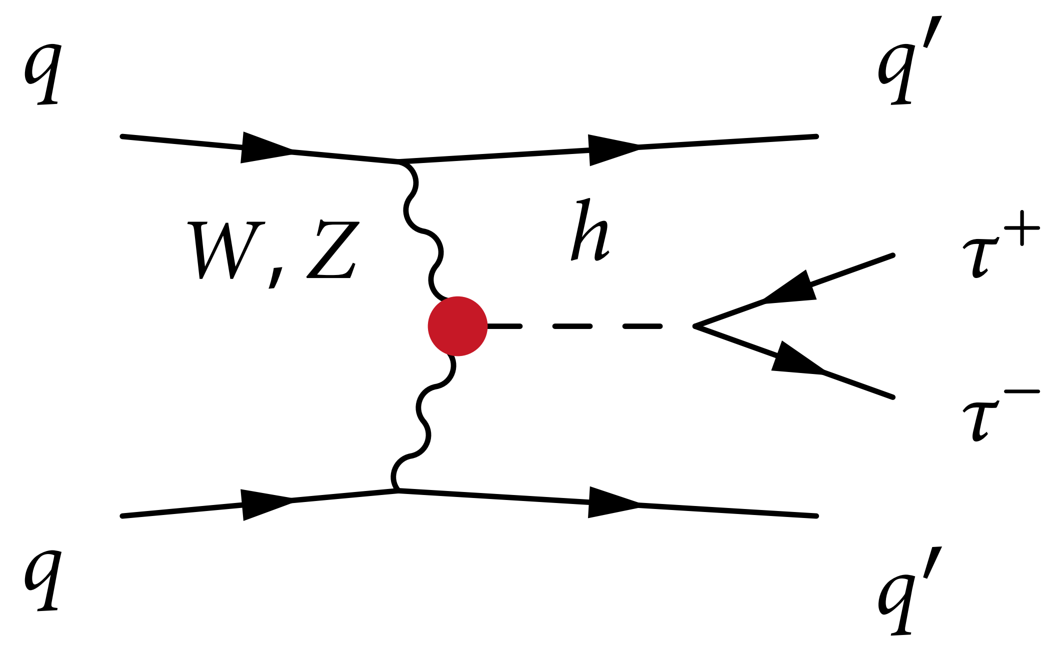
[W. Buchmuller, D. Wyler 85; K. Hagiwara, S. Ishihara, S. R. Szalapski, D. Zeppenfeld 93;
B. Grzadkowski, M. Iskrzynski, M. Misiak, J. Rosiek 1008.4884; …]

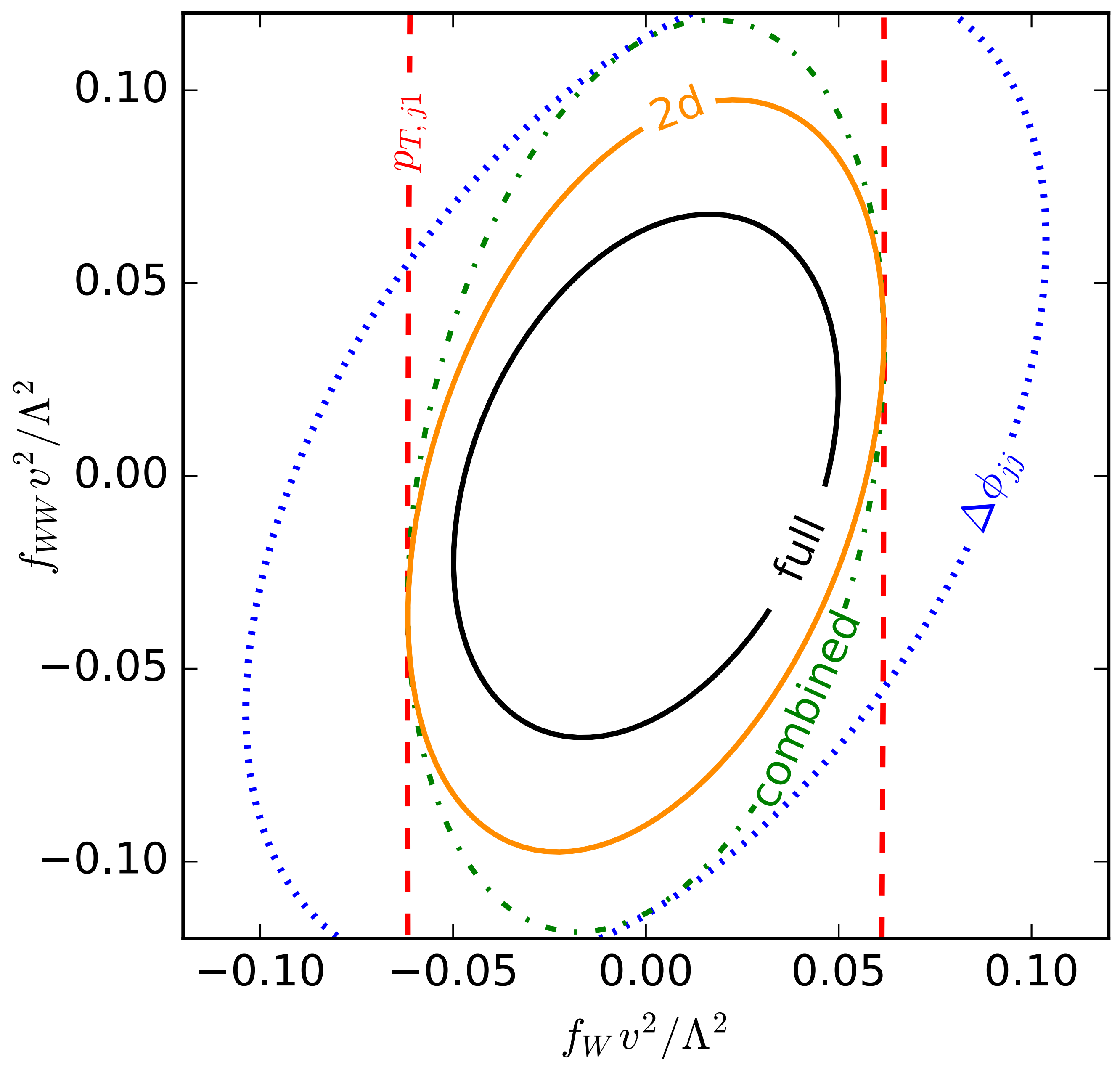
  - Total rate: $\mathcal{O}_{\phi,2} = \frac{1}{2} \partial^\mu (\phi^\dagger \phi) \, \partial_\mu (\phi^\dagger \phi)$
  - New kinematic structures:

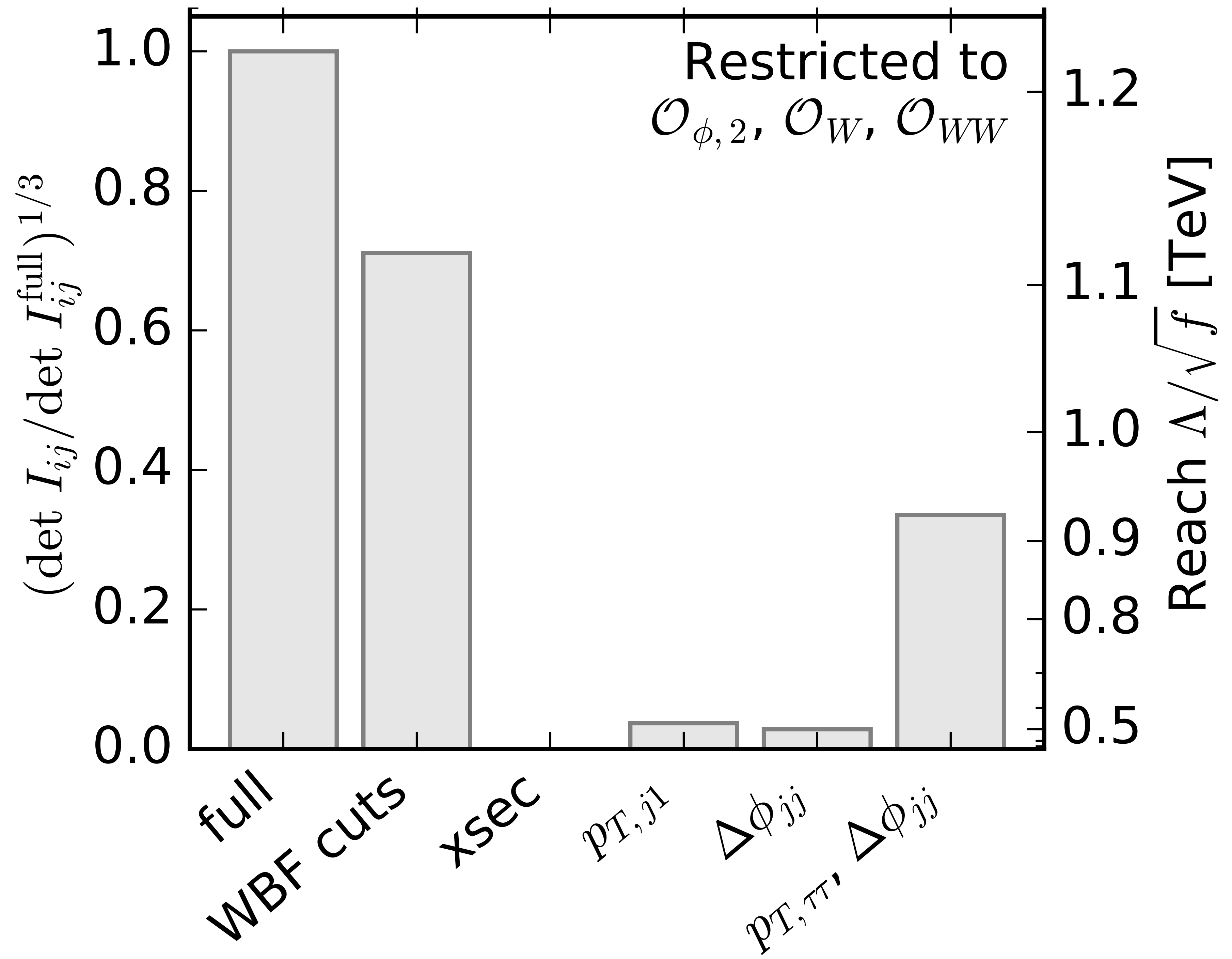
$$\mathcal{O}_B = i \frac{g}{2} (D^\mu \phi^\dagger)(D^\nu \phi) \, B_{\mu\nu} \qquad \mathcal{O}_W = i \frac{g}{2} (D^\mu \phi)^\dagger \sigma^k (D^\nu \phi) \, W^k_{\mu\nu}$$

$$\mathcal{O}_{BB} = -\frac{g'^2}{4} (\phi^\dagger \phi) \, B_{\mu\nu} \, B^{\mu\nu} \qquad \mathcal{O}_{WW} = -\frac{g^2}{4} (\phi^\dagger \phi) \, W^k_{\mu\nu} \, W^{\mu\nu\, k}$$

  - $CP$ violation: $\mathcal{O}_{W\widetilde{W}} = -\frac{g^2}{4} (\phi^\dagger \phi) \, W^k_{\mu\nu} \, \widetilde{W}^{\mu\nu\, k}$
  - Others strongly constrained by EWPD or redundant



# Information Geometry

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

- Theory language: dimension-6 operators of SM EFT, $\mathcal{L} \supset \sum_i \frac{f_i}{\Lambda^2} \mathcal{O}_i$

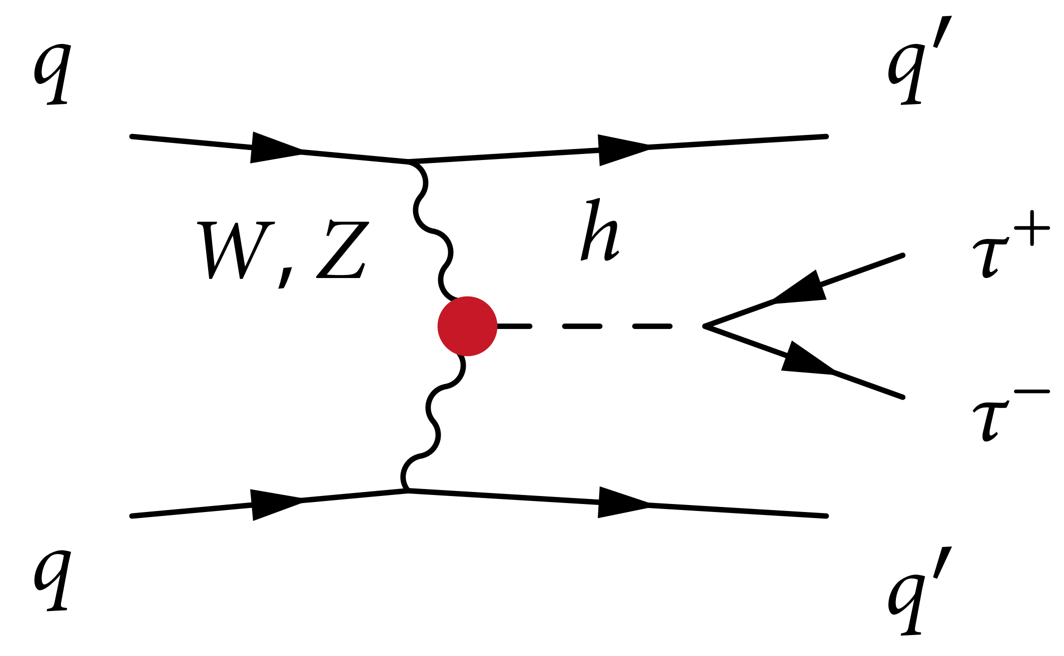
  [W. Buchmuller, D. Wyler 85; K. Hagiwara, S. Ishihara, S. R. Szalapski, D. Zeppenfeld 93; B. Grzadkowski, M. Iskrzynski, M. Misiak, J. Rosiek 1008.4884; …]

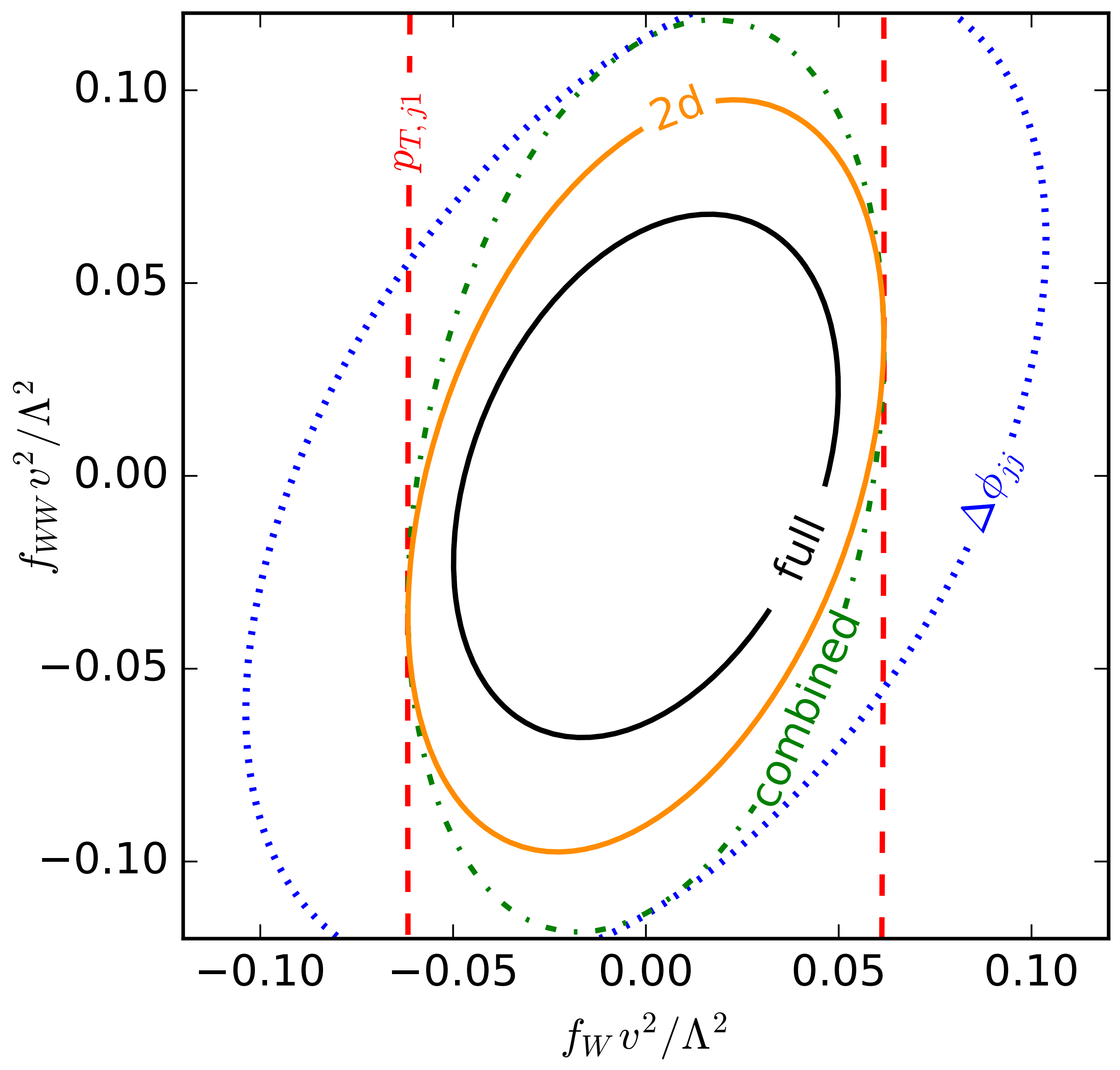
  - Total rate: $\mathcal{O}_{\phi,2} = \frac{1}{2} \partial^\mu (\phi^\dagger \phi)\, \partial_\mu (\phi^\dagger \phi)$
  - New kinematic structures:

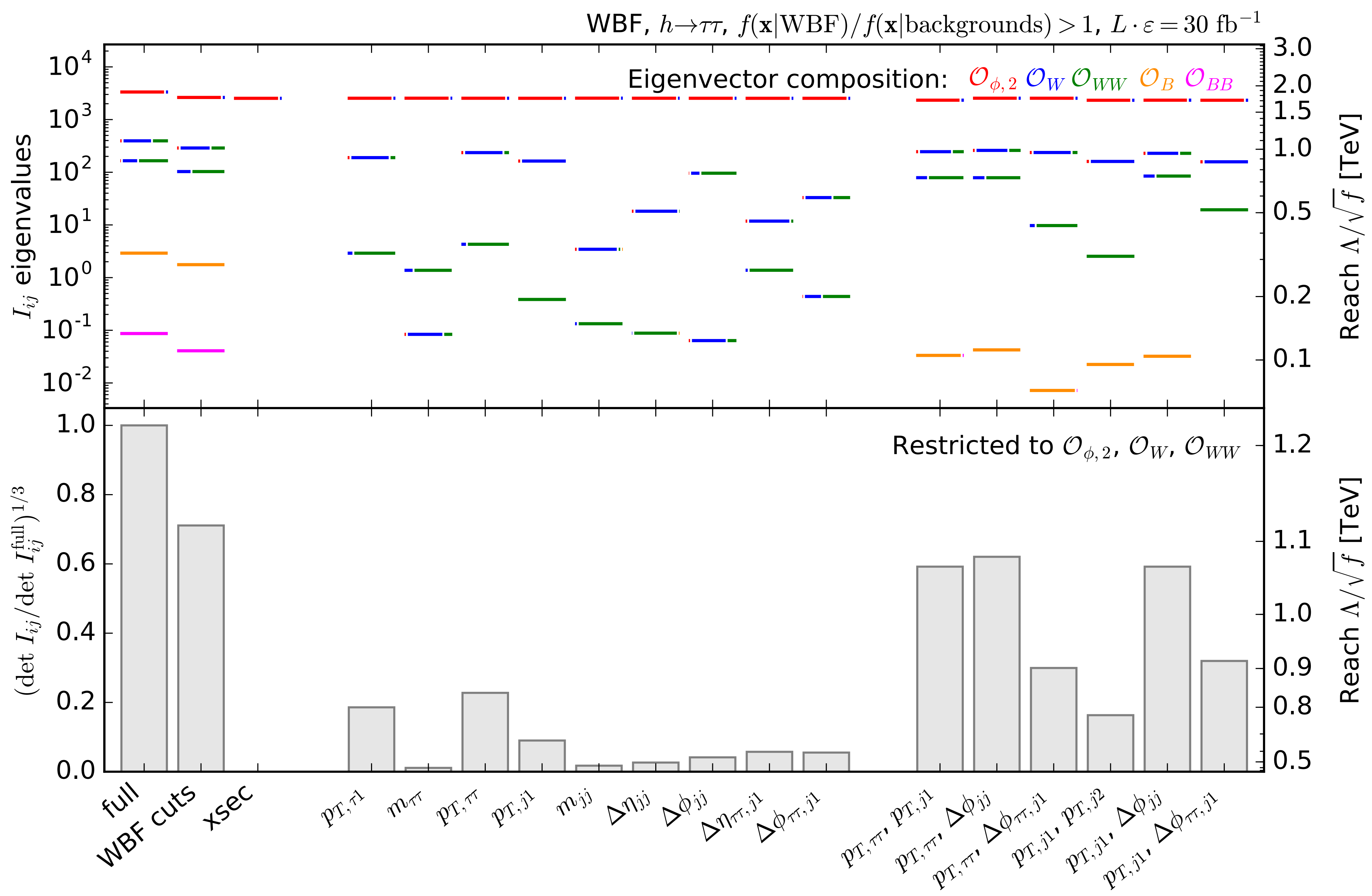
$$\mathcal{O}_B = \mathrm{i}\frac{g}{2} (D^\mu \phi^\dagger)(D^\nu \phi)\, B_{\mu\nu} \qquad \mathcal{O}_W = \mathrm{i}\frac{g}{2} (D^\mu \phi)^\dagger \sigma^k (D^\nu \phi)\, W^k_{\mu\nu}$$

$$\mathcal{O}_{BB} = -\frac{g'^2}{4} (\phi^\dagger \phi)\, B_{\mu\nu}\, B^{\mu\nu} \qquad \mathcal{O}_{WW} = -\frac{g^2}{4} (\phi^\dagger \phi)\, W^k_{\mu\nu}\, W^{\mu\nu\, k}$$

  - *CP* violation: $\mathcal{O}_{W\widetilde{W}} = -\frac{g^2}{4} (\phi^\dagger \phi)\, W^k_{\mu\nu}\, \widetilde{W}^{\mu\nu\, k}$
  - Others strongly constrained by EWPD or redundant



# Fisher Information

Brehmer, Kling, Plehn, Cranmer [arXiv:1612.05261]
Brehmer, Kling, Plehn, Tait [arXiv: 1712.02350]

## Information geometry provides a very powerful tool for phenomenology of EFT

- formal bounds on how well parameters can be measured
- exploit fully differential cross-section
- Eg: a global fit (13 parameters) & can profile/marginalize parameters you aren't interested in (eg. CP violating vs. CP conserving)

For an effective Higgs-gauge Lagrangian truncated at mass dimension six,

$$\mathcal{L} = \mathcal{L}_{\rm SM} + \frac{f_i}{\Lambda^2}\mathcal{O}_i \tag{9}$$

our $CP$-even reference scenario consists of the renormalizable Standard Model Lagrangian combined with the five $CP$-even dimension-six operators in the HISZ basis [6, 7, 35],

$$\begin{aligned}
\mathcal{O}_B &= i\frac{g}{2}\,(D^\mu\phi^\dagger)(D^\nu\phi)\,B_{\mu\nu} & \mathcal{O}_W &= i\frac{g}{2}\,(D^\mu\phi)^\dagger\sigma^k(D^\nu\phi)\,W^k_{\mu\nu}\\
\mathcal{O}_{BB} &= -\frac{g'^2}{4}\,(\phi^\dagger\phi)\,B_{\mu\nu}\,B^{\mu\nu} & \mathcal{O}_{WW} &= -\frac{g^2}{4}\,(\phi^\dagger\phi)\,W^k_{\mu\nu}\,W^{\mu\nu\,k}\\
\mathcal{O}_{\phi,2} &= \frac{1}{2}\,\partial^\mu(\phi^\dagger\phi)\,\partial_\mu(\phi^\dagger\phi)\,.
\end{aligned} \tag{10}$$

At the same mass dimension, $CP$-odd couplings are described by operators

$$\begin{aligned}
\mathcal{O}_{B\widetilde{B}} &= -\frac{g'^2}{4}\,(\phi^\dagger\phi)\,\widetilde{B}_{\mu\nu}\,B^{\mu\nu} \equiv -\frac{g'^2}{4}\,(\phi^\dagger\phi)\,\epsilon_{\mu\nu\rho\sigma}B^{\rho\sigma}\,B^{\mu\nu}\\
\mathcal{O}_{W\widetilde{W}} &= -\frac{g^2}{4}\,(\phi^\dagger\phi)\,\widetilde{W}^k_{\mu\nu}\,W^{\mu\nu\,k} \equiv -\frac{g^2}{4}\,(\phi^\dagger\phi)\,\epsilon_{\mu\nu\rho\sigma}W^{\rho\sigma\,k}\,W^{\mu\nu\,k}\,.
\end{aligned} \tag{11}$$

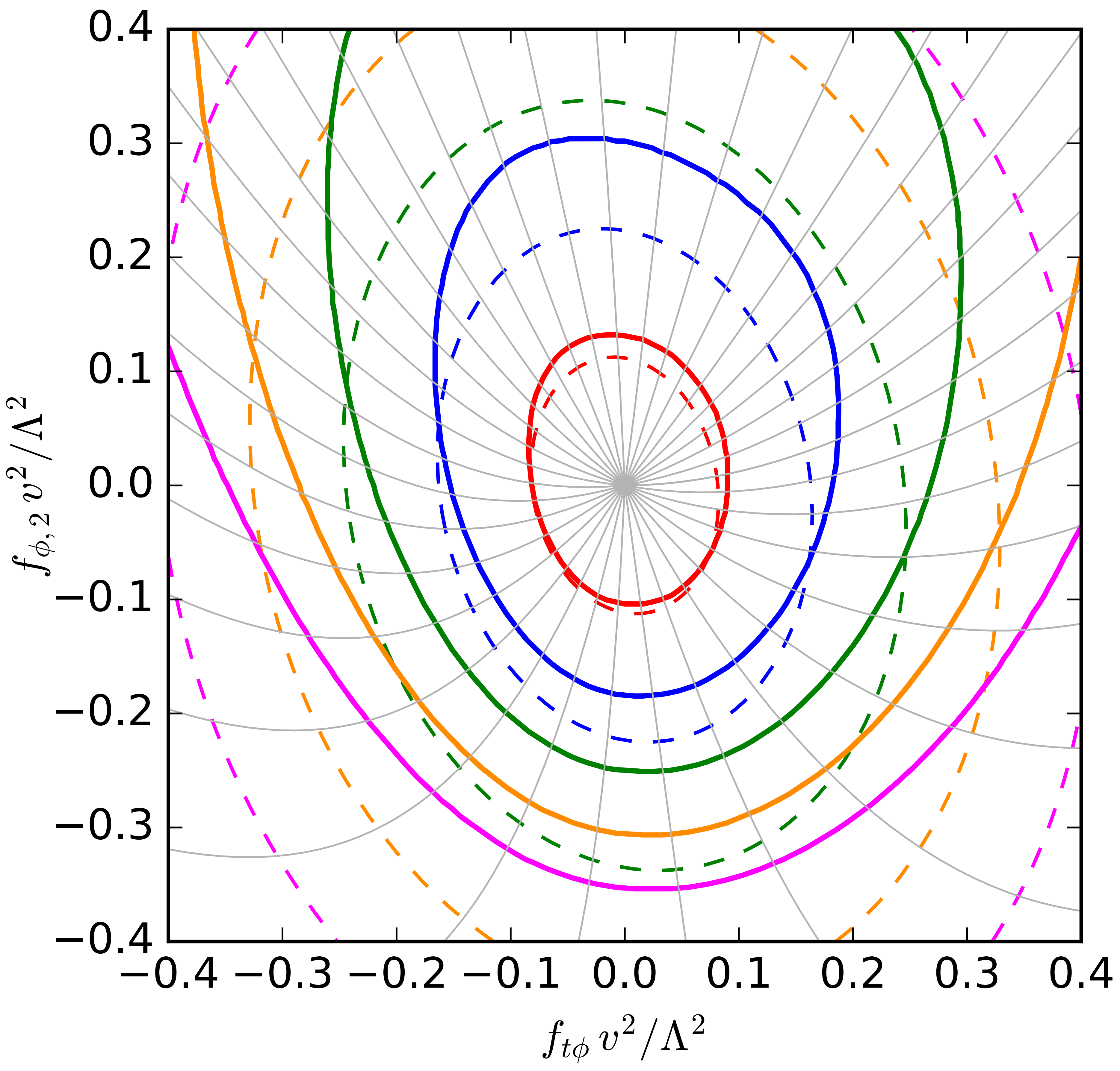
With the Levi-Civita tensor, these operators break down as $C$-conserving and $P$-violating.

$$I_{ij} = \begin{array}{c} \begin{array}{ccccccccccccc} f_{\phi,2} & f_W & f_B & f_{WW} & f_{BB} & f_{W\widetilde{W}} & f_{B\widetilde{B}} & \text{Im}\,f_W & \text{Im}\,f_B & \text{Im}\,f_{WW} & \text{Im}\,f_{BB} & \text{Im}\,f_{W\widetilde{W}} & \text{Im}\,f_{B\widetilde{B}} \end{array} \\
\left(\begin{array}{ccccccccccccc}
4942 & -968 & -50 & 54 & 2 & -7 & 0 & -1 & 0 & 2 & 0 & 36 & 0\\
-968 & 715 & 35 & -191 & -3 & 1 & 0 & 0 & 0 & 0 & 0 & -55 & -1\\
-50 & 35 & 6 & -9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0\\
54 & -191 & -9 & 321 & 3 & -1 & 0 & 0 & 0 & 1 & 0 & 72 & 1\\
2 & -3 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\
-7 & 1 & 0 & -1 & 0 & 359 & 4 & 41 & 1 & -81 & -1 & -1 & 0\\
0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0\\
-1 & 0 & 0 & 0 & 0 & 41 & 0 & 6 & 0 & -12 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
2 & 0 & 0 & 1 & 0 & -81 & -1 & -12 & 0 & 23 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
36 & -55 & -2 & 72 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 21 & 0\\
0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right) \end{array}
\begin{array}{c} f_{\phi,2}\\ f_W\\ f_B\\ f_{WW}\\ f_{BB}\\ f_{W\widetilde{W}}\\ f_{B\widetilde{B}}\\ \text{Im}\,f_W\\ \text{Im}\,f_B\\ \text{Im}\,f_{WW}\\ \text{Im}\,f_{BB}\\ \text{Im}\,f_{W\widetilde{W}}\\ \text{Im}\,f_{B\widetilde{B}} \end{array}, \tag{30}$$

# The Information Geometry of an EFT



- With a metric, it now makes sense to talk about how far two points in theory space are from one another. And we can go beyond the ellipse and calculate geodesics!

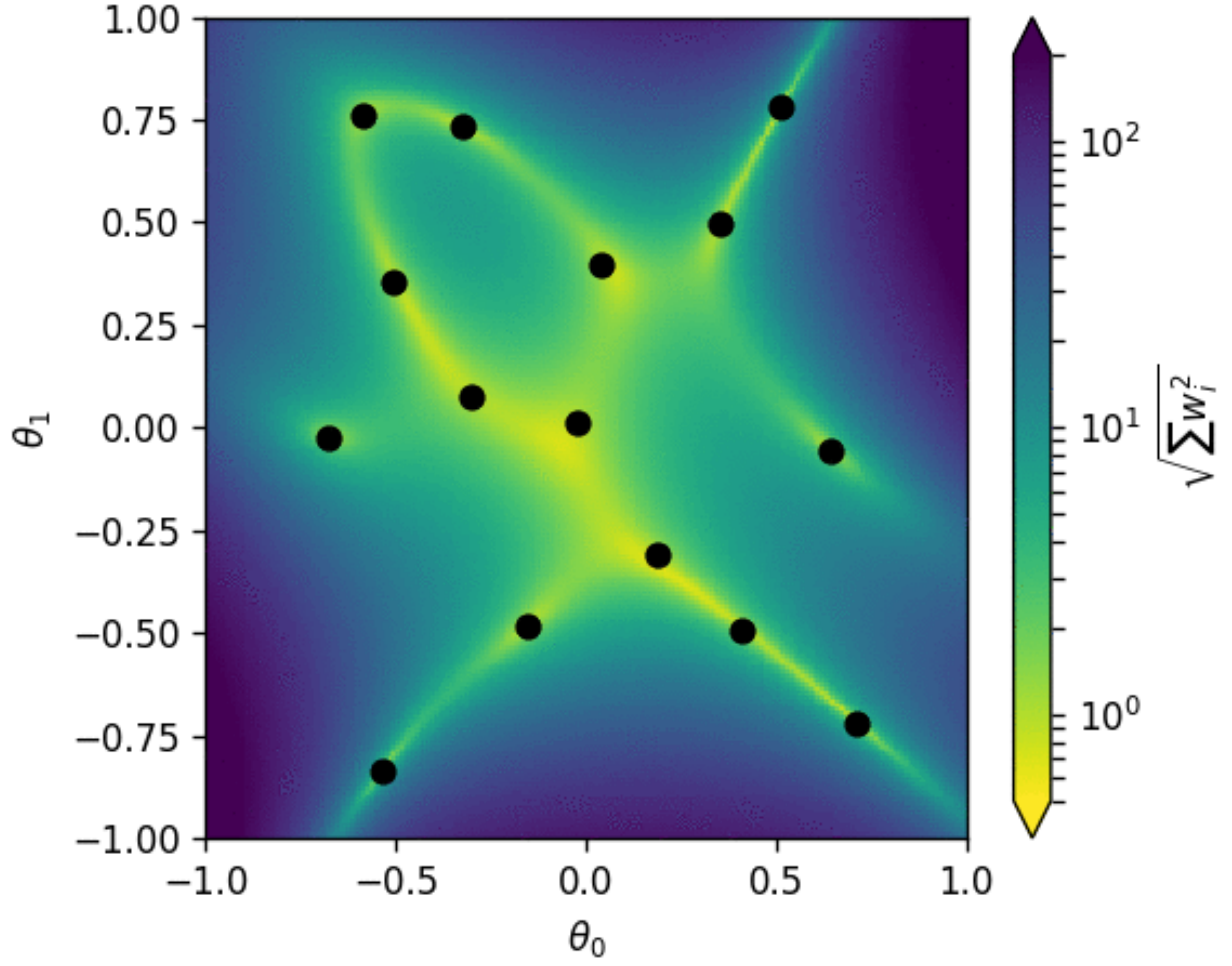



- Parametrization-independent, geometric description of sensitivity based on Fisher information

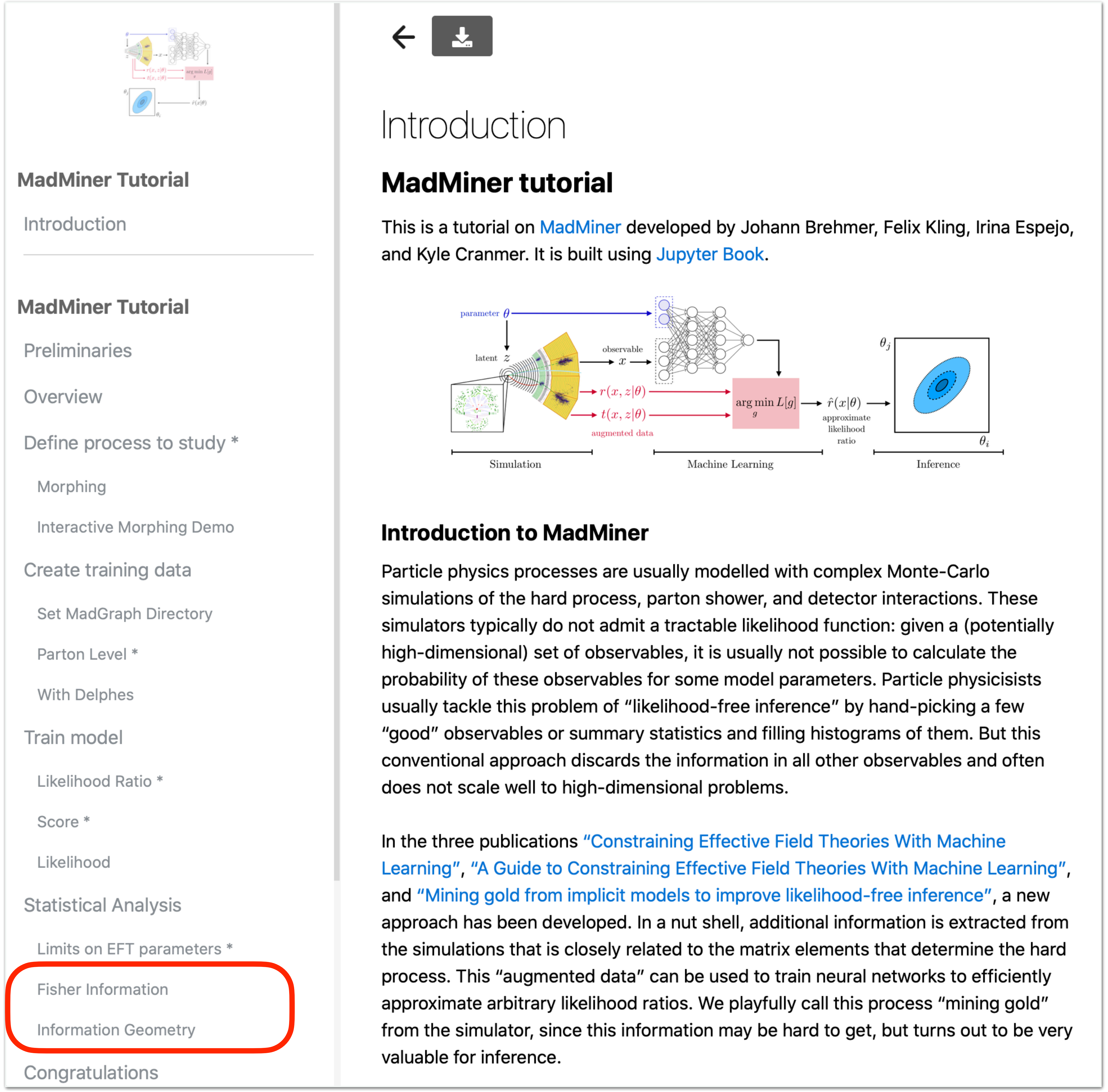
# MadMiner: Hands-on Tutorial

MadMiner automates the morphing for an arbitrary EFT and has utilities to to exploit information geometry to understand the phenomenology of EFT at a specific collider

- formal bounds on how well parameters can be measured
- exploit fully differential cross-section
- useful for Snowmass!




https://cranmer.github.io/madminer-tutorial/



MadMiner Tutorial

Introduction

MadMiner Tutorial

Preliminaries

Overview

Define process to study *

Morphing

Interactive Morphing Demo

Create training data

Set MadGraph Directory

Parton Level *

With Delphes

Train model

Likelihood Ratio *

Score *

Likelihood

Statistical Analysis

Limits on EFT parameters *

Fisher Information

Information Geometry

Congratulations

Introduction

**MadMiner tutorial**

This is a tutorial on MadMiner developed by Johann Brehmer, Felix Kling, Irina Espejo, and Kyle Cranmer. It is built using Jupyter Book.



**Introduction to MadMiner**

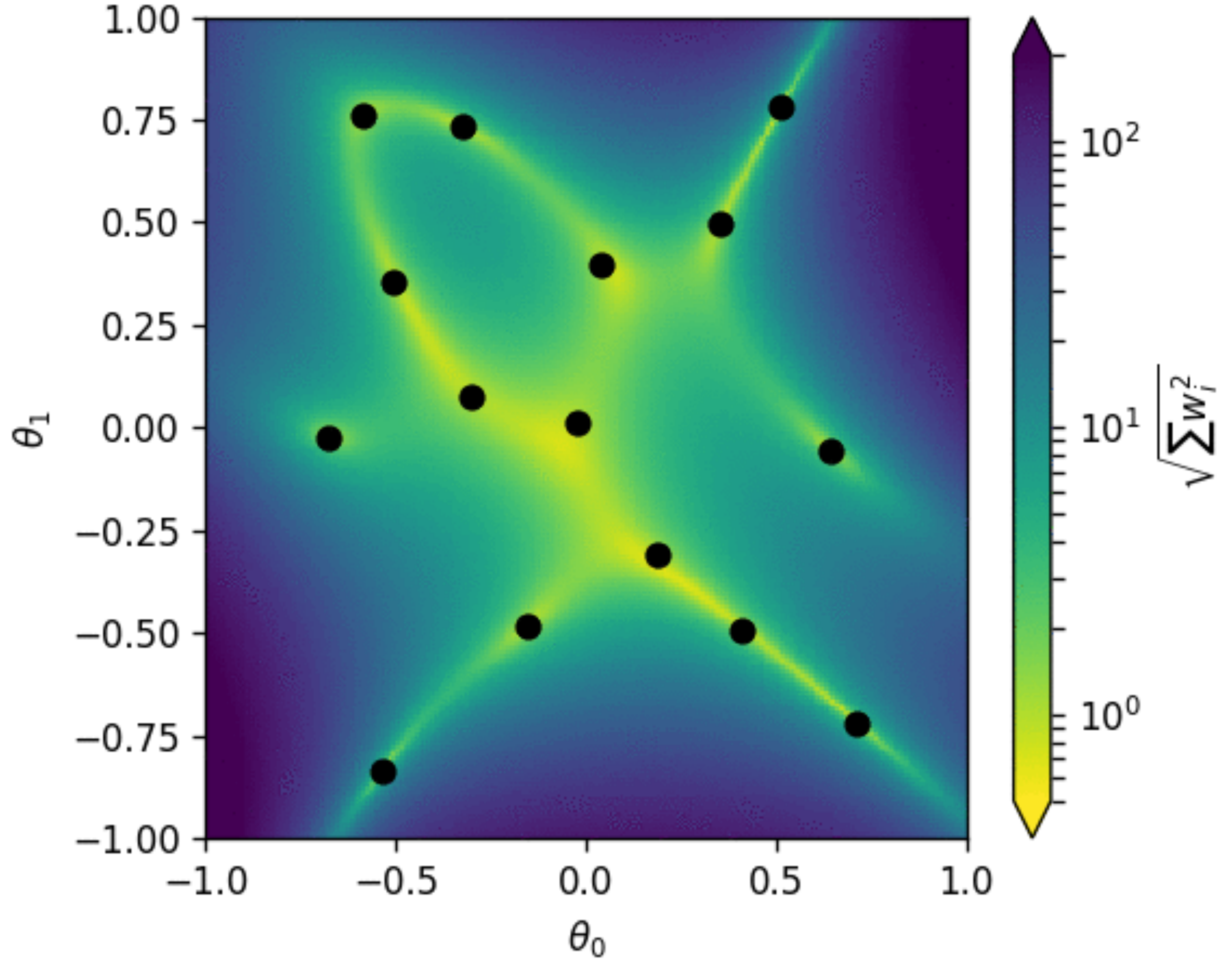
Particle physics processes are usually modelled with complex Monte-Carlo simulations of the hard process, parton shower, and detector interactions. These simulators typically do not admit a tractable likelihood function: given a (potentially high-dimensional) set of observables, it is usually not possible to calculate the probability of these observables for some model parameters. Particle physicisists usually tackle this problem of "likelihood-free inference" by hand-picking a few "good" observables or summary statistics and filling histograms of them. But this conventional approach discards the information in all other observables and often does not scale well to high-dimensional problems.

In the three publications "Constraining Effective Field Theories With Machine Learning", "A Guide to Constraining Effective Field Theories With Machine Learning", and "Mining gold from implicit models to improve likelihood-free inference", a new approach has been developed. In a nut shell, additional information is extracted from the simulations that is closely related to the matrix elements that determine the hard process. This "augmented data" can be used to train neural networks to efficiently approximate arbitrary likelihood ratios. We playfully call this process "mining gold" from the simulator, since this information may be hard to get, but turns out to be very valuable for inference.

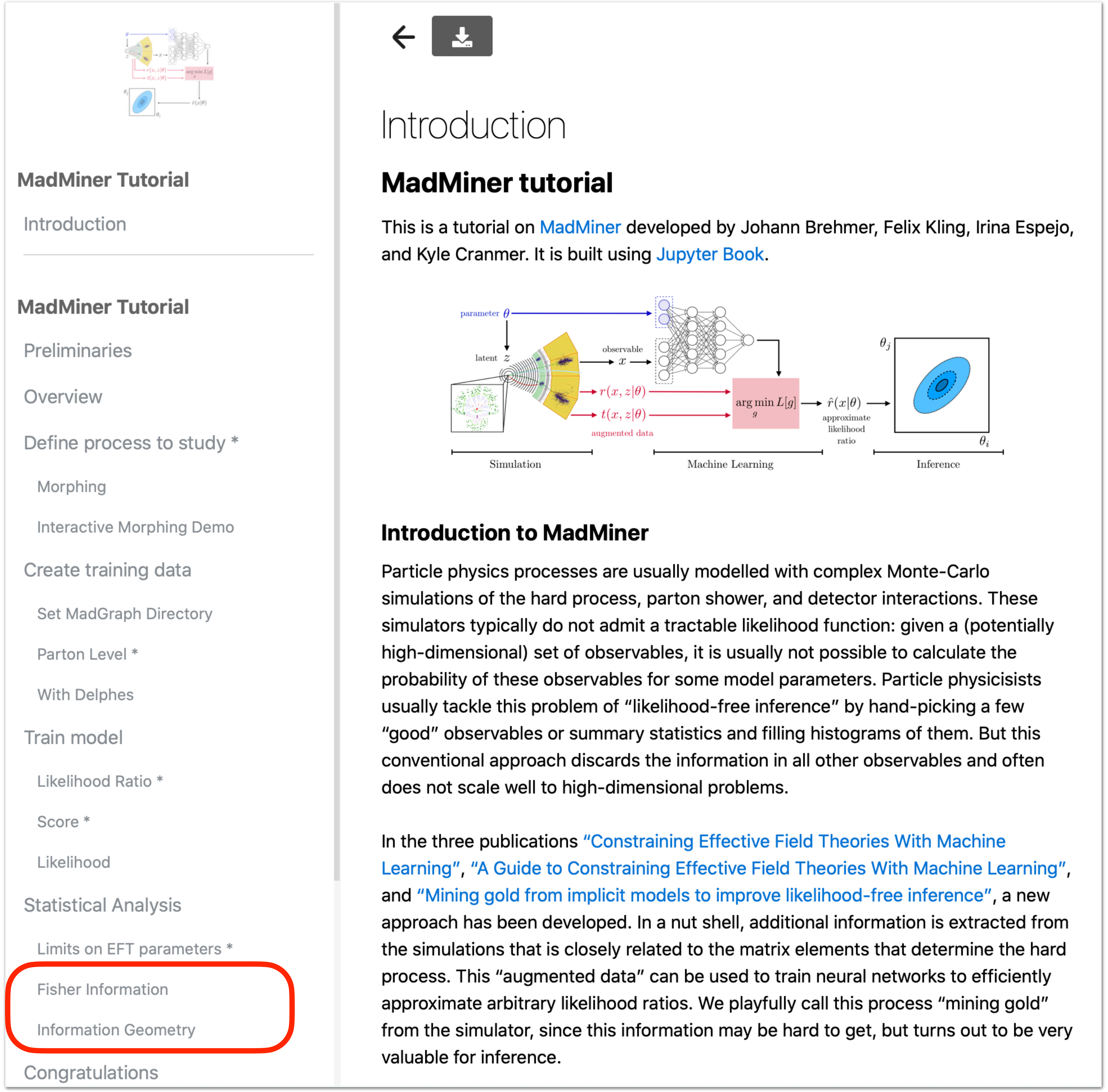
# MadMiner: Hands-on Tutorial

MadMiner automates the morphing for an arbitrary EFT and has utilities to to exploit information geometry to understand the phenomenology of EFT at a specific collider

- formal bounds on how well parameters can be measured
- exploit fully differential cross-section
- useful for Snowmass!




https://cranmer.github.io/madminer-tutorial/



MadMiner Tutorial

Introduction

MadMiner Tutorial

Preliminaries

Overview

Define process to study *

Morphing

Interactive Morphing Demo

Create training data

Set MadGraph Directory

Parton Level *

With Delphes

Train model

Likelihood Ratio *

Score *

Likelihood

Statistical Analysis

Limits on EFT parameters *

Fisher Information

Information Geometry

Congratulations

Introduction

**MadMiner tutorial**

This is a tutorial on MadMiner developed by Johann Brehmer, Felix Kling, Irina Espejo, and Kyle Cranmer. It is built using Jupyter Book.



**Introduction to MadMiner**

Particle physics processes are usually modelled with complex Monte-Carlo simulations of the hard process, parton shower, and detector interactions. These simulators typically do not admit a tractable likelihood function: given a (potentially high-dimensional) set of observables, it is usually not possible to calculate the probability of these observables for some model parameters. Particle physicisists usually tackle this problem of "likelihood-free inference" by hand-picking a few "good" observables or summary statistics and filling histograms of them. But this conventional approach discards the information in all other observables and often does not scale well to high-dimensional problems.

In the three publications "Constraining Effective Field Theories With Machine Learning", "A Guide to Constraining Effective Field Theories With Machine Learning", and "Mining gold from implicit models to improve likelihood-free inference", a new approach has been developed. In a nut shell, additional information is extracted from the simulations that is closely related to the matrix elements that determine the hard process. This "augmented data" can be used to train neural networks to efficiently approximate arbitrary likelihood ratios. We playfully call this process "mining gold" from the simulator, since this information may be hard to get, but turns out to be very valuable for inference.

Now for some bad news….

Particle physics processes do not have a tractable likelihood function.

# Modeling particle physics processes

Theory parameters

$\theta$

Evolution

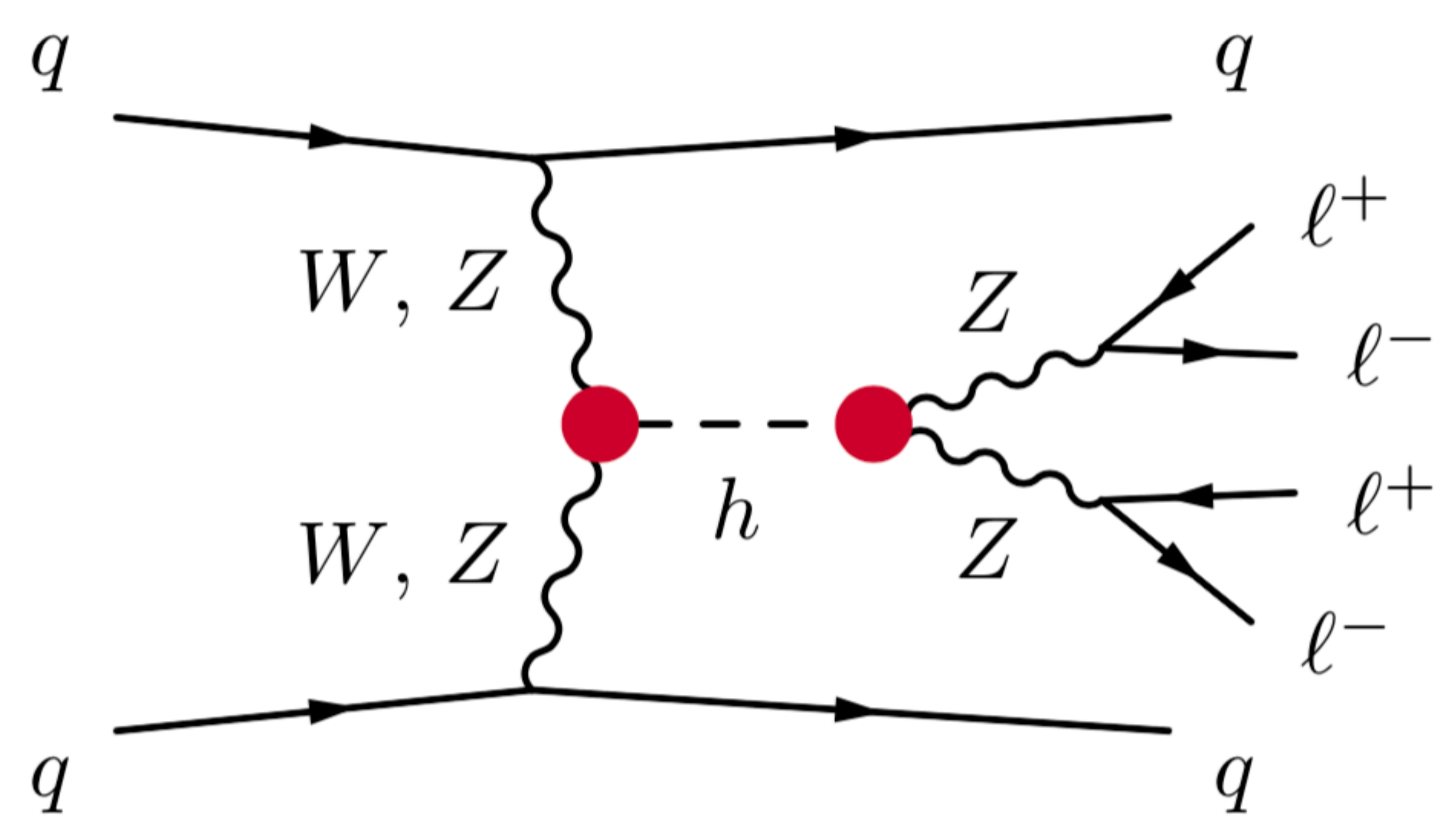
# Modeling particle physics processes

Latent variables

Parton-level momenta

Theory parameters

$z_p \longleftarrow \theta$

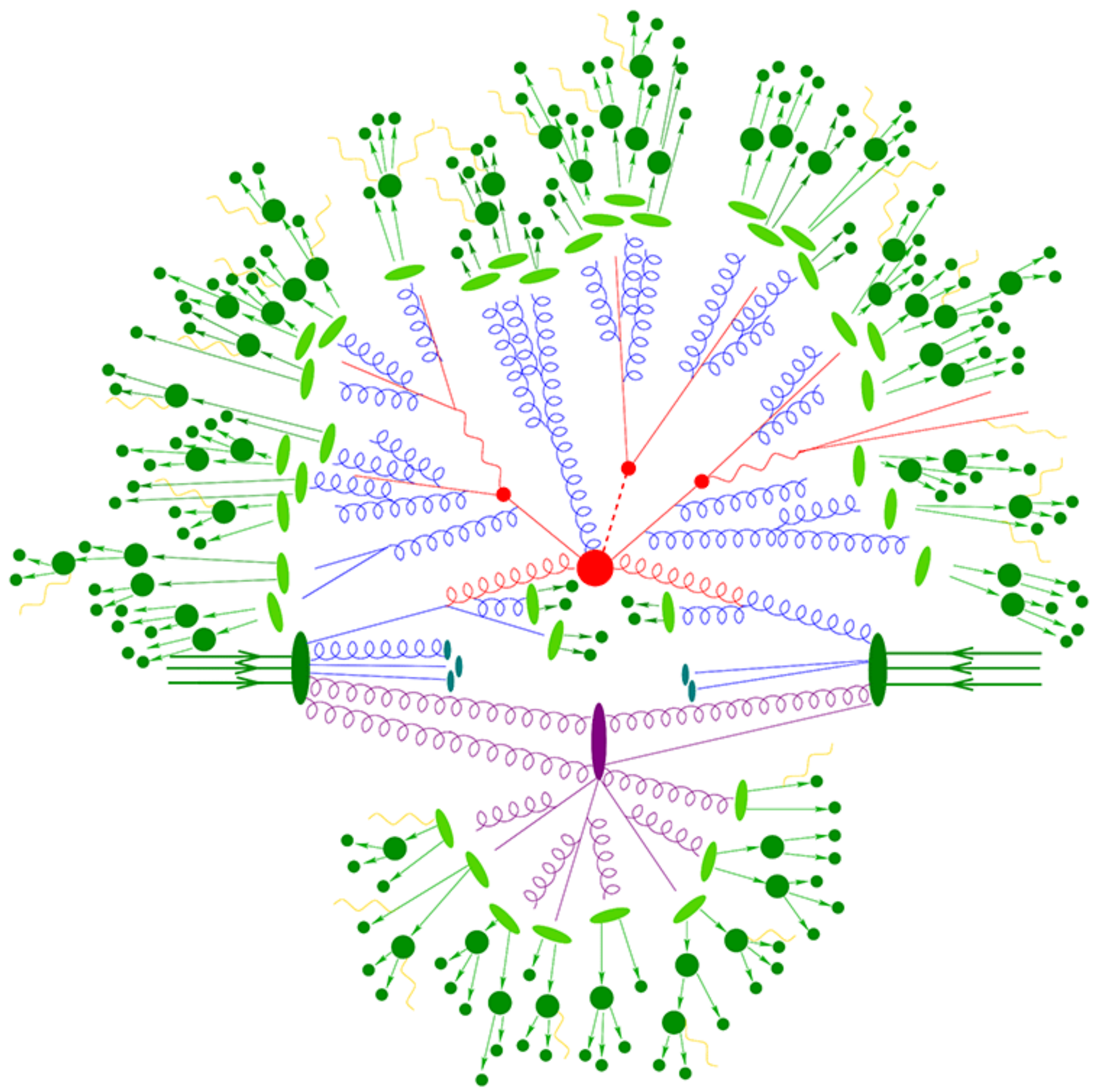



Evolution

# Modeling particle physics processes

Latent variables

| Shower splittings | Parton-level momenta | Theory parameters |
|---|---|---|
| $z_s$ ← | $z_p$ ← | $\theta$ |



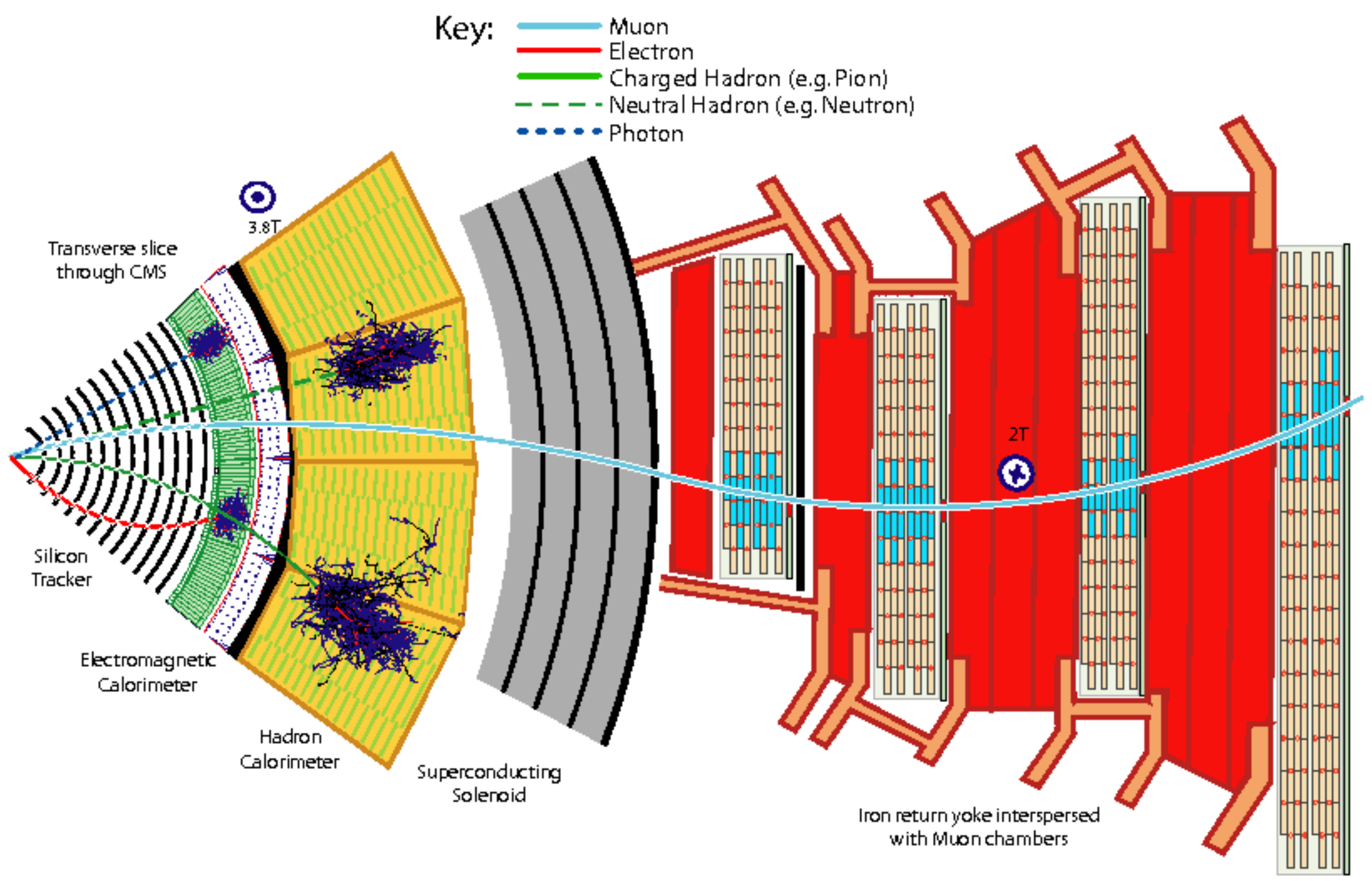
[F. Krauss]

← Evolution

# Modeling particle physics processes

Latent variables

| Detector interactions | Shower splittings | Parton-level momenta | Theory parameters |
| --- | --- | --- | --- |
| $z_d$ ← | $z_s$ ← | $z_p$ ← | $\theta$ |



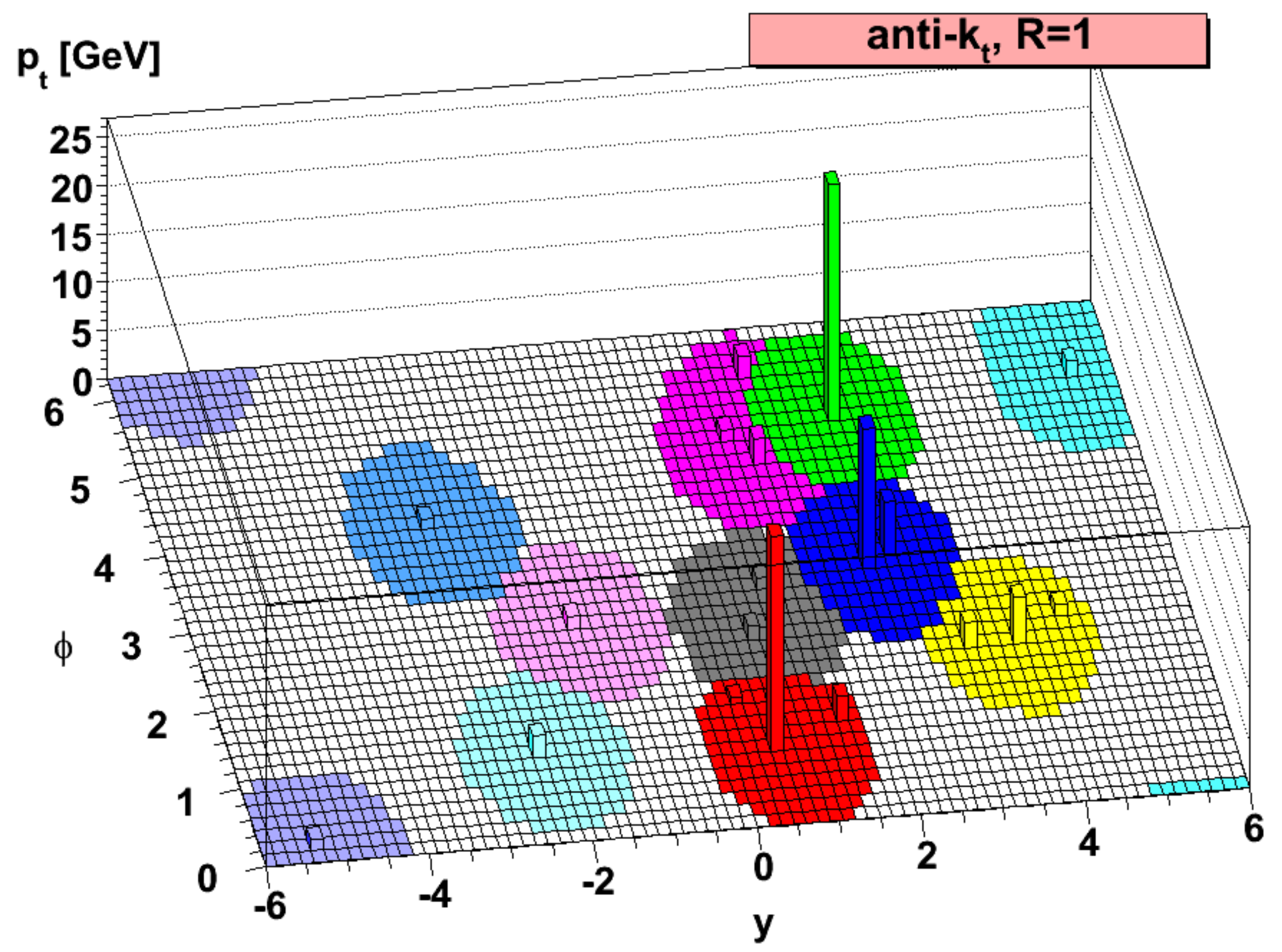

[CMS]

Evolution

# Modeling particle physics processes

Latent variables

Observables $x$ ← Detector interactions $z_d$ ← Shower splittings $z_s$ ← Parton-level momenta $z_p$ ← Theory parameters $\theta$




[M. Cacciari, G. Salam, G. Soyez 0802.1189]

Evolution

# Modeling particle physics processes

Latent variables

| Observables | | Detector interactions | Shower splittings | Parton-level momenta | Theory parameters |
|---|---|---|---|---|---|
| $x$ | ← | $z_d$ ← | $z_s$ ← | $z_p$ ← | $\theta$ |

Sample from $p(x|z_d)$ $p(z_d|z_s)$ $p(z_s|z_p)$ $p(z_p|\theta)$

GEANT4 A SIMULATION TOOLKIT

MADGRAPH5_aMC@NLO

← Prediction (simulation)

# Modeling particle physics processes

Latent variables

| Observables | Detector interactions | Shower splittings | Parton-level momenta | Theory parameters |
|---|---|---|---|---|
| $x$ | $z_d$ | $z_s$ | $z_p$ | $\theta$ |

$$x \longleftarrow z_d \longleftarrow z_s \longleftarrow z_p \longleftarrow \theta$$

$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \quad p(x|z_d) \quad p(z_d|z_s) \quad p(z_s|z_p) \quad p(z_p|\theta)$$

Inference →

# Modeling particle physics processes

Latent variables

| Observables | Detector interactions | Shower splittings | Parton-level momenta | Theory parameters |
| --- | --- | --- | --- | --- |
| $x$ | $z_d$ | $z_s$ | $z_p$ | $\theta$ |

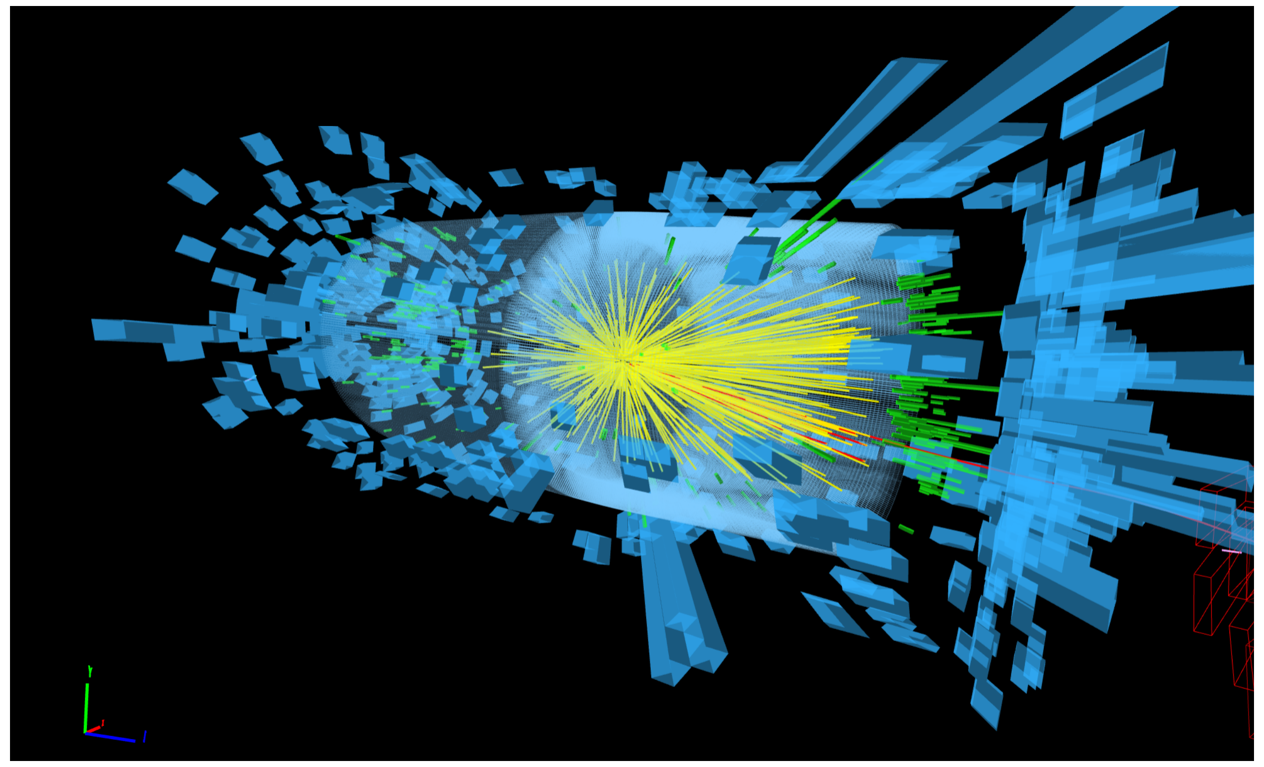
$x \leftarrow z_d \leftarrow z_s \leftarrow z_p \leftarrow \theta$

$$p(x|\theta) = \boxed{\int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p} \; p(x|z_d) \quad p(z_d|z_s) \quad p(z_s|z_p) \quad p(z_p|\theta)$$

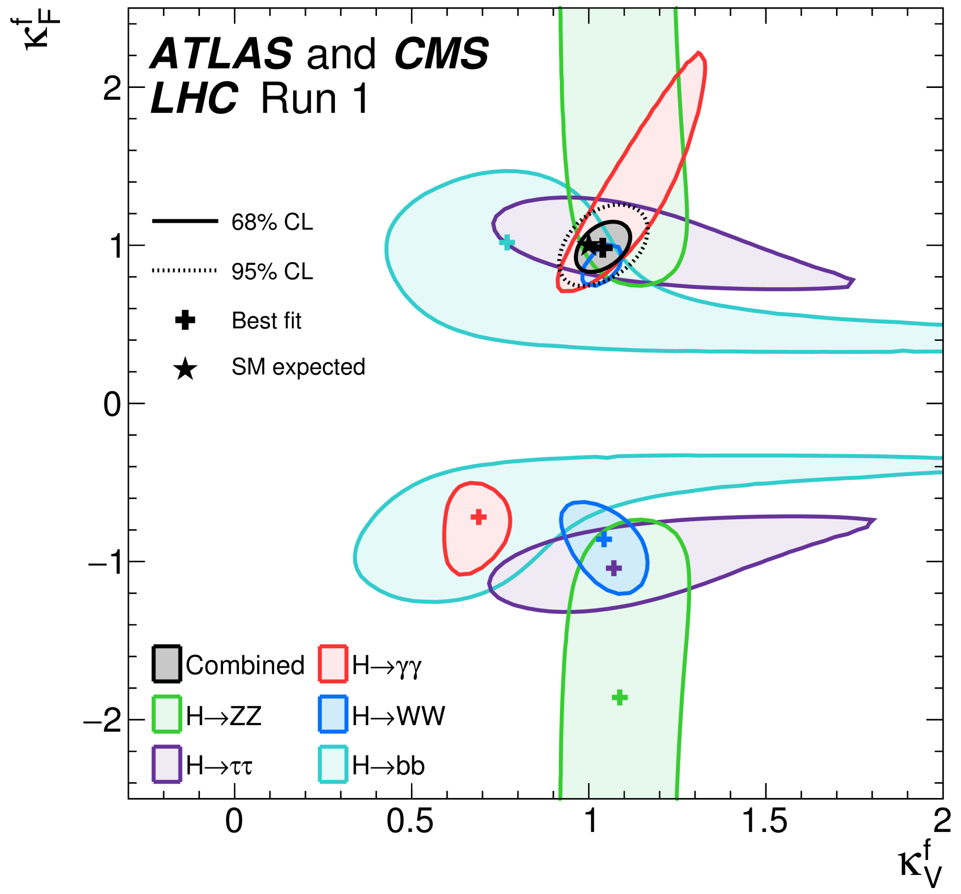
It's infeasible to calculate the integral over this enormous space!

(More subtle: We cannot sample from $p(z|x,\theta)$ efficiently.)

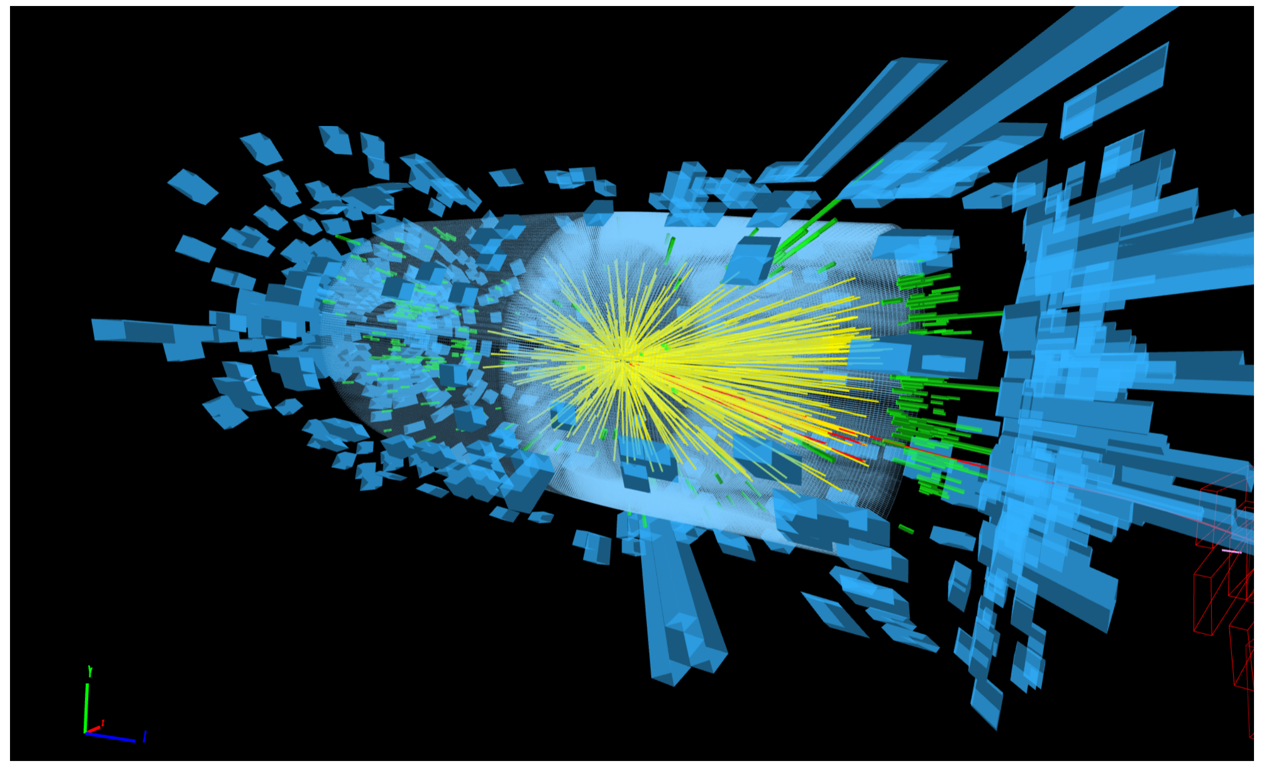
→ Inference



High-dimensional
event data $x$




Constraints on
parameters $\theta$

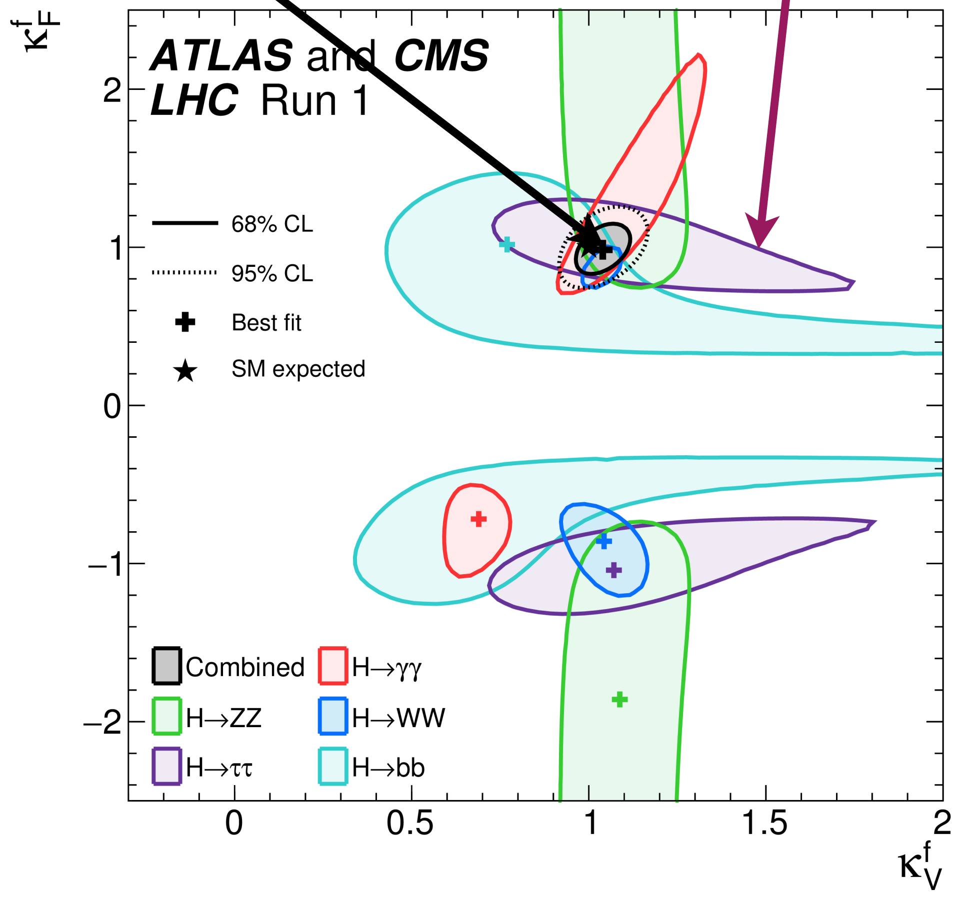


High-dimensional event data $x$

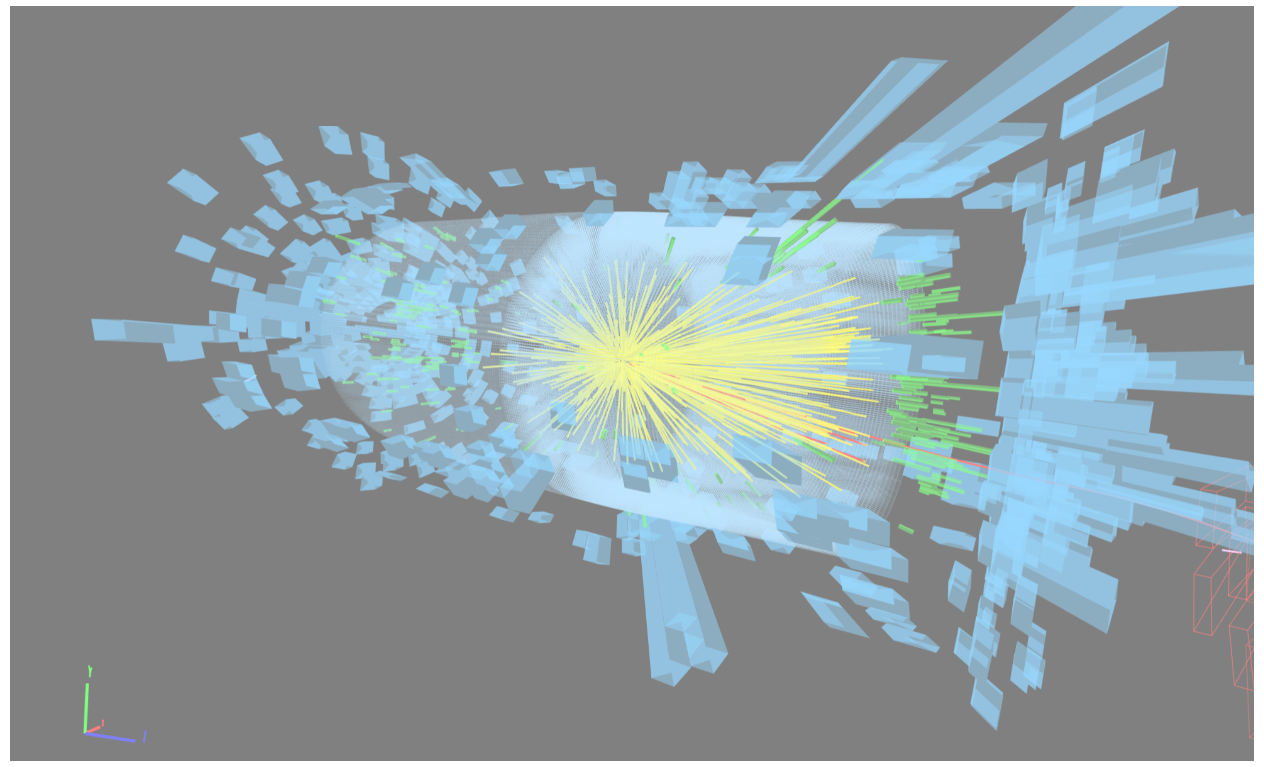
Likelihood function $p(x|\theta)$

Maximum-likelihood estimator

Confidence limits based on likelihood ratio tests

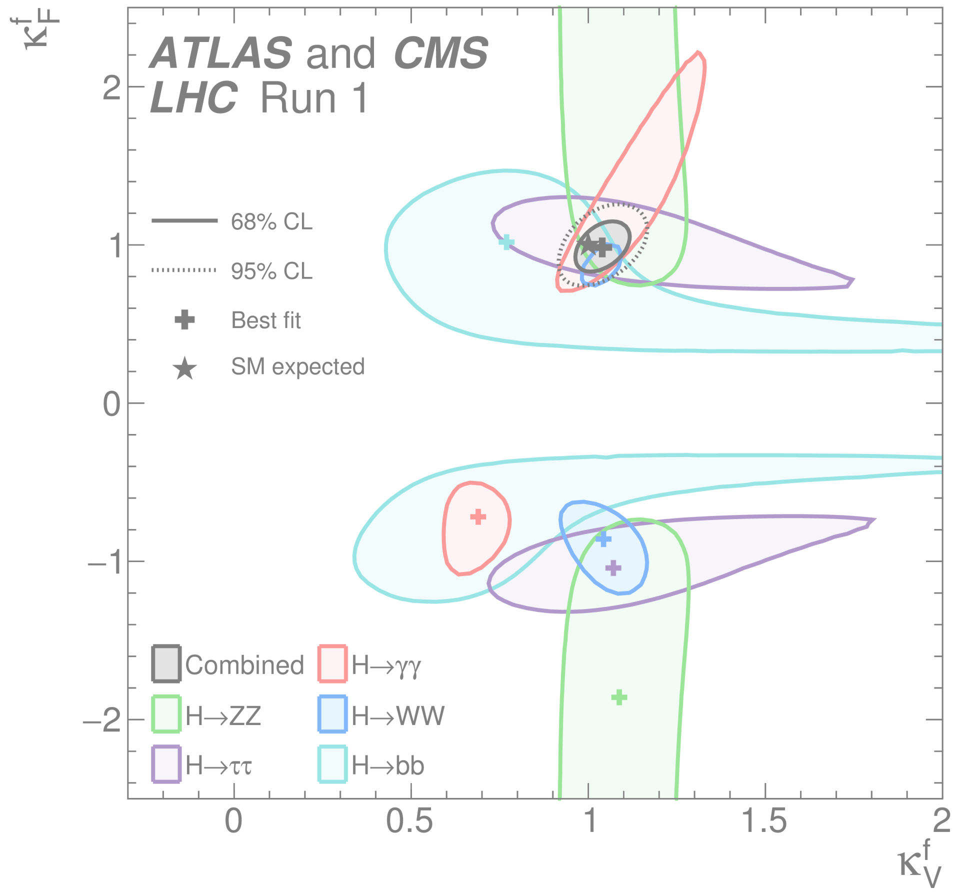



Constraints on parameters $\theta$



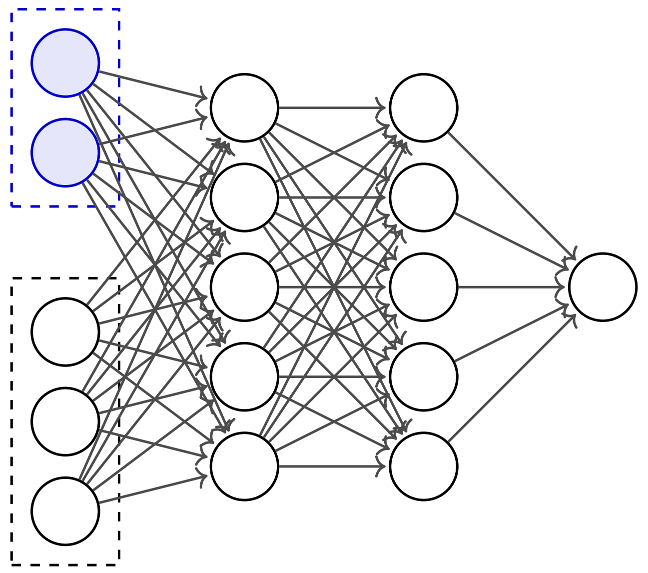
High-dimensional event data $x$

Likelihood function
$p(x|\theta)$

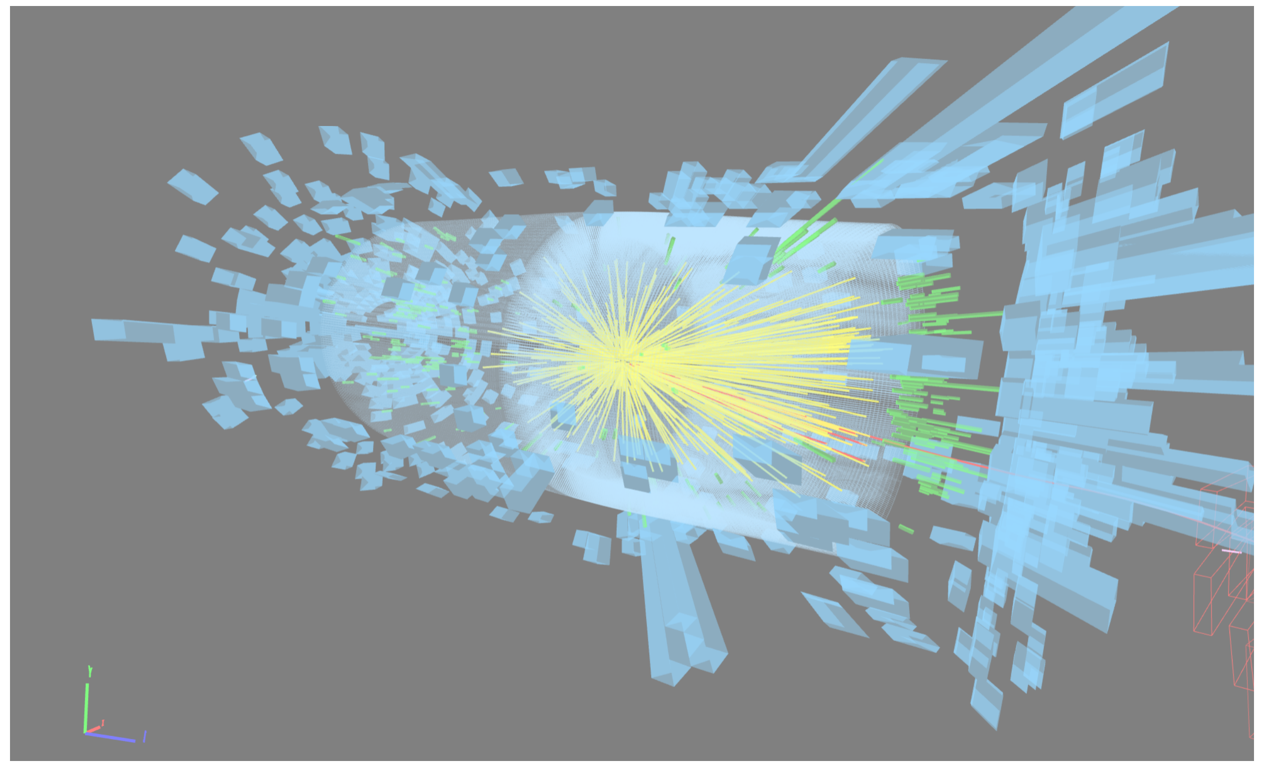



Constraints on parameters $\theta$

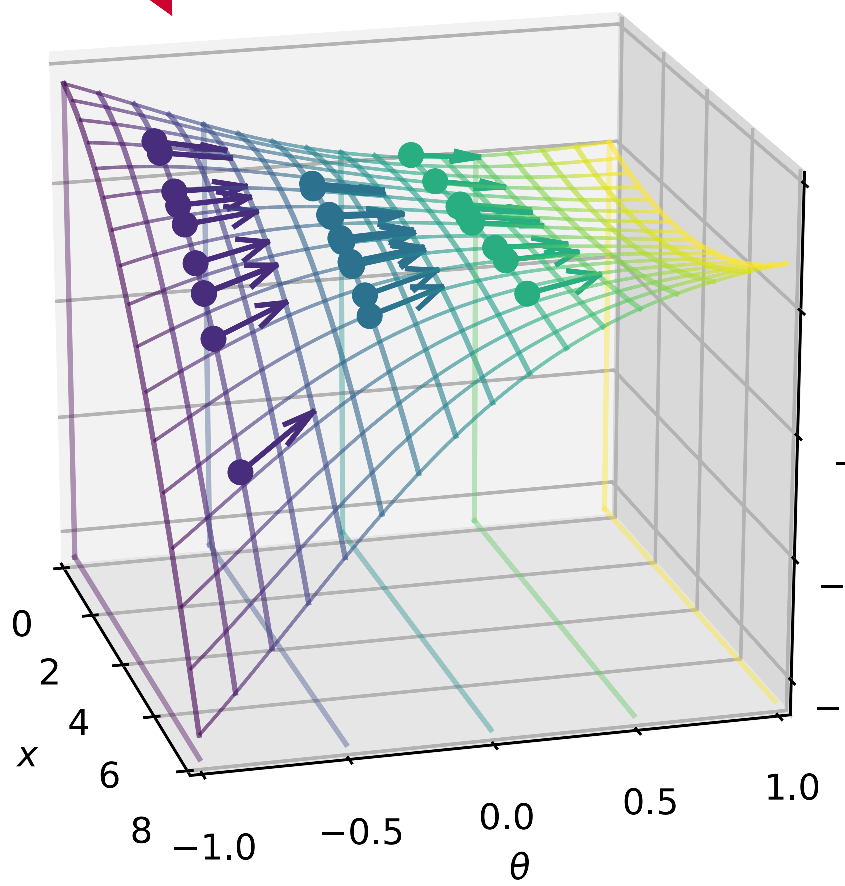
Surprisingly, when we want to use high-dimensional data and have to deal with the detector response, we do not have a good way to calculate the likelihood.



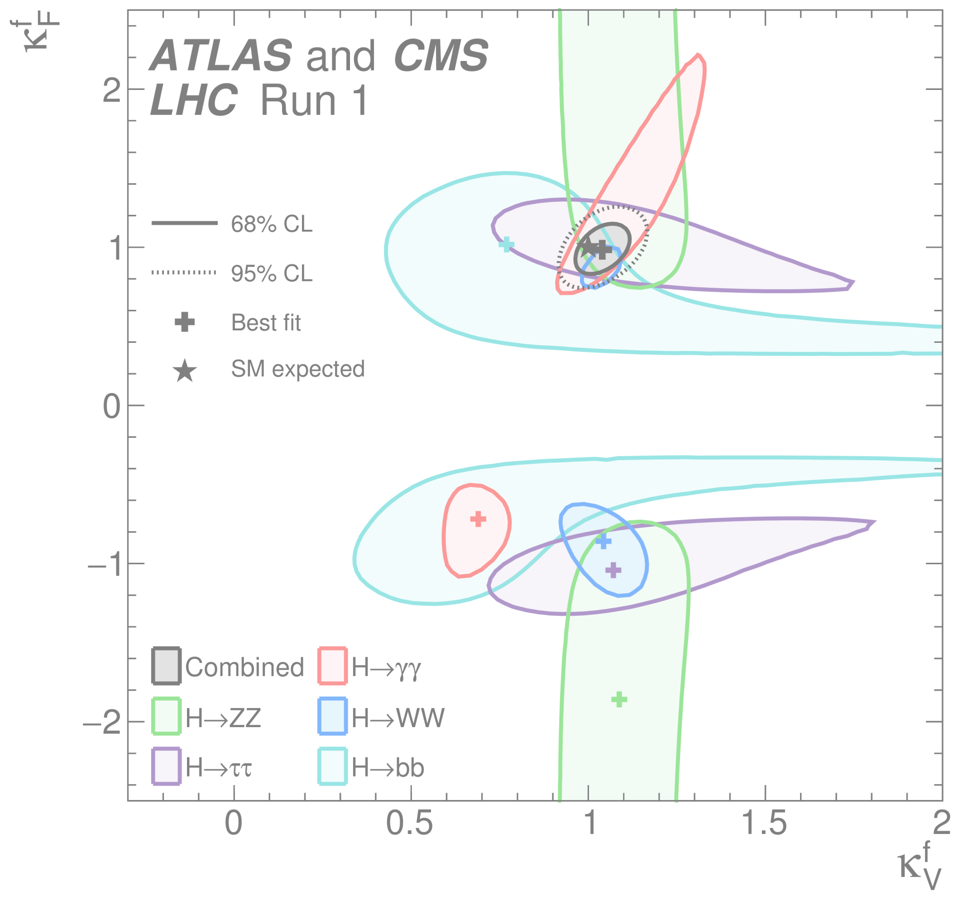
Machine learning



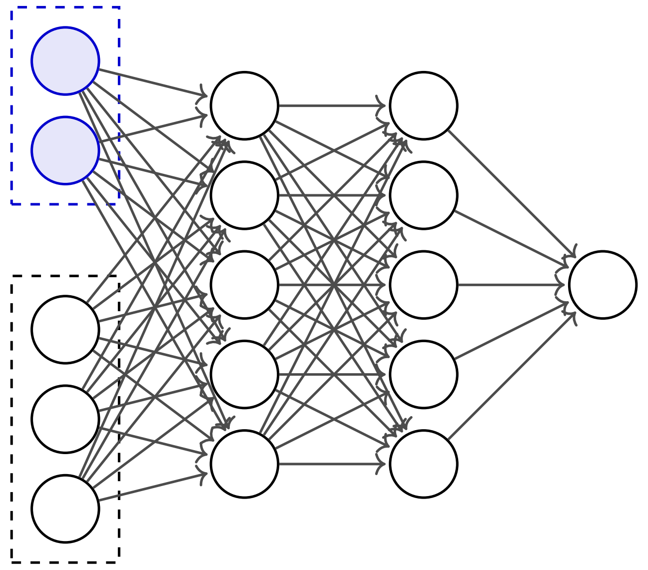
High-dimensional event data $x$



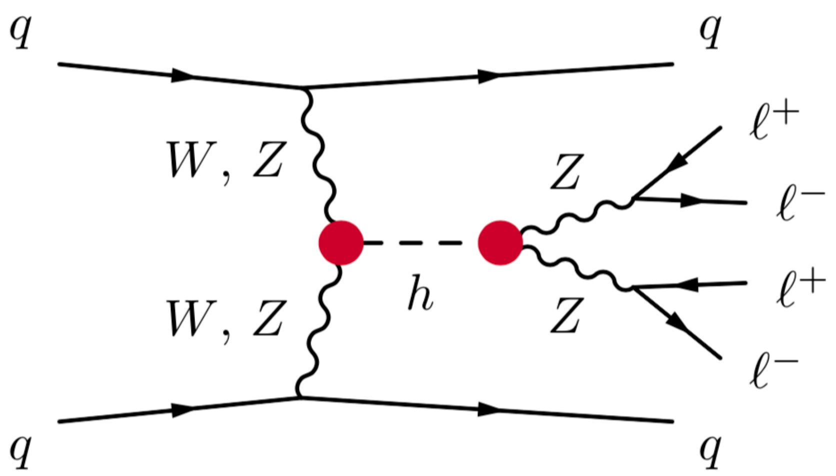

Estimator of the likelihood $p(x|\theta)$



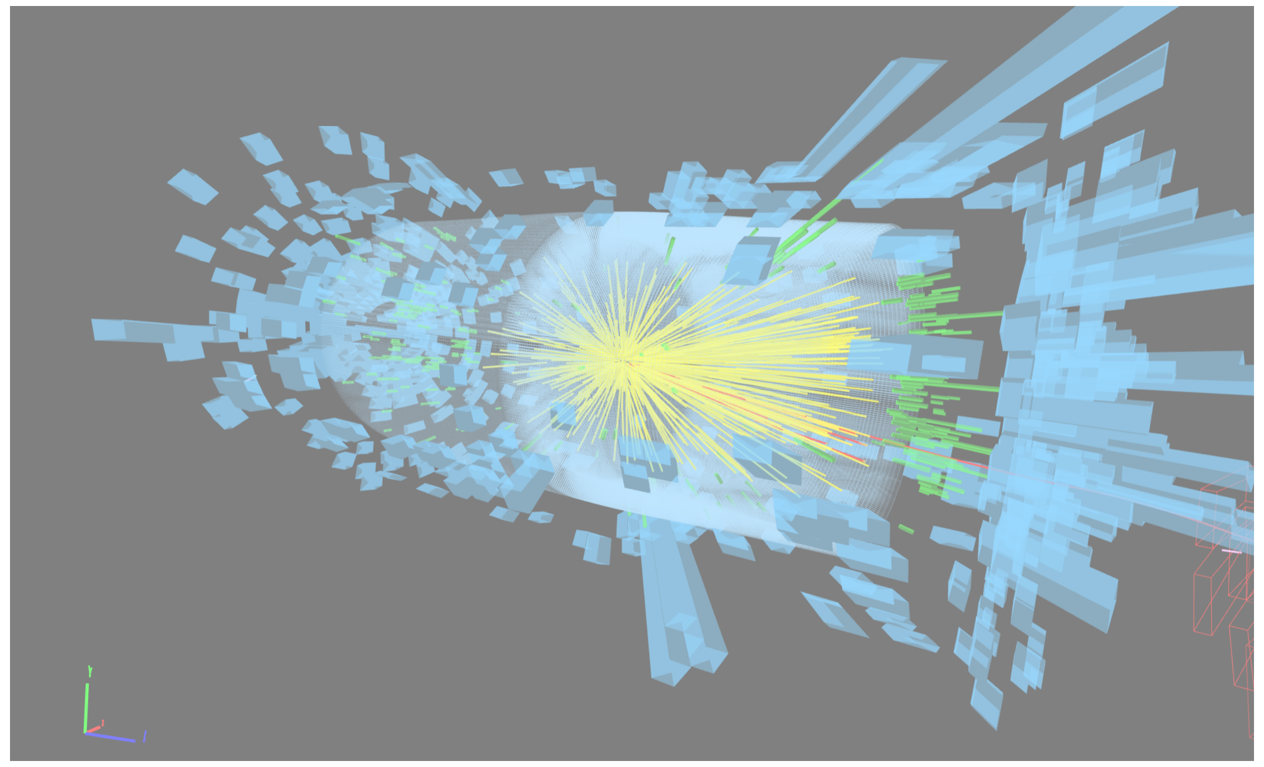

Constraints on parameters $\theta$



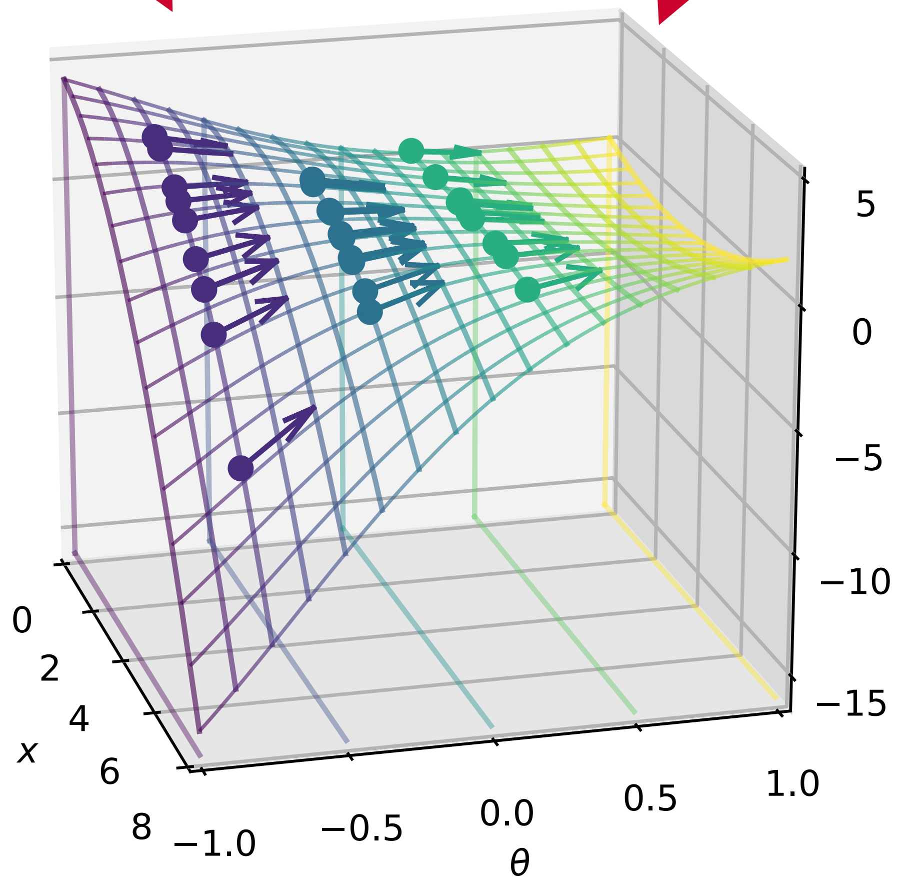
Machine learning



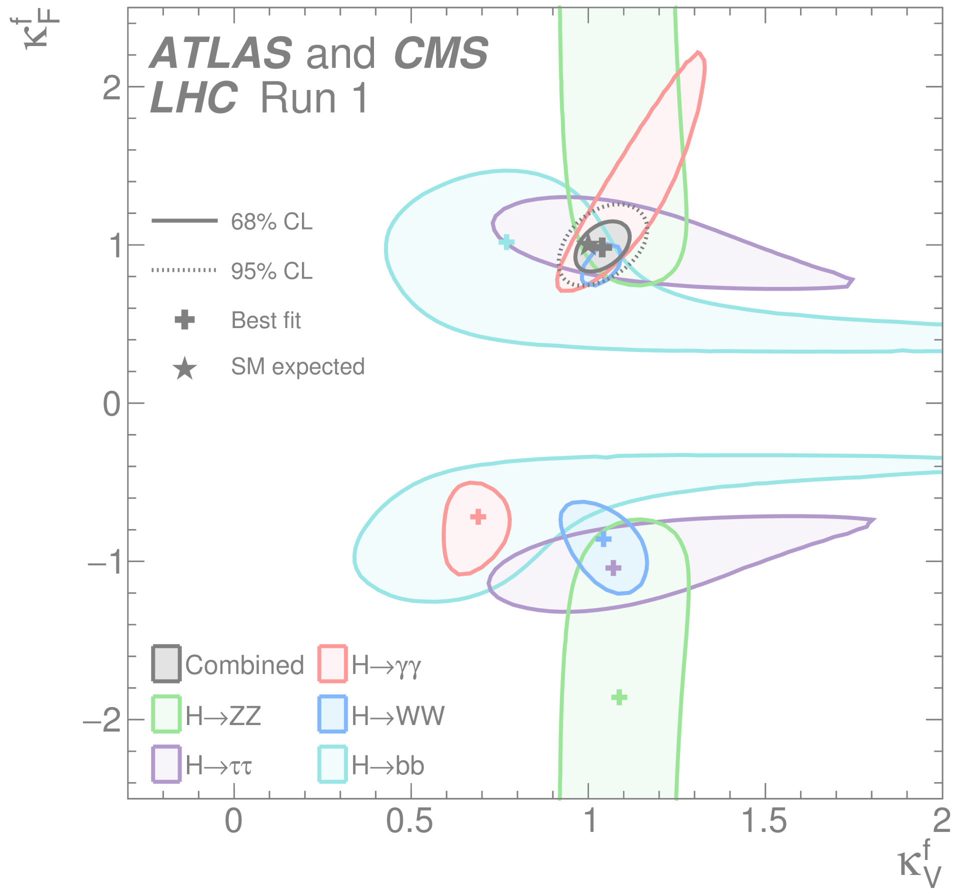

Physics insight:
matrix element information



High-dimensional
event data $x$

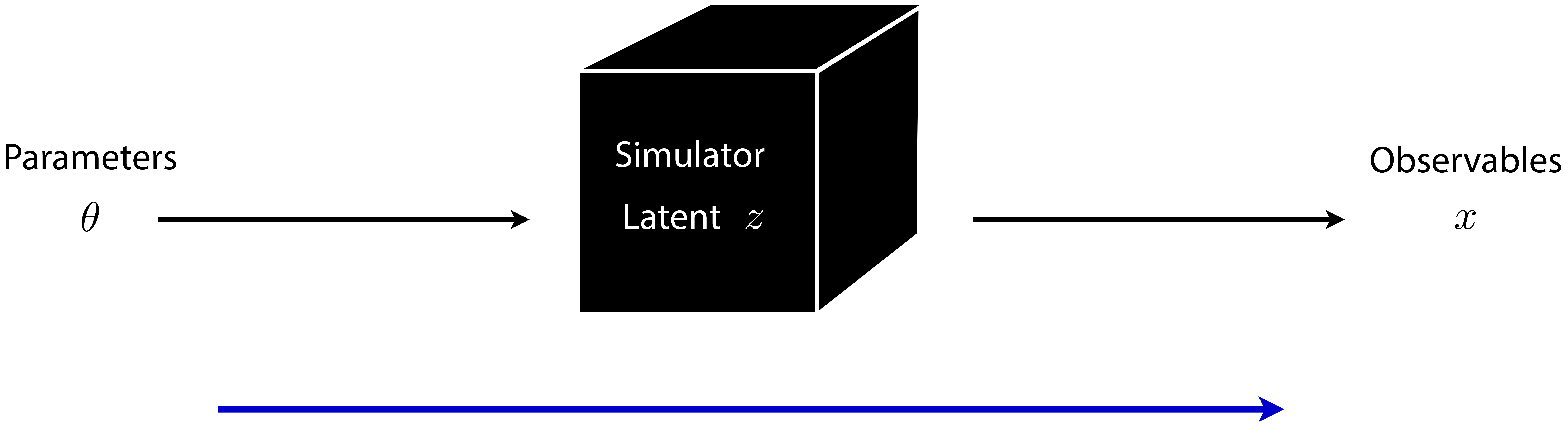



Estimator of the
likelihood $p(x|\theta)$




Constraints on
parameters $\theta$

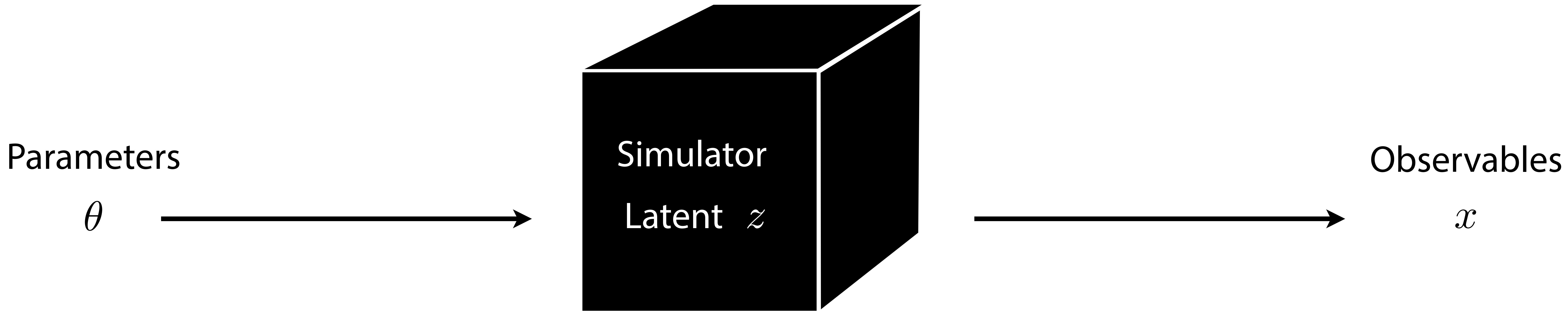
# Simulation-based ("likelihood-free") inference




Prediction:

- Well-understood mechanistic model
- Simulator can generate samples $x \sim p(x|\theta)$

# Simulation-based ("likelihood-free") inference

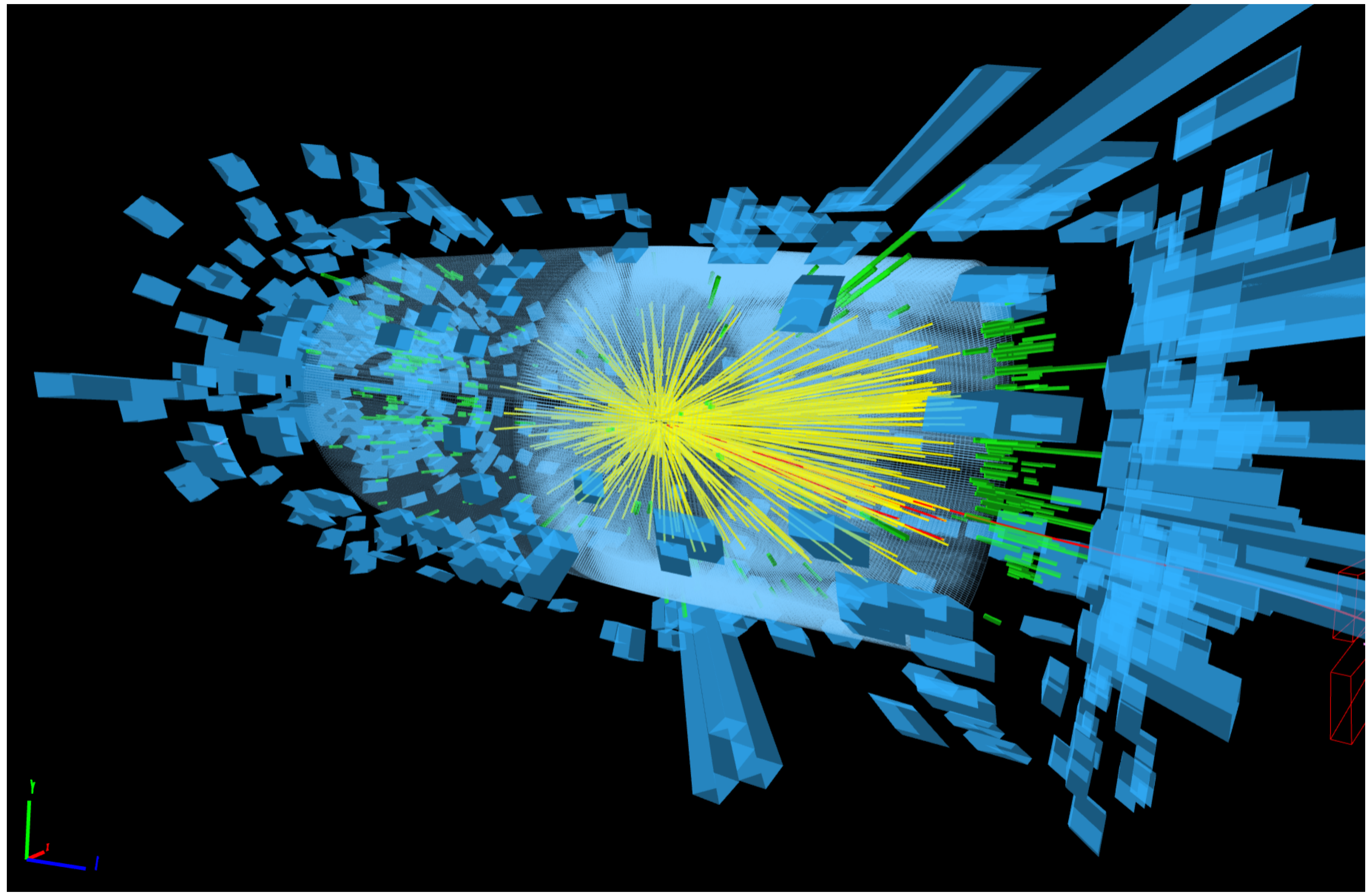



Prediction:
- Well-understood mechanistic model
- Simulator can generate samples $x \sim p(x|\theta)$

Inference:
- Likelihood $p(x|\theta) = \int \mathrm{d}z \; p(x, z|\theta)$ is intractable
- Inference is challenging

Traditionally, inference is made possible with summary statistics.

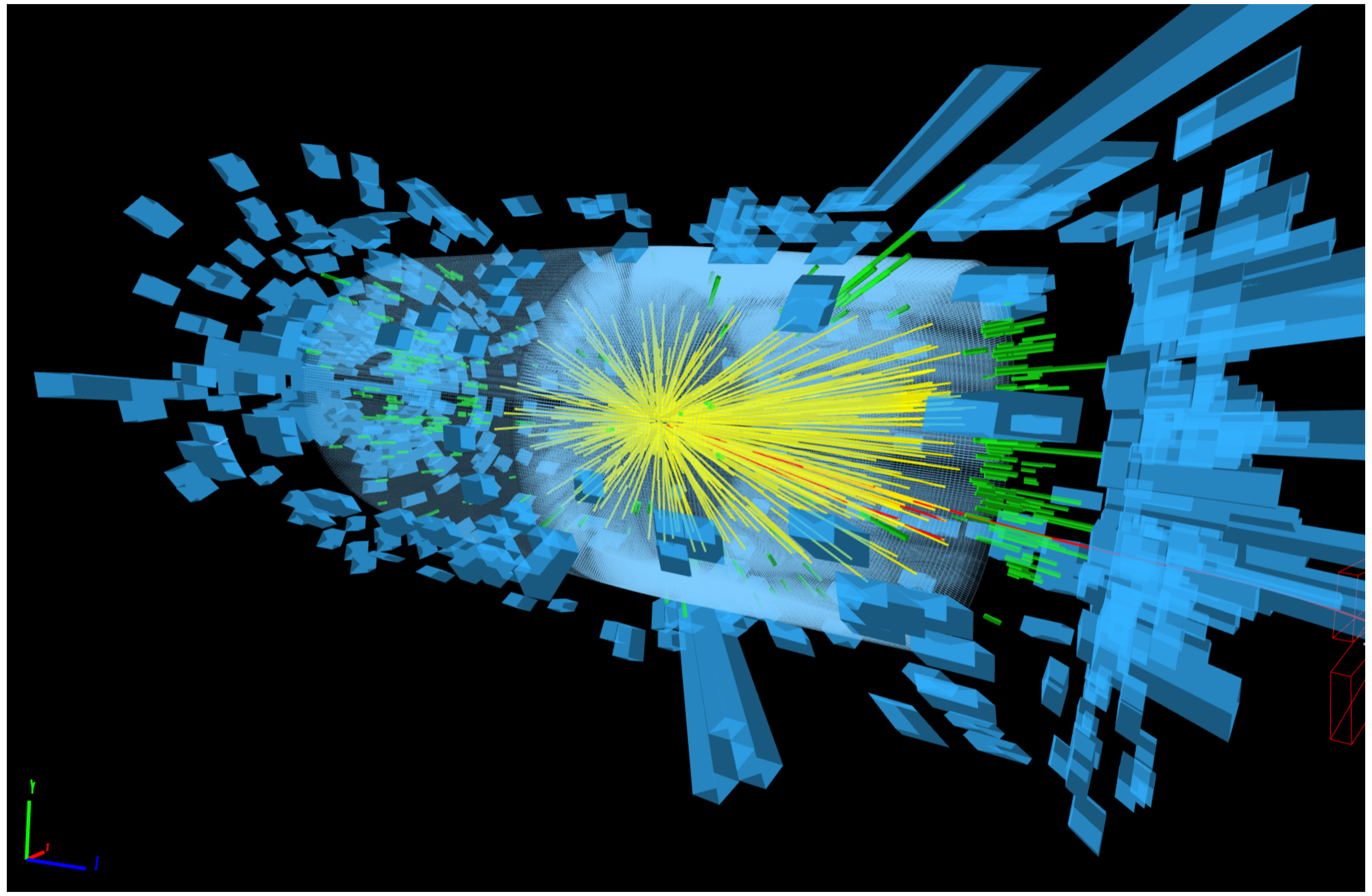
# Solve it with summary statistics



High-dimensional event data $x$

$p(x|\theta)$ cannot be calculated

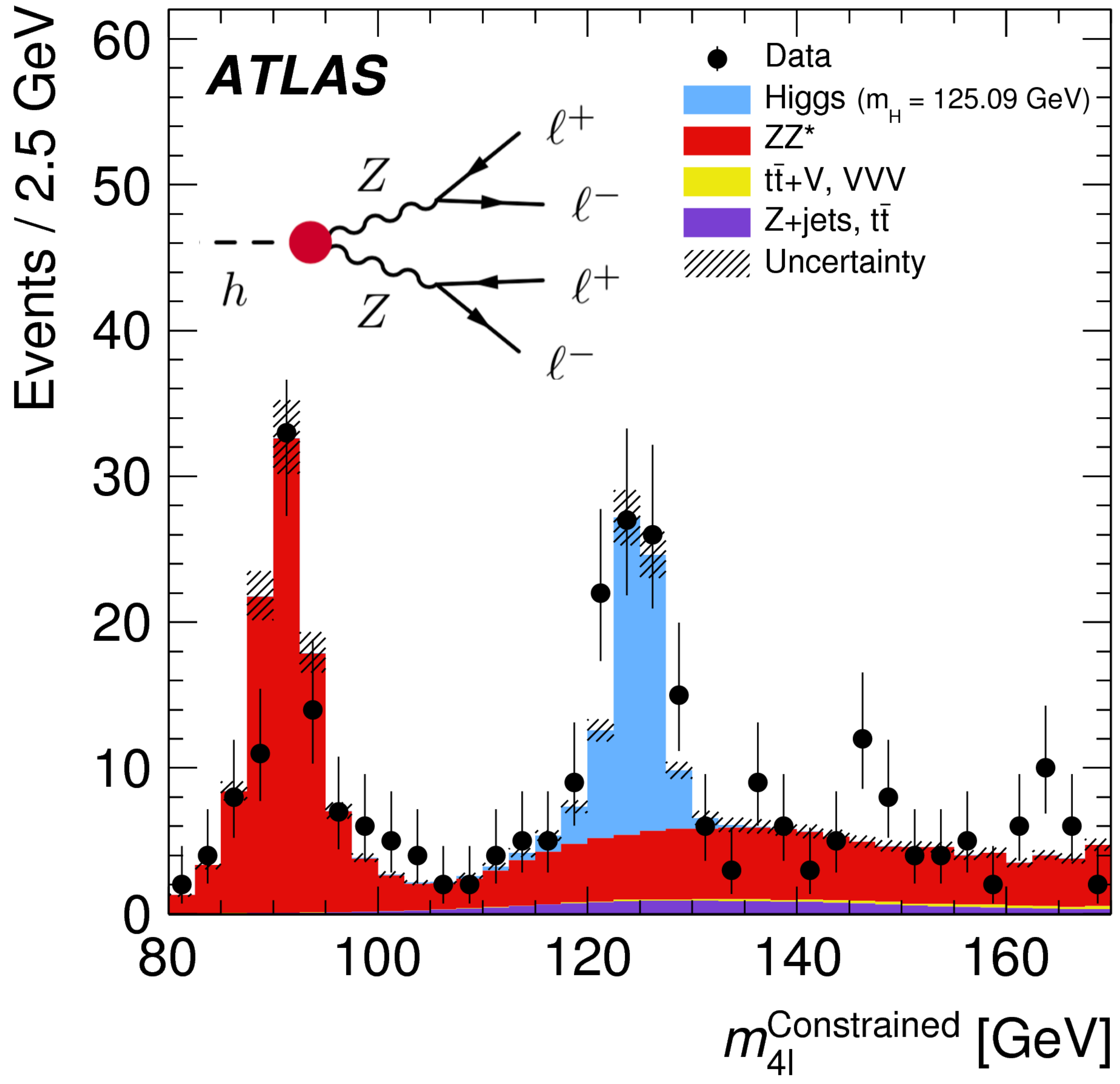
# Solve it with summary statistics



High-dimensional event data $x$

$p(x|\theta)$ cannot be calculated

→



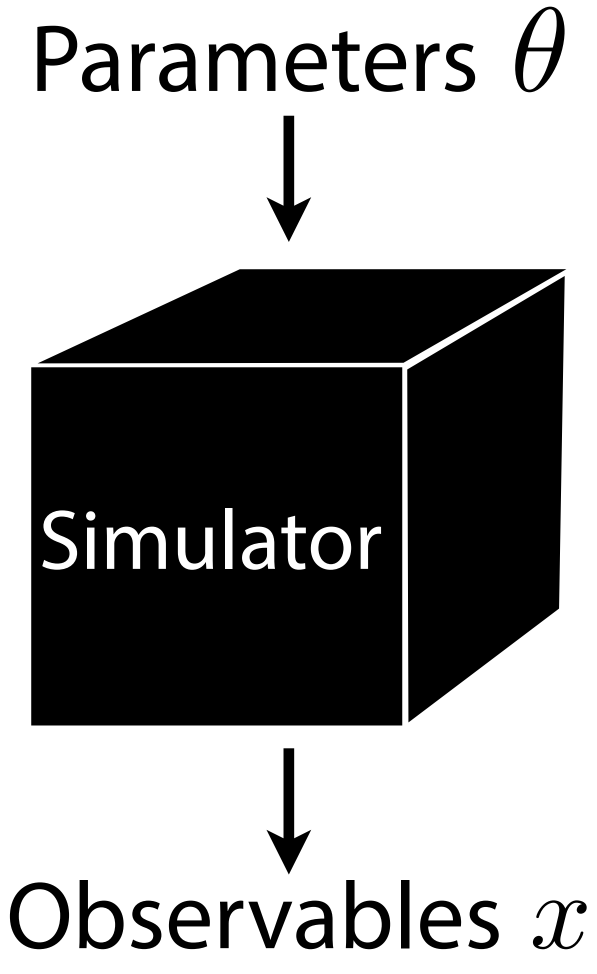

[ATLAS 1712.02304]

One or two summary statistics $x'$

$p(x'|\theta)$ can be estimated with histograms, KDE, …

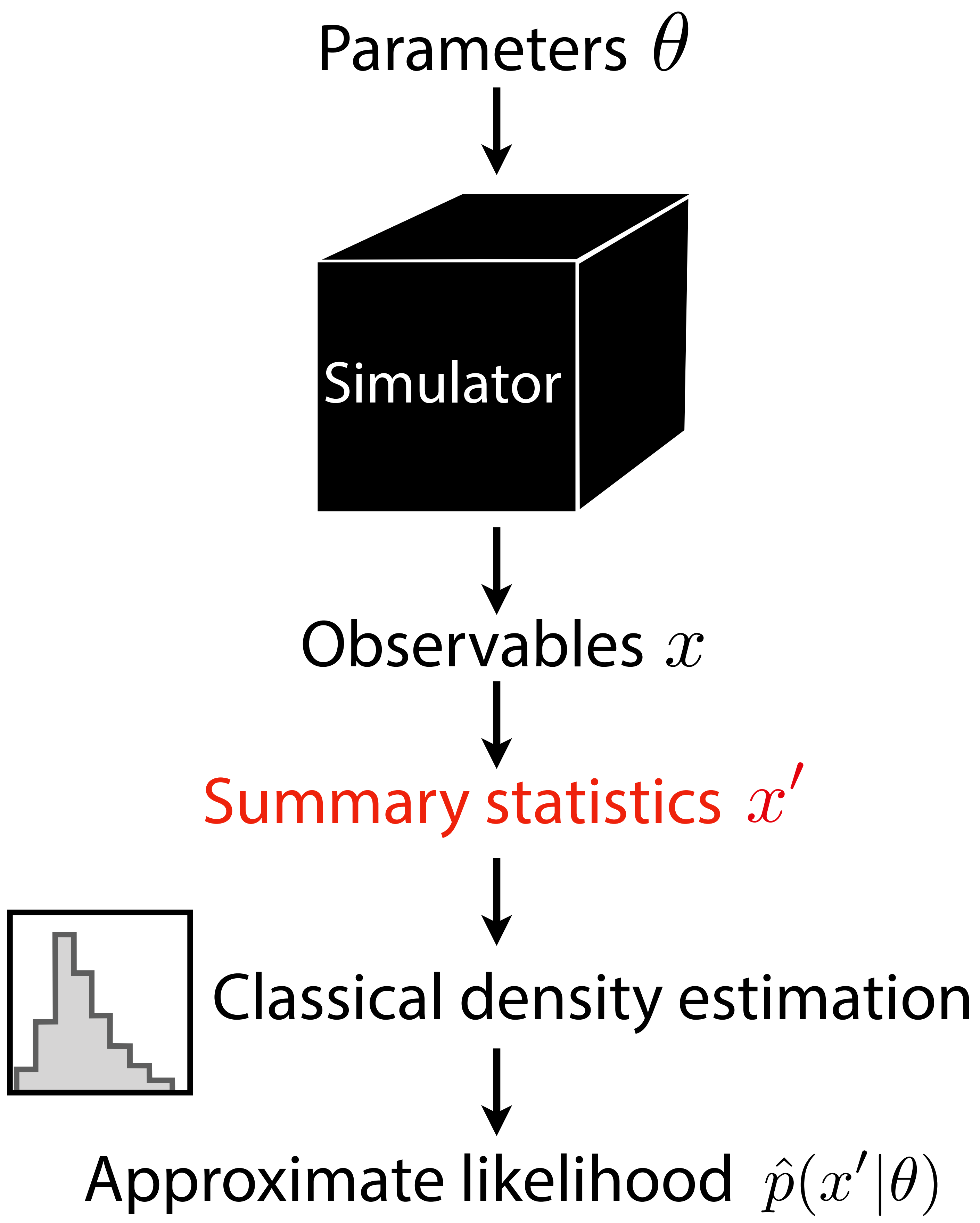
# Inference by estimating the likelihood

[e.g. P. Diggle, R. Gratton 1984]



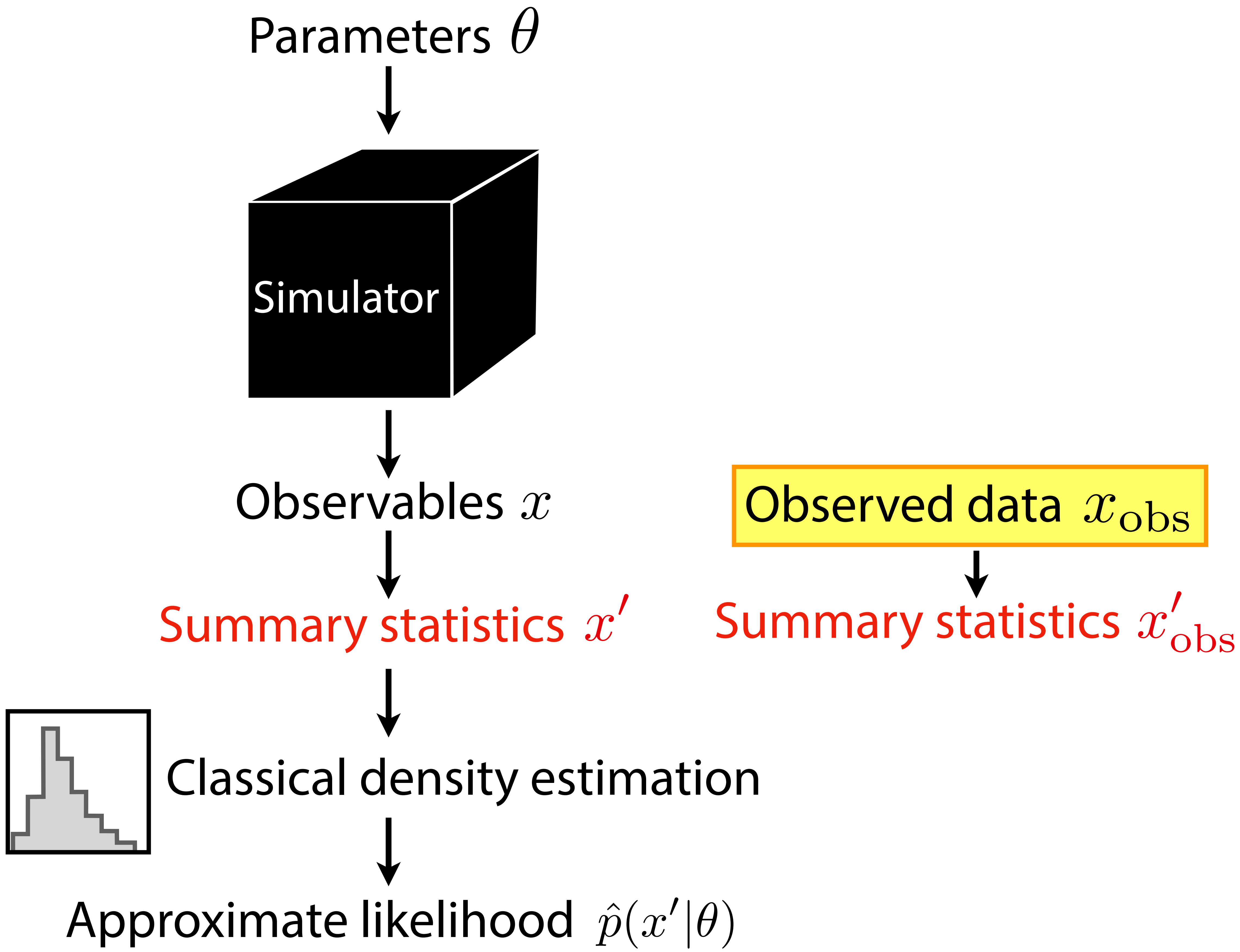
# Inference by estimating the likelihood

[e.g. P. Diggle, R. Gratton 1984]



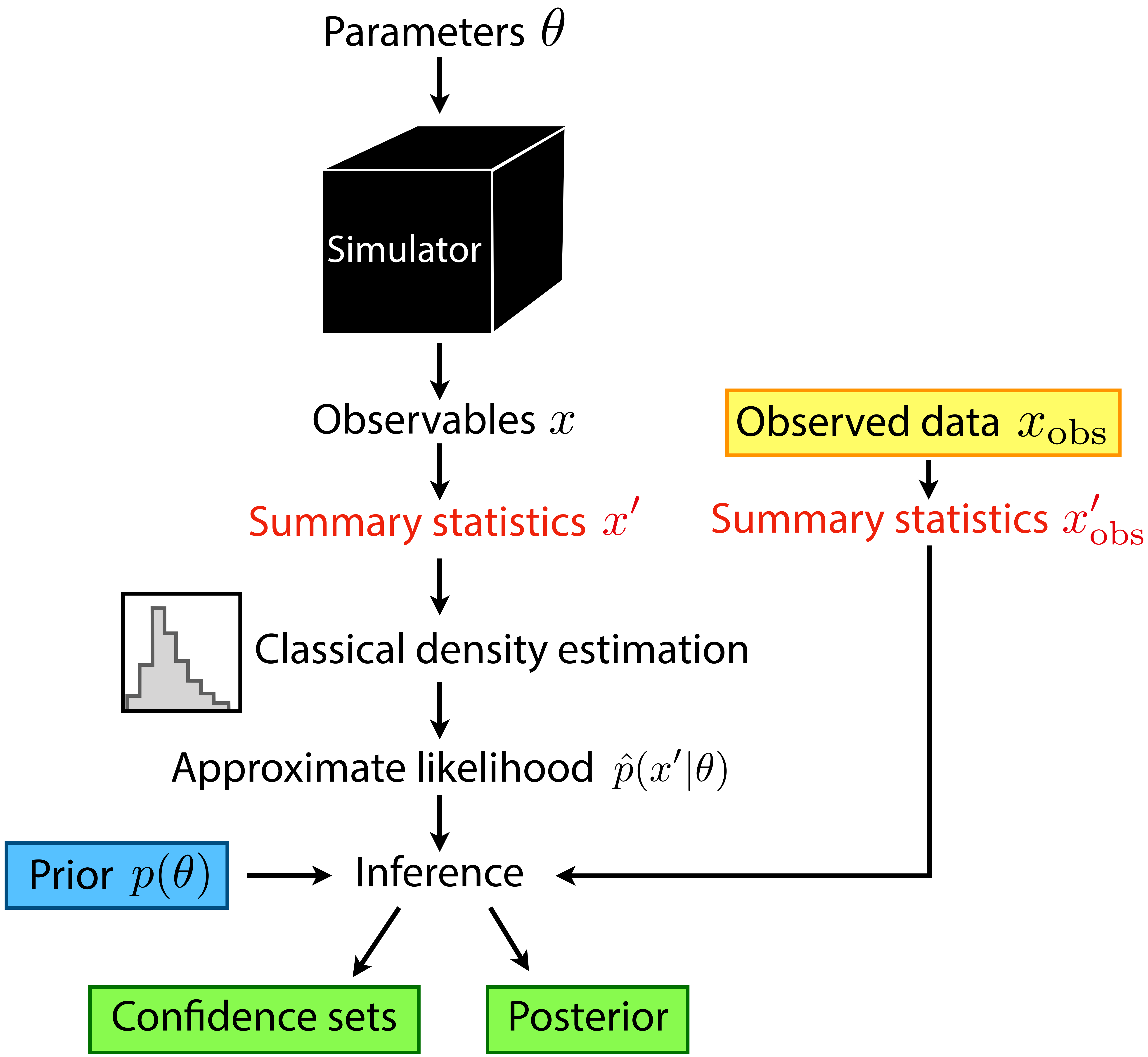
# Inference by estimating the likelihood

[e.g. P. Diggle, R. Gratton 1984]



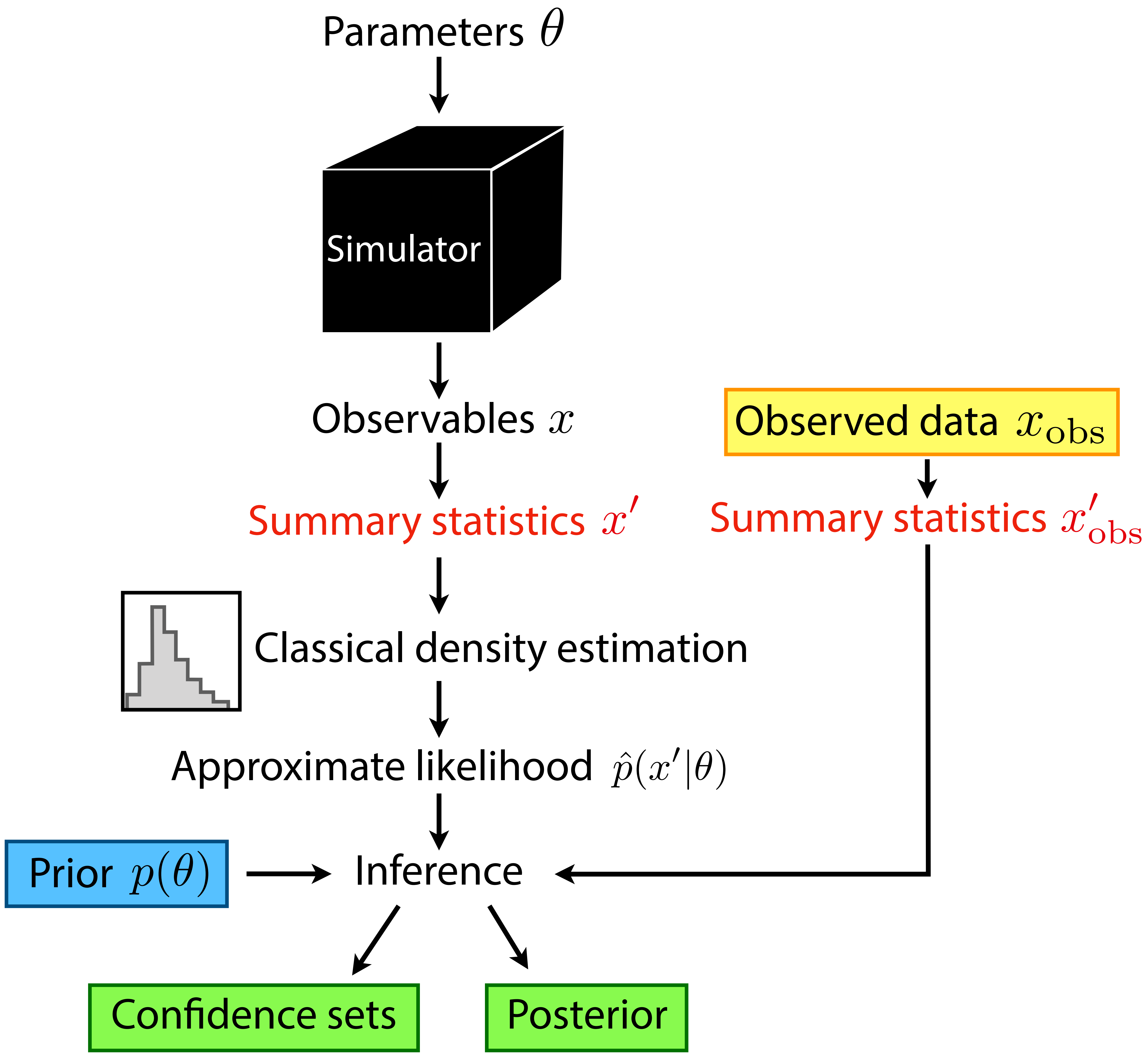
# Inference by estimating the likelihood

[e.g. P. Diggle, R. Gratton 1984]



# Inference by estimating the likelihood

[e.g. P. Diggle, R. Gratton 1984]




- Compression to summary statistics loses information & reduces quality of inference
- Curse of dimensionality: does not scale to more than a few summary statistics
- Related alternative: Approximate Bayesian Computation (ABC) [D. Rubin 1984]

# Summary statistics for LHC measurements?

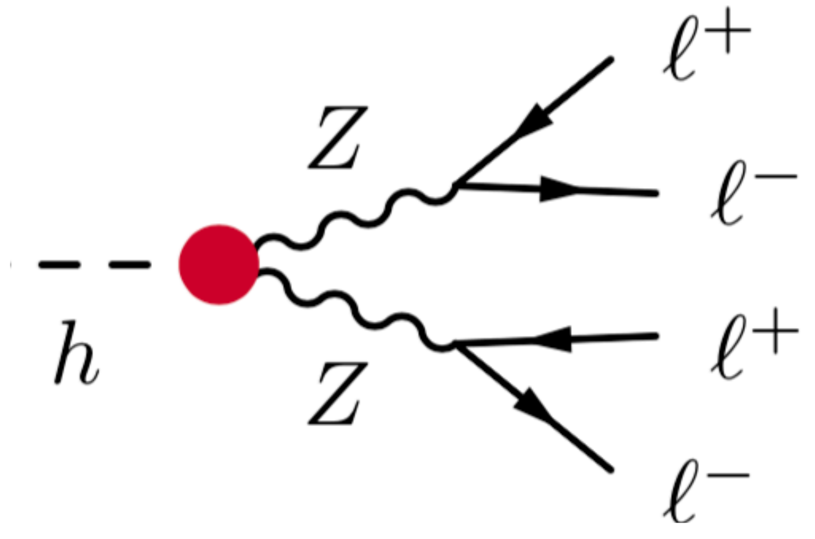
- In many LHC problems there is no single good summary statistic: compressing to any $x'$ loses information!
  [JB, K. Cranmer, F. Kling, T. Plehn 1612.05261; JB, F. Kling, T. Plehn, T. Tait 1712.02350]

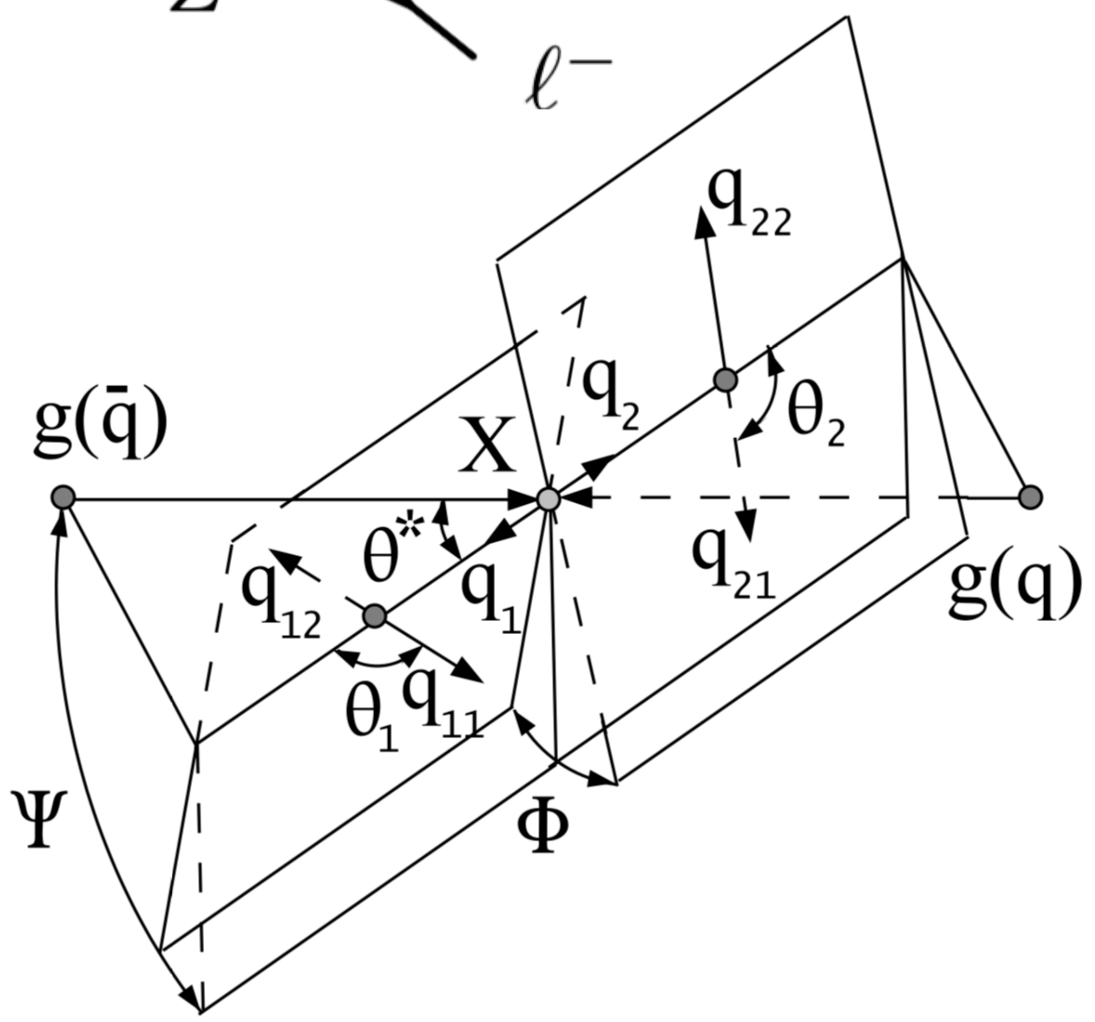
- Ideally: analyze all trustworthy high-level features (reconstructed four-momenta…), or some form of low-level features, including correlations
  (“fully differential cross section”)

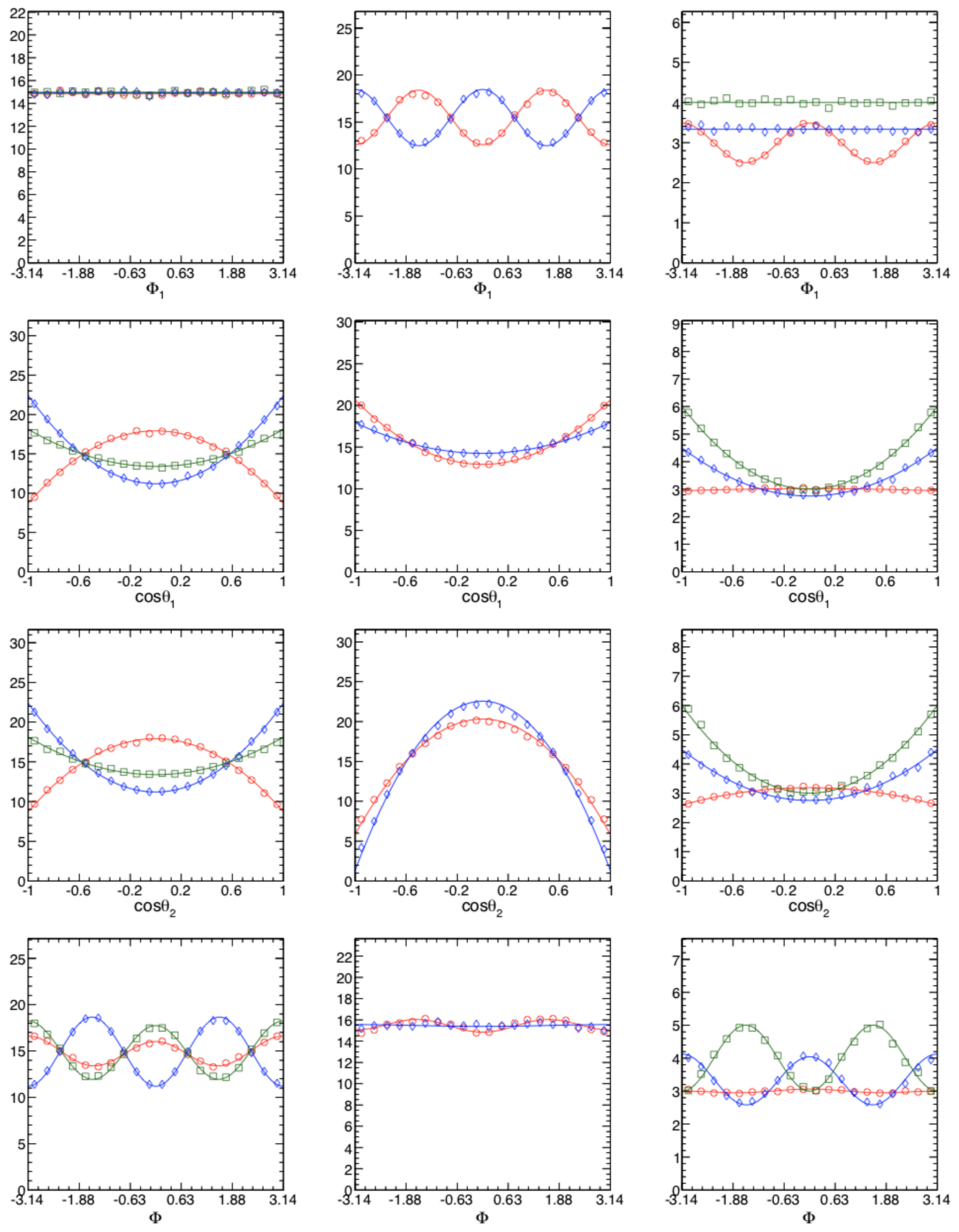
# Summary statistics for LHC measurements?

- In many LHC problems there is no single good summary statistic: compressing to any $x'$ loses information!
  [JB, K. Cranmer, F. Kling, T. Plehn 1612.05261; JB, F. Kling, T. Plehn, T. Tait 1712.02350]

- Ideally: analyze all trustworthy high-level features (reconstructed four-momenta…), or some form of low-level features, including correlations
  (“fully differential cross section”)



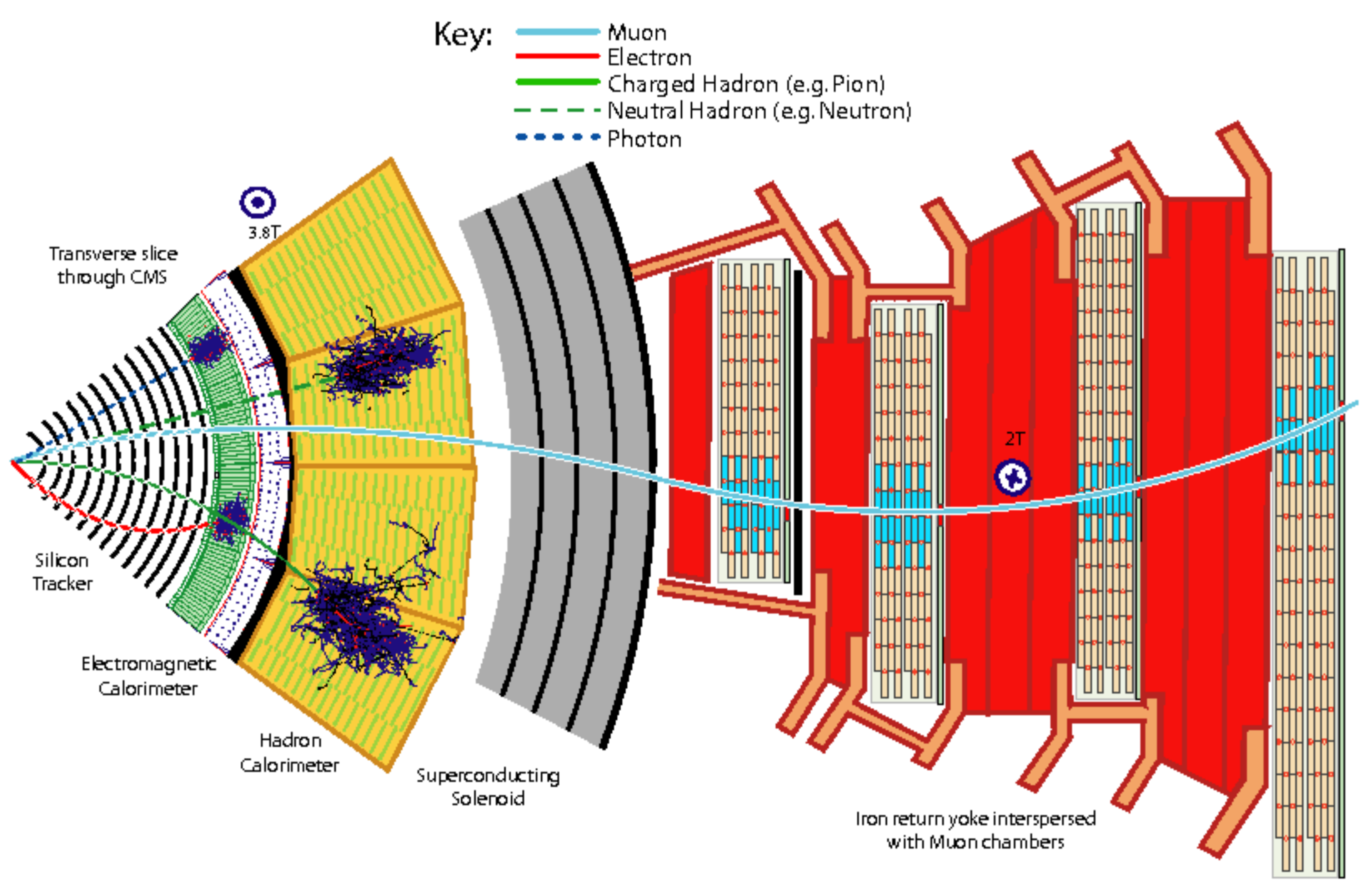




[Bolognesi et al. 1208.4018]

# Solve it by approximating the integral

- Problem: high-dimensional integral over shower / detector trajectories

$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \; p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p(z_p|\theta)$$

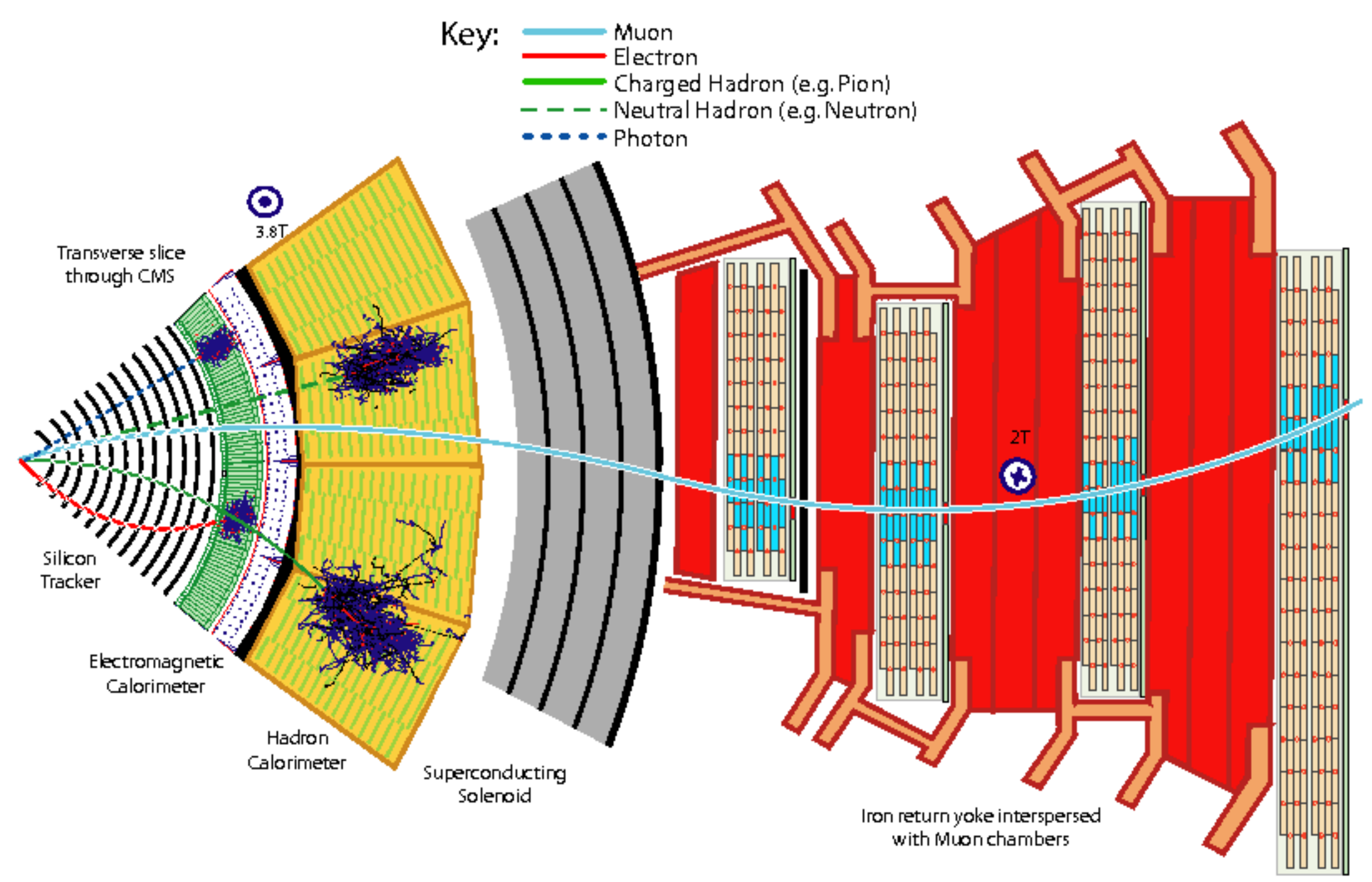



[CMS]

# Solve it by approximating the integral

- Problem: high-dimensional integral over shower / detector trajectories

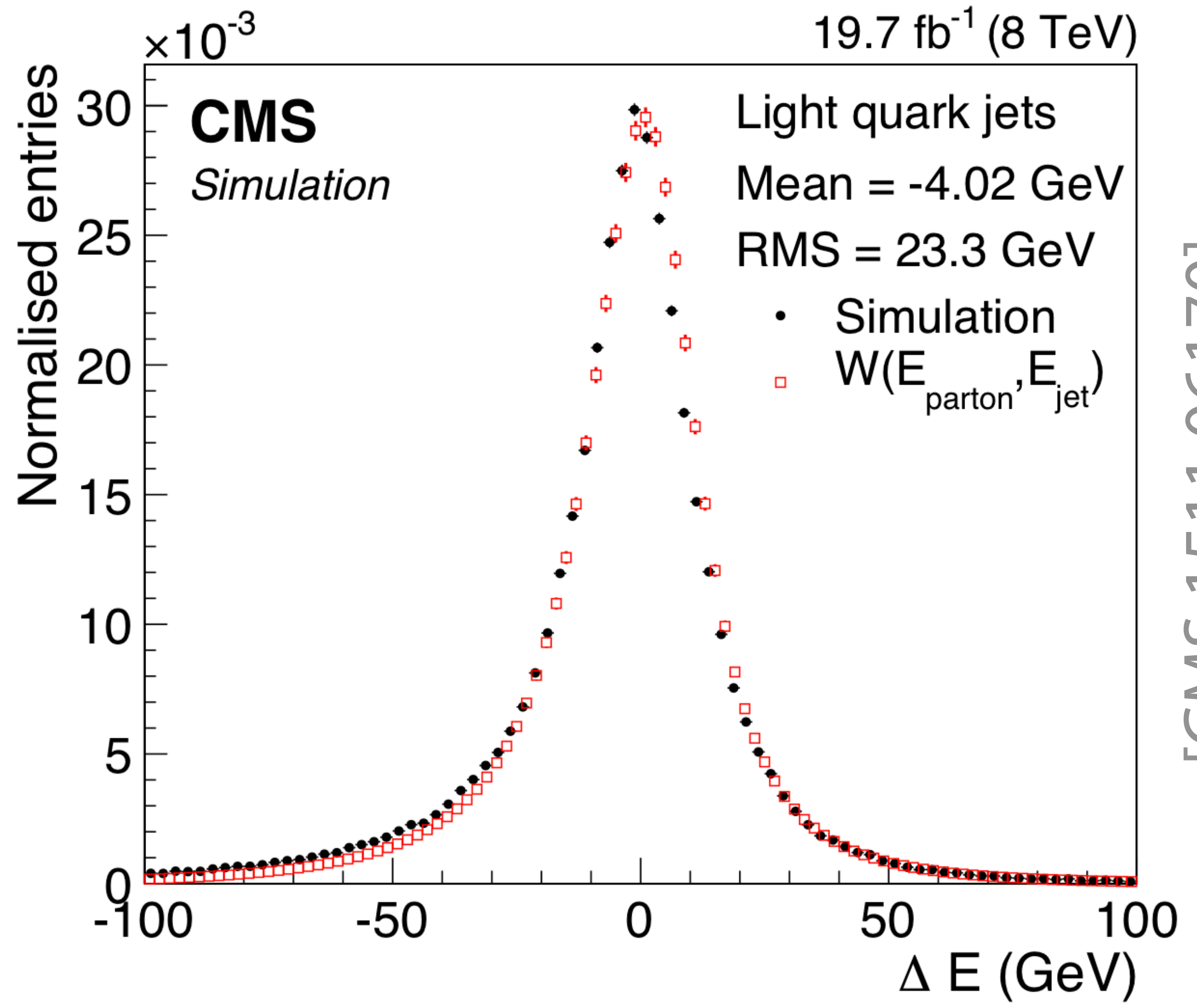
$$p(x|\theta) = \int dz_d \int dz_s \int dz_p \; p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p(z_p|\theta)$$




[CMS]

- Matrix Element Method (and similarly Optimal Observables): [K. Kondo 1988]
    - approximate shower + detector effects into transfer function $\hat{p}(x|z_p)$
    - explicitly calculate remaining integral

$$\hat{p}(x|\theta) = \int dz_p \; \hat{p}(x|z_p)\, p(z_p|\theta)$$

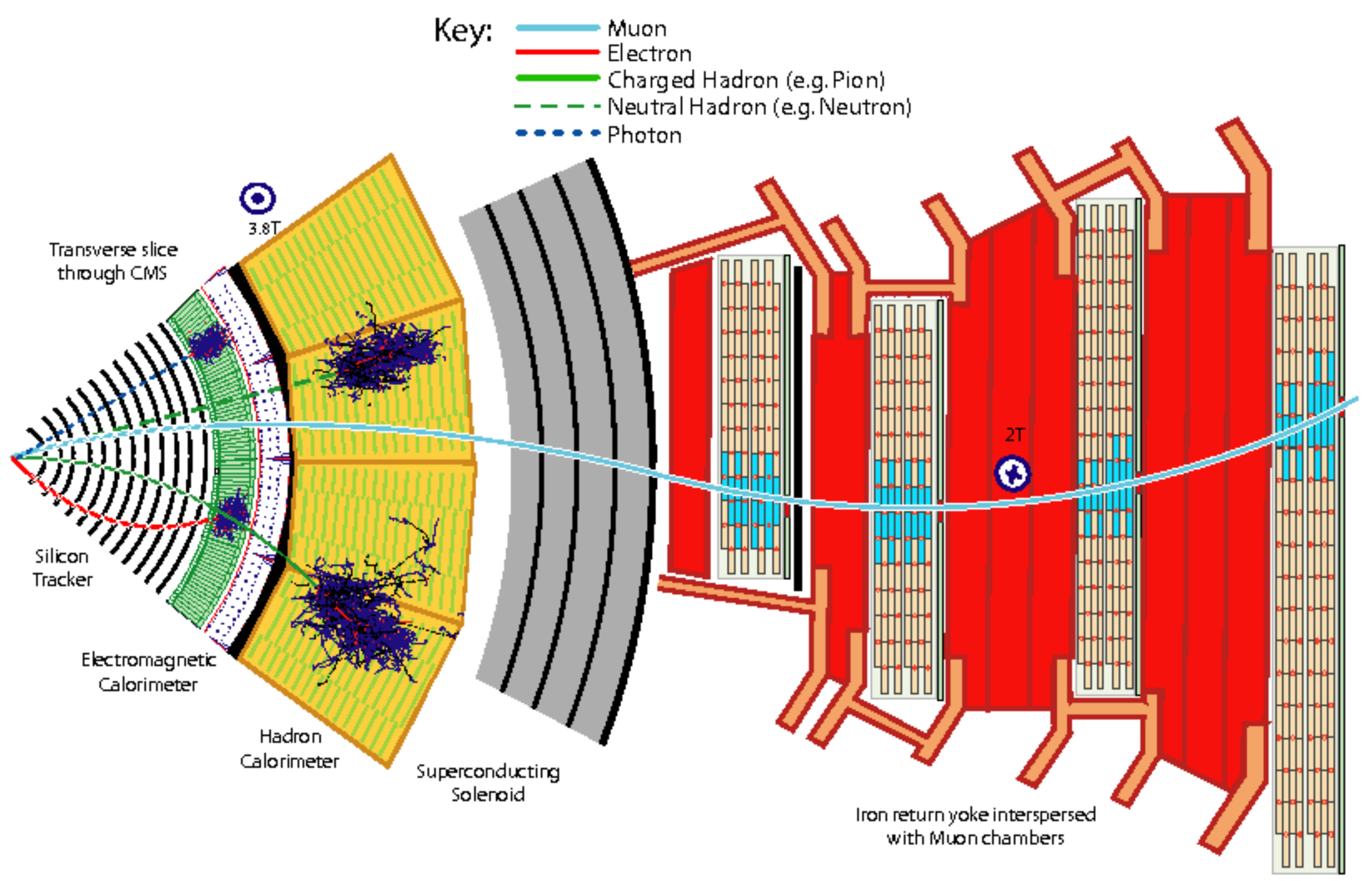



[CMS 1511.06170]

# Solve it by approximating the integral

- Problem: high-dimensional integral over shower / detector trajectories

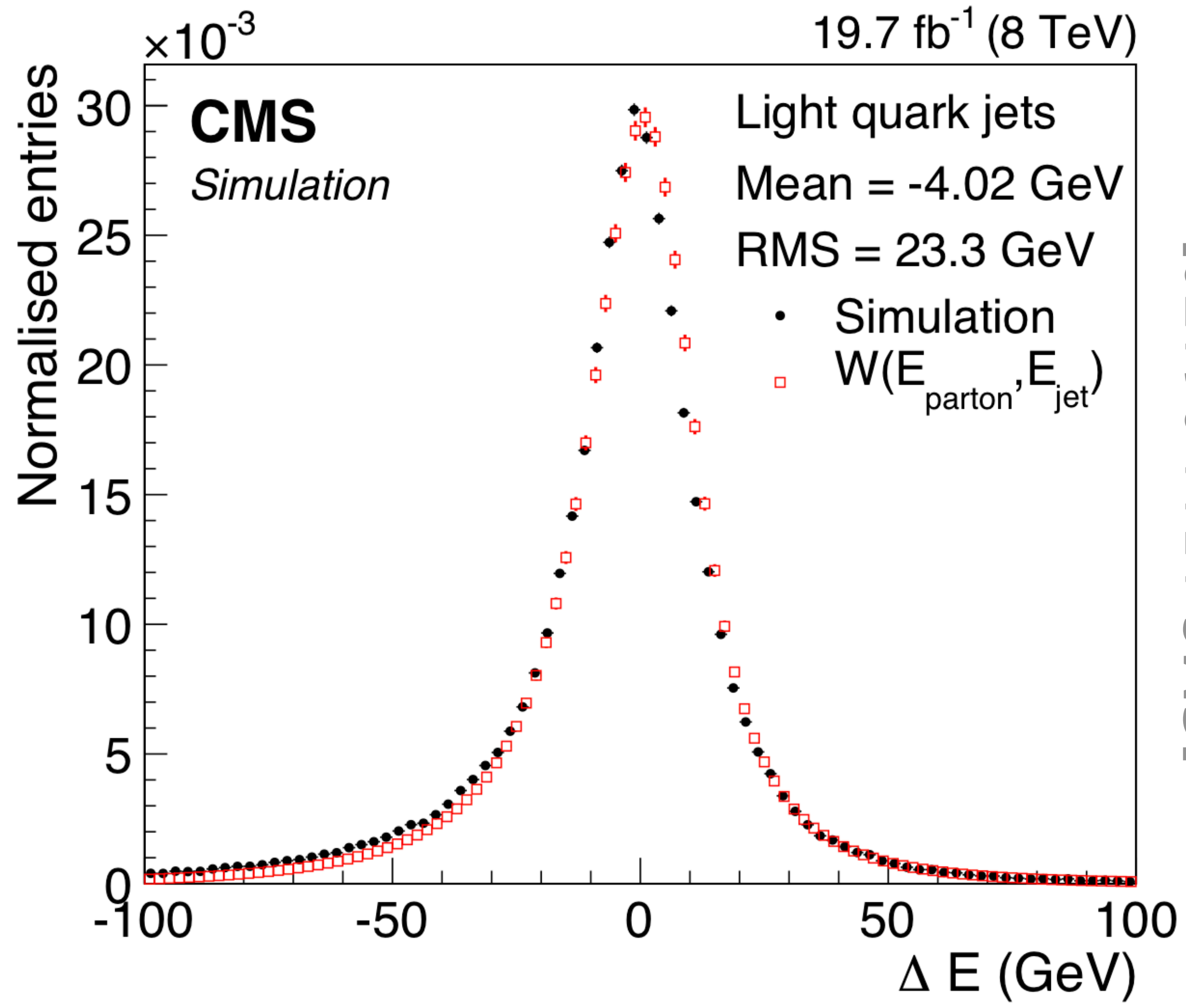
$$p(x|\theta) = \int dz_d \int dz_s \int dz_p \; p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p(z_p|\theta)$$




[CMS]

- Matrix Element Method (and similarly Optimal Observables): [K. Kondo 1988]
  - approximate shower + detector effects into transfer function $\hat{p}(x|z_p)$
  - explicitly calculate remaining integral

$$\hat{p}(x|\theta) = \int dz_p \; \hat{p}(x|z_p)\, p(z_p|\theta)$$




[CMS 1511.06170]

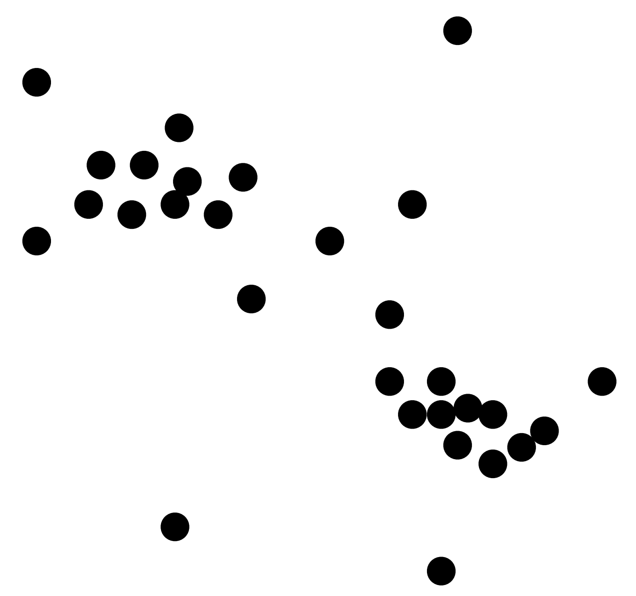
⇒ Uses matrix-element information, no summary statistics necessary, but:

- ad-hoc transfer functions (what about extra radiation?)
- evaluation still requires calculating an expensive integral

But we can use machine learning to make inference possible.

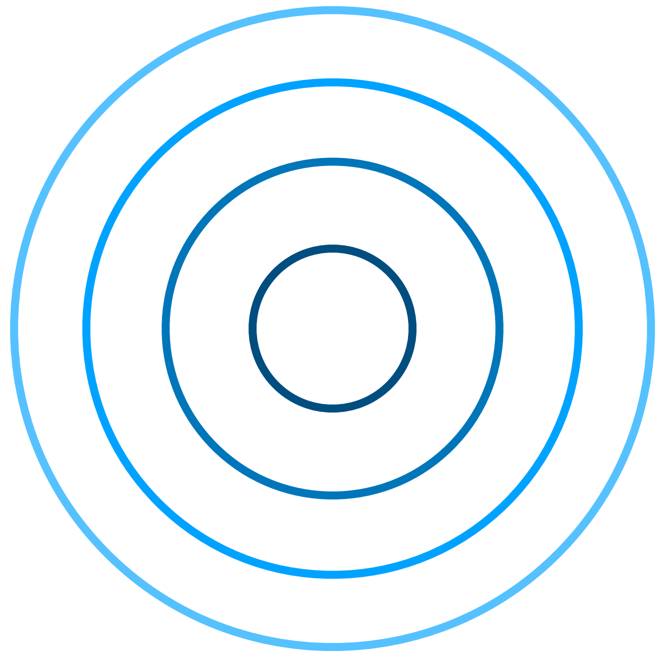
[K. Cranmer, JB, G. Louppe 1911.01429]

# High-dimensional density estimation with normalizing flows



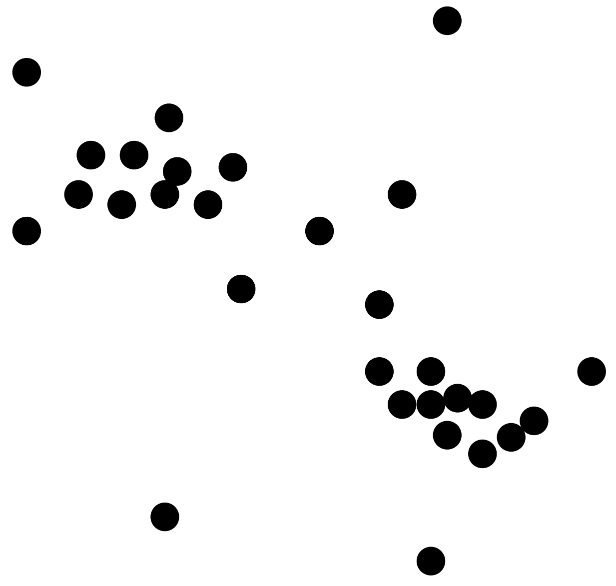
[K.C., Gilles Louppe https://doi.org/10.5281/zenodo.198541
lots of great work recently; see G. Papamakarios et al 1912.02762 for a review]

# High-dimensional density estimation with normalizing flows



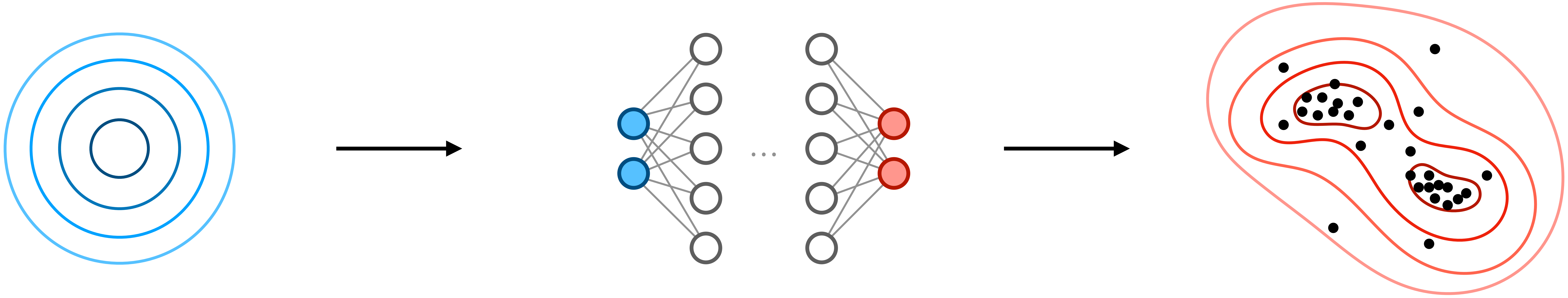
Simple base density

$$u \sim \pi(u)$$



[K.C., Gilles Louppe https://doi.org/10.5281/zenodo.198541
lots of great work recently; see G. Papamakarios et al 1912.02762 for a review]

# High-dimensional density estimation with normalizing flows




Simple base density
$u \sim \pi(u)$

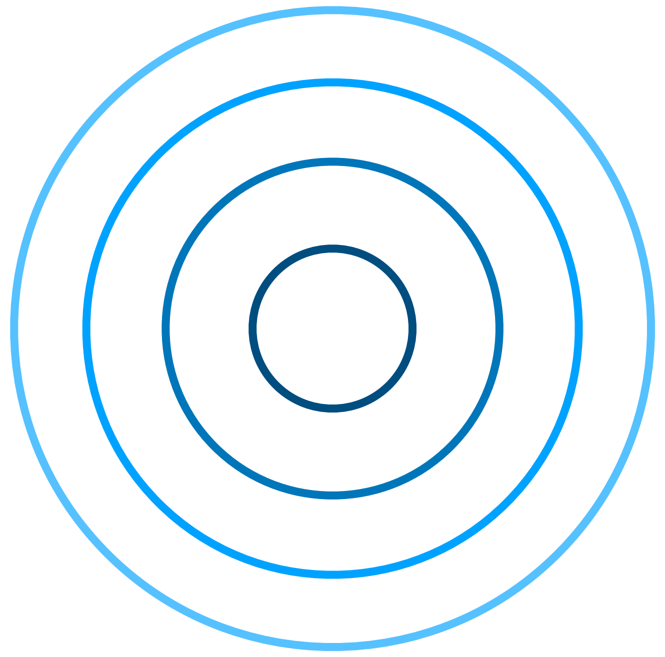
NN: transformation $x = f(u)$

- one-to-one and invertible
- differentiable
- $f^{-1}$ and $\det \nabla f$ are tractable

Target density is given by
$\hat{p}(x) = \pi(f^{-1}(x)) \, |\det \nabla f|^{-1}$

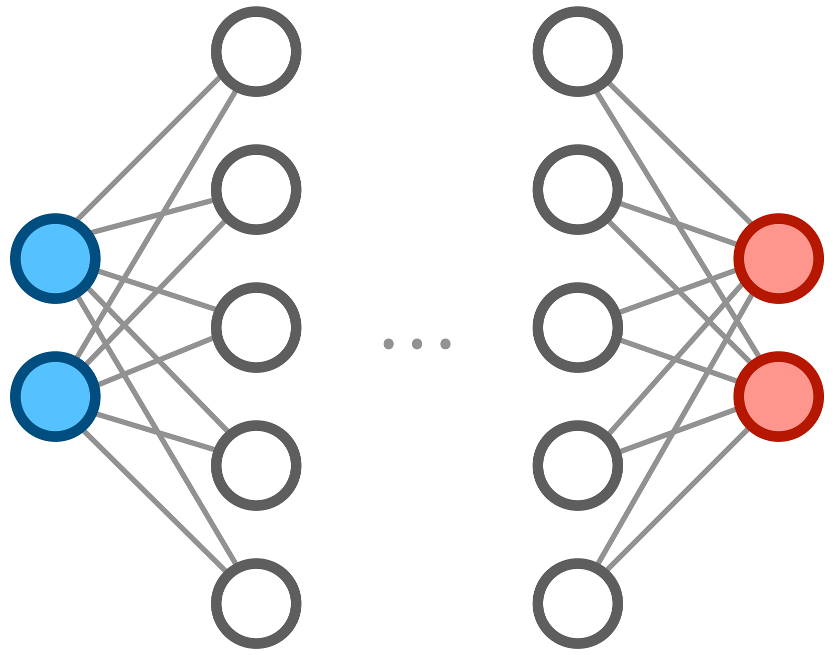
[K.C., Gilles Louppe https://doi.org/10.5281/zenodo.198541
lots of great work recently; see G. Papamakarios et al 1912.02762 for a review]

# High-dimensional density estimation with normalizing flows



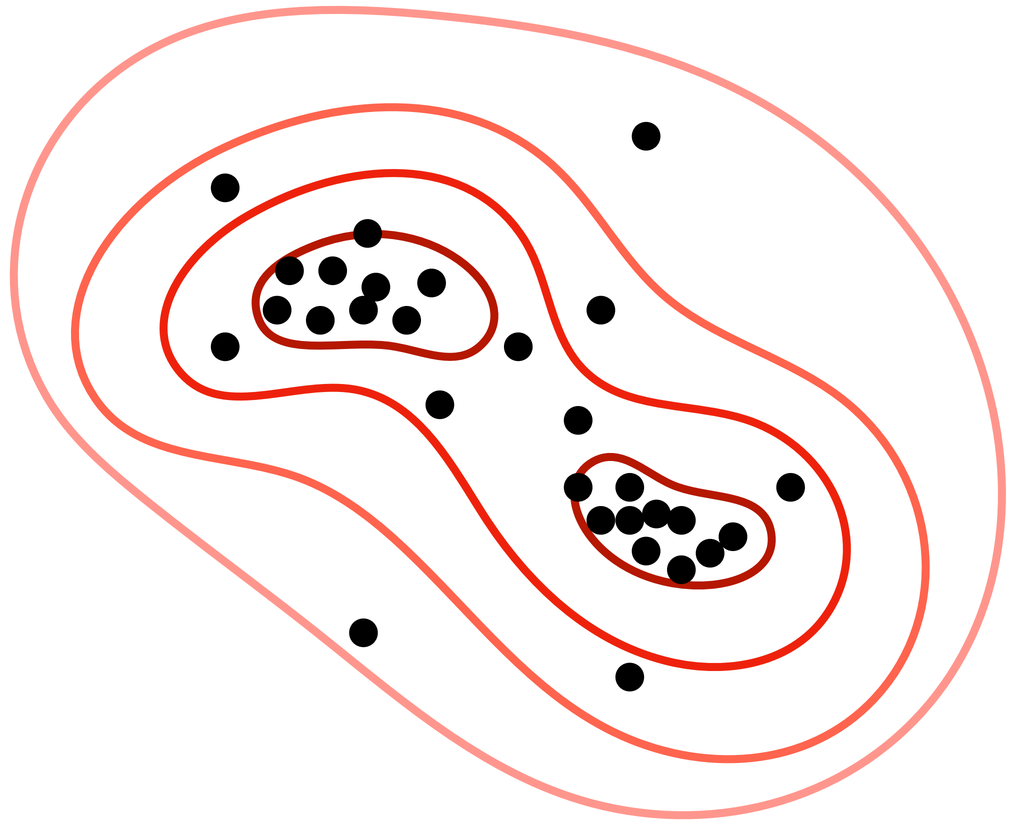
Simple base density

$u \sim \pi(u)$



NN: transformation $x = f(u)$

- one-to-one and invertible
- differentiable
- $f^{-1}$ and $\det \nabla f$ are tractable



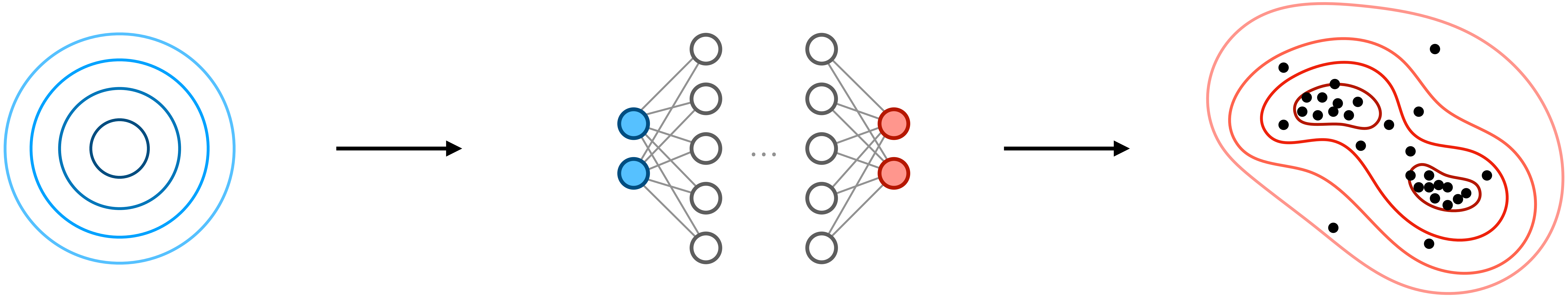
Target density is given by

$\hat{p}(x) = \pi(f^{-1}(x)) \, |\det \nabla f|^{-1}$

Train transformation by maximizing $\log \hat{p}(x)$

[K.C., Gilles Louppe https://doi.org/10.5281/zenodo.198541
lots of great work recently; see G. Papamakarios et al 1912.02762 for a review]

# High-dimensional density estimation with normalizing flows




Simple base density

$u \sim \pi(u)$

NN: transformation $x = f(u)$

- one-to-one and invertible
- differentiable
- $f^{-1}$ and $\det \nabla f$ are tractable

Target density is given by

$\hat{p}(x) = \pi(f^{-1}(x))\,|\det \nabla f|^{-1}$

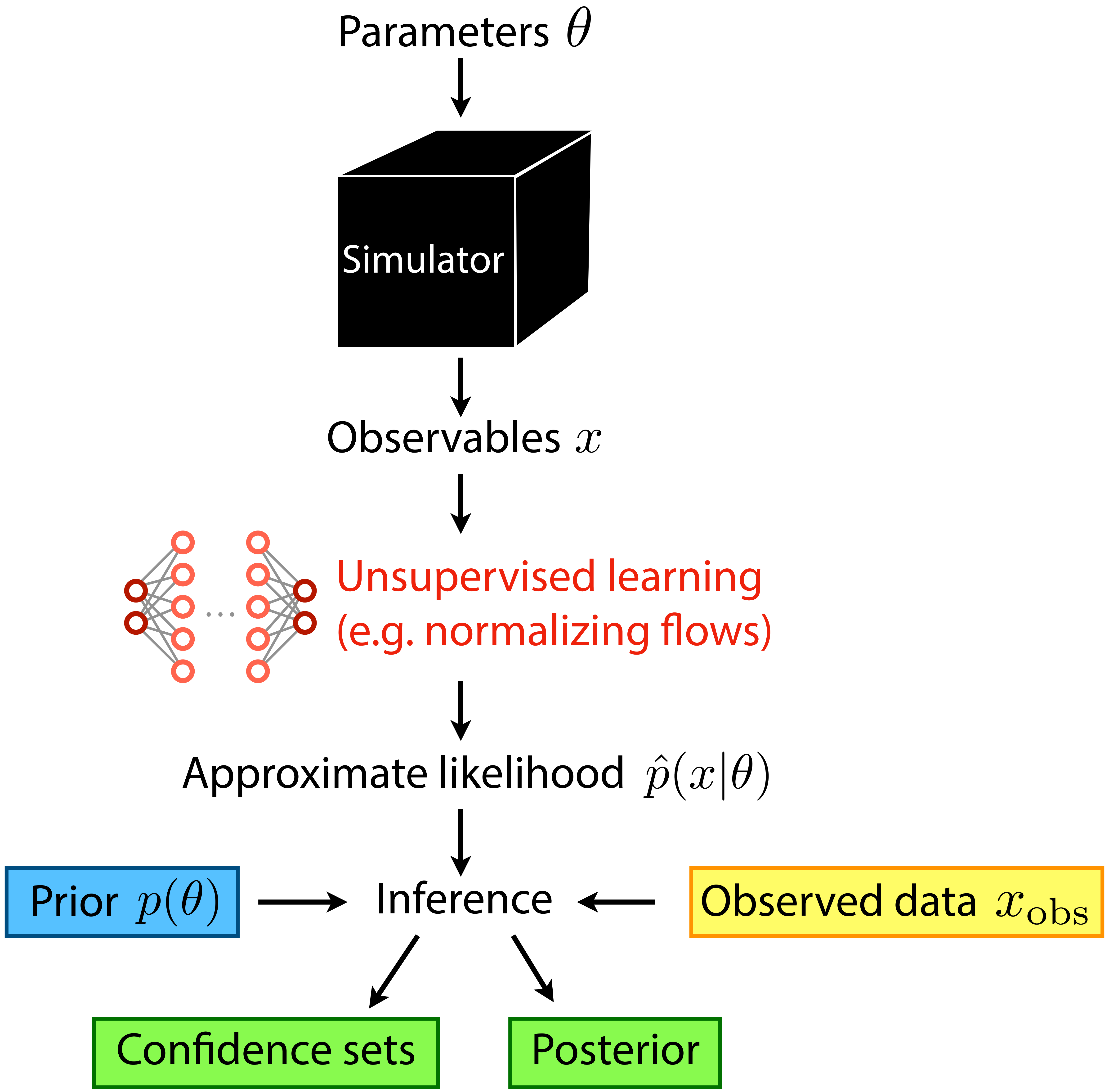
Train transformation by maximizing $\log \hat{p}(x)$

Transformation can depend on $\theta$ to model conditional density $\log \hat{p}(x|\theta)$

[K.C., Gilles Louppe https://doi.org/10.5281/zenodo.198541
lots of great work recently; see G. Papamakarios et al 1912.02762 for a review]

# Inference with neural likelihood estimation

[K.C., Gilles Louppe https://doi.org/10.5281/zenodo.198541 ;
G. Papamakarios, D. Sterrart, I. Murray 1805.07226;
J.-M. Lueckmann, G. Bassetto, T. Karaletsos, J. Macke 1805.09294]




- Conditional neural density estimator (e.g. normalizing flow) as tractable surrogate for simulator likelihood
- Scales well to high-dimensional data (no compression to summary stats necessary)
- Amortized: After upfront simulation + training phase, inference is efficient for new data or prior
- Related alternative: learn posterior $\hat{p}(\theta|x_{\mathrm{obs}})$ [G. Papamakarios et al 1605.06376; J.-M. Lueckmann et al 1711.01861]

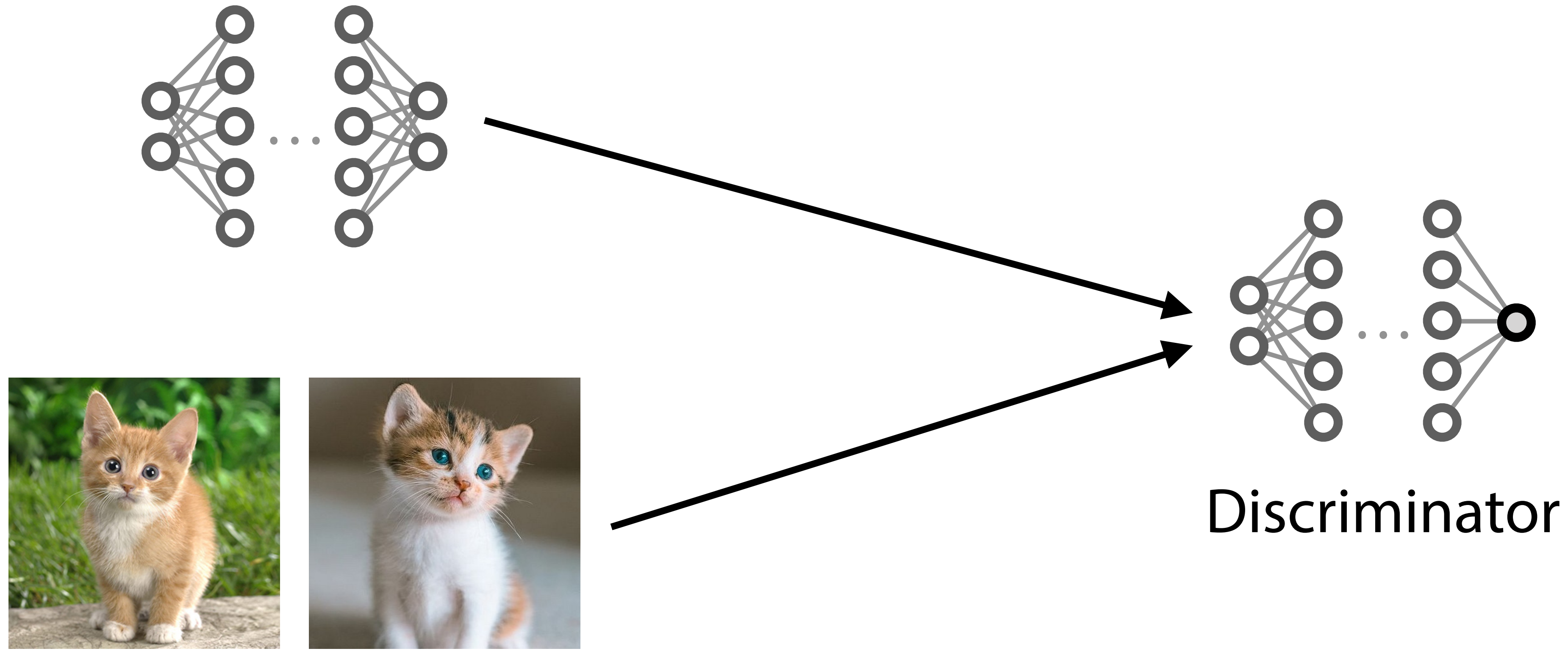
# The likelihood ratio trick

- Generative Adversarial Networks (GANs):

[I. Goodfellow et al. 1406.2661]

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

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



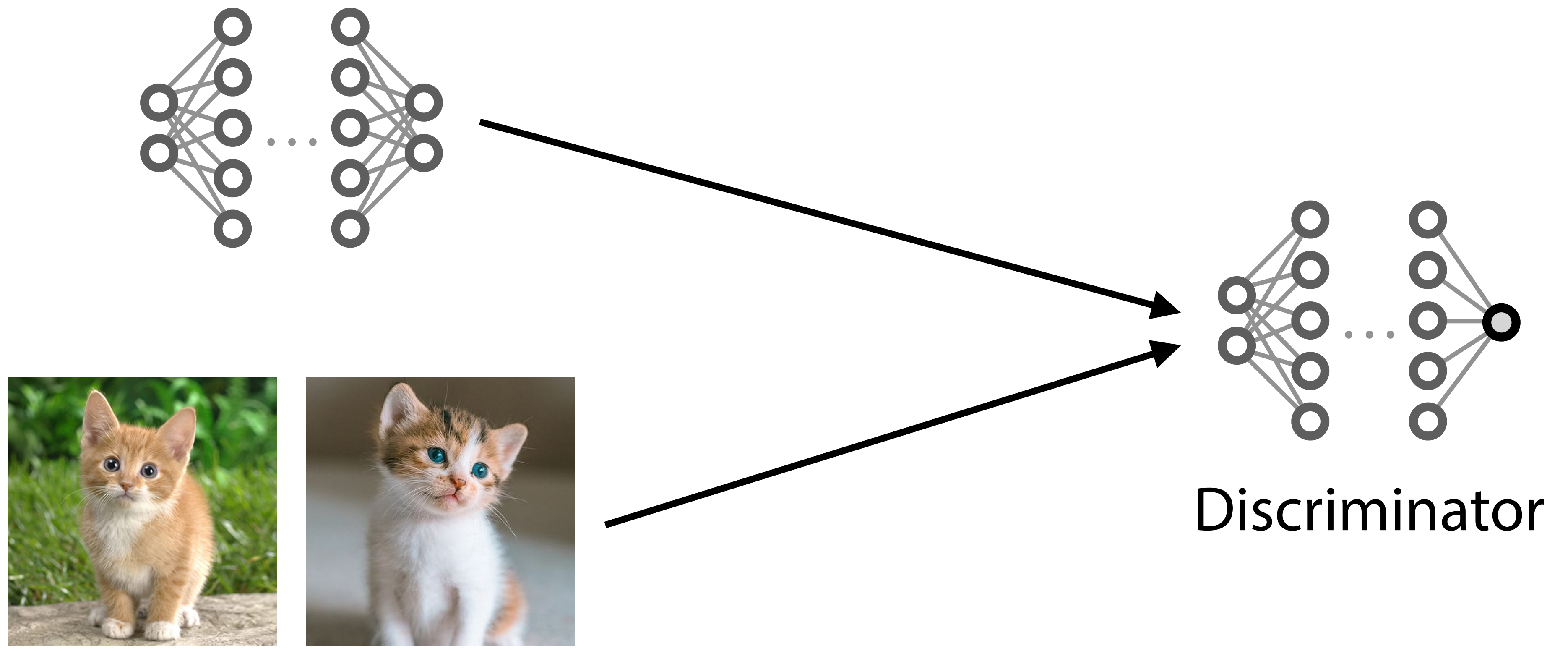
# The likelihood ratio trick

- 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

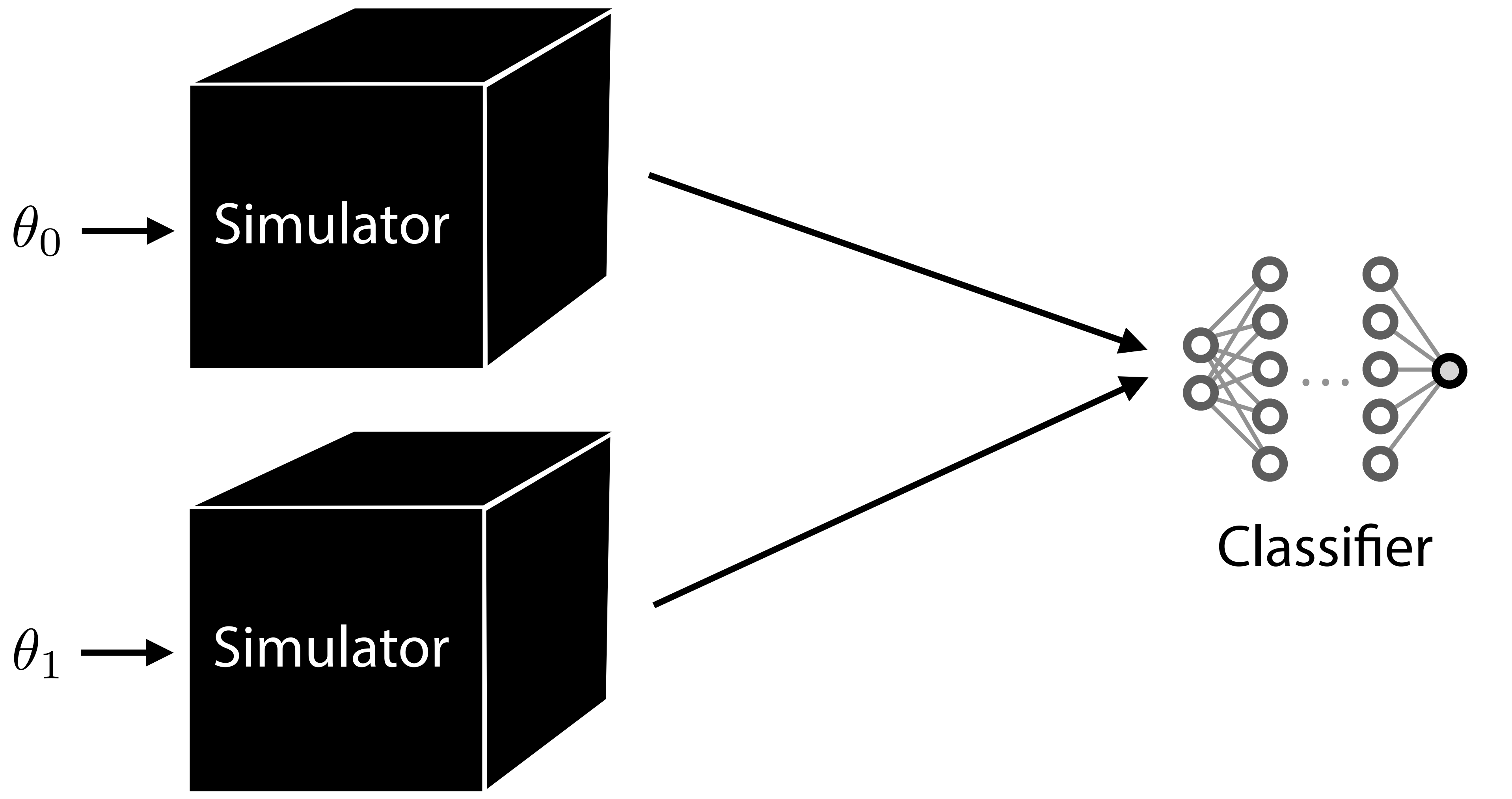
$$s(x) \to \frac{p_{\text{true}}(x)}{p_{\text{gen}}(x) + p_{\text{true}}(x)}$$

- 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 $p(x|\theta_1)$)



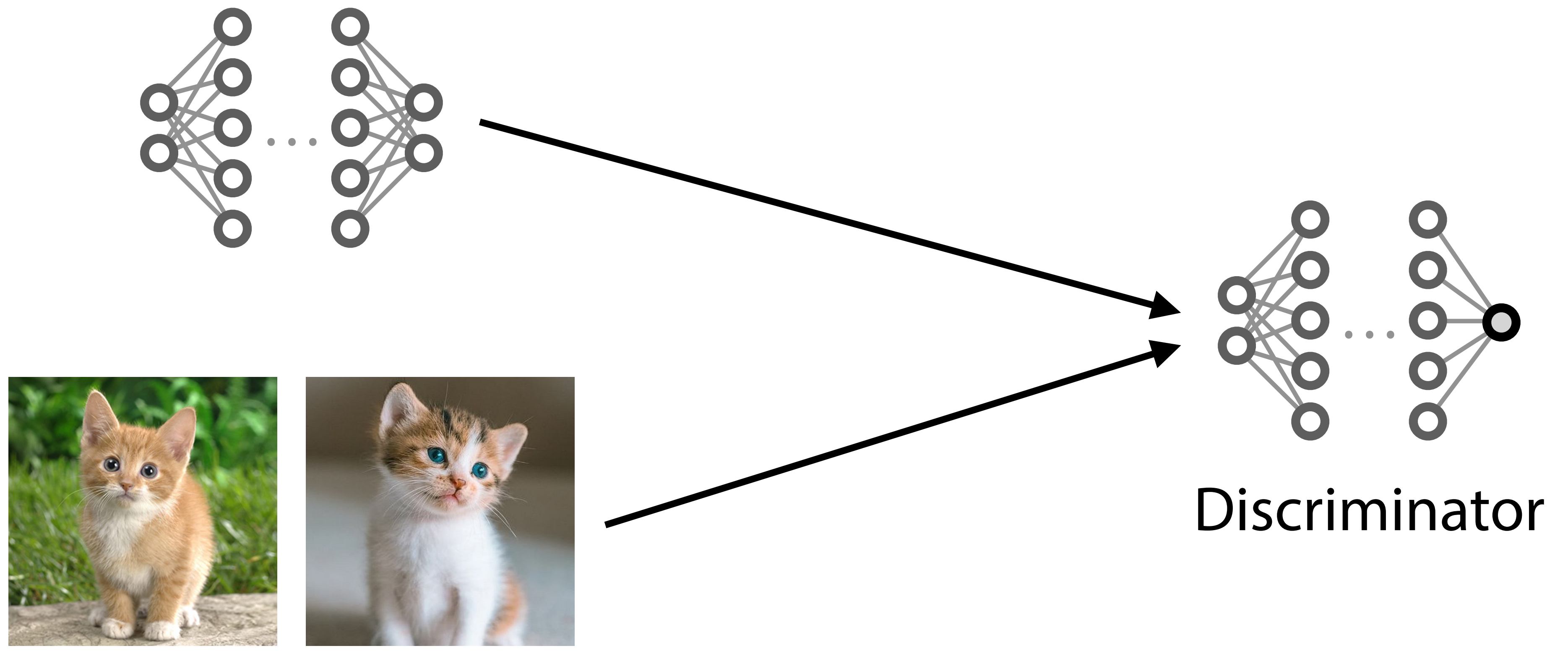
# The likelihood ratio trick

- Generative Adversarial Networks (GANs):

[I. Goodfellow et al. 1406.2661]

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

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




Discriminator learns decision function

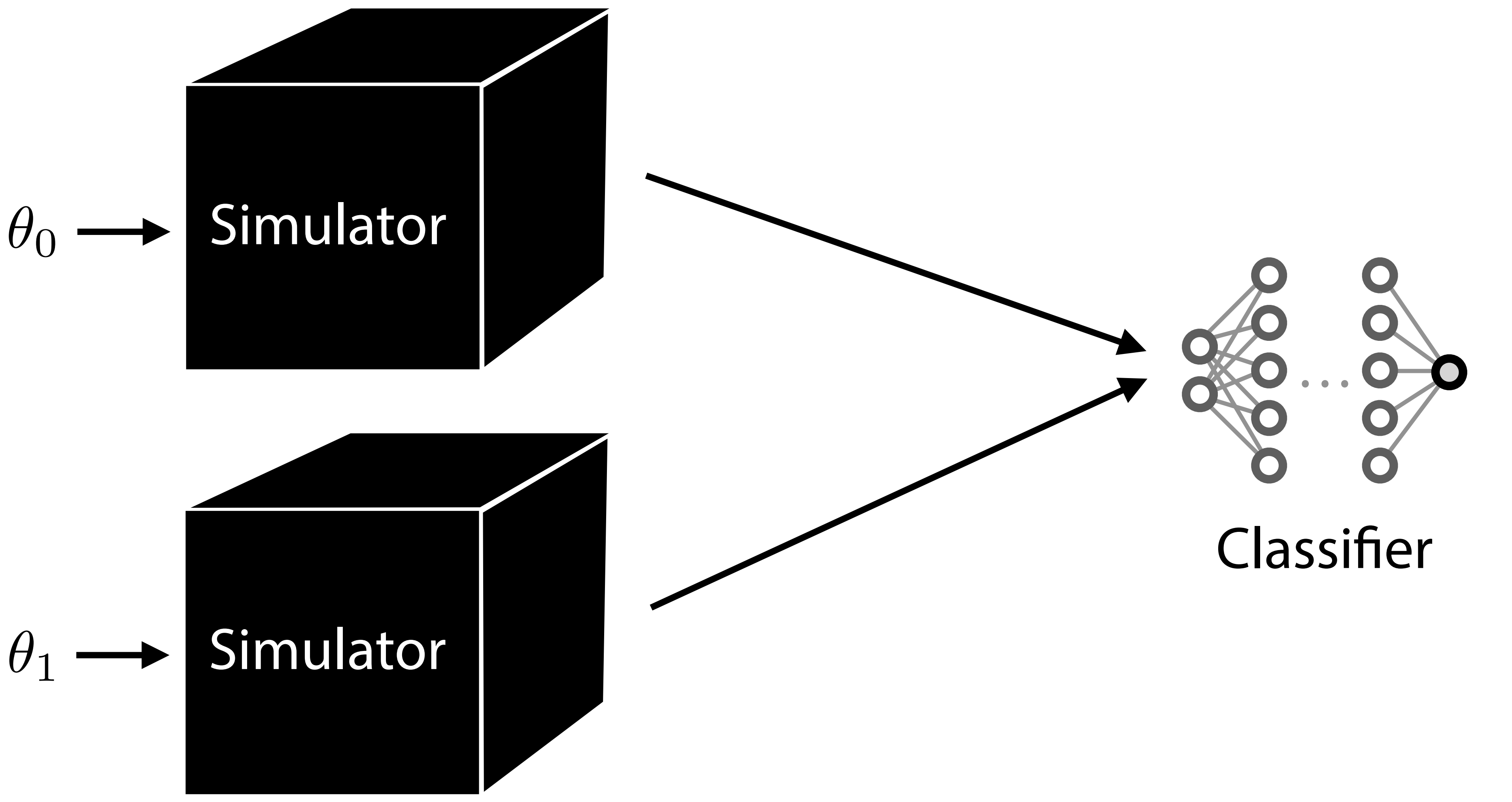
$$s(x) \to \frac{p_{\mathrm{true}}(x)}{p_{\mathrm{gen}}(x) + p_{\mathrm{true}}(x)}$$

- 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

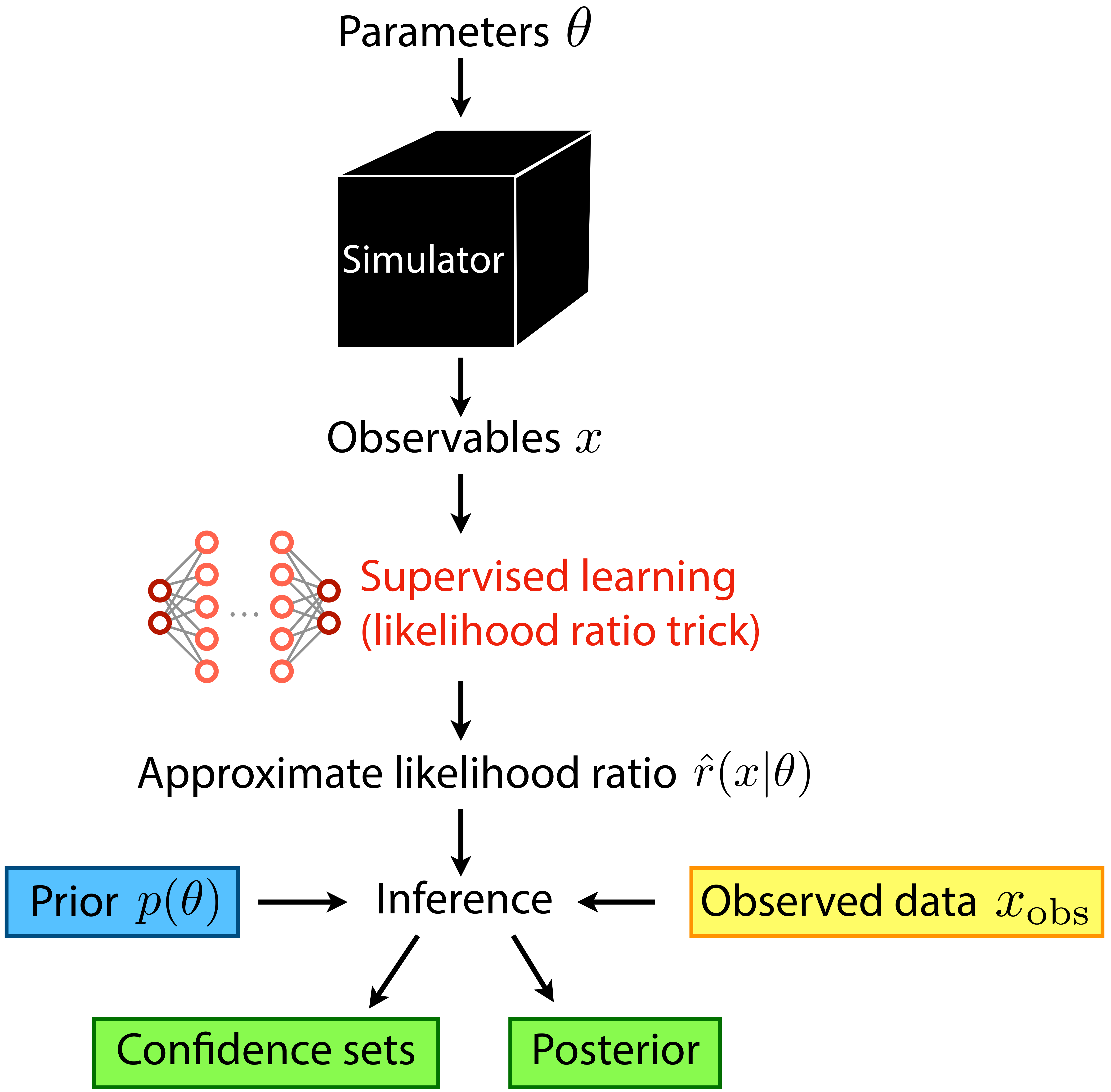
$$s(x) \to \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)}$$

⇒ 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]



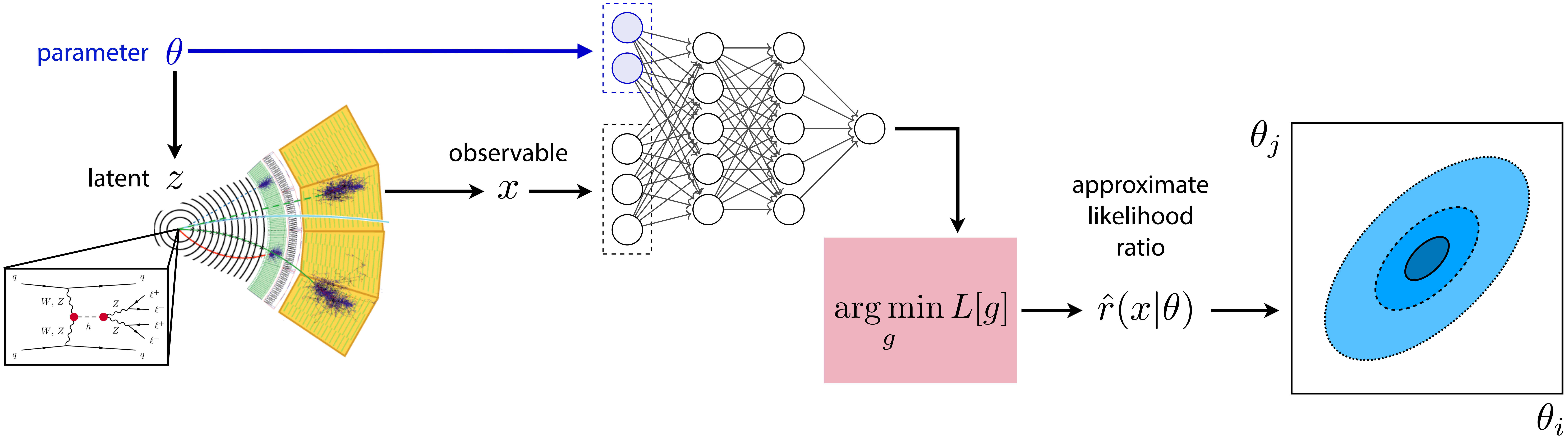

- For inference, likelihood and likelihood ratio are interchangeable
- Advantage: Learning the likelihood ratio can be a simpler task than learning the likelihood
- Disadvantage: Cannot sample from likelihood ratio

“Mining gold”:
Physics insights can make these inference methods more efficient.

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

# Learning with Simulated Data

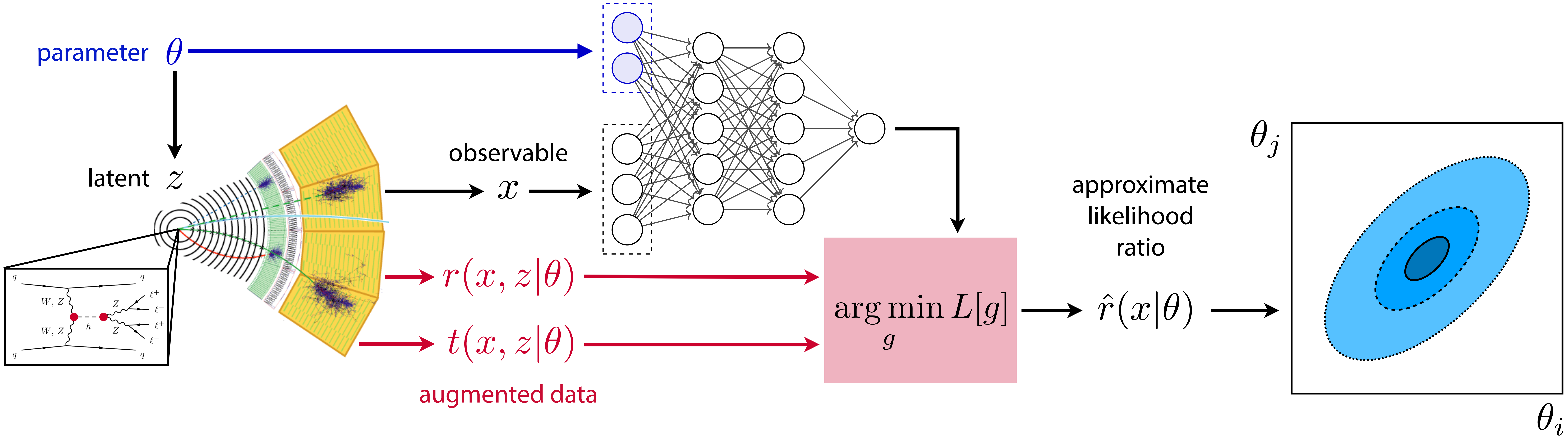
arXiv:1805.12244
PRL, arXiv:1805.00013
PRD, arXiv:1805.00020
physics.aps.org/articles/v11/90




| Simulation | Machine Learning | Inference |
|---|---|---|
| "Mining gold": Extract additional information from simulator | Use this information to train estimator for likelihood ratio | Limit setting with standard hypothesis tests |

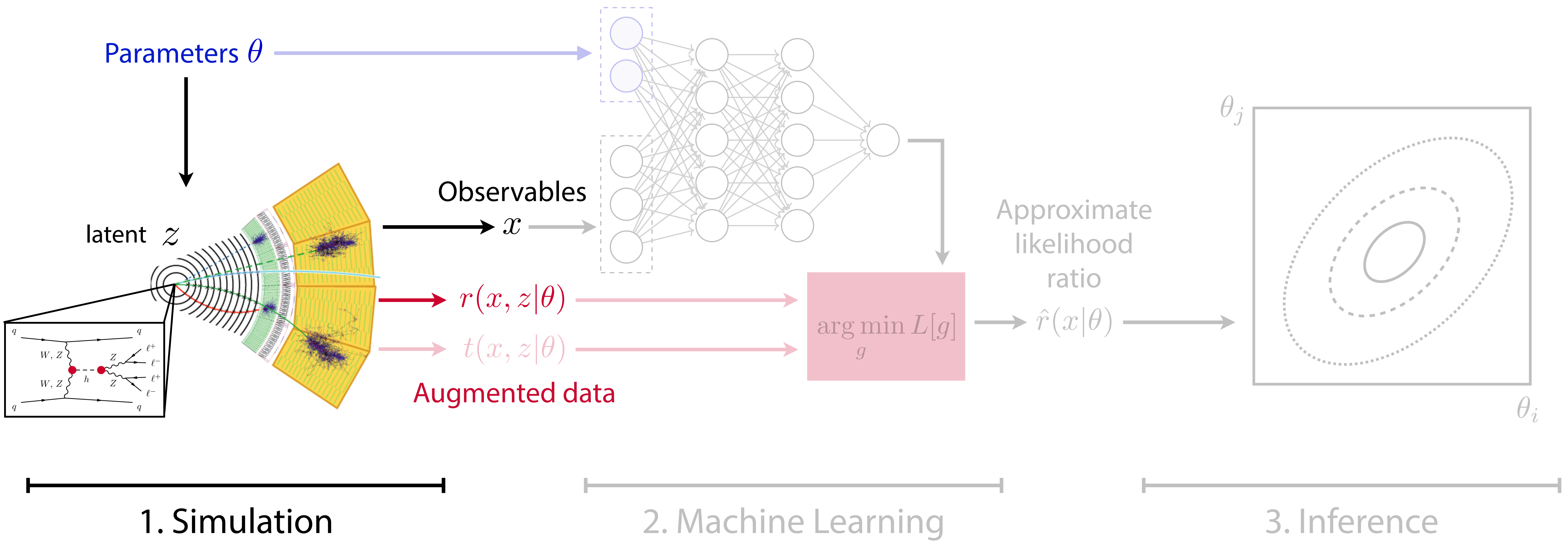
# Learning with Augmented Data

arXiv:1805.12244
PRL, arXiv:1805.00013
PRD, arXiv:1805.00020
physics.aps.org/articles/v11/90




| Simulation | Machine Learning | Inference |
|---|---|---|
| "Mining gold": Extract additional information from simulator | Use this information to train estimator for likelihood ratio | Limit setting with standard hypothesis tests |

# Learning with Augmented Data



# Mining gold from the simulator

Latent variables

| Observables | Detector interactions | Shower splittings | Parton-level momenta | Theory parameters |
|---|---|---|---|---|
| $x$ ← | $z_d$ ← | $z_s$ ← | $z_p$ ← | $\theta$ |

$$p(x|\theta) = \boxed{\int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p} \; p(x|z_d) \quad p(z_d|z_s) \quad p(z_s|z_p) \quad p(z_p|\theta)$$

Intractable integrals

# Mining gold from the simulator

Latent variables

| Observables | | Detector interactions | Shower splittings | Parton-level momenta | Theory parameters |
|---|---|---|---|---|---|
| $x$ | $\longleftarrow$ | $z_d$ $\longleftarrow$ | $z_s$ $\longleftarrow$ | $z_p$ $\longleftarrow$ | $\theta$ |

$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \quad p(x|z_d) \quad p(z_d|z_s) \quad p(z_s|z_p) \quad \boxed{p(z_p|\theta)}$$

Parton-level likelihood is given by matrix element and can be evaluated!

⇒ For each simulated event, we can calculate the joint likelihood ratio which depends on the specific evolution of the simulation:

$$r(x,z|\theta_0,\theta_1) \equiv \frac{p(x,z_d,z_s,z_p|\theta_0)}{p(x,z_d,z_s,z_p|\theta_1)} = \frac{p(x|z_d)}{p(x|z_d)} \frac{p(z_d|z_s)}{p(z_d|z_s)} \frac{p(z_s|z_p)}{p(z_s|z_p)} \boxed{\frac{p(z_p|\theta_0)}{p(z_p|\theta_1)} \sim \frac{|\mathcal{M}(z_p|\theta_0)|^2}{|\mathcal{M}(z_p|\theta_1)|^2}}$$

# The value of gold

We can calculate the joint likelihood ratio

$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p|\theta_0)}{p(x, z_d, z_s, z_p|\theta_1)}$$

("How much more likely is this simulated event, including all intermediate states, for $\theta_0$ compared to $\theta_1$?")

↓

We want the likelihood ratio function

$$r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

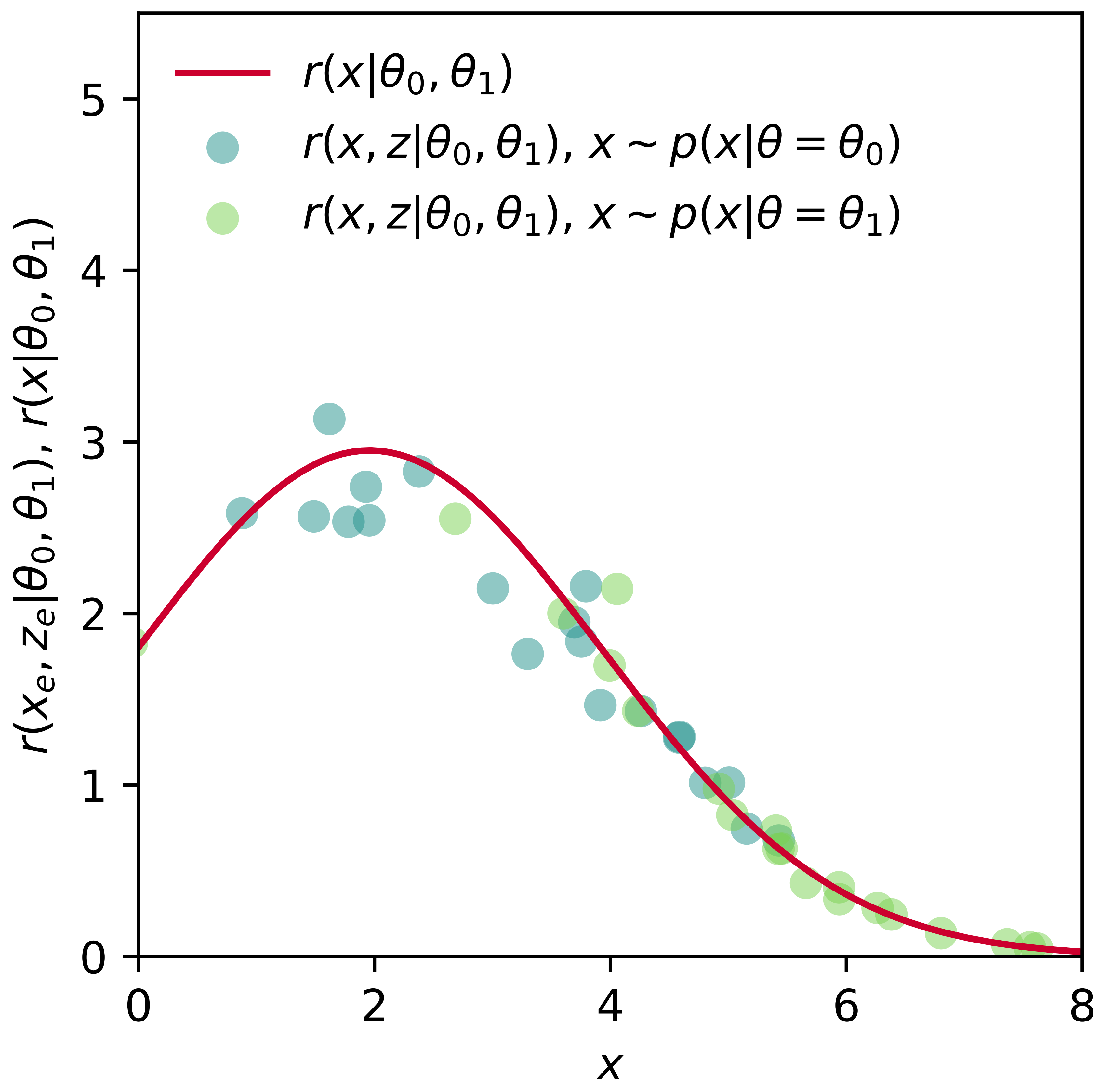
("How much more likely is the observation $x$ for $\theta_0$ compared to $\theta_1$?")

# The value of gold

We can calculate the joint likelihood ratio

$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p|\theta_0)}{p(x, z_d, z_s, z_p|\theta_1)}$$

$r(x, z|\theta_0, \theta_1)$ are scattered around $r(x|\theta_0, \theta_1)$




We want the likelihood ratio function

$$r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

# The value of gold

We can calculate the joint likelihood ratio

$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p|\theta_0)}{p(x, z_d, z_s, z_p|\theta_1)}$$

↓

We want the likelihood ratio function

$$r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

With $r(x, z|\theta_0, \theta_1)$, we define a functional like

$$L_r[\hat{r}(x|\theta_0, \theta_1)] = \int \mathrm{d}x \int \mathrm{d}z \; p(x, z|\theta_1) \left[ (\hat{r}(x|\theta_0, \theta_1) - r(x, z|\theta_0, \theta_1))^2 \right]$$

It is minimized by

$$\mathbb{E}_{z \sim p(z|x,\theta_1)} [r(x, z|\theta_0, \theta_1)] = \underset{\hat{r}(x|\theta_0,\theta_1)}{\arg\min} \; L_r[\hat{r}(x|\theta_0, \theta_1)]!$$

(And we can sample from $p(x, z|\theta)$ by running the simulator.)

# The value of gold

We can calculate the joint likelihood ratio

$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p|\theta_0)}{p(x, z_d, z_s, z_p|\theta_1)}$$

We want the likelihood ratio function

$$r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

With $r(x, z|\theta_0, \theta_1)$, we define a functional like

$$L_r[\hat{r}(x|\theta_0, \theta_1)] = \int \mathrm{d}x \int \mathrm{d}z \; p(x, z|\theta_1) \left[ (\hat{r}(x|\theta_0, \theta_1) - r(x, z|\theta_0, \theta_1))^2 \right]$$

It is minimized by

$$\mathbb{E}_{z \sim p(z|x,\theta_1)} [r(x, z|\theta_0, \theta_1)] = \underset{\hat{r}(x|\theta_0,\theta_1)}{\arg\min} \; L_r[\hat{r}(x|\theta_0, \theta_1)]!$$

(And we can sample from $p(x, z|\theta)$ by running the simulator.)

**.... and then magic ...**

$$\mathbb{E}_{z \sim p(z|x,\theta_1)} [r(x, z|\theta_0, \theta_1)] = \int \mathrm{d}z \; p(z|x, \theta_1) \frac{p(x, z|\theta_0)}{p(x, z|\theta_1)}$$

$$= \int \mathrm{d}z \; \frac{p(x, z|\theta_1)}{p(x|\theta_1)} \frac{p(x, z|\theta_0)}{p(x, z|\theta_1)}$$

$$= r(x|\theta_0, \theta_1) \; !$$

# Machine learning = applied calculus of variatios

So to get a good estimator of the likelihood ratio, we need to mimimize a functional numerically:

Variational family

Extremization

Functional with integral

$$r(x|\theta_0, \theta_1) = \arg\min_{\hat{r}(x|\theta_0,\theta_1)} L_r[\hat{r}(x|\theta_0, \theta_1)]$$

# Machine learning = applied calculus of variatios

So to get a good estimator of the likelihood ratio, we need to mimimize a functional numerically:

Variational family

Extremization

Functional with integral

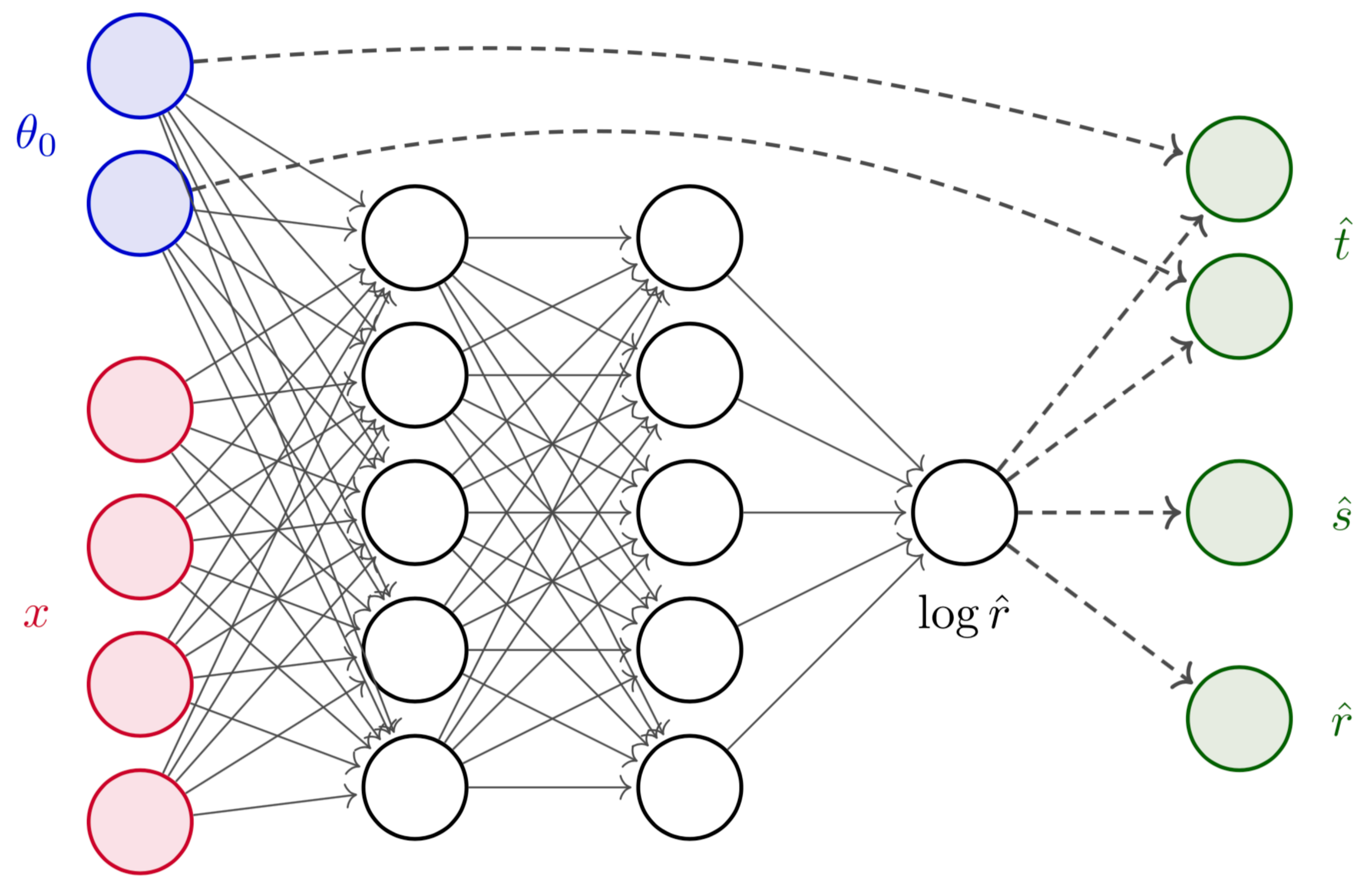
$$r(x|\theta_0,\theta_1) = \arg\min_{\hat{r}(x|\theta_0,\theta_1)} L_r[\hat{r}(x|\theta_0,\theta_1)]$$

Loss function with finite sum over samples

Stochastic gradient descent

This is where machine learning comes in!

Neural network



# Machine learning = applied calculus of variatios

So to get a good estimator of the likelihood ratio, we need to mimimize a functional numerically:

Variational family

Extremization

Functional with integral

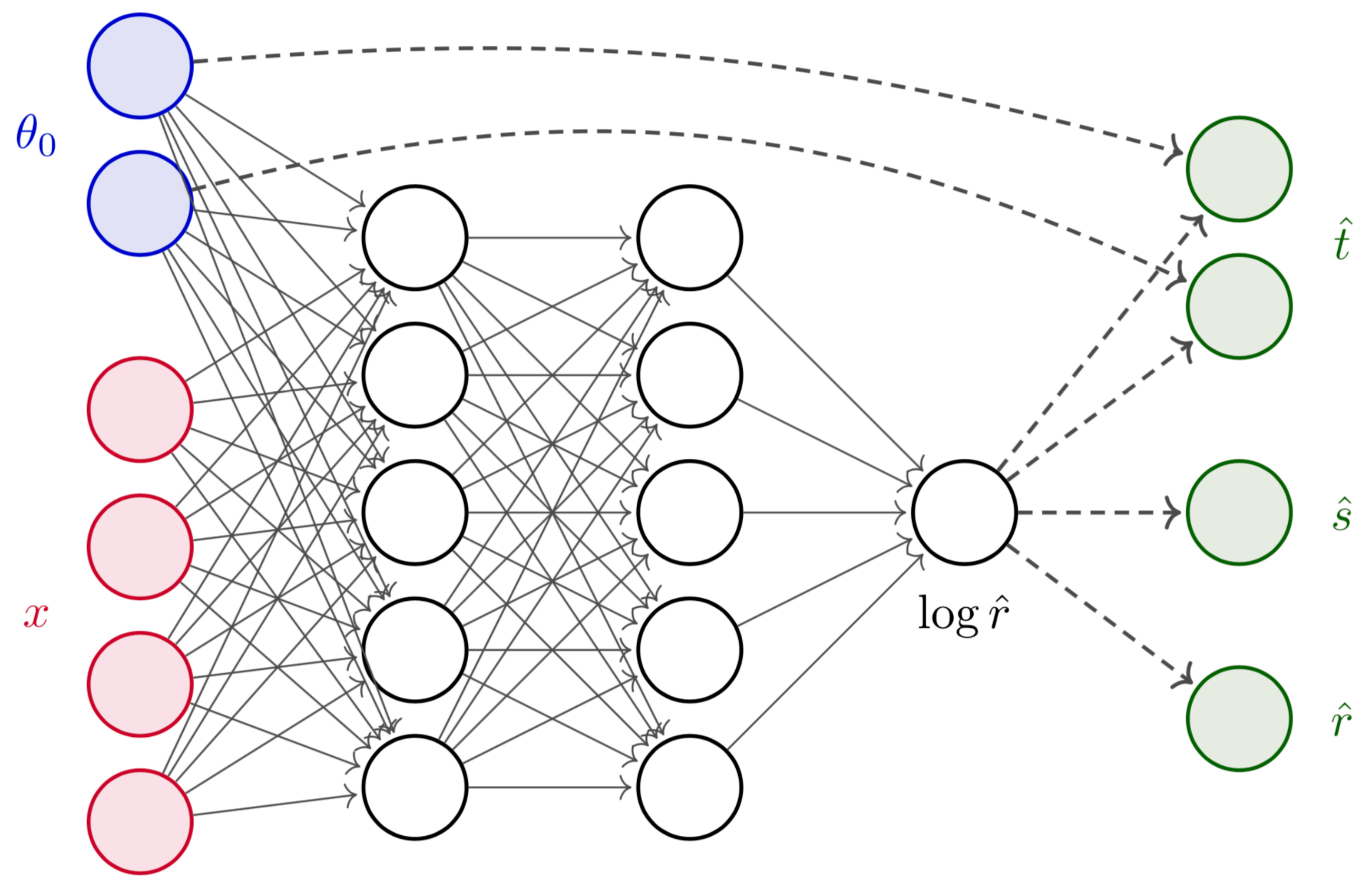
$$r(x|\theta_0, \theta_1) = \arg\min_{\hat{r}(x|\theta_0,\theta_1)} L_r[\hat{r}(x|\theta_0, \theta_1)]$$

This is where machine learning comes in!

Neural network

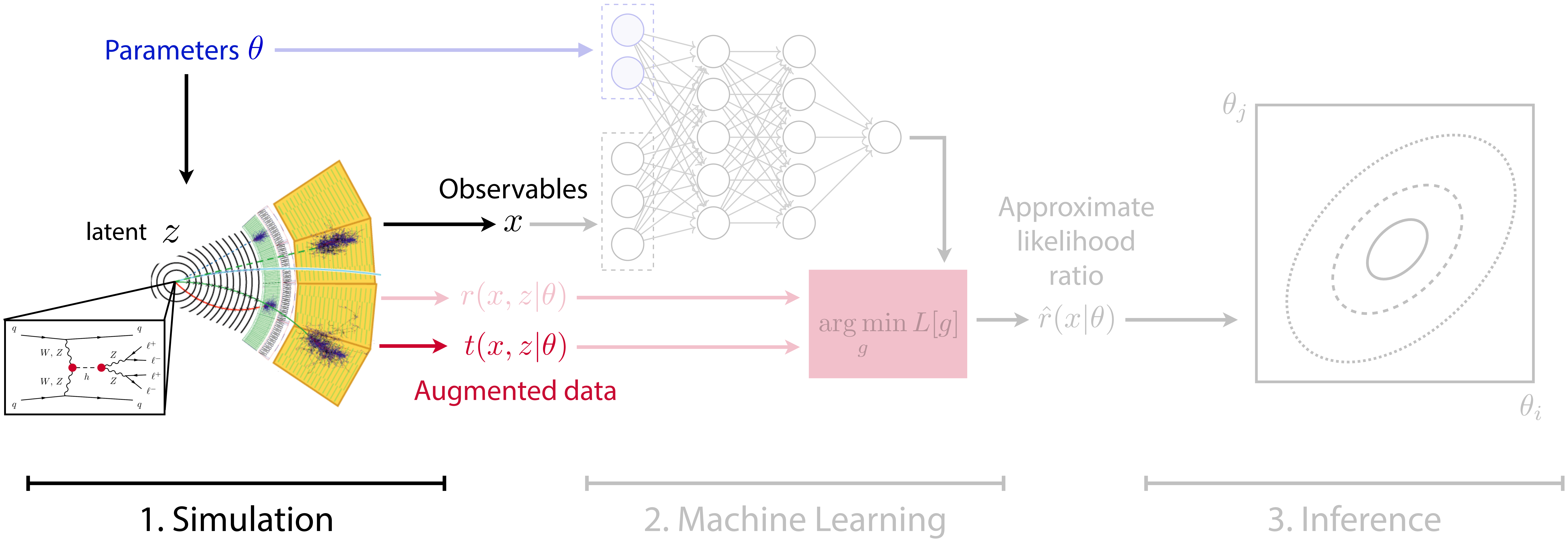
Stochastic gradient descent

Loss function with finite sum over samples



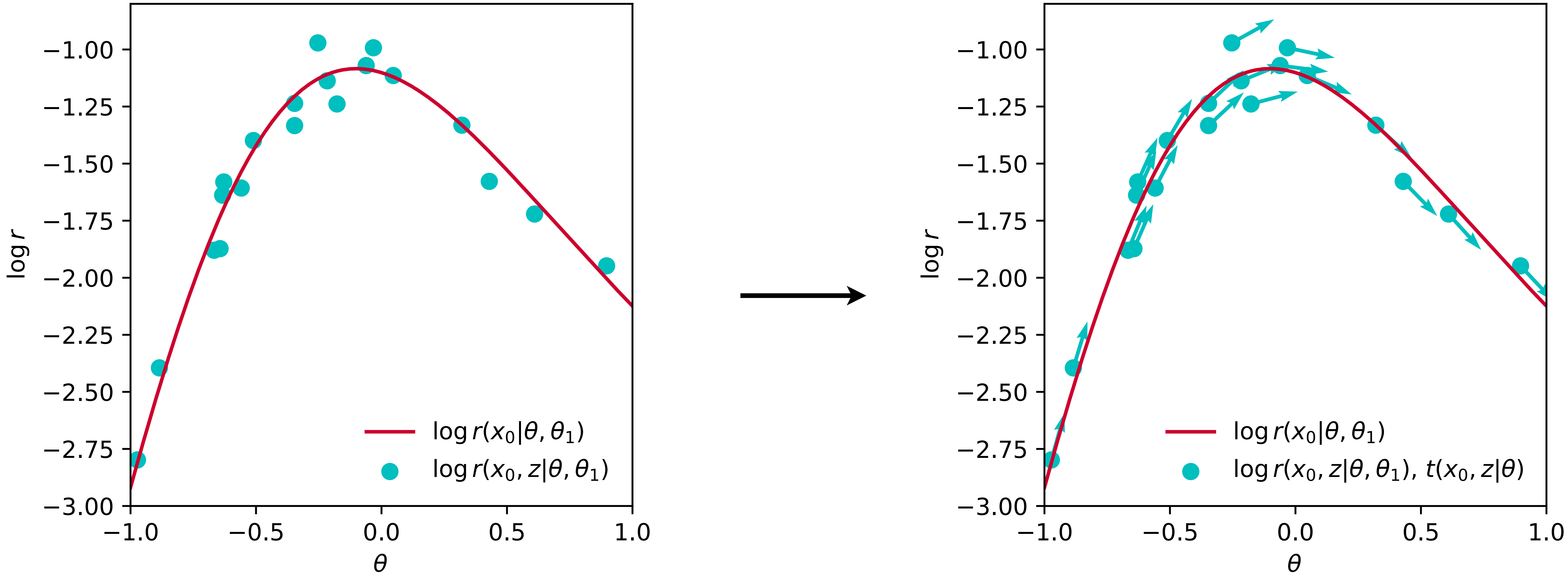

A sufficiently expressive neural network efficiently trained in this way with enough data will learn the likelihood ratio function $r(x|\theta_0, \theta_1)$!

# Learning with Augmented Data



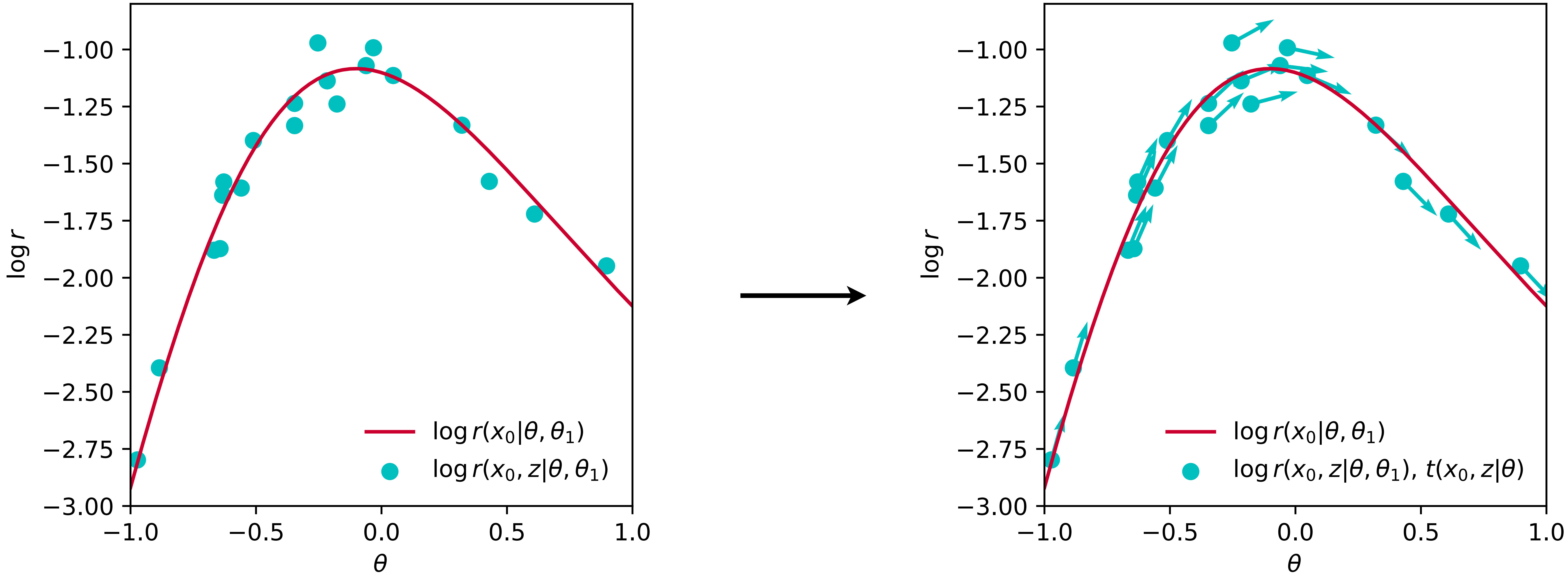
# One more piece: the score

- Knowing derivative often helps fitting:



# One more piece: the score

- Knowing derivative often helps fitting:




- In our case, the relevant quantity is the score $t(x|\theta_0) \equiv \nabla_\theta \log p(x|\theta)\Big|_{\theta_0}$ .
  - The score itself is intractable. But…

# Learning the score

Similar to the joint likelihood ratio, from the simulator we can extract the joint score

$$t(x, z|\theta_0) \equiv \nabla_\theta \log p(x, z_d, z_s, z_p|\theta)\Big|_{\theta_0}$$

↓

We want the score

$$t(x|\theta_0) \equiv \nabla_\theta \log p(x|\theta)\Big|_{\theta_0}$$

# Learning the score

Similar to the joint likelihood ratio, from the simulator we can extract the joint score

$$t(x, z|\theta_0) \equiv \nabla_\theta \log p(x, z_d, z_s, z_p|\theta)\Big|_{\theta_0}$$

We want the score

$$t(x|\theta_0) \equiv \nabla_\theta \log p(x|\theta)\Big|_{\theta_0}$$

Given $t(x, z|\theta_0)$,
we define the functional

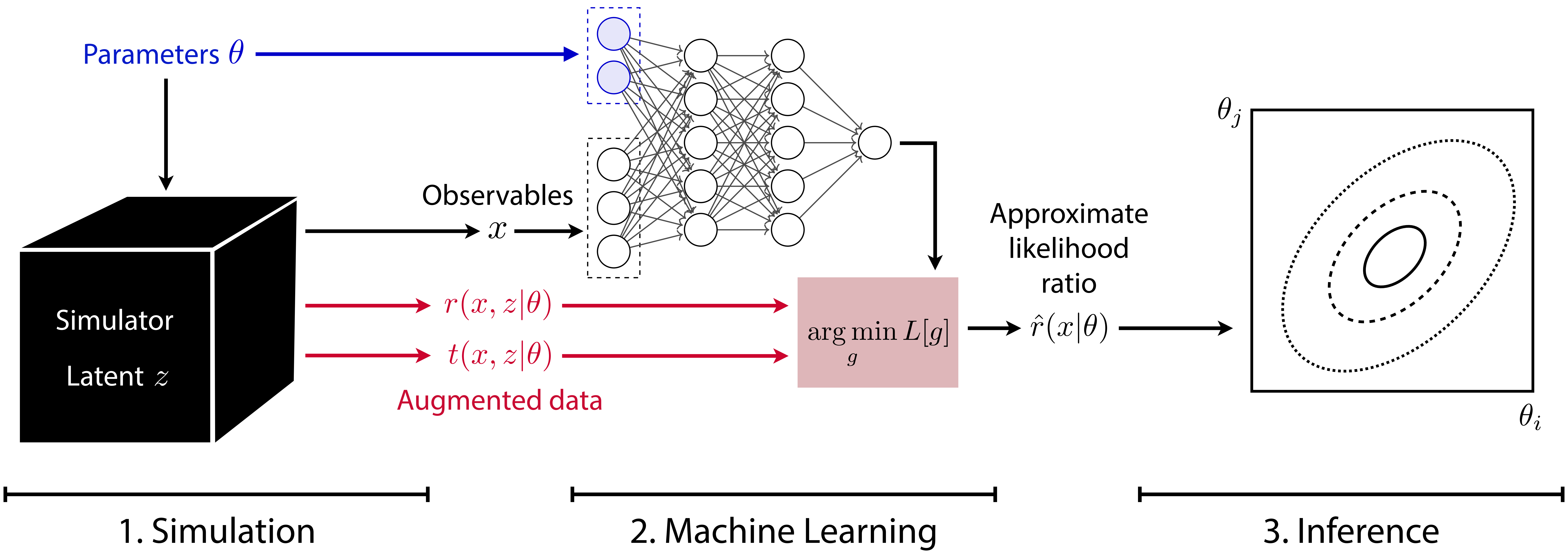
$$L_t[\hat{t}(x|\theta_0)] = \int \mathrm{d}x \int \mathrm{d}z \;\; p(x, z|\theta_0) \; \Big[\big(\hat{t}(x|\theta_0) - t(x, z|\theta_0)\big)^2\Big].$$

One can show it is minimized by

$$t(x|\theta_0) = \underset{\hat{t}(x|\theta_0)}{\arg\min}\, L_t[\hat{t}(x|\theta_0)]\,.$$

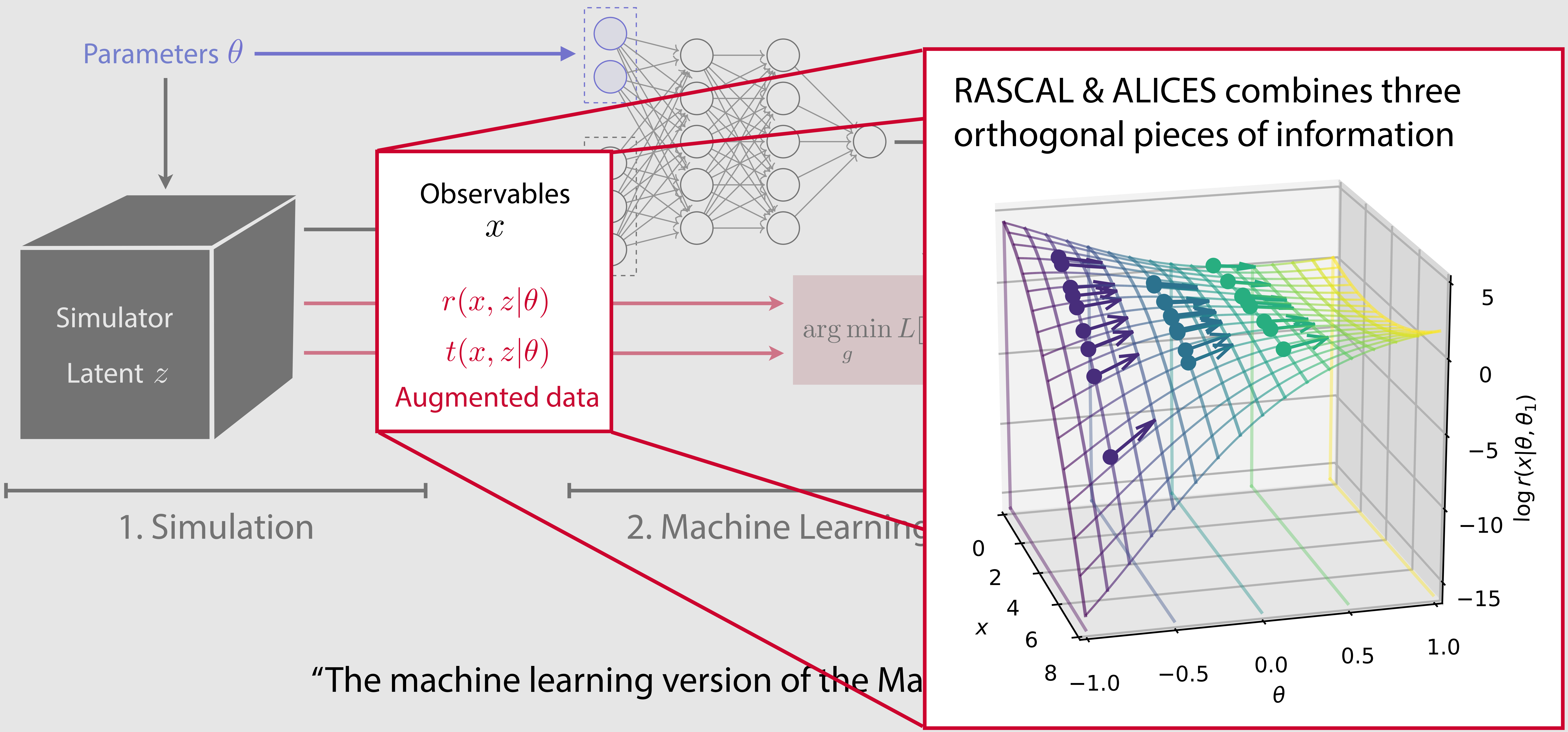
Again, we implement this minimization through machine learning.

# Putting the pieces together: RASCAL & ALICES



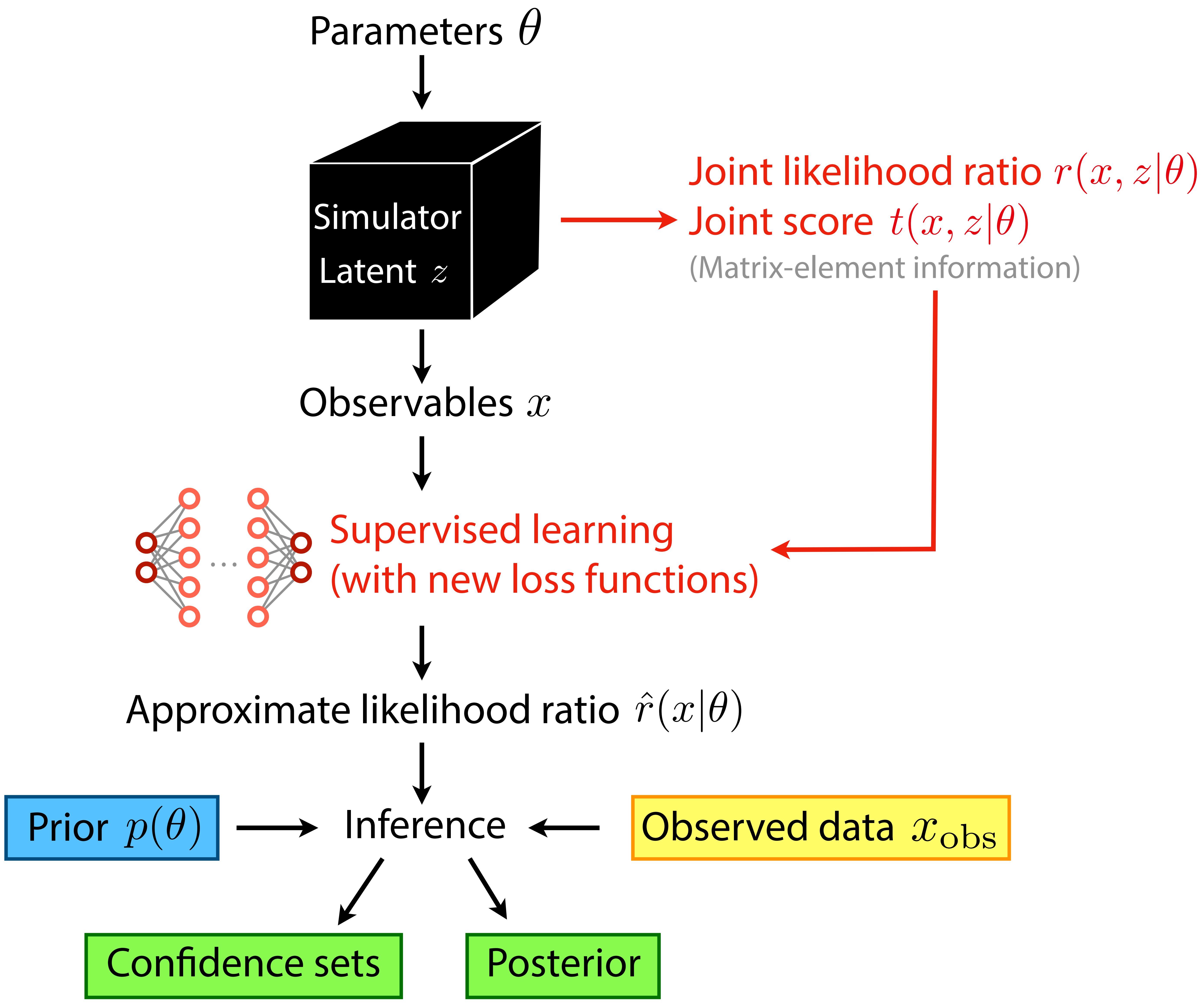

"The machine learning version of the Matrix Element Method"

# Putting the pieces together: RASCAL & ALICES




“The machine learning version of the Ma

# Mining gold: Summary




"The machine learning version of the Matrix Element Method"

- Scales to high-dimensional data (no summary statistics necessary)
- Uses matrix-element information to improve sample efficiency
- Neural networks learn effect of shower + detector (no transfer functions & works with ME+PS matching)
- Amortized (evaluation in μs)

MadMiner automates all of these methods.

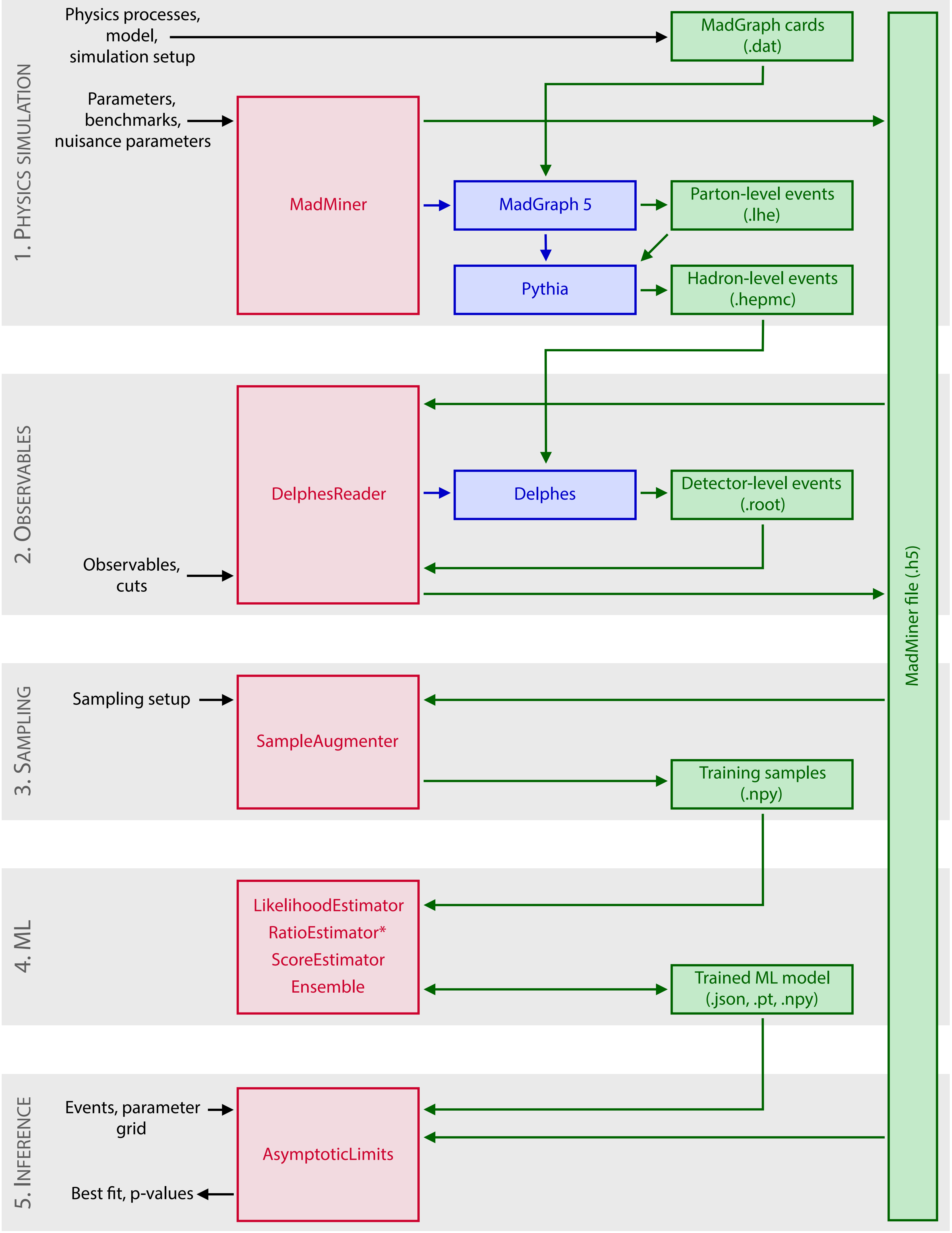
[JB, F. Kling, I. Espejo, K. Cranmer 1907.10621]

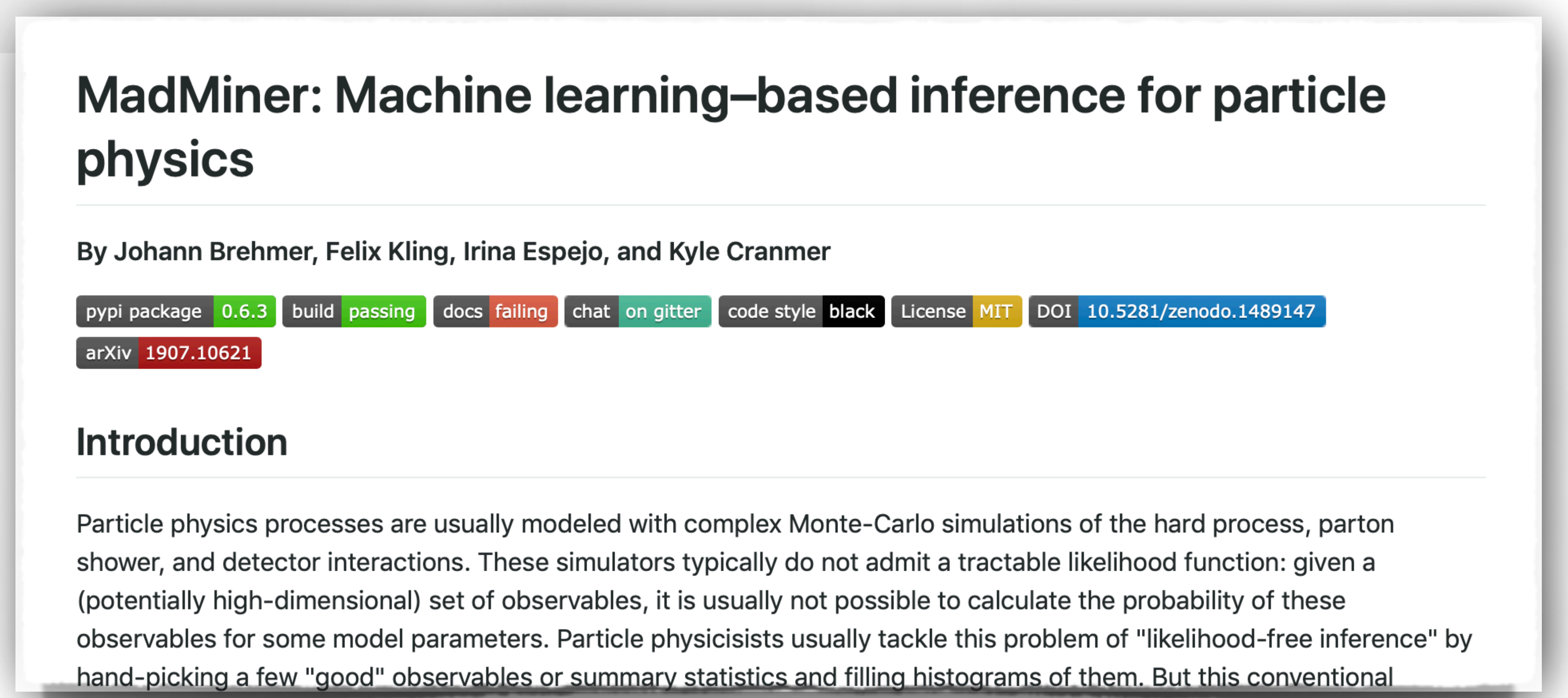
# Automation

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

New Python package MadMiner makes it straightforward to apply the new techniques to LHC problems

- Out of the box: Pheno-level analyses
  - MadGraph, Pythia, Delphes
  - Systematic uncertainties from PDF / scale variation
- Scalable to state-of-the-art experimental tools
  - Mostly requires bookkeeping of fully differential cross sections
- Modular interface
  - Extensive documentation
  - Embedded into Python / ML ecosystem





# MadMiner: Machine learning–based inference for particle physics

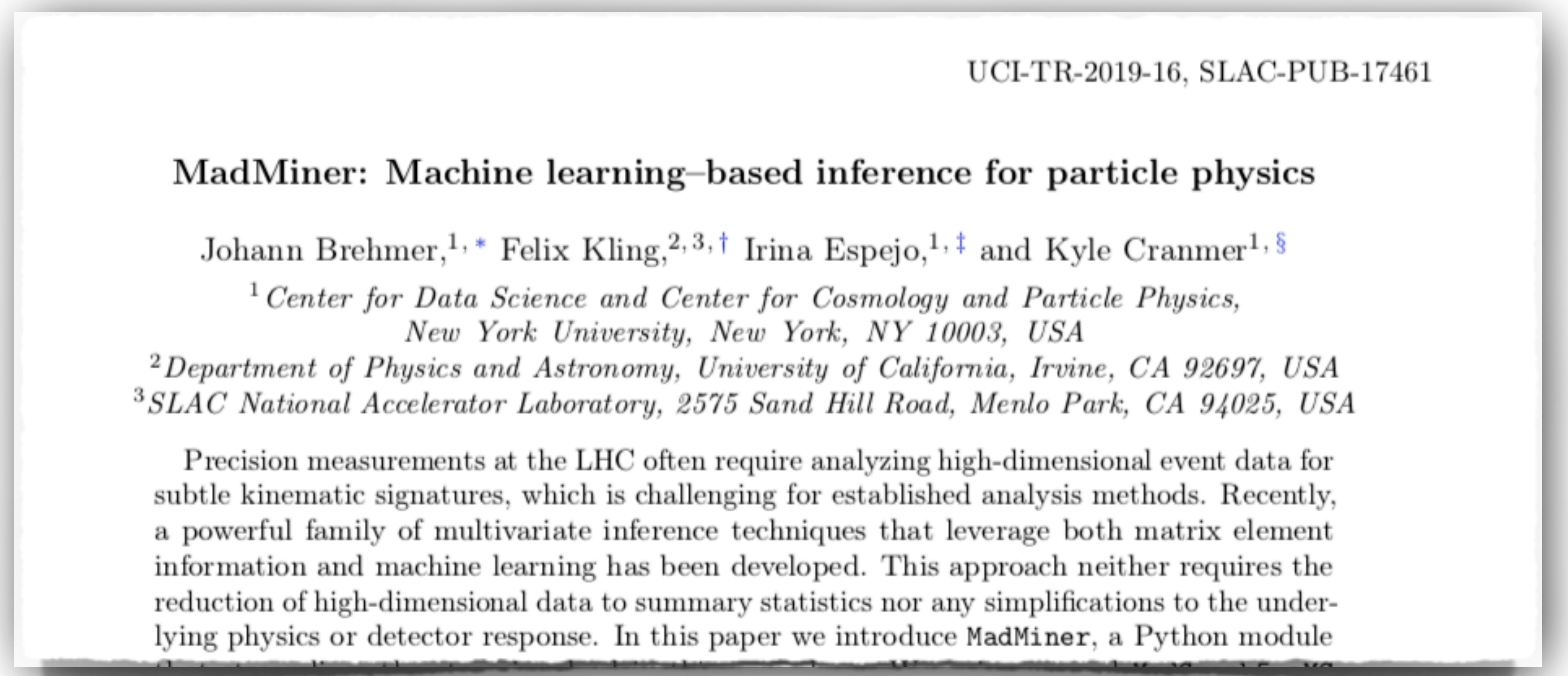
By Johann Brehmer, Felix Kling, Irina Espejo, and Kyle Cranmer

pypi package 0.6.3 | build passing | docs failing | chat on gitter | code style black | License MIT | DOI 10.5281/zenodo.1489147 | arXiv 1907.10621

## Introduction

Particle physics processes are usually modeled with complex Monte-Carlo simulations of the hard process, parton shower, and detector interactions. These simulators typically do not admit a tractable likelihood function: given a (potentially high-dimensional) set of observables, it is usually not possible to calculate the probability of these observables for some model parameters. Particle physicisists usually tackle this problem of "likelihood-free inference" by hand-picking a few "good" observables or summary statistics and filling histograms of them. But this conventional

Repository and tutorials:
github.com/johannbrehmer/madminer



UCI-TR-2019-16, SLAC-PUB-17461

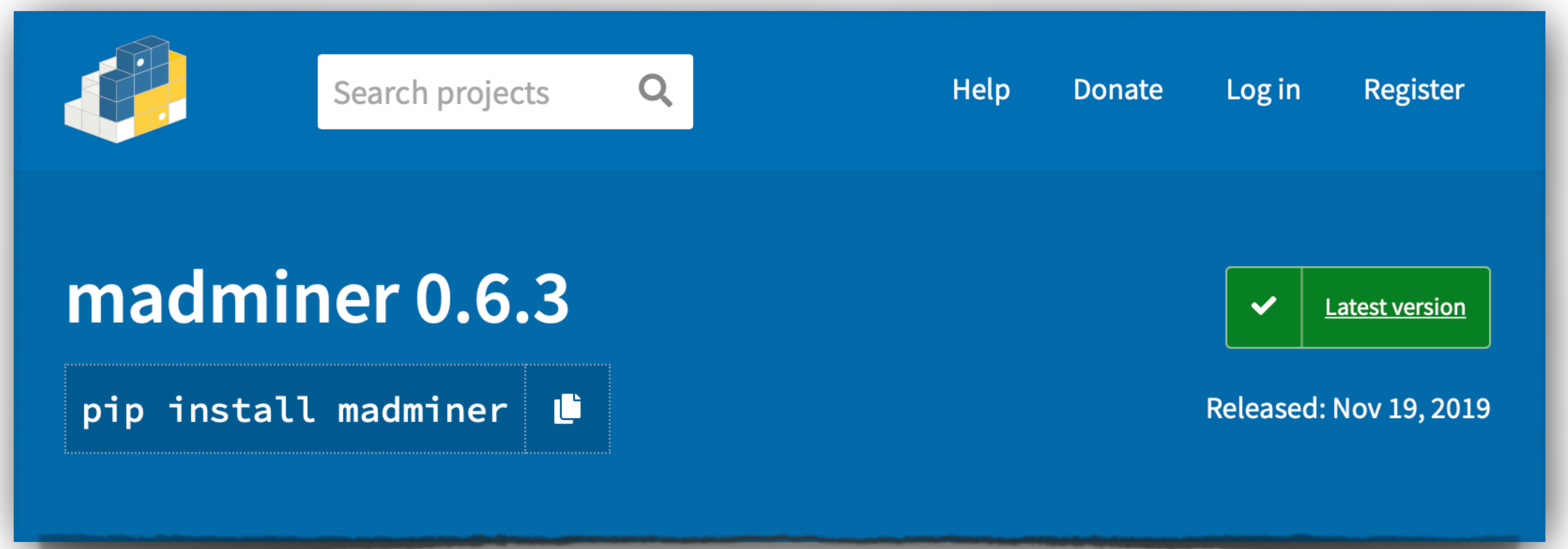
MadMiner: Machine learning–based inference for particle physics

Johann Brehmer,[1,*] Felix Kling,[2,3,†] Irina Espejo,[1,‡] and Kyle Cranmer[1,§]

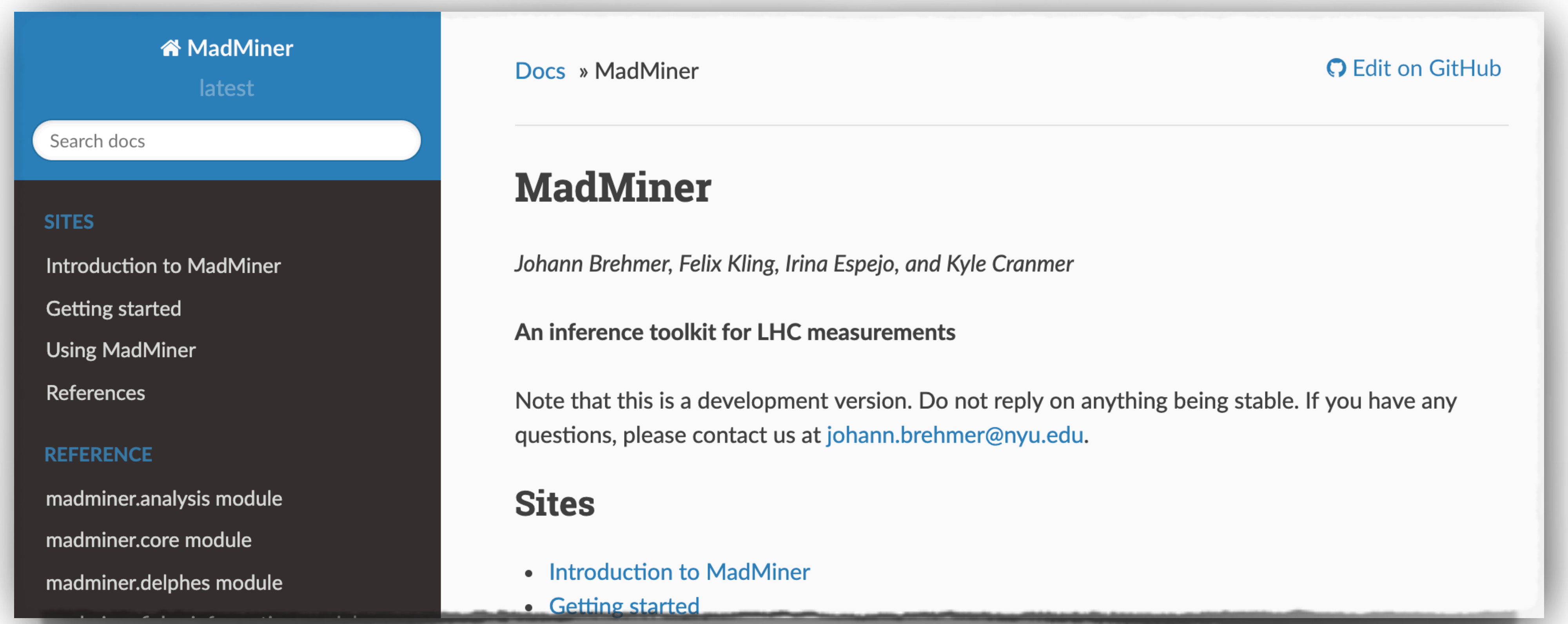
[1] *Center for Data Science and Center for Cosmology and Particle Physics, New York University, New York, NY 10003, USA*
[2] *Department of Physics and Astronomy, University of California, Irvine, CA 92697, USA*
[3] *SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA*

Precision measurements at the LHC often require analyzing high-dimensional event data for subtle kinematic signatures, which is challenging for established analysis methods. Recently, a powerful family of multivariate inference techniques that leverage both matrix element information and machine learning has been developed. This approach neither requires the reduction of high-dimensional data to summary statistics nor any simplifications to the underlying physics or detector response. In this paper we introduce MadMiner, a Python module

Paper with detailed explanations:
1907.10621




Installation:
`pip install madminer`




API documentation:
madminer.readthedocs.io

We can also machine learn optimal observables.

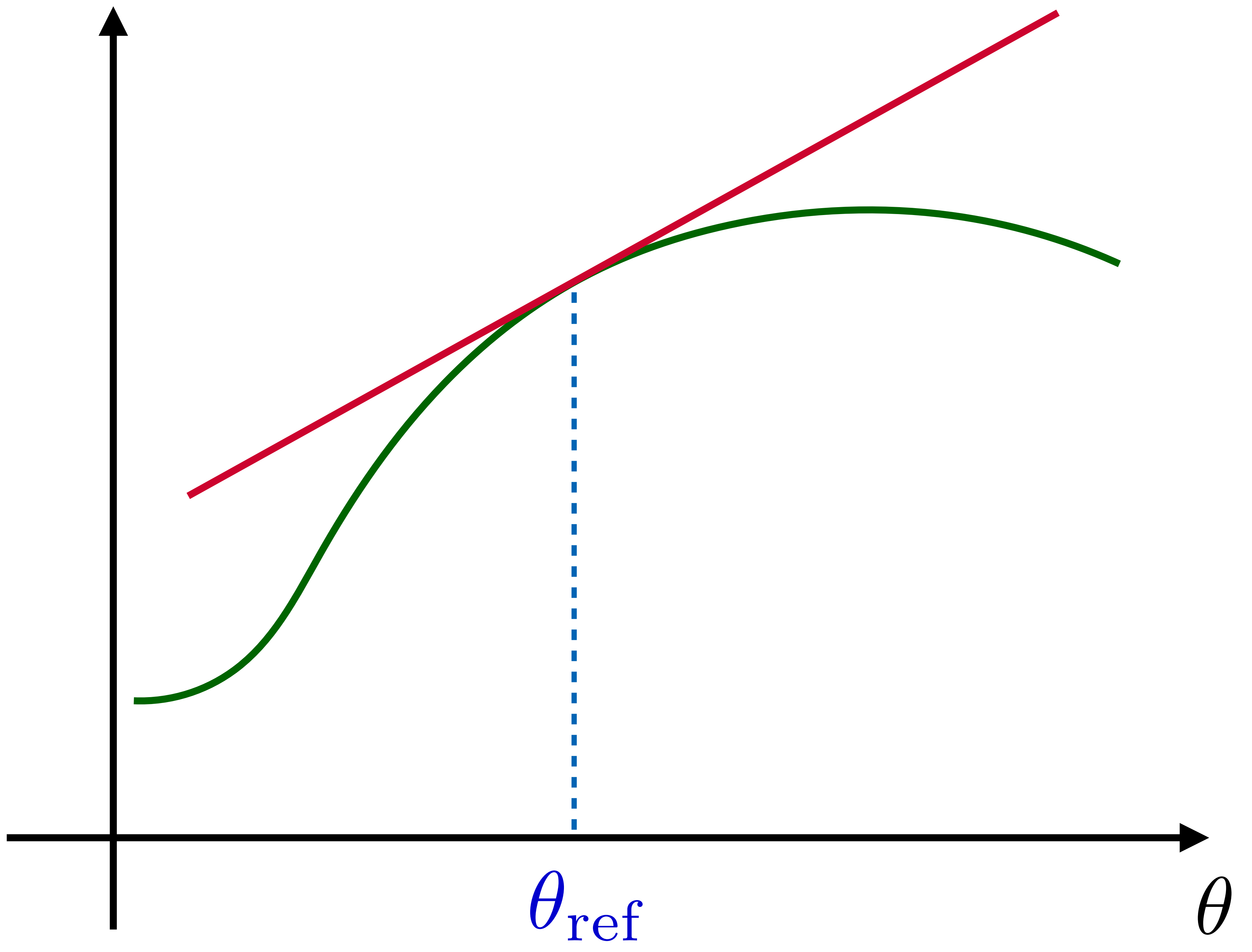
[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013, 1805.00020, 1805.12244]

# The local model

[see also J. Alsing, B. Wandelt 1712.00012; J. Alsing, B. Wandelt. S. Freeney 1801.01497; P. de Castro, T. Dorigo 1806.04743; J. Alsing, B. Wandelt 1903.01473]

Taylor expansion of $\log p(x|\theta)$ around $\theta_{\text{ref}}$:

$$\log p(x|\theta) = \log p(x|\theta_{\text{ref}}) + \underbrace{\nabla_\theta \log p(x|\theta)\Big|_{\theta_{\text{ref}}}}_{\equiv t(x|\theta_{\text{ref}})} \cdot (\theta - \theta_{\text{ref}}) + \mathcal{O}\left((\theta - \theta_{\text{ref}})^2\right)$$



# The local model

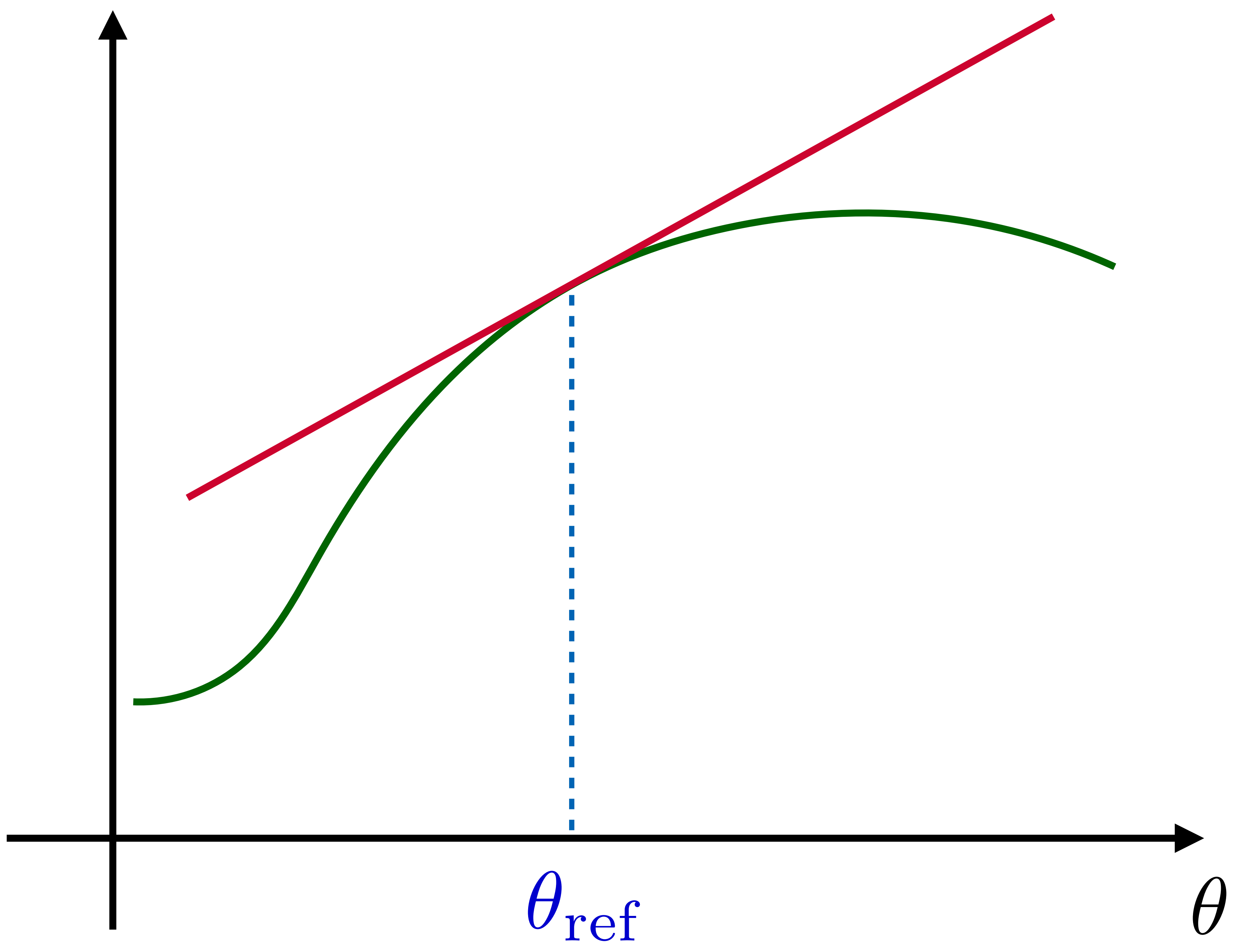
[see also J. Alsing, B. Wandelt 1712.00012; J. Alsing, B. Wandelt. S. Freeney 1801.01497; P. de Castro, T. Dorigo 1806.04743; J. Alsing, B. Wandelt 1903.01473]

Taylor expansion of $\log p(x|\theta)$ around $\theta_{\mathrm{ref}}$:

$$\log p(x|\theta) = \log p(x|\theta_{\mathrm{ref}}) + \underbrace{\nabla_\theta \log p(x|\theta)\Big|_{\theta_{\mathrm{ref}}}}_{\equiv t(x|\theta_{\mathrm{ref}})} \cdot (\theta - \theta_{\mathrm{ref}}) + \mathcal{O}\left((\theta - \theta_{\mathrm{ref}})^2\right)$$

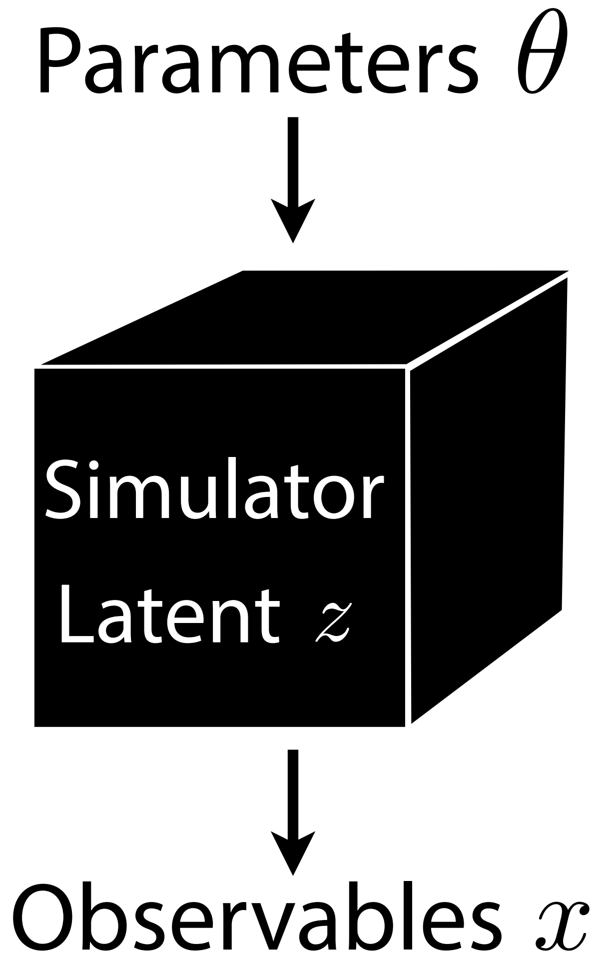
In the neighborhood of $\theta_{\mathrm{ref}}$ (e.g. close to the SM):

- the score vector $t(x|\theta_{\mathrm{ref}})$ components are sufficient statistics
- knowing $t(x|\theta_{\mathrm{ref}})$ is just as powerful as knowing the full function $\log p(x|\theta)$
- $t(x|\theta_{\mathrm{ref}})$ are the most powerful observables

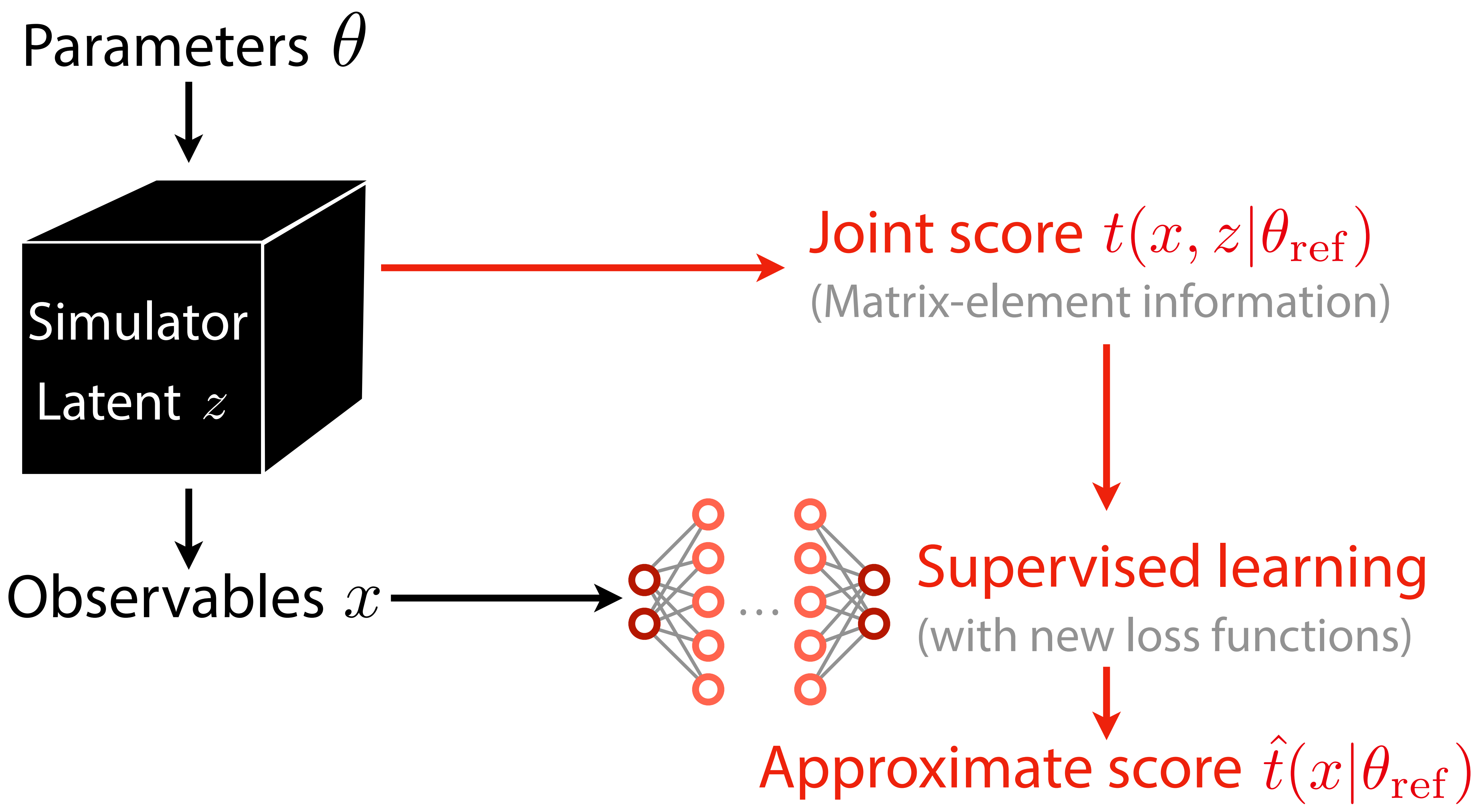



The score itself is intractable. But we can use the same trick as for the likelihood ratio!

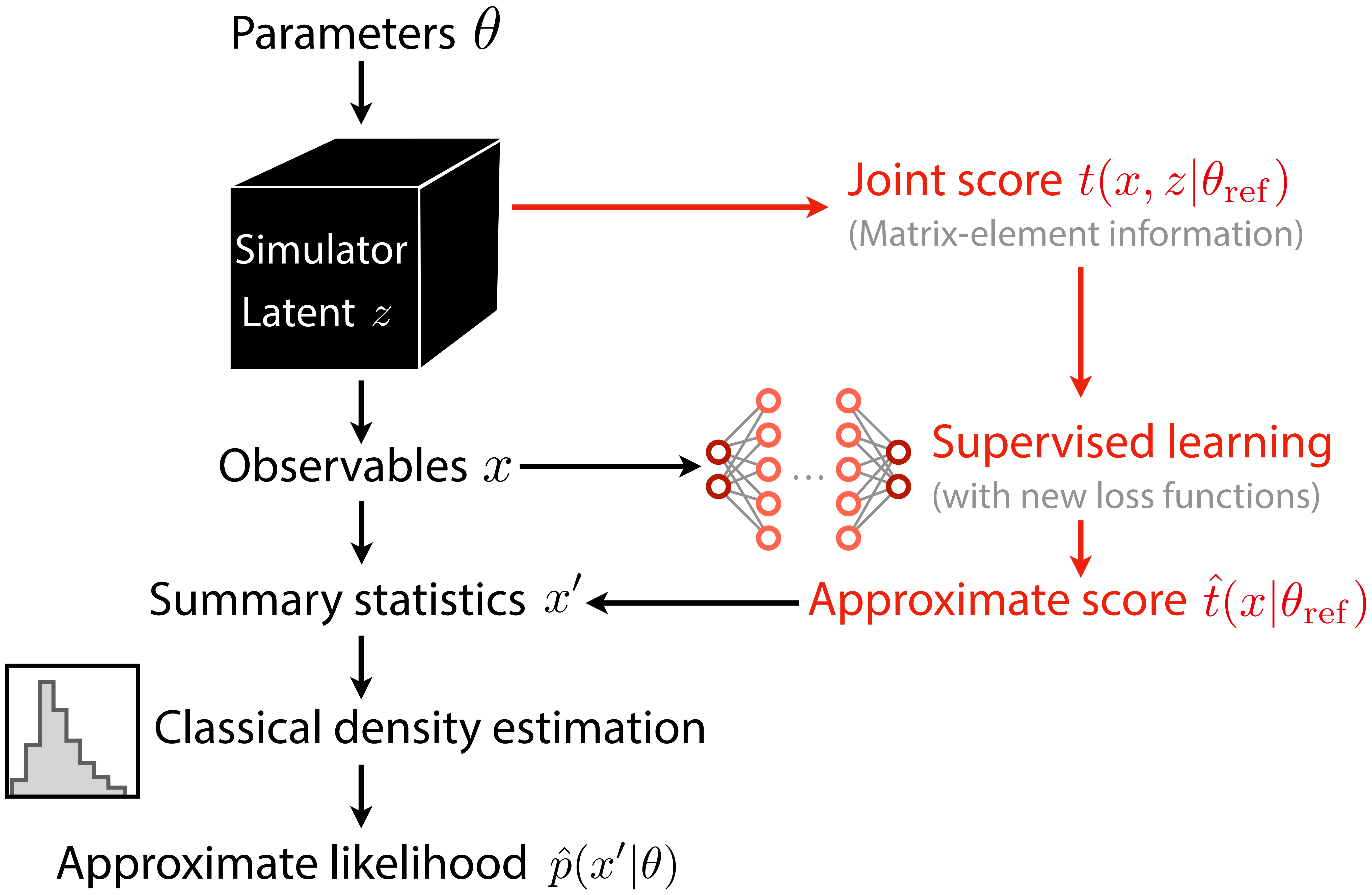
# Neural optimal observables (SALLY)



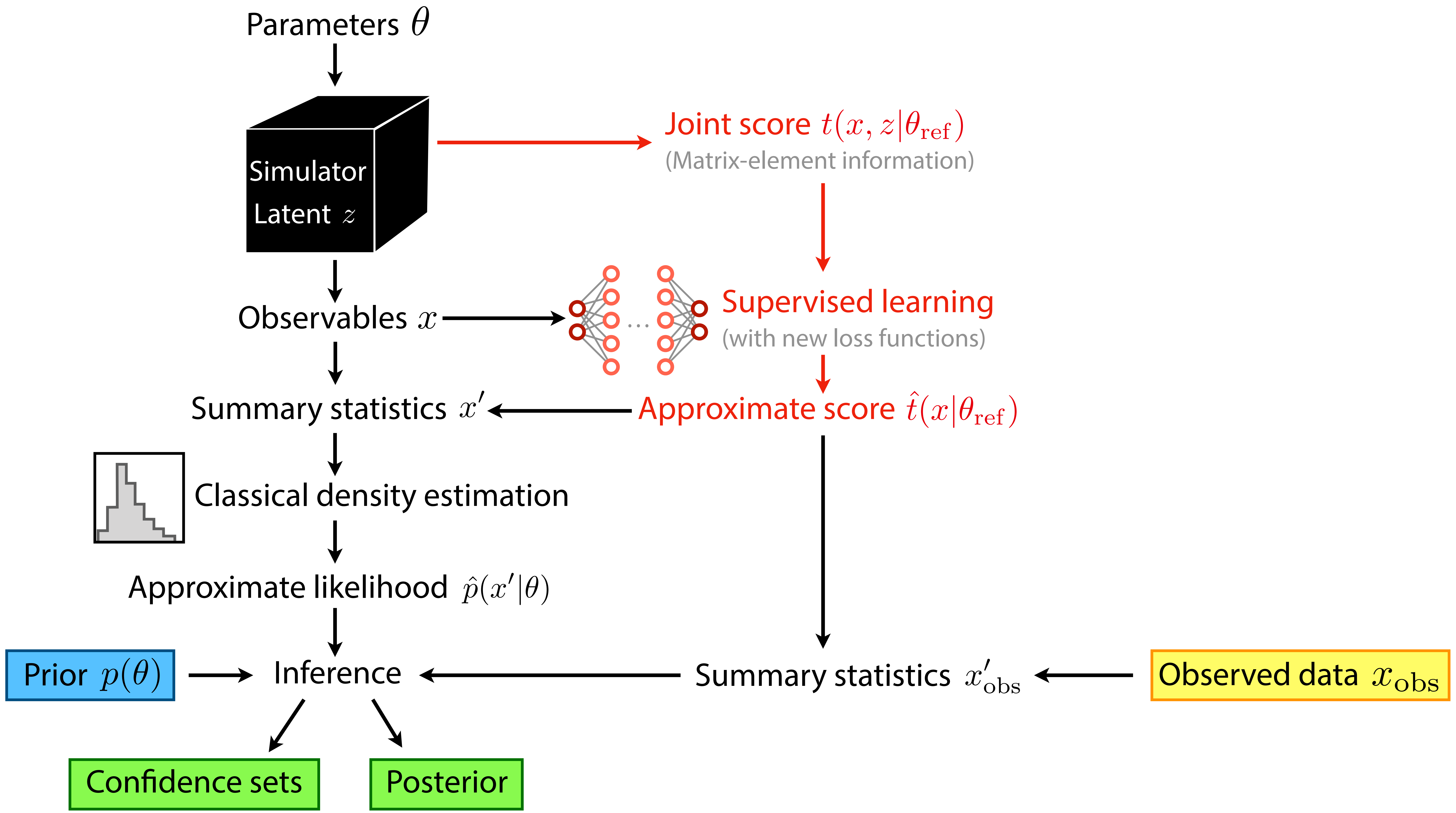
# Neural optimal observables (SALLY)



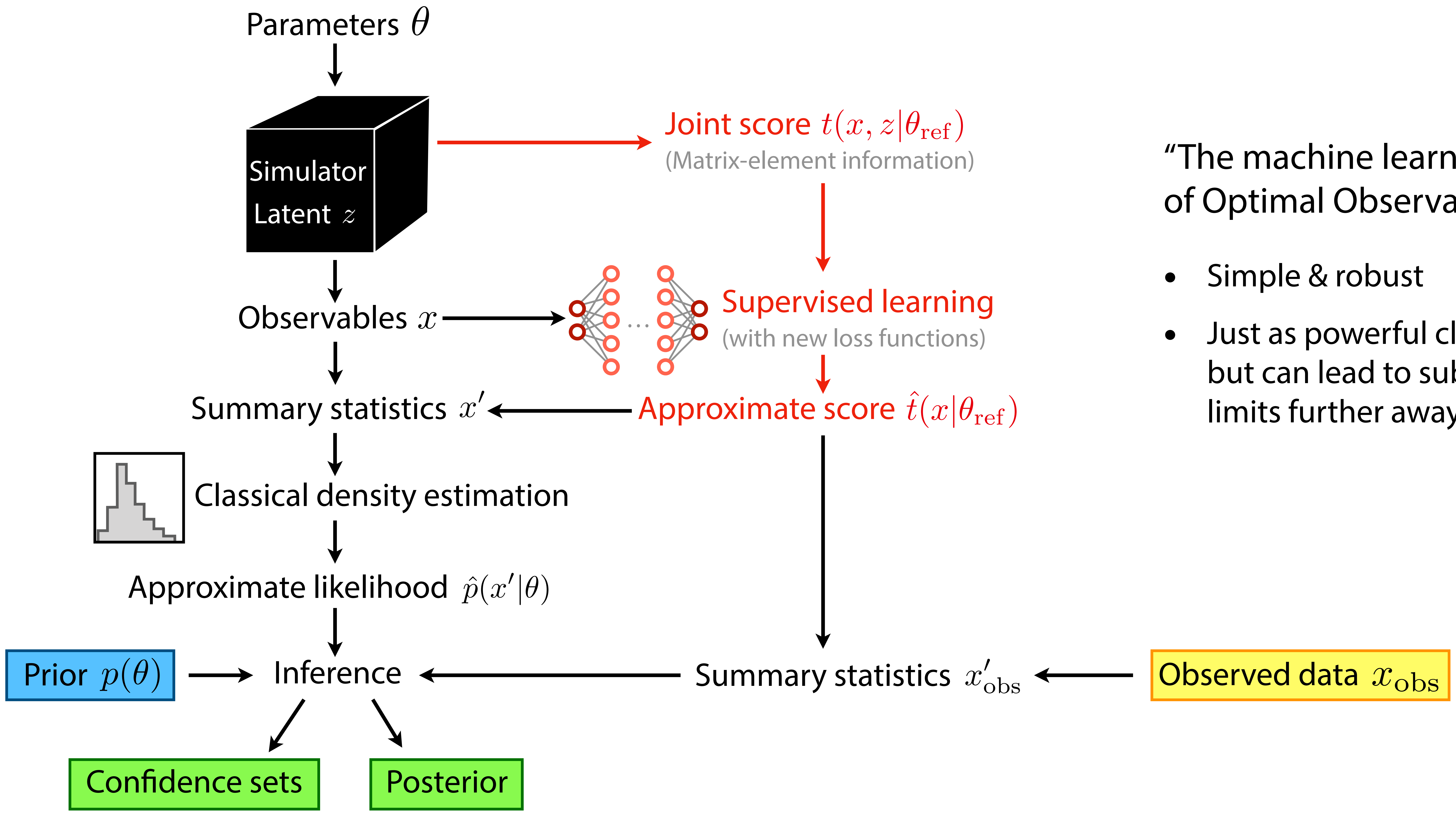
# Neural optimal observables (SALLY)



# Neural optimal observables (SALLY)



# Neural optimal observables (SALLY)



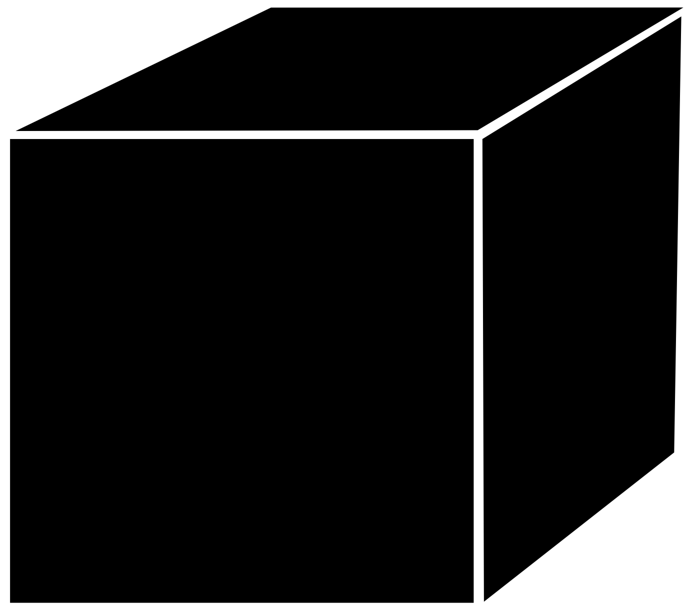

"The machine learning version of Optimal Observables"

- Simple & robust
- Just as powerful close to $\theta_{\text{ref}}$, but can lead to suboptimal limits further away

# An incomplete wrap-up of simulation-based inference methods

| Method | Approximations | Upfront cost | Eval |
| --- | --- | --- | --- |
| Summary statistics: | | | |
| Likelihood for summary stats (standard histograms) | Reduction to summary stats | Fast | Fast |
| Approximate Bayesian Computation | Reduction to summary stats | Depends | Depends |
| Matrix elements: | | | |
| Matrix Element Method | Transfer fns | Fast | Slow |
| Optimal Observables | Transfer fns, optimal only locally | Fast | Slow |
| Neural networks: | | | |
| Neural likelihood | NN | Needs many samples | Fast |
| Neural posterior | NN | Needs many samples | Fast |
| Neural likelihood ratio | NN | Needs many samples | Fast |
| Neural networks + matrix elements: | | | |
| Neural likelihood (ratio) + gold mining (RASCAL etc) | NN | Needs less samples | Fast |
| Neural optimal observables (SALLY) | NN, optimal only locally | Needs less samples | Fast |

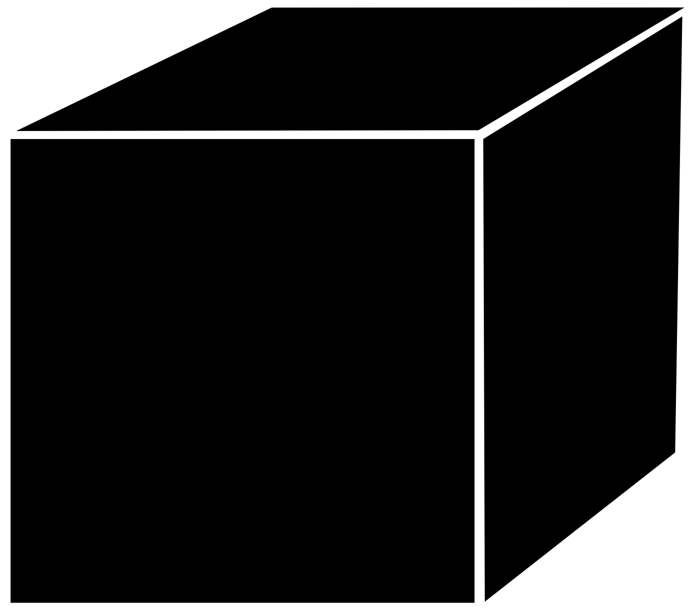
# Systematics



Don't fully trust the simulator?

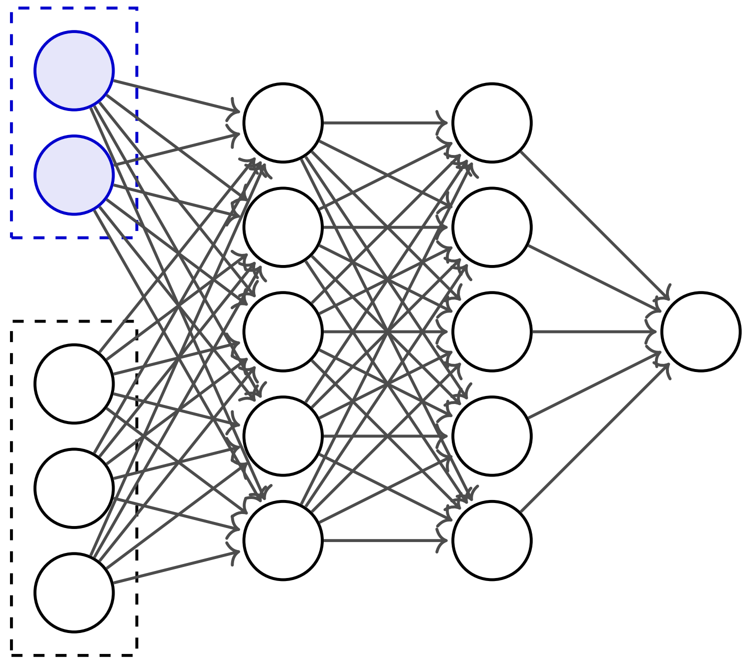
- Nuisance parameters to model systematic uncertainties
- Methods learn dependence both on parameters of interest and nuisance parameters. Then we can construct profile likelihood and "nuisance-hardened" score
  [J. Alsing, B. Wandelt 1903.01473;
  see also P. de Castro, T. Dorigo 1806.04743]
- Alternatively: Robustness to nuisance with adversarial training
  [G. Louppe, M. Kagan, K. Cranmer 1611.01046]

# Systematics



Don't fully trust the simulator?

- Nuisance parameters to model systematic uncertainties
- Methods learn dependence both on parameters of interest and nuisance parameters. Then we can construct profile likelihood and "nuisance-hardened" score
  [J. Alsing, B. Wandelt 1903.01473;
  see also P. de Castro, T. Dorigo 1806.04743]
- Alternatively: Robustness to nuisance with adversarial training
  [G. Louppe, M. Kagan, K. Cranmer 1611.01046]



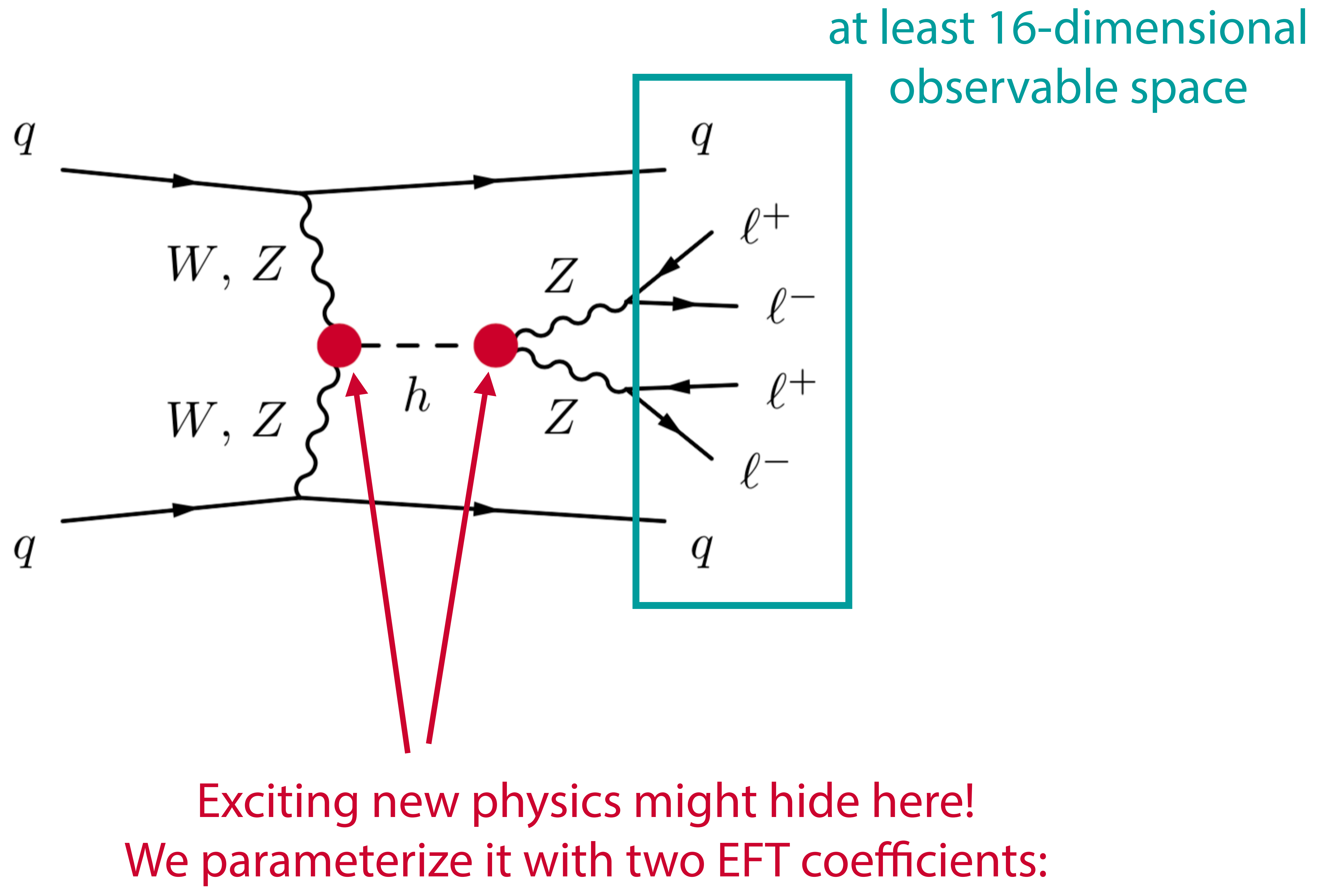
Don't blindly trust the neural network?

- Sanity checks: expectation values, "critic" tests
- Calibration / Neyman construction with toys
  (badly trained network can lead to suboptimal limits, but not to wrong limits)

These techniques let us constrain effective theories more effectively.

# Proof of concept: Higgs production in weak boson fusion

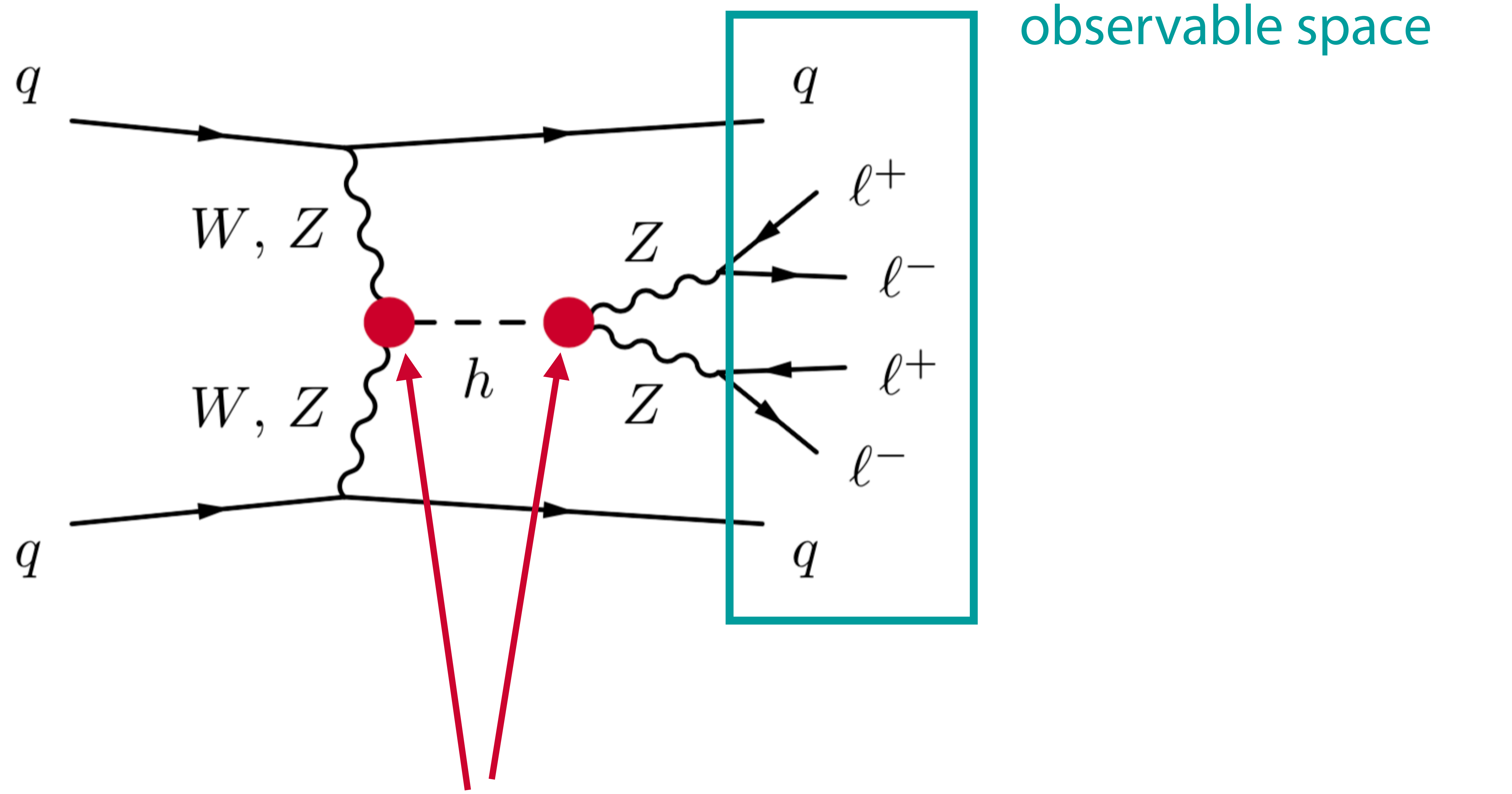
[JB, K. Cranmer, G. Louppe, J. Pavez
1805.00013, 1805.00020]




$$\mathcal{L} = \mathcal{L}_{SM} + \boxed{\frac{f_W}{\Lambda^2}} \underbrace{\frac{ig}{2} (D^\mu \phi)^\dagger \sigma^a D^\nu \phi \, W^a_{\mu\nu}}_{\mathcal{O}_W} - \boxed{\frac{f_{WW}}{\Lambda^2}} \underbrace{\frac{g^2}{4} (\phi^\dagger \phi) \, W^a_{\mu\nu} W^{\mu\nu\, a}}_{\mathcal{O}_{WW}}$$

# Proof of concept: Higgs production in weak boson fusion

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




Exciting new physics might hide here!
We parameterize it with two EFT coefficients:

$$\mathcal{L} = \mathcal{L}_{\rm SM} + \frac{f_W}{\Lambda^2} \underbrace{\frac{ig}{2}\,(D^\mu\phi)^\dagger\,\sigma^a\,D^\nu\phi\,W^a_{\mu\nu}}_{\mathcal{O}_W} - \frac{f_{WW}}{\Lambda^2} \underbrace{\frac{g^2}{4}\,(\phi^\dagger\phi)\,W^a_{\mu\nu}\,W^{\mu\nu\,a}}_{\mathcal{O}_{WW}}$$

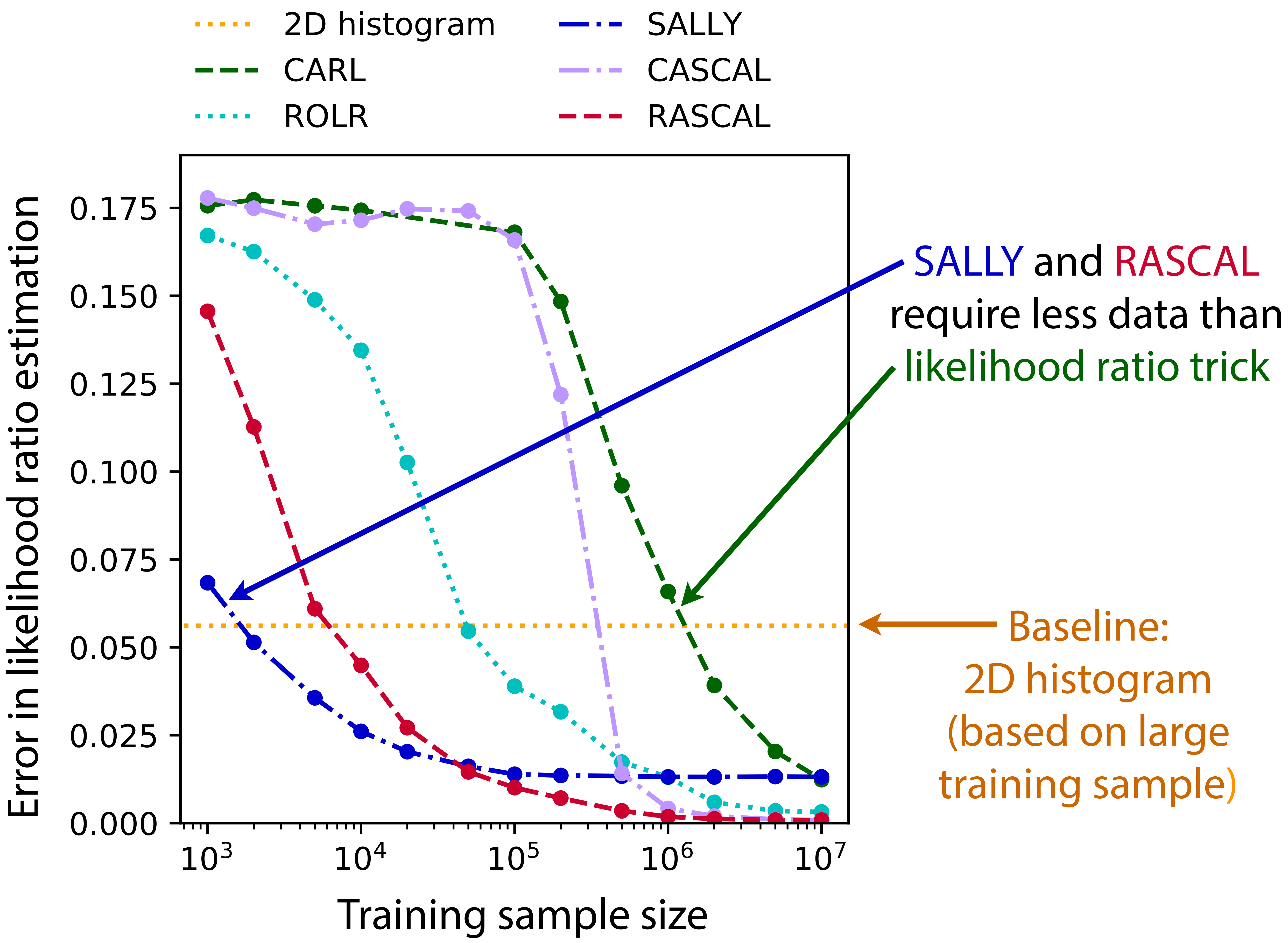
Goal: constrain the two EFT parameters

- new inference methods
- baseline: 2d histogram analysis of jet momenta & angular correlations

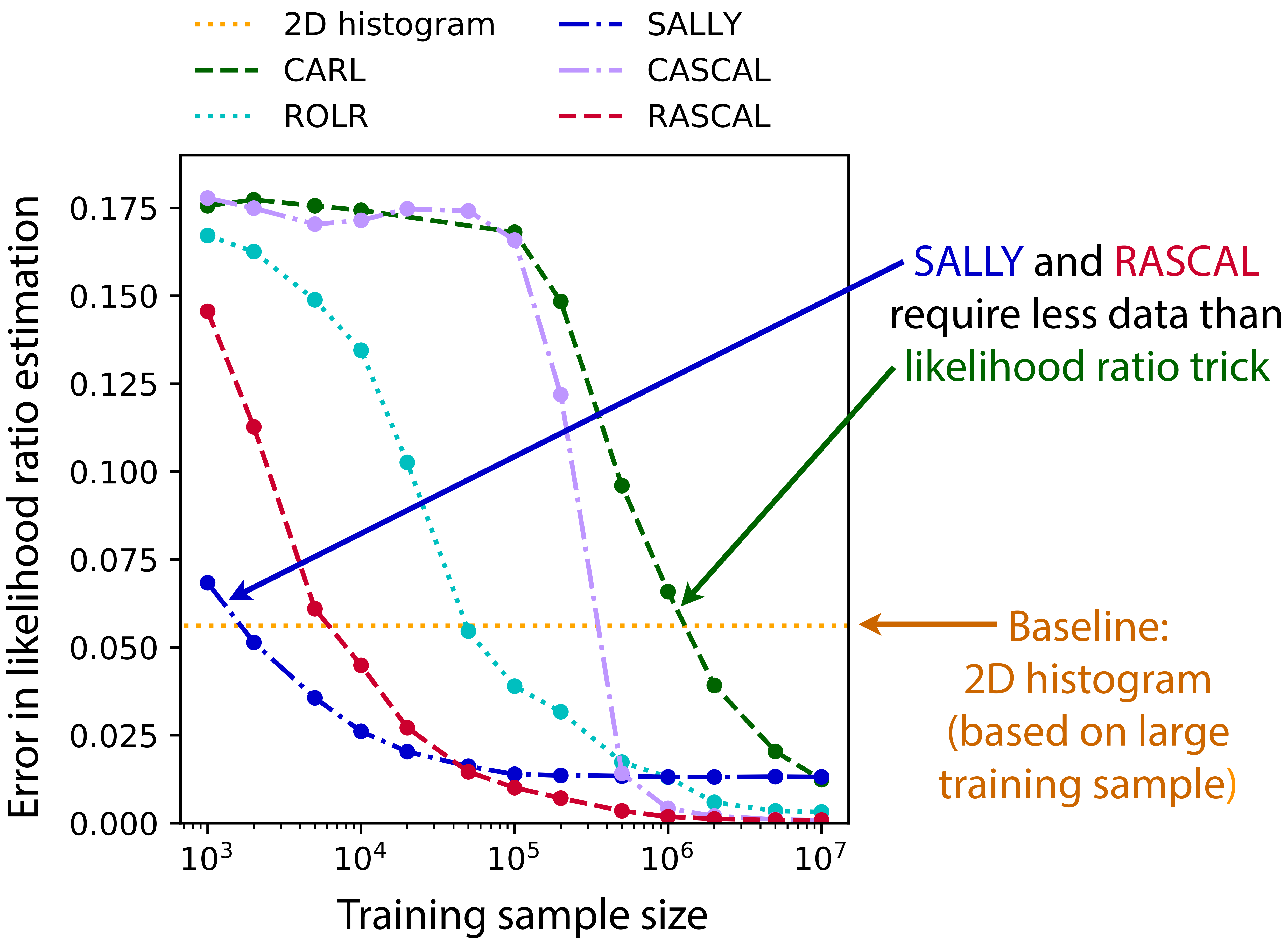
Two scenarios:

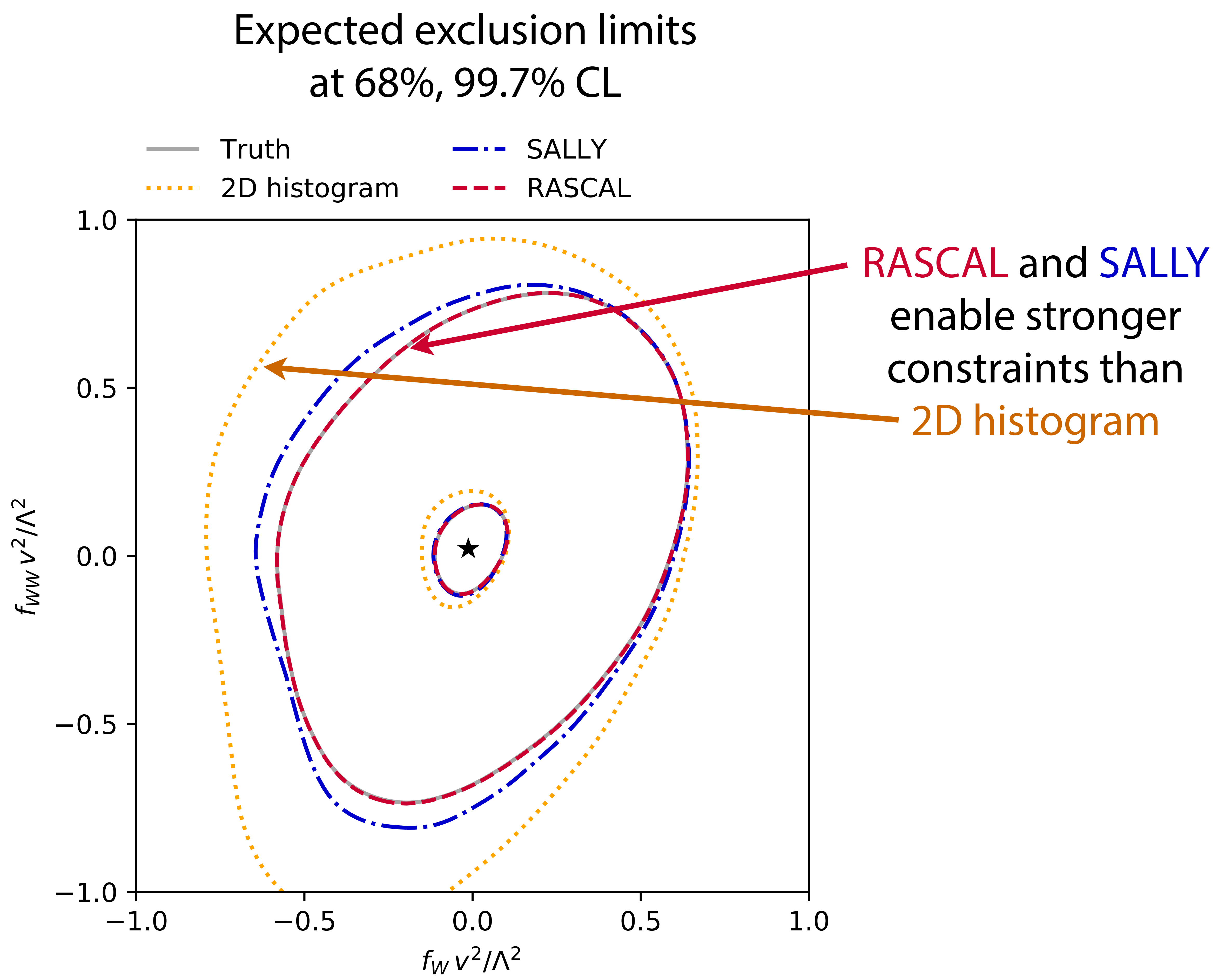
- Simplified setup in which we can compare to true likelihood
- “Realistic” simulation with approximate detector effects

# Proof of concept: Stronger constraints with less training data

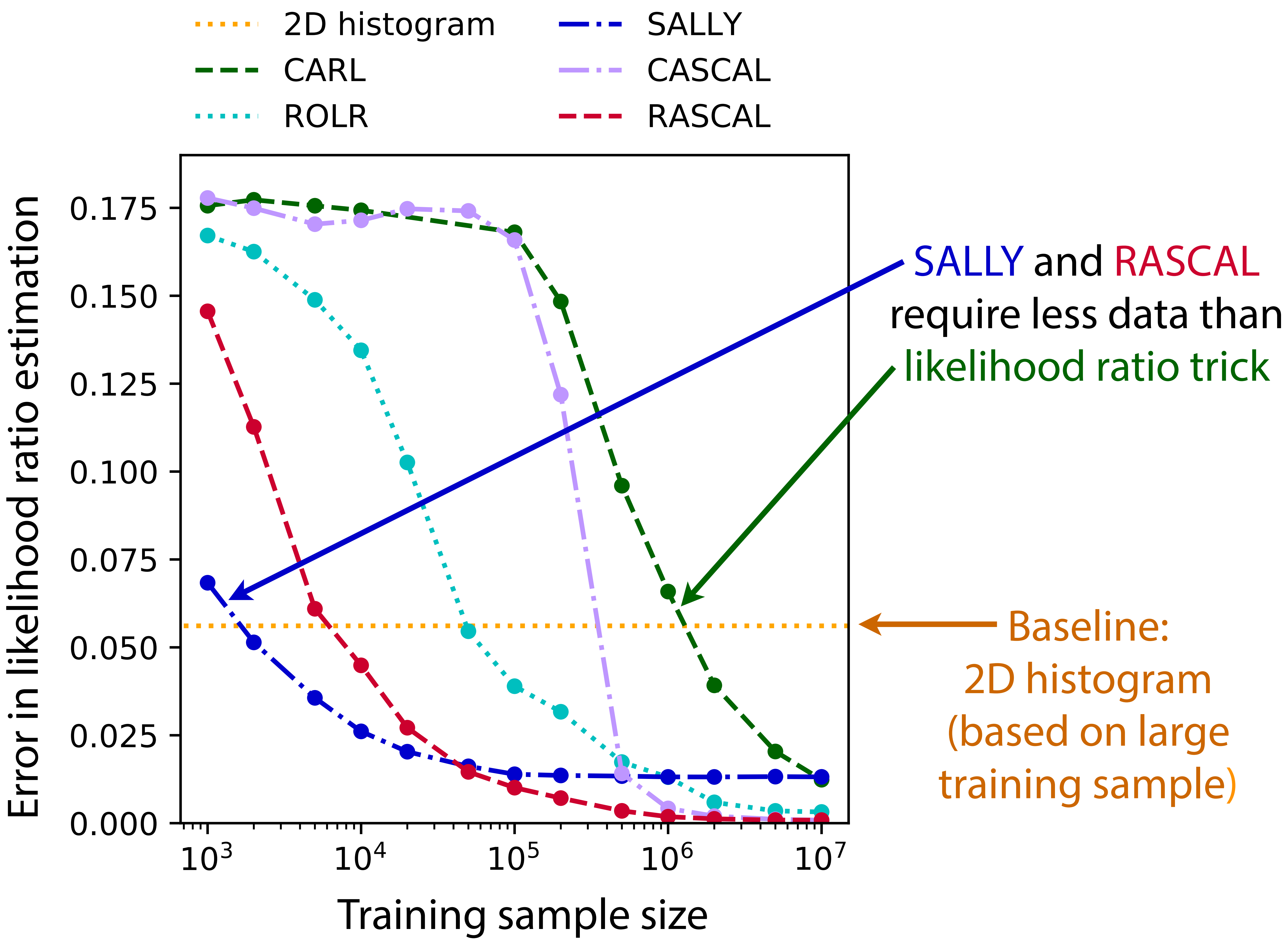


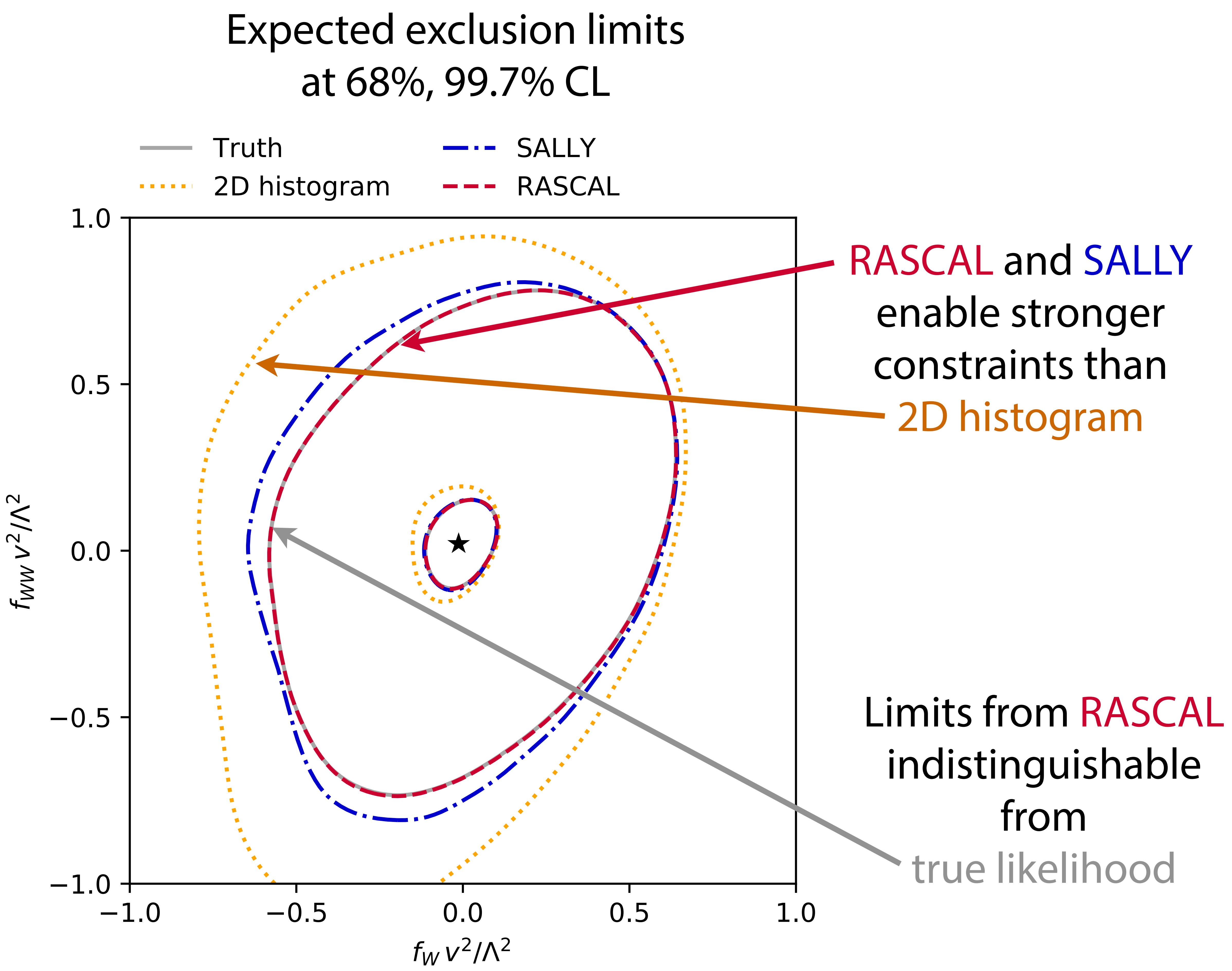
# Proof of concept: Stronger constraints with less training data



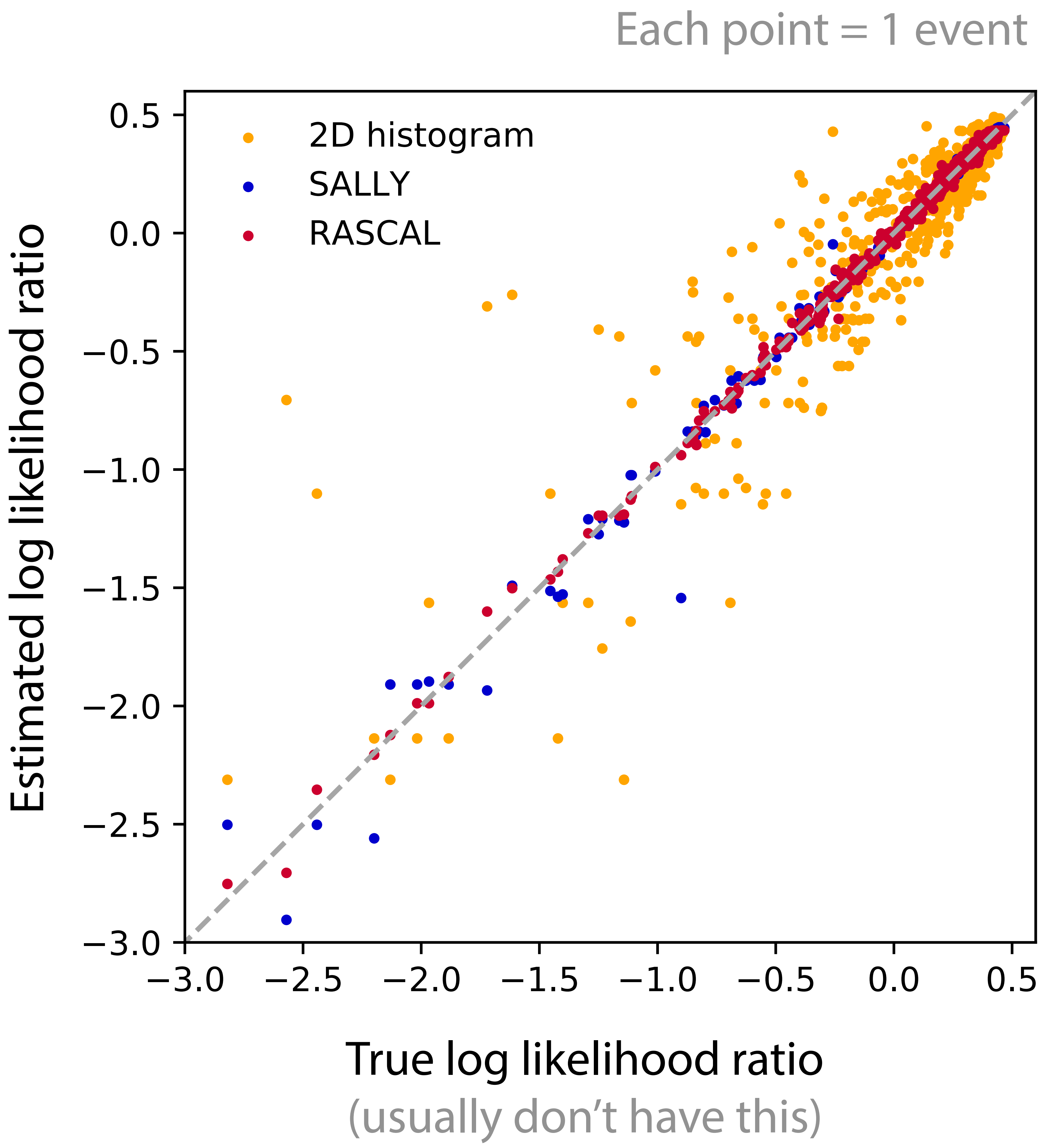



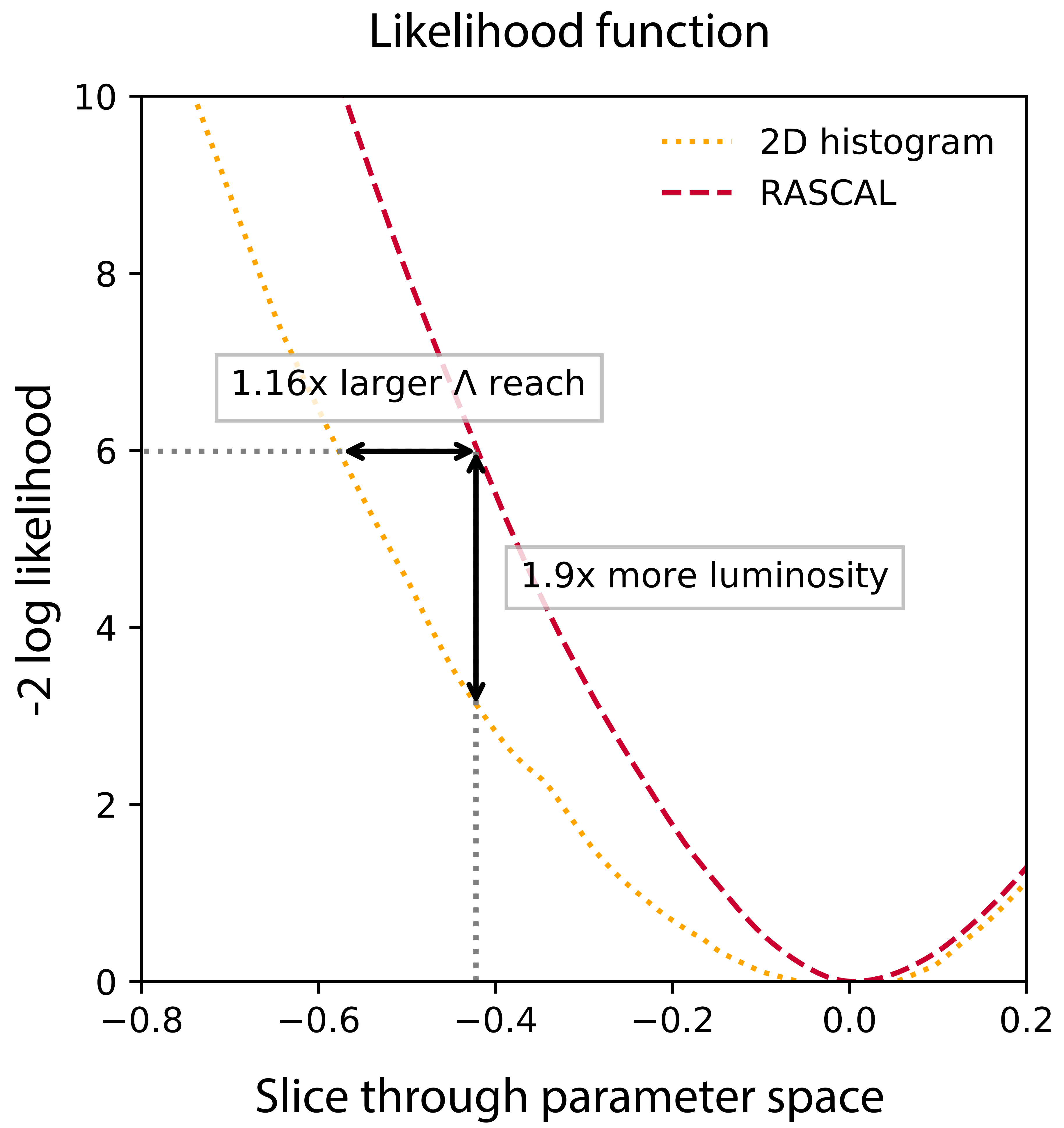
# Proof of concept: Stronger constraints with less training data



# Proof of concept: 16% greater reach (~90% more data)



# Constraining operators in ttH effectively

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

- Pheno-level analysis of

  $$pp \to t\bar{t}\,h \to (b\ell^+)\,(\bar{b}\ell^-)\,(\gamma\gamma)\,E_T^{\text{miss}}$$

  with MadGraph + Pythia + Delphes

- Inference on three EFT operators:

  $$\mathcal{O}_u = -\frac{1}{v^2}(H^\dagger H)(H^\dagger \bar{Q}_L)u_R\,,\ \mathcal{O}_G = \frac{g_s^2}{m_W^2}(H^\dagger H)G^a_{\mu\nu}G_a^{\mu\nu}\,,$$

  $$\mathcal{O}_{uG} = -\frac{4g_s}{m_W^2}y_u(H^\dagger \bar{Q}_L)\gamma^{\mu\nu}T_a u_R G^a_{\mu\nu}$$

# Constraining operators in ttH effectively

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

- Pheno-level analysis of

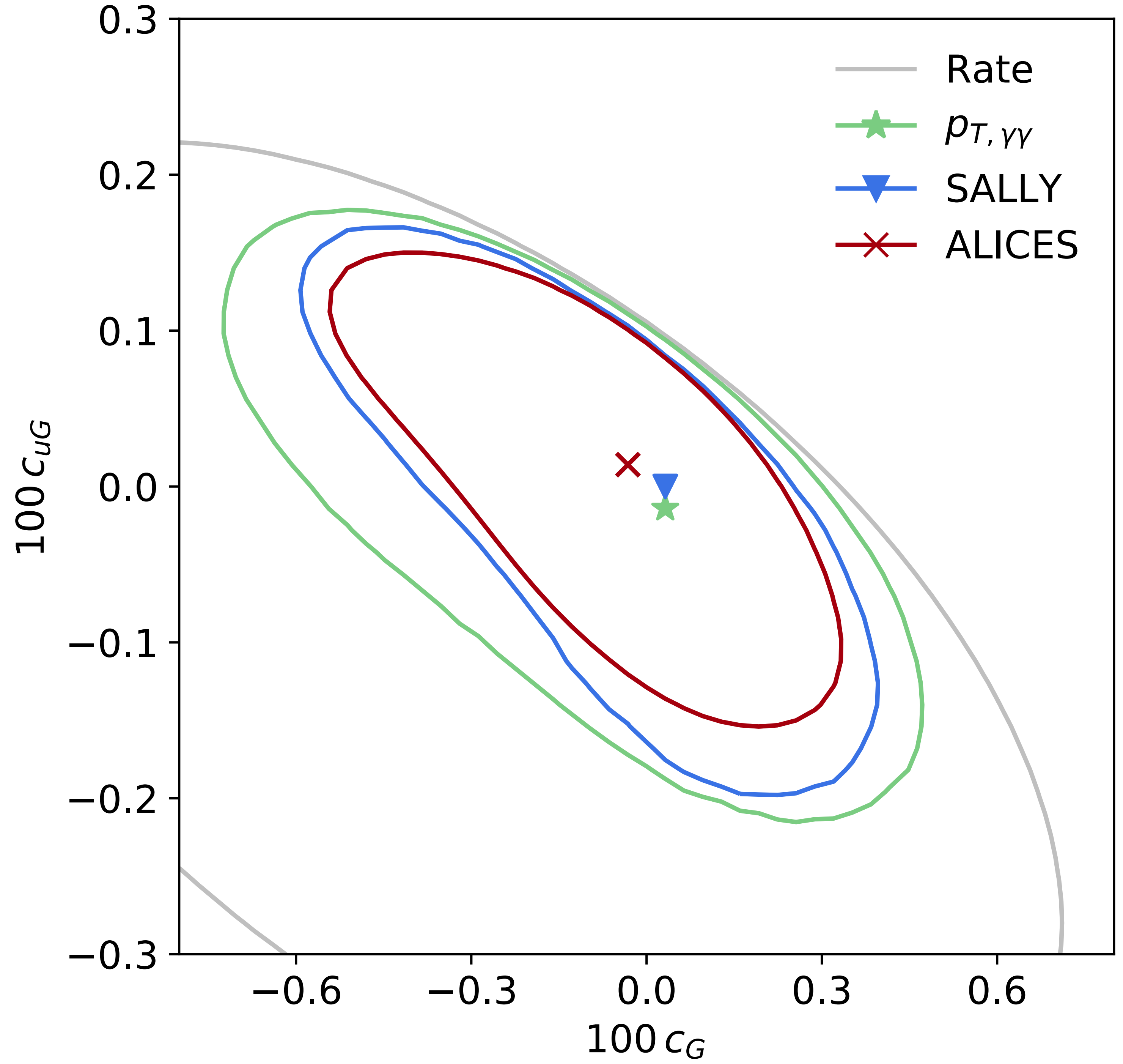
$$pp \to t\bar{t}\,h \to (b\ell^+)\,(\bar{b}\ell^-)\,(\gamma\gamma)\,E_T^{\text{miss}}$$

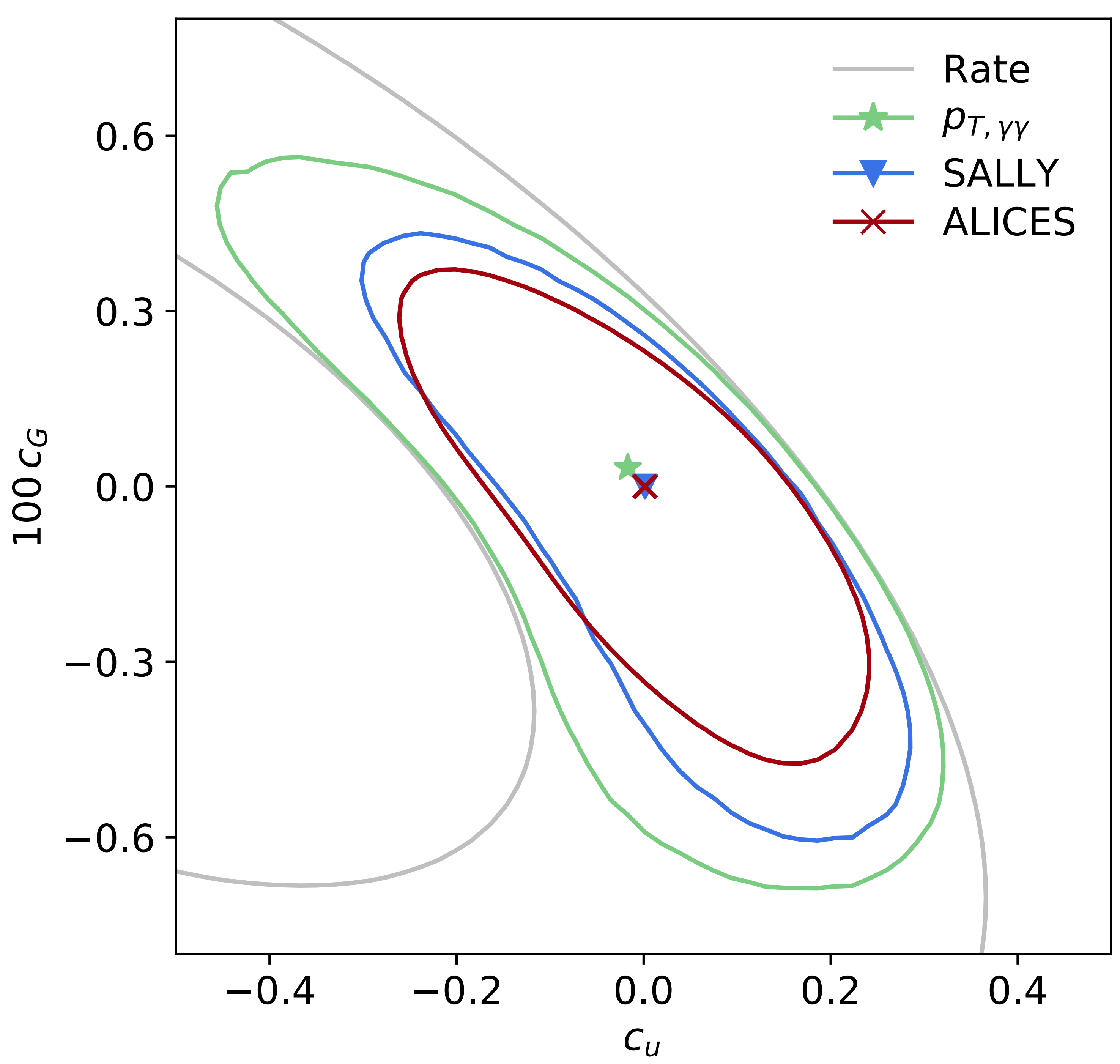
with MadGraph + Pythia + Delphes

- Inference on three EFT operators:

$$\mathcal{O}_u = -\frac{1}{v^2}(H^\dagger H)(H^\dagger \bar{Q}_L)u_R\,,\;\; \mathcal{O}_G = \frac{g_s^2}{m_W^2}(H^\dagger H)G^a_{\mu\nu}G_a^{\mu\nu}\,,$$
$$\mathcal{O}_{uG} = -\frac{4g_s}{m_W^2}y_u(H^\dagger \bar{Q}_L)\gamma^{\mu\nu}T_a u_R G^a_{\mu\nu}$$

- New inference techniques improve expected HL-LHC limits compared to histogram baseline:

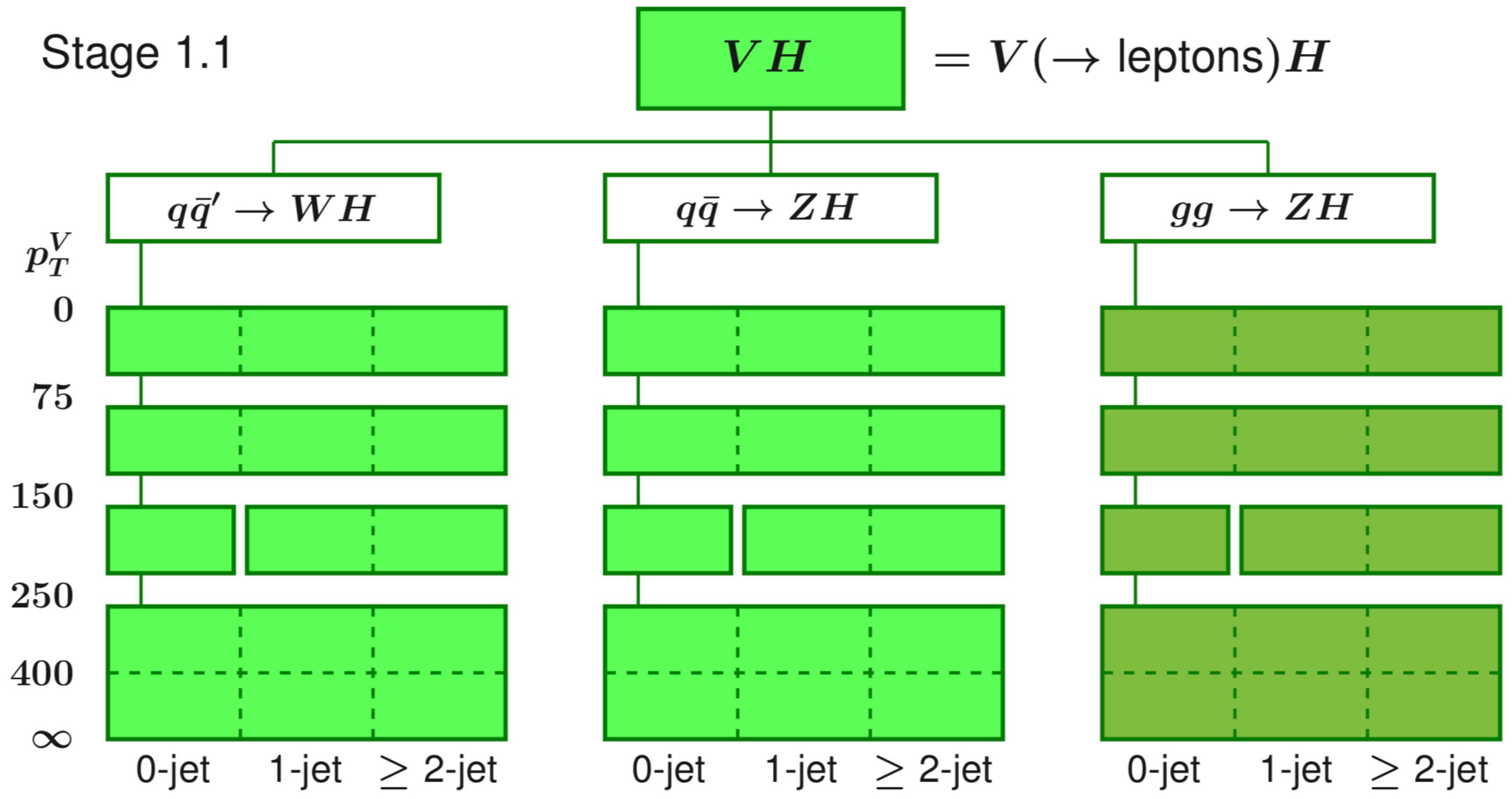


# Benchmarking STXS in WH

[JB, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

- Simplified Template Cross-Sections (STXS) define observable bins that are supposed to capture as much information on NP as possible
  [N. Berger et al. 1906.02754; HXSWG YR4]




- Let's check! How much information on

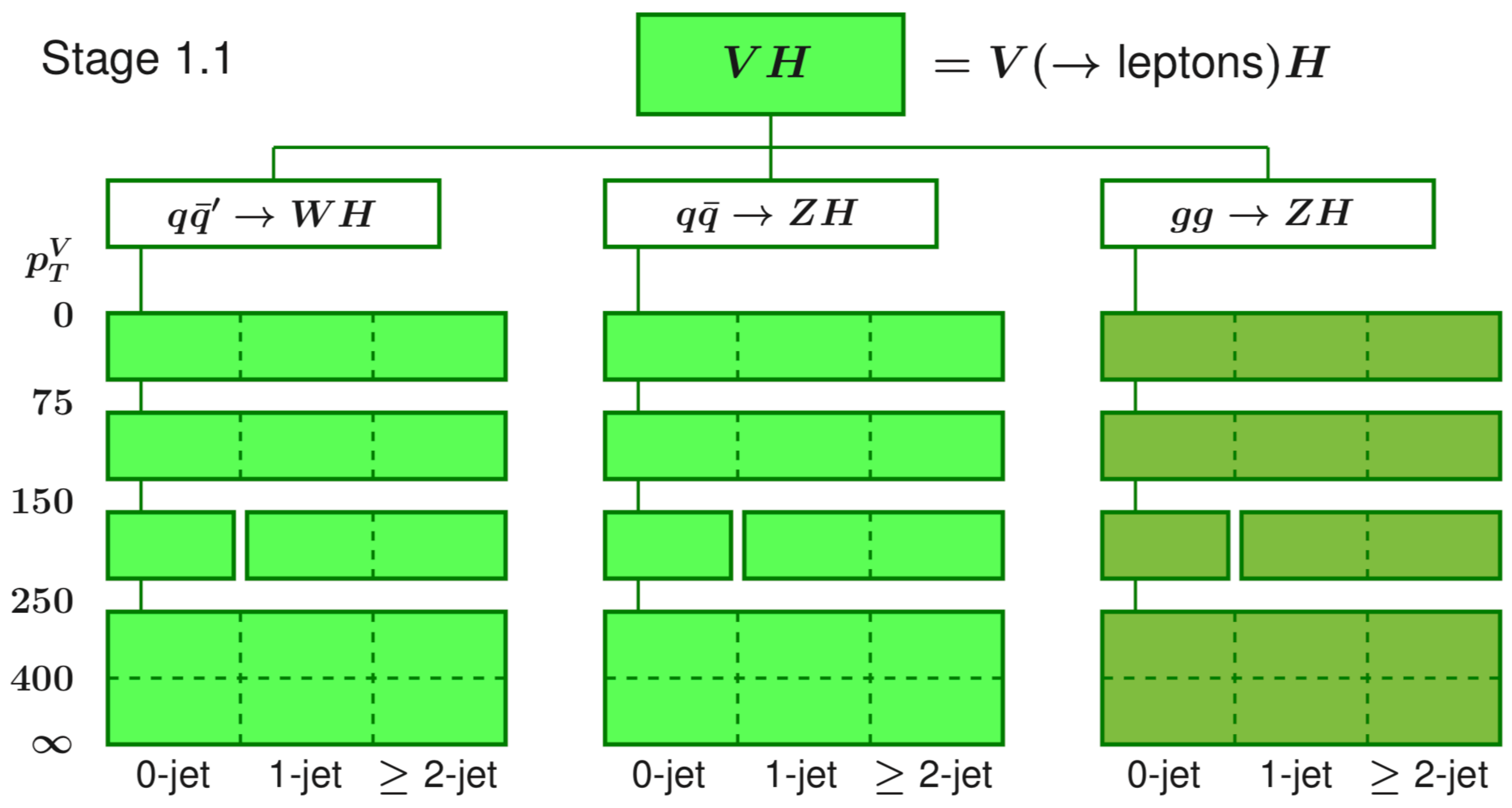
$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\Box} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger \phi)\Box(\phi^\dagger \phi) - \frac{1}{4}(\phi^\dagger D^\mu \phi)^*(\phi^\dagger D_\mu \phi)$$
$$\mathcal{O}_{HW} = \phi^\dagger \phi W^a_{\mu\nu} W^{\mu\nu a}$$
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i \overleftrightarrow{D}^a_\mu \phi)(\overline{Q}_L \sigma^a \gamma^\mu Q_L) \,,$$

can we extract from $pp \to WH \to \ell\nu \, b\bar{b}$ ?

# Benchmarking STXS in WH

[JB, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

- Simplified Template Cross-Sections (STXS) define observable bins that are supposed to capture as much information on NP as possible
  [N. Berger et al. 1906.02754; HXSWG YR4]

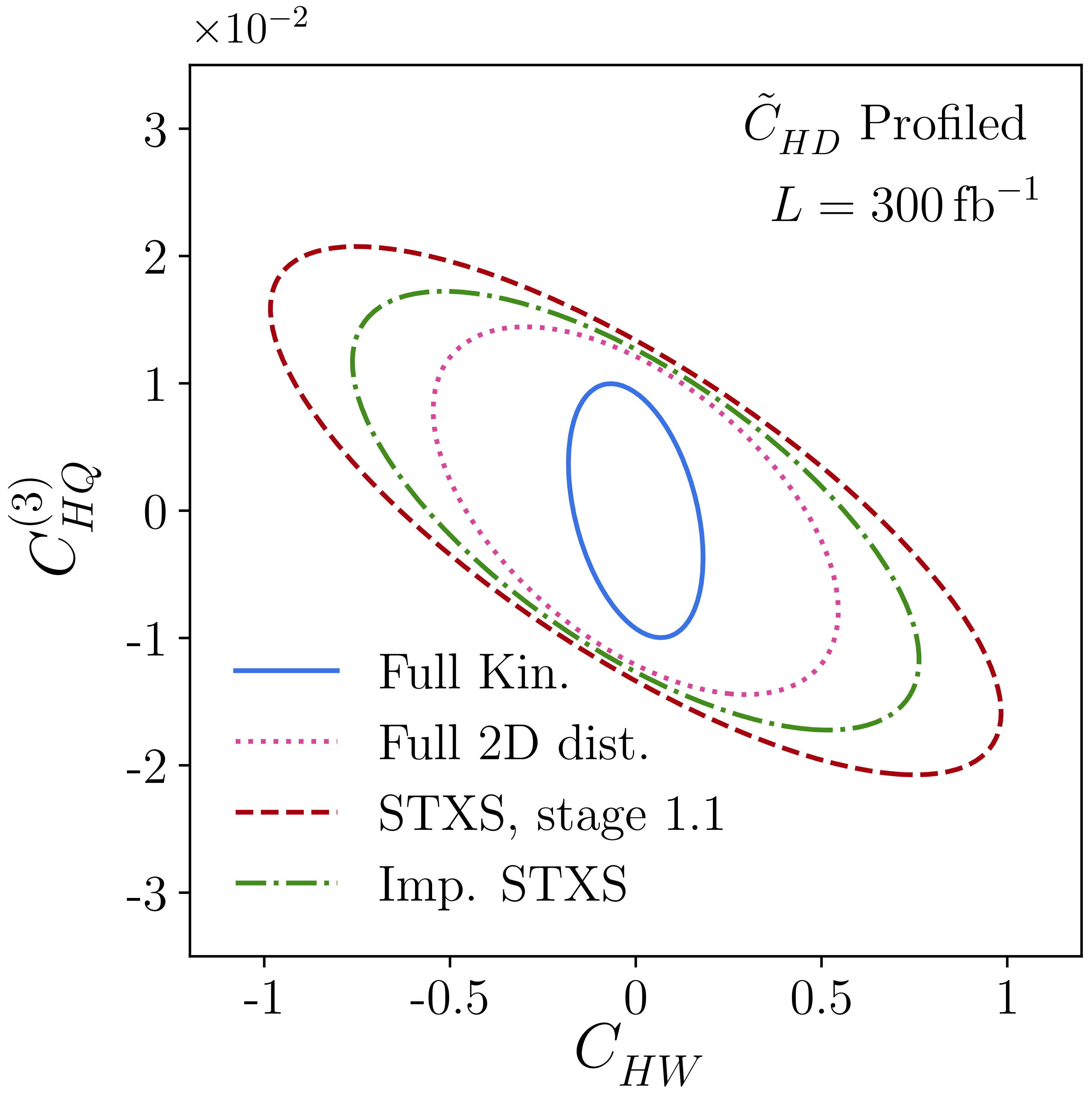



- Let's check! How much information on

$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\Box} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger\phi)\Box(\phi^\dagger\phi) - \frac{1}{4}(\phi^\dagger D^\mu\phi)^*(\phi^\dagger D_\mu\phi)$$
$$\mathcal{O}_{HW} = \phi^\dagger\phi W^a_{\mu\nu}W^{\mu\nu a}$$
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu\phi)(\overline{Q}_L\sigma^a\gamma^\mu Q_L)\ ,$$

can we extract from $pp \to WH \to \ell\nu\, b\bar{b}$ ?

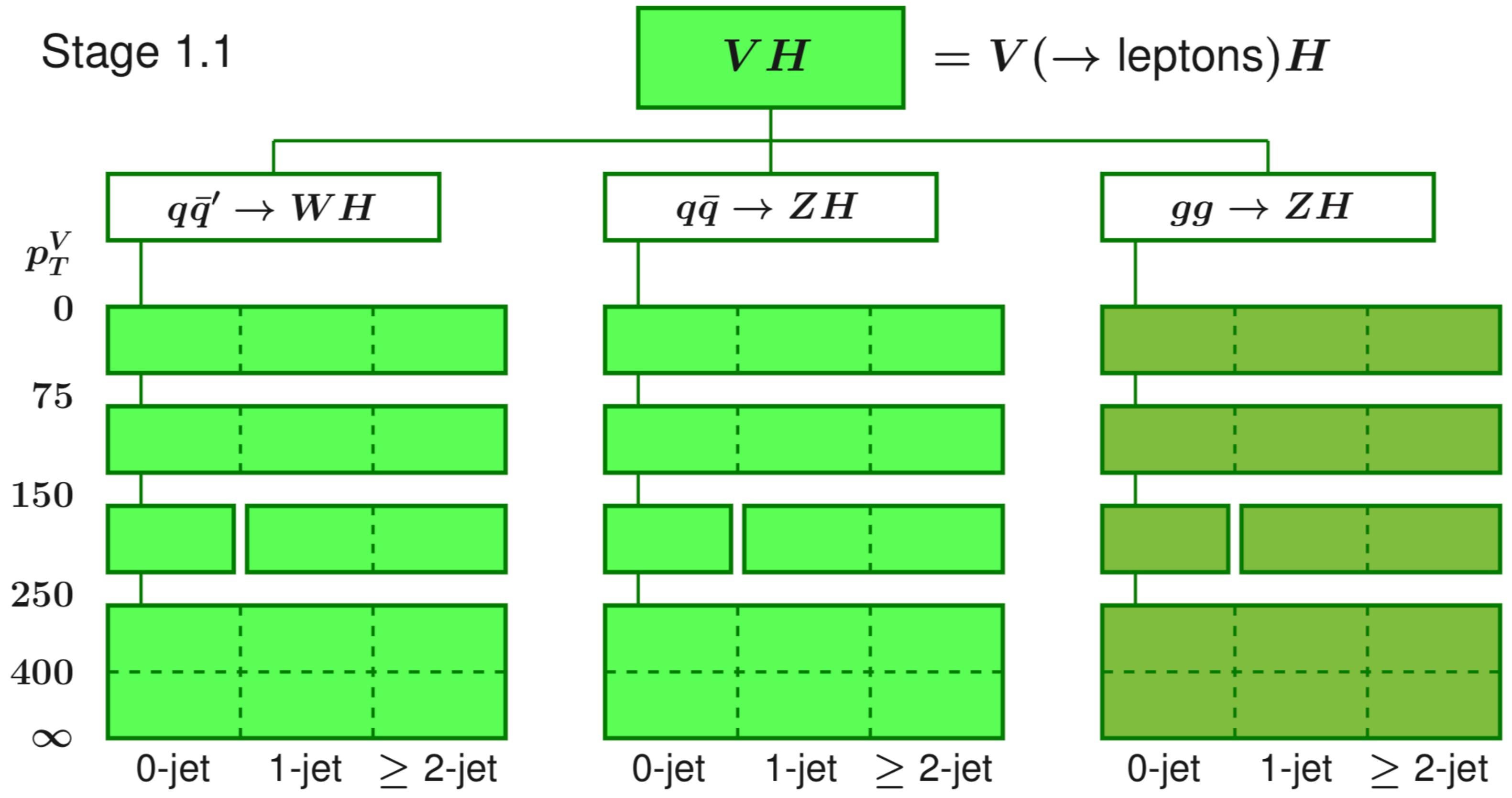
- Results: STXS are indeed sensitive to operators, adding a few more bins improve them, but a multivariate analysis is still stronger



# Benchmarking STXS in WH

[JB, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

- Simplified Template Cross-Sections (STXS) define observable bins that are supposed to capture as much information on NP as possible
  [N. Berger et al. 1906.02754; HXSWG YR4]

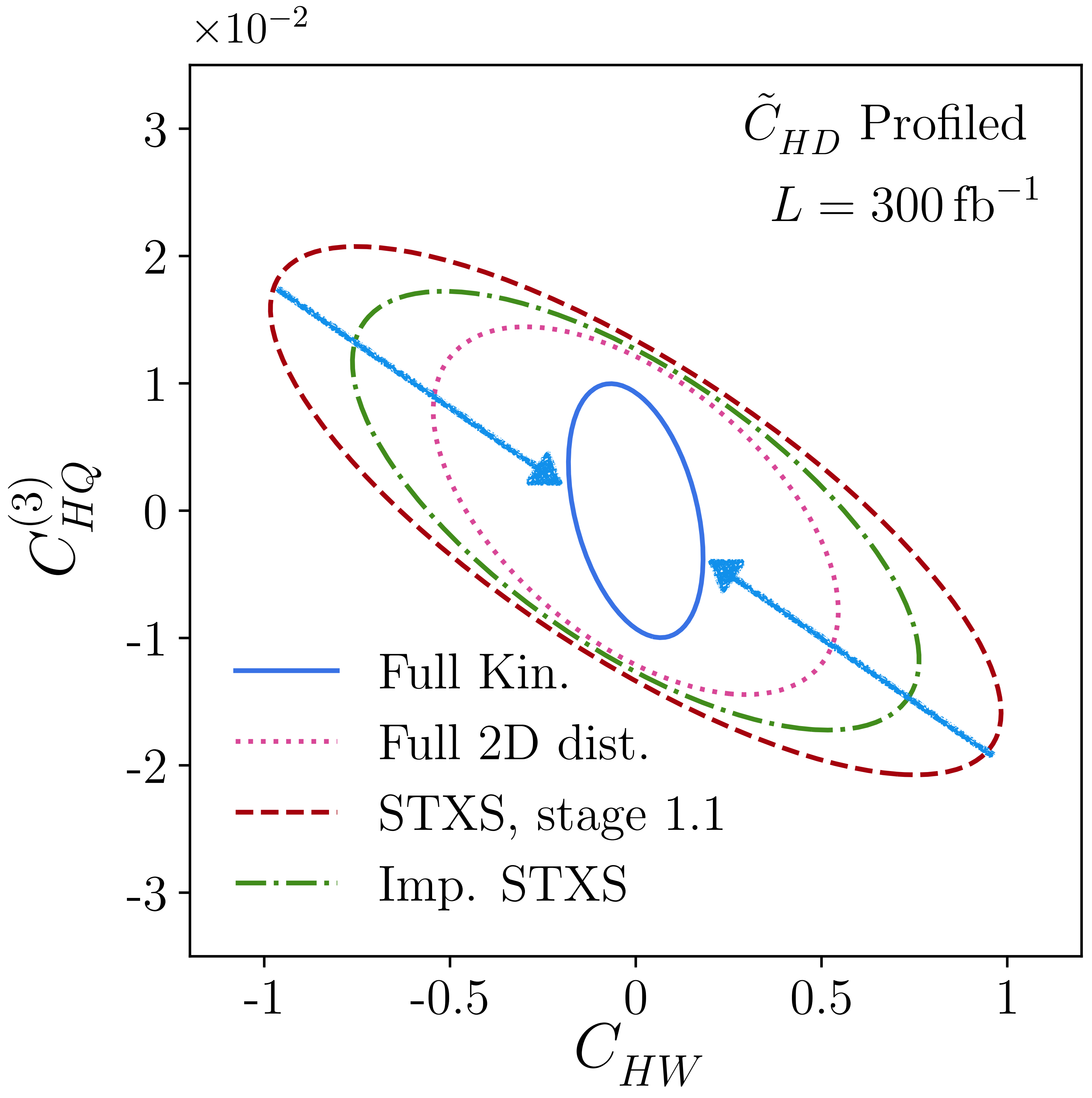



- Let's check! How much information on

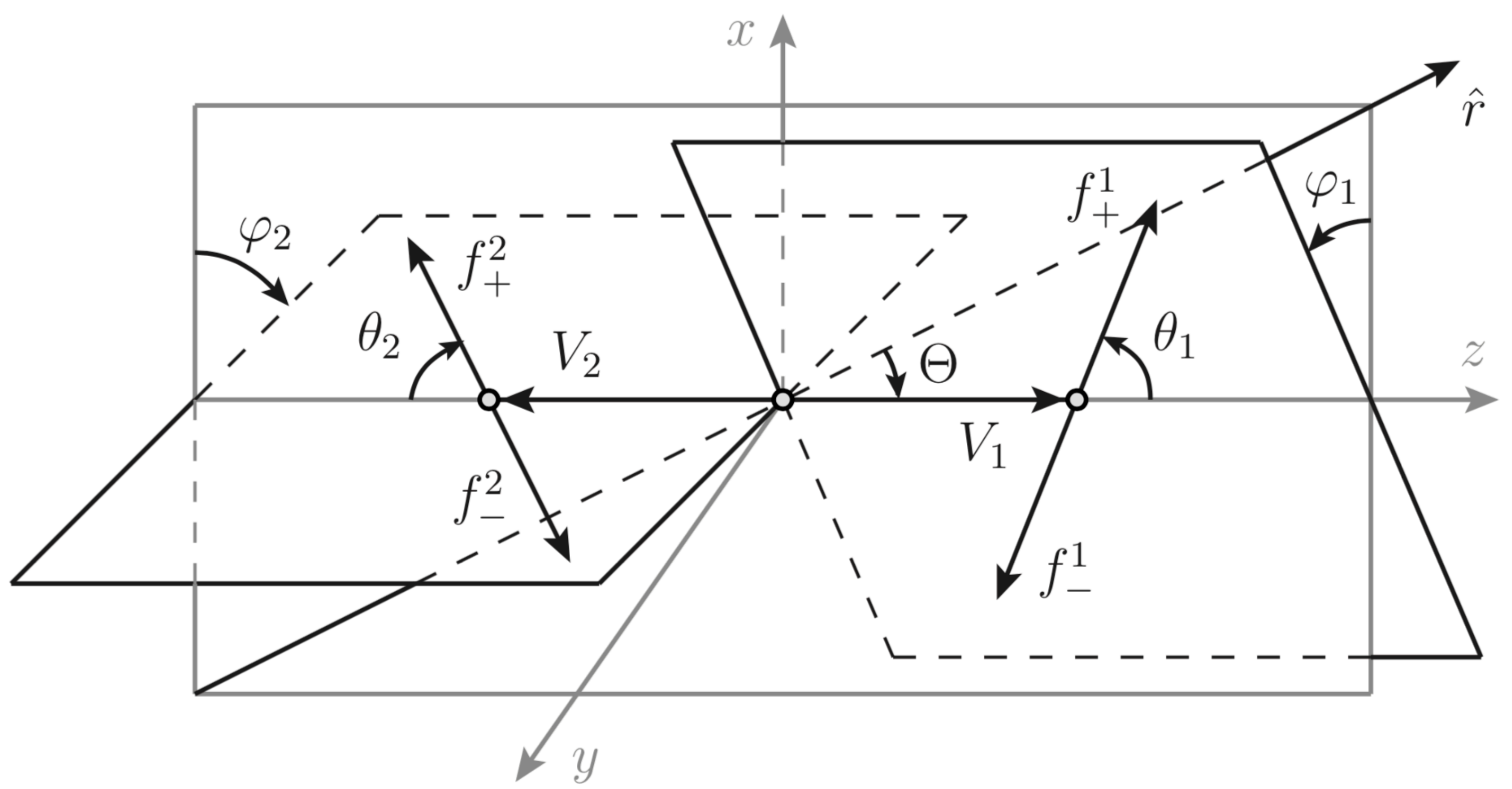
$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\Box} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger\phi)\Box(\phi^\dagger\phi) - \frac{1}{4}(\phi^\dagger D^\mu\phi)^*(\phi^\dagger D_\mu\phi)$$
$$\mathcal{O}_{HW} = \phi^\dagger\phi W^a_{\mu\nu}W^{\mu\nu a}$$
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu\phi)(\overline{Q}_L\sigma^a\gamma^\mu Q_L)\ ,$$

can we extract from $pp \to WH \to \ell\nu\, b\bar{b}$ ?

- Results: STXS are indeed sensitive to operators, adding a few more bins improve them, but a multivariate analysis is still stronger



# Diboson production




- In inclusive observables, the interference between SM and new physics amplitudes vanishes

  ⇒ Reduced sensitivity to new physics

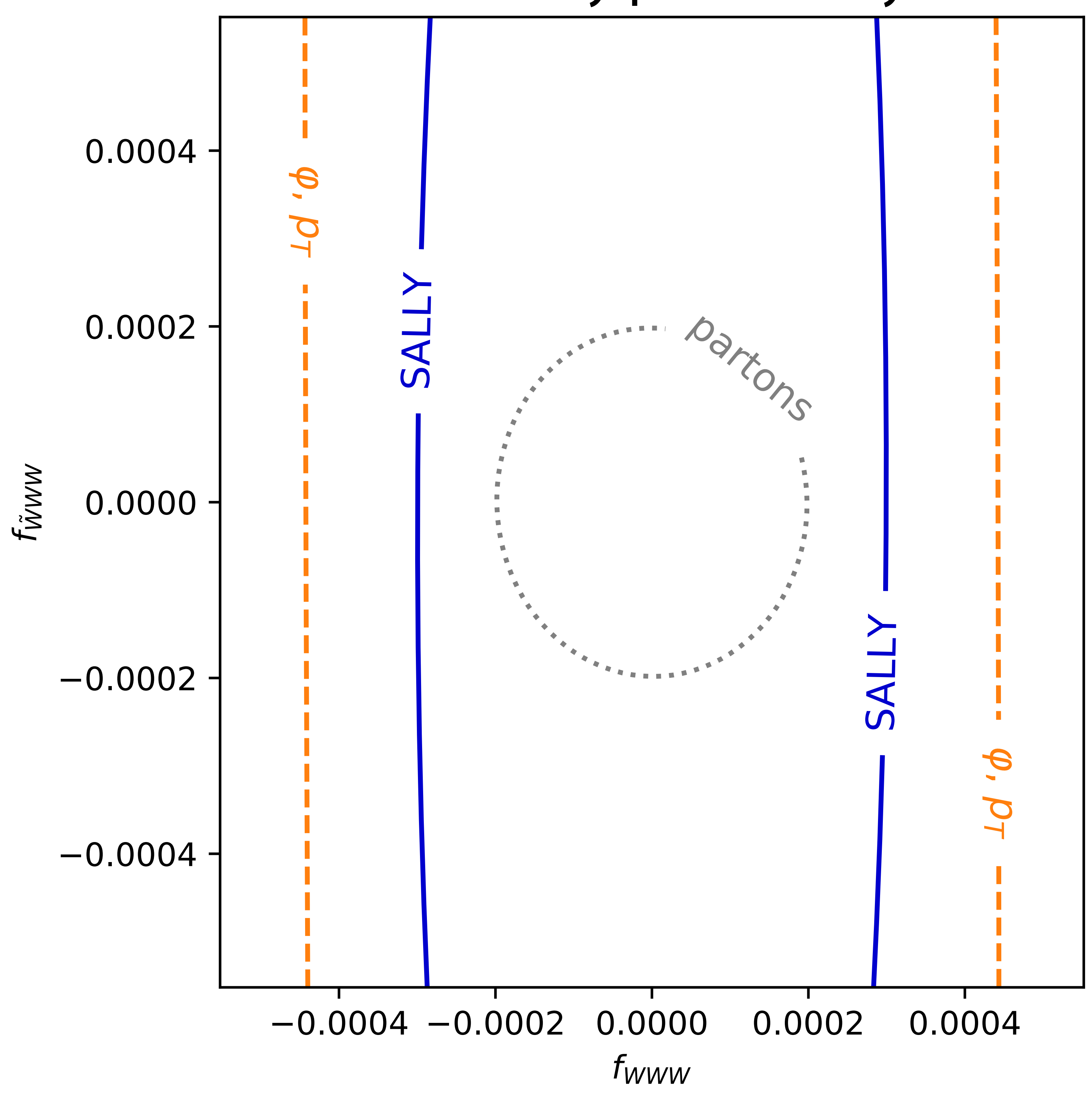
- "Diboson interference resurrection": an angular variable $\varphi$ can be constructed to be sensitive to this interference
  [G. Panico, F. Riva, A. Wulzer 1708.07823;
  A. Azatov, D. Barducci, E. Venturini 1901.04821]

- We test the ML approach in EFT measurements in $W\gamma \to \ell\nu\ \gamma$
  [JB, K. Cranmer, M. Farina, F. Kling, D. Pappadopulo, J. Ruderman in progress]

  **New**: WZ→ ℓℓ ℓν by Chen, Glioti, Panico, Wulzer arXiv:2007.10356

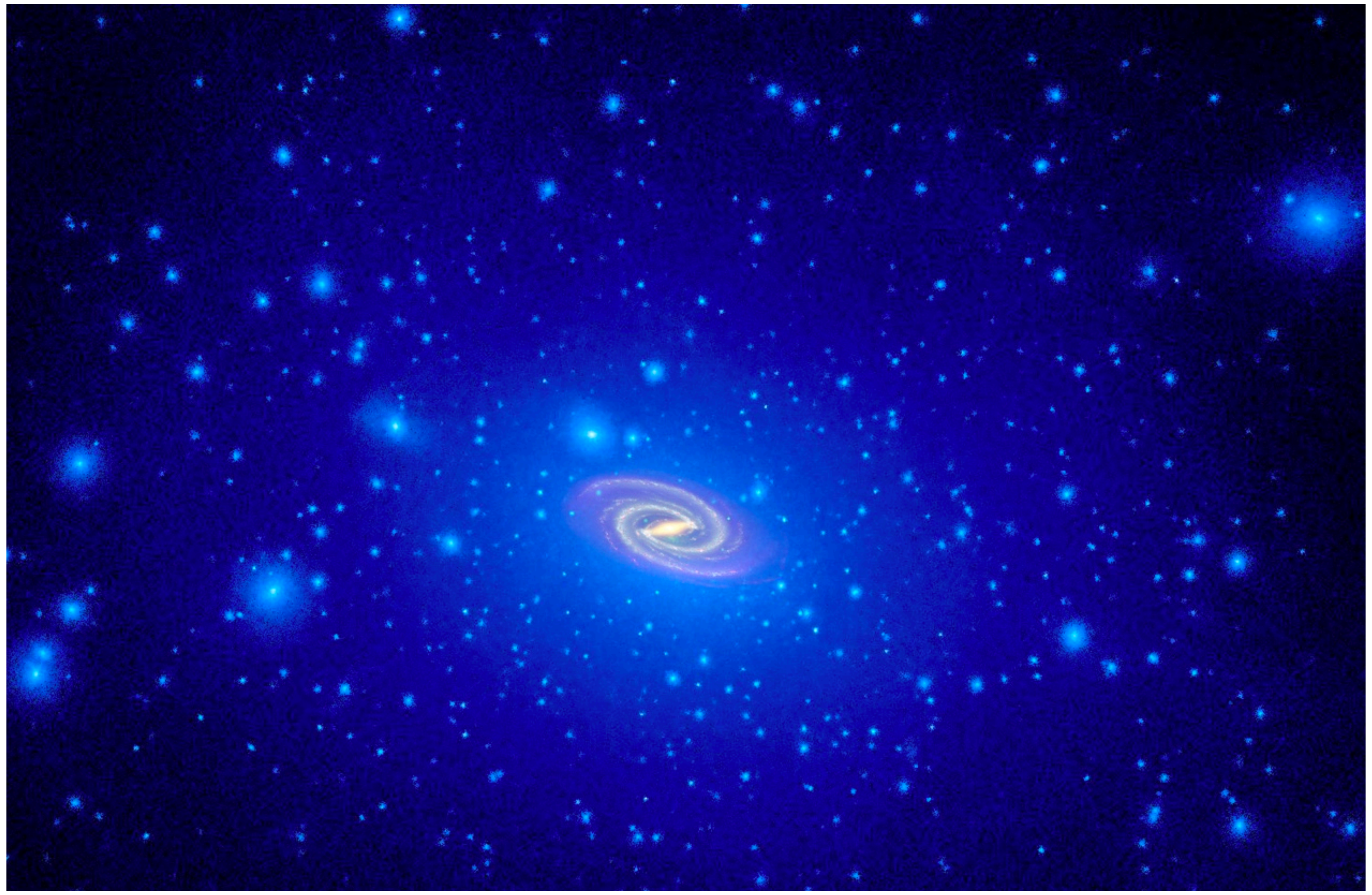
- Preliminary results: we can extract more information when we analyze events $\varphi$ with SALLY than with histograms of $\varphi$ and standard observables

Very preliminary results



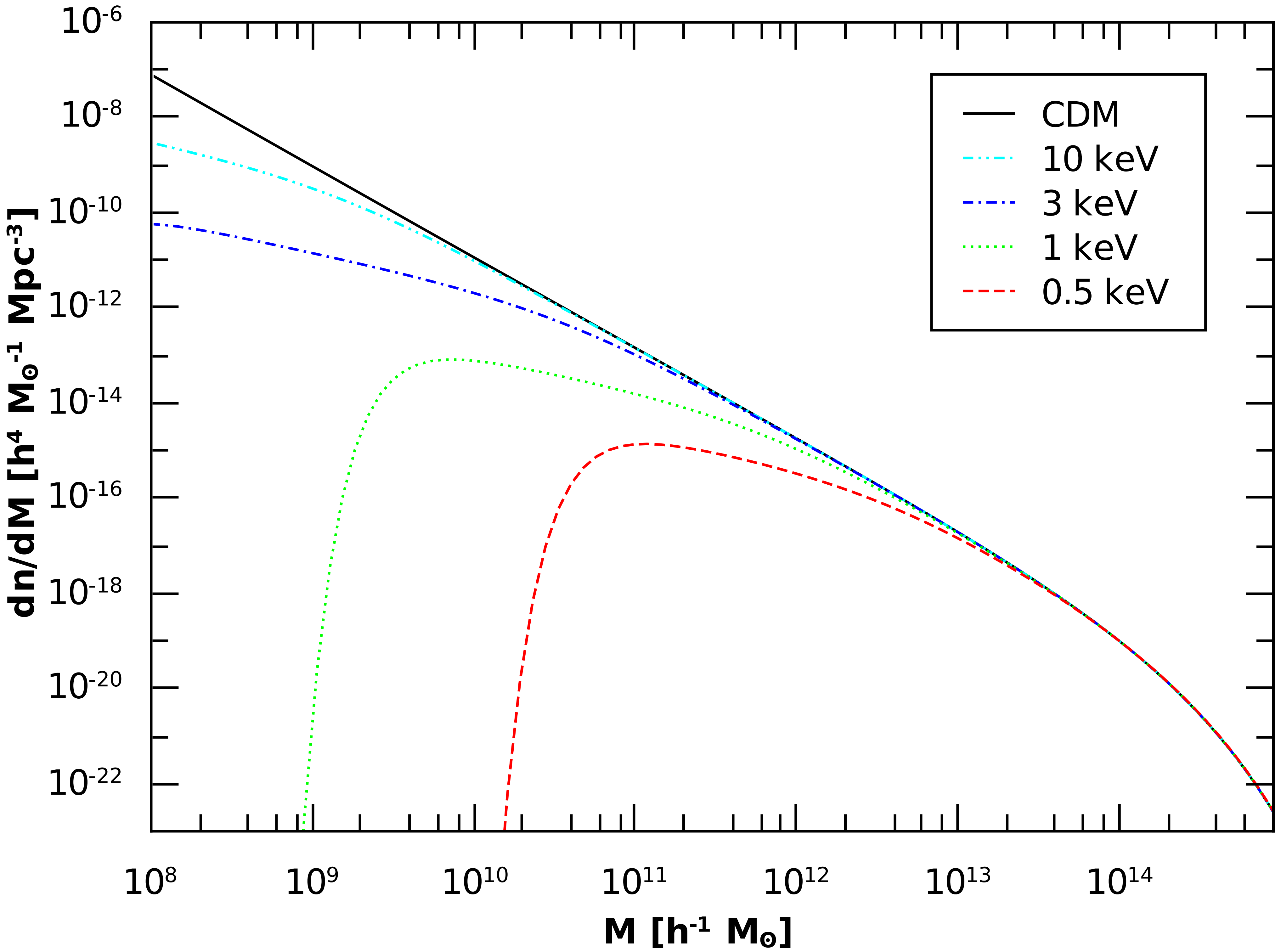
These methods work beyond the LHC.

# DM small-scale structure as a probe of DM particle properties



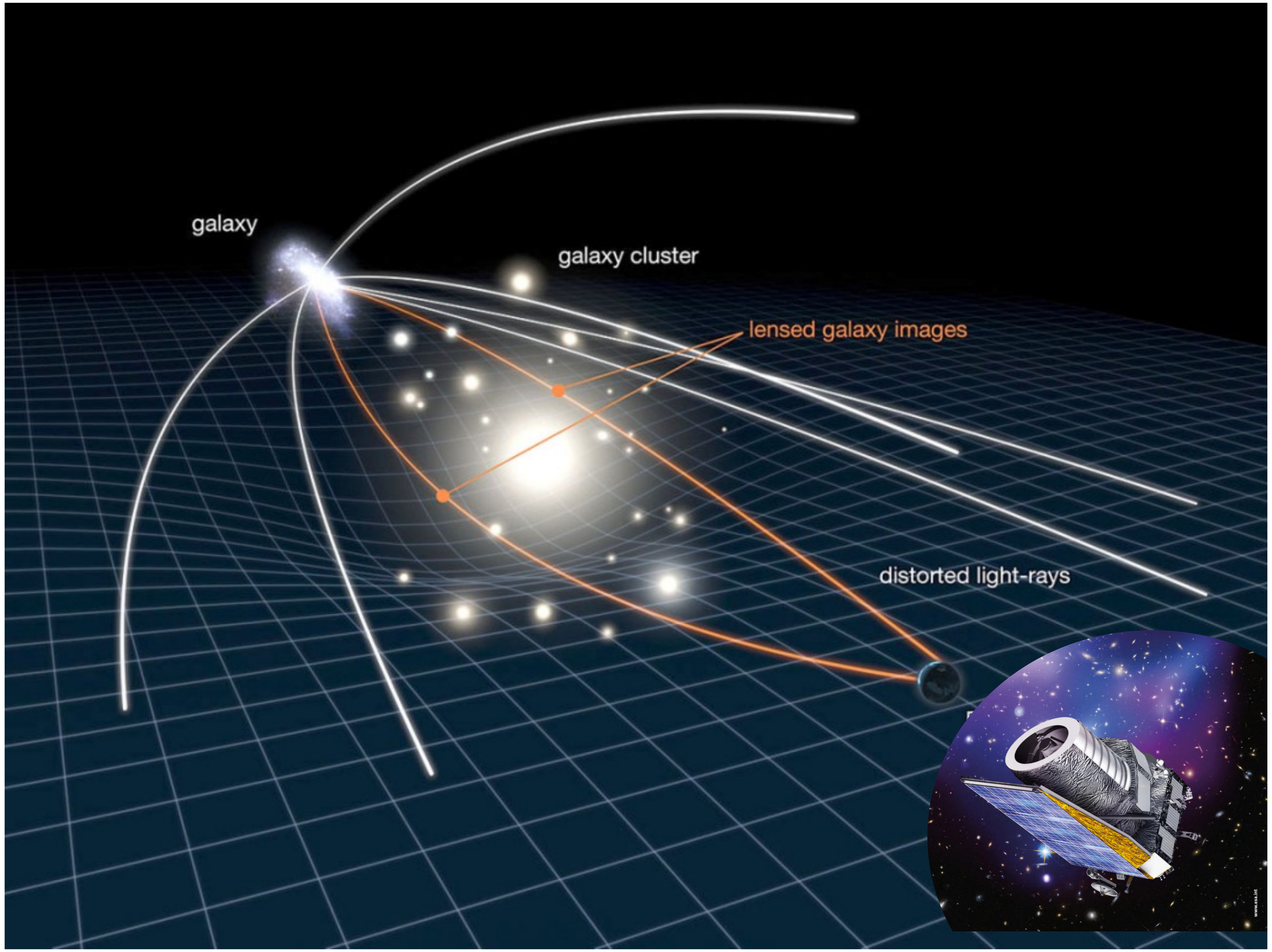
[T. Brown, J.Tumlinson]

Abundance of DM subhalos vs mass:



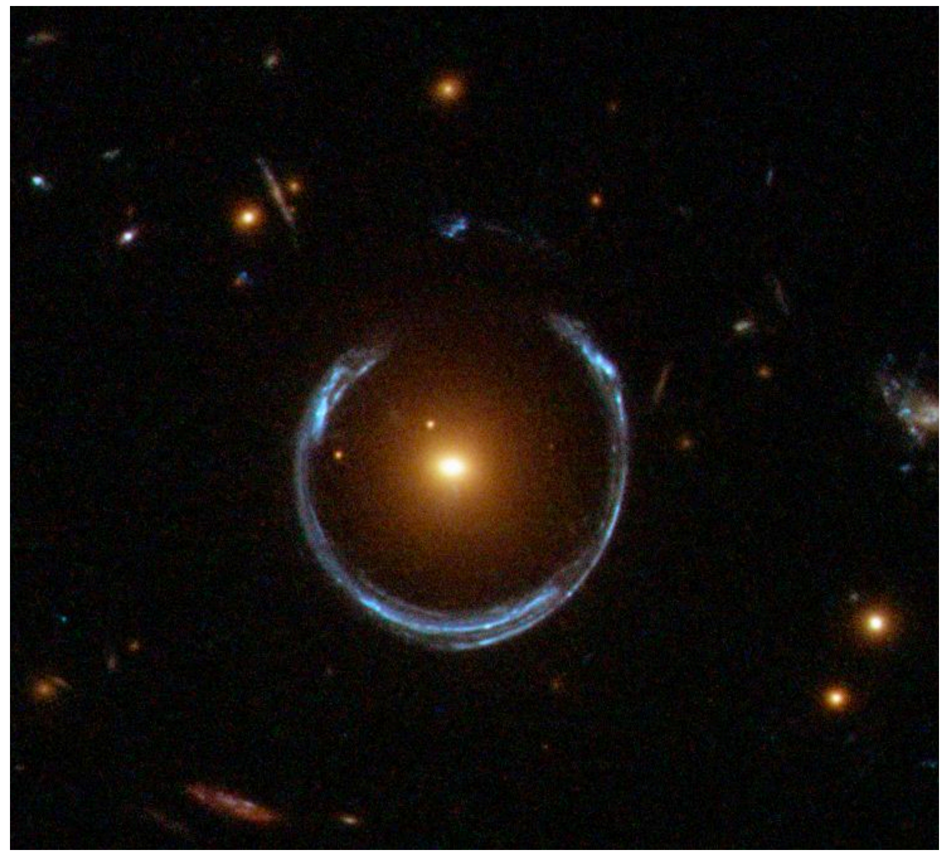

[R. *Dunstan et al* 1109.6291]

# Strong gravitational lensing



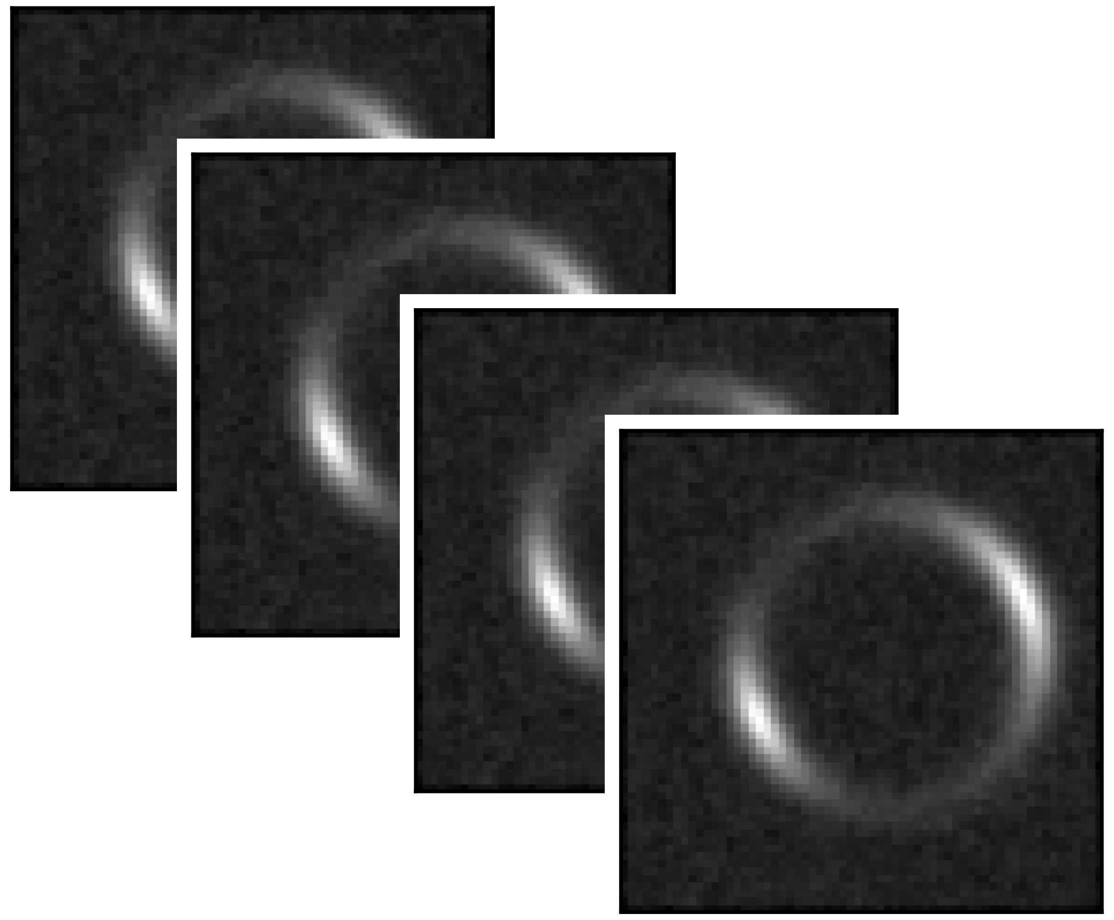

Multiple images of quasars

Extended arcs from galaxies

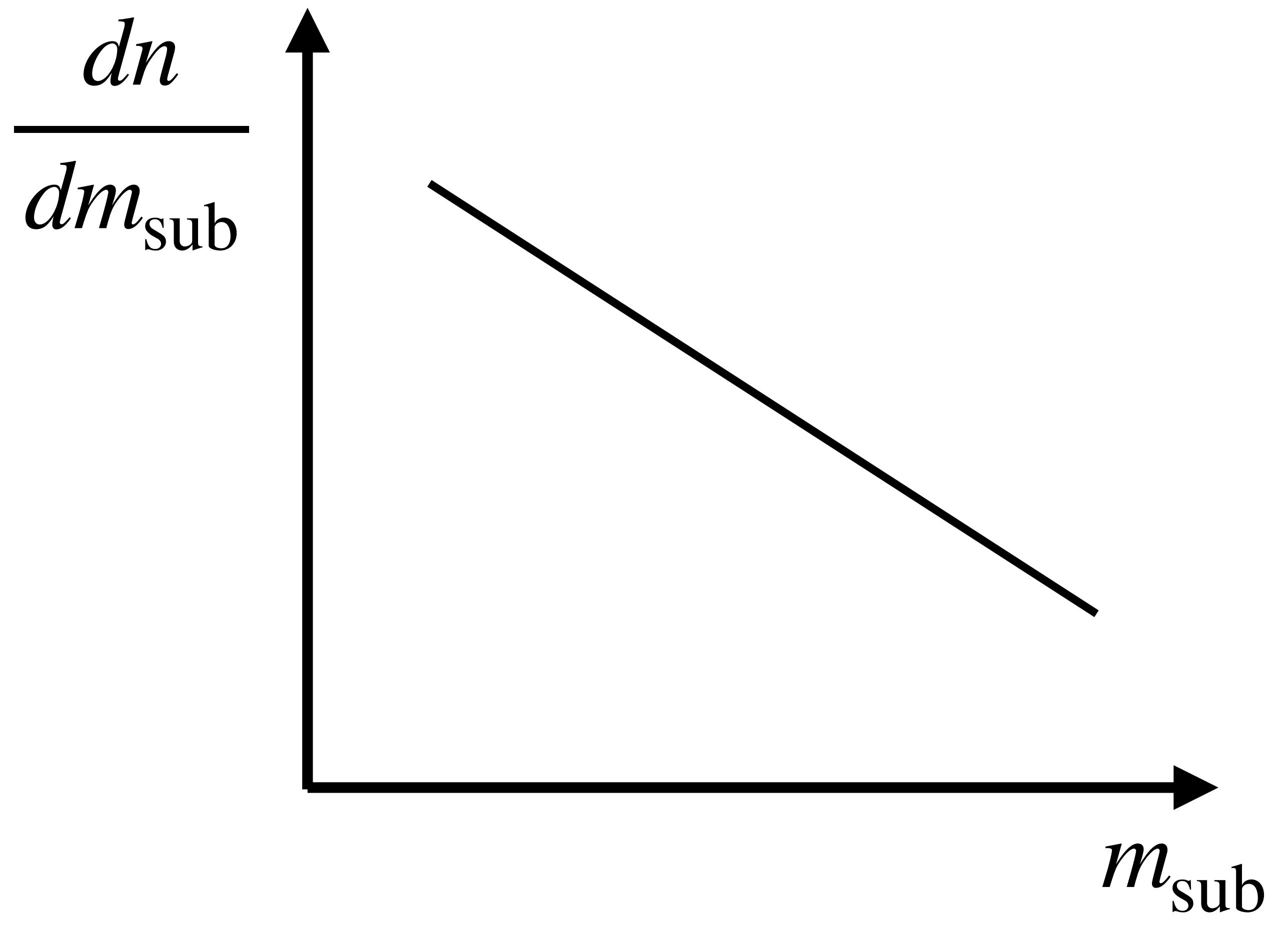


Based on slides by Sid Mishra-Sharma

# Scalable inference for small subhalos



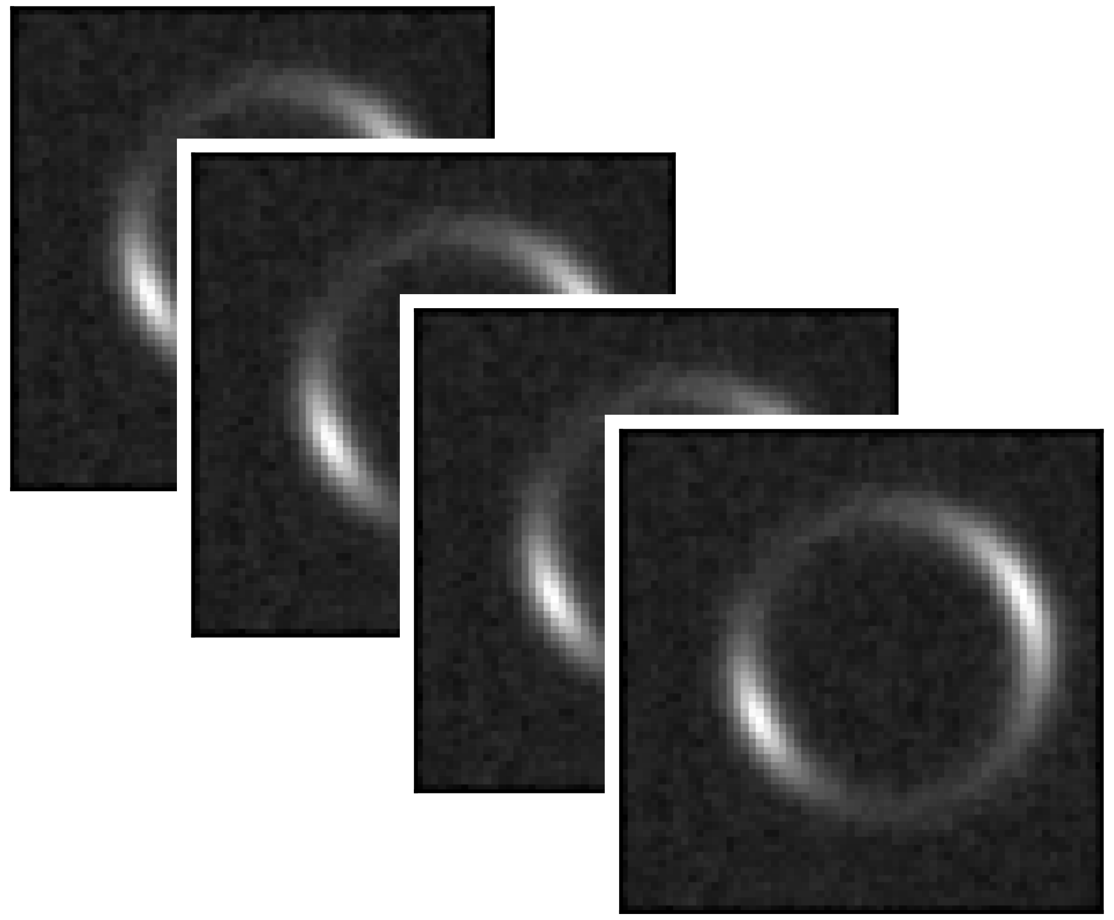
Future surveys (LSST, Euclid) are expected to deliver large samples of galaxy-galaxy strong lenses [Collett et al 1507.02657]



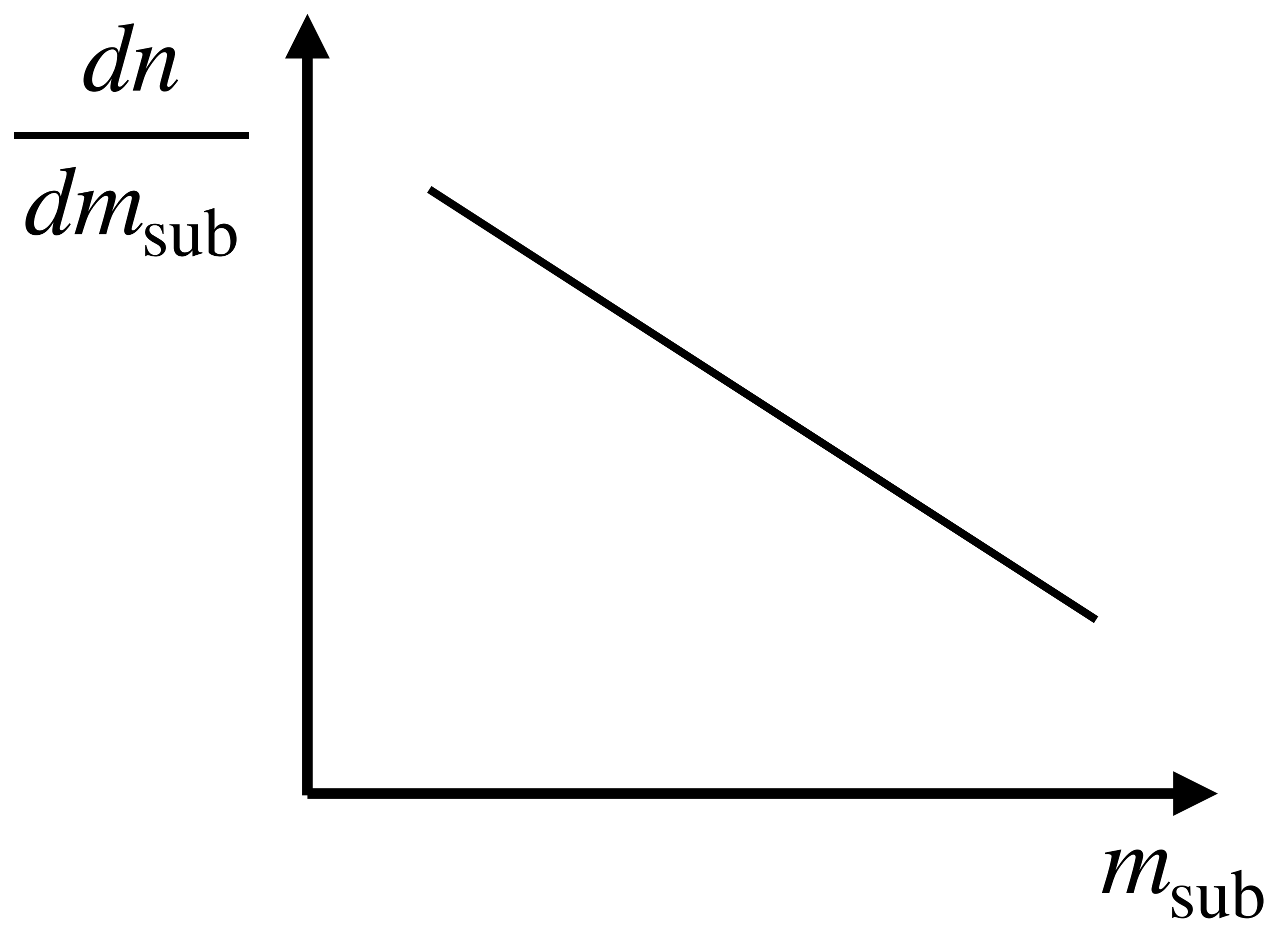

Goal: infer subhalo mass distribution through collective effects of many light subhalos

Based on slides by Sid Mishra-Sharma

# Scalable inference for small subhalos



Future surveys (LSST, Euclid) are expected to deliver large samples of galaxy-galaxy strong lenses [Collett et al 1507.02657]




Goal: infer subhalo mass distribution through collective effects of many light subhalos

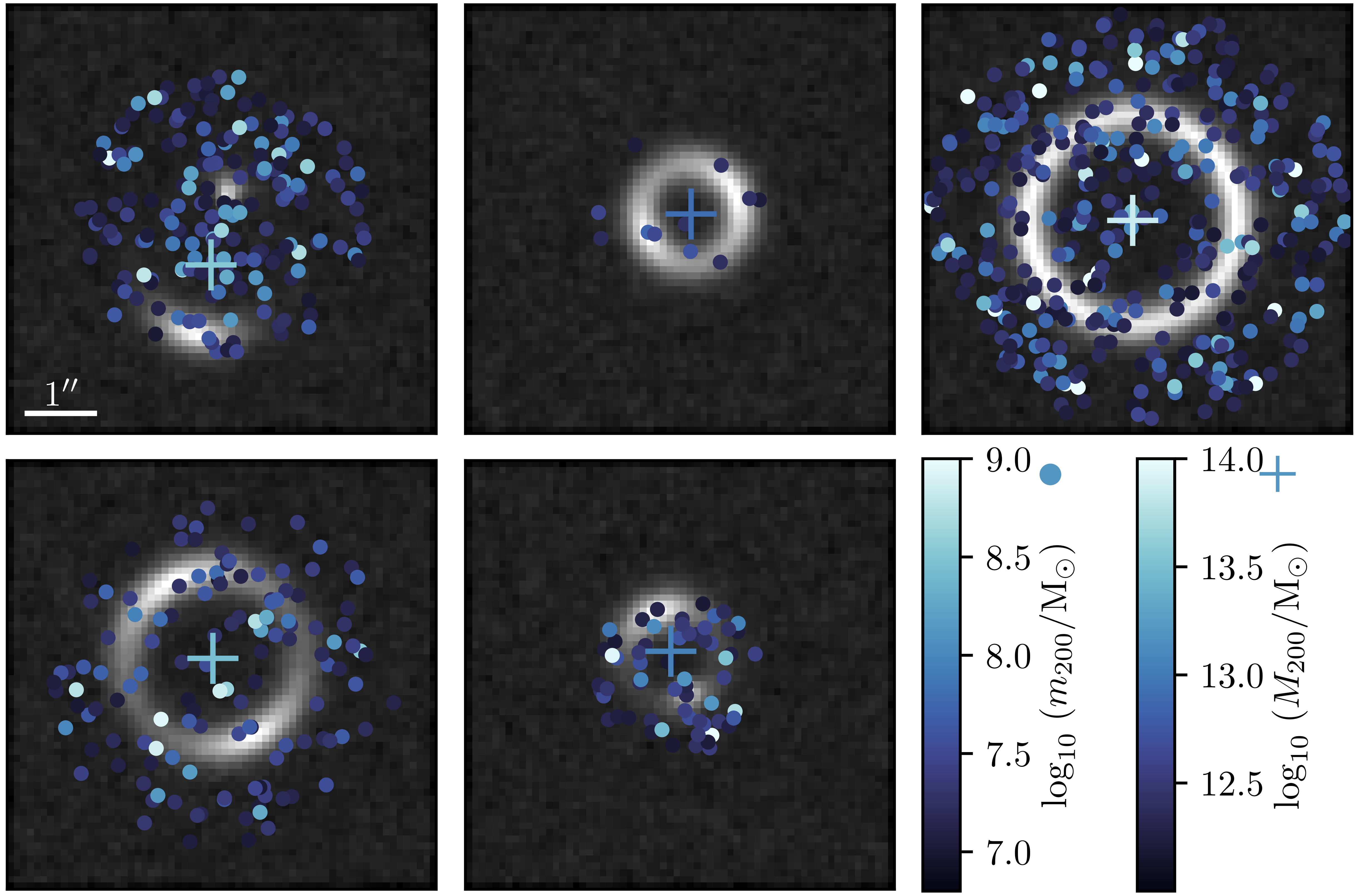
⇒ Need inference technique that

- scales to many lenses (fast evaluation)
- captures subtle effects in high-dimensional image data
- can deal with a large number of subhalos (latent variables)

Based on slides by Sid Mishra-Sharma

# Proof-of-principle simulator

[following T. Collett 1507.02657]

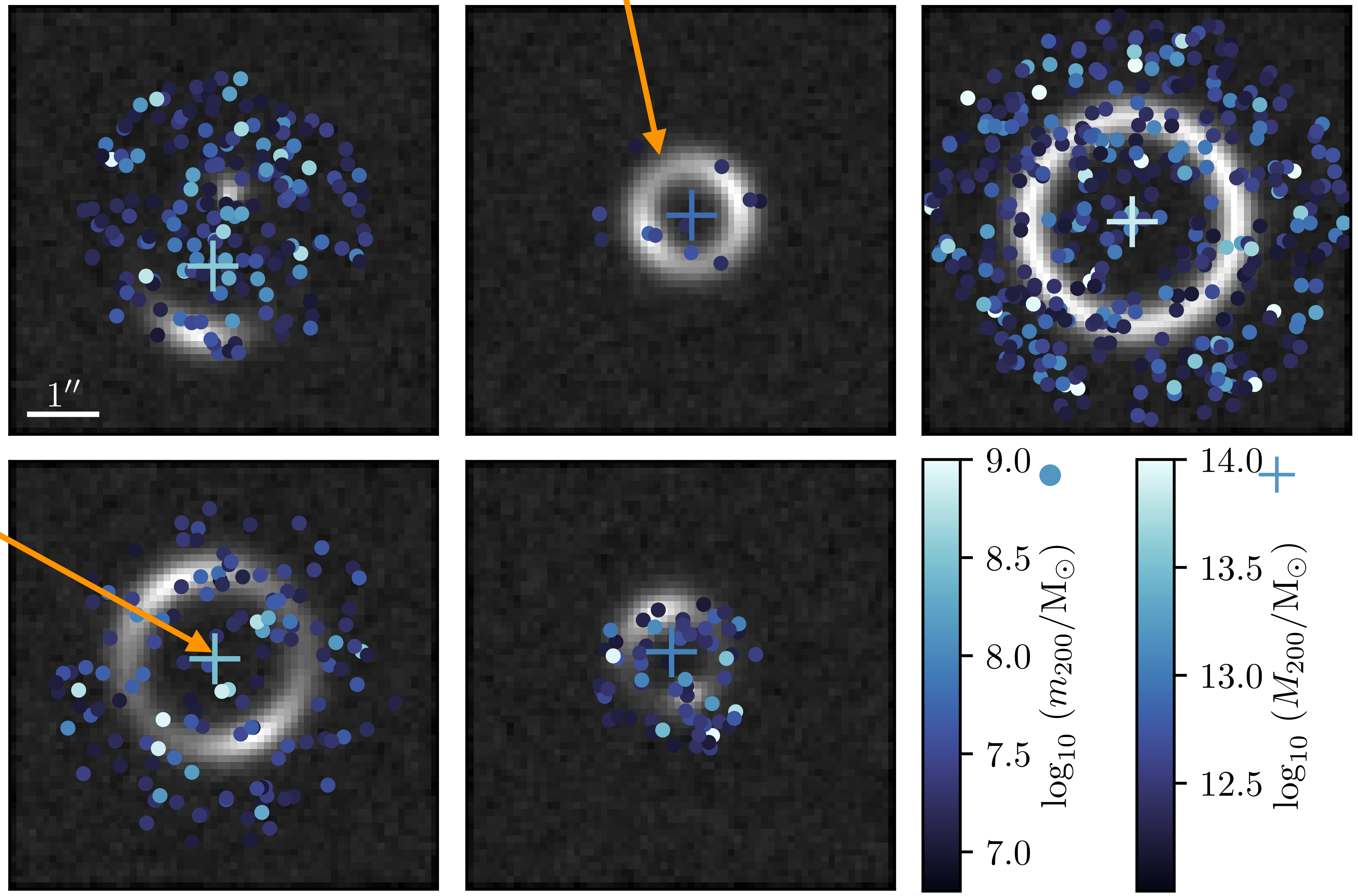


# Proof-of-principle simulator

[following T. Collett 1507.02657]

Spherical lensing host galaxies
at redshift ~0.5…1

Extended galaxy sources
at redshift 1.5

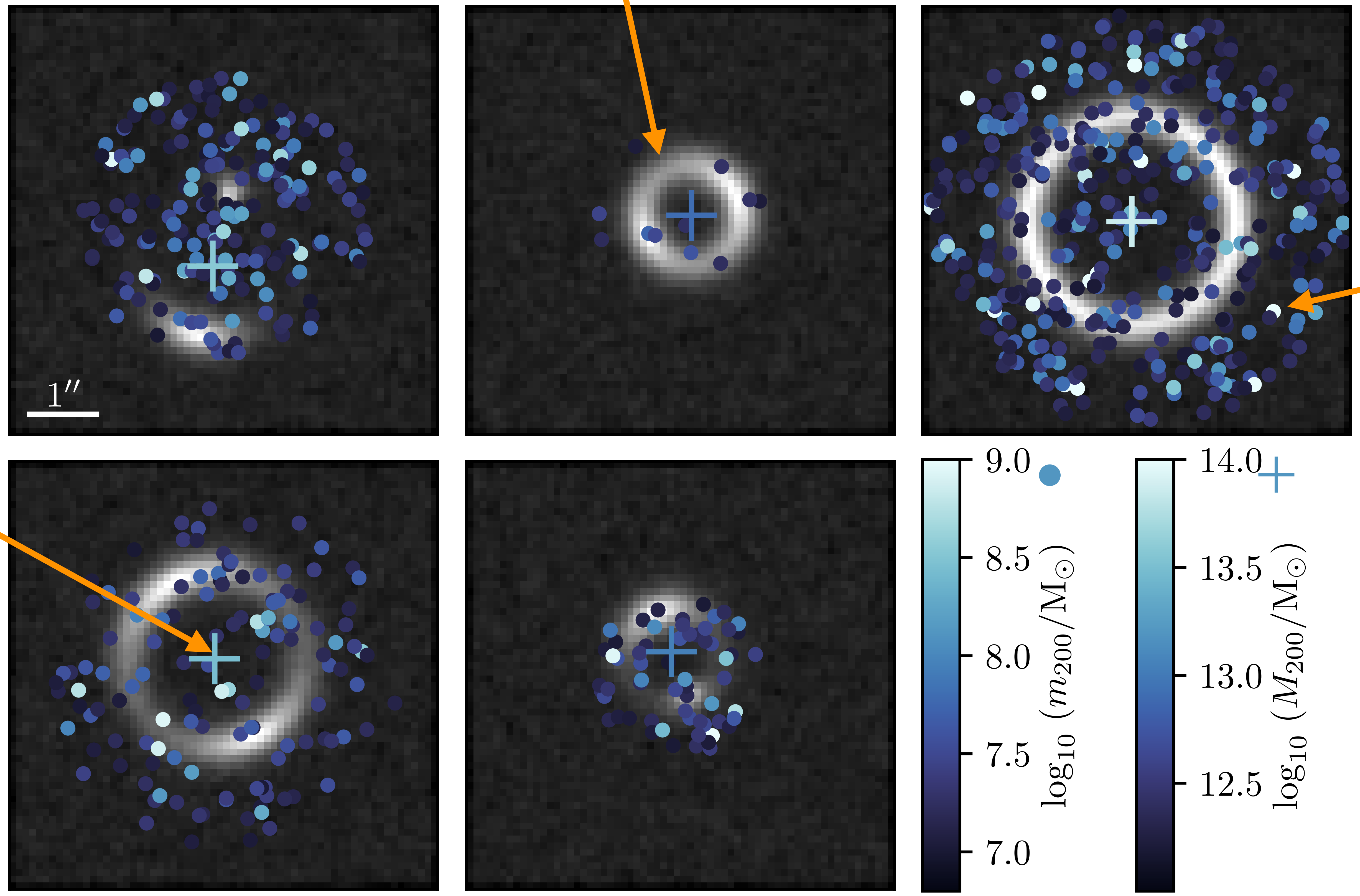


# Proof-of-principle simulator

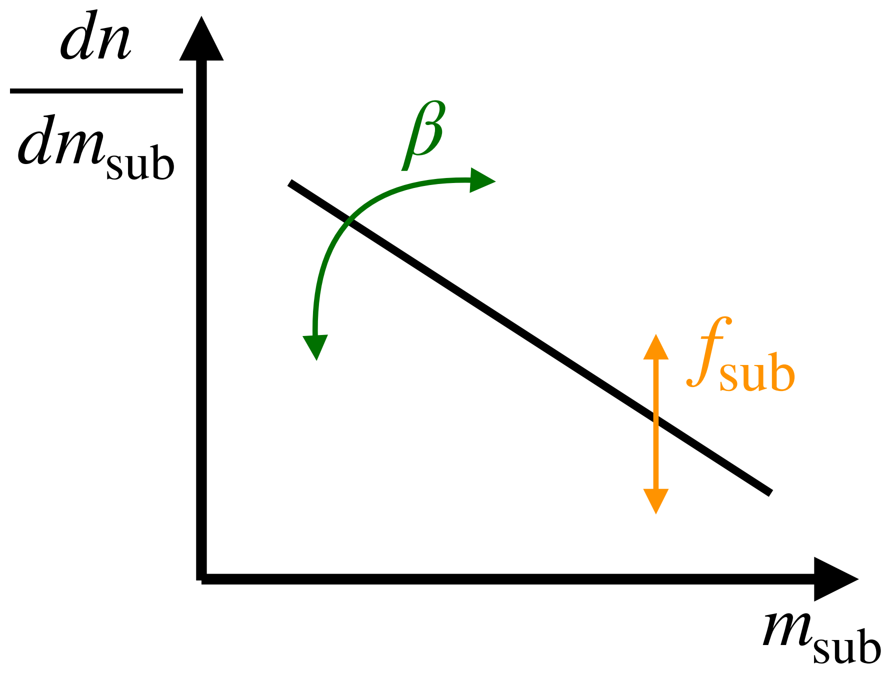
[following T. Collett 1507.02657]

Spherical lensing host galaxies
at redshift ~0.5…1

Extended galaxy sources
at redshift 1.5




Subhalos follow mass distribution with two parameters



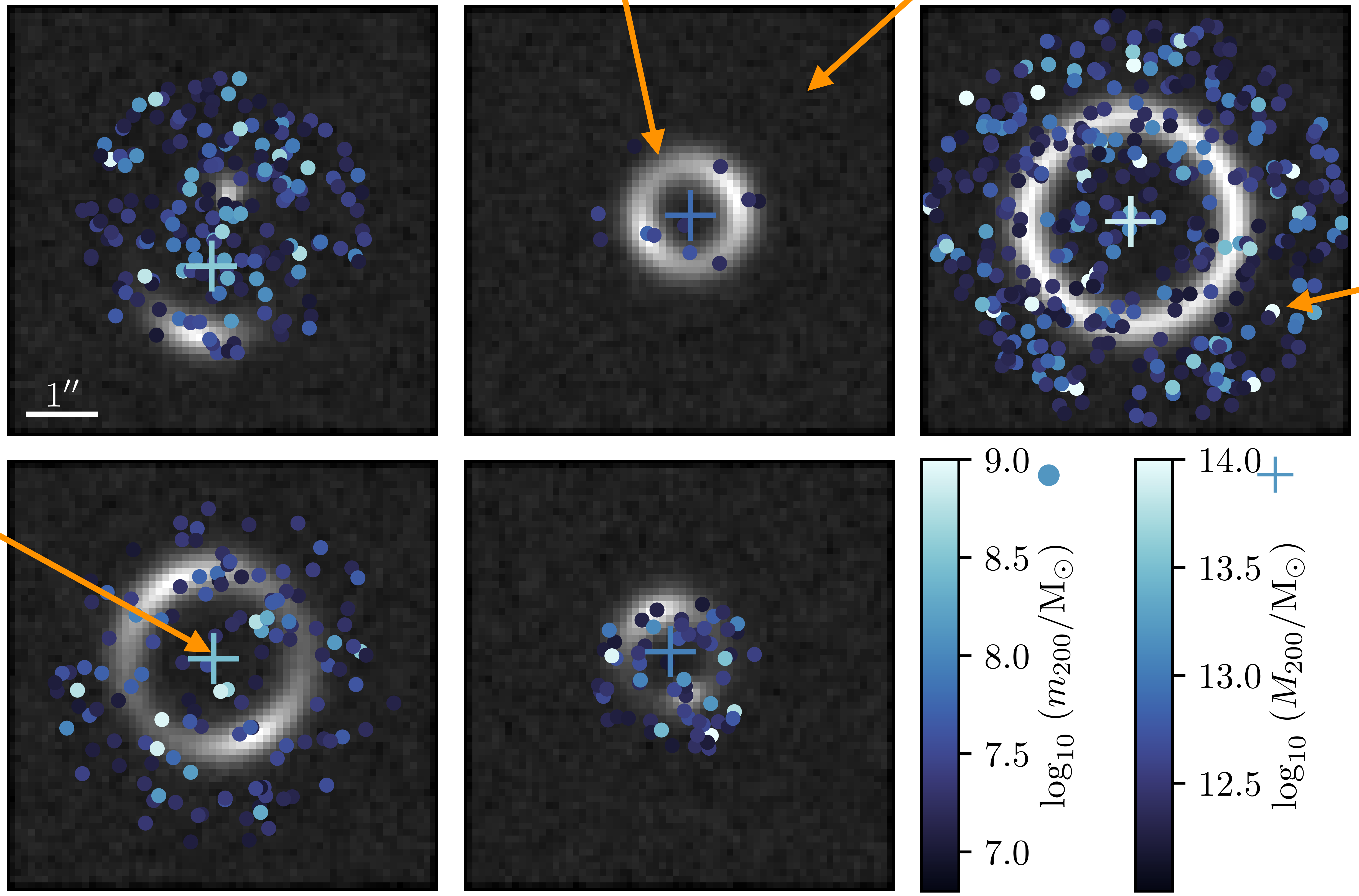
# Proof-of-principle simulator

[following T. Collett 1507.02657]

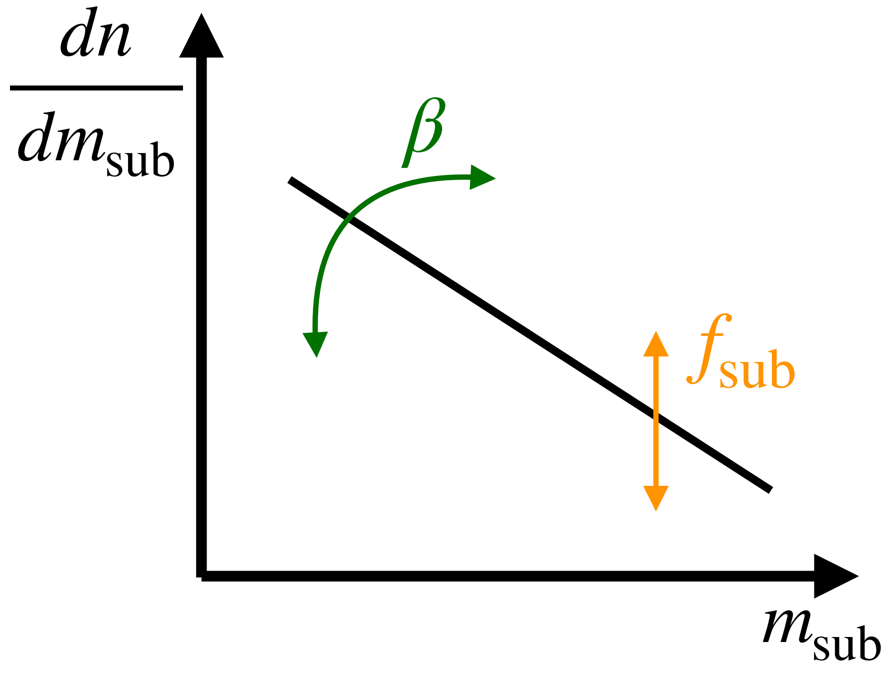
Spherical lensing host galaxies
at redshift ~0.5…1

Poisson fluctuations

Extended galaxy sources
at redshift 1.5




Subhalos follow mass distribution with two parameters



# Proof-of-principle simulator

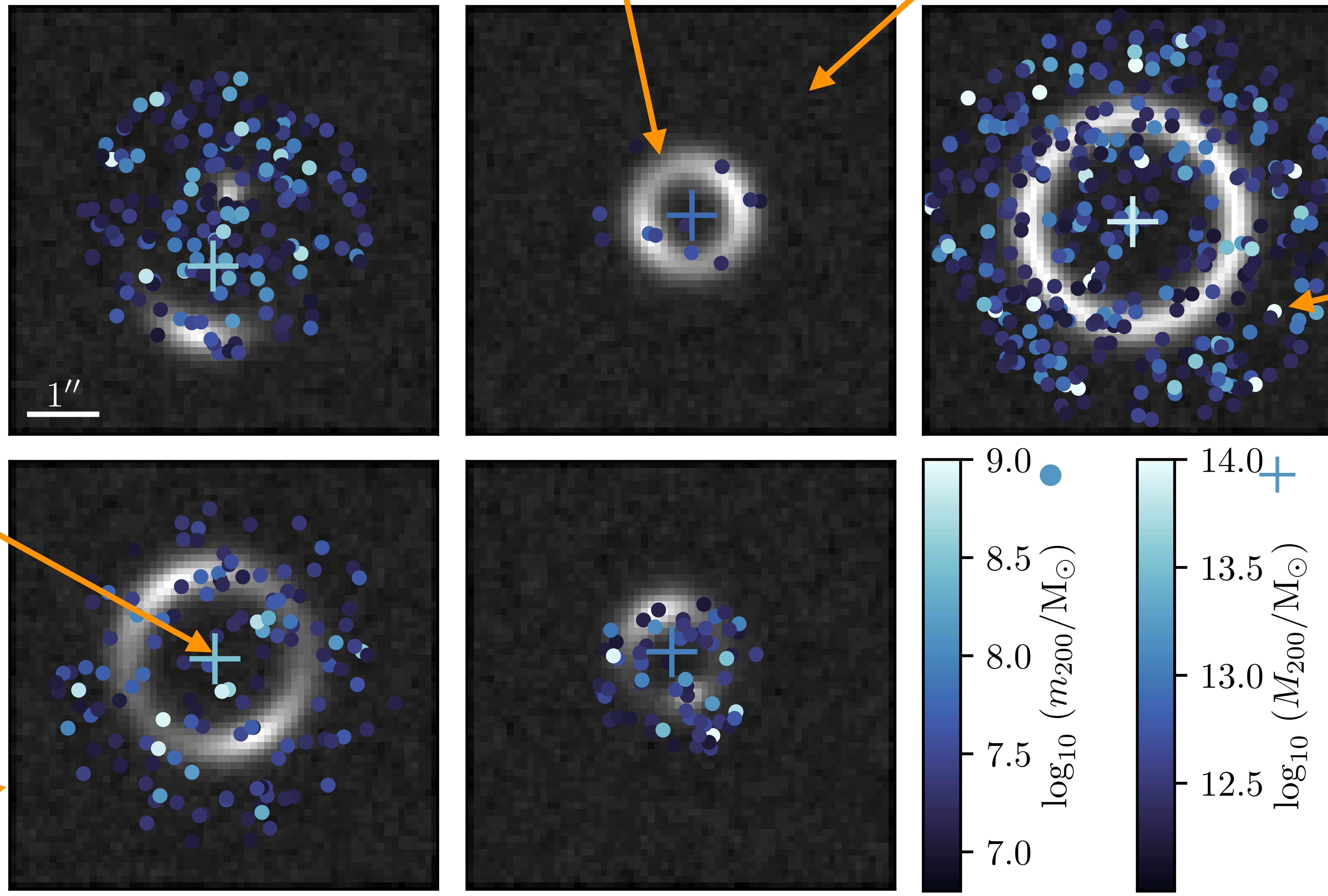
[following T. Collett 1507.02657]

Spherical lensing host galaxies
at redshift ~0.5…1

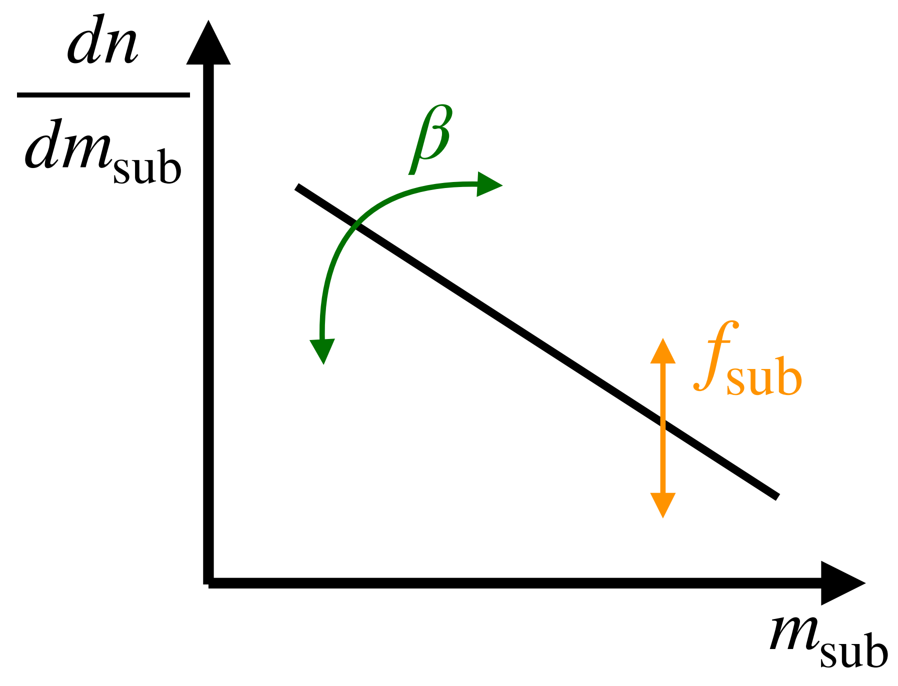
Poisson fluctuations

Extended galaxy sources
at redshift 1.5

Euclid detector model
(single optical band,
$64^2$ pixel, Gaussian PSF)




Subhalos follow mass distribution with two parameters

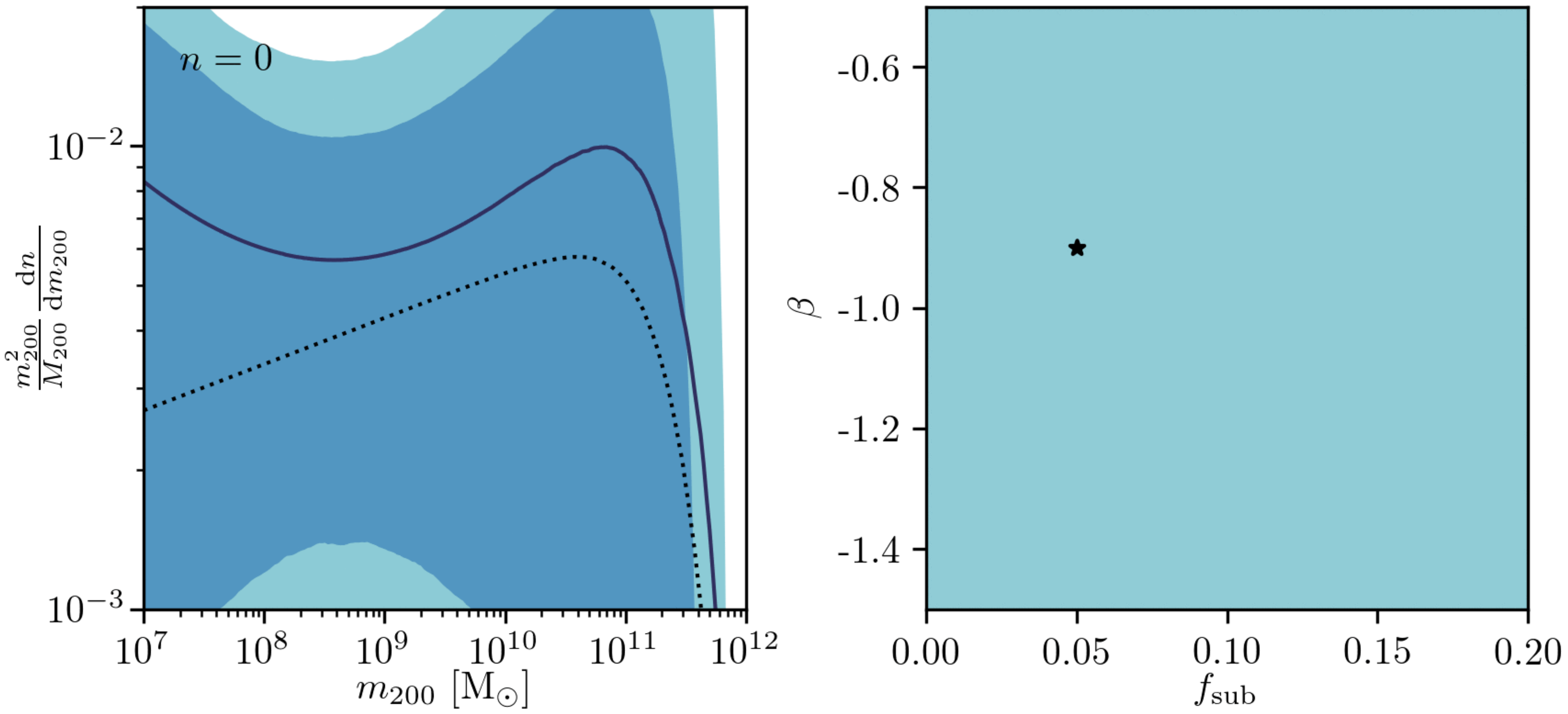


# Bayesian inference

Sid Mishra-Sharma   Johann Brehmer   Gilles Louppe   Joeri Hermans

+ K.C. published in ApJ, 886 (2019) arXiv:1909.02005 ]

Watch how knowledge of the subhalos mass distribution improves as data comes in. See posterior for two parameters concentrate around true value used to generate mock data




- fsub, defined as the fraction of the total dark matter halo mass contained in bound substructure in a given mass range
- The halo virial mass M200 describes the total mass contained with the virial radius r200, defined as the radius within which the mean density is 200 times the critical density of the universe

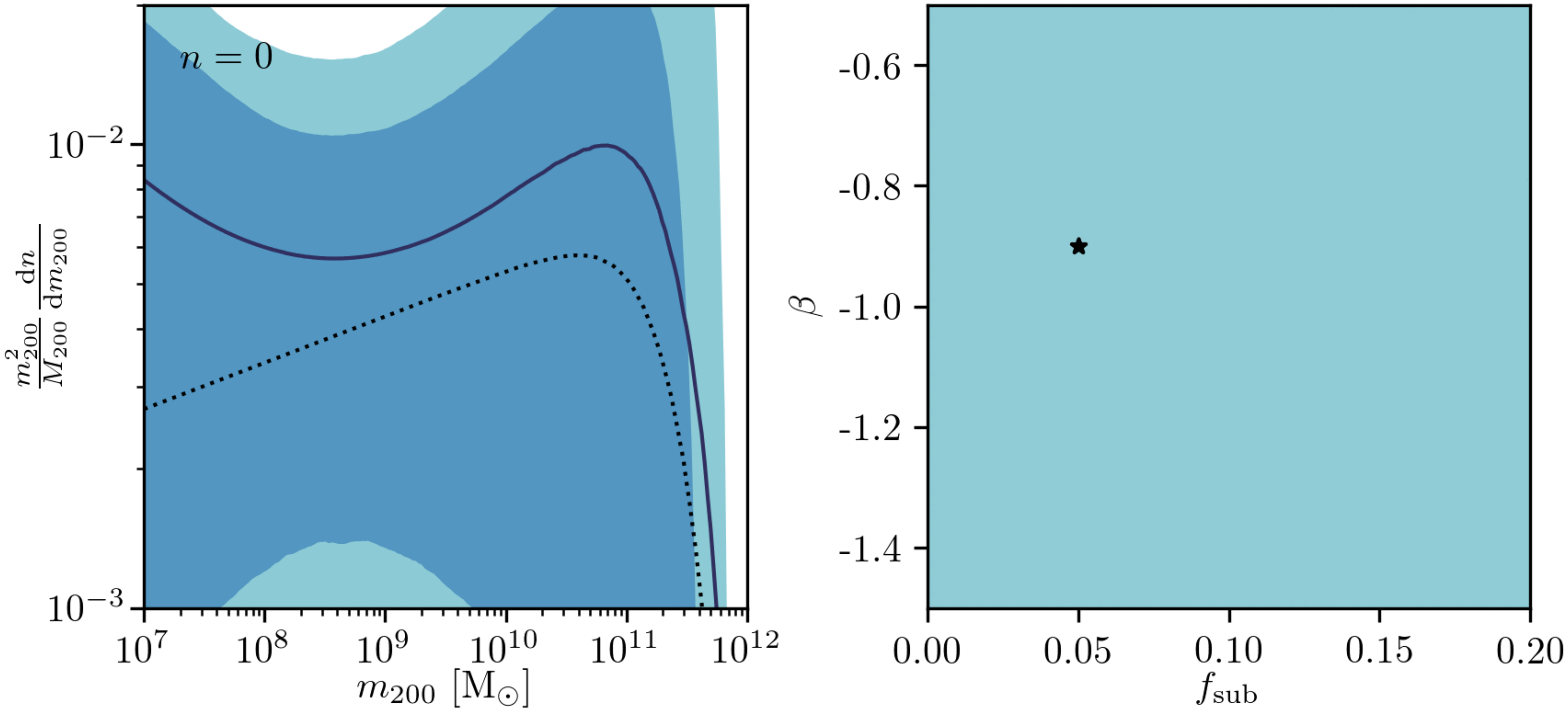
# Bayesian inference




Sid Mishra-Sharma Johann Brehmer Gilles Louppe Joeri Hermans

+ K.C. published in ApJ, 886 (2019) arXiv:1909.02005 ]

Watch how knowledge of the subhalos mass distribution improves as data comes in. See posterior for two parameters concentrate around true value used to generate mock data




- fsub, defined as the fraction of the total dark matter halo mass contained in bound substructure in a given mass range
- The halo virial mass M200 describes the total mass contained with the virial radius r200, defined as the radius within which the mean density is 200 times the critical density of the universe

# Conclusions

Information geometry is a useful concept for phenomenology and could be very useful for Snowmass studies

Simulation-based inference techniques powered by machine learning have the potential to greatly improve our sensitivity for a wide range of searches and measurements in particle physics

- In particular, it allows us to model the likelihood for a lower-level, high-dimensional representation of the data including complicated detector effects

# References


**Opinionated review**
K. Cranmer, JB, G. Louppe:
"The frontier of simulation-based inference"
[1911.01429]

**Do It Yourself (for LHC physics)**
JB, F. Kling, I. Espejo, K. Cranmer:
"MadMiner: Machine learning—based inference for particle physics"
[CSBS, 1907.10621, https://github.com/diana-hep/madminer]

**Strong lensing**
JB, 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]

**LHC HXSWG YR4 STXS**
JB, S. Dawson, S. Homiller, F. Kling, T. Plehn:
"Benchmarking simplified template cross sections in WH production"
[JHEP, 1908.06980]

**Meh, the original trilogy was better**
JB, K. Cranmer, G. Louppe, J. Pavez:
"Constraining Effective Field Theories with machine learning"
[PRL, 1805.00013]

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

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

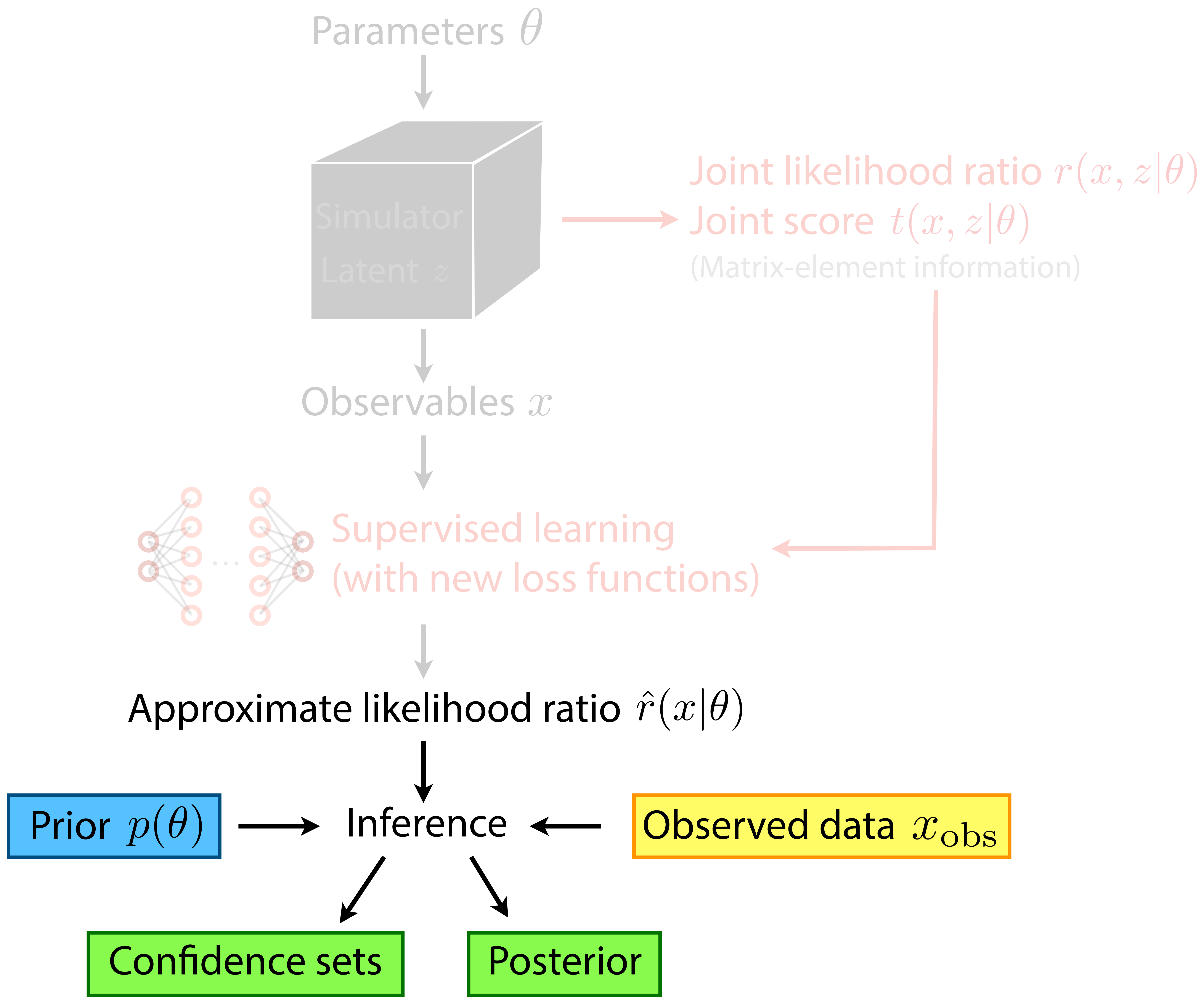
**Follow-up with incremental improvements**
M. Stoye, JB, K. Cranmer, G. Louppe, J. Pavez:
"Likelihood-free inference with an improved cross-entropy estimator"
[NeurIPS workshop, 1808.00973]

Backup

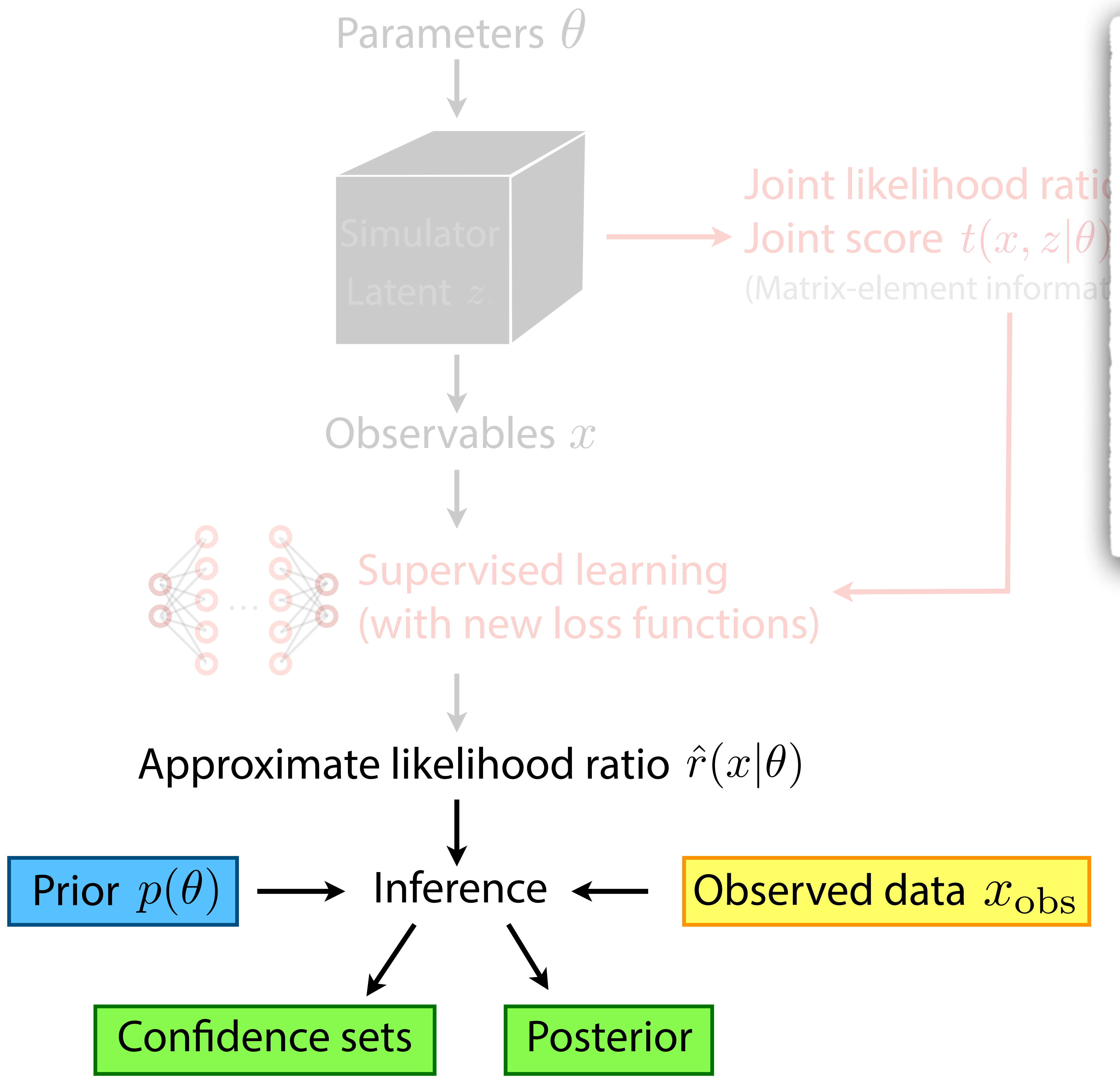
# Inference, Limits, & Fisher Information Geometry

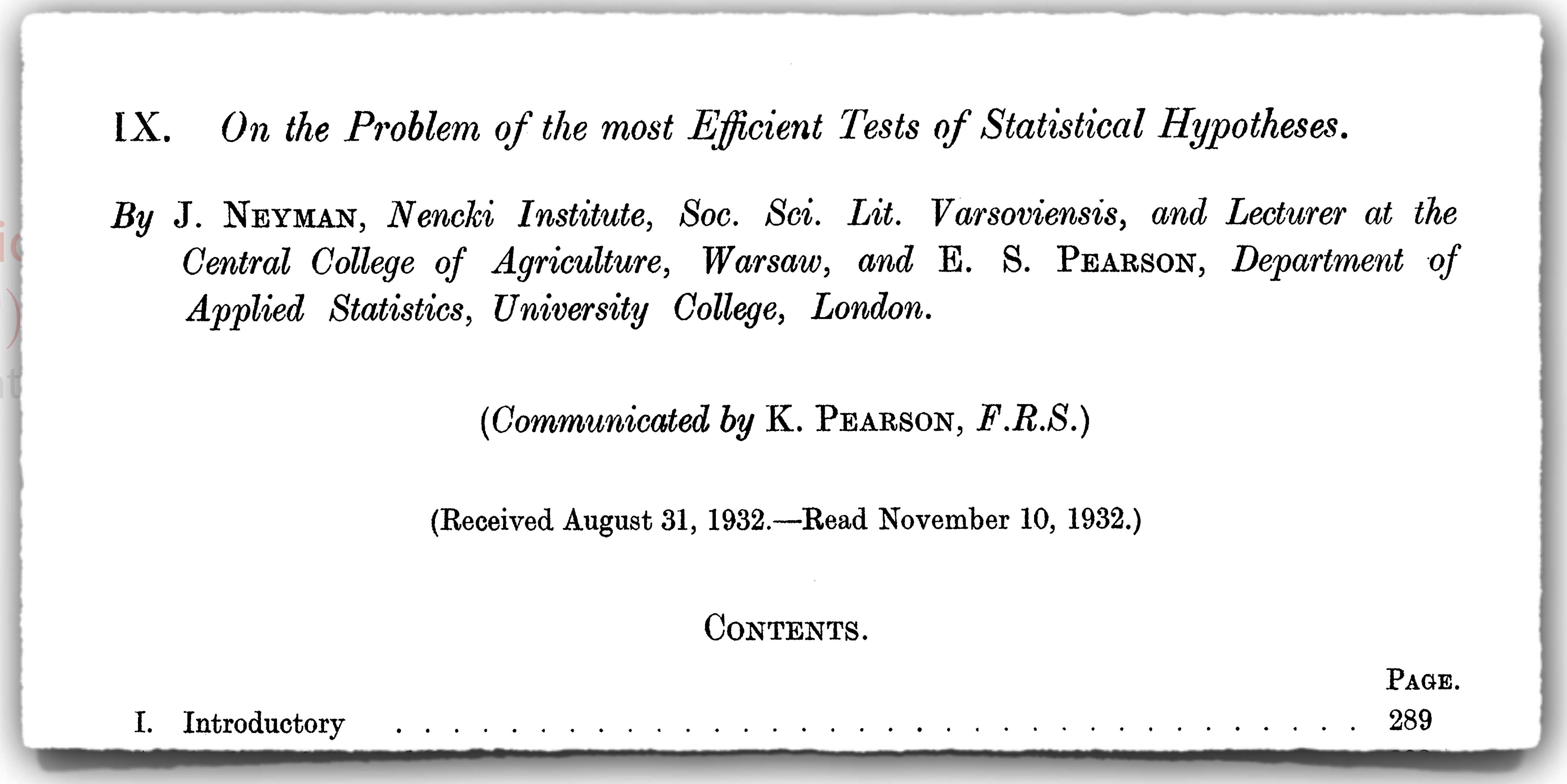
[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013, 1805.00020, 1805.12244]

# Step 3: Inference



# Step 3: Inference



IX. *On the Problem of the most Efficient Tests of Statistical Hypotheses.*

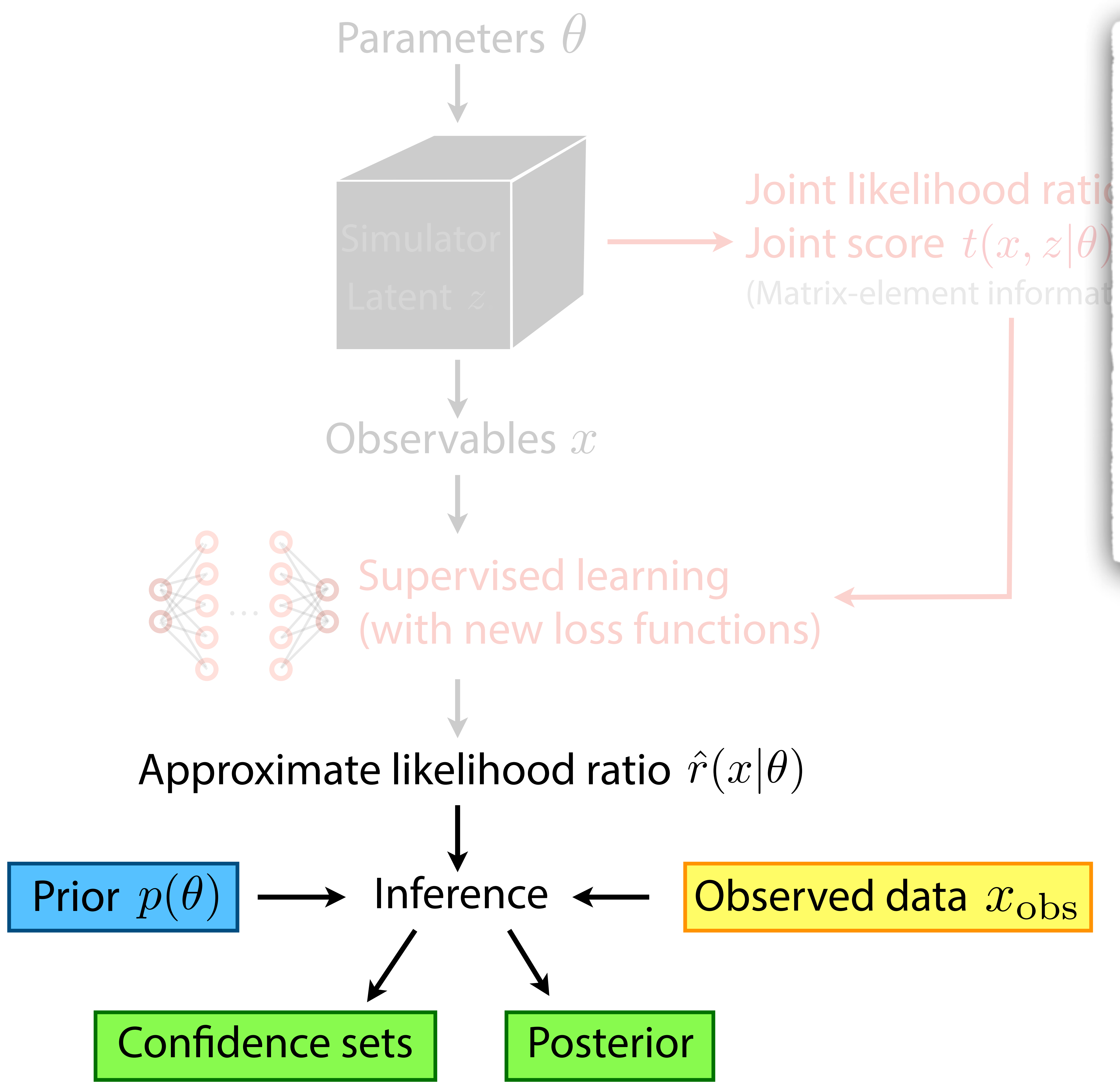
*By* J. Neyman, *Nencki Institute, Soc. Sci. Lit. Varsoviensis, and Lecturer at the Central College of Agriculture, Warsaw, and* E. S. Pearson, *Department of Applied Statistics, University College, London.*

(*Communicated by* K. Pearson, *F.R.S.*)

(Received August 31, 1932.—Read November 10, 1932.)

Contents.

# Step 3: Inference




IX. *On the Problem of the most Efficient Tests of Statistical Hypotheses.*

*By* J. NEYMAN, *Nencki Institute, Soc. Sci. Lit. Varsoviensis, and Lecturer at the Central College of Agriculture, Warsaw, and* E. S. PEARSON, *Department of Applied Statistics, University College, London.*

(*Communicated by* K. PEARSON, *F.R.S.*)

(Received August 31, 1932.—Read November 10, 1932.)


Eur. Phys. J. C (2011) 71: 1554
DOI 10.1140/epjc/s10052-011-1554-0

THE EUROPEAN PHYSICAL JOURNAL C

Special Article - Tools for Experiment and Theory


**Asymptotic formulae for likelihood-based tests of new physics**


Glen Cowan[1], Kyle Cranmer[2], Eilam Gross[3], Ofer Vitells[3,a]

[1]Physics Department, Royal Holloway, University of London, Egham TW20 0EX, UK
[2]Physics Department, New York University, New York, NY 10003, USA
[3]Weizmann Institute of Science, Rehovot 76100, Israel



Received: 15 October 2010 / Revised: 6 January 2011 / Published online: 9 February 2011

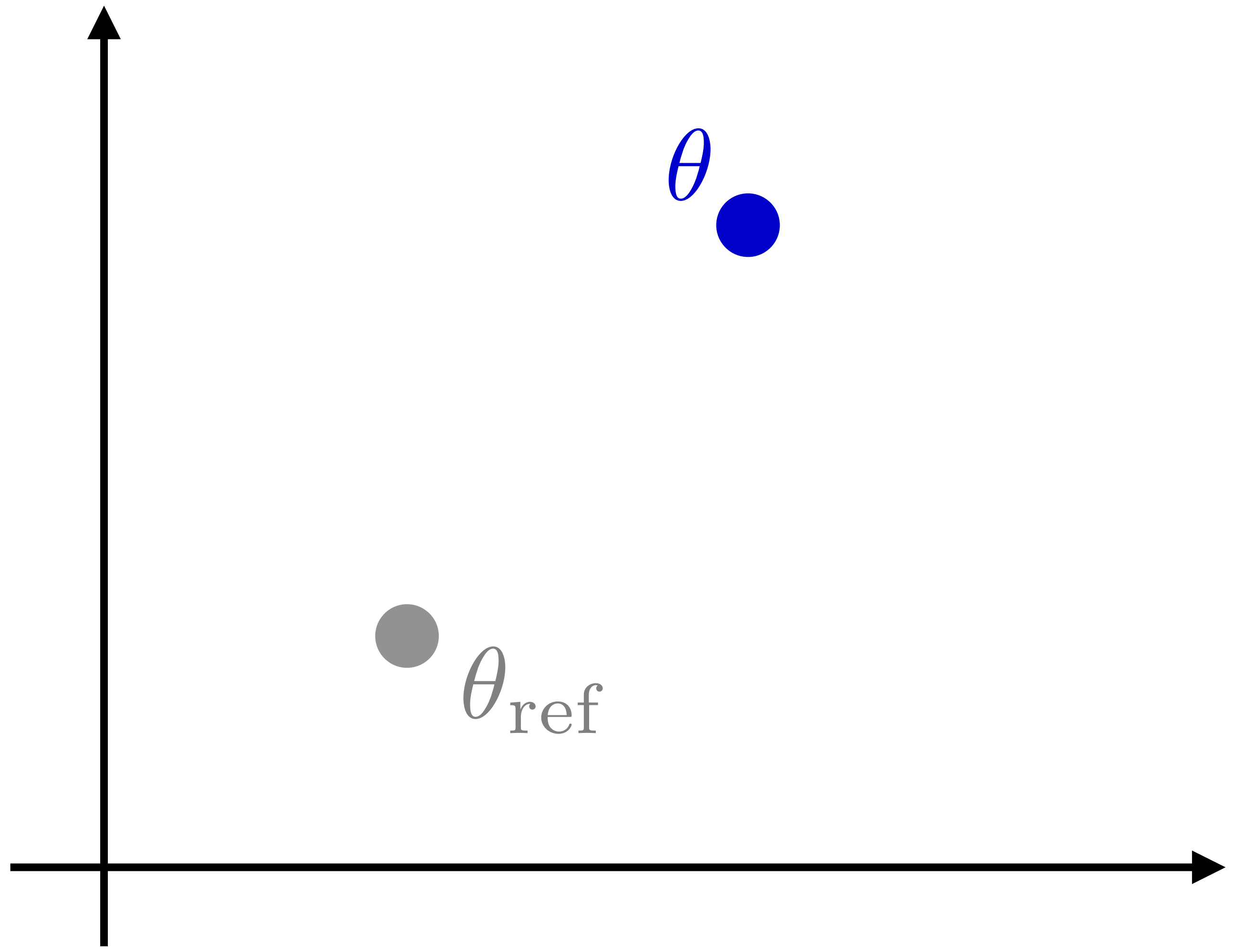



**Abstract** We describe likelihood-based statistical tests for use in high energy physics for the discovery of new phenomena and for construction of confidence intervals on model data sets by a single representative one, referred to here as the "Asimov" data set.[1] In the past, this method has been used and justified intuitively (e.g., [4, 5]). Here we provide

# Limit setting (frequentist, standard ATLAS / CMS practice)

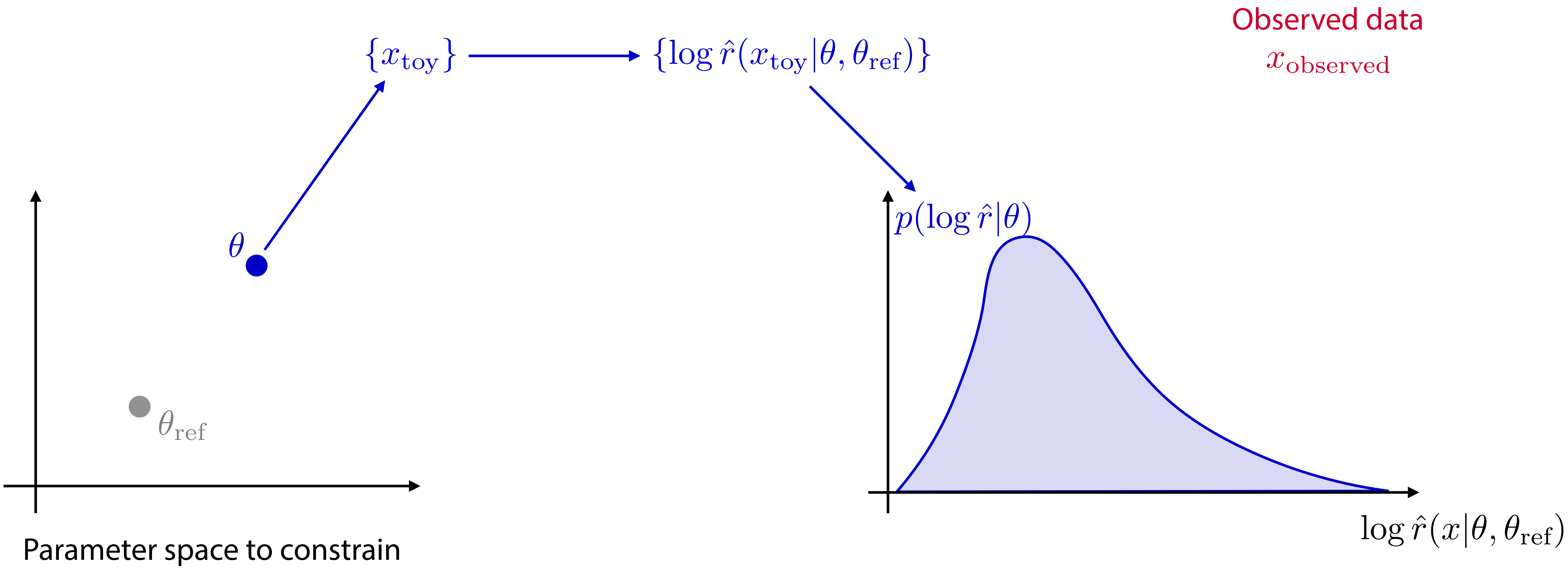
Observed data

$x_{\text{observed}}$

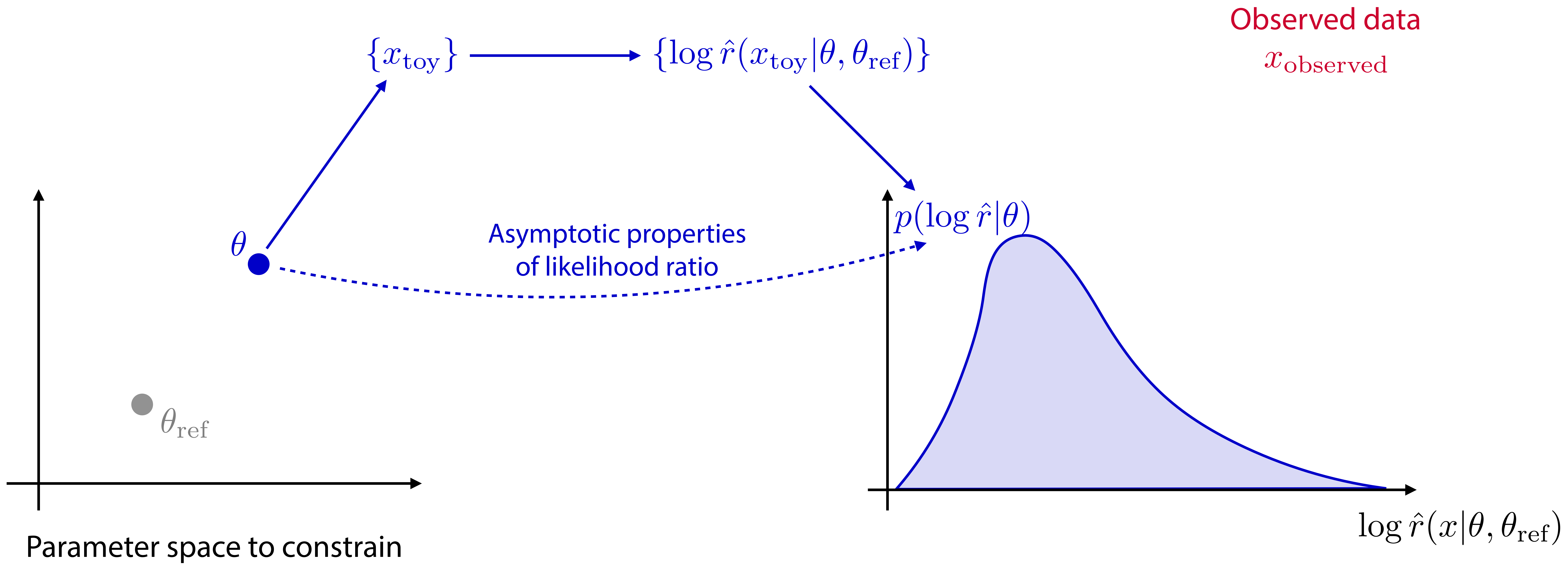



Parameter space to constrain

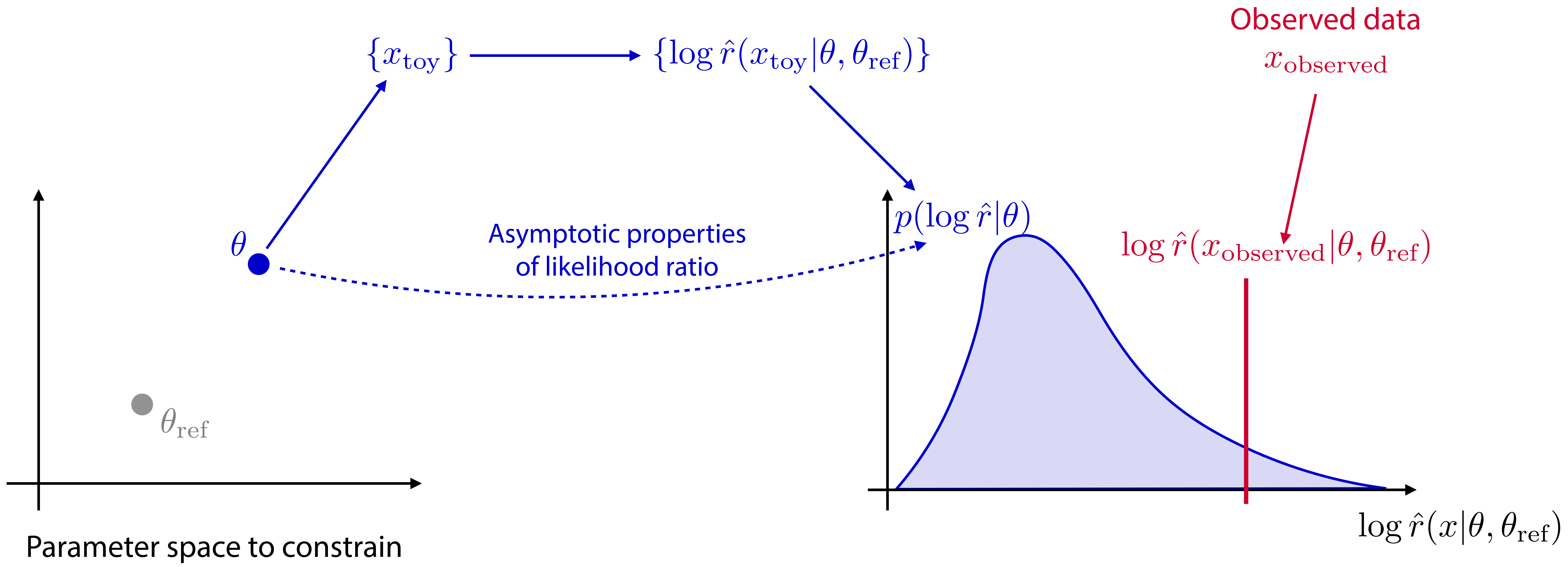
# Limit setting (frequentist, standard ATLAS / CMS practice)



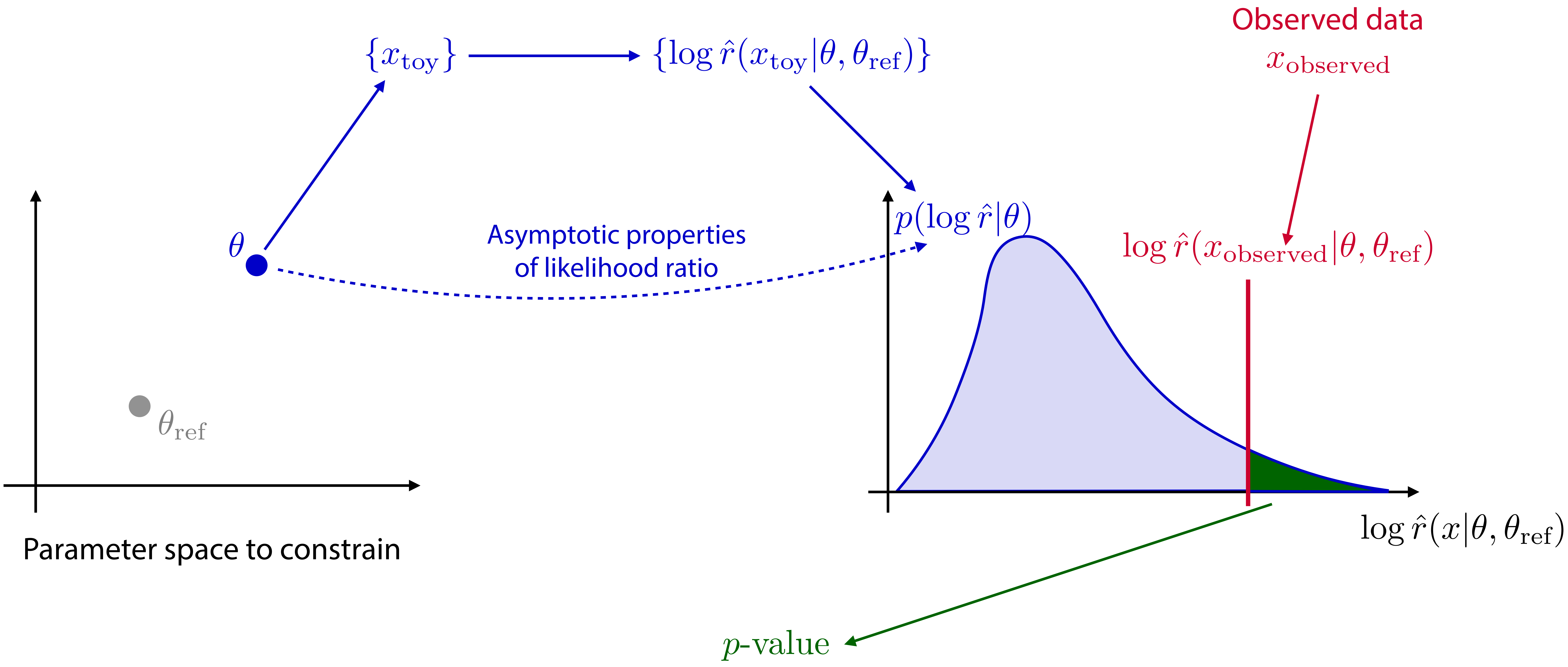
# Limit setting (frequentist, standard ATLAS / CMS practice)



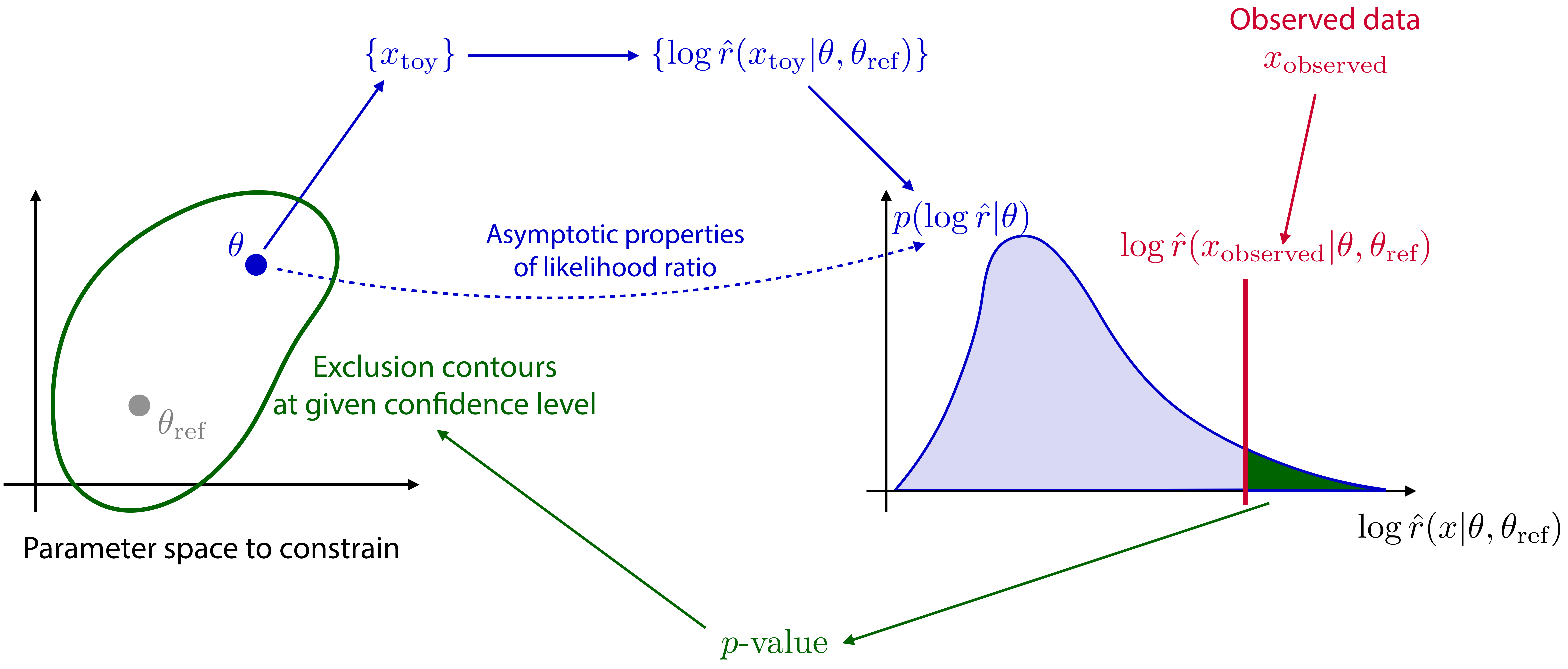
# Limit setting (frequentist, standard ATLAS / CMS practice)



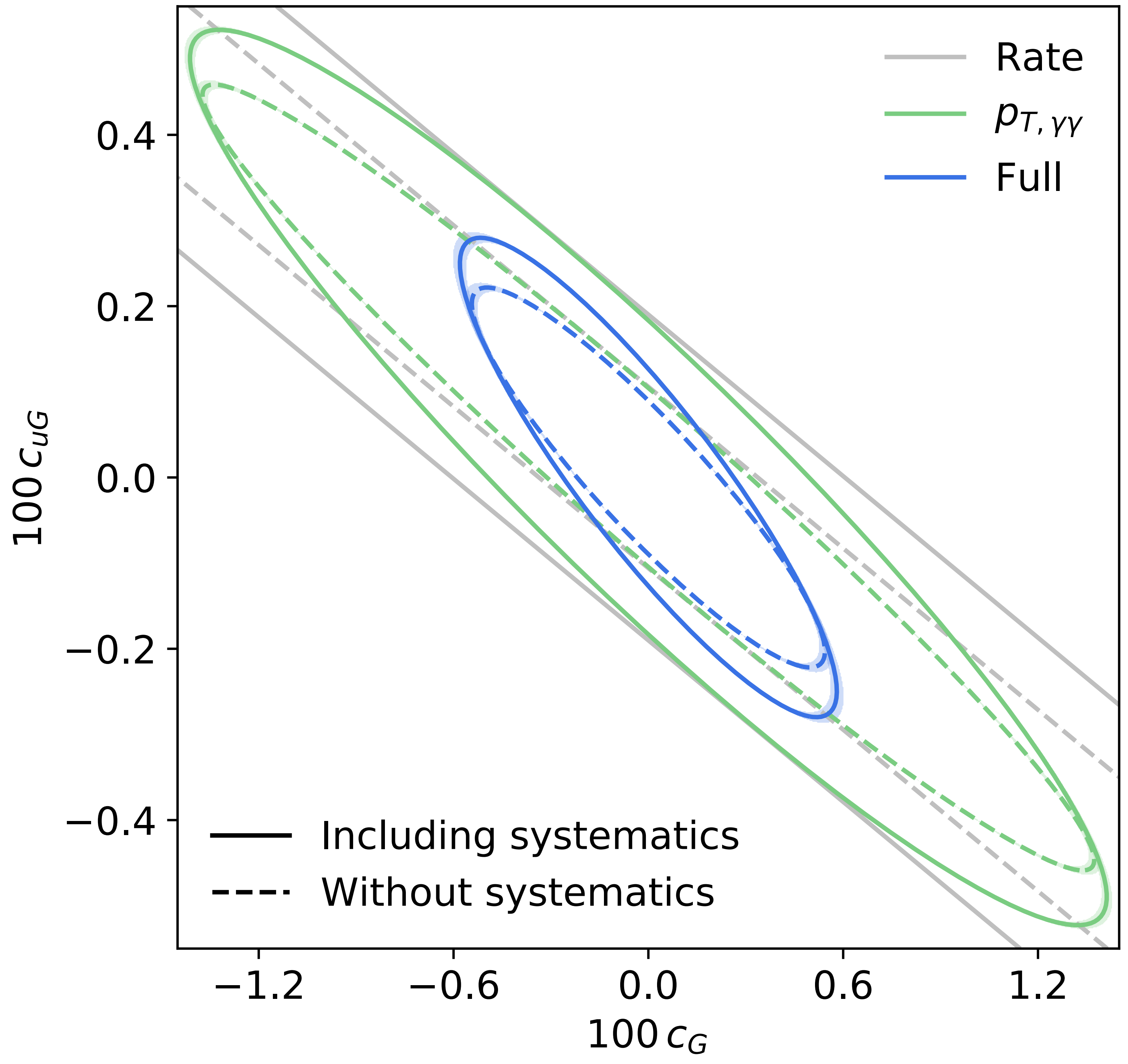
# Limit setting (frequentist, standard ATLAS / CMS practice)



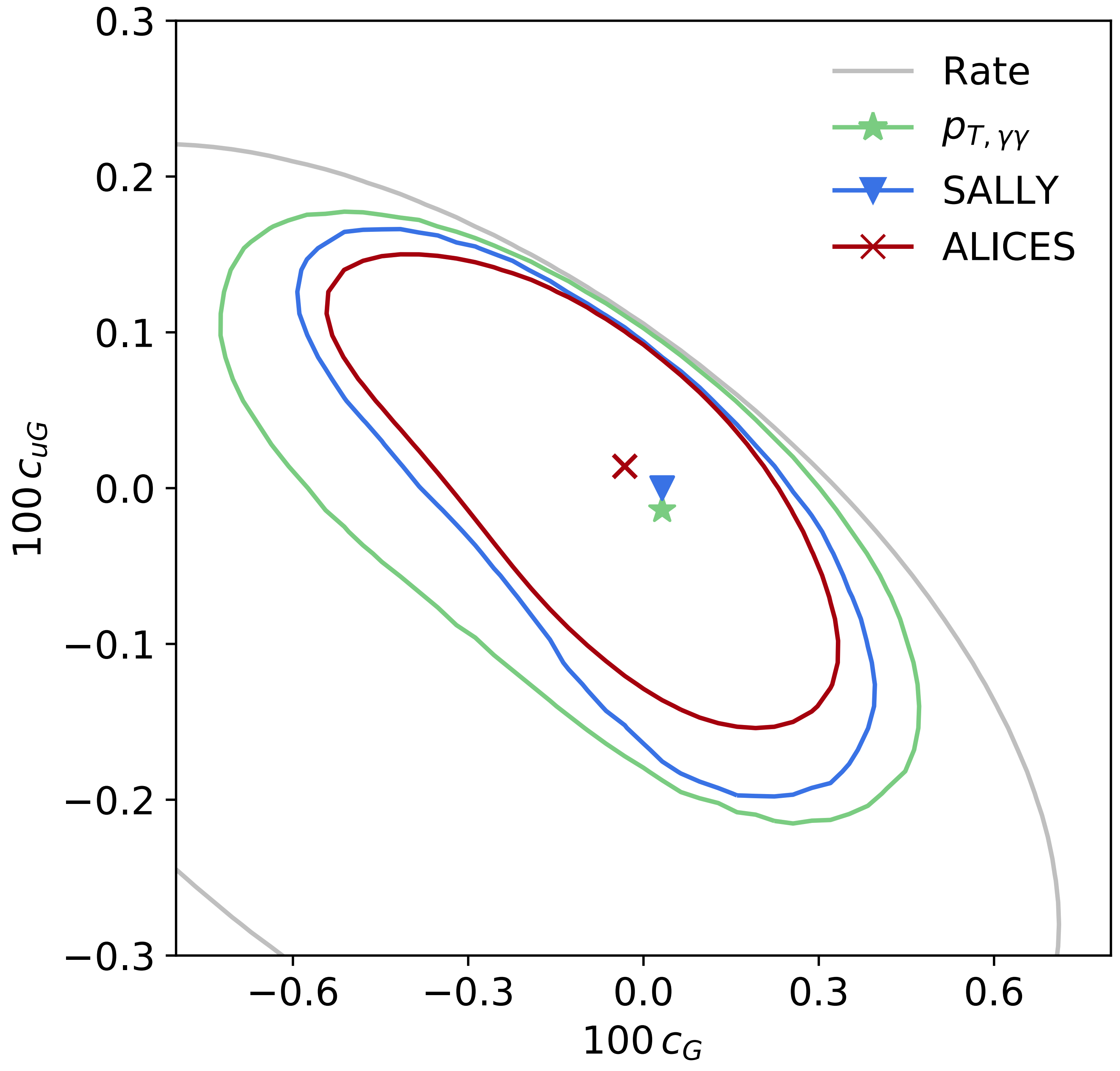
# Limit setting (frequentist, standard ATLAS / CMS practice)



# Types of Inference Results & Sensitivity Summaries



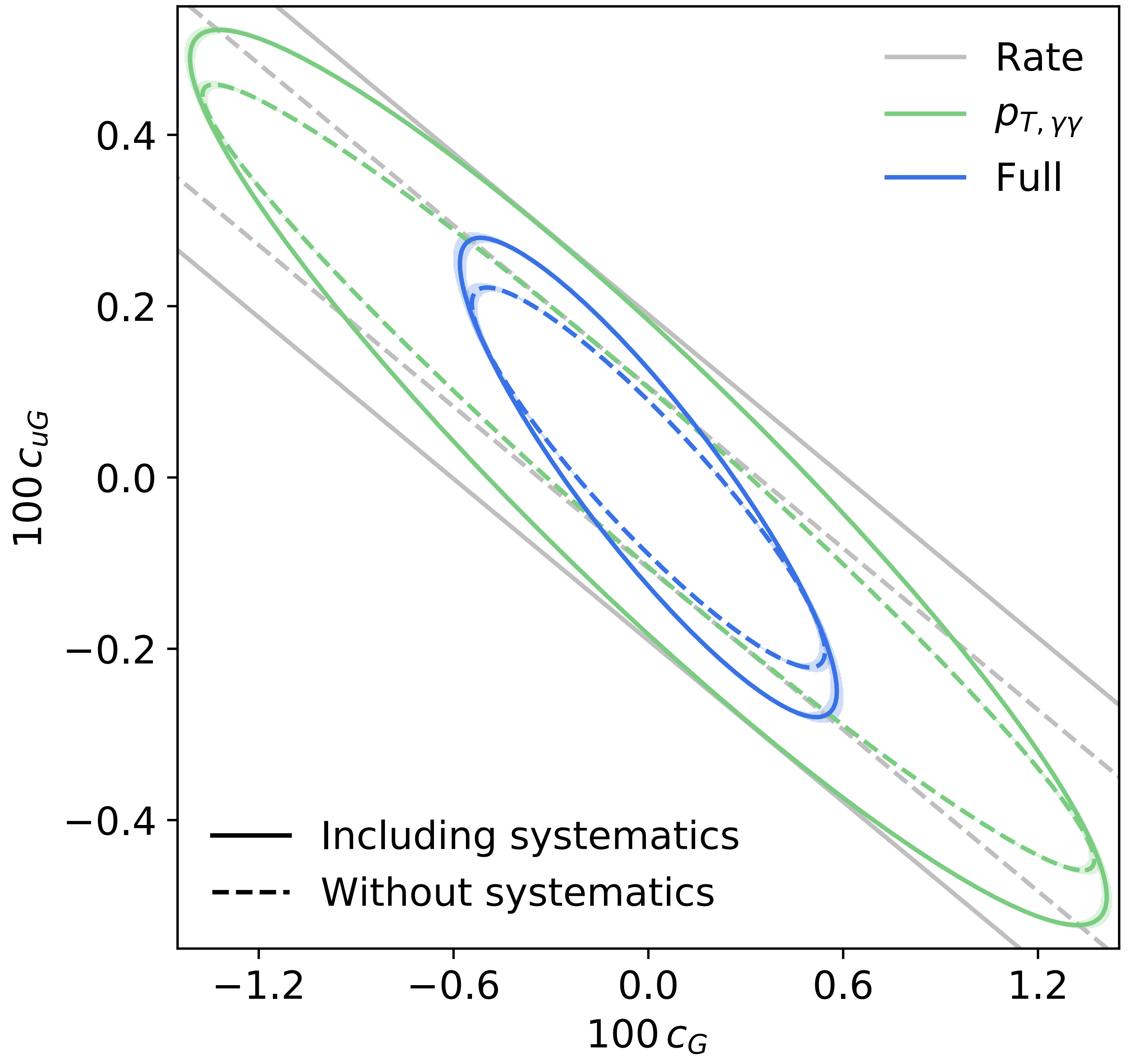

Elliptical Contours
(with / without systematics)



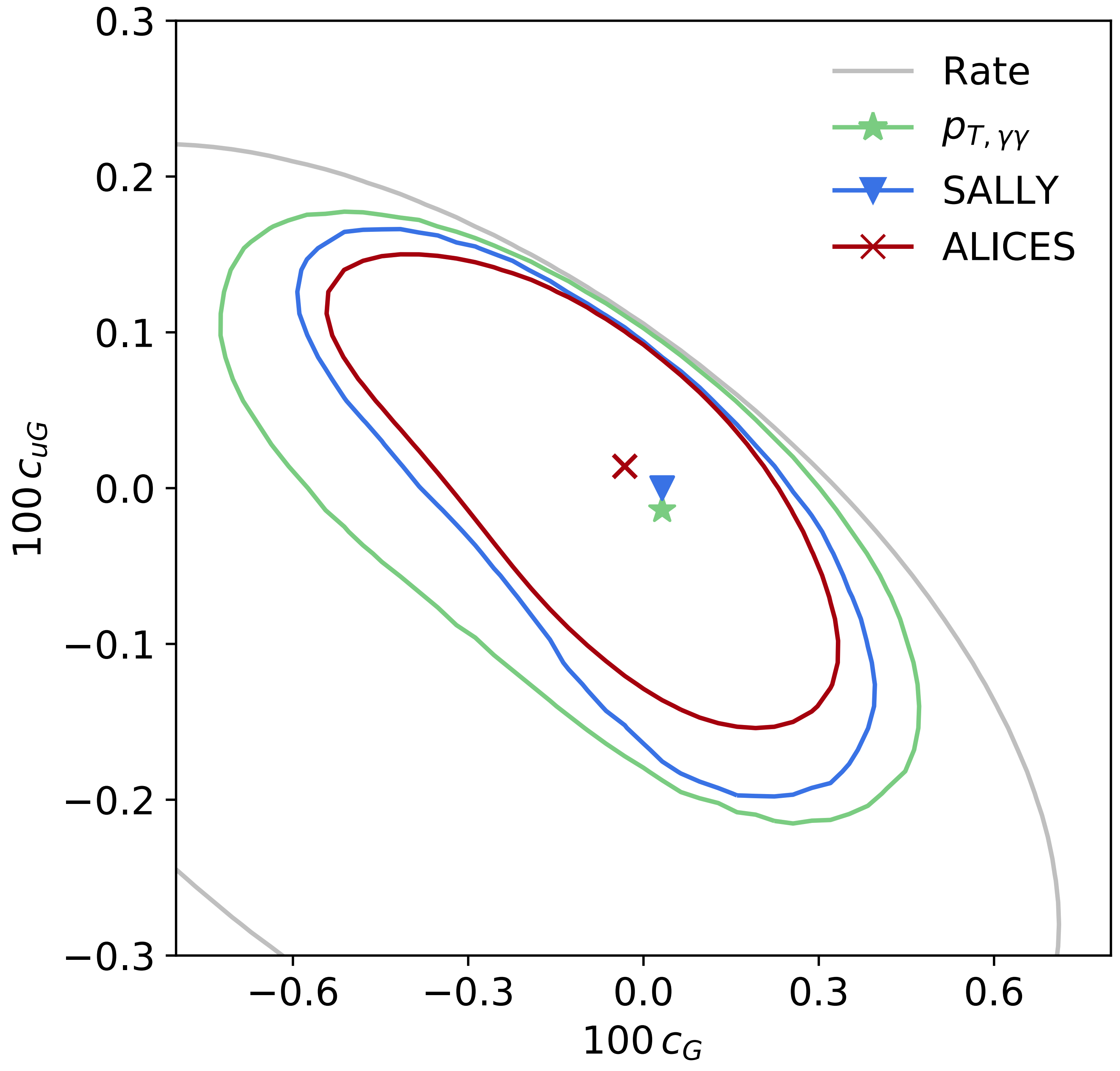

Non-Eliptical Contours

Fisher Information
&
Information Geometry

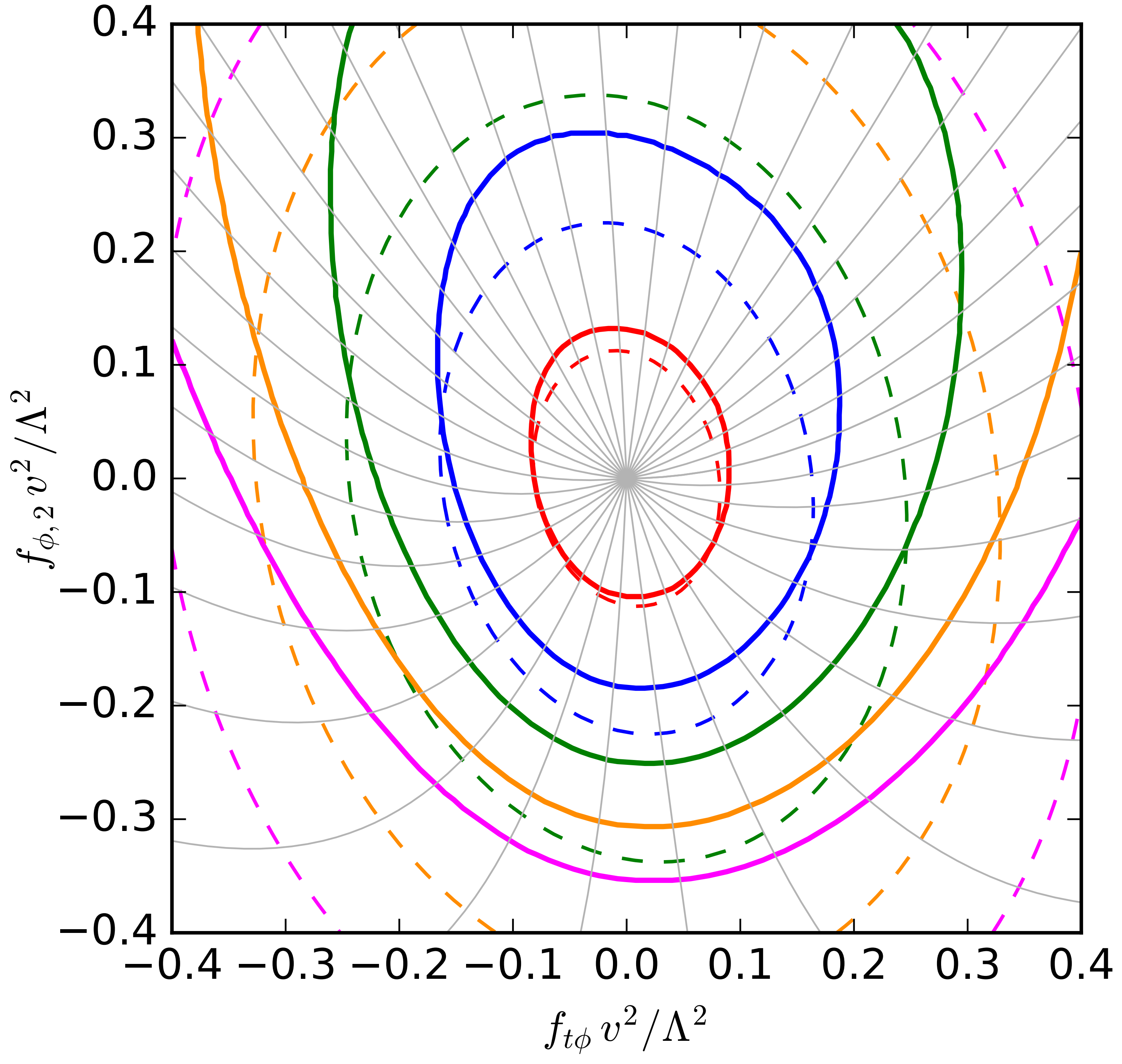
# Types of Inference Results & Sensitivity Summaries




Elliptical Contours
(with / without systematics)




Non-Eliptical Contours




Fisher Information
&
Information Geometry

MadMiner automates all of these methods.

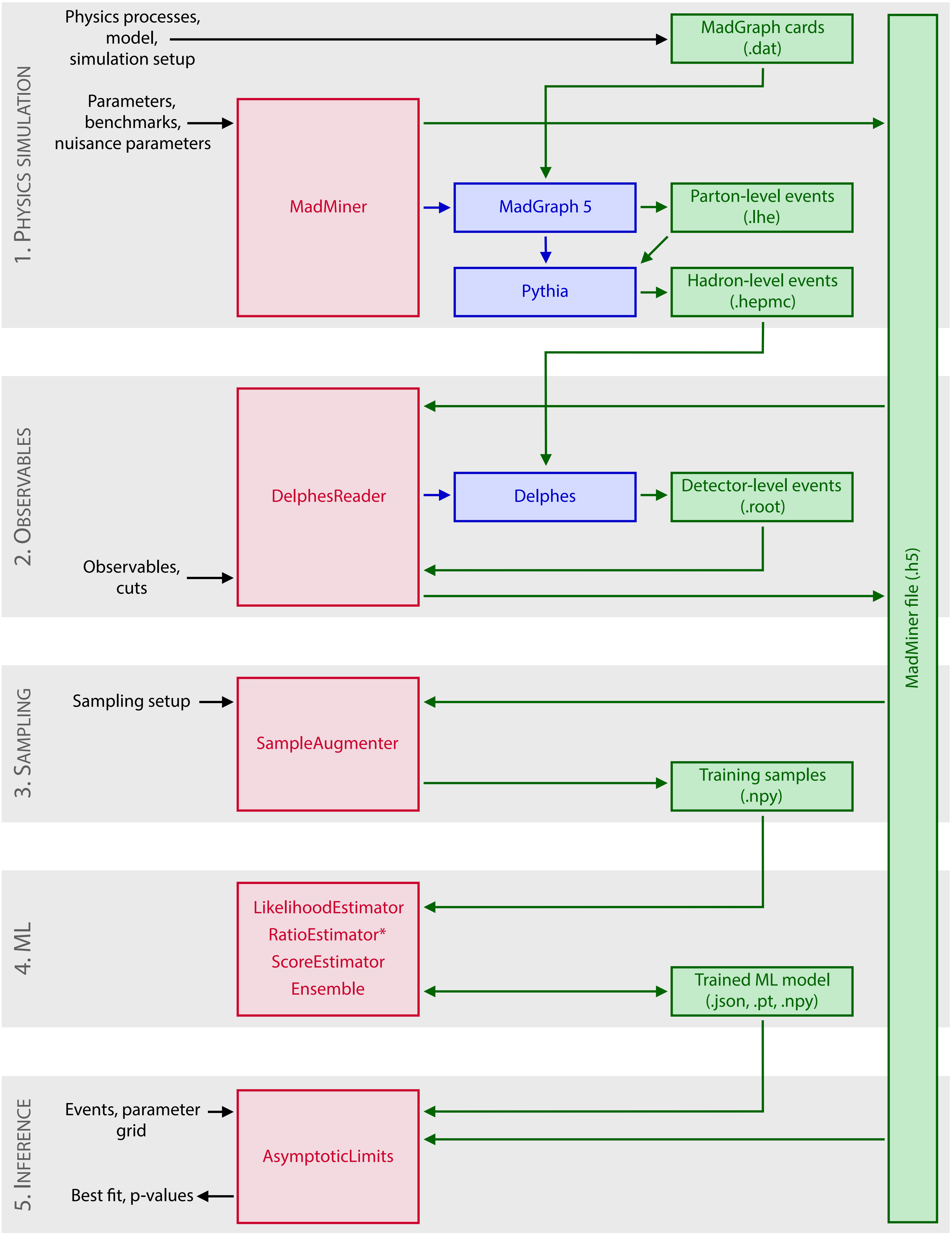
[JB, F. Kling, I. Espejo, K. Cranmer 1907.10621]

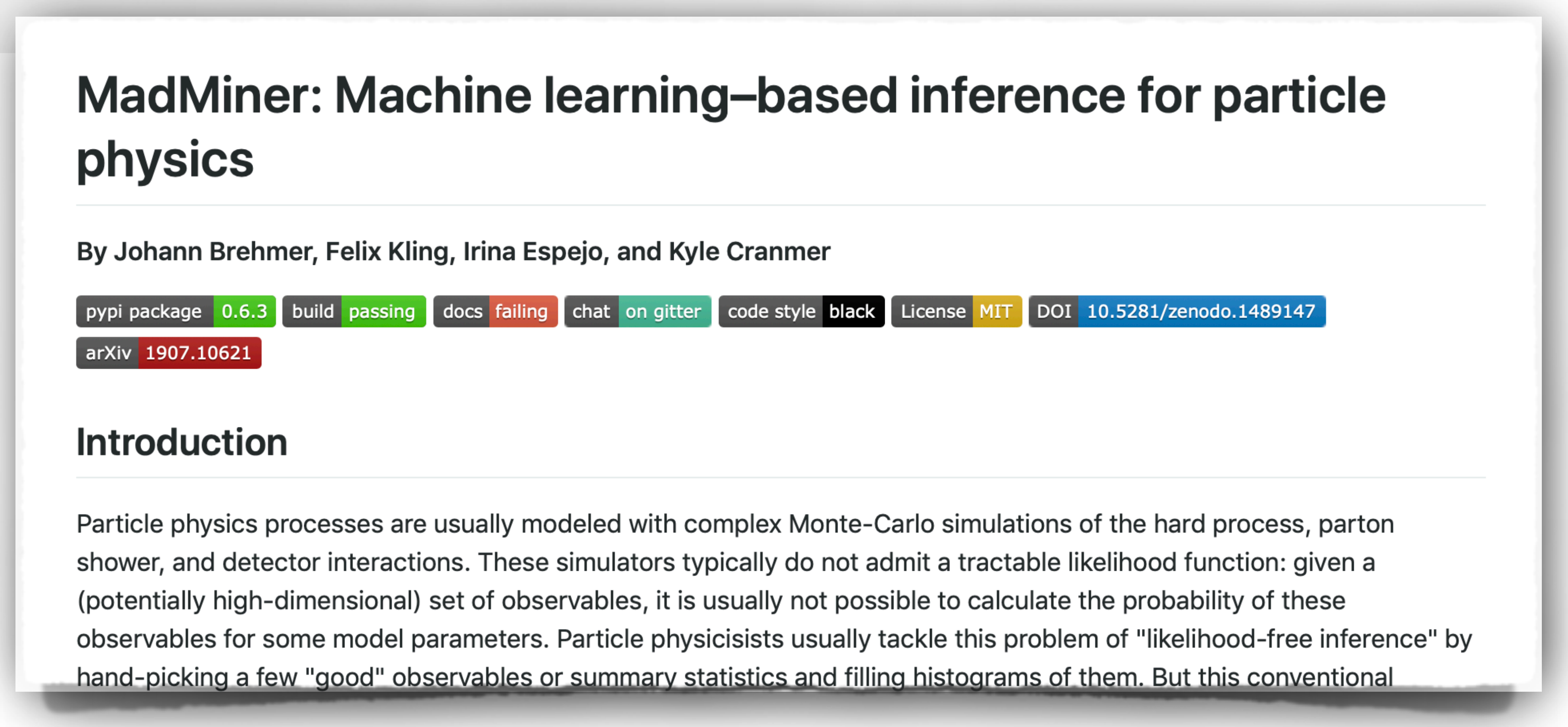
# Automation

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

New Python package MadMiner makes it straightforward to apply the new techniques to LHC problems

- Out of the box: Pheno-level analyses
  - MadGraph, Pythia, Delphes
  - Systematic uncertainties from PDF / scale variation
- Scalable to state-of-the-art experimental tools
  - Mostly requires bookkeeping of fully differential cross sections
- Modular interface
  - Extensive documentation
  - Embedded into Python / ML ecosystem





## MadMiner: Machine learning–based inference for particle physics

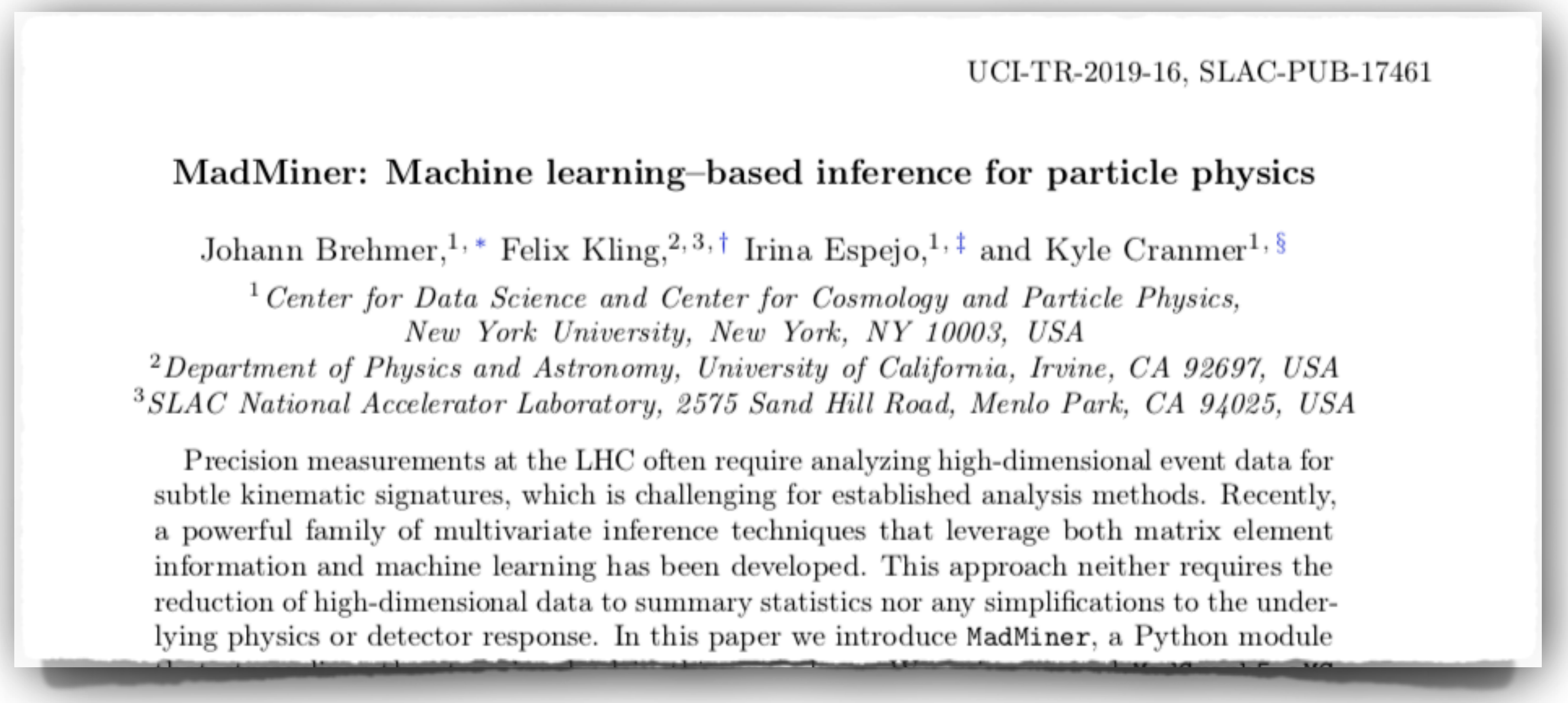
By Johann Brehmer, Felix Kling, Irina Espejo, and Kyle Cranmer

pypi package 0.6.3 | build passing | docs failing | chat on gitter | code style black | License MIT | DOI 10.5281/zenodo.1489147 | arXiv 1907.10621

### Introduction

Particle physics processes are usually modeled with complex Monte-Carlo simulations of the hard process, parton shower, and detector interactions. These simulators typically do not admit a tractable likelihood function: given a (potentially high-dimensional) set of observables, it is usually not possible to calculate the probability of these observables for some model parameters. Particle physicisists usually tackle this problem of "likelihood-free inference" by hand-picking a few "good" observables or summary statistics and filling histograms of them. But this conventional

Repository and tutorials:
github.com/johannbrehmer/madminer



UCI-TR-2019-16, SLAC-PUB-17461

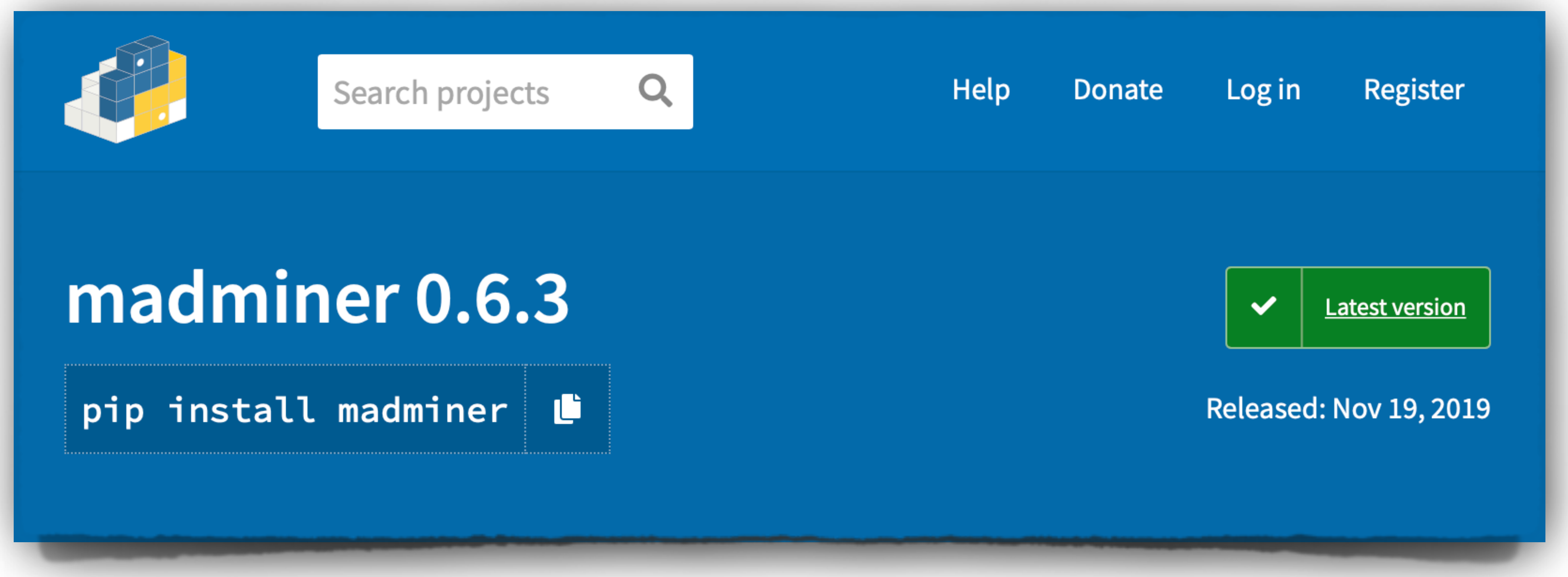
### MadMiner: Machine learning–based inference for particle physics

Johann Brehmer,[1,*] Felix Kling,[2,3,†] Irina Espejo,[1,‡] and Kyle Cranmer[1,§]

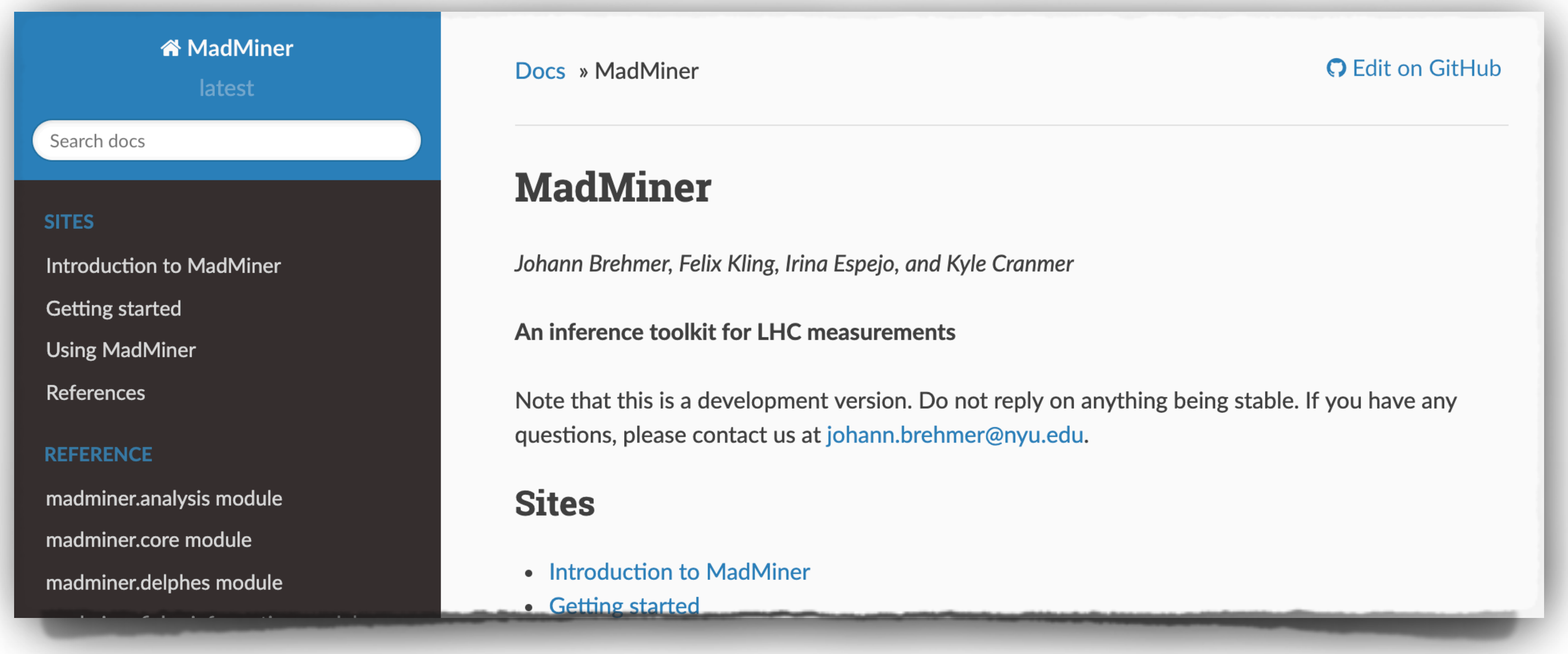
[1] *Center for Data Science and Center for Cosmology and Particle Physics, New York University, New York, NY 10003, USA*
[2] *Department of Physics and Astronomy, University of California, Irvine, CA 92697, USA*
[3] *SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA*

Precision measurements at the LHC often require analyzing high-dimensional event data for subtle kinematic signatures, which is challenging for established analysis methods. Recently, a powerful family of multivariate inference techniques that leverage both matrix element information and machine learning has been developed. This approach neither requires the reduction of high-dimensional data to summary statistics nor any simplifications to the underlying physics or detector response. In this paper we introduce `MadMiner`, a Python module

Paper with detailed explanations:
1907.10621




Installation:
`pip install madminer`




API documentation:
madminer.readthedocs.io

# Forward modeling and inverse problems

forward modeling
generation
simulation

PREDICTION

**θ**
parameters of interest

**p( x, z | θ, ν )**

**x**
observed data
simulated data

**z**
latent variables

**ν**
nuisance parameters

INFERENCE

inverse problem
measurement
parameter estimation

# Forward modeling and inverse problems

forward modeling
generation
simulation

PREDICTION

**θ**
parameters of interest

**p( x, z | θ, ν )**

**x** → **s(x)**

**z**
latent variables

observed data
simulated data

summary statistic
engineered feature

**ν**
nuisance parameters

INFERENCE

inverse problem
measurement
parameter estimation

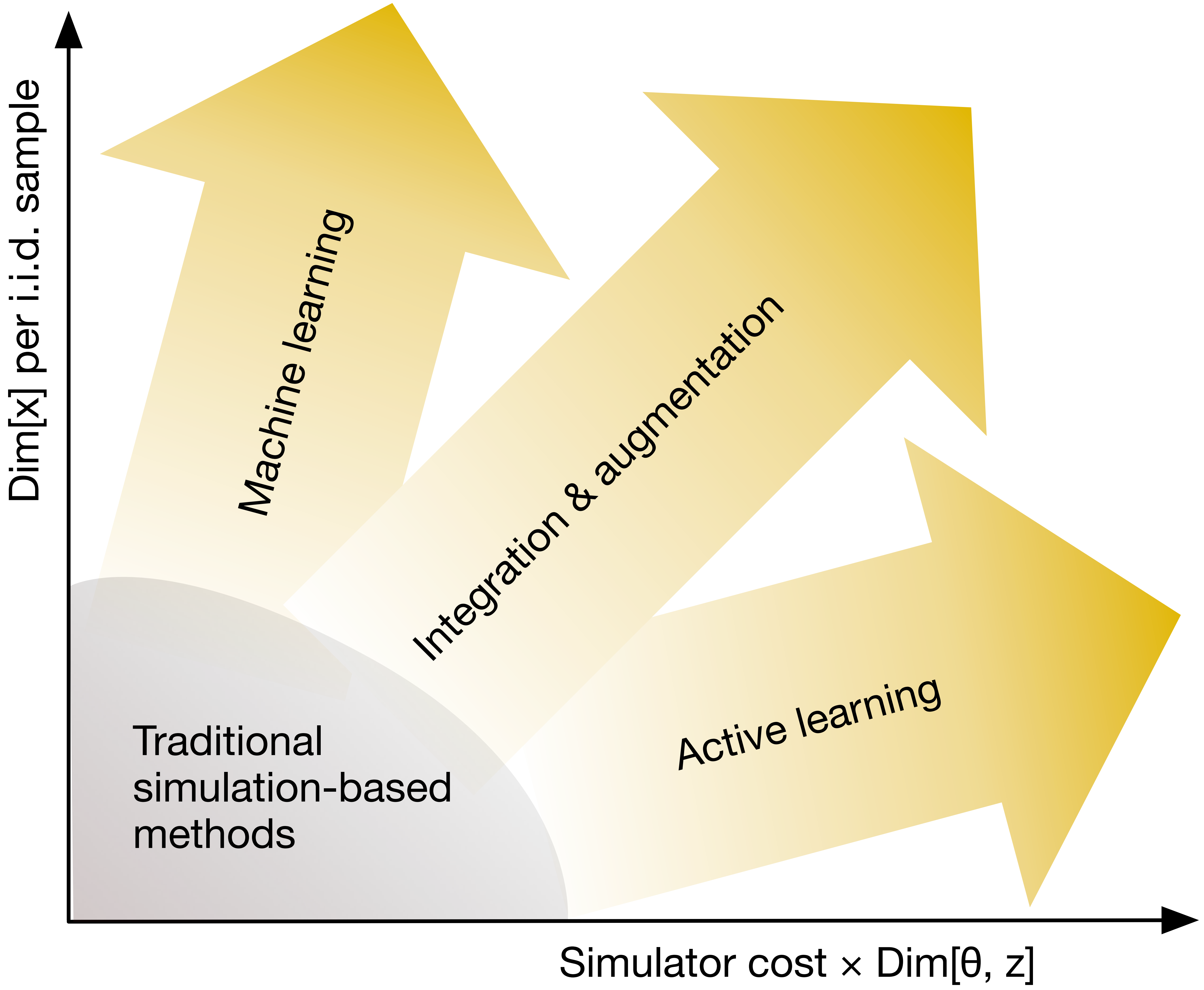
Published in Proceedings of the National Academy of Sciences

# 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, USA; [b]Center for Data Science, New York University, USA; [c]Montefiore Institute, University of Liège, Belgium

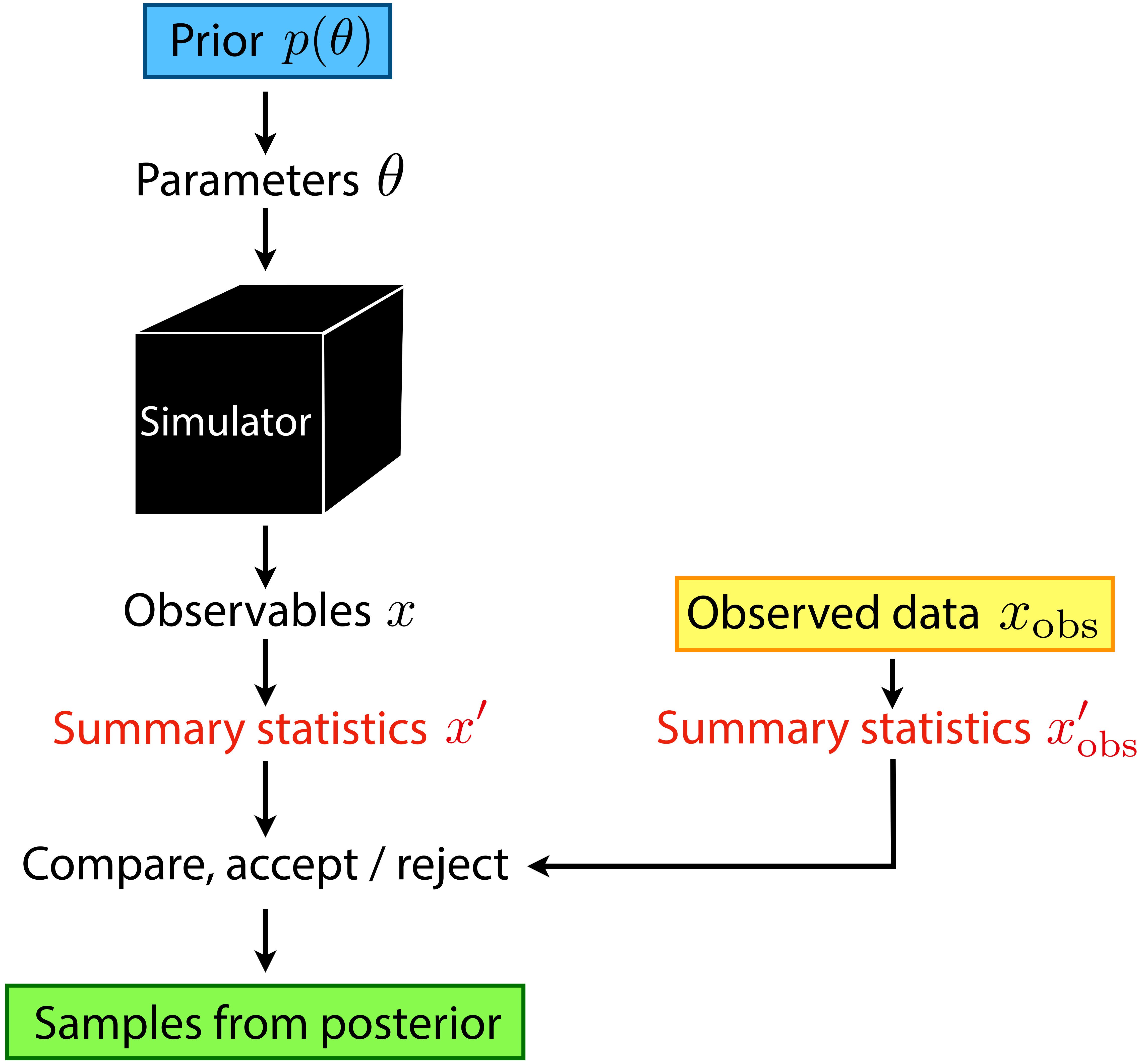
April 3, 2020




[PNAS (2020), arXiv:1911.01429; DOI: 10.1073/pnas.1912789117]

# Approximate Bayesian Computation (ABC)

[D. Rubin 1984]

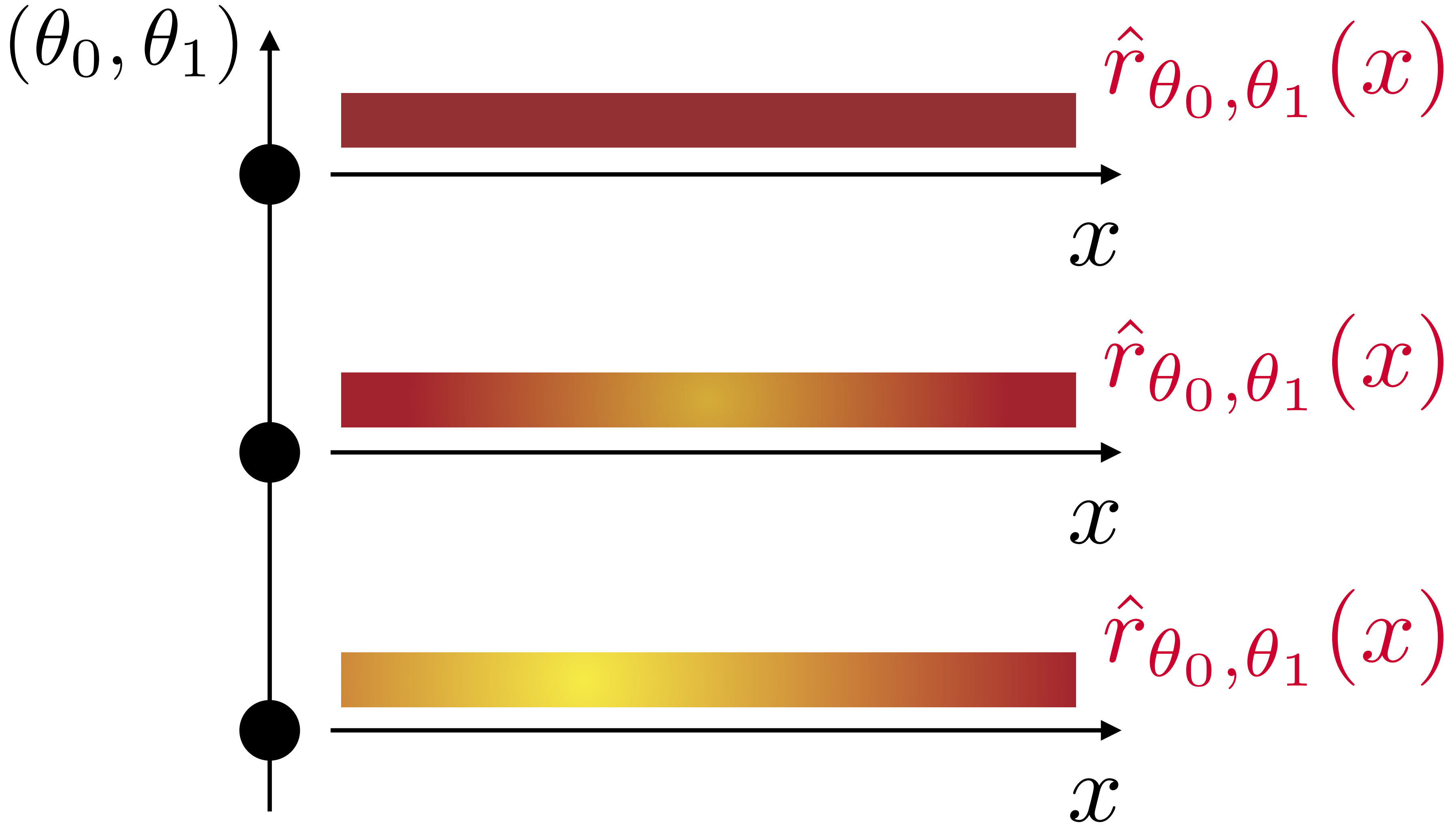



- Compression to summary statistics and acceptance threshold reduce quality of inference
- Rejection algorithm can be very sample inefficient

# Two types of likelihood ratio estimators

## A) Point by point:

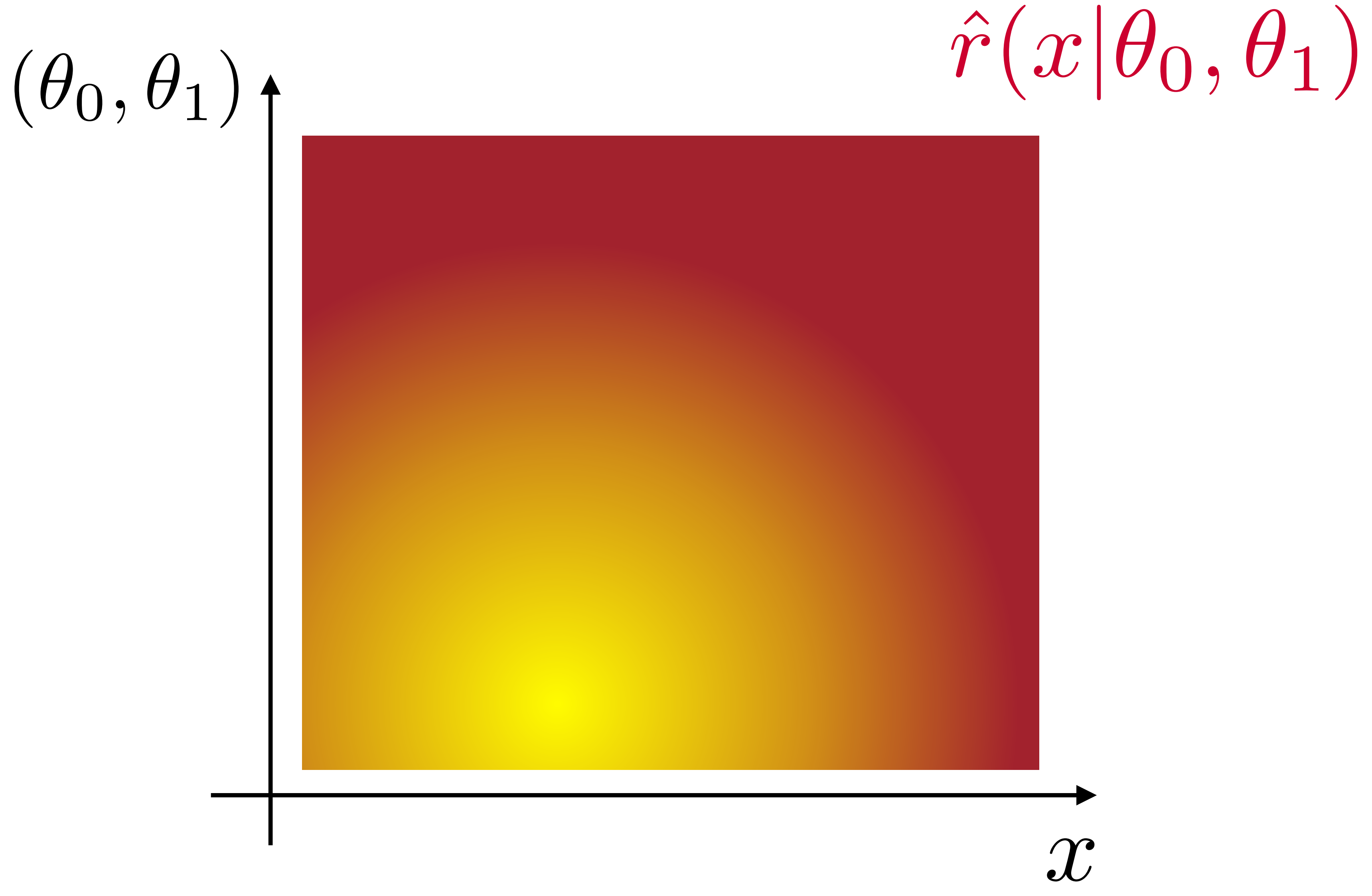
- first, define grid of parameter points $\{(\theta_0, \theta_1)\}$
- for each combination $(\theta_0, \theta_1)$ ,
  create separate estimator $\hat{r}_{\theta_0,\theta_1}(x)$
- final results can be interpolated between grid points



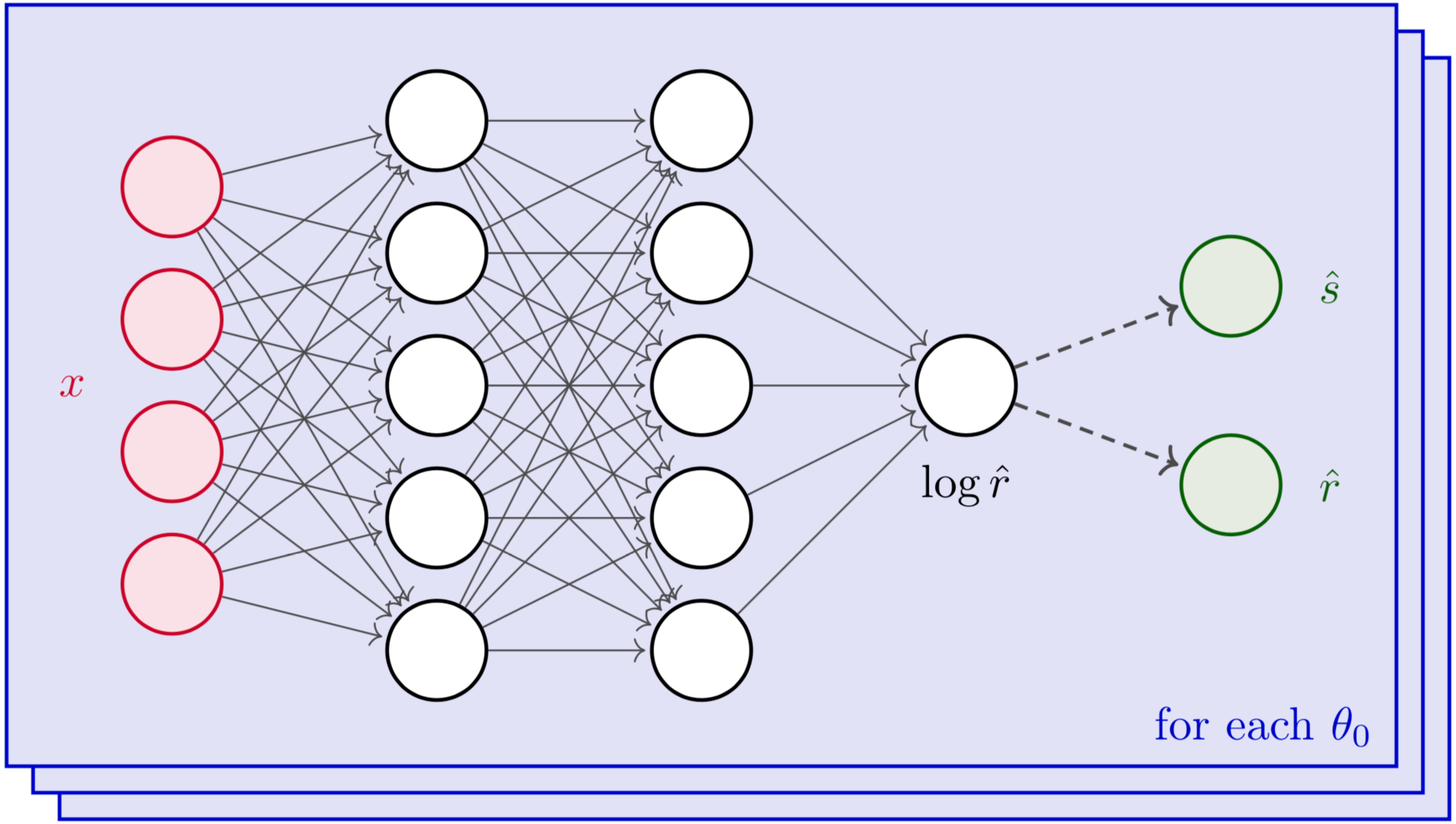

## B) Parameterized:

[K. Cranmer, J. Pavez, G. Louppe 1506.02169;
P. Baldi et al. 1601.07913]

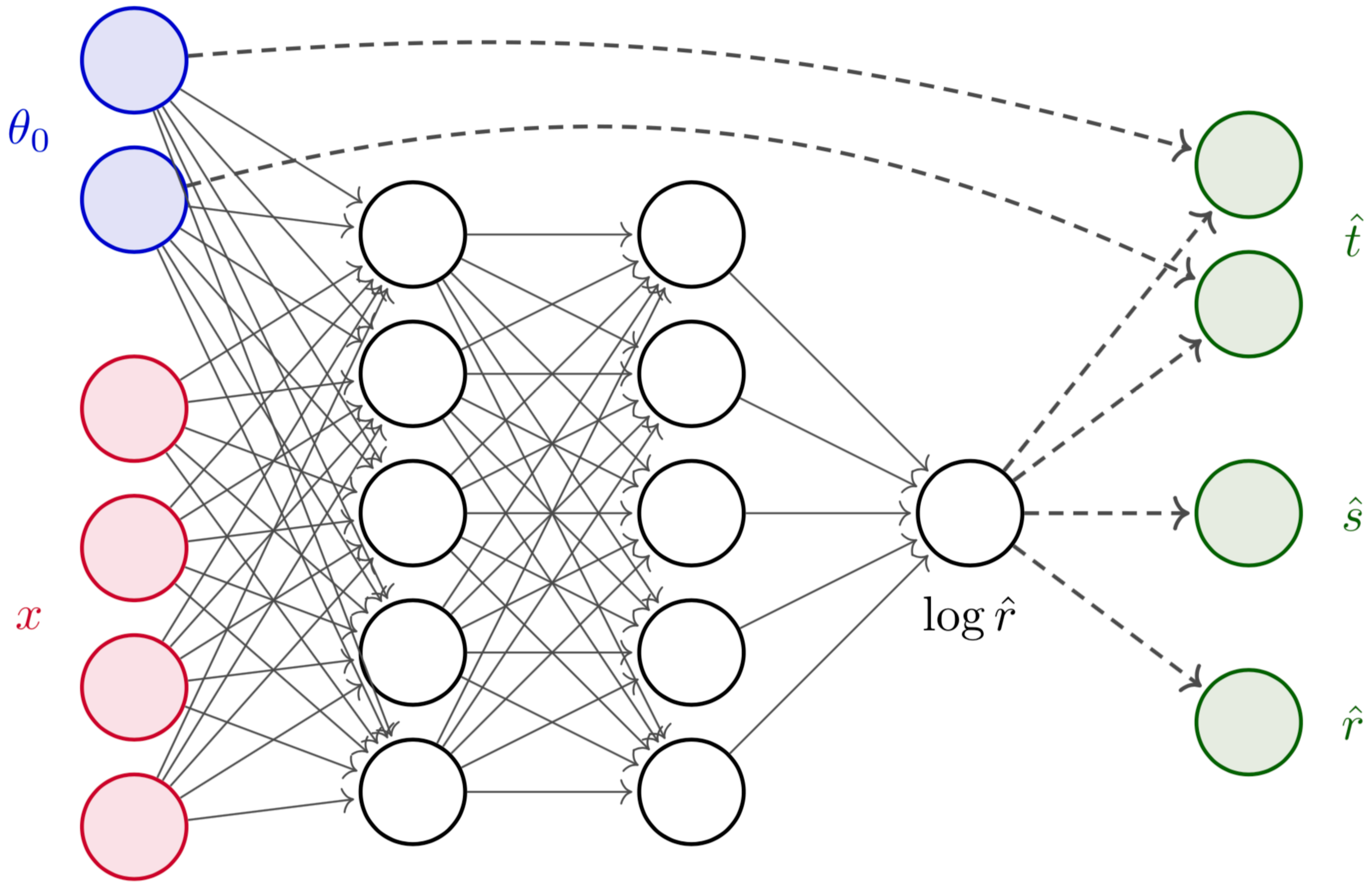
- create one estimator $\hat{r}(x|\theta_0, \theta_1)$
  that is a function of $\theta_0$ and $\theta_1$
- no further interpolation necessary
- "borrows information" from close points



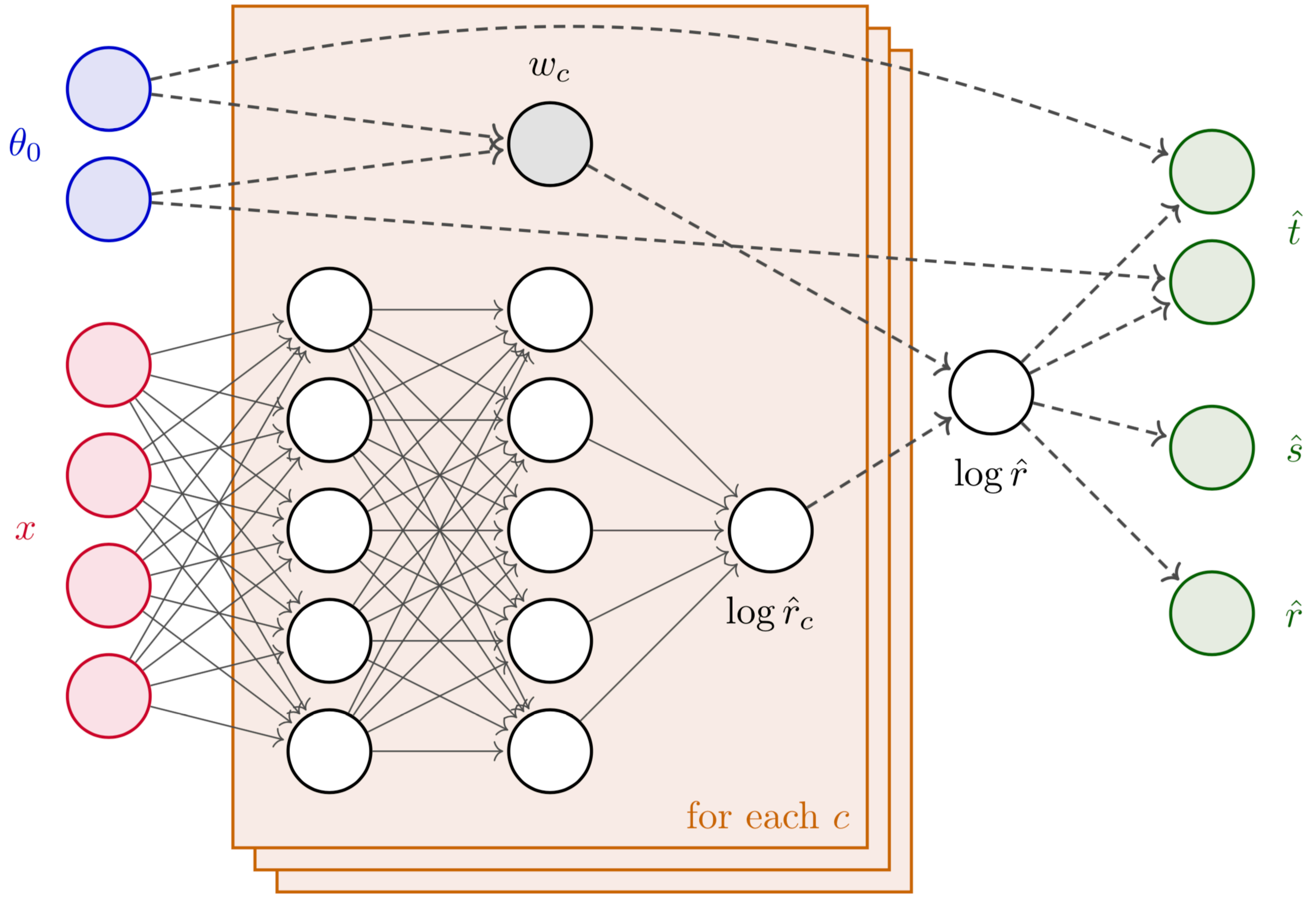
# Point by point



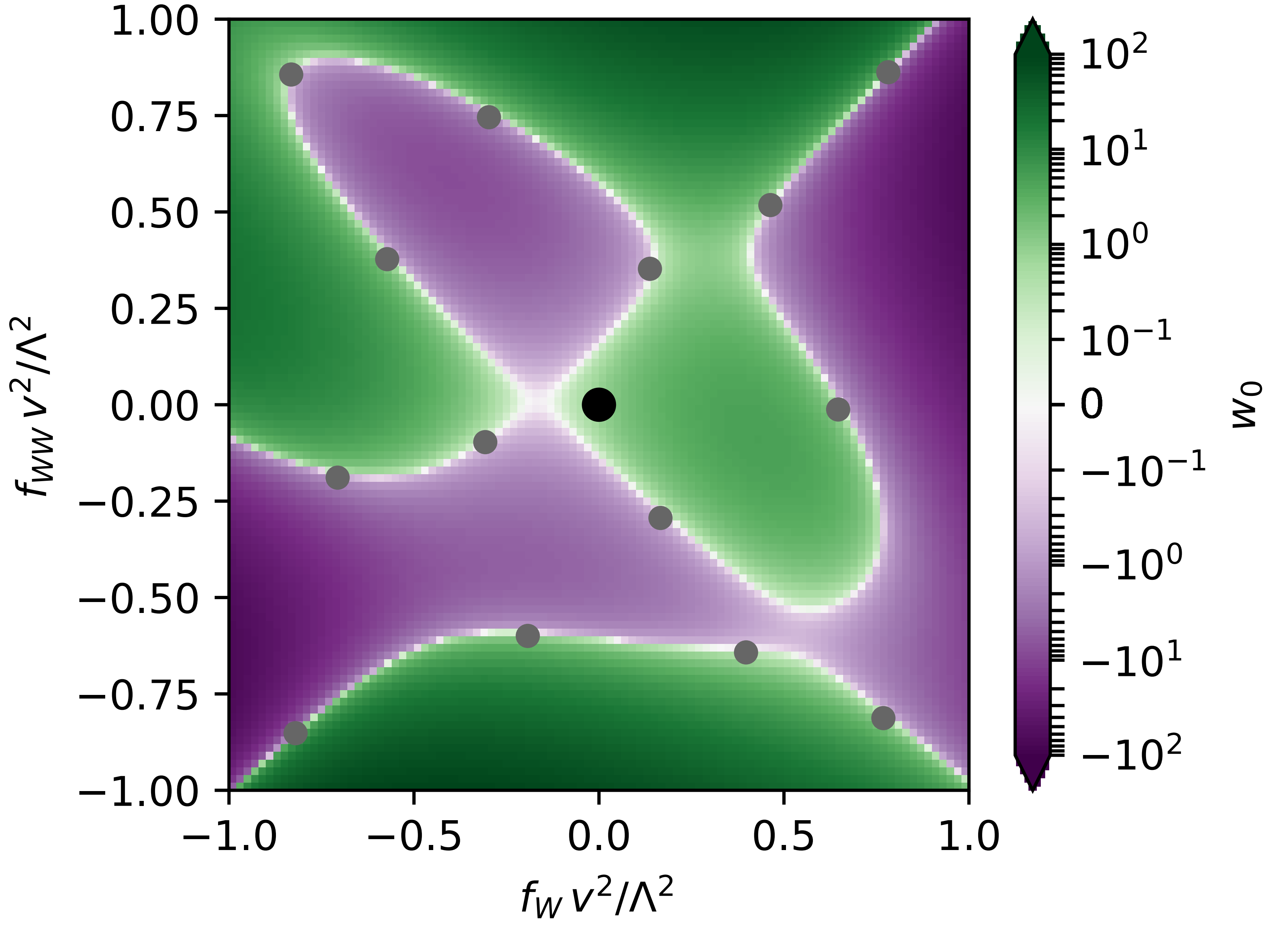
# (Physics-agnostic) parameterized estimators



# Morphing-aware parameterized estimators



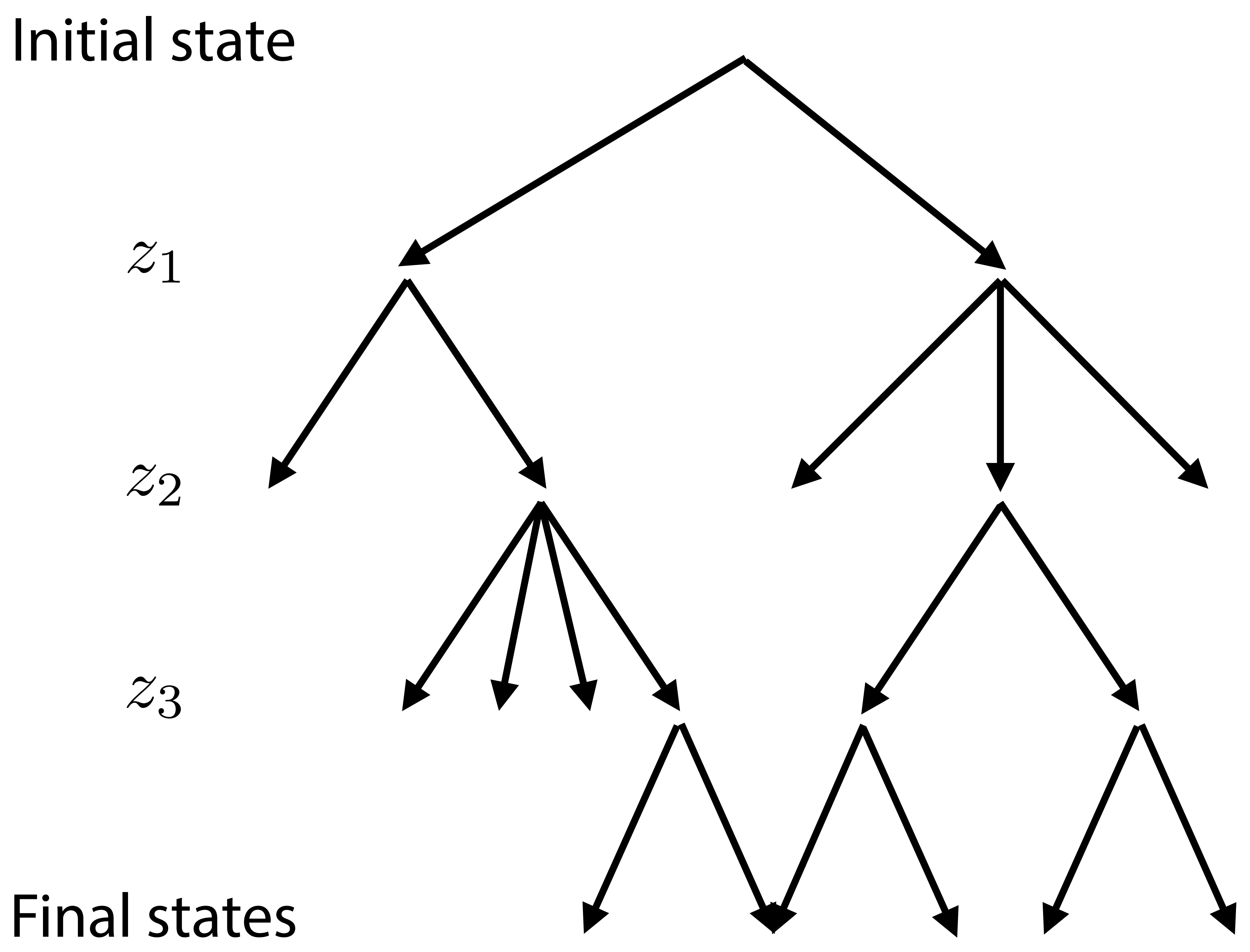

$$\hat{r}(x|\theta_0, \theta_1) = \sum_c w_c(\theta_0)\, \hat{r}_c(x)$$



# Mining gold from any simulation

- Computer simulation typically evolve along a tree-like structure of successive random branchings
- The probabilities of each branching $p_i(z_i|z_{i-1}, \theta)$ are often clearly defined in the code:

```
if random() > 0.1 + 2.5 * model_parameter:
    do_one_thing()
else:
    do_another_thing()
```



# Mining gold from any simulation

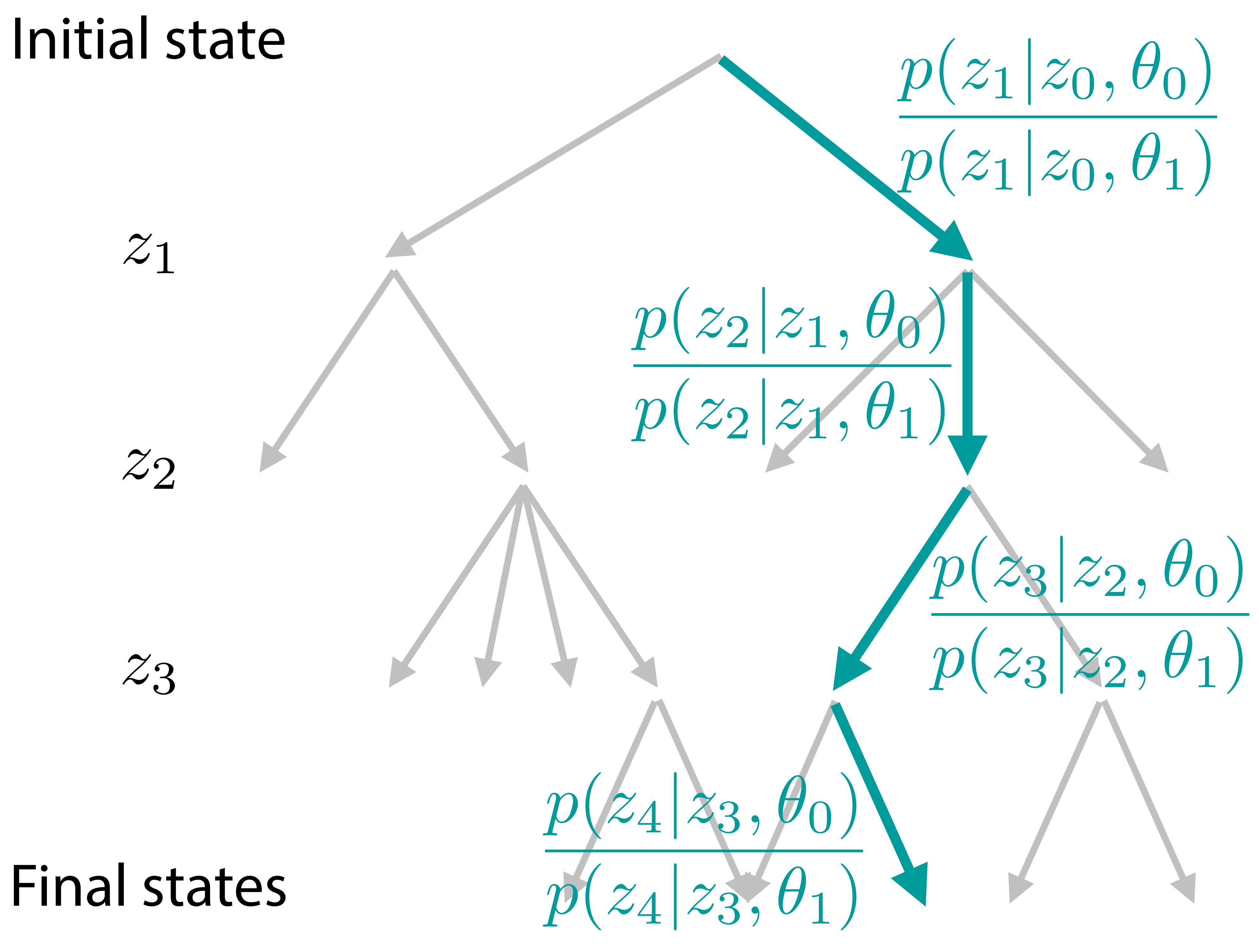
- Computer simulation typically evolve along a tree-like structure of successive random branchings

- The probabilities of each branching $p_i(z_i|z_{i-1},\theta)$ are often clearly defined in the code:

```
if random() > 0.1 + 2.5 * model_parameter:
    do_one_thing()
else:
    do_another_thing()
```

- For each run of the simulator, we can calculate the probability of the chosen path for different values of the parameters, and the "joint likelihood ratio":

$$r(x,z|\theta_0,\theta_1) = \frac{p(x,z|\theta_0)}{p(x,z|\theta_1)} = \prod_i \frac{p(z_i|z_{i-1},\theta_0)}{p(z_i|z_{i-1},\theta_1)}$$



# Mining gold: A family of new inference techniques

| Method | Simulate | Extract $r(x,z)$ | Extract $t(x,z)$ | NN estimates | Asympt. exact | Generative |
|---|---|---|---|---|---|---|
| ROLR | $\theta_0 \sim \pi(\theta), \theta_1$ | ✓ | | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| CASCAL | $\theta_0 \sim \pi(\theta), \theta_1$ | | ✓ | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| ALICE | $\theta_0 \sim \pi(\theta), \theta_1$ | | ✓ | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| RASCAL | $\theta_0 \sim \pi(\theta), \theta_1$ | ✓ | ✓ | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| ALICES | $\theta_0 \sim \pi(\theta), \theta_1$ | ✓ | ✓ | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| SCANDAL | $\theta \sim \pi(\theta)$ | | ✓ | $\hat{p}(x\|\theta)$ | ✓ | ✓ |
| SALLY | $\theta_{\text{ref}}$ | | ✓ | $\hat{t}(x\|\theta_{\text{ref}})$ | in local approx. | |
| SALLINO | $\theta_{\text{ref}}$ | | ✓ | $\hat{t}(x\|\theta_{\text{ref}})$ | in local approx. | |

# Mining gold: A family of new inference techniques

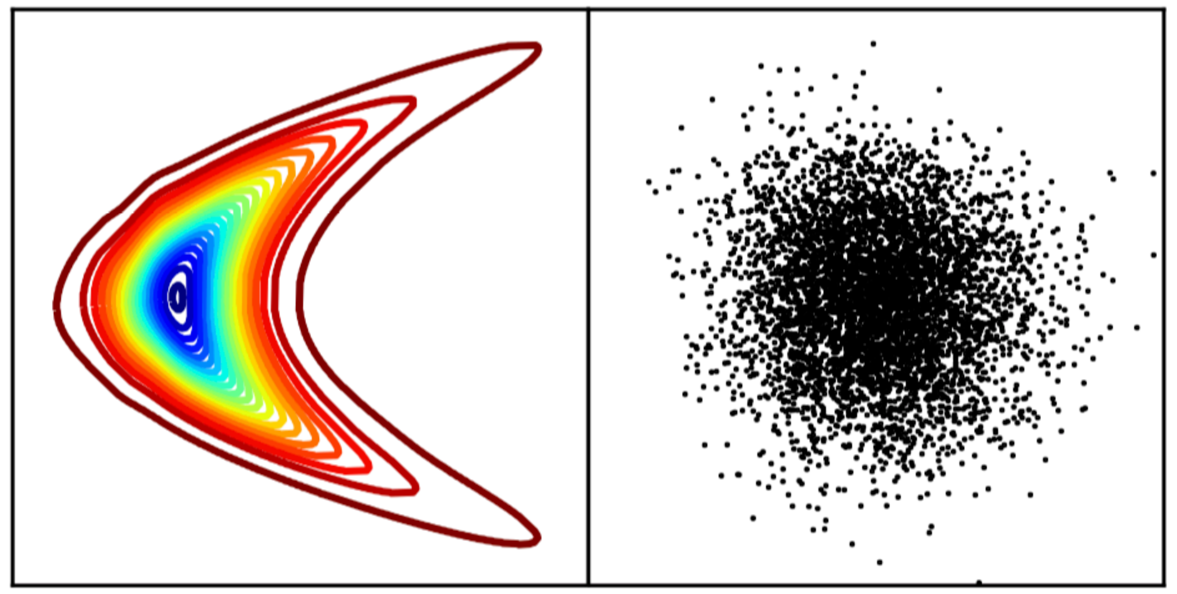
| Method | Simulate | Extract $r(x,z)$ | Extract $t(x,z)$ | NN estimates | Asympt. exact | Generative |
|---|---|---|---|---|---|---|
| ROLR | $\theta_0 \sim \pi(\theta), \theta_1$ | ✓ | | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| CASCAL | $\theta_0 \sim \pi(\theta), \theta_1$ | | ✓ | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| ALICE | $\theta_0 \sim \pi(\theta), \theta_1$ | | ✓ | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| RASCAL | $\theta_0 \sim \pi(\theta), \theta_1$ | ✓ | ✓ | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| ALICES | $\theta_0 \sim \pi(\theta), \theta_1$ | ✓ | ✓ | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| SCANDAL | $\theta \sim \pi(\theta)$ | | ✓ | $\hat{p}(x\|\theta)$ | ✓ | ✓ |
| SALLY | $\theta_{\text{ref}}$ | | ✓ | $\hat{t}(x\|\theta_{\text{ref}})$ | in local approx. | |
| SALLINO | $\theta_{\text{ref}}$ | | ✓ | $\hat{t}(x\|\theta_{\text{ref}})$ | in local approx. | |

Performance gains with cross-entropy-based loss

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

# Mining gold: A family of new inference techniques

| Method | Simulate | Extract $r(x,z)$ | Extract $t(x,z)$ | NN estimates | Asympt. exact | Generative |
|---|---|---|---|---|---|---|
| ROLR | $\theta_0 \sim \pi(\theta), \theta_1$ | ✓ | | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| CASCAL | $\theta_0 \sim \pi(\theta), \theta_1$ | | ✓ | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| ALICE | $\theta_0 \sim \pi(\theta), \theta_1$ | | ✓ | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| RASCAL | $\theta_0 \sim \pi(\theta), \theta_1$ | ✓ | ✓ | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| ALICES | $\theta_0 \sim \pi(\theta), \theta_1$ | ✓ | ✓ | $\hat{r}(x\|\theta_0, \theta_1)$ | ✓ | |
| SCANDAL | $\theta \sim \pi(\theta)$ | | ✓ | $\hat{p}(x\|\theta)$ | ✓ | ✓ |
| SALLY | $\theta_{\text{ref}}$ | | ✓ | $\hat{t}(x\|\theta_{\text{ref}})$ | in local approx. | |
| SALLINO | $\theta_{\text{ref}}$ | | ✓ | $\hat{t}(x\|\theta_{\text{ref}})$ | in local approx. | |



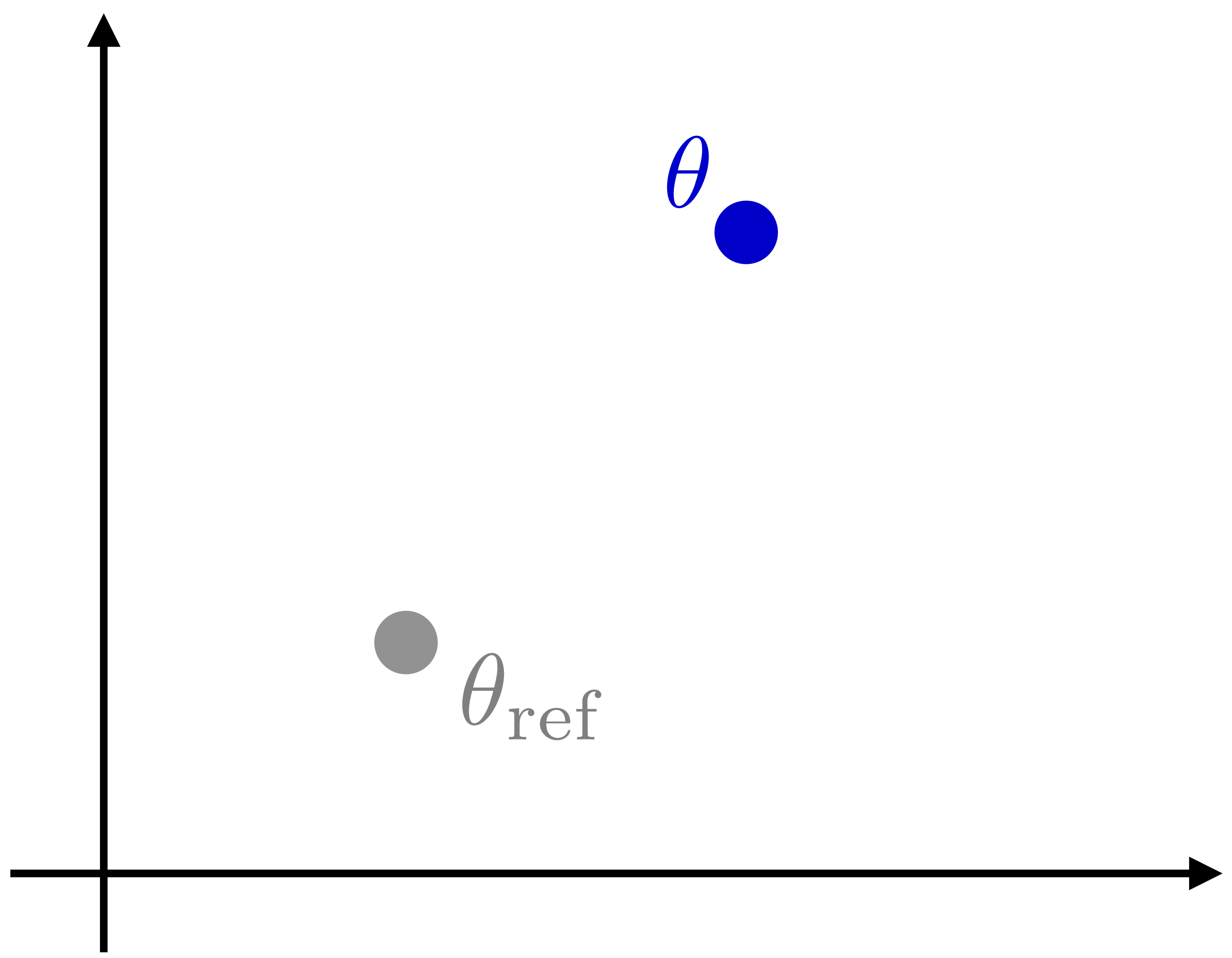
Combination with state-of-the-art conditional neural density estimators, e.g. normalizing flows

[everything by G. Papamakarios: G. Papamakarios, T. Pavlakou, I. Murray 1705.07057; G. Papamakarios, D. Sterratt, I. Murray 1805.07226; …]

# Limit setting (frequentist, standard ATLAS / CMS practice)

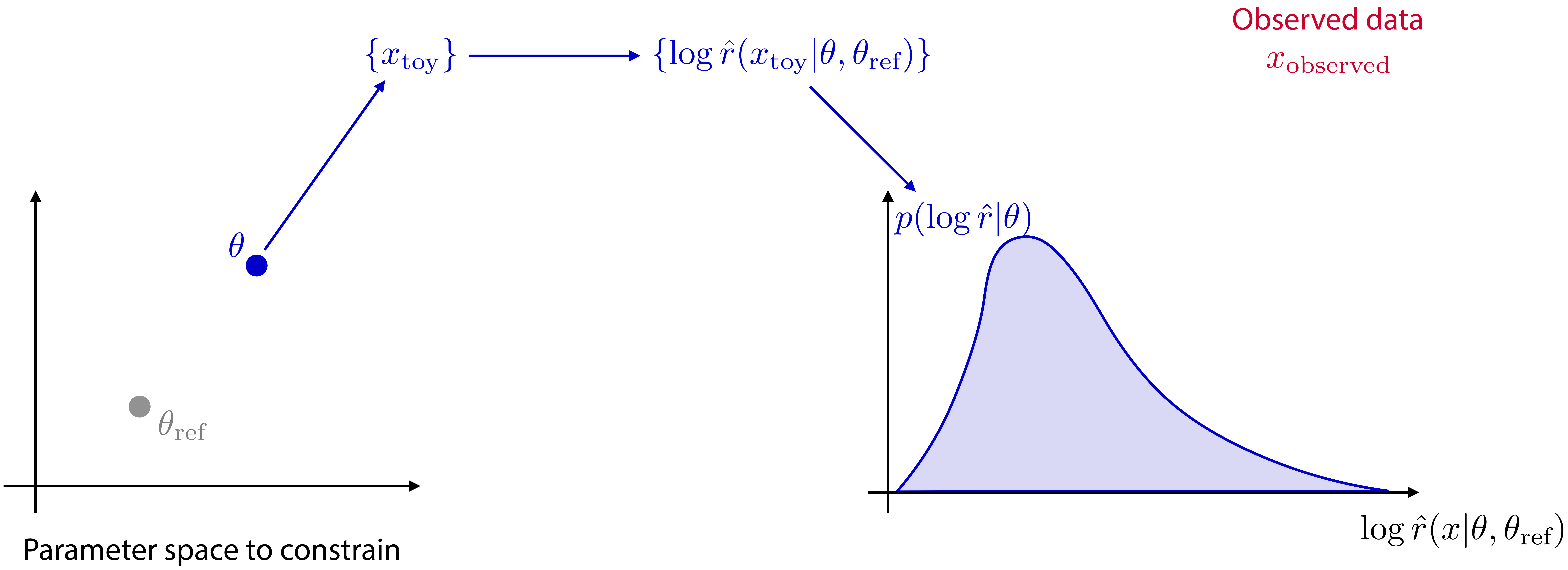
Observed data

$x_{\text{observed}}$

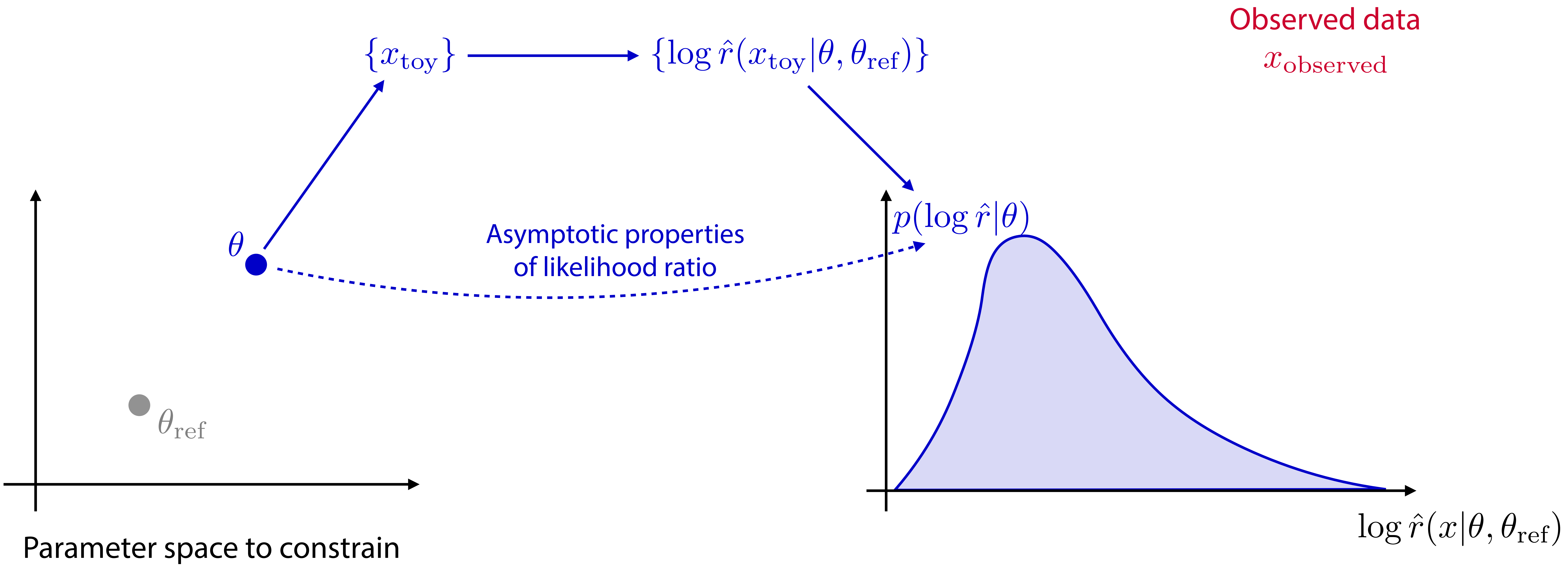



Parameter space to constrain

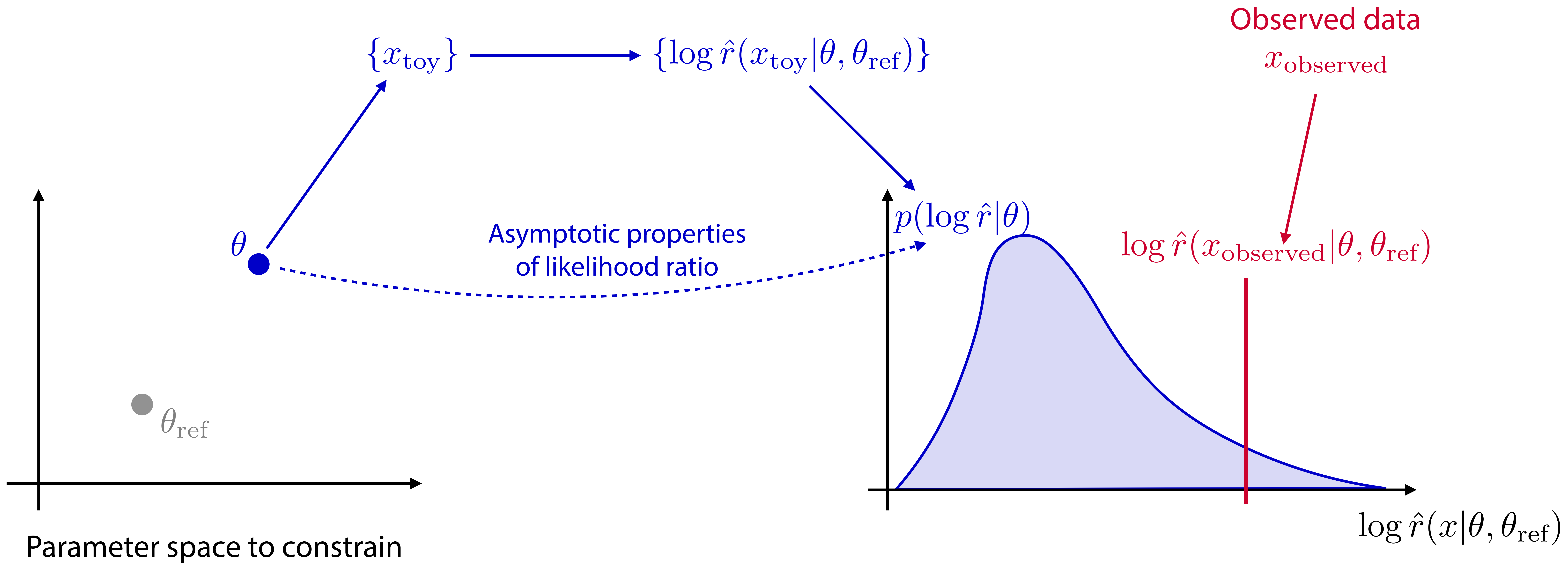
# Limit setting (frequentist, standard ATLAS / CMS practice)



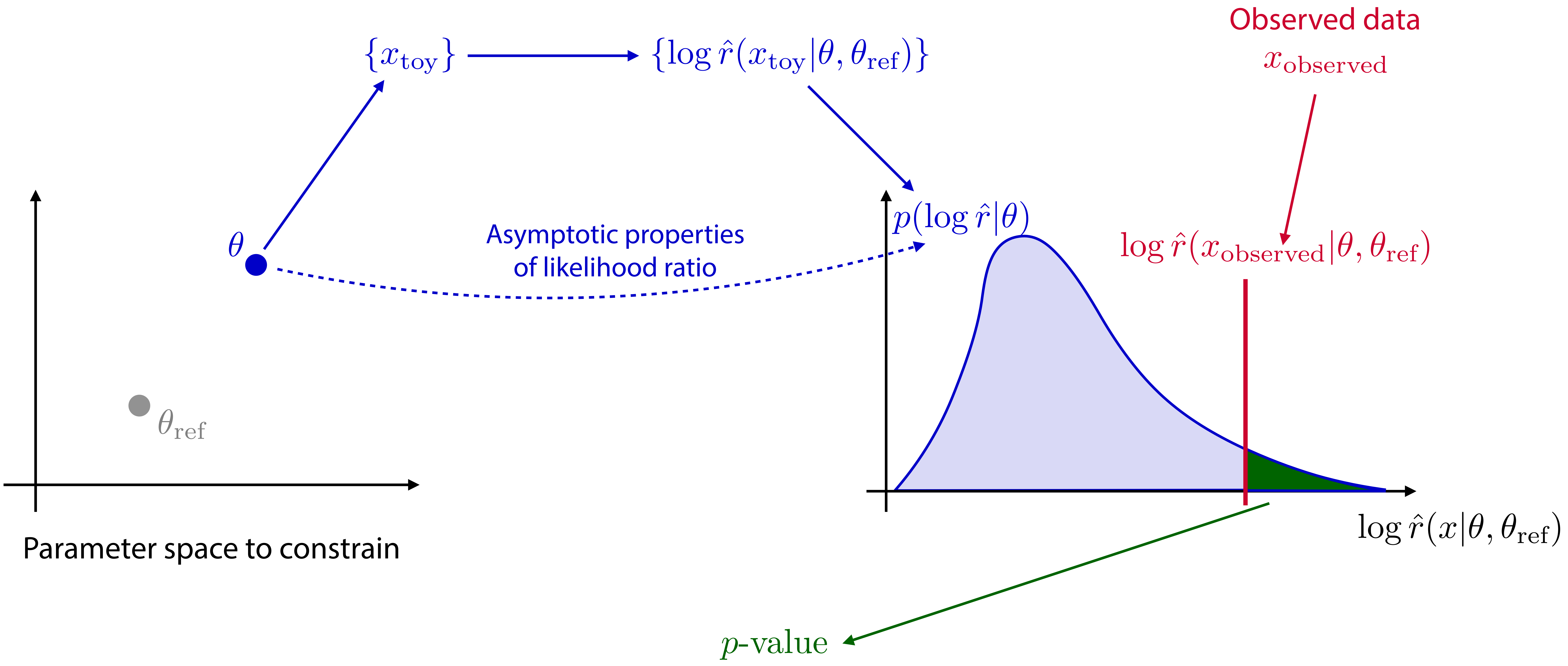
# Limit setting (frequentist, standard ATLAS / CMS practice)



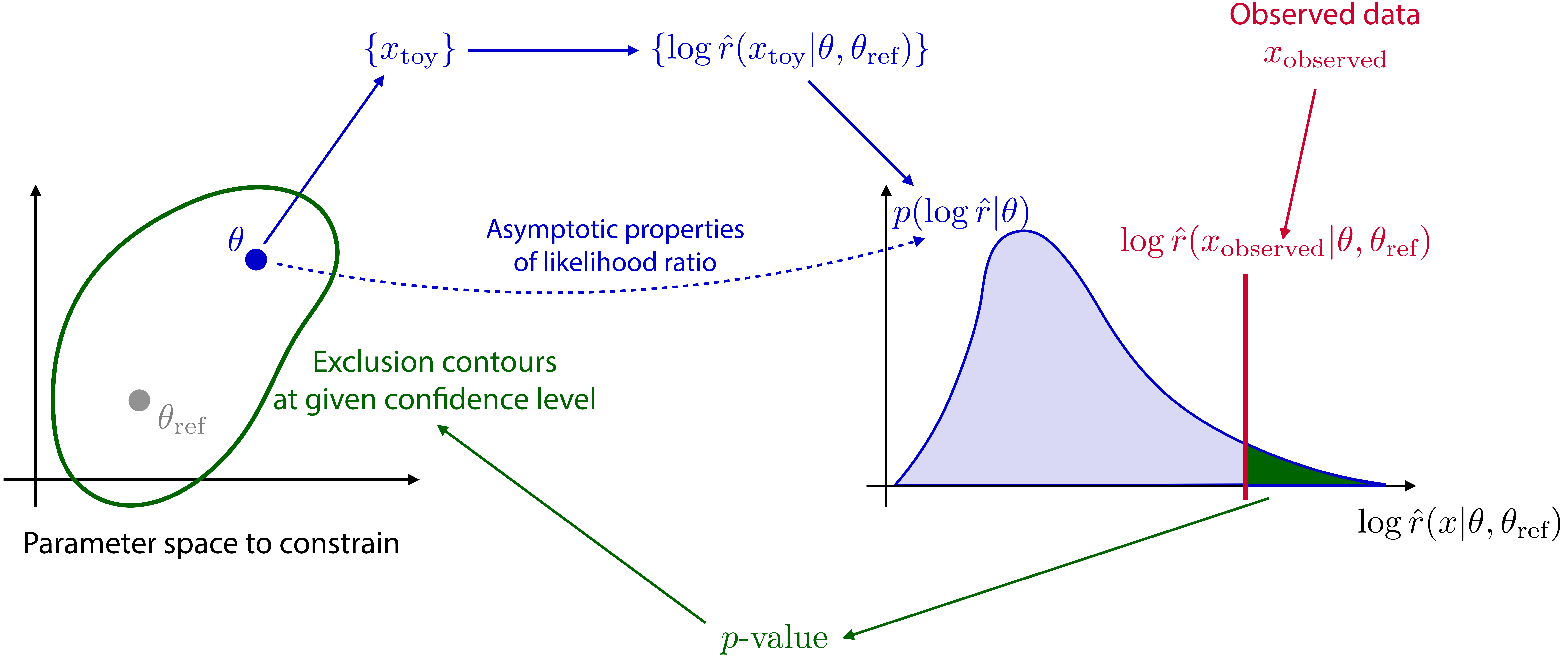
# Limit setting (frequentist, standard ATLAS / CMS practice)



# Limit setting (frequentist, standard ATLAS / CMS practice)



# Limit setting (frequentist, standard ATLAS / CMS practice)

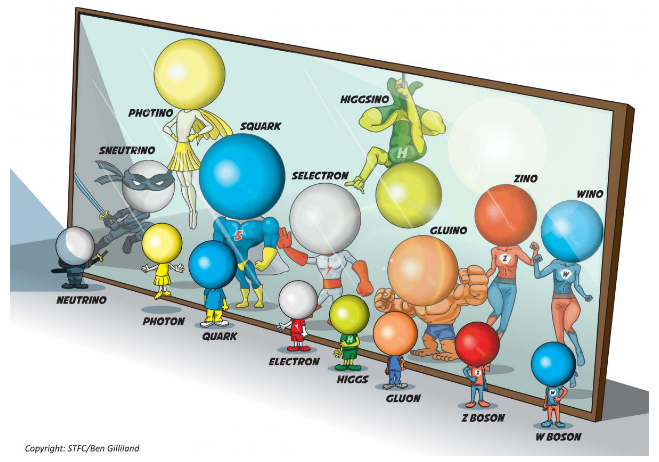


# Effective field theory

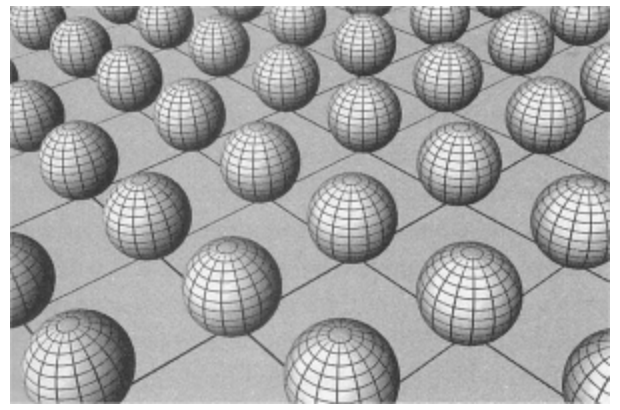
[STFC / Ben Gililand, Sean Carroll, Friedrich Justin Bertuch 1806, symmetry]

Energy

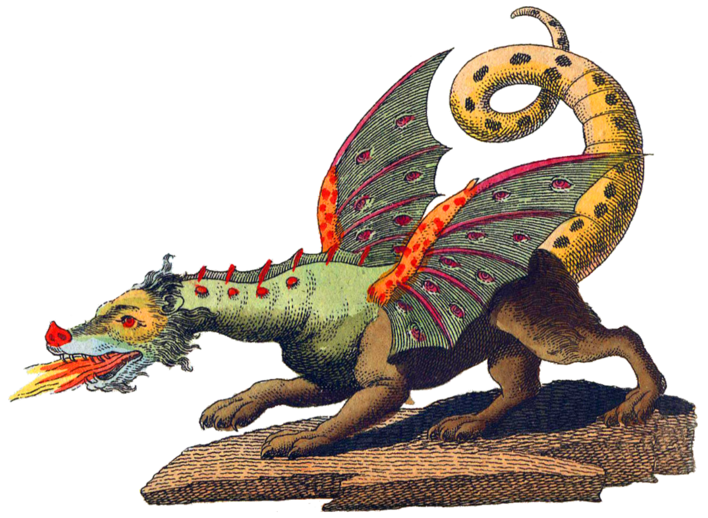
$\Lambda$



Supersymmetry?



Extra dimensions?



Dragons?


Composite Higgs?

…

Matching

$$\mathcal{L}_{\text{EFT}} = \mathcal{L}_{\text{SM}} + \sum_i \frac{f_i}{\Lambda^2}\, \mathcal{O}_i + \ldots$$

Effective field theory

Measurements

$E \ll \Lambda$

LHC Higgs

Other LHC measurements

LEP

Flavour

…

# SMEFT (Standard Model Effective Field Theory)

[W. Buchmuller, D. Wyler 85;
B. Grzadkowski, M. Iskrzynski, M. Misiak, J. Rosiek 1008.4884; …]

$$\mathcal{L}_{\text{EFT}} = \mathcal{L}_{\text{SM}} + \sum_i \frac{f_i}{\Lambda^2} \mathcal{O}_i + \ldots$$

## Operators

- all possible interactions between SM particles mediated by new physics
- fixed by SM particles + SM symmetries + expansion in $1/\Lambda$, independent of high-energy physics
- affect rates + kinematics

| | |
|---|---|
| $\mathcal{O}_{\phi,1} = (D_\mu\phi)^\dagger \phi\, \phi^\dagger D^\mu \phi$ | $\mathcal{O}_{GG} = (\phi^\dagger\phi)\, G^a_{\mu\nu}\, G^{\mu\nu\, a}$ |
| $\mathcal{O}_{\phi,2} = \frac{1}{2}\partial_\mu(\phi^\dagger\phi)\, \partial^\mu(\phi^\dagger\phi)$ | $\mathcal{O}_{BB} = -\frac{g'^2}{4}(\phi^\dagger\phi)\, B_{\mu\nu}\, B^{\mu\nu}$ |
| $\mathcal{O}_{\phi,3} = \frac{1}{3}(\phi^\dagger\phi)^3$ | $\mathcal{O}_{WW} = -\frac{g^2}{4}(\phi^\dagger\phi)\, W^a_{\mu\nu}\, W^{\mu\nu\, a}$ |
| $\mathcal{O}_{\phi,4} = (\phi^\dagger\phi)\,(D_\mu\phi)^\dagger D^\mu\phi$ | $\mathcal{O}_{BW} = -\frac{g\,g'}{4}(\phi^\dagger\sigma^a\phi)\, B_{\mu\nu}\, W^{\mu\nu\, a}$ |
| | $\mathcal{O}_{B} = \frac{ig'}{2}(D^\mu\phi)^\dagger D^\nu\phi\, B_{\mu\nu}$ |
| | $\mathcal{O}_{W} = \frac{ig}{2}(D^\mu\phi)^\dagger \sigma^a D^\nu\phi\, W^a_{\mu\nu}$ |

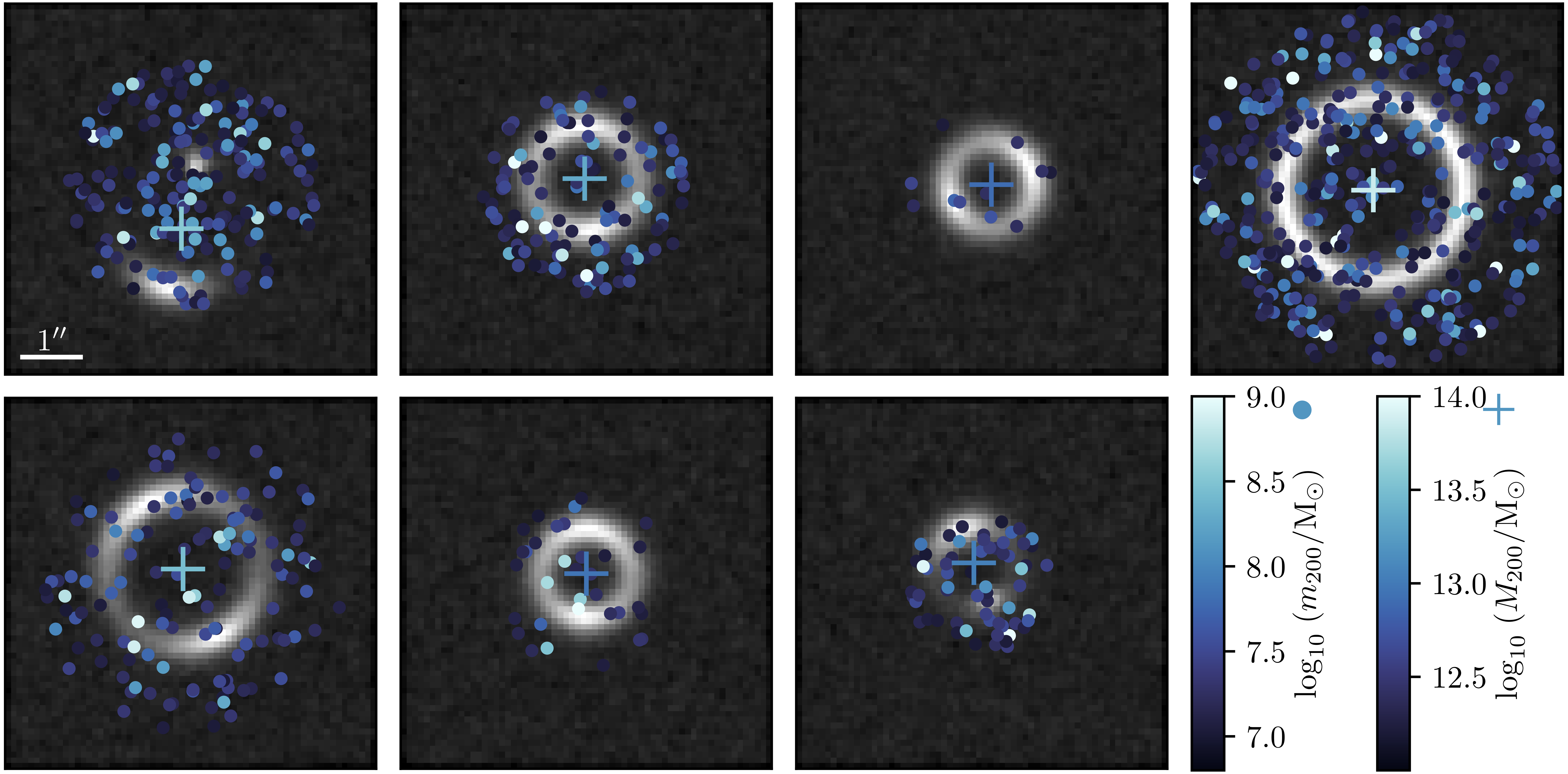
## Wilson coefficients

- precise measurement of these parameters is one of the most important goals of the LHC
- can be translated to high-energy physics parameters

## Higher-order terms

- suppressed by additional factors of $E^2 / \Lambda^2$

# Strong lensing: simulated images



# Strong lensing: expected posterior