

The frontier of simulation-based inference

Johann Brehmer

New York University

Machine Learning at LHC workshop, Nagoya University

February 5, 2020



Kyle Cranmer



Gilles Louppe



Juan Pavez



Markus Stoye



Felix Kling



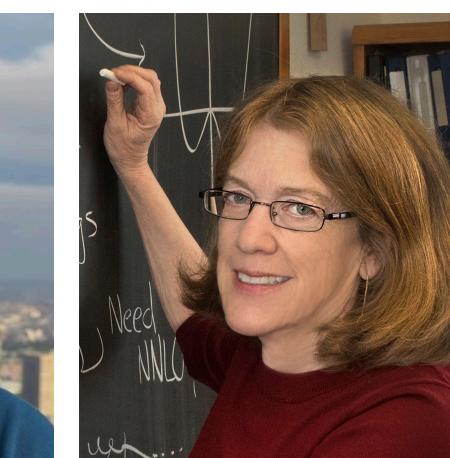
Irina Espejo



Tilman Plehn



Sam Homiller



Sally Dawson



Sid Mishra-Sharma

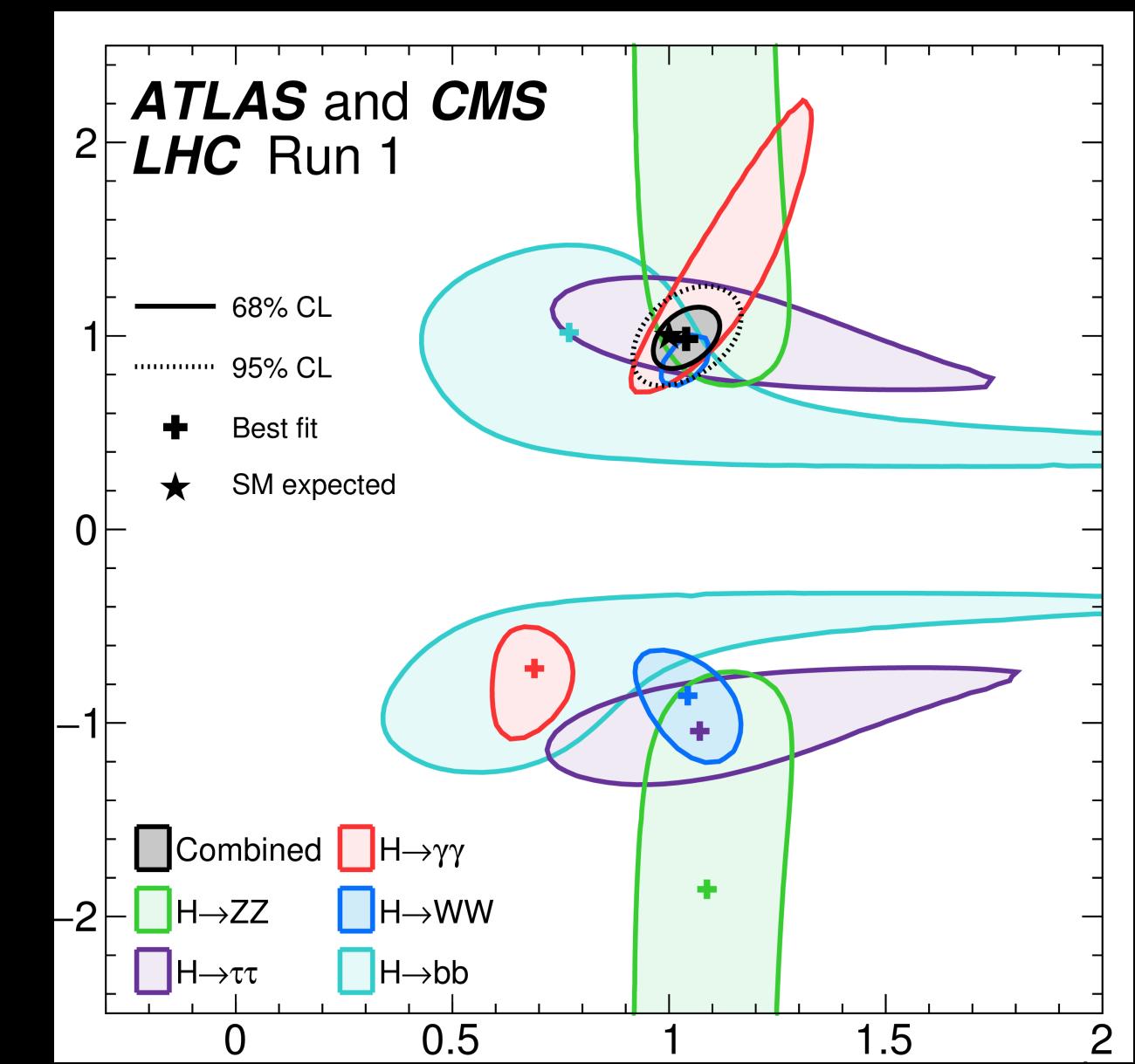
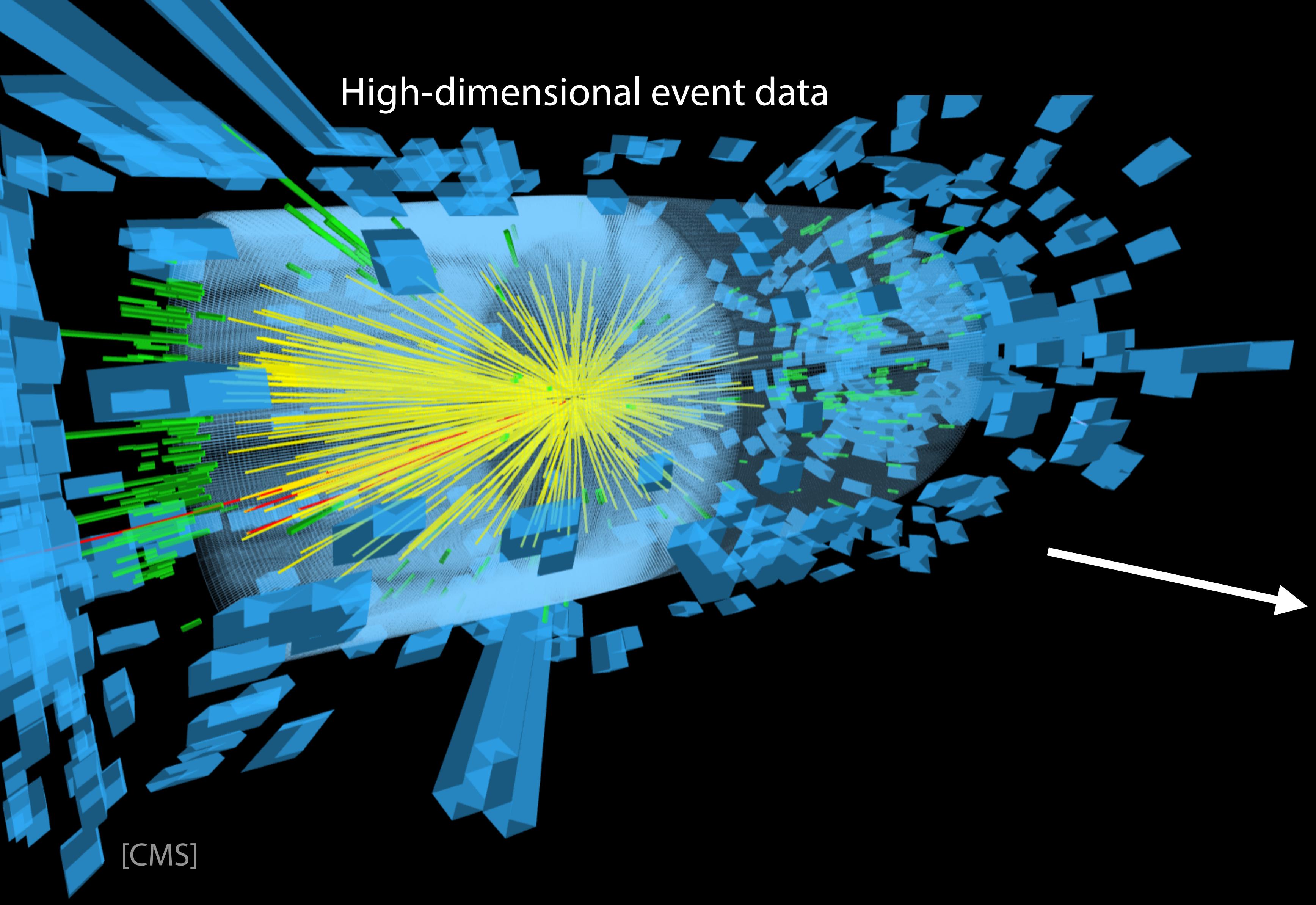


Zubair Bhatti

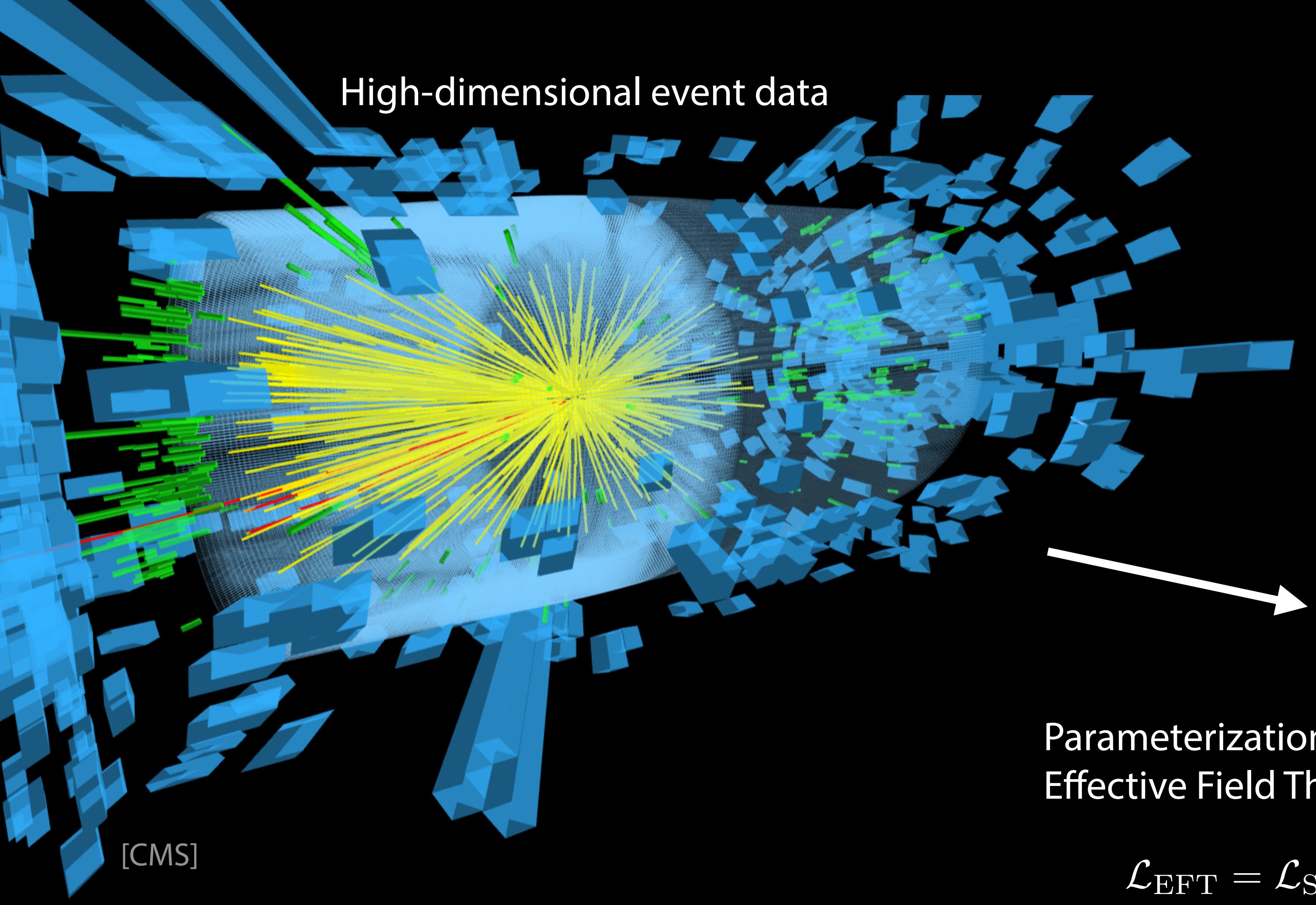


The SCAILFIN Project
scailfin.github.io





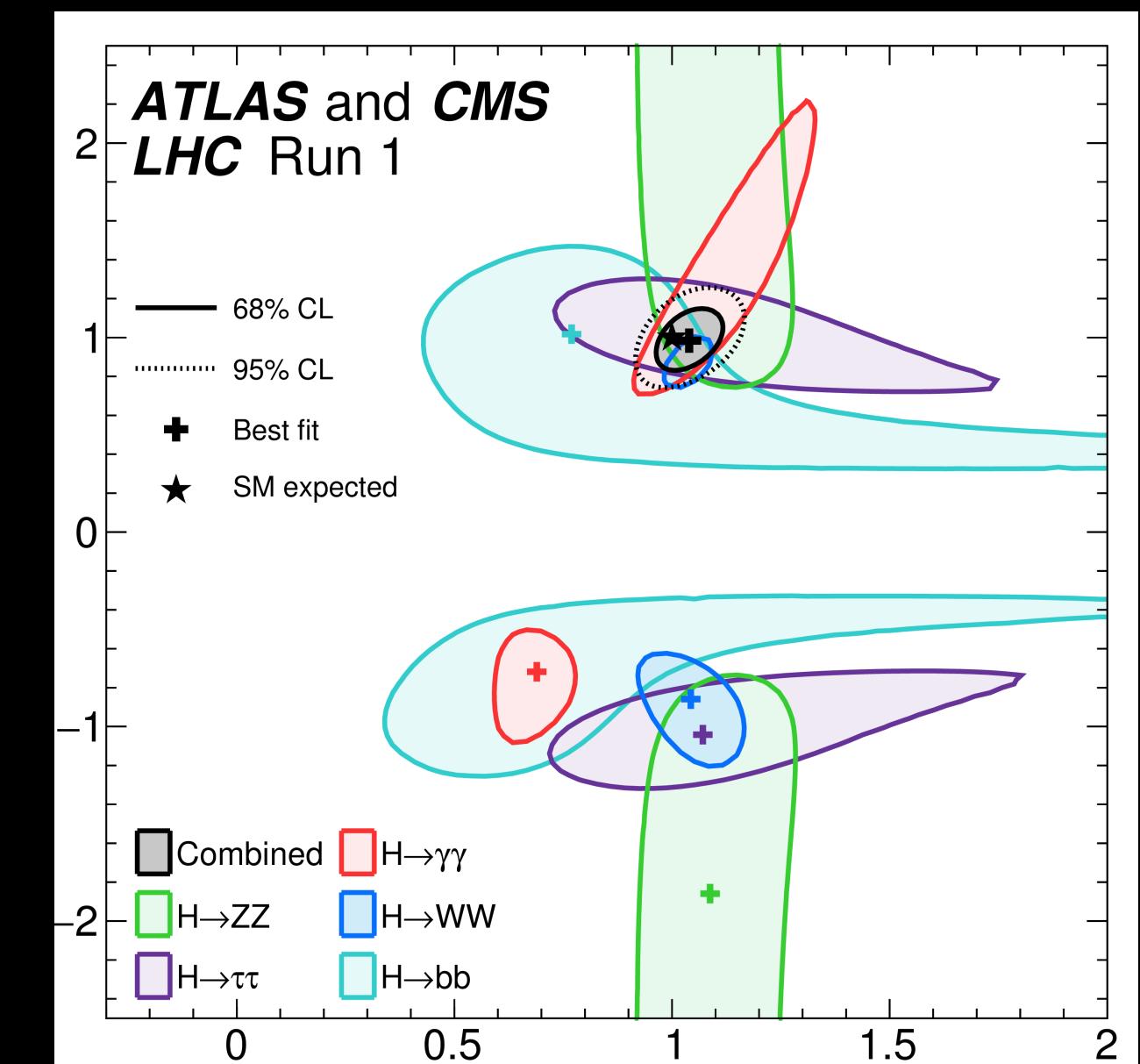
Precision constraints on
new physics



Parameterization e.g. in
Effective Field Theory:

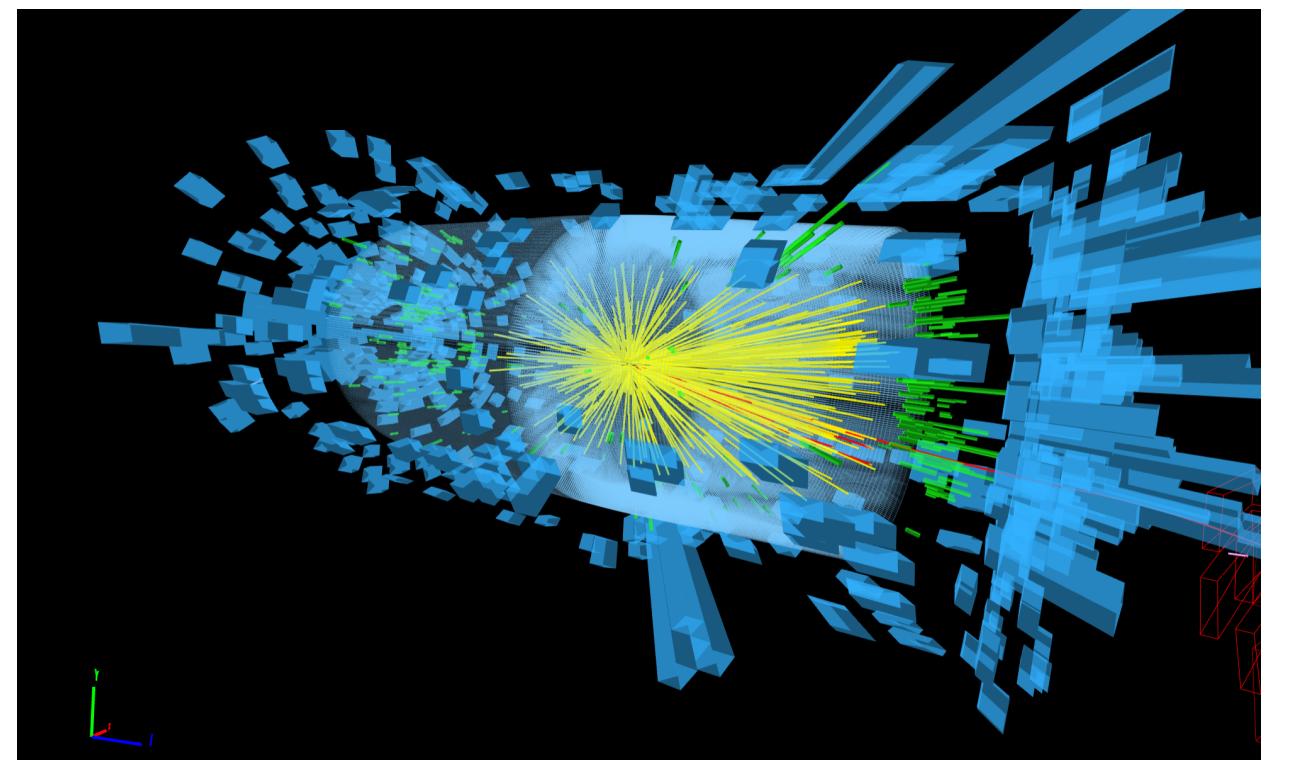
$$\mathcal{L}_{\text{EFT}} = \mathcal{L}_{\text{SM}} + \sum_i \frac{f_i}{\Lambda^2} \mathcal{O}_i + \dots$$

10s to 100s “universal”
parameters to measure

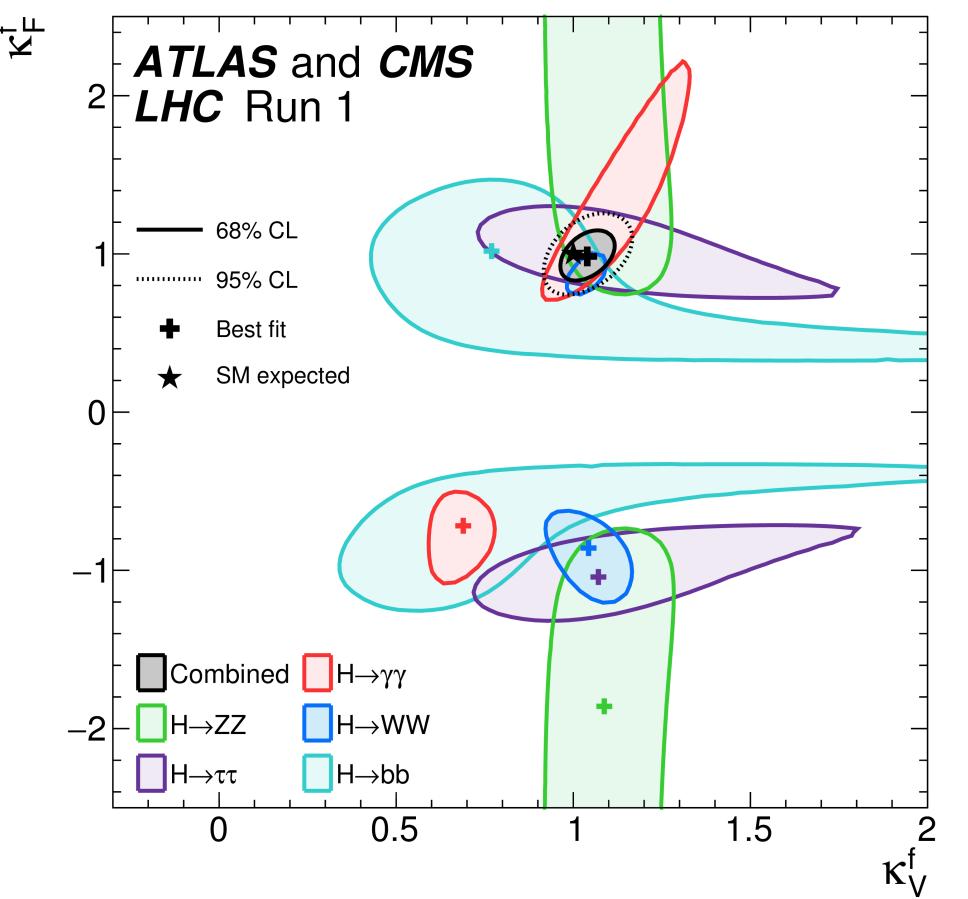


Precision constraints on
new physics

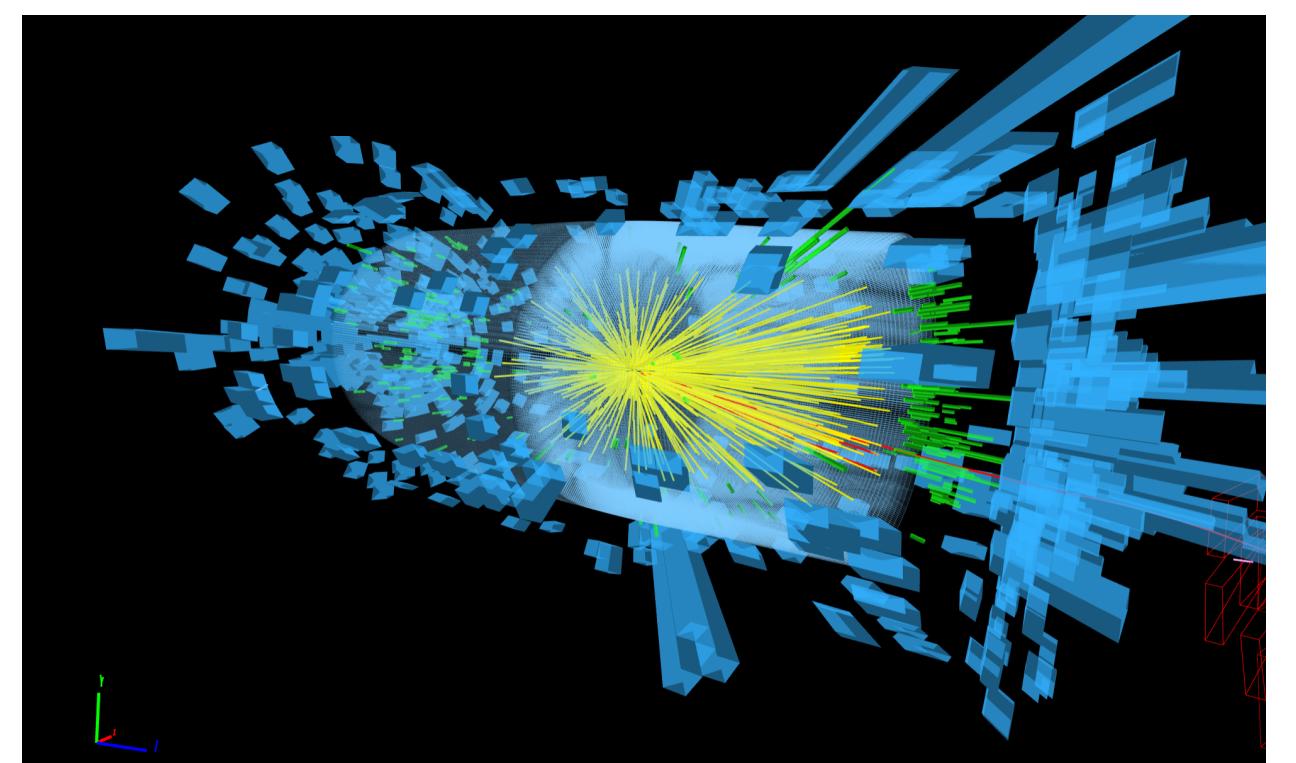
systematic expansion of
new physics around
Standard Model



High-dimensional
event data x



Constraints on
parameters θ

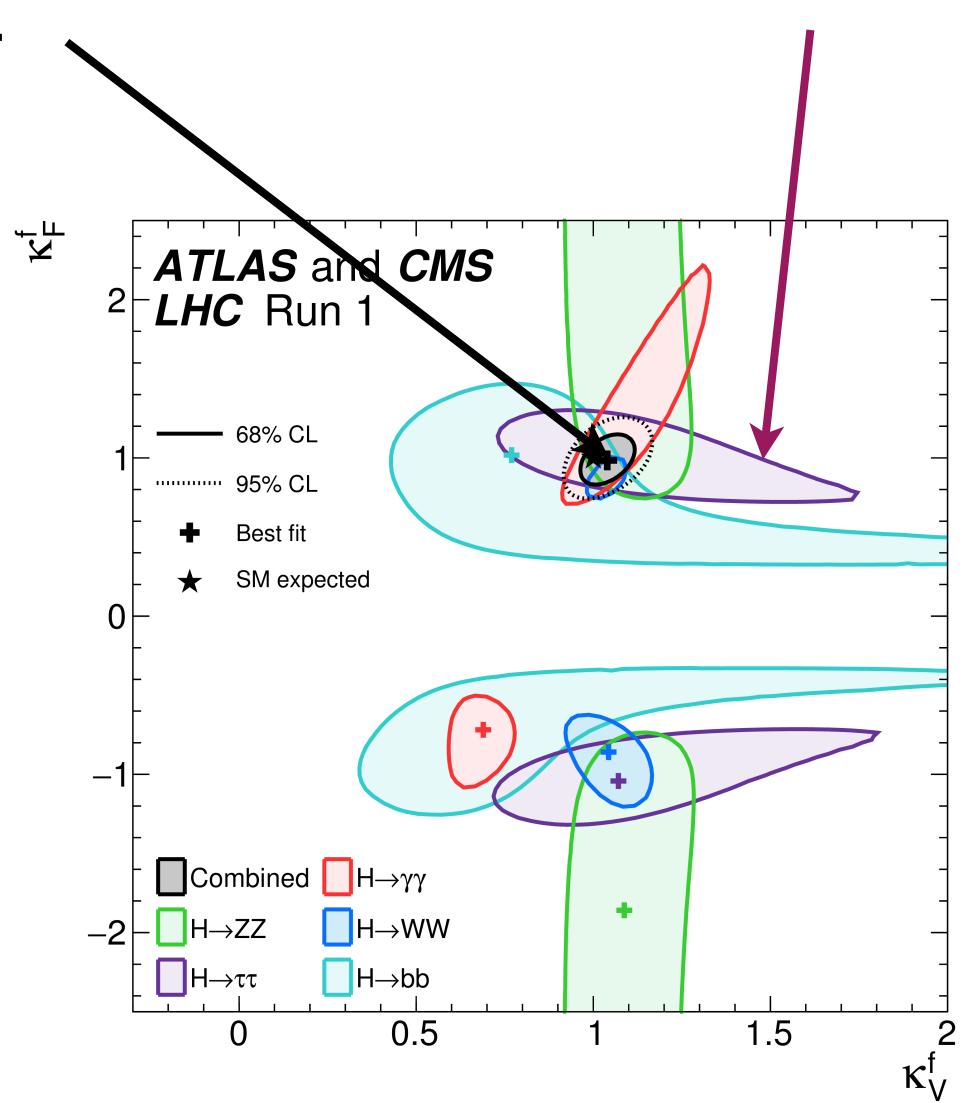


High-dimensional
event data x



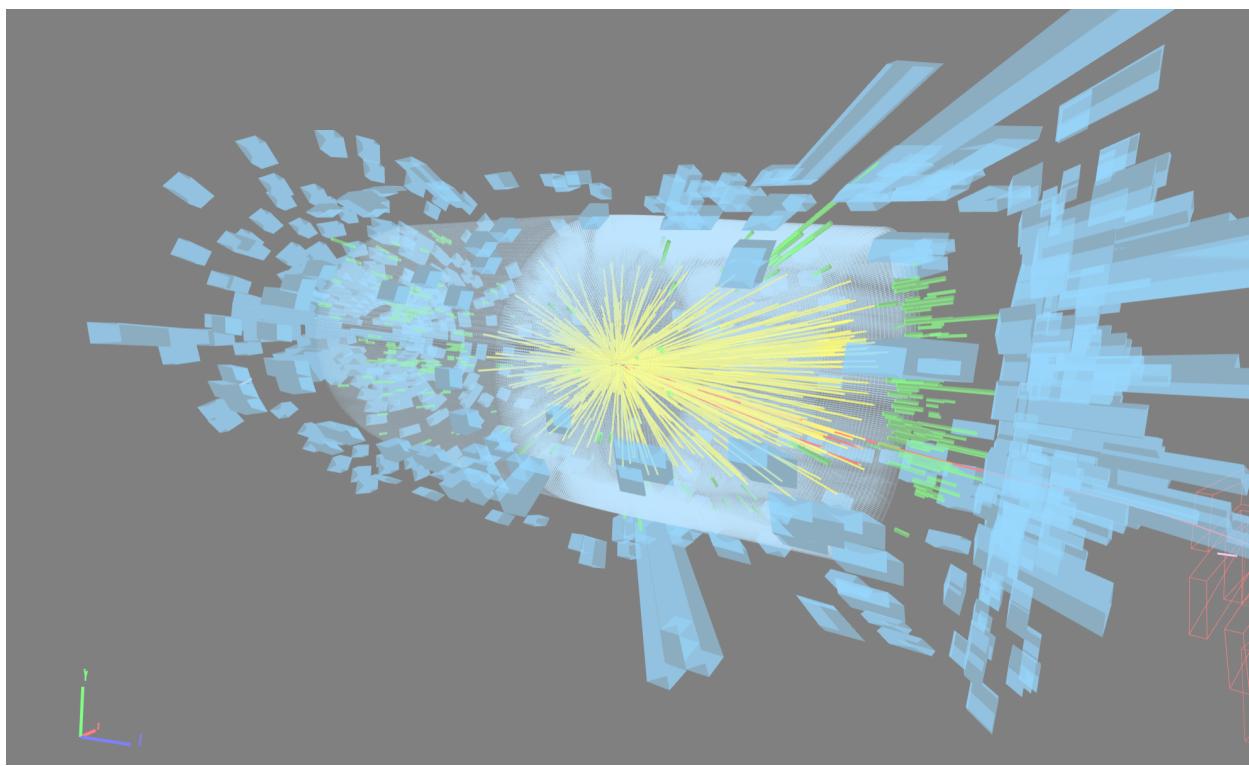
Likelihood function
 $p(x|\theta)$

Maximum-likelihood
estimator



Confidence limits based
on likelihood ratio tests

Constraints on
parameters θ

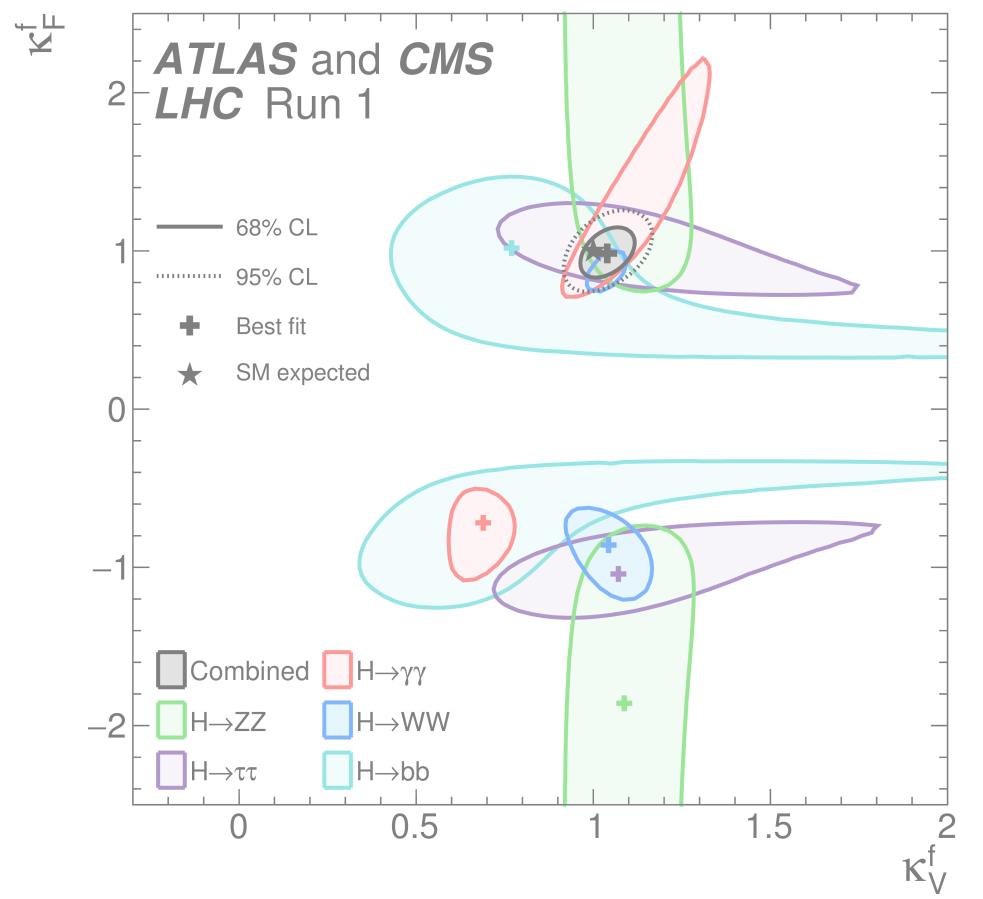


High-dimensional
event data x

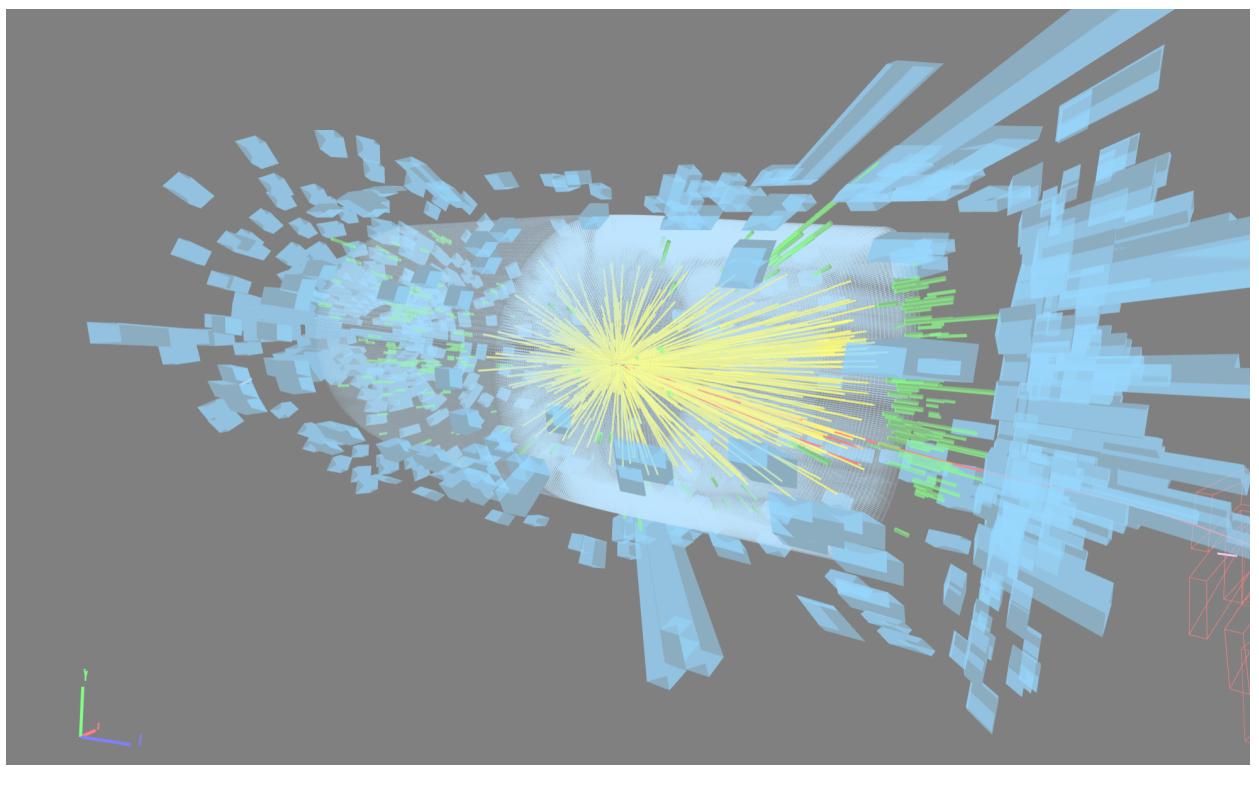
Surprisingly, when we want to use high-dimensional data and have to deal with the detector response, we do not have a good way to calculate the likelihood.



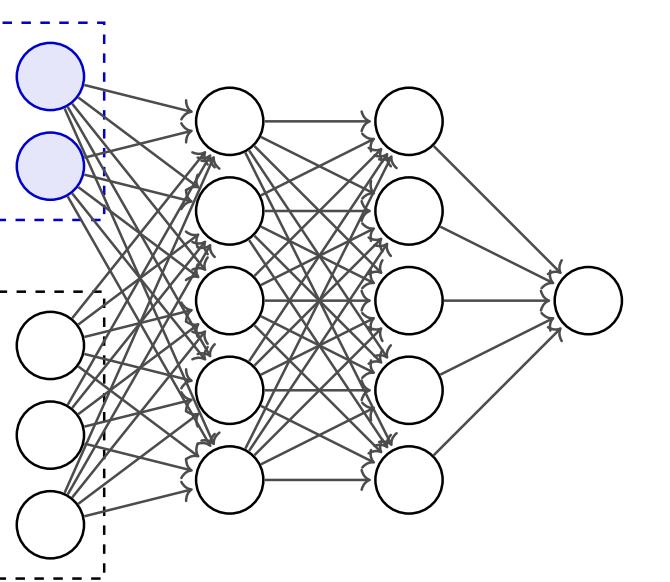
Likelihood function
 $p(x|\theta)$



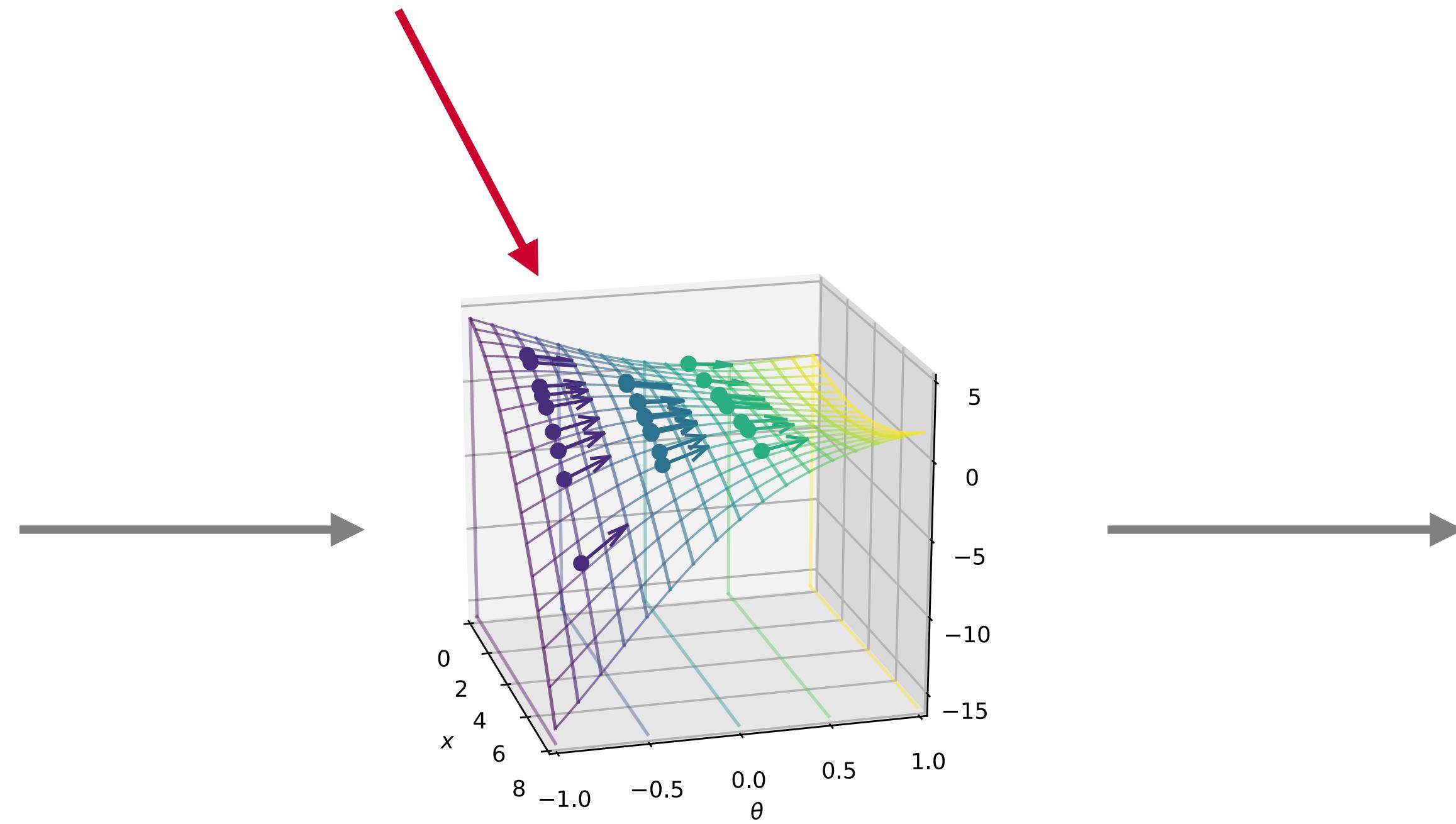
Constraints on
parameters θ



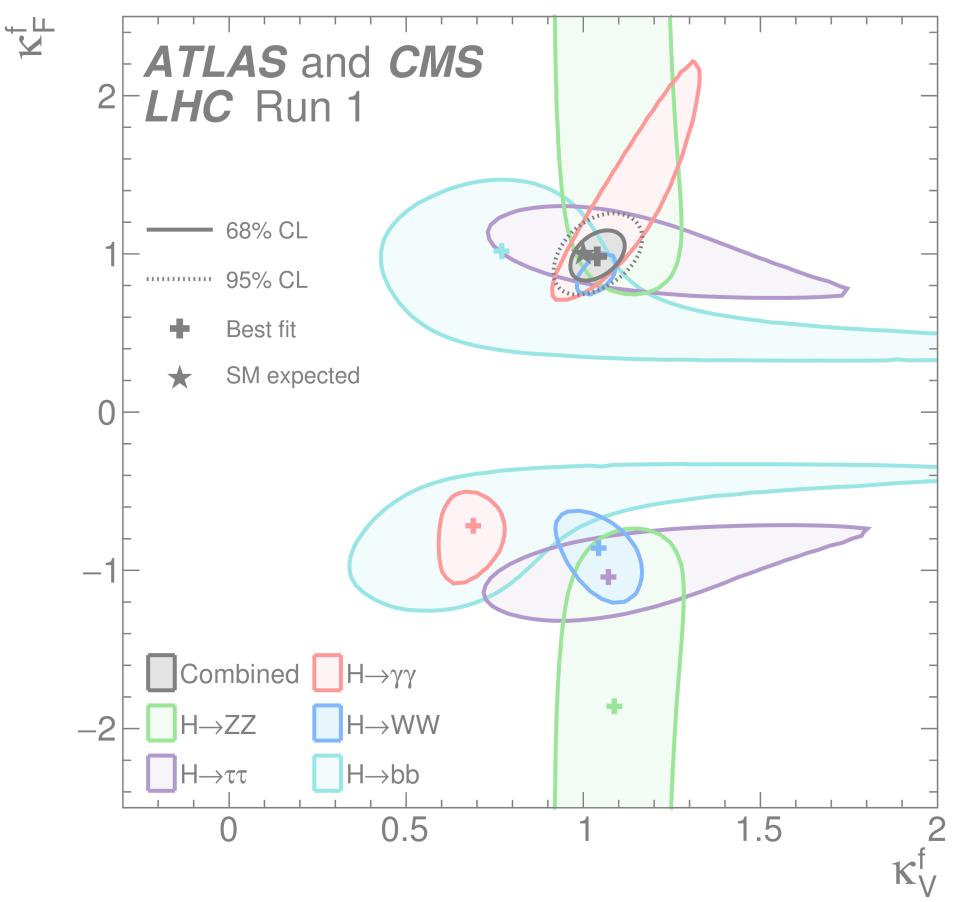
High-dimensional
event data x



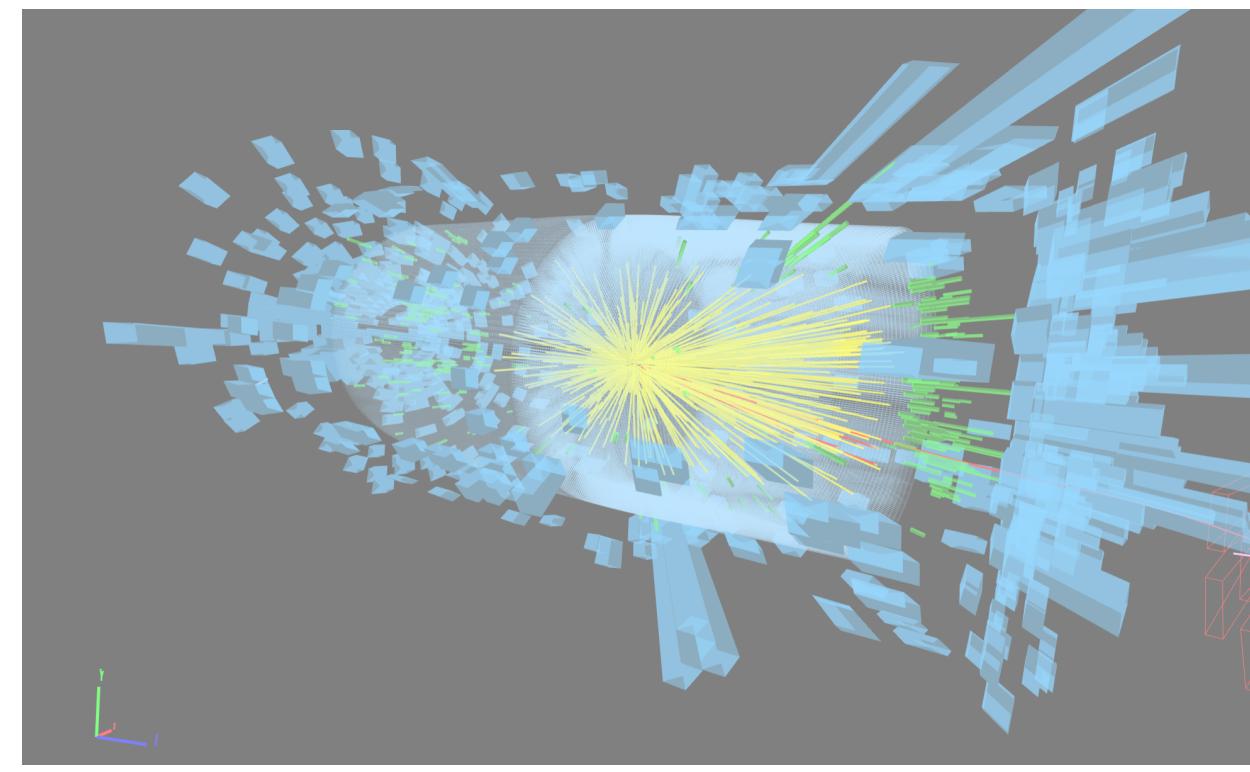
Machine learning



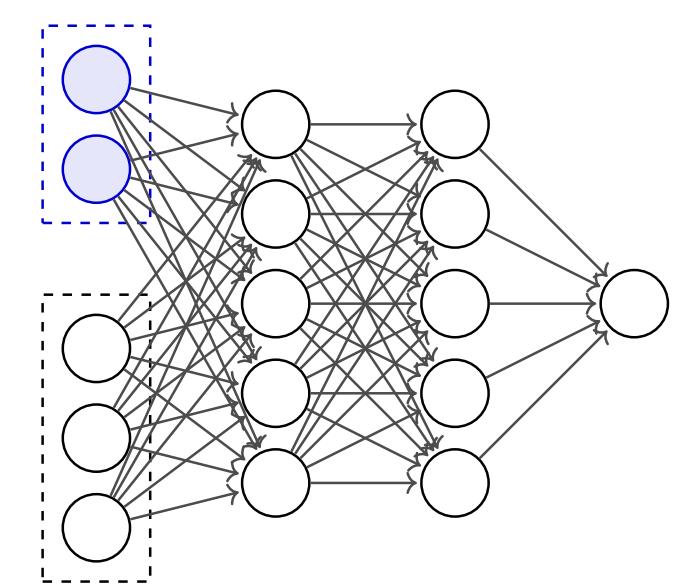
Estimator of the
likelihood $p(x|\theta)$



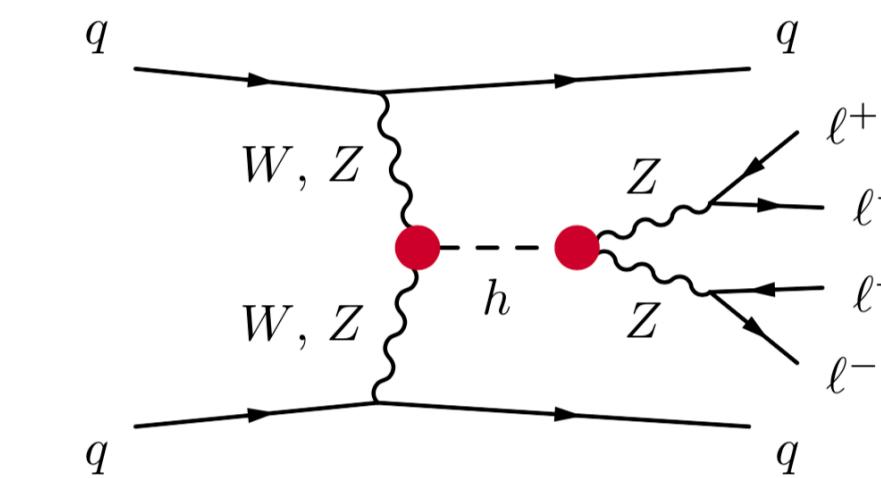
Constraints on
parameters θ



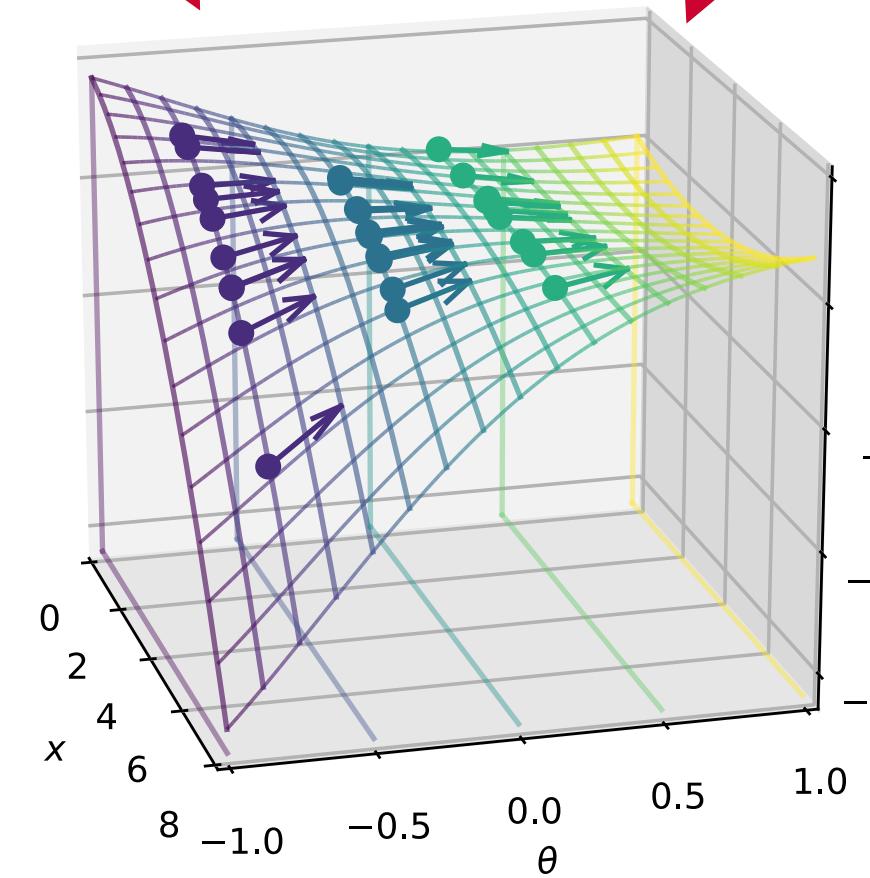
High-dimensional
event data x



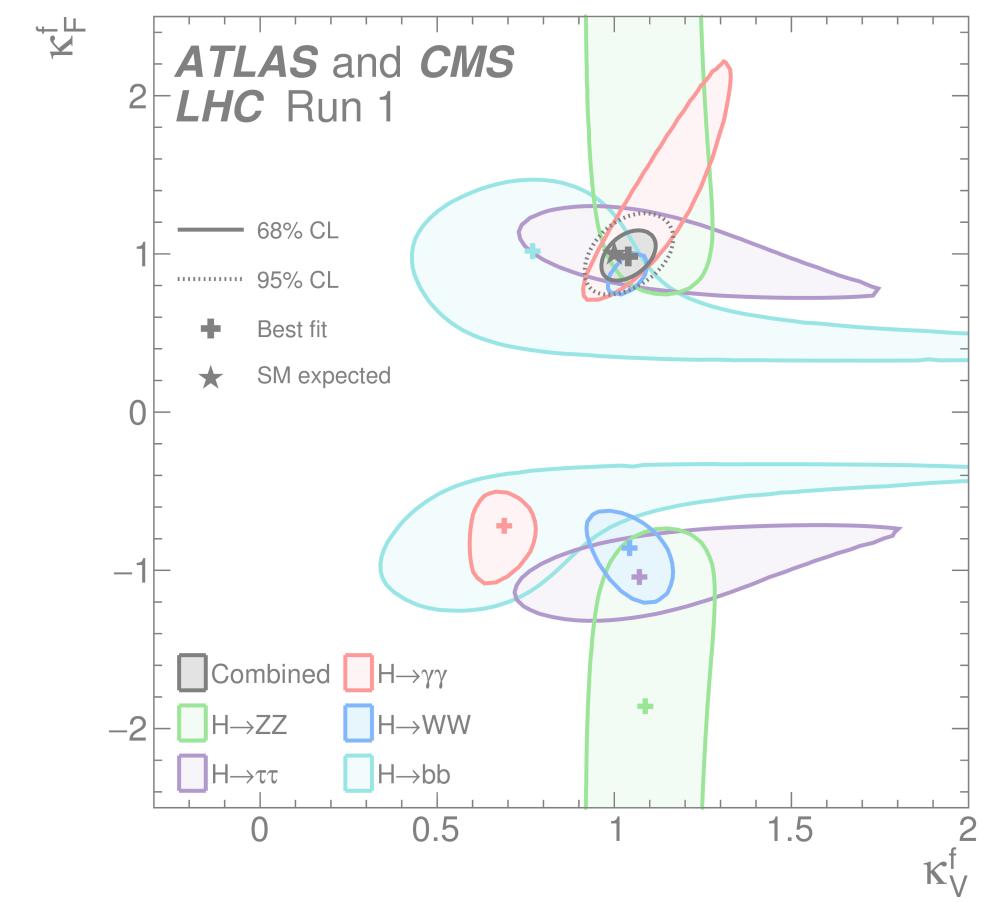
Machine learning



Physics insight:
matrix element information



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



Constraints on
parameters θ

Particle physics processes do not have a tractable likelihood function.

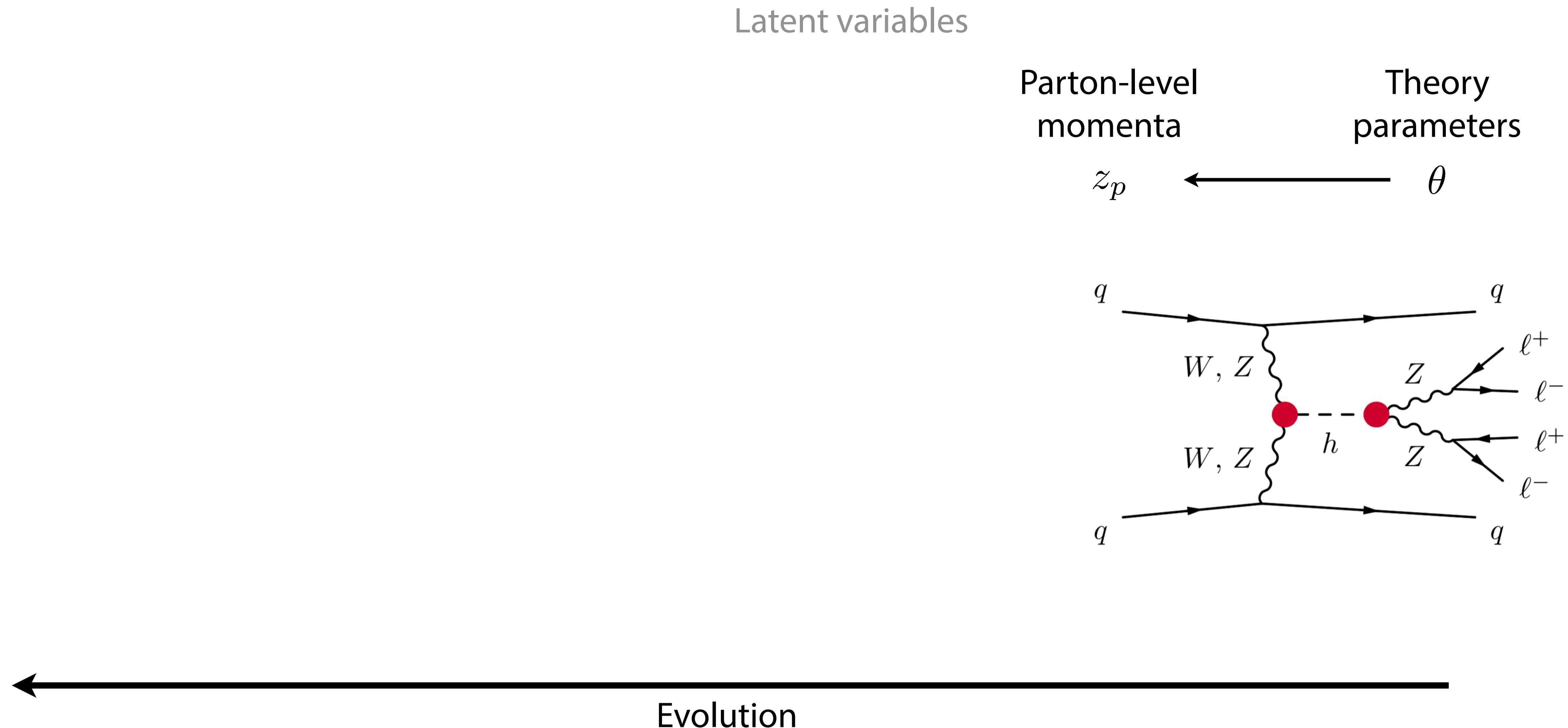
Modelling particle physics processes

Theory
parameters
 θ

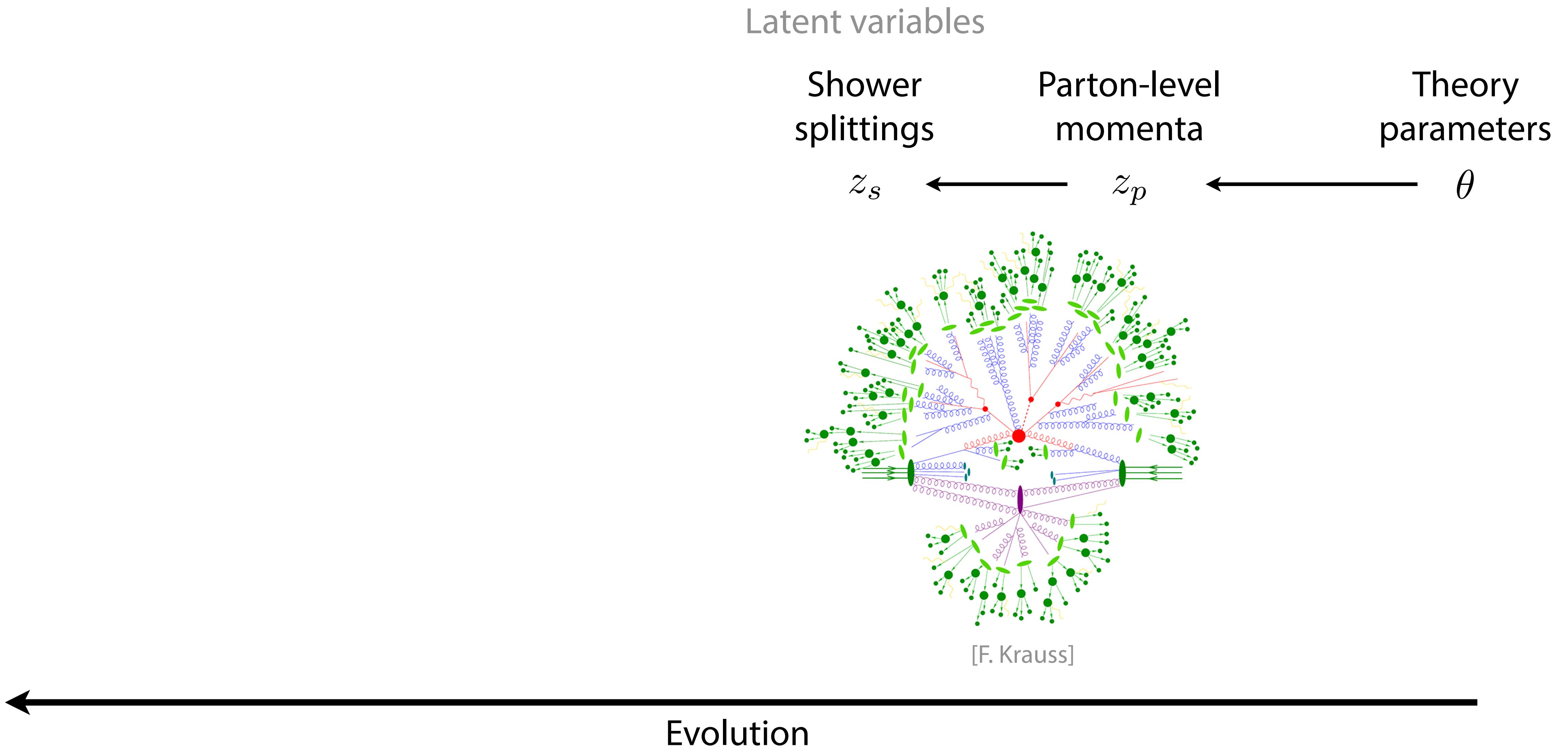


Evolution

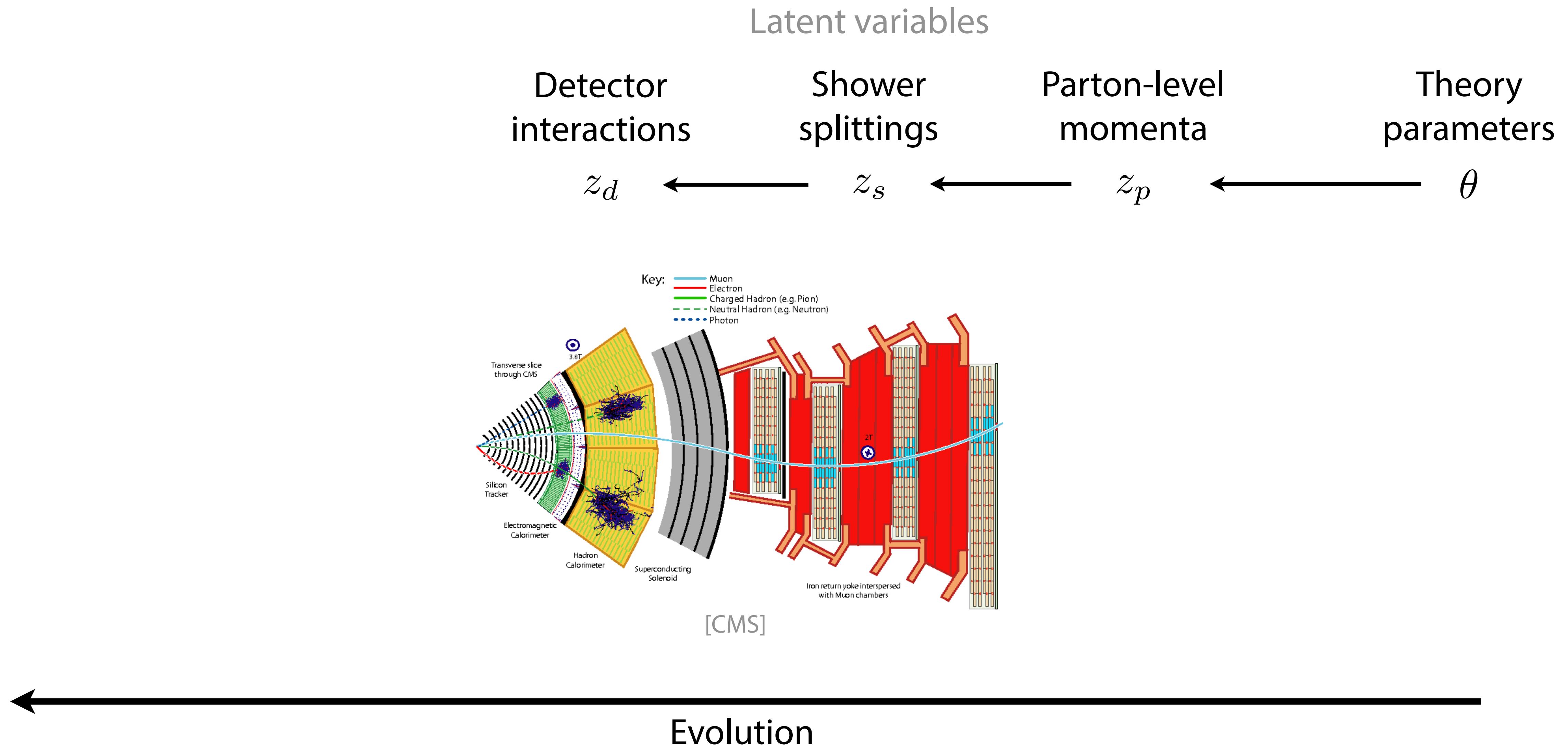
Modelling particle physics processes



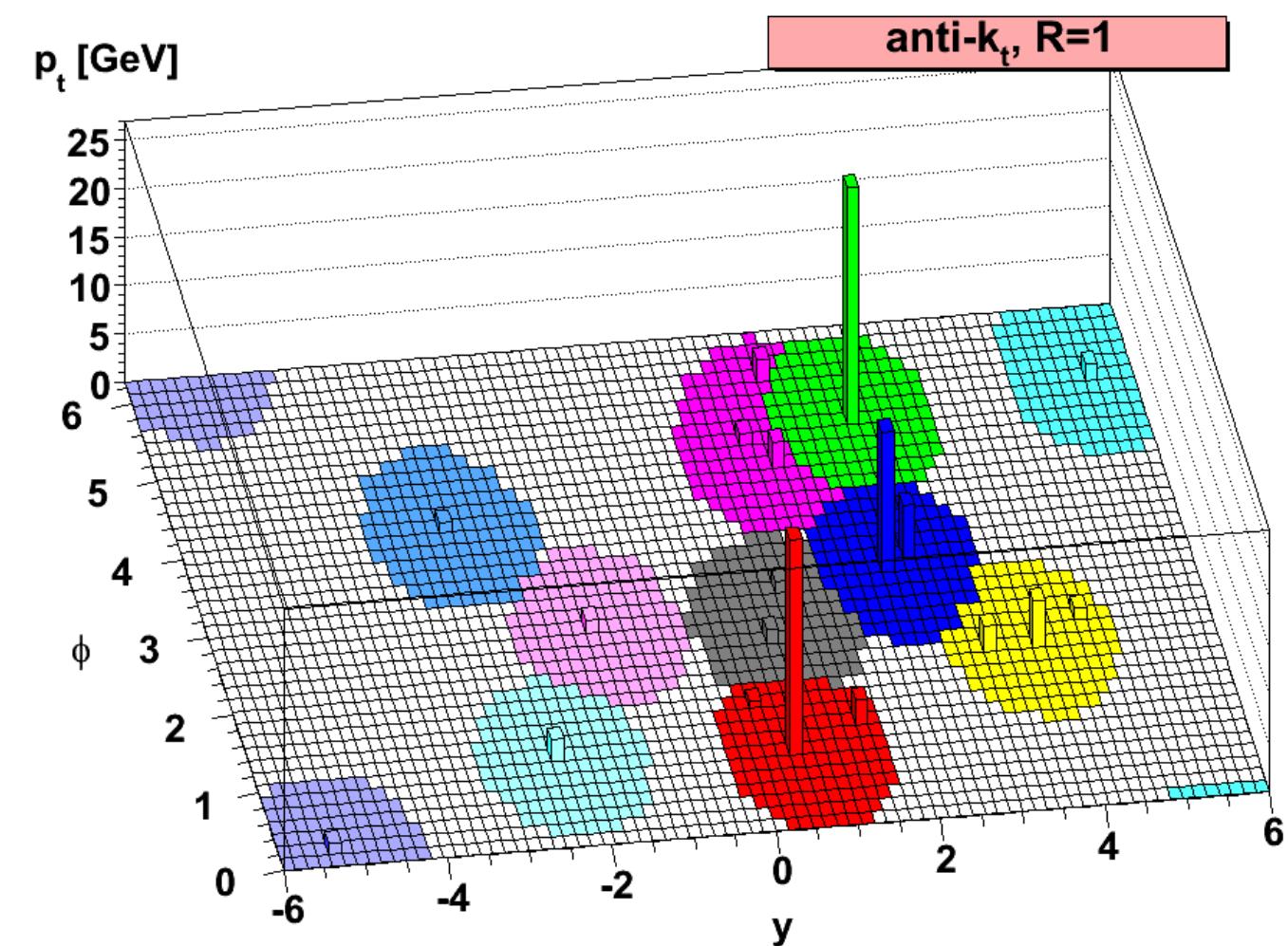
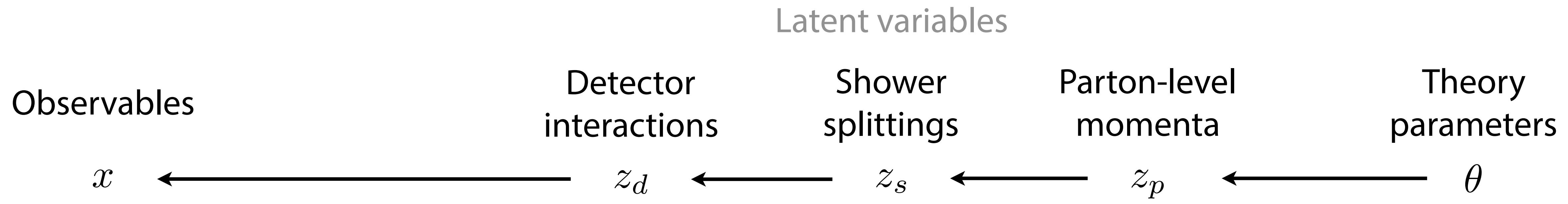
Modelling particle physics processes



Modelling particle physics processes



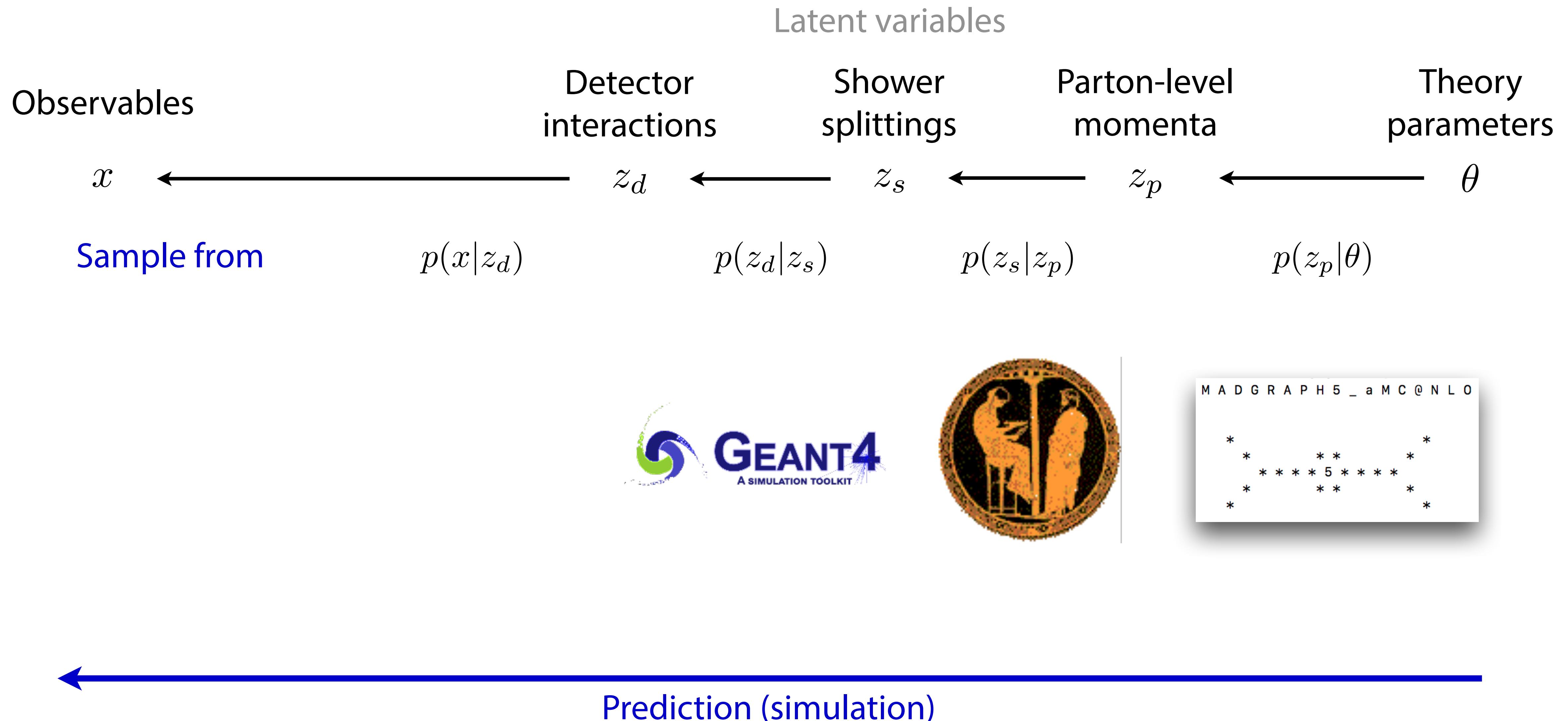
Modelling particle physics processes



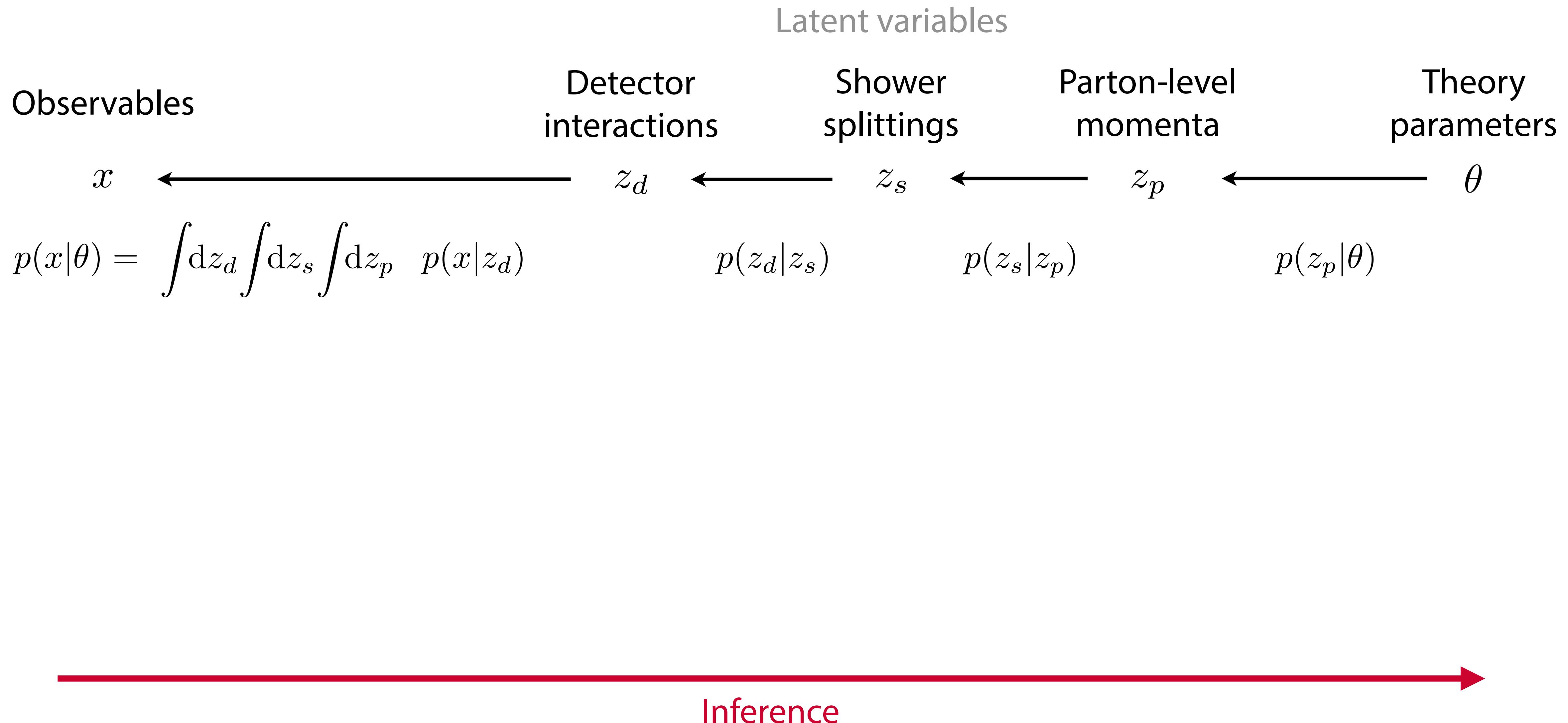
[M. Cacciari, G. Salam, G. Soyez 0802.1189]

Evolution

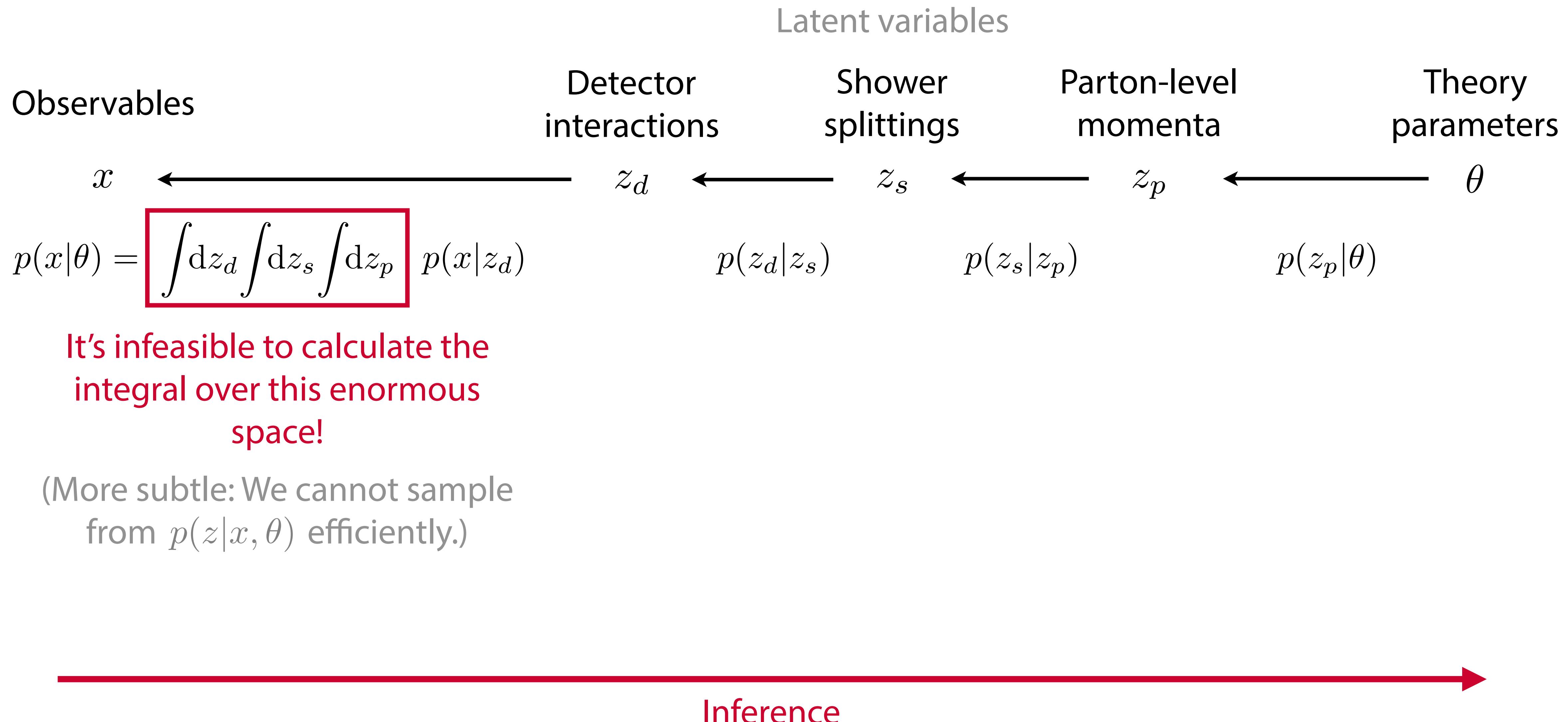
Modelling particle physics processes



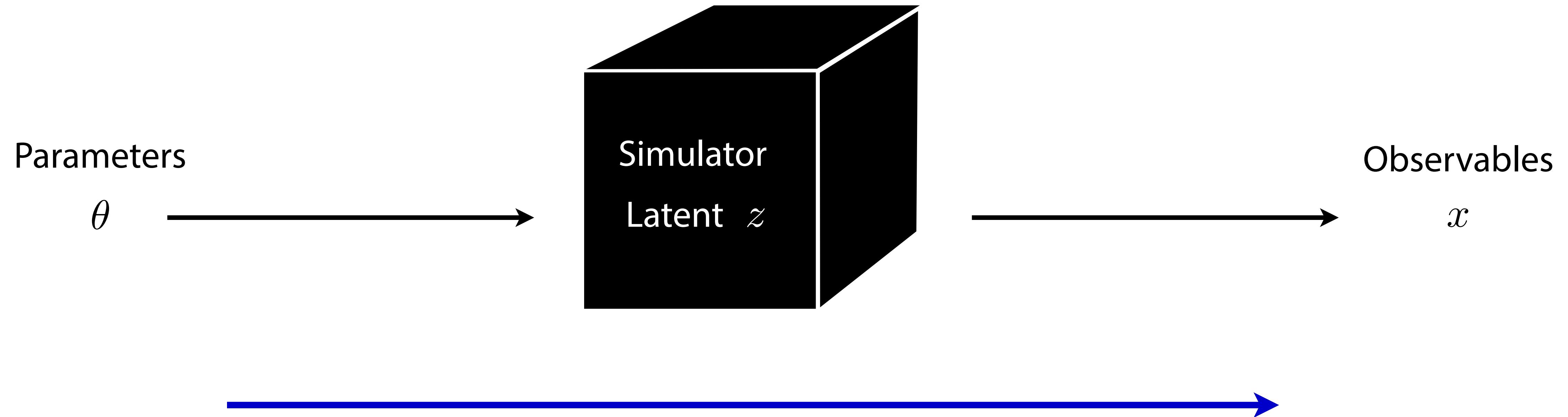
Modelling particle physics processes



Modelling particle physics processes

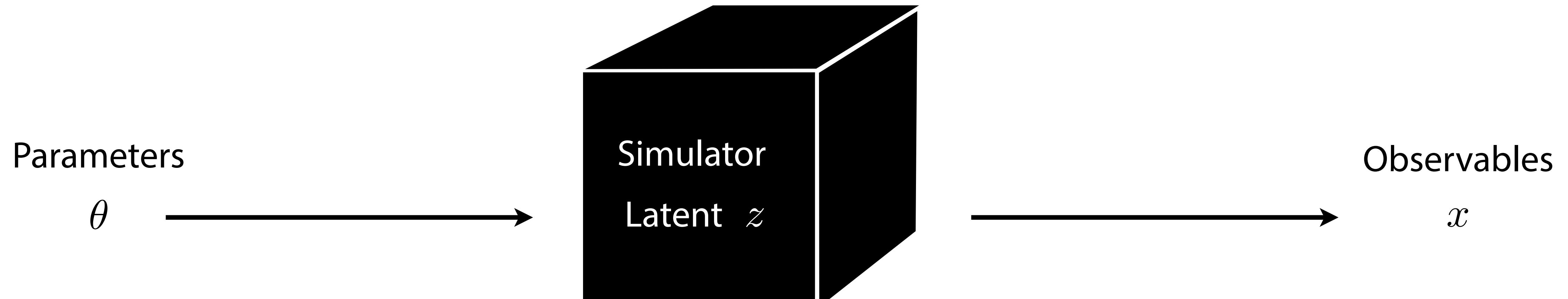


Simulation-based (“likelihood-free”) inference



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

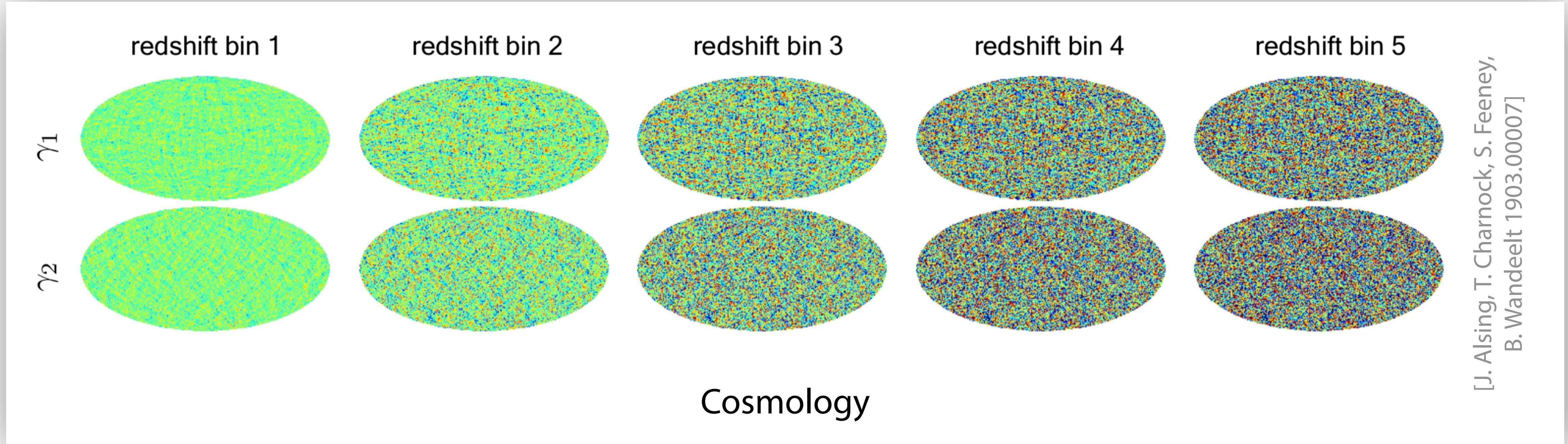
Simulation-based (“likelihood-free”) inference



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

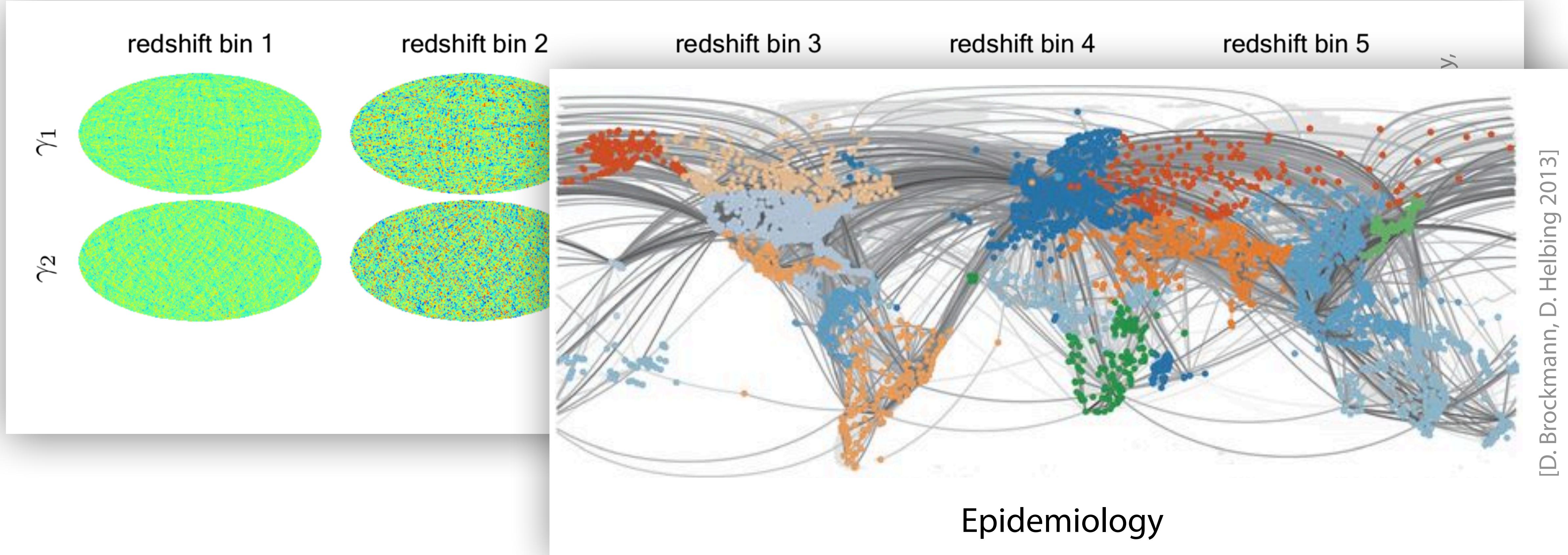
- Inference:
- Likelihood $p(x|\theta) = \int dz p(x, z|\theta)$ is intractable
 - Inference is challenging

It's not just particle physics!

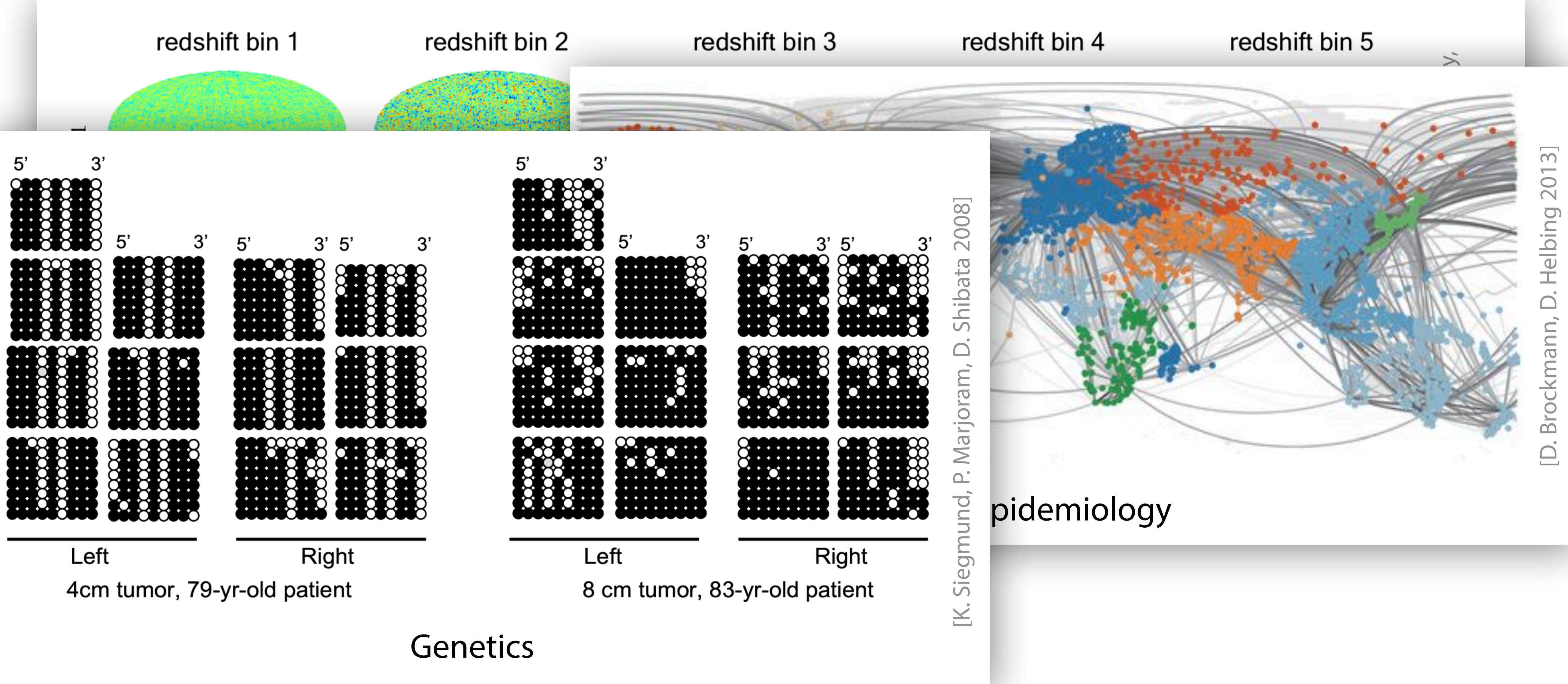


[J. Alsing, T. Charnock, S. Feeney,
B. Wandelt 1903.00007]

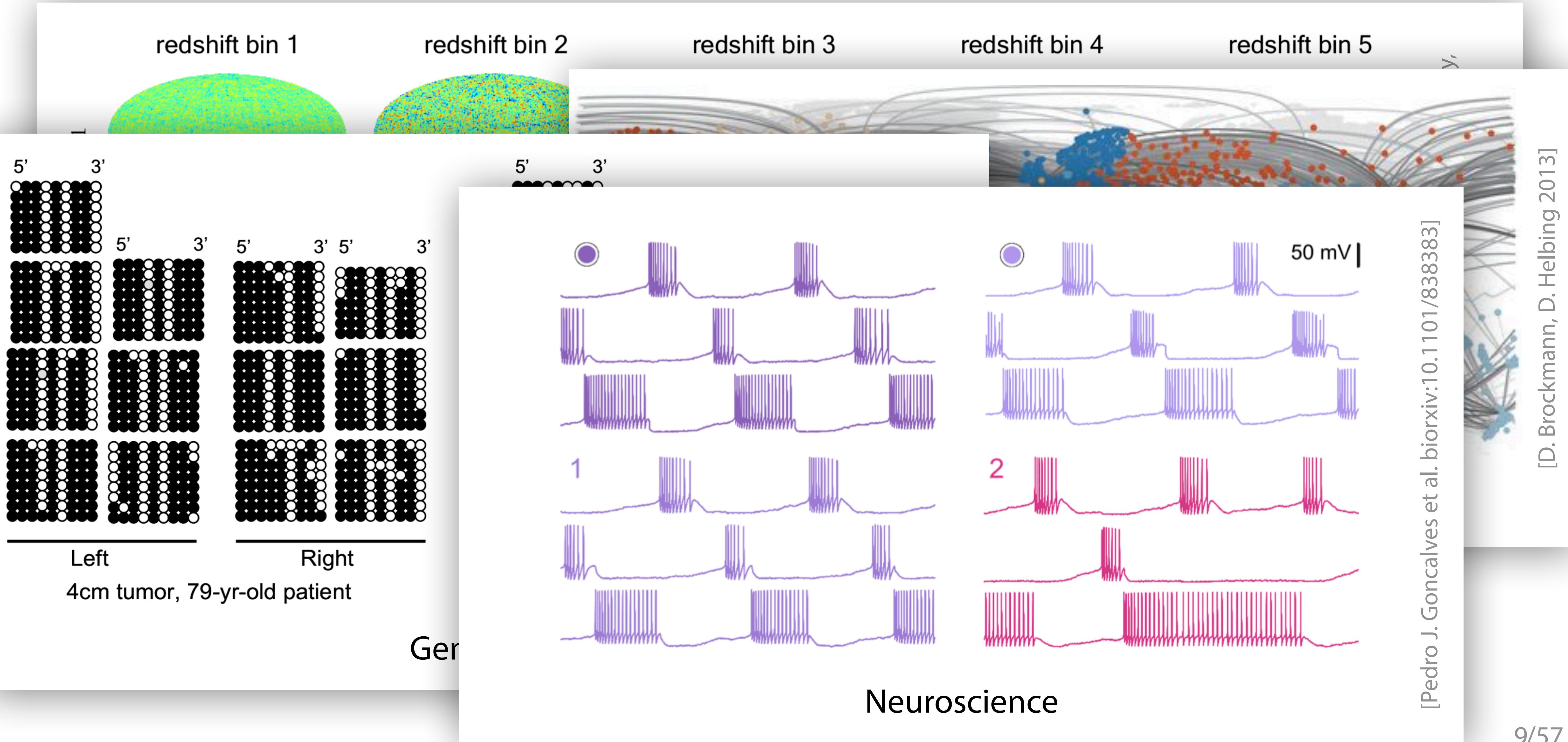
It's not just particle physics!



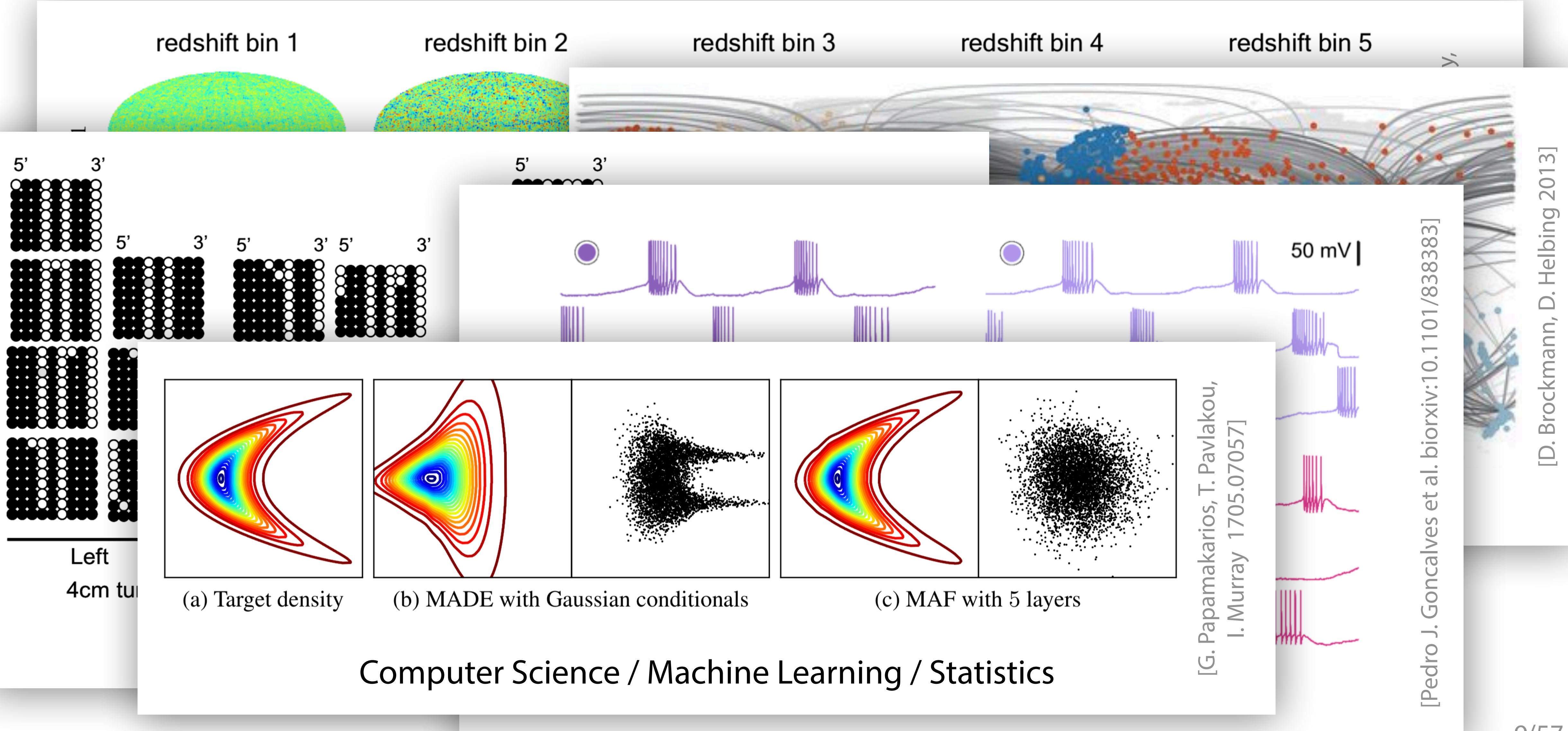
It's not just particle physics!



It's not just particle physics!



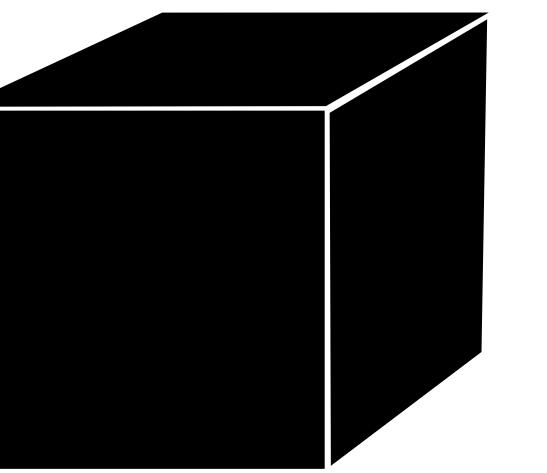
It's not just particle physics!



Formalizing the problem

You are given

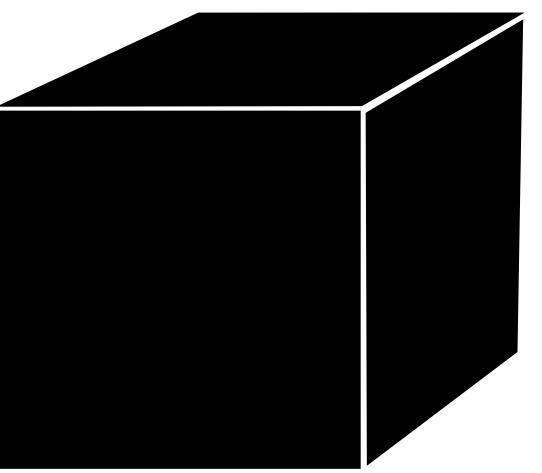
- a simulator that lets you generate N samples $x_i \sim p(x_i|\theta_i)$ for parameter points θ_i of your choice
- observed data $x_{\text{obs}} \sim p(x_{\text{obs}}|\theta_{\text{true}})$
- prior belief $p(\theta)$



Formalizing the problem

You are given

- a simulator that lets you generate N samples $x_i \sim p(x_i|\theta_i)$ for parameter points θ_i of your choice
- observed data $x_{\text{obs}} \sim p(x_{\text{obs}}|\theta_{\text{true}})$
- prior belief $p(\theta)$



Goals: estimate either...

- true parameters $\hat{\theta}_{\text{true}}$
- confidence sets based on likelihood $\hat{p}(x_{\text{obs}}|\theta)$
- posterior $\hat{p}(\theta|x_{\text{obs}}) = \frac{\hat{p}(x_{\text{obs}}|\theta) p(\theta)}{\int d\theta' \hat{p}(x_{\text{obs}}|\theta') p(\theta')}$
or samples from posterior $\theta \sim \hat{p}(\theta|x_{\text{obs}})$

... depending on domain conventions

LHC footnotes

- Full LHC likelihood: $p_{\text{full}}(\{x\}|\theta) = \text{Pois}(n|L\sigma(\theta)) \prod_{\text{events } x} p(x|\theta)$

LHC footnotes

- Full LHC likelihood:

$$p_{\text{full}}(\{x\}|\theta) = \text{Pois}(n|L\sigma(\theta)) \prod_{\text{events } x} p(x|\theta)$$

Total rate term:

- How likely is it to observe n events after cuts?
- “Easy” to compute
- For simplicity, we ignore this part in this talk

LHC footnotes

- Full LHC likelihood:

$$p_{\text{full}}(\{x\}|\theta) = \text{Pois}(n|L\sigma(\theta)) \prod_{\text{events } x} p(x|\theta)$$

Total rate term:

- How likely is it to observe n events after cuts?
- “Easy” to compute
- For simplicity, we ignore this part in this talk

Kinematic term for each event:

- How likely is it that an event looks like it does?
- \sim normalized differential xsec
- This is the intractable part of the likelihood
- Focus of this talk

LHC footnotes

- Full LHC likelihood:

$$p_{\text{full}}(\{x\}|\theta) = \text{Pois}(n|L\sigma(\theta)) \prod_{\text{events } x} p(x|\theta)$$

Total rate term:

- How likely is it to observe n events after cuts?
- “Easy” to compute
- For simplicity, we ignore this part in this talk

Kinematic term for each event:

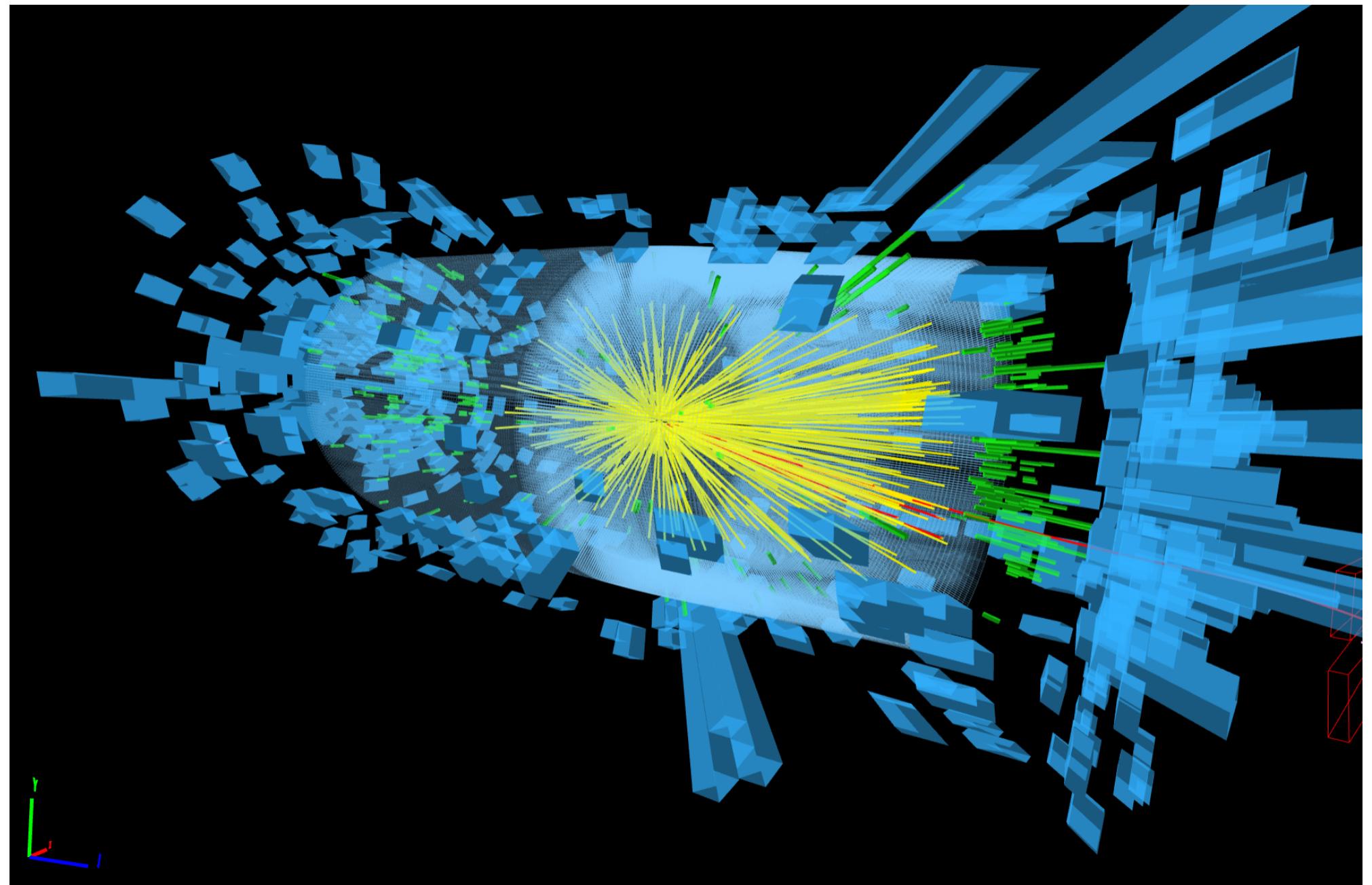
- How likely is it that an event looks like it does?
- \sim normalized differential xsec
- This is the intractable part of the likelihood
- Focus of this talk

- Event selection:

- Choice of cuts shifts information between rate and kinematic part
- “Good” cuts depend on inference strategy
- This talk: assume fixed event selection

Traditionally, inference is made possible
with summary statistics.

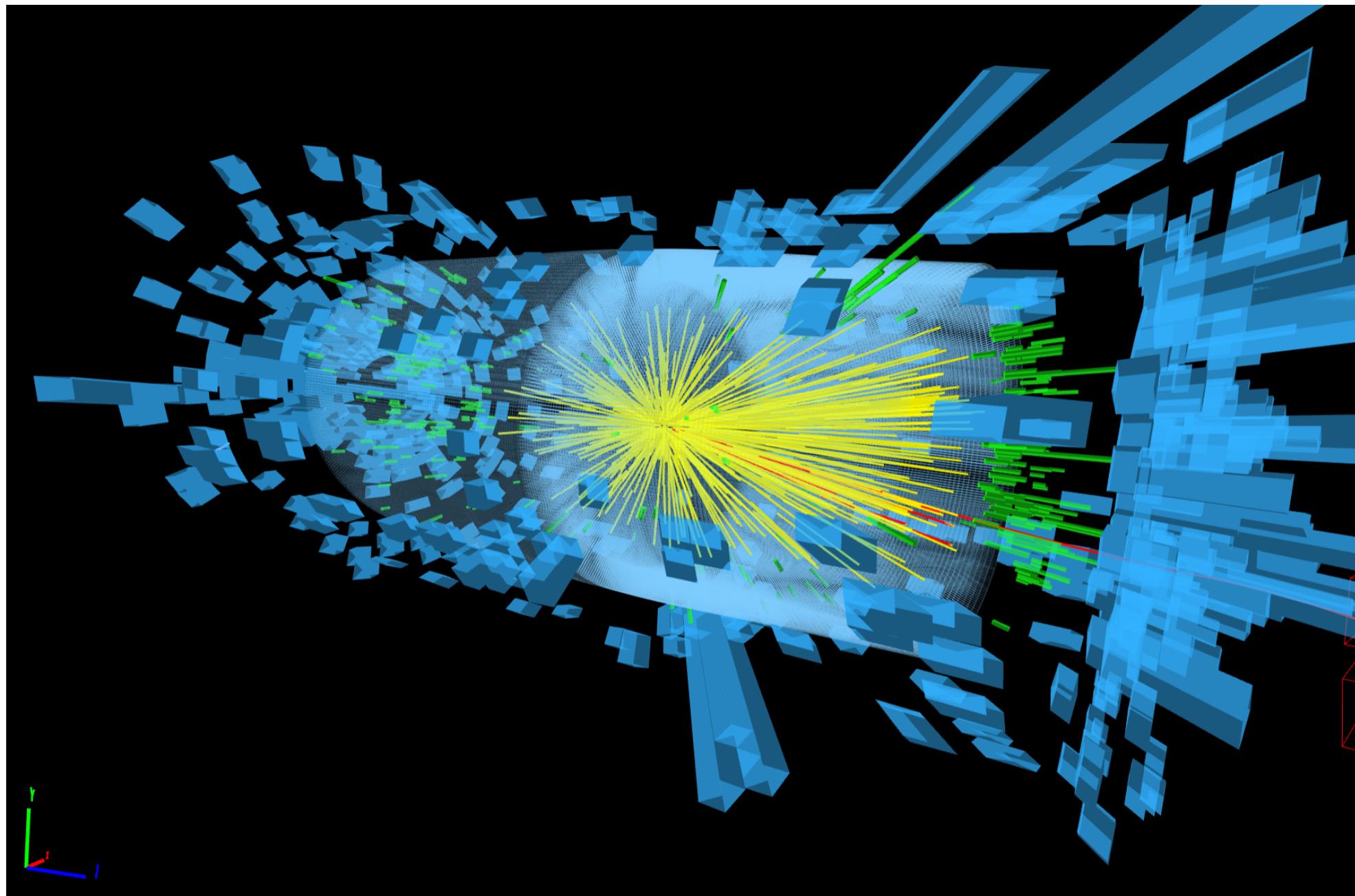
Solve it with summary statistics



High-dimensional event data x

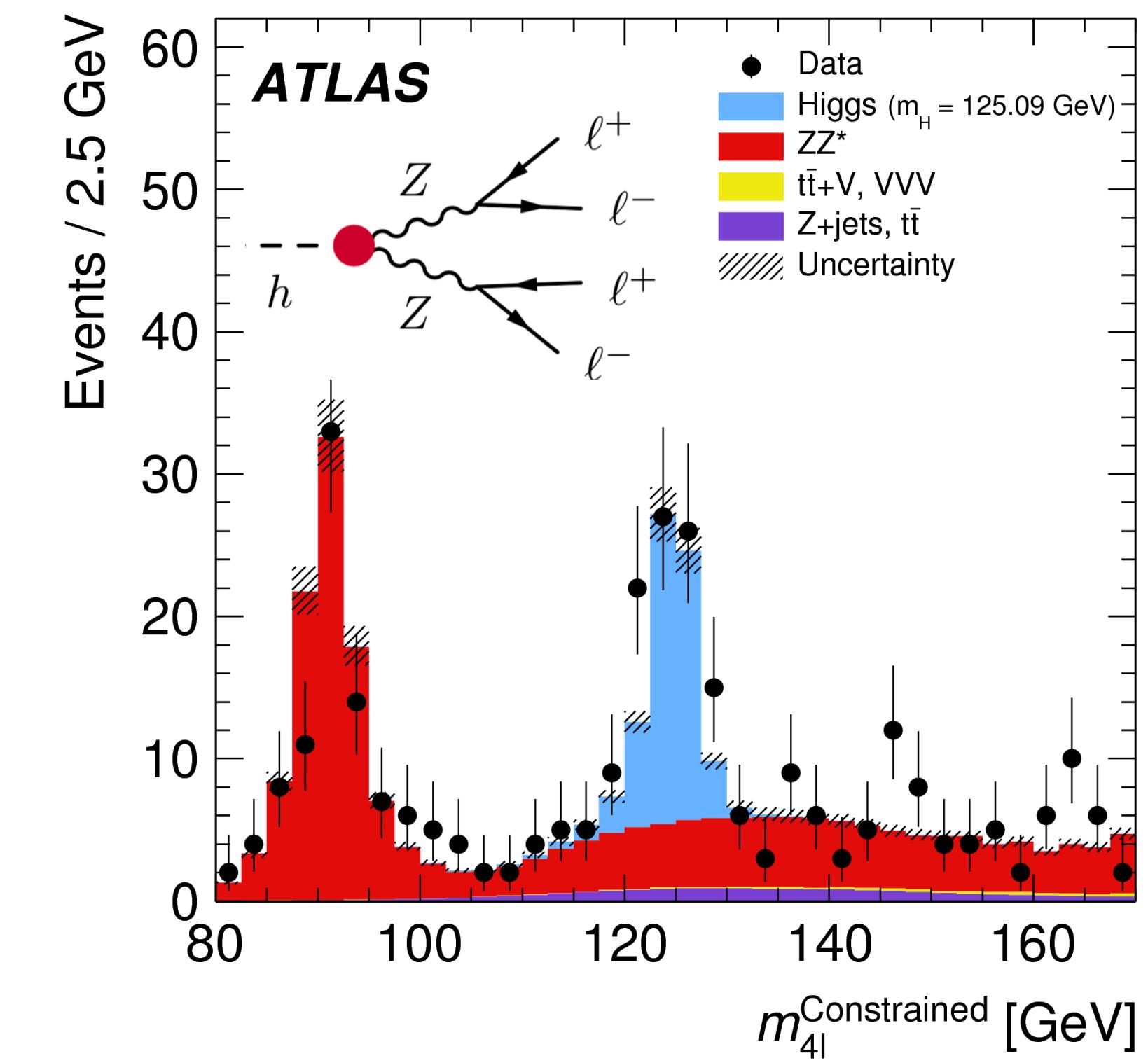
$p(x|\theta)$ cannot be calculated

Solve it with summary statistics



High-dimensional event data x

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

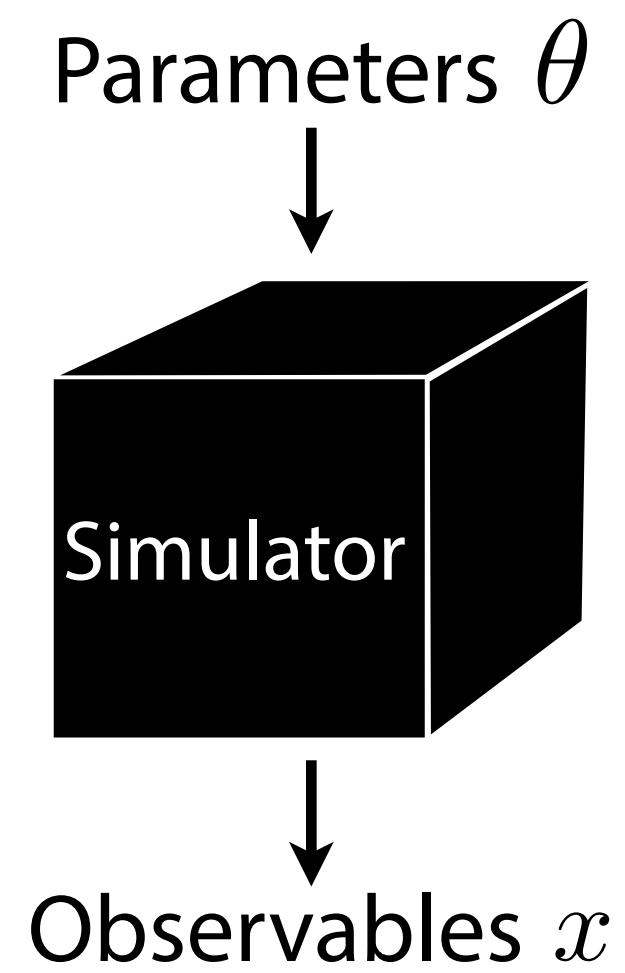


One or two summary statistics x'

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

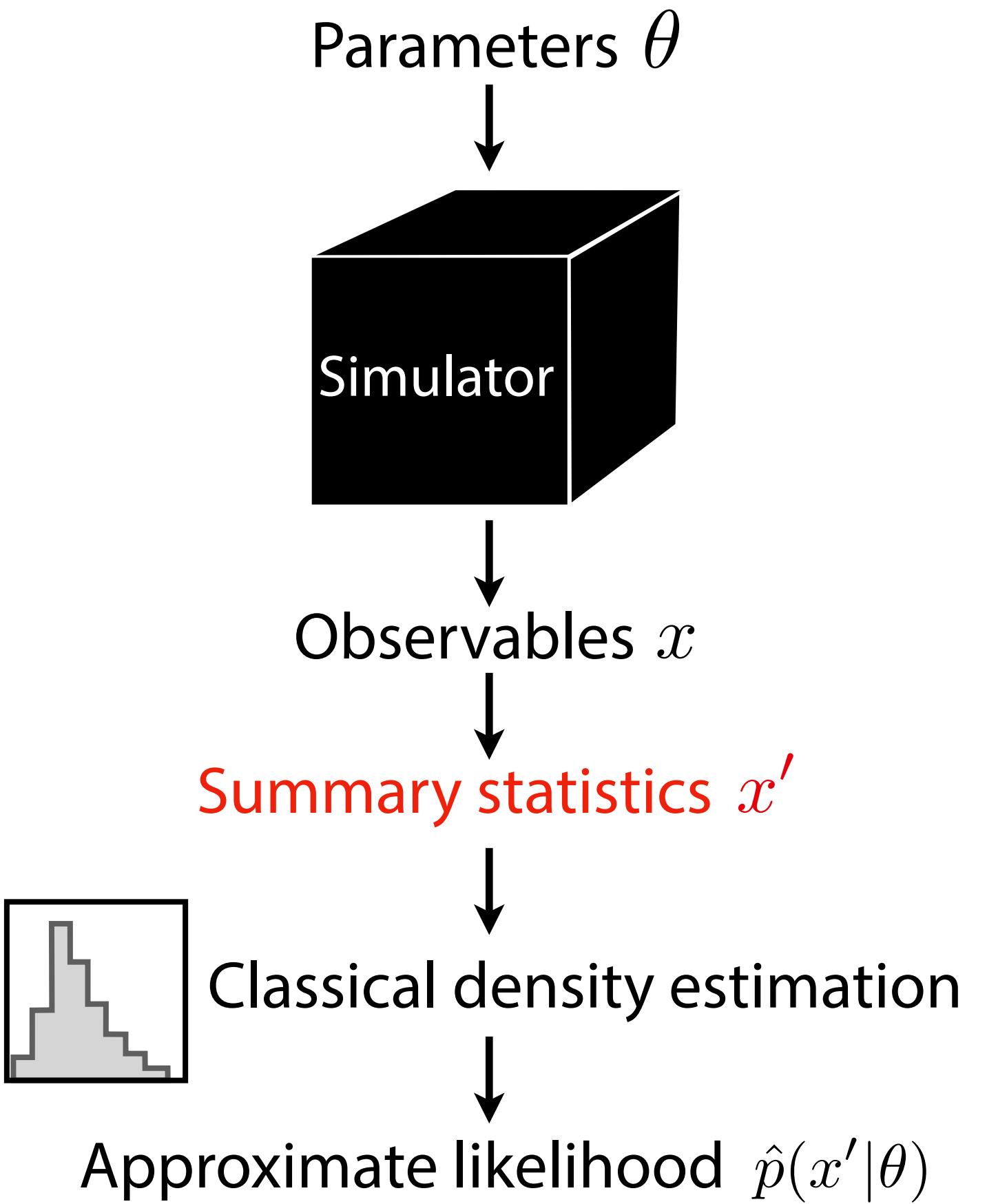
Inference by estimating the likelihood

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



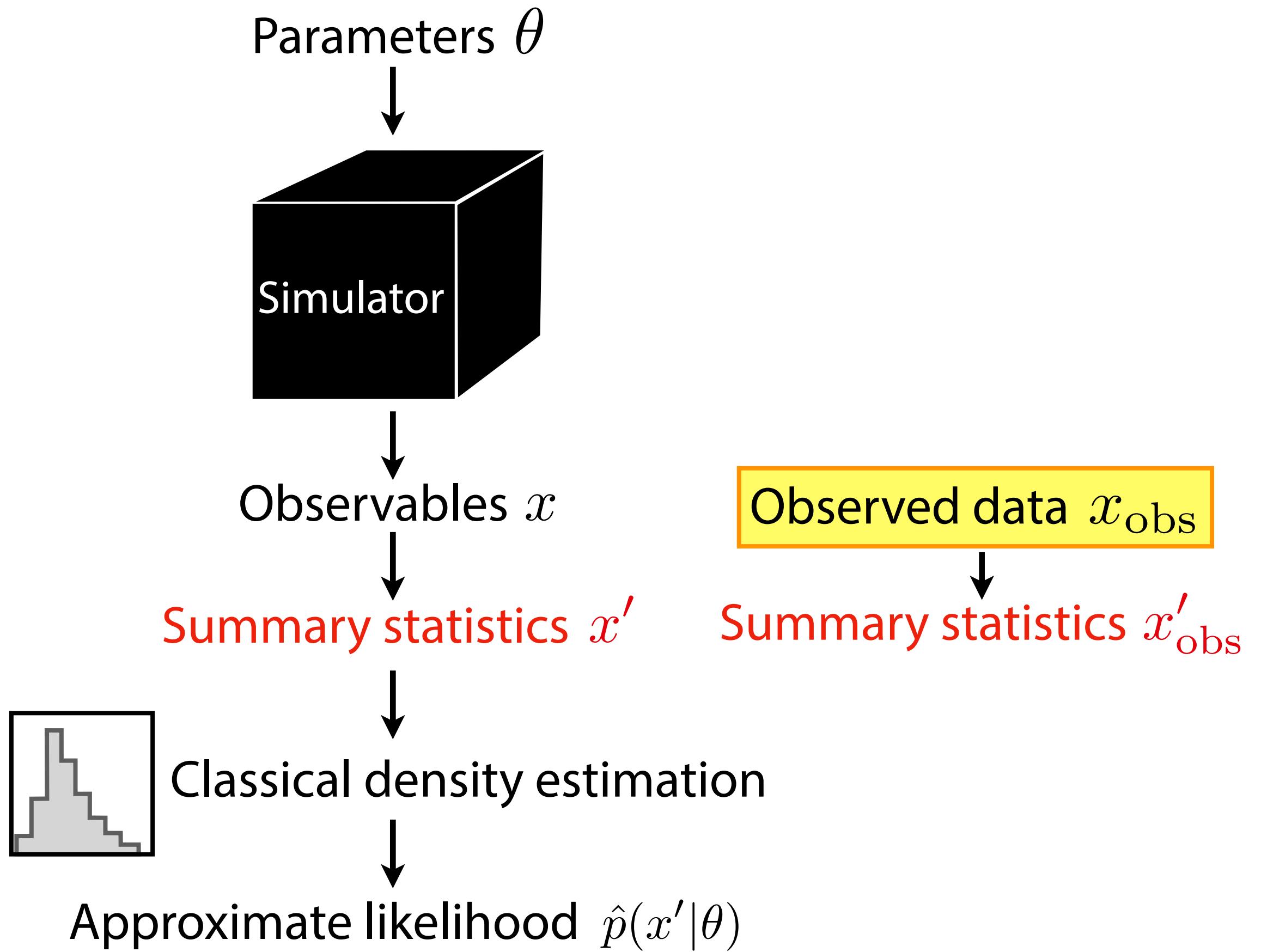
Inference by estimating the likelihood

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



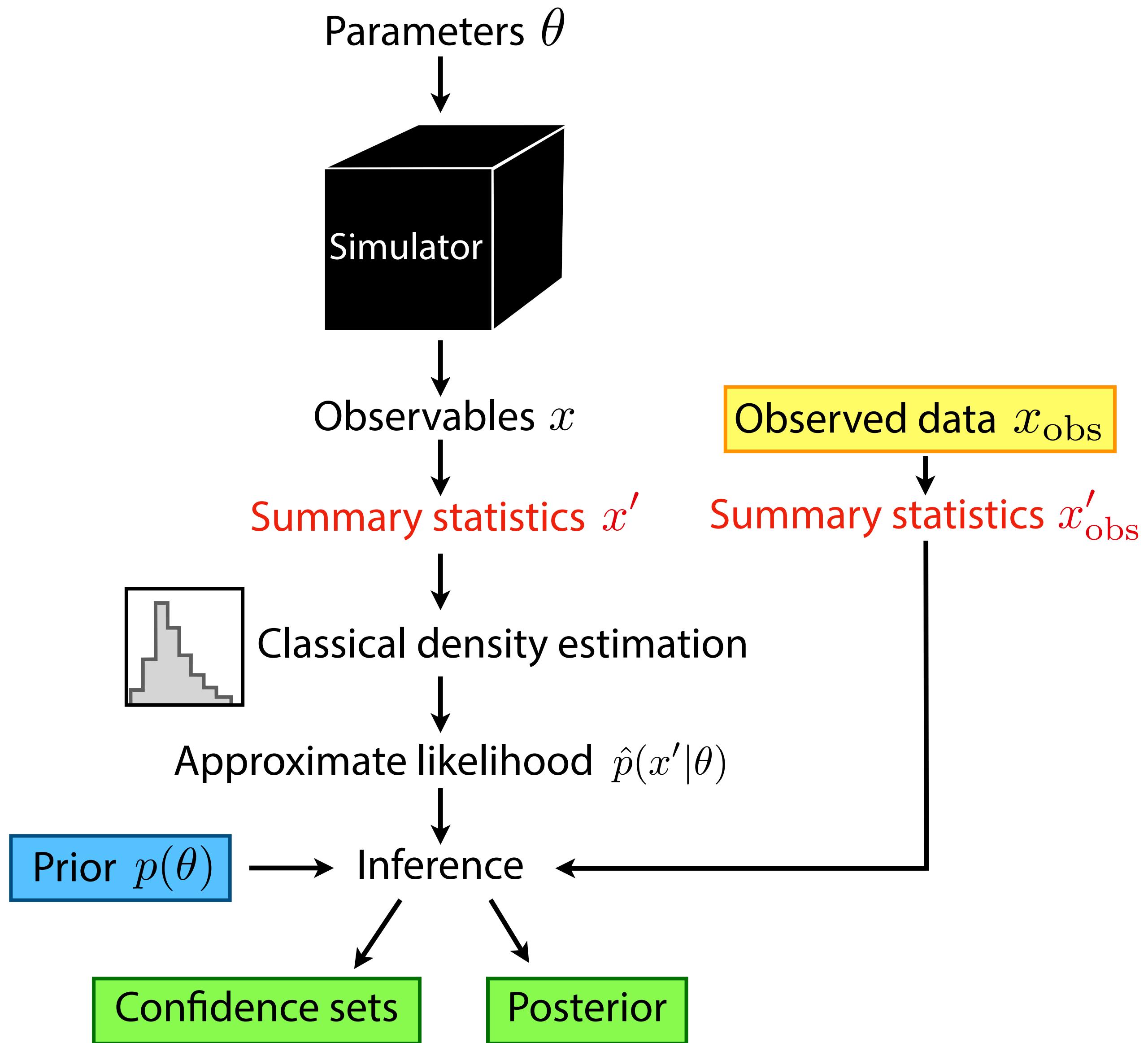
Inference by estimating the likelihood

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



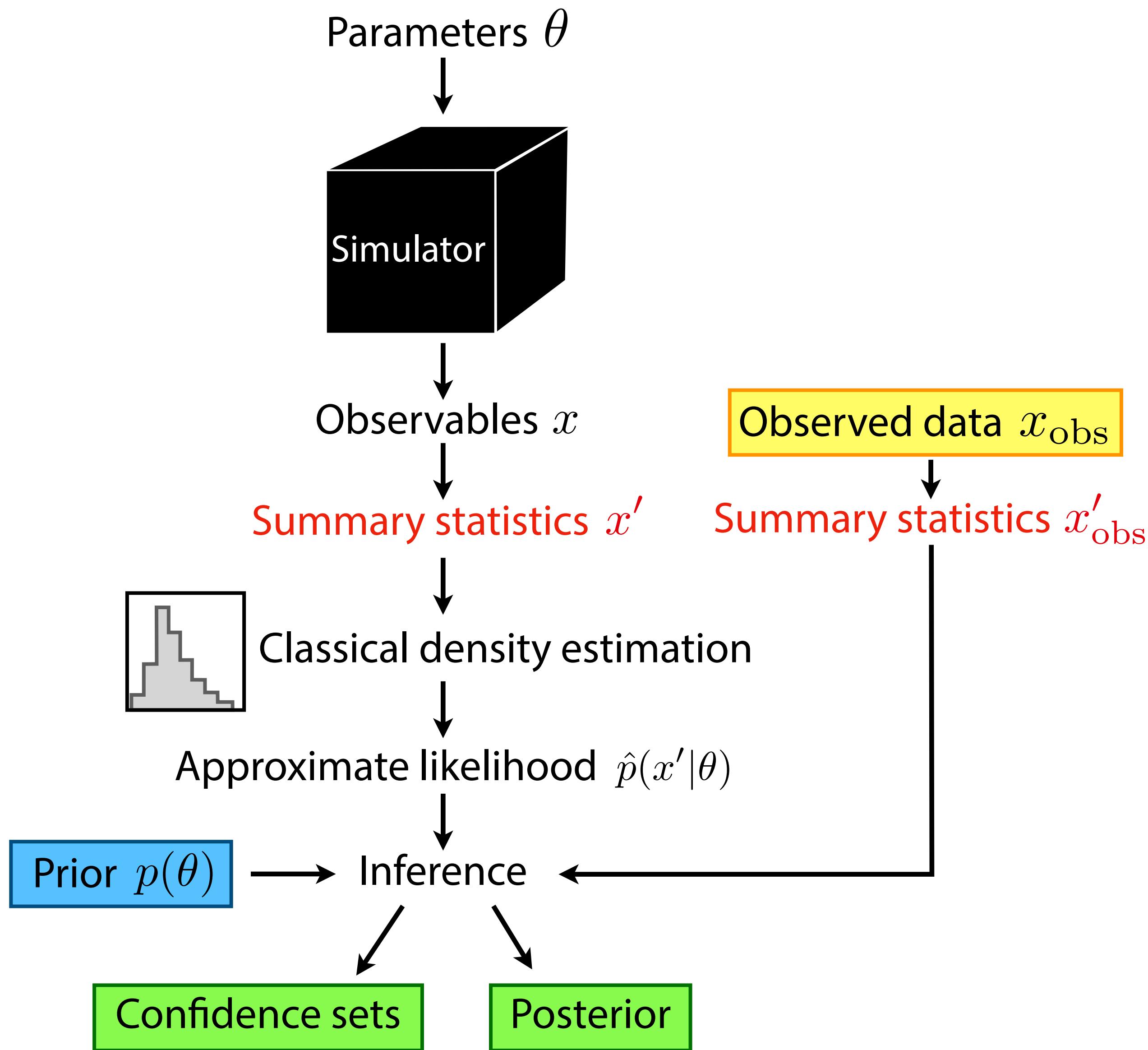
Inference by estimating the likelihood

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



Inference by estimating the likelihood

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



- Compression to summary statistics 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 statistics: 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

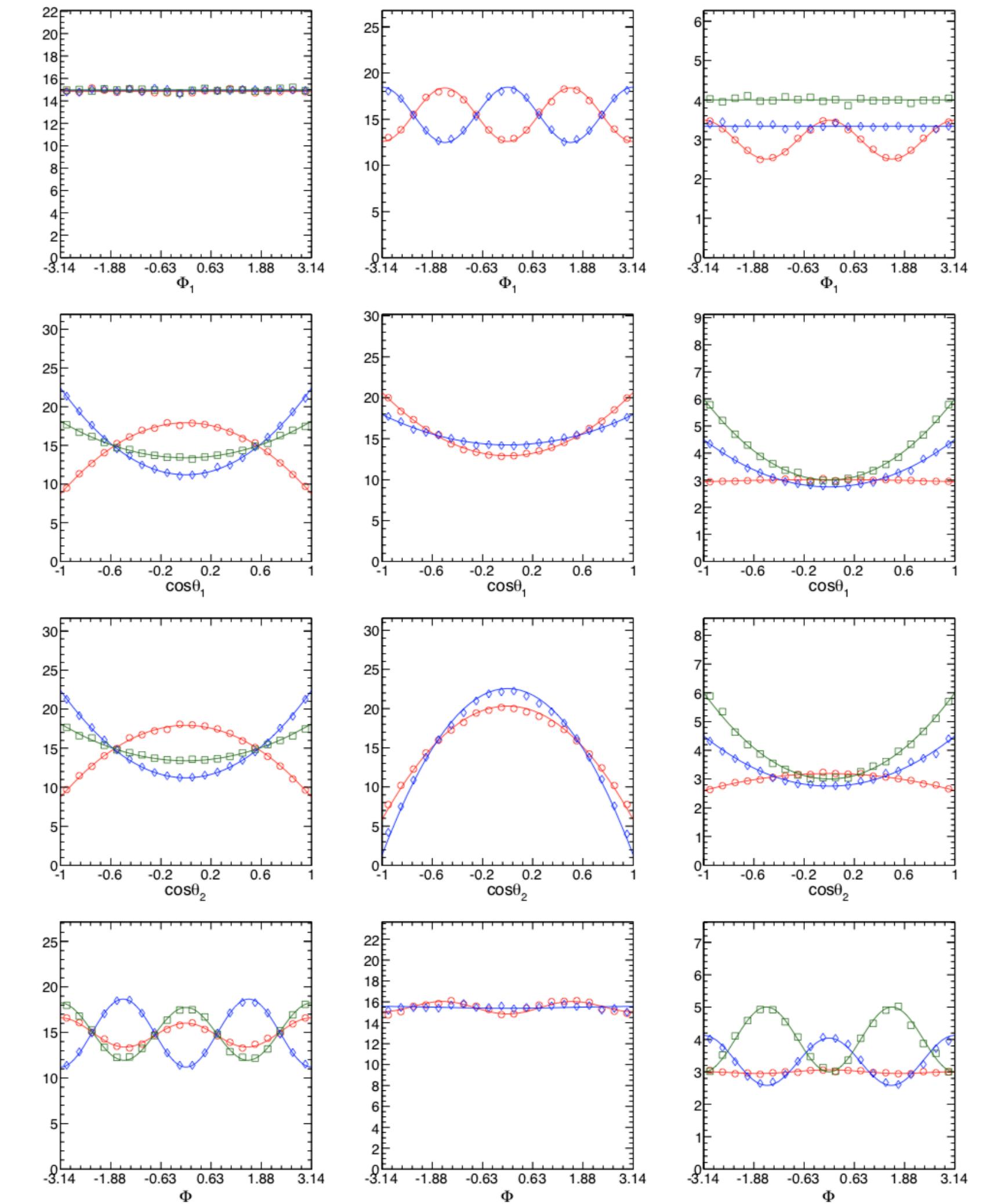
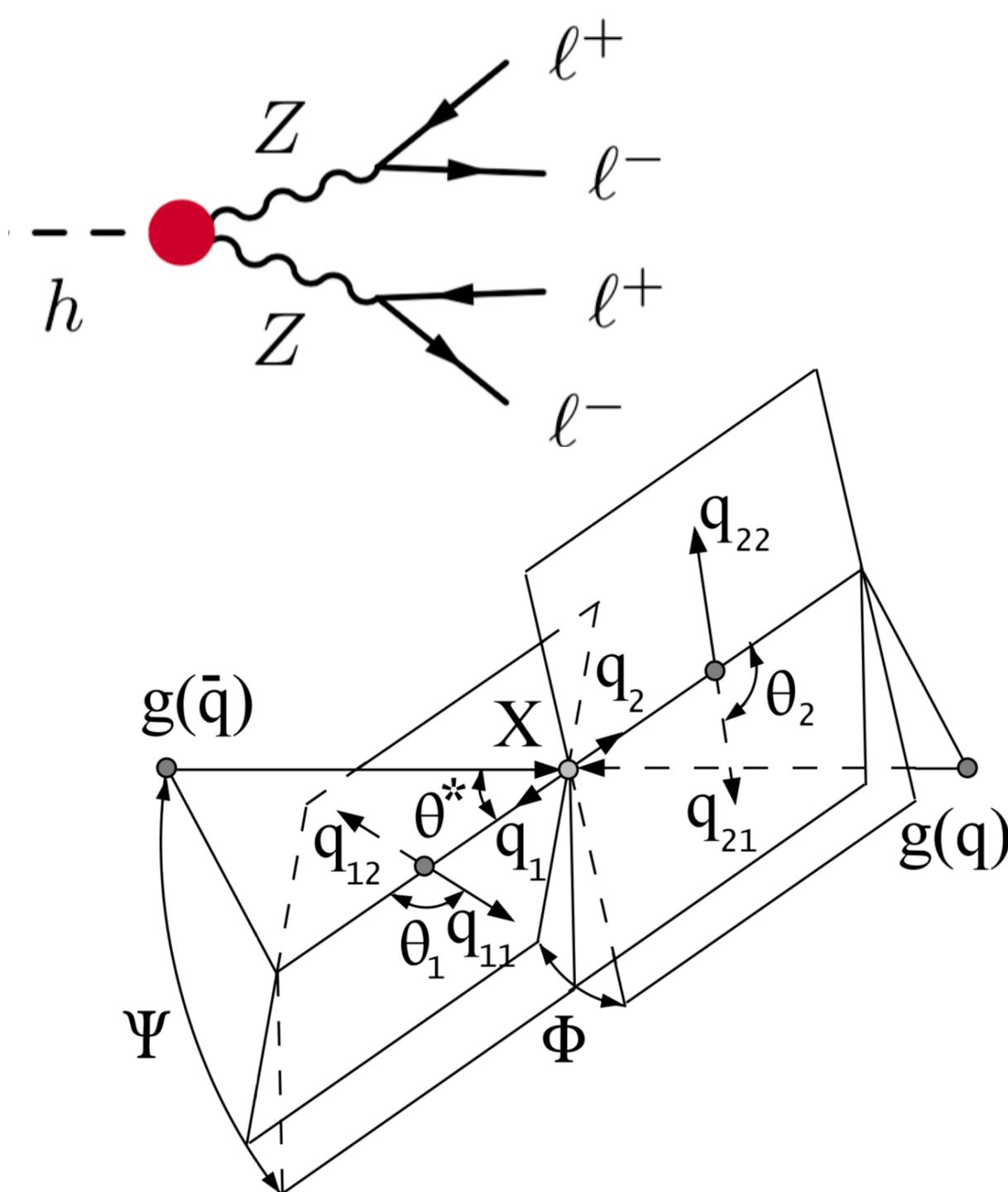
("fully differential cross section")

Summary statistics for LHC measurements?

- In many LHC problems there is no single good summary statistics: 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”)

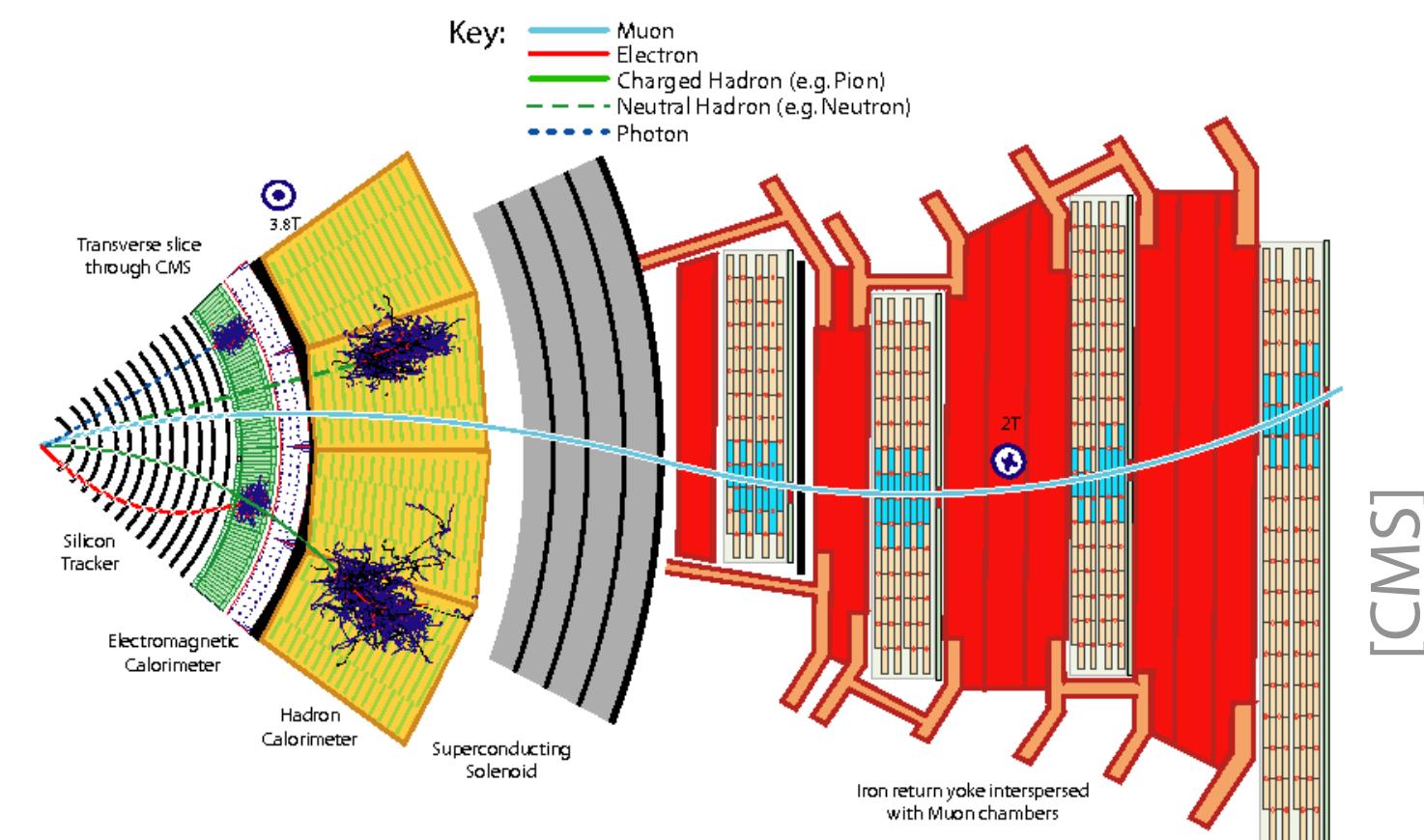


[Bolognesi et al. 1208.4018]

Solve it by approximating the integral

- Problem: high-dim. integral over shower / detector trajectories

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



Solve it by approximating the integral

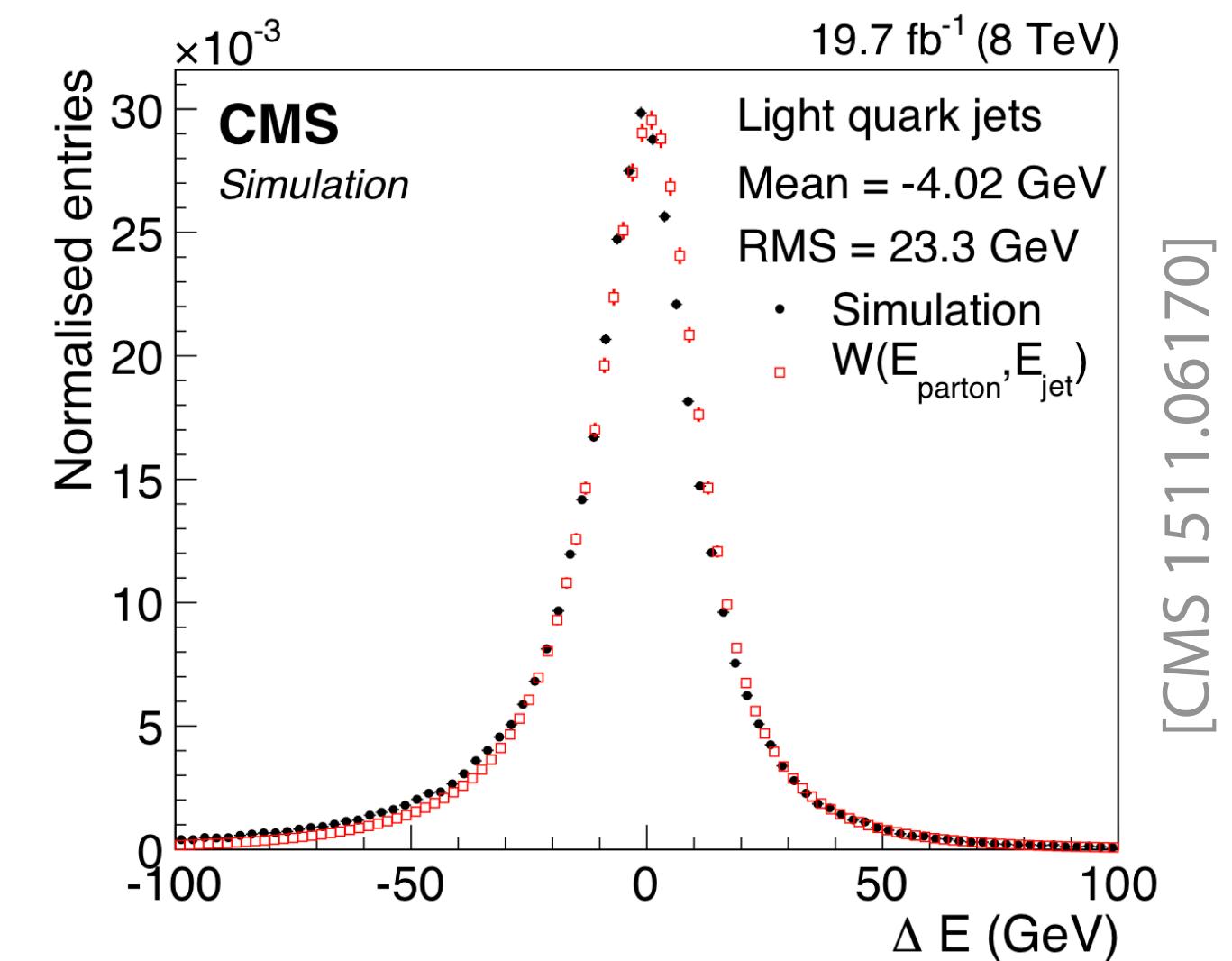
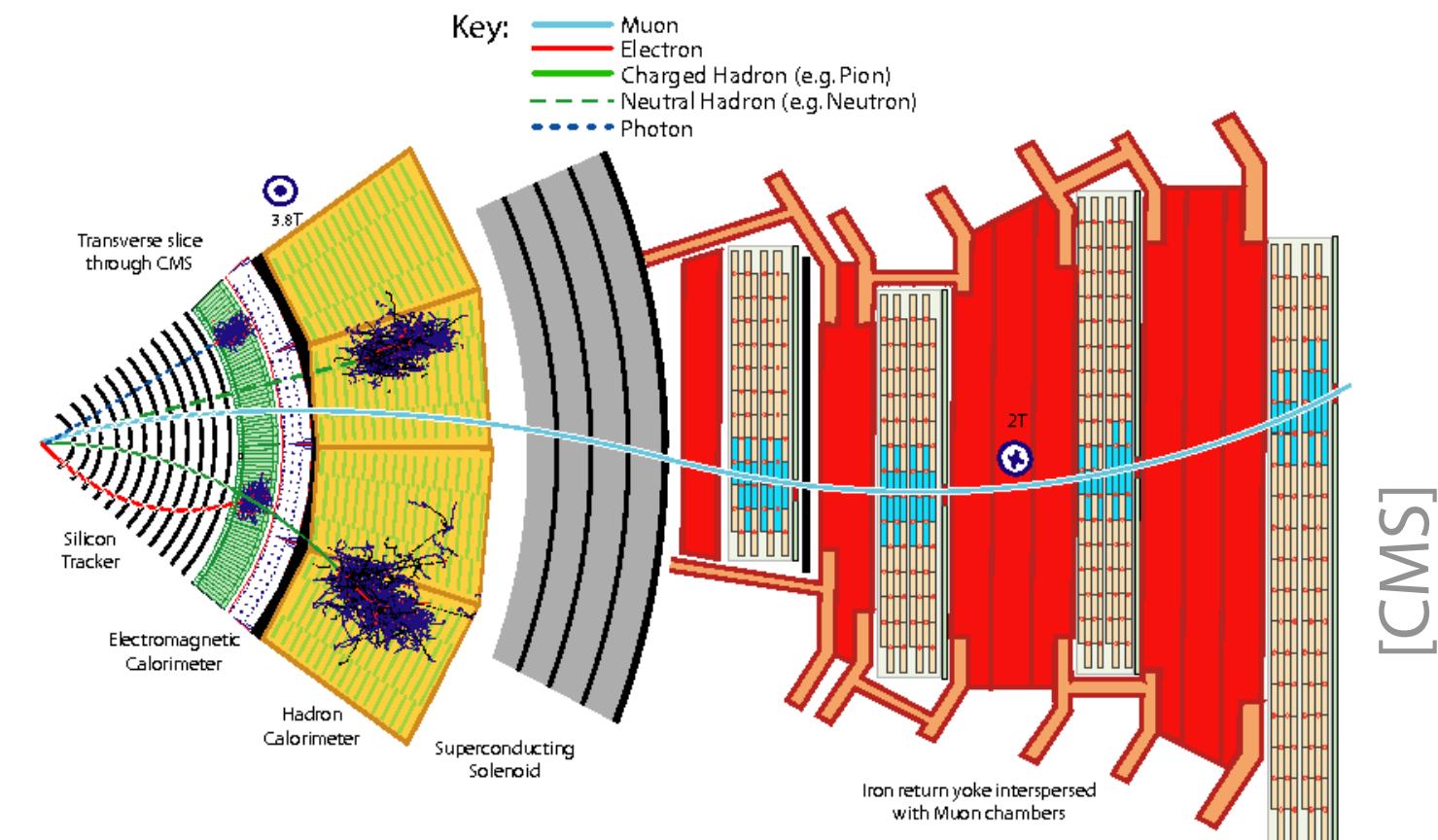
- Problem: high-dim. integral over **shower / detector trajectories**

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

- 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

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



Solve it by approximating the integral

- Problem: high-dim. integral over **shower / detector trajectories**

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

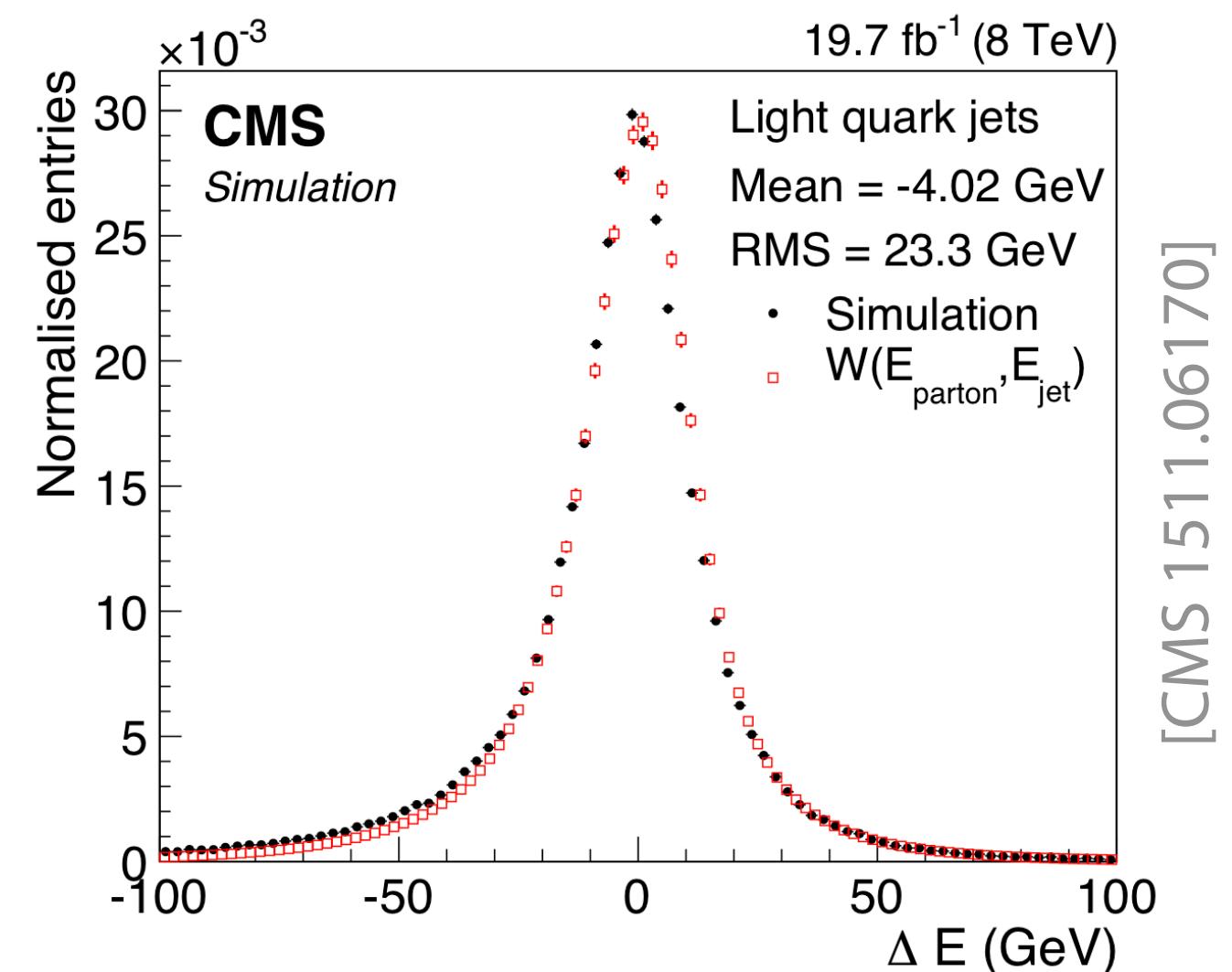
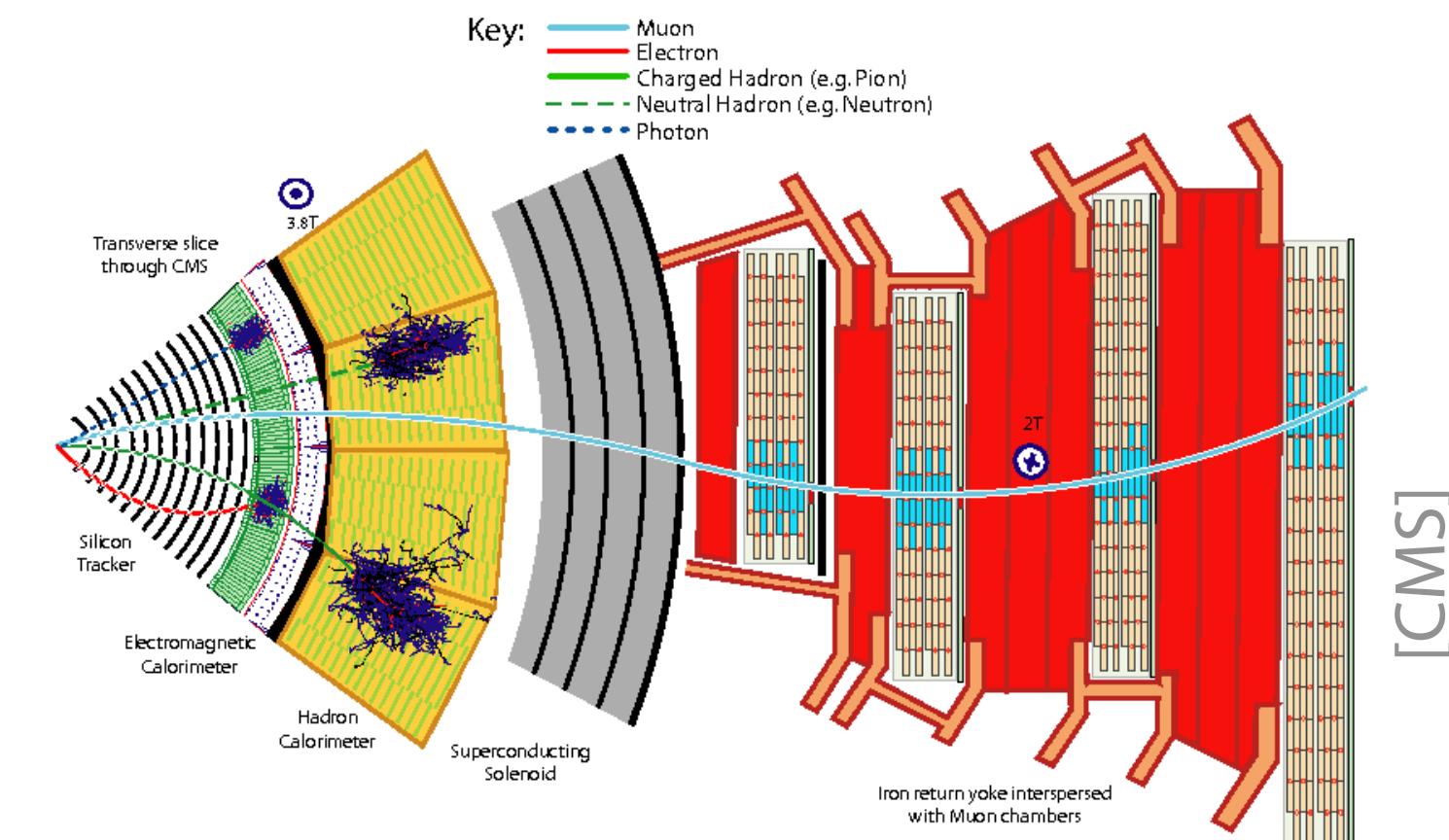
- Matrix Element Method (and similarly Optimal Observables): [K. Kondo 1988]

- approximate **shower + detector effects** into **transfer function** $\hat{p}(x|z_p)$
- explicitly calculate remaining integral

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

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

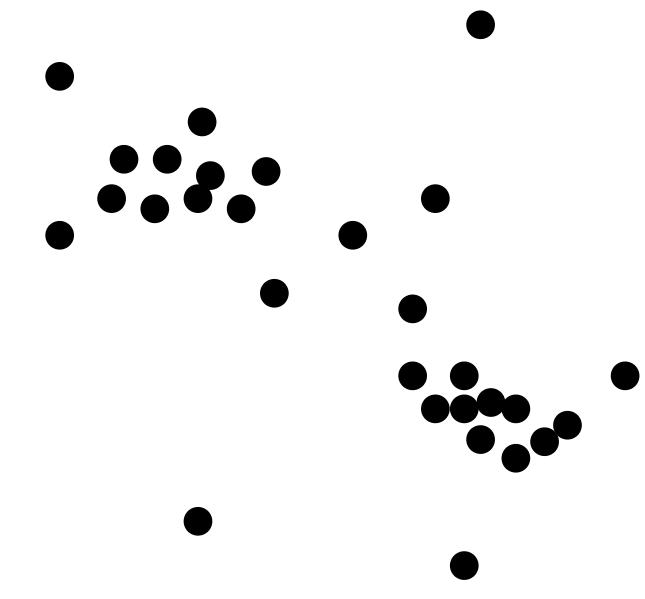
- ad-hoc transfer functions (what about extra radiation?)
- evaluation still requires calculating an expensive integral



Now, we can also use machine learning
to make inference possible.

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

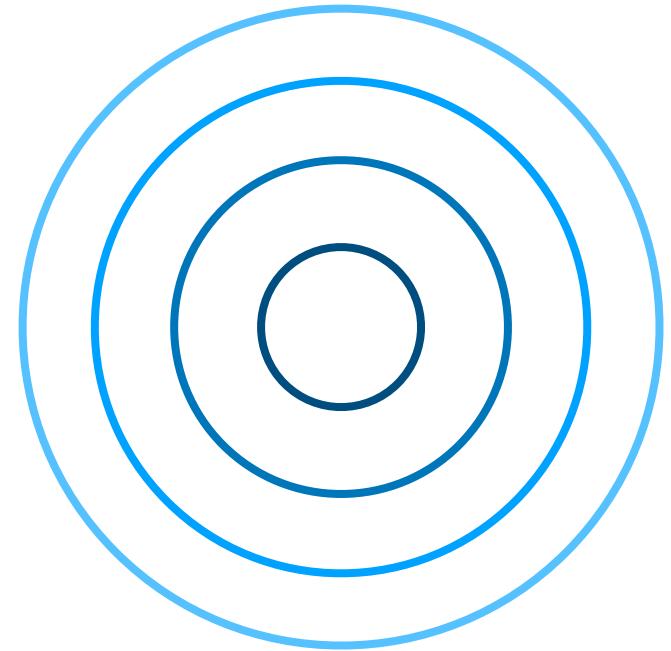
High-dimensional density estimation with normalizing flows



[lots of great work recently; see G. Papamakarios et al 1912.02762 for a review]

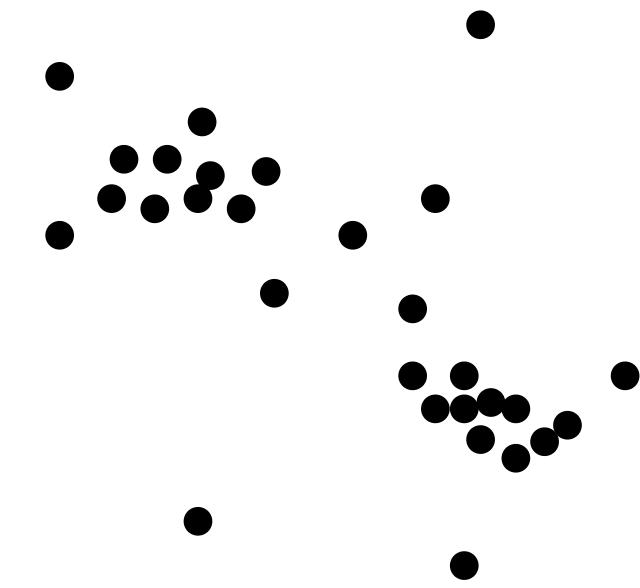
18/57

High-dimensional density estimation with normalizing flows

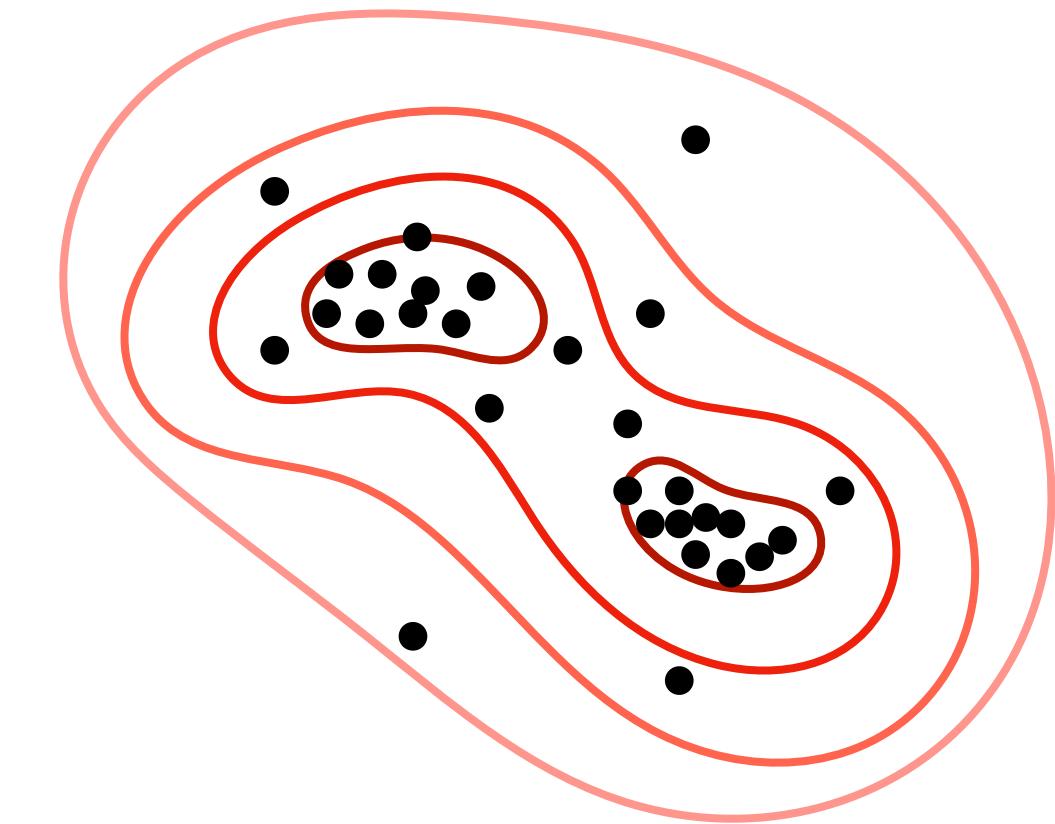
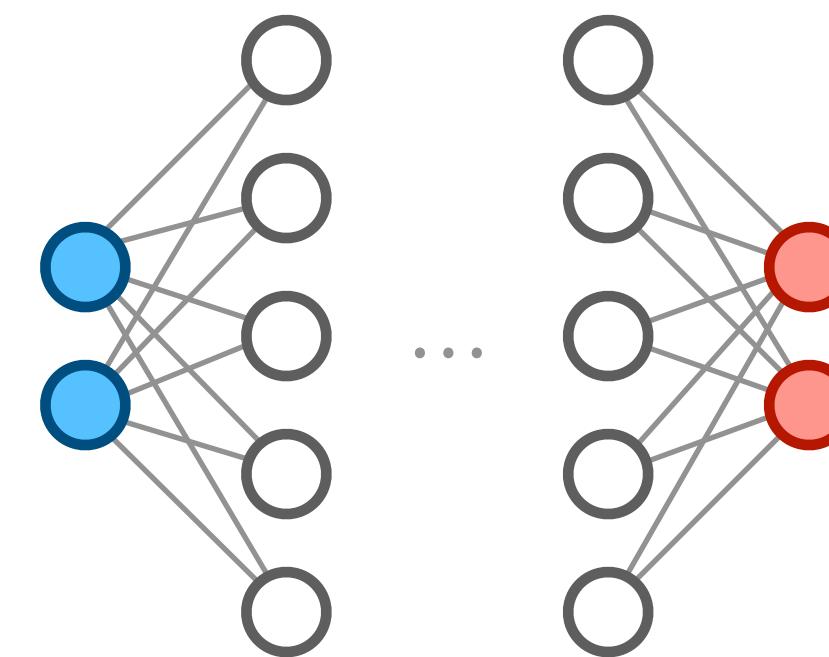
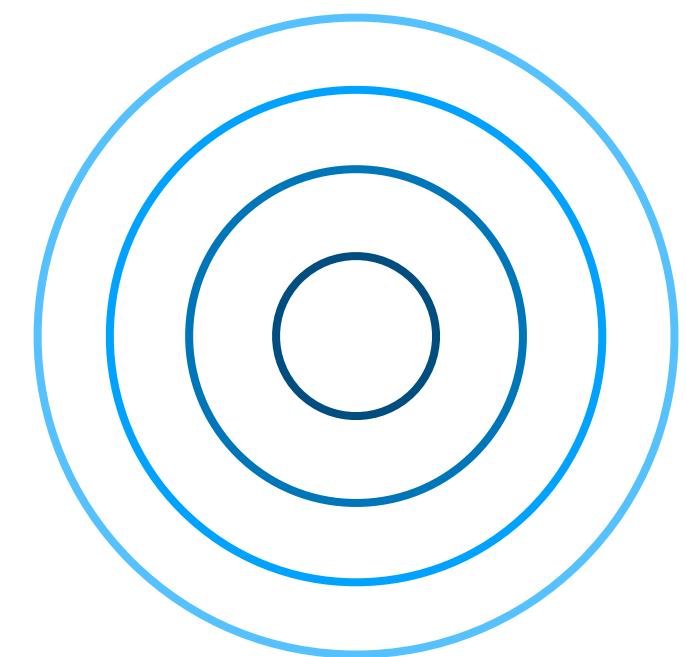


Simple base density

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



High-dimensional density estimation with normalizing flows



Simple base density

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

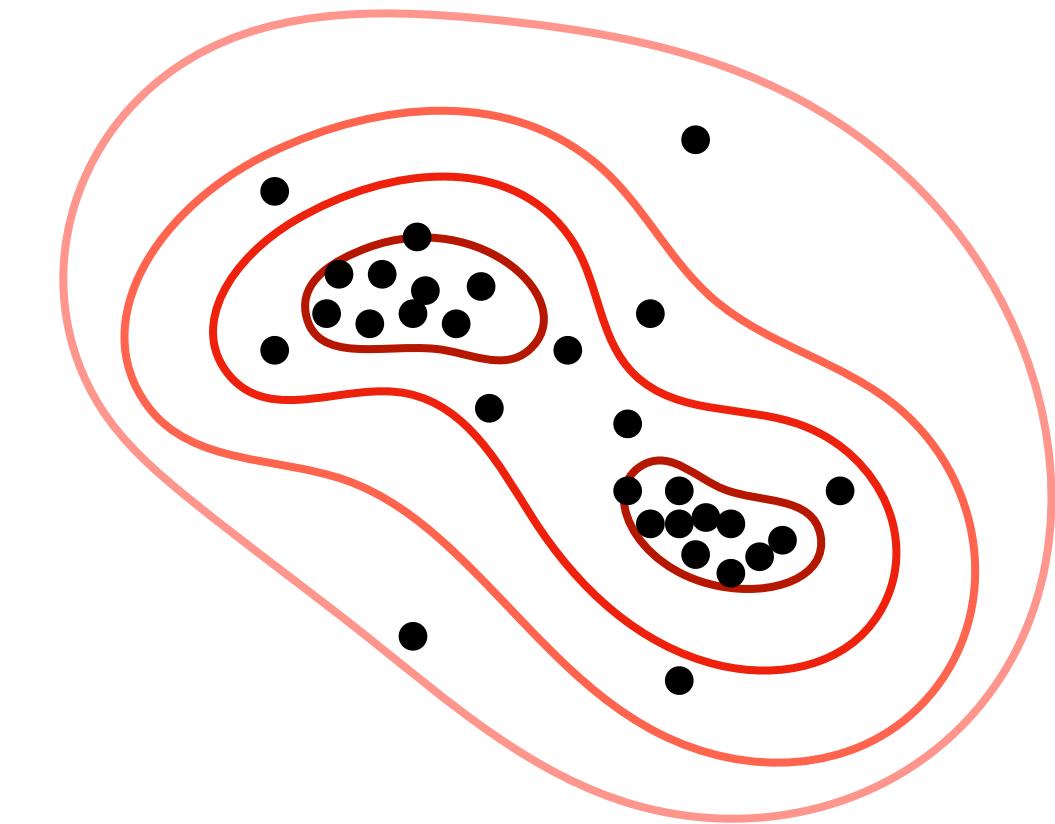
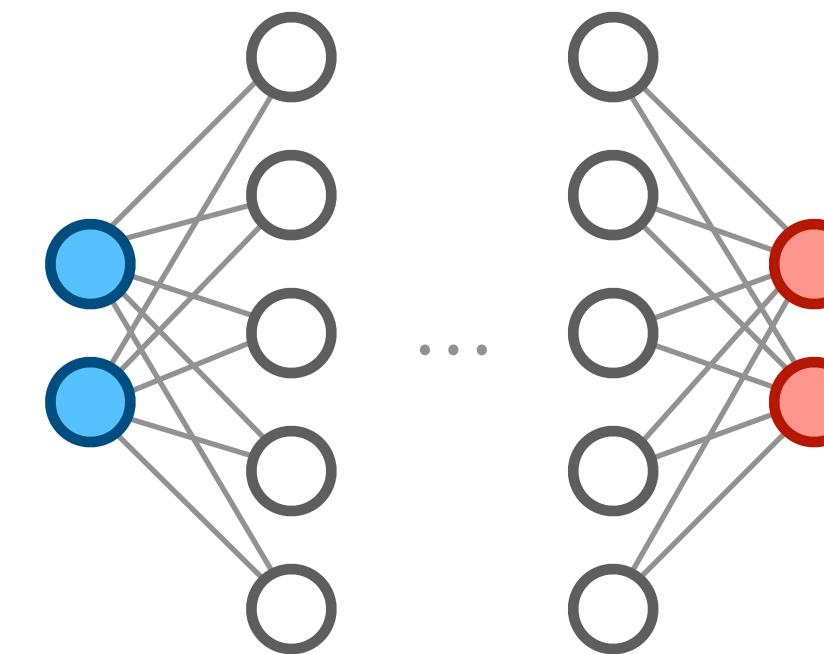
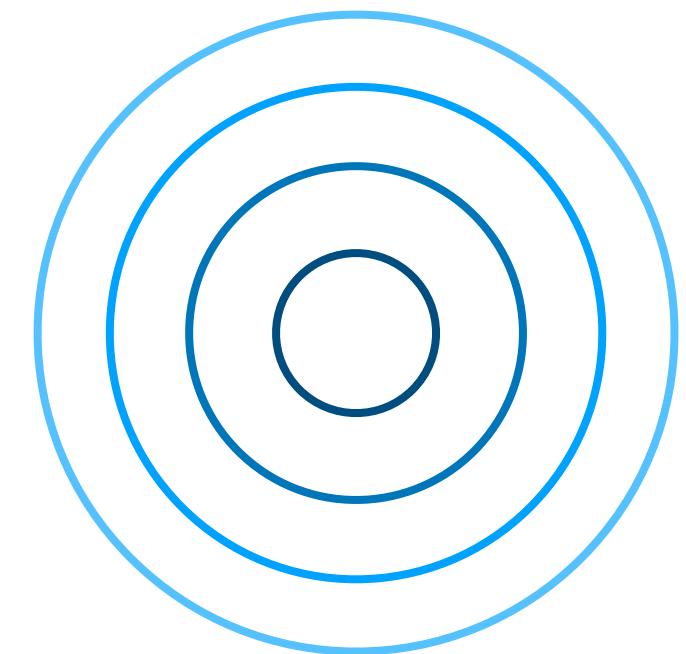
NN: transformation $x = f(u)$

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

Target density is given by

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

High-dimensional density estimation with normalizing flows



Simple base density

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

NN: transformation $x = f(u)$

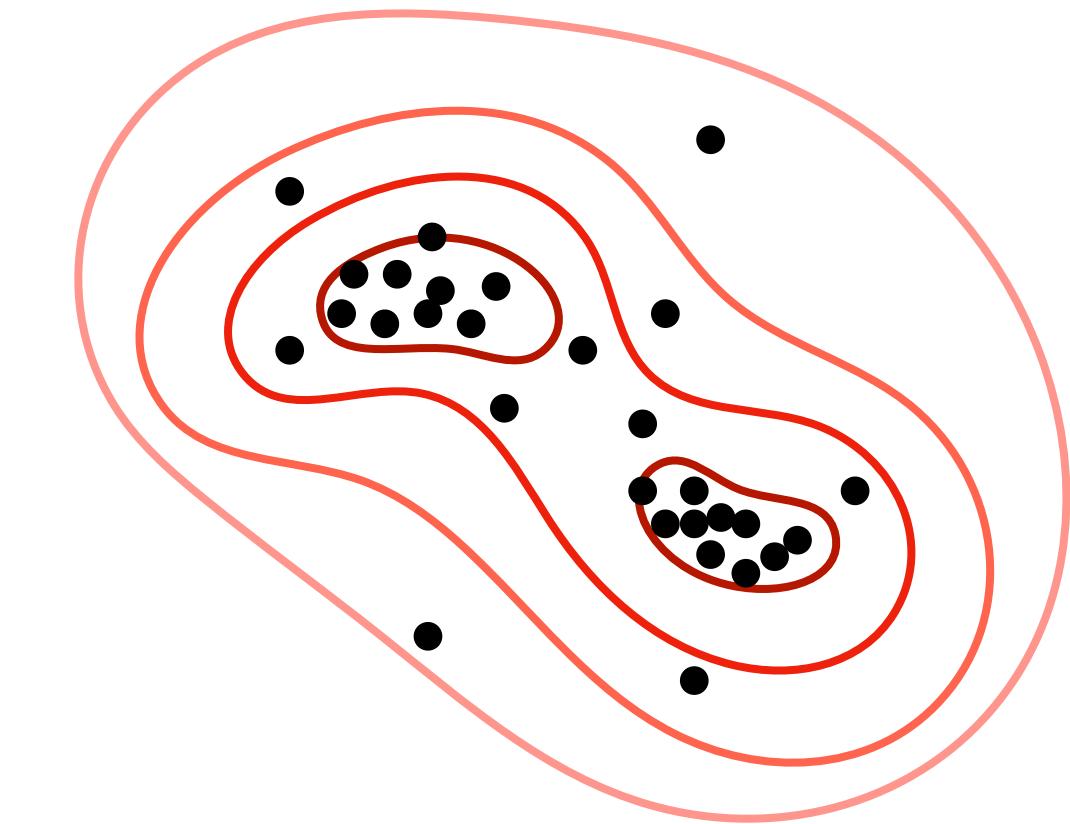
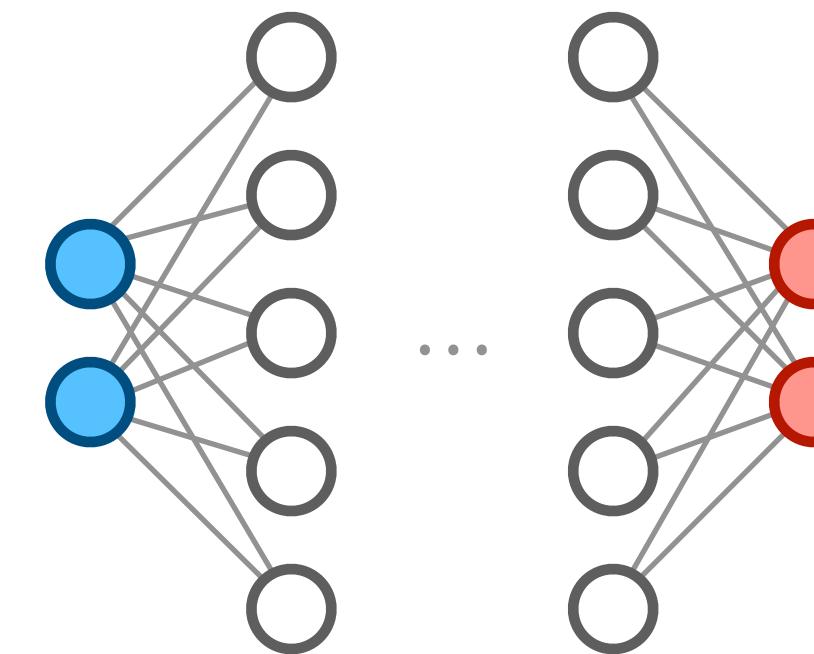
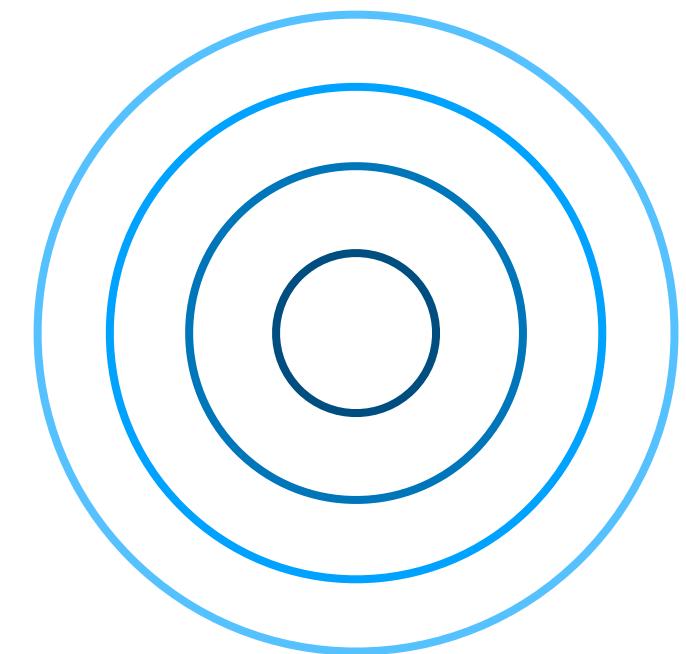
- one-to-one and invertible
- differentiable
- f^{-1} and $\det \nabla f$ are tractable

Target density is given by

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

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

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

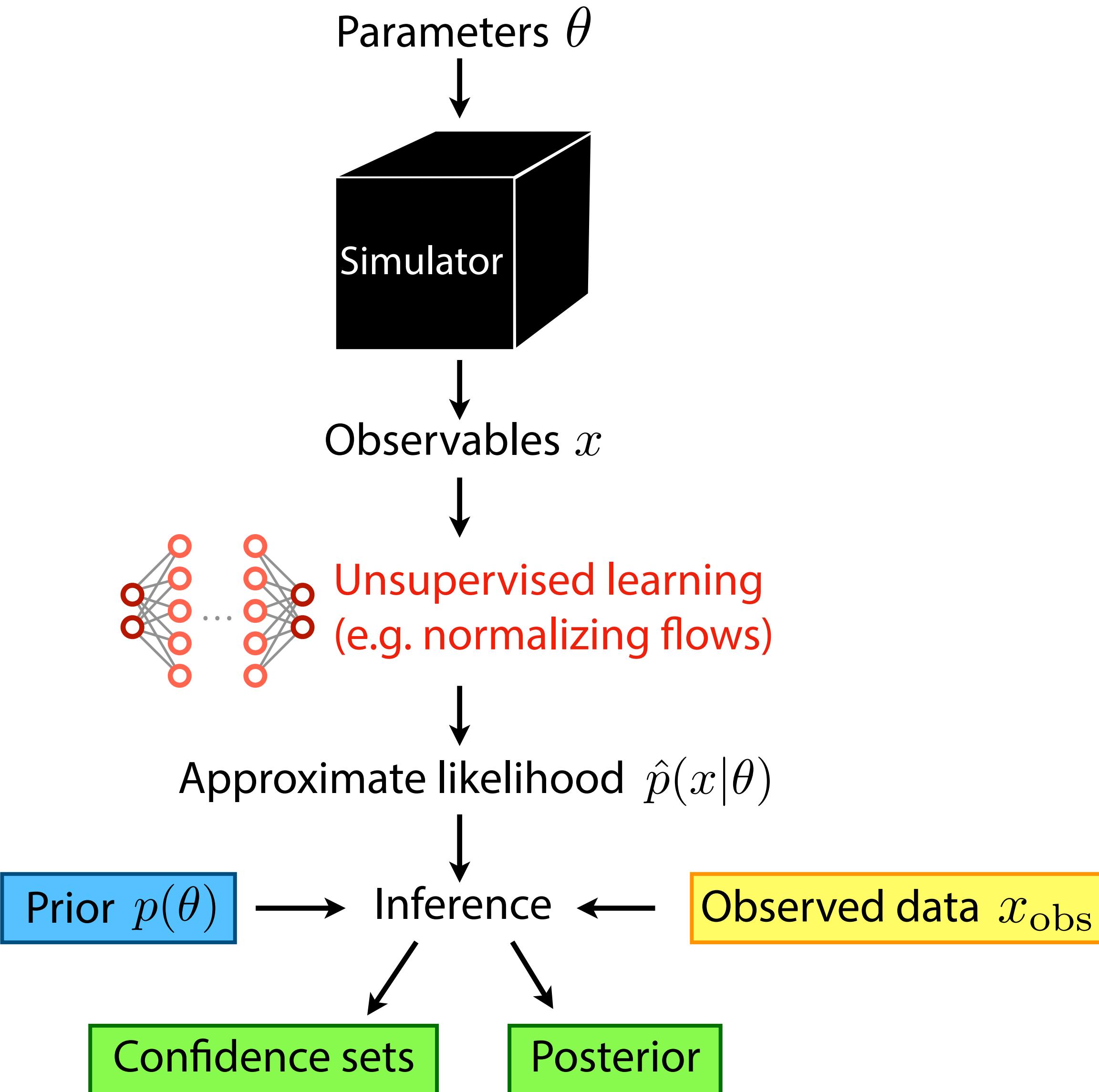
Target density is given by

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

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

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

Inference with neural likelihood estimation



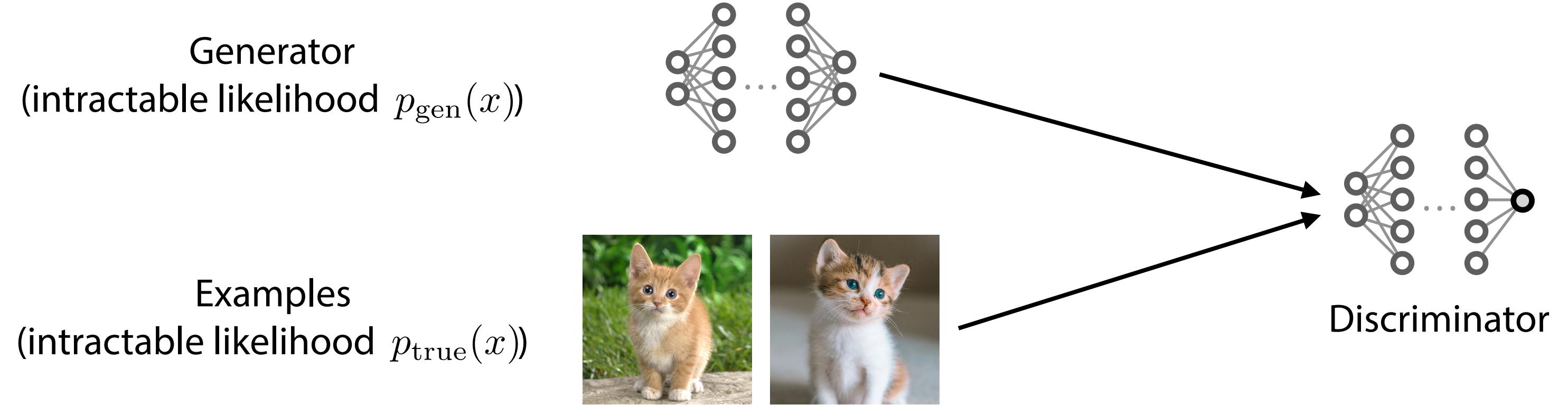
[G. Papamakarios, D. Sterratt, I. Murray 1805.07226;
J.-M. Lueckmann, G. Bassetto, T. Karaletsos, J. Macke 1805.09294]

- 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 $\hat{p}(\theta|x_{\text{obs}})$

[G. Papamakarios et al 1605.06376;
J.-M. Lueckmann et al 1711.01861]

The likelihood ratio trick

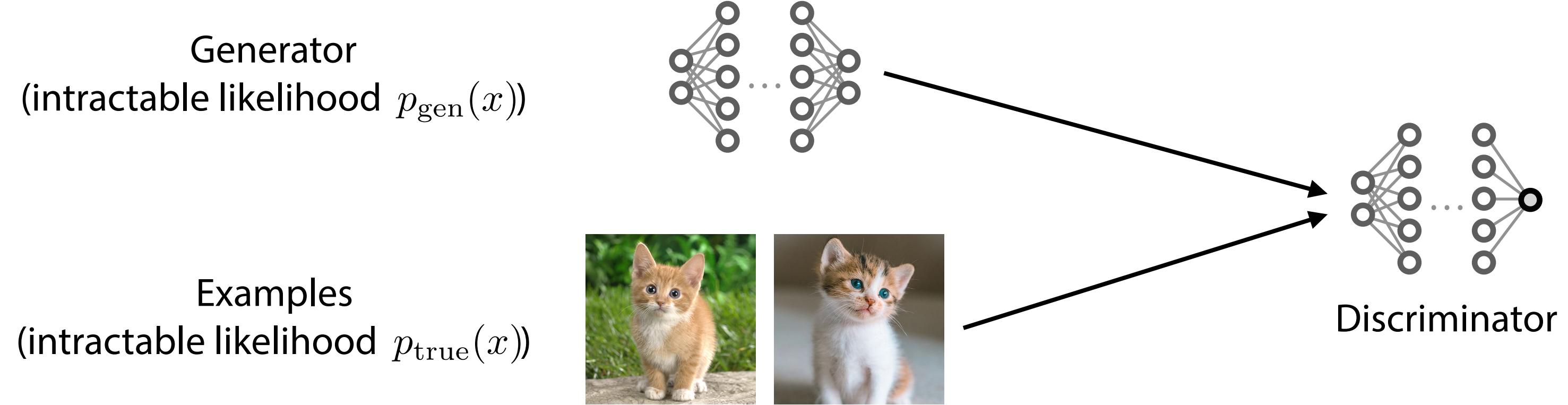
- Generative Adversarial Networks (GANs):



[I. Goodfellow et al. 1406.2661]

The likelihood ratio trick

- Generative Adversarial Networks (GANs):



[I. Goodfellow et al. 1406.2661]

Discriminator learns decision function

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

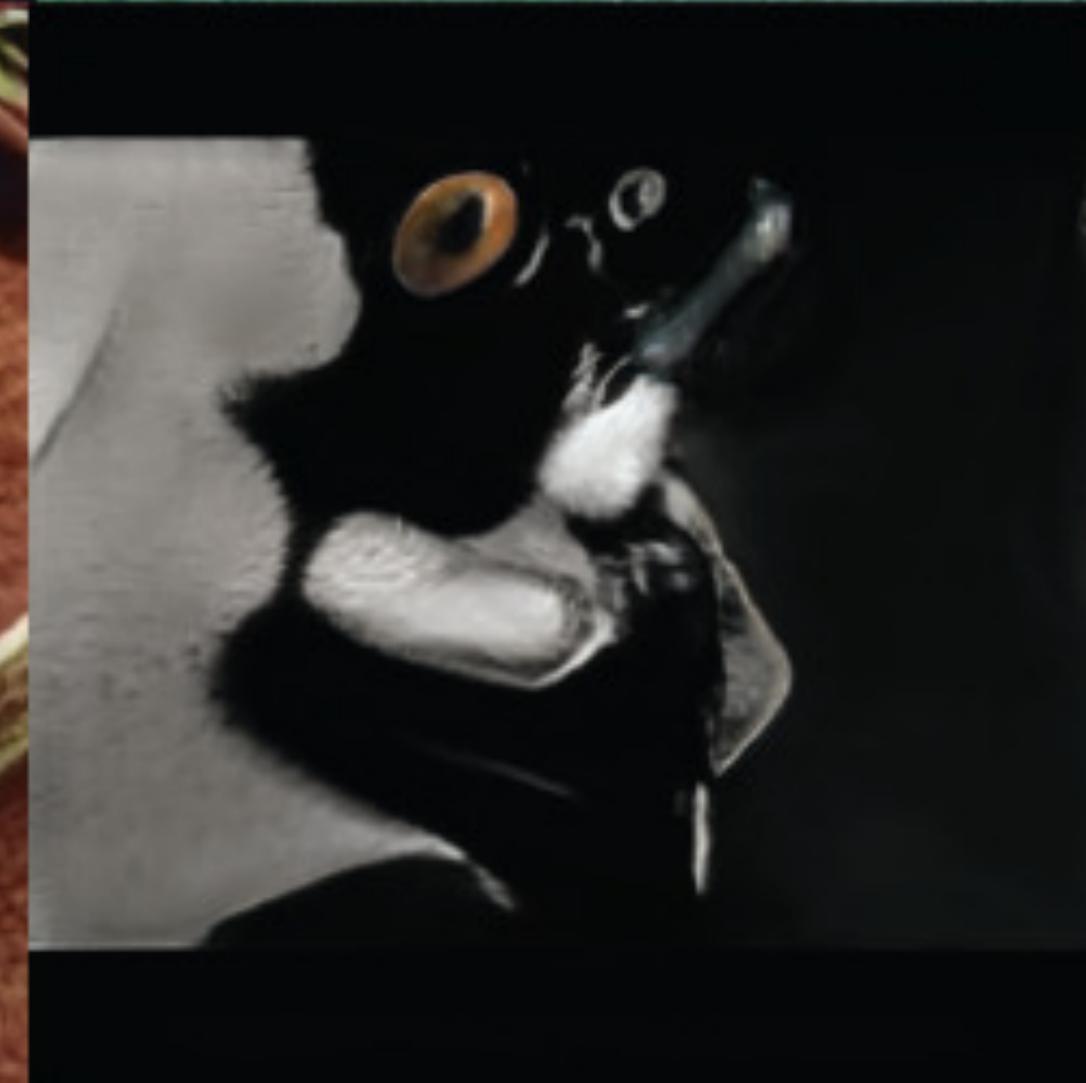
The likelihood ratio trick

- Generative Adversarial Networks (GANs):

Generator
(intractable likelihood $p_g(x)$)



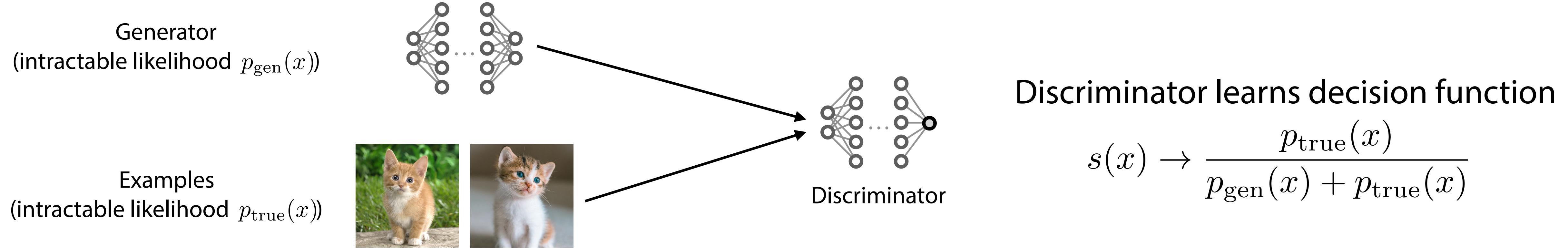
Examples
(intractable likelihood $p_t(x)$)



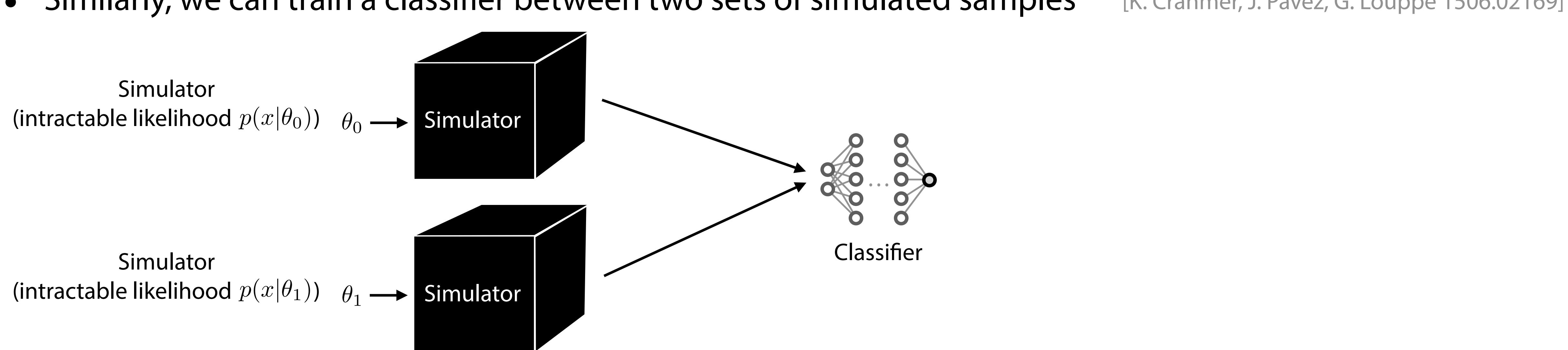
$$\text{decision function} = \frac{p_{\text{true}}(x)}{p_{\text{true}}(x) + p_{\text{generated}}(x)}$$

The likelihood ratio trick

- Generative Adversarial Networks (GANs):

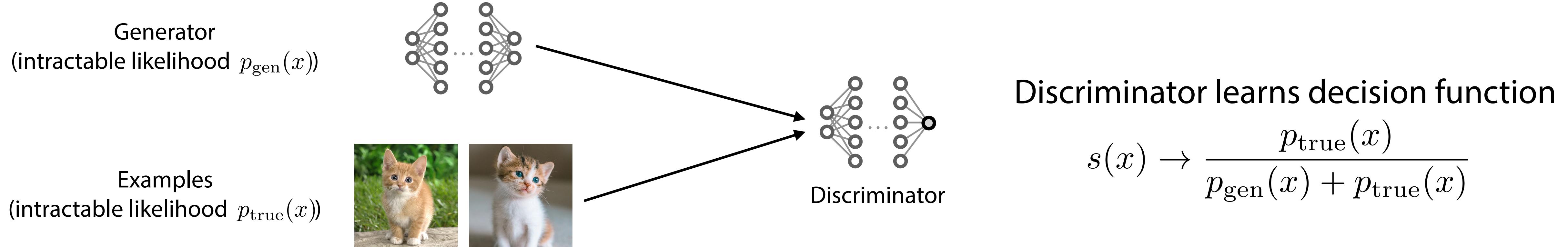


- Similarly, we can train a classifier between two sets of simulated samples

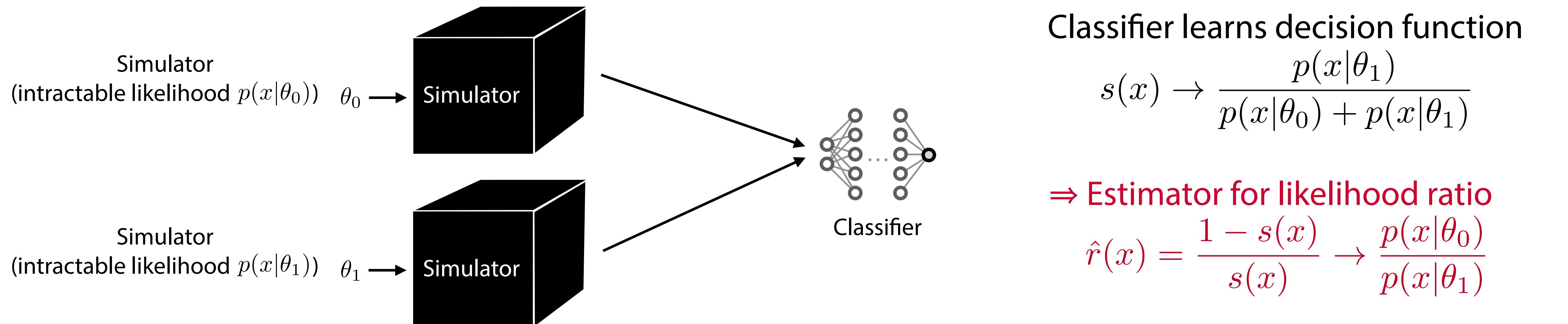


The likelihood ratio trick

- Generative Adversarial Networks (GANs):

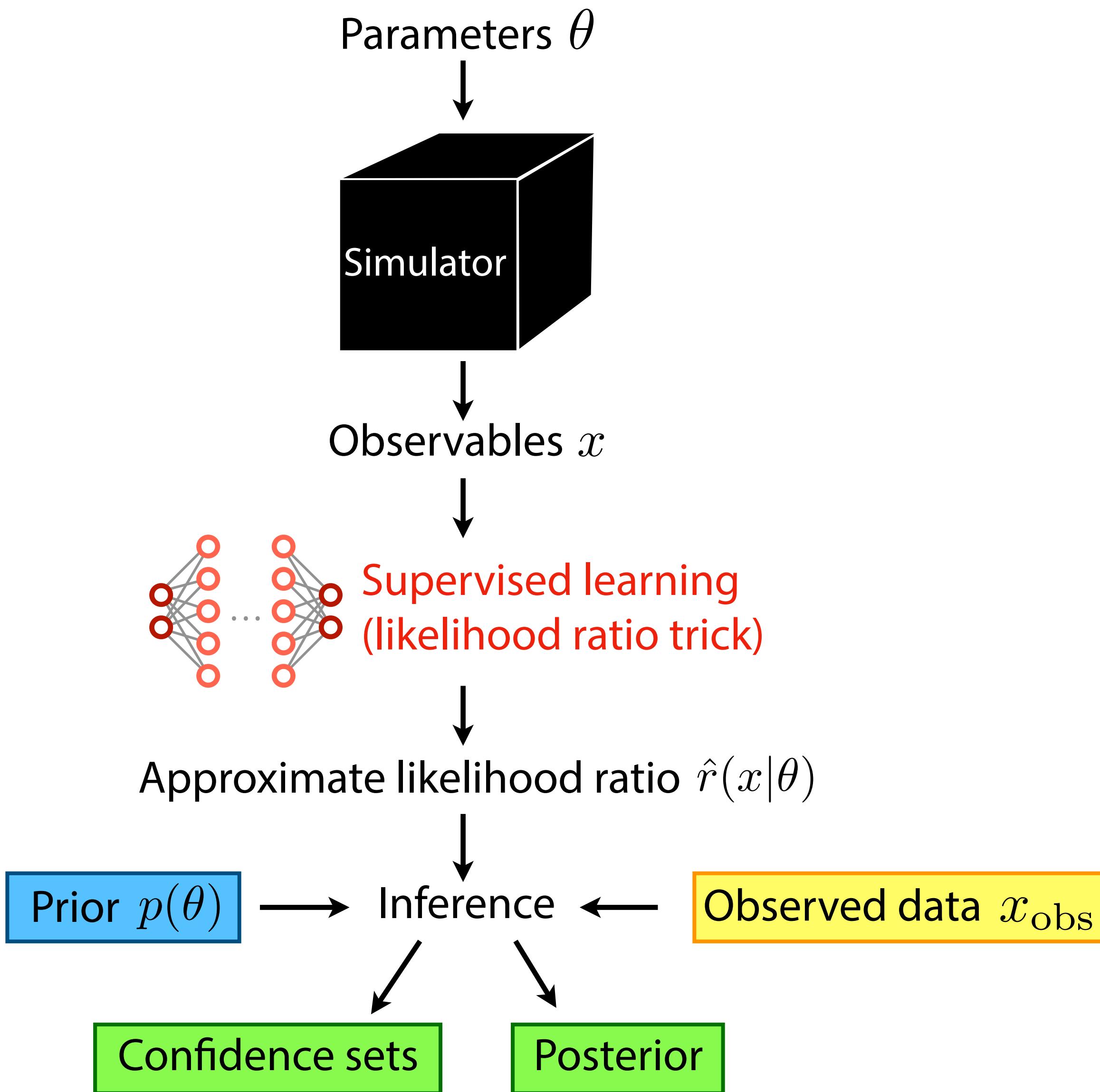


- Similarly, we can train a classifier between two sets of simulated samples



Inference by likelihood ratio trick

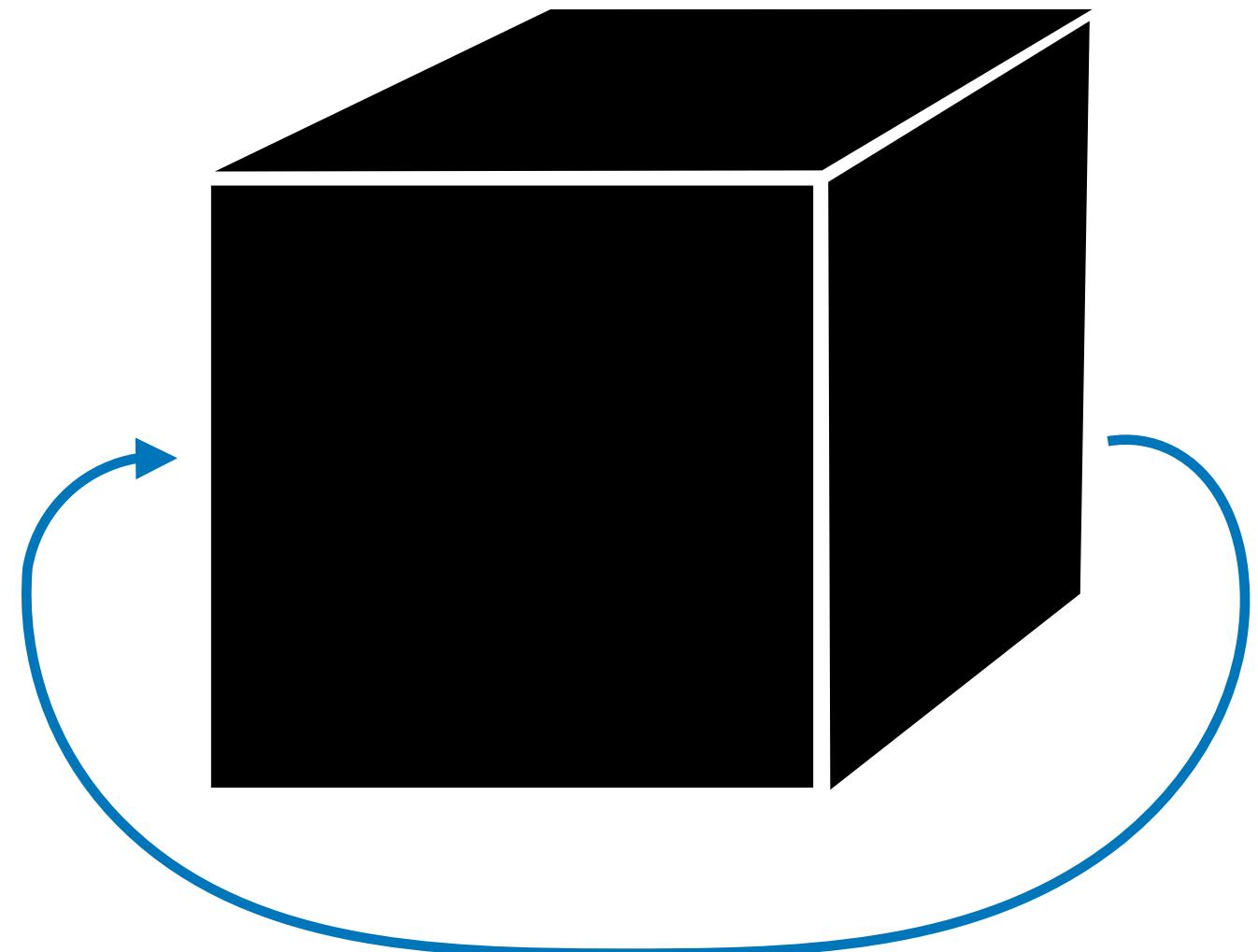
[K. Cranmer J. Pavez, G. Louppe 1506.02169]



- 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

Frontiers of simulation-based inference

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

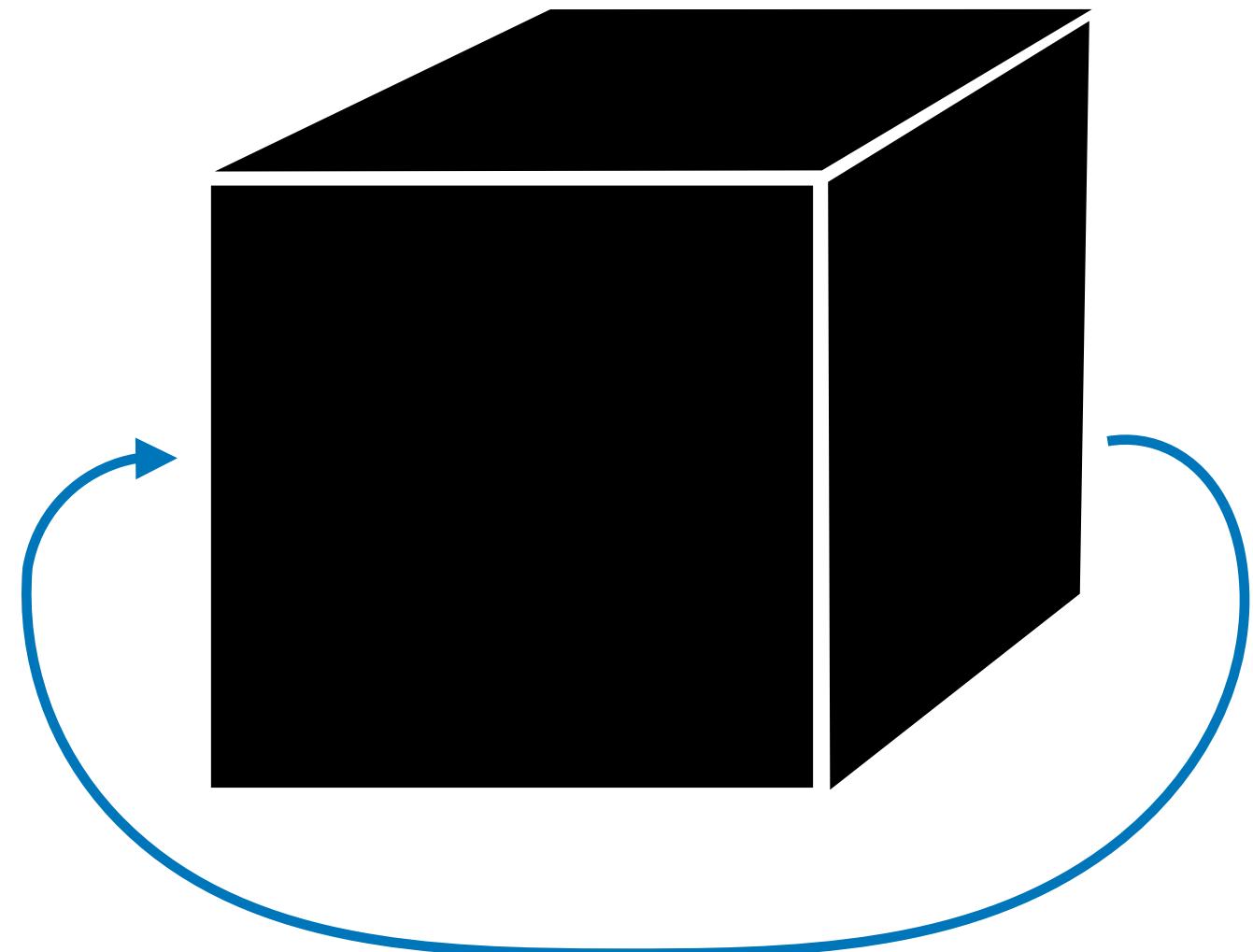


Active learning:

Iteratively guide simulator to
important parameter regions
based on past results

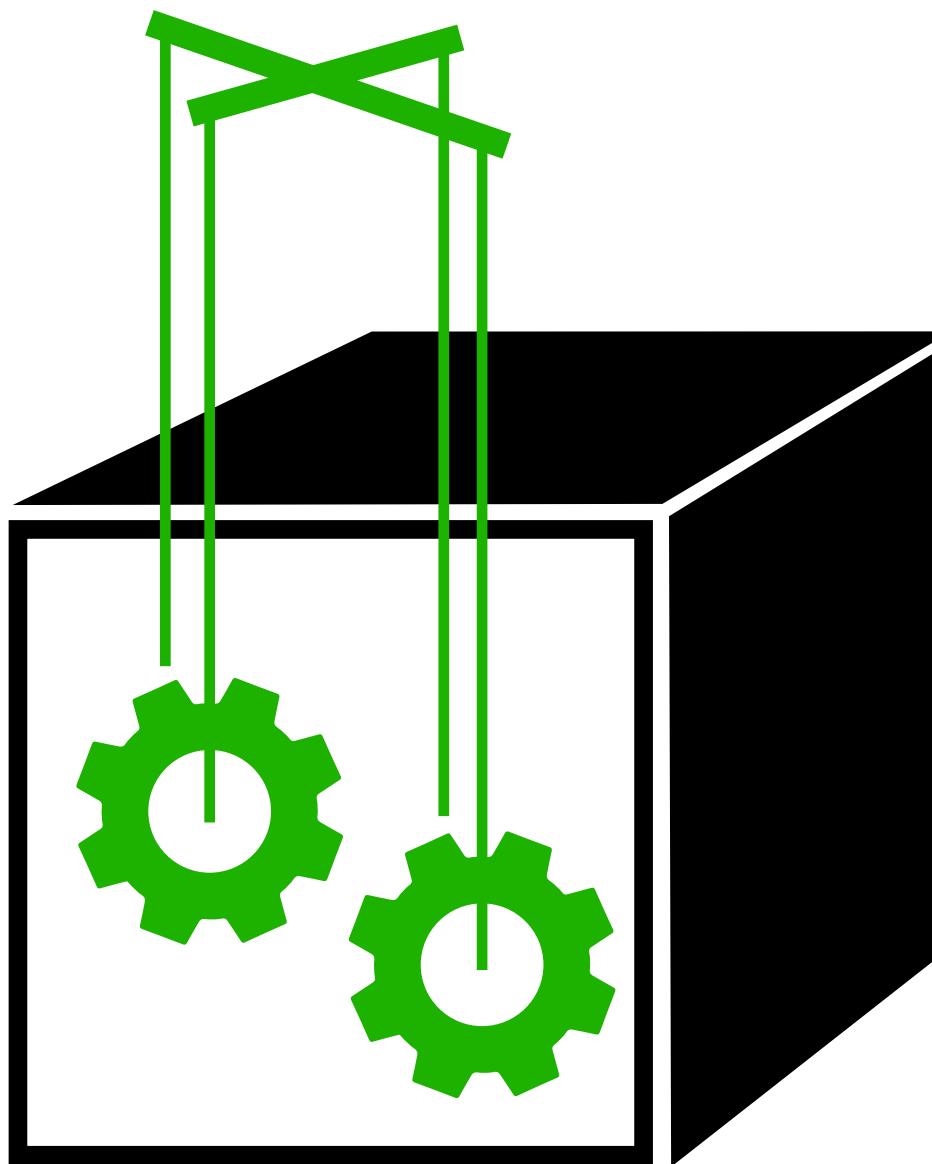
Frontiers of simulation-based inference

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



Active learning:

Iteratively guide simulator to important parameter regions based on past results

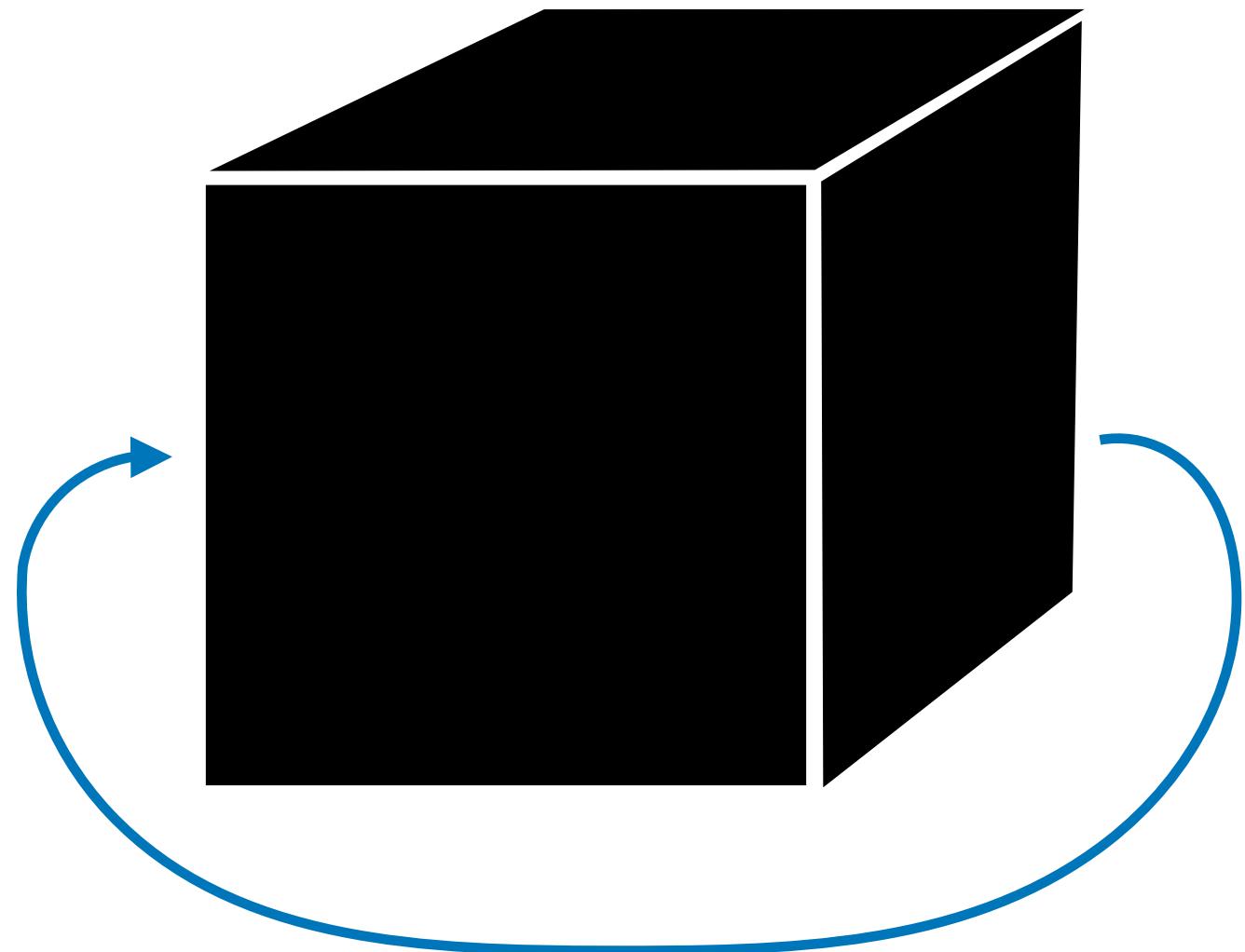


Probabilistic programming:

Explicitly write simulator as probabilistic program, with ability to condition execution trace on observations

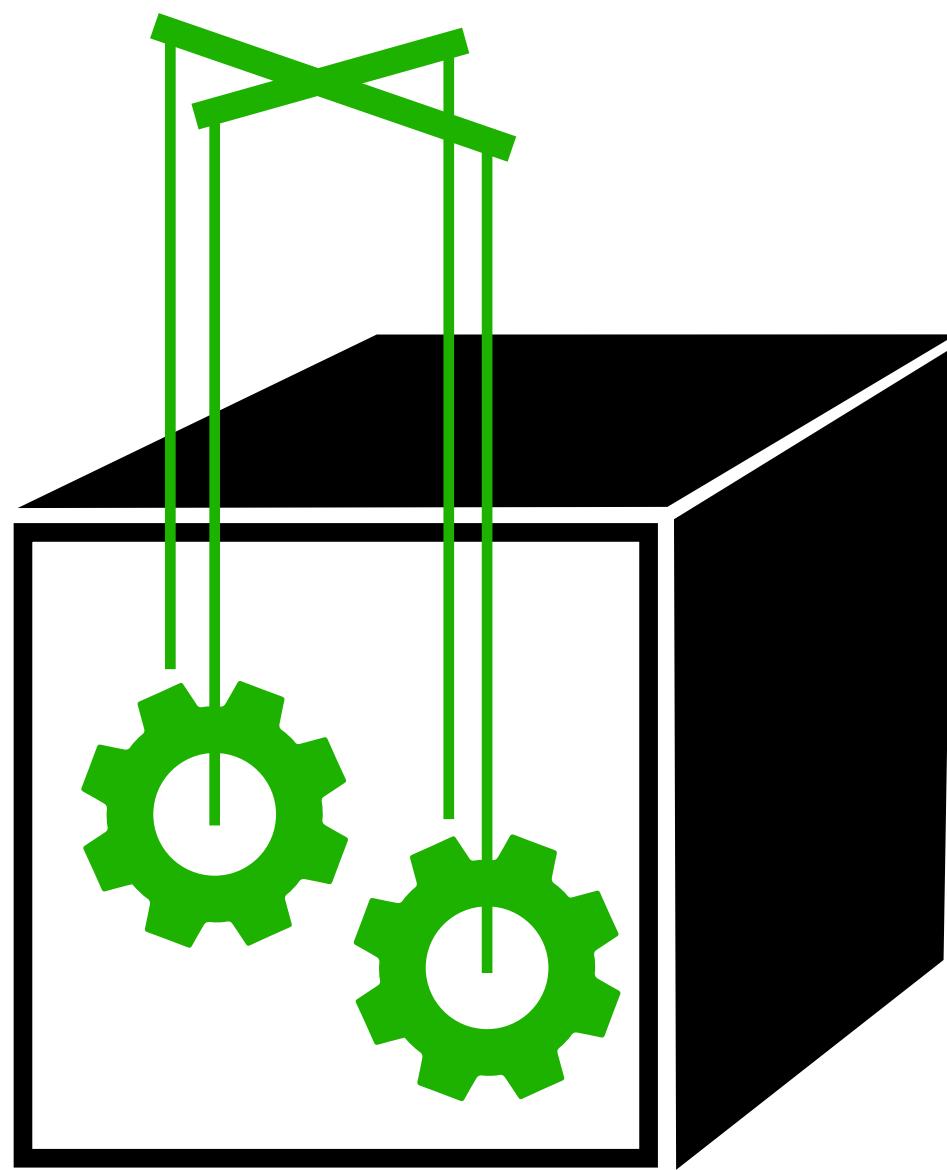
Frontiers of simulation-based inference

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



Active learning:

Iteratively guide simulator to important parameter regions based on past results



Probabilistic programming:

Explicitly write simulator as probabilistic program, with ability to condition execution trace on observations



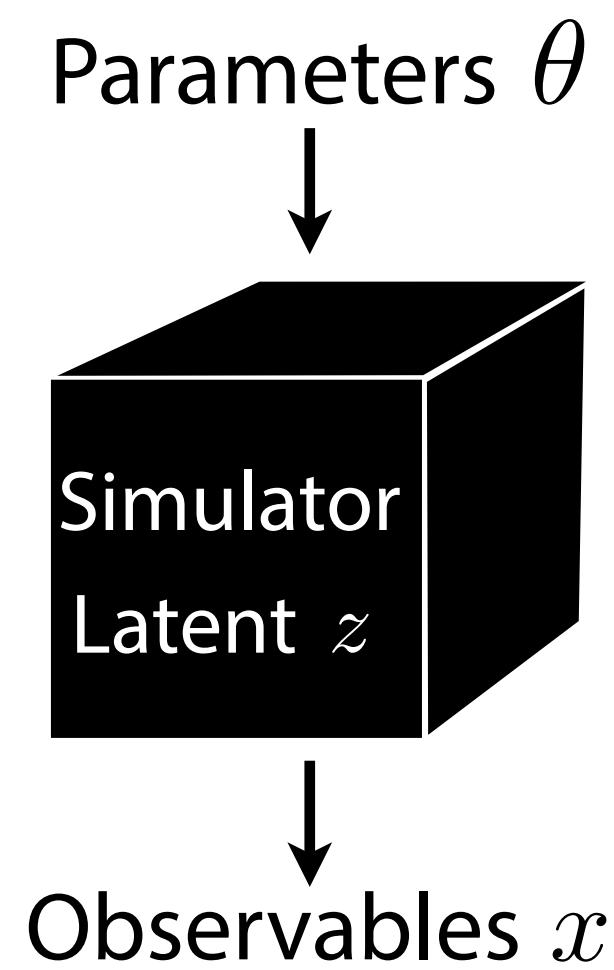
Mining gold:

Extract and leverage information from the simulator that characterizes the latent process

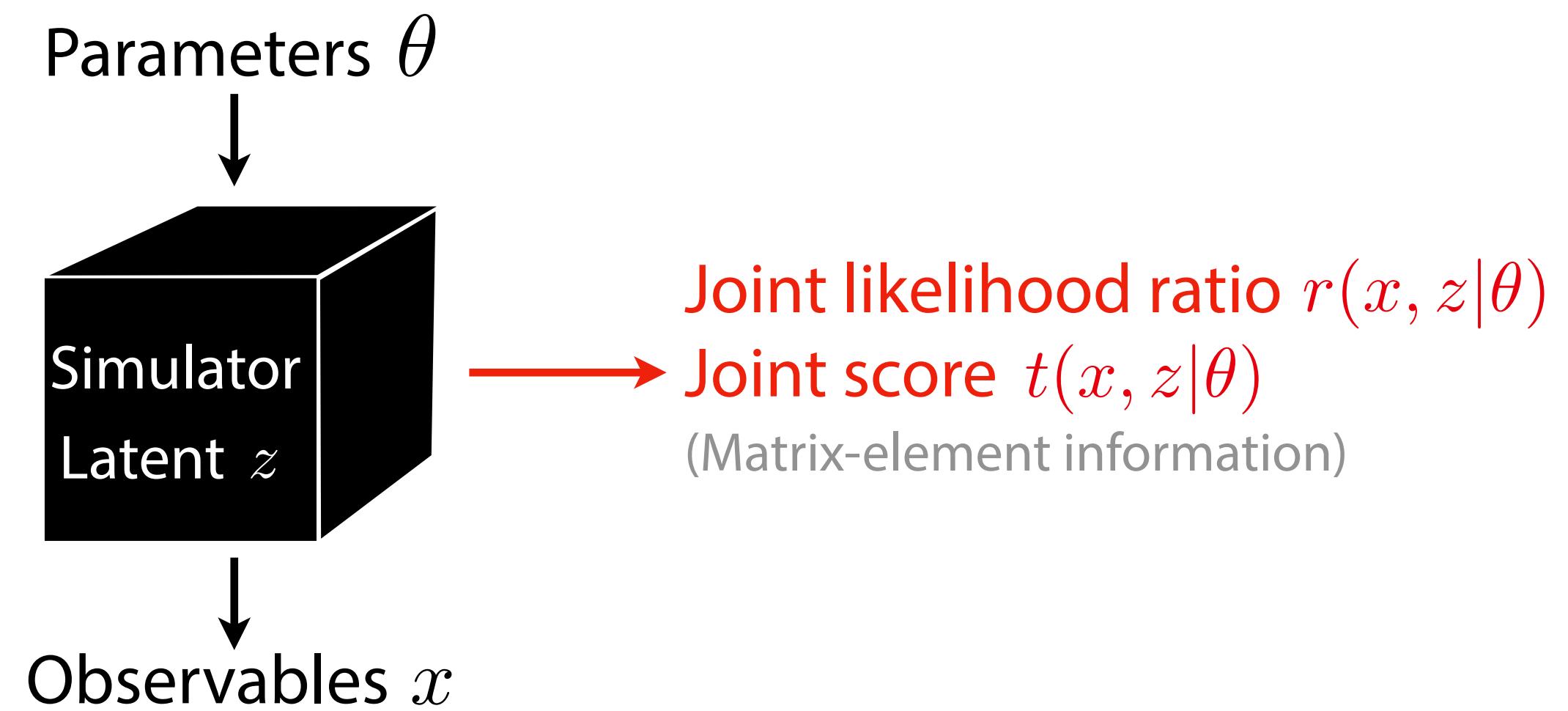
“Mining gold”:
Physics insights can make these inference methods
more efficient.

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

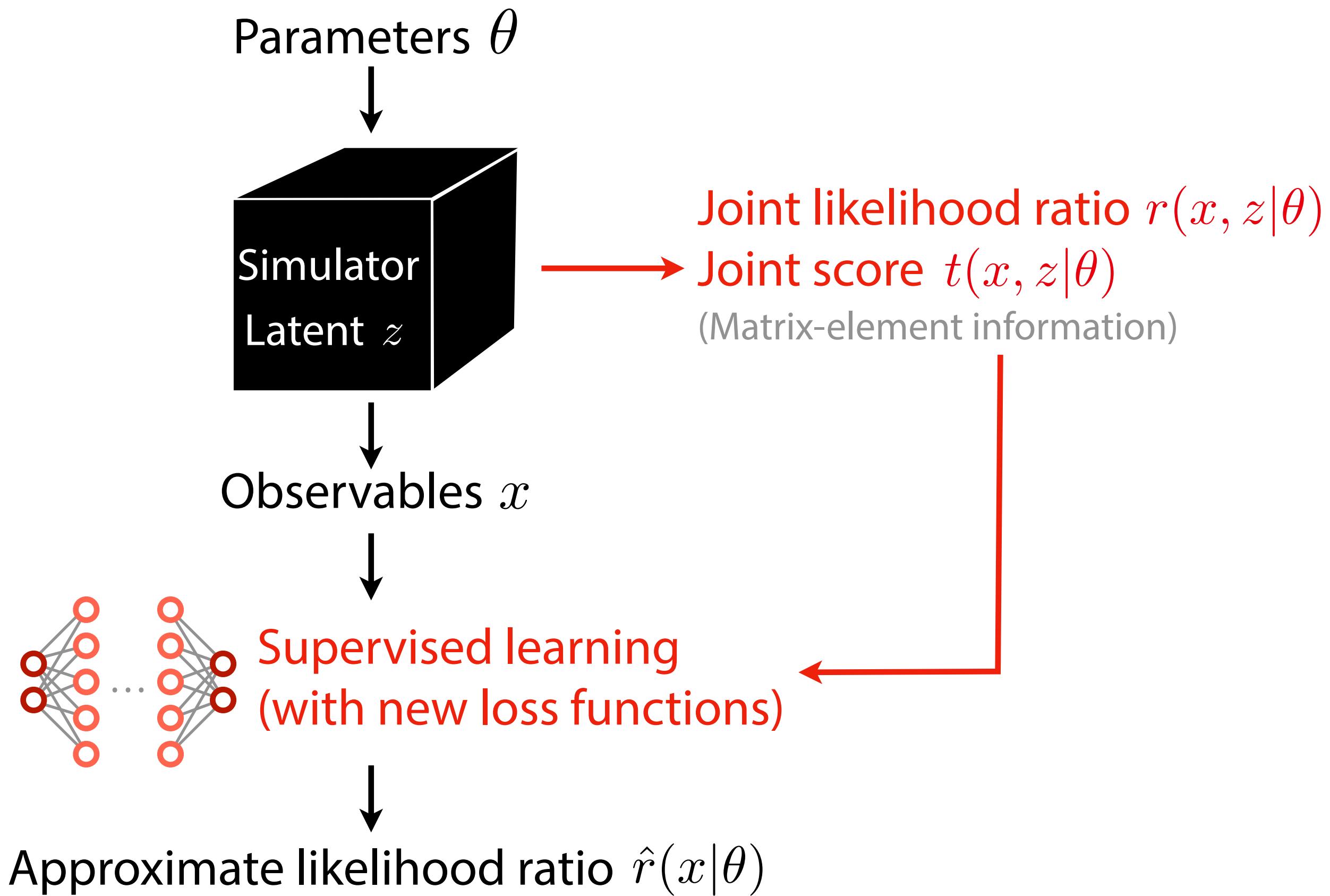
Mining gold



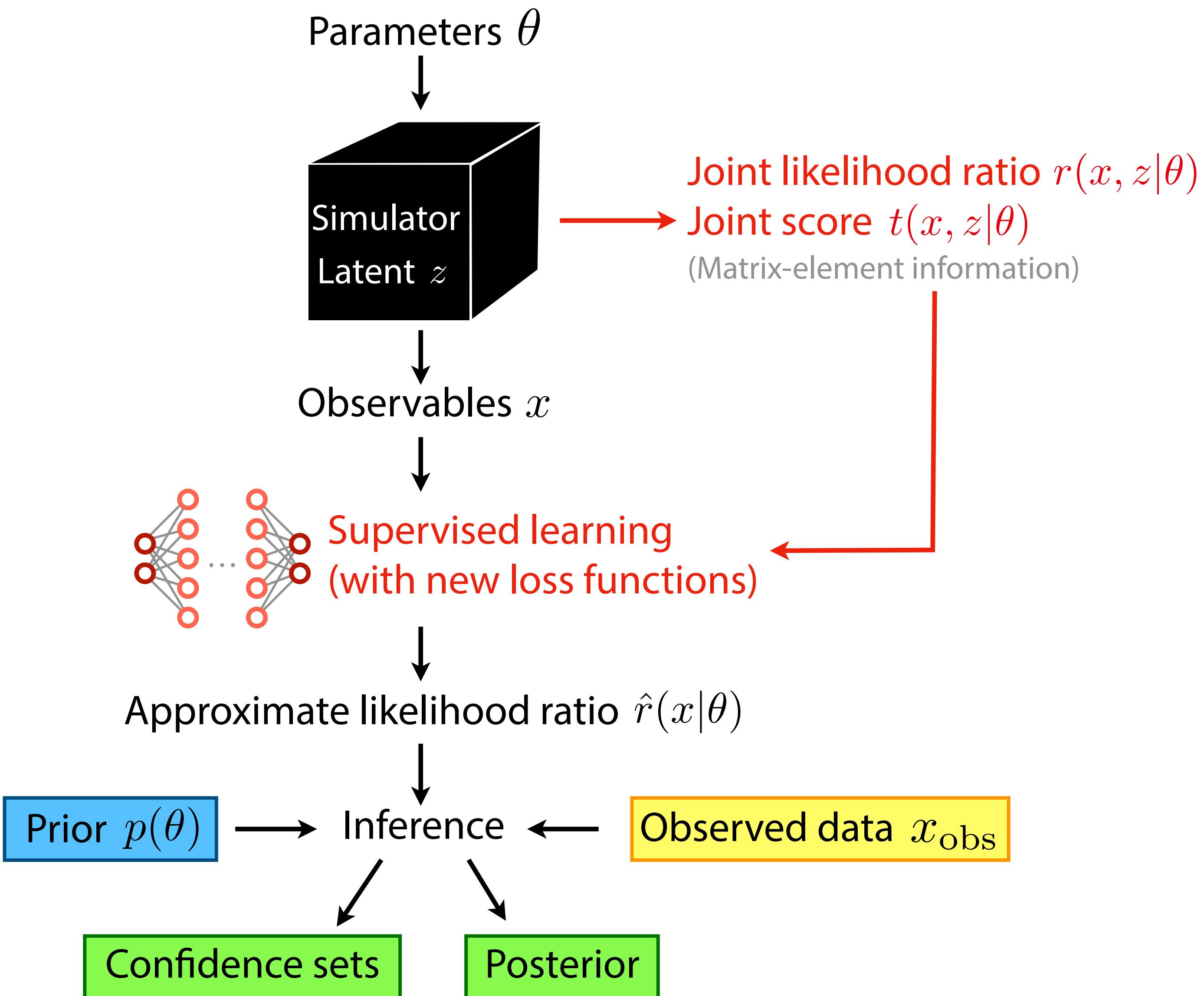
Mining gold



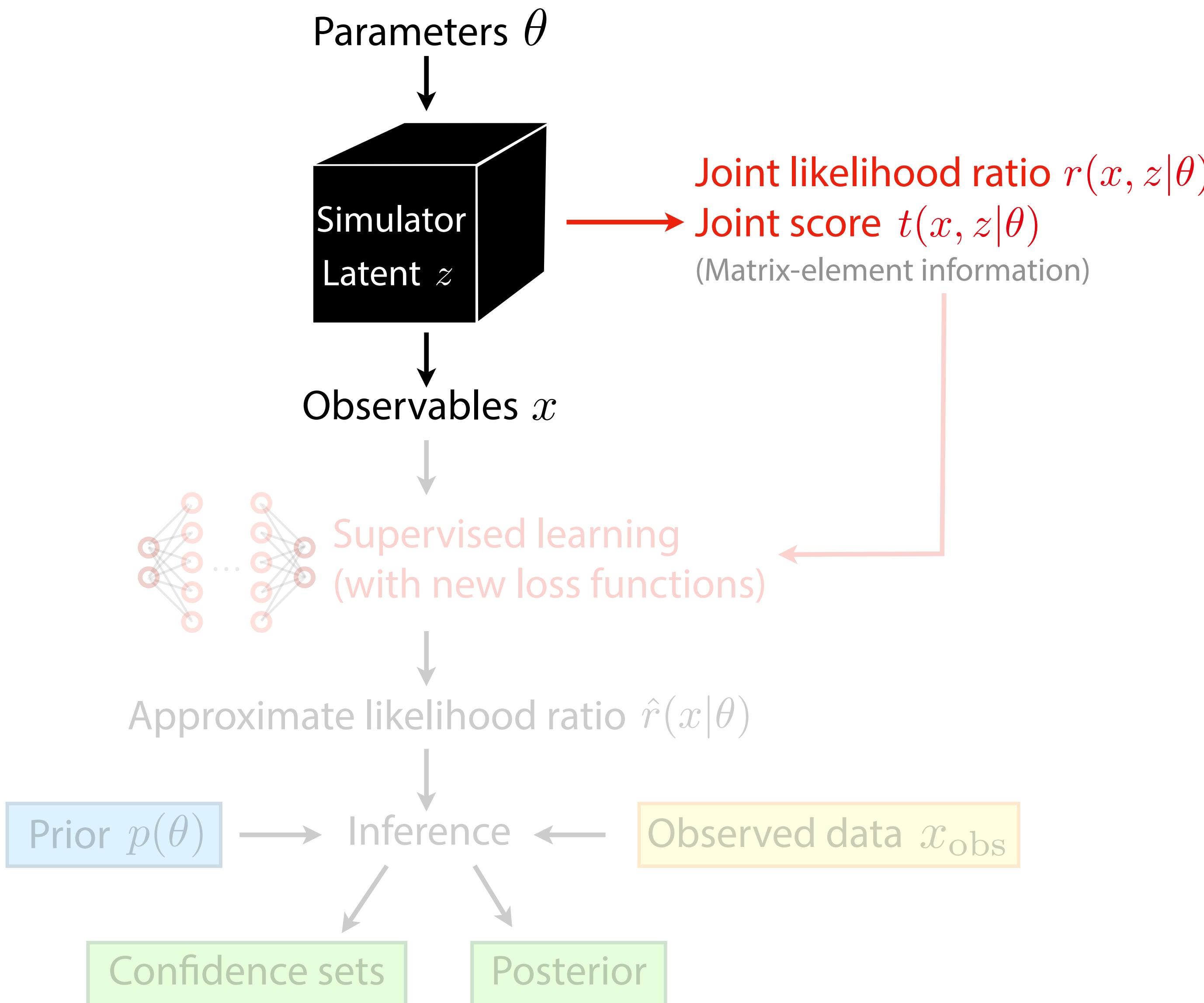
Mining gold



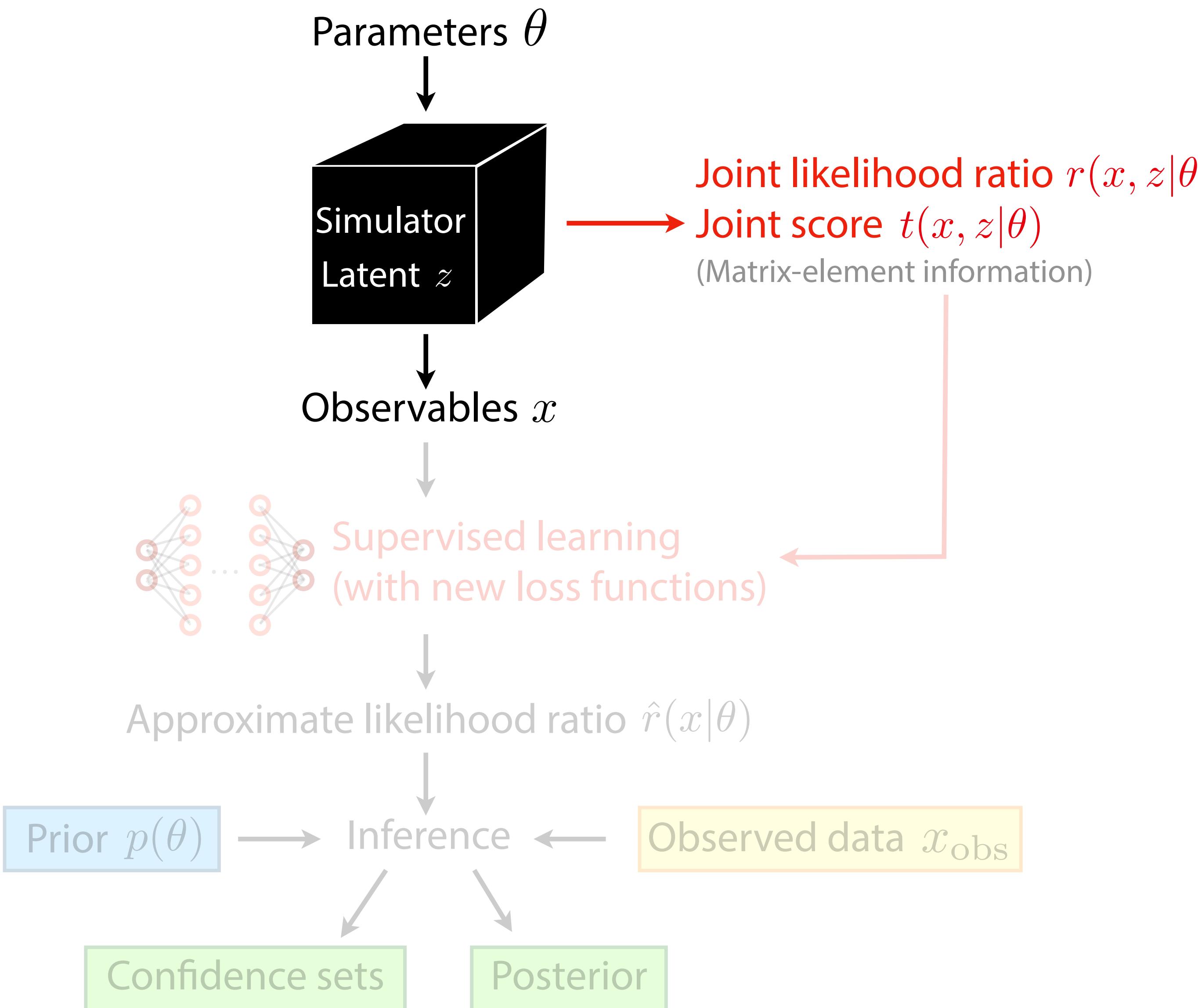
Mining gold



Step 1: Extracting more information from simulations

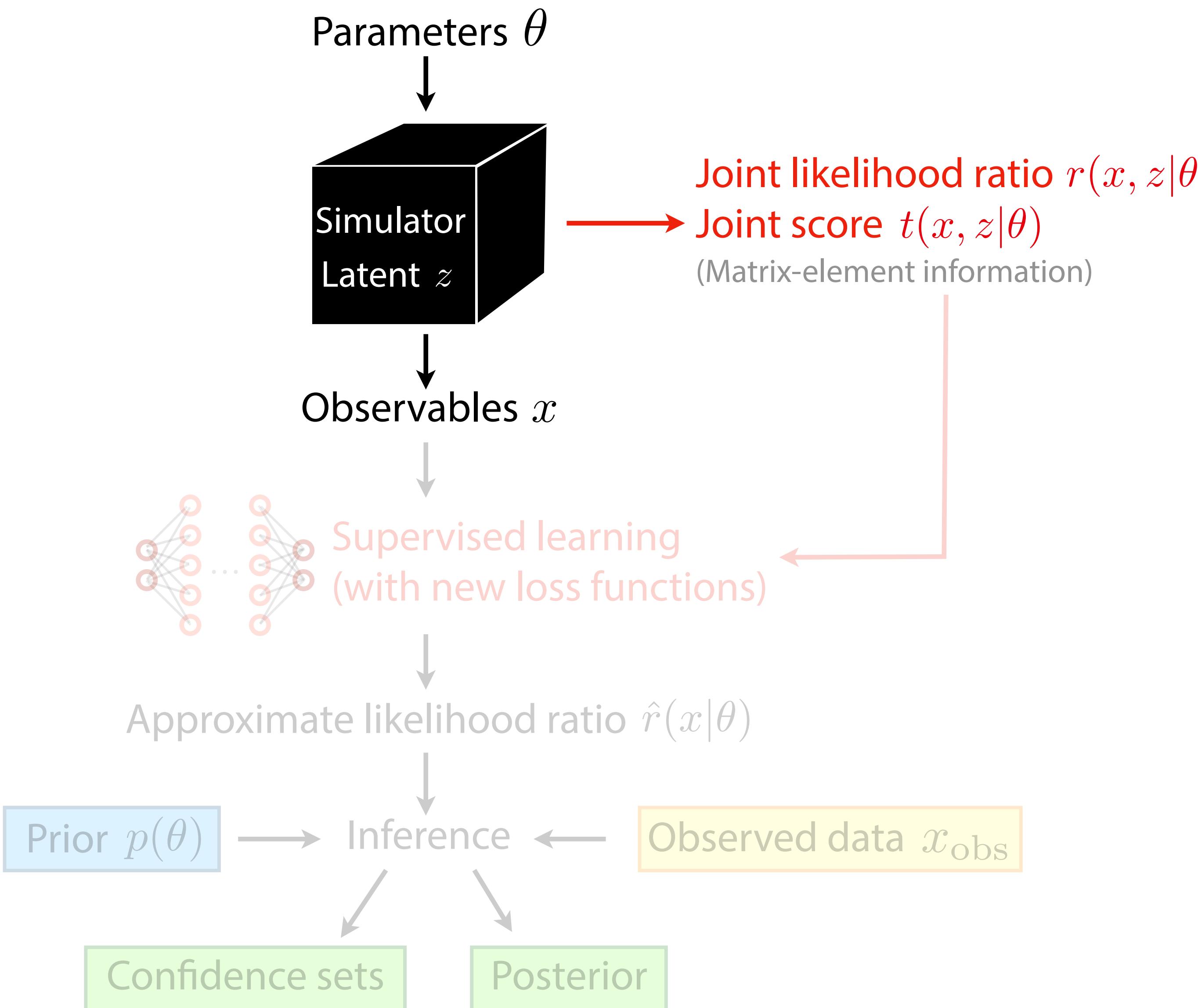


Step 1: Extracting more information from simulations



- For each simulated event, calculate
 - joint likelihood ratio $r(x, z|\theta) = \frac{p(x, z|\theta)}{p_{\text{ref}}(x, z)}$
 - joint score $t(x, z|\theta) = \nabla_{\theta} \log p(x, z|\theta)$
- (How much more or less likely would this simulated event (fixing all latent variables) be when changing the theory parameters?)

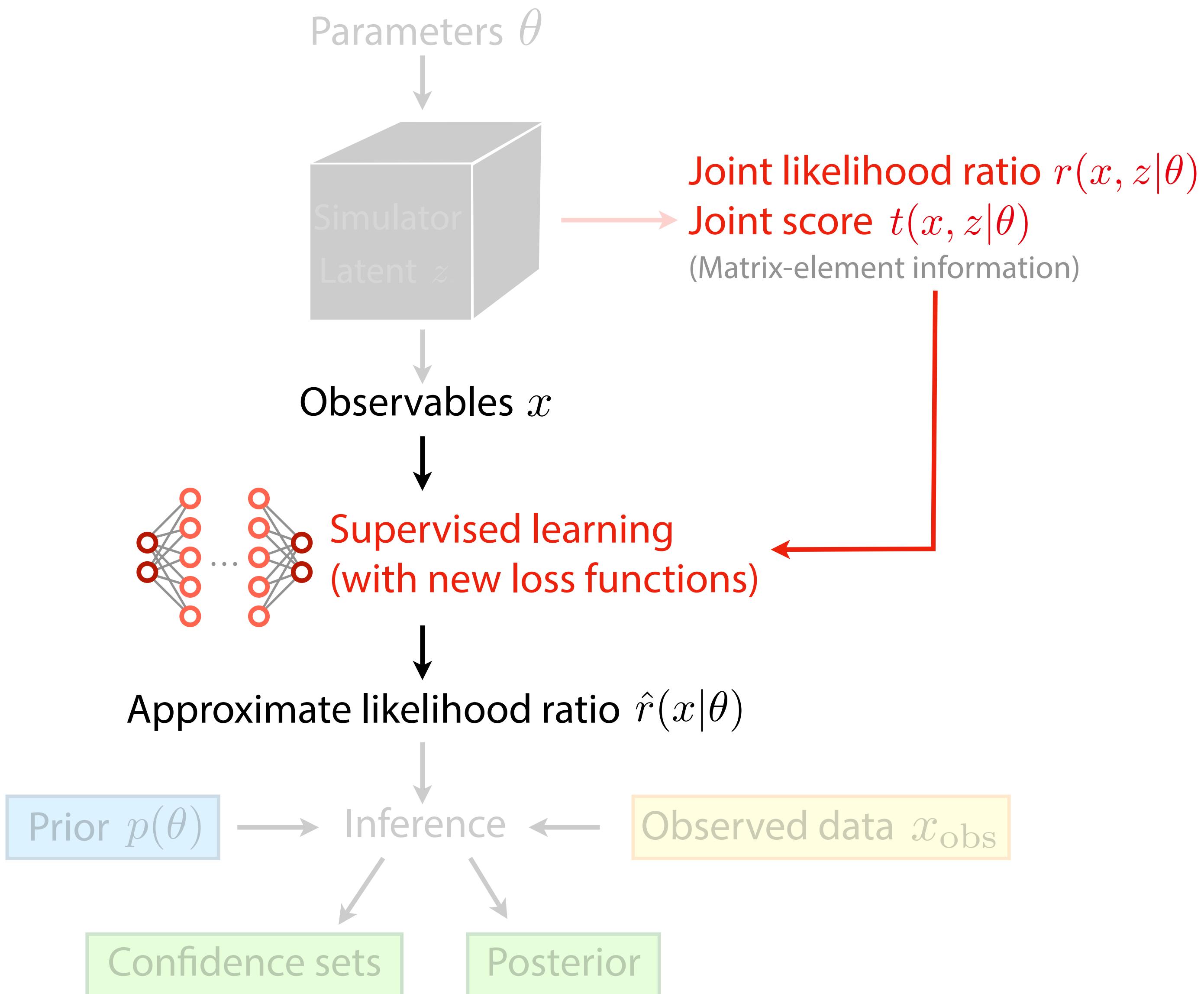
Step 1: Extracting more information from simulations



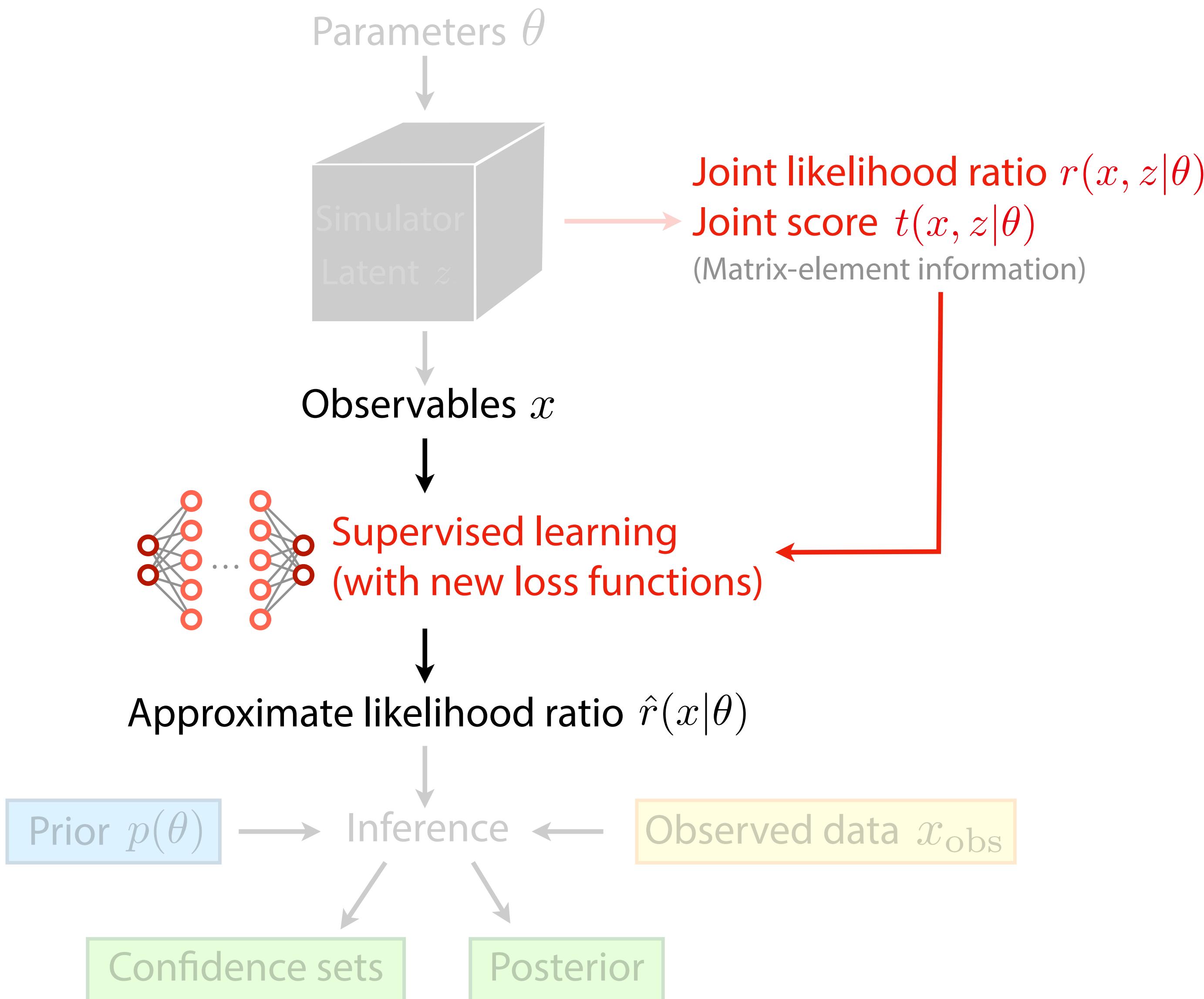
- For each simulated event, calculate
 - joint likelihood ratio $r(x, z|\theta) = \frac{p(x, z|\theta)}{p_{\text{ref}}(x, z)}$
 - joint score $t(x, z|\theta) = \nabla_{\theta} \log p(x, z|\theta)$

(How much more or less likely would this simulated event (fixing all latent variables) be when changing the theory parameters?)
 - Shower + detector simulation are often independent of theory parameters θ , and
- $$r(x, z|\theta) \sim \frac{|\mathcal{M}|^2(\theta)}{|\mathcal{M}|_{\text{ref}}^2}$$
- $$t(x, z|\theta) \sim \frac{\nabla_{\theta} |\mathcal{M}|^2(\theta)}{|\mathcal{M}|^2(\theta)}$$
- can be calculated easily

Step 2: Machine learning



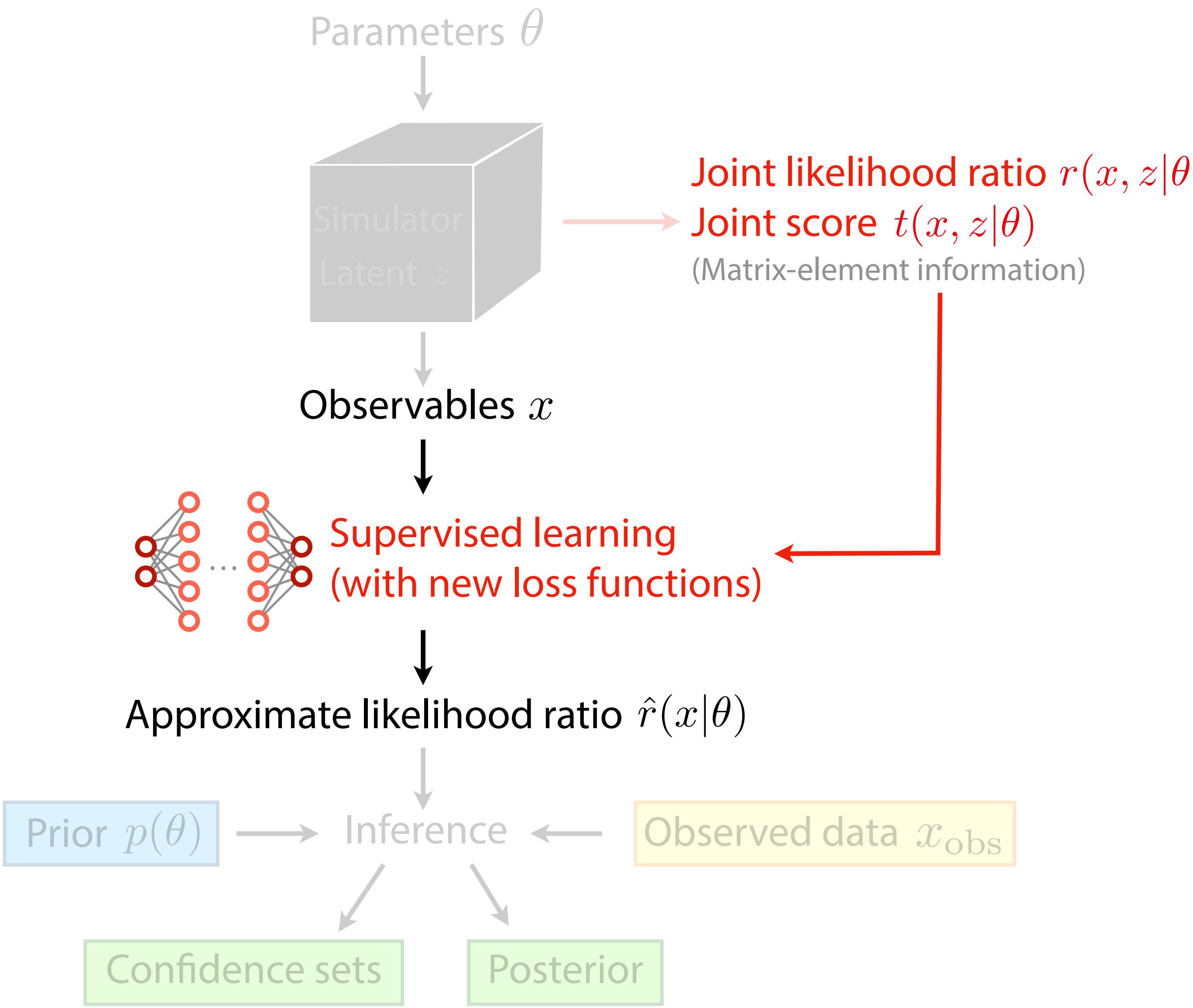
Step 2: Machine learning



- Train a neural network $g(x, \theta)$ on loss functionals like

$$L[g] = \frac{1}{N} \sum_i |g(x_i, \theta_i) - r(x_i, z_i|\theta_i)|^2$$

Step 2: Machine learning



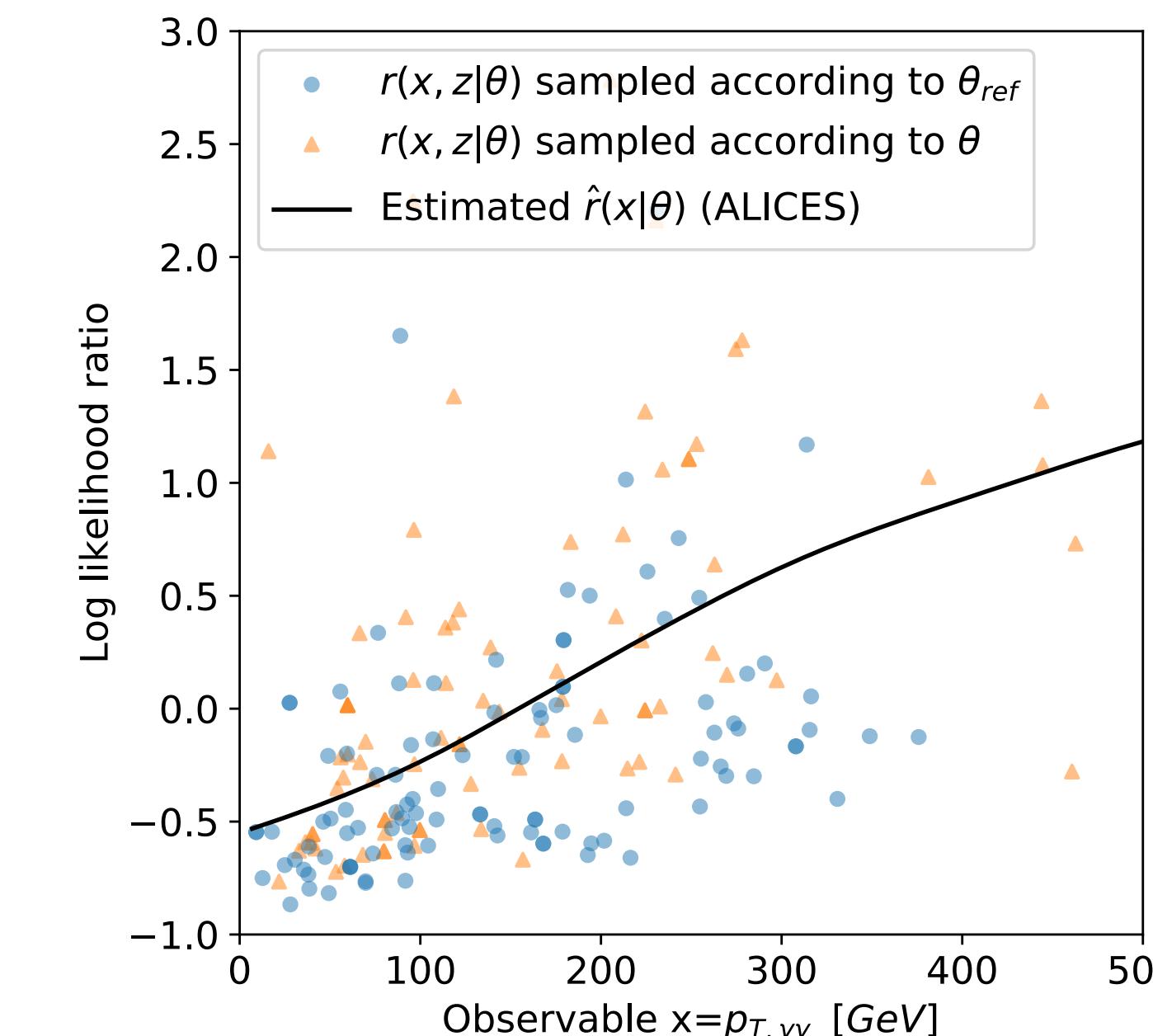
- Train a neural network $g(x, \theta)$ on loss functionals like

$$L[g] = \frac{1}{N} \sum_i |g(x_i, \theta_i) - r(x_i, z_i|\theta_i)|^2$$

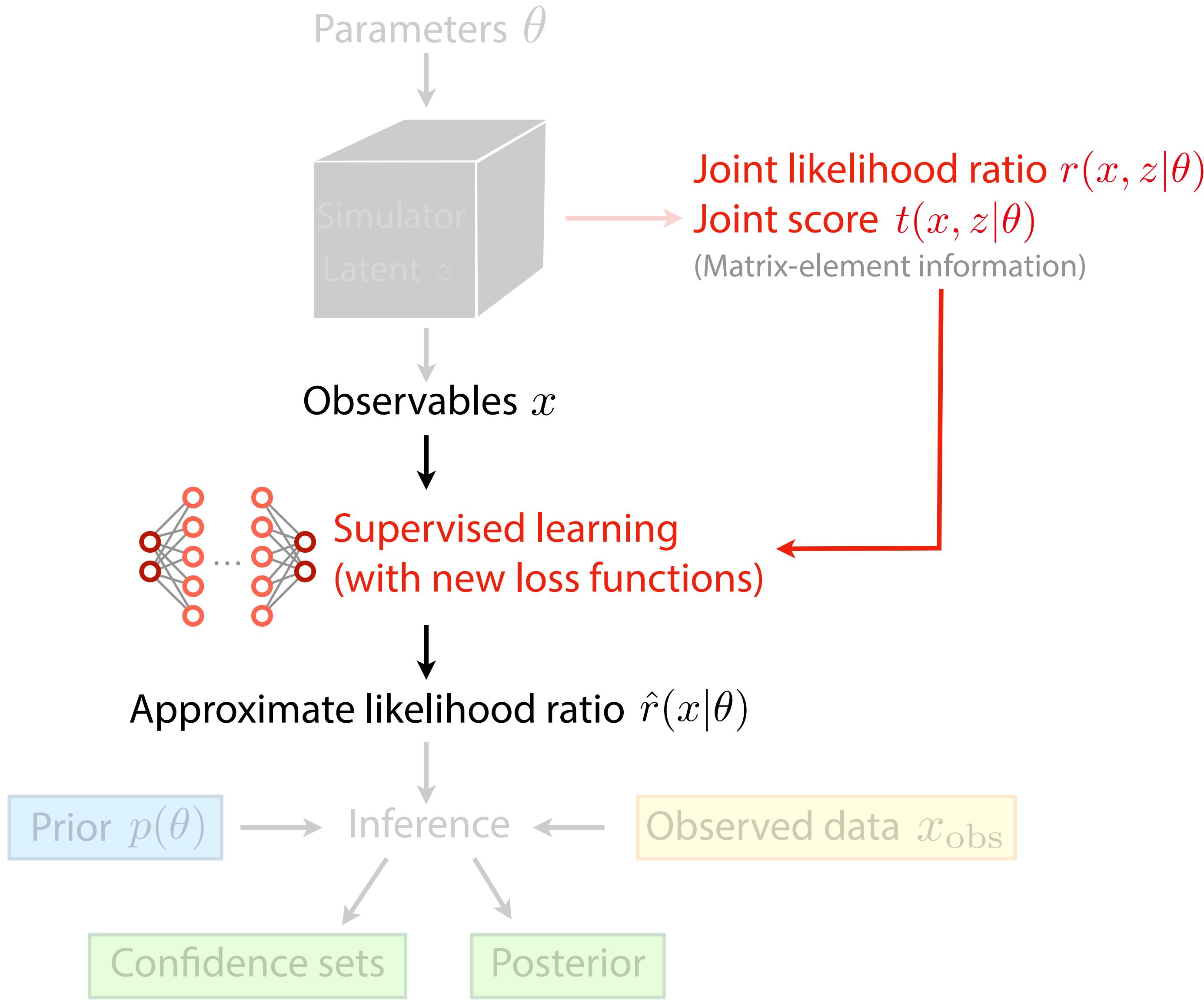
- The network will converge to

$$g(x, \theta) \rightarrow \arg \min L[g] = r(x|\theta) = \frac{p(x|\theta)}{p_{\text{ref}}(x)} !$$

(for sufficient training data, NN capacity, efficient optimization)



Step 2: Machine learning



- Train a neural network $g(x, \theta)$ on loss functionals like

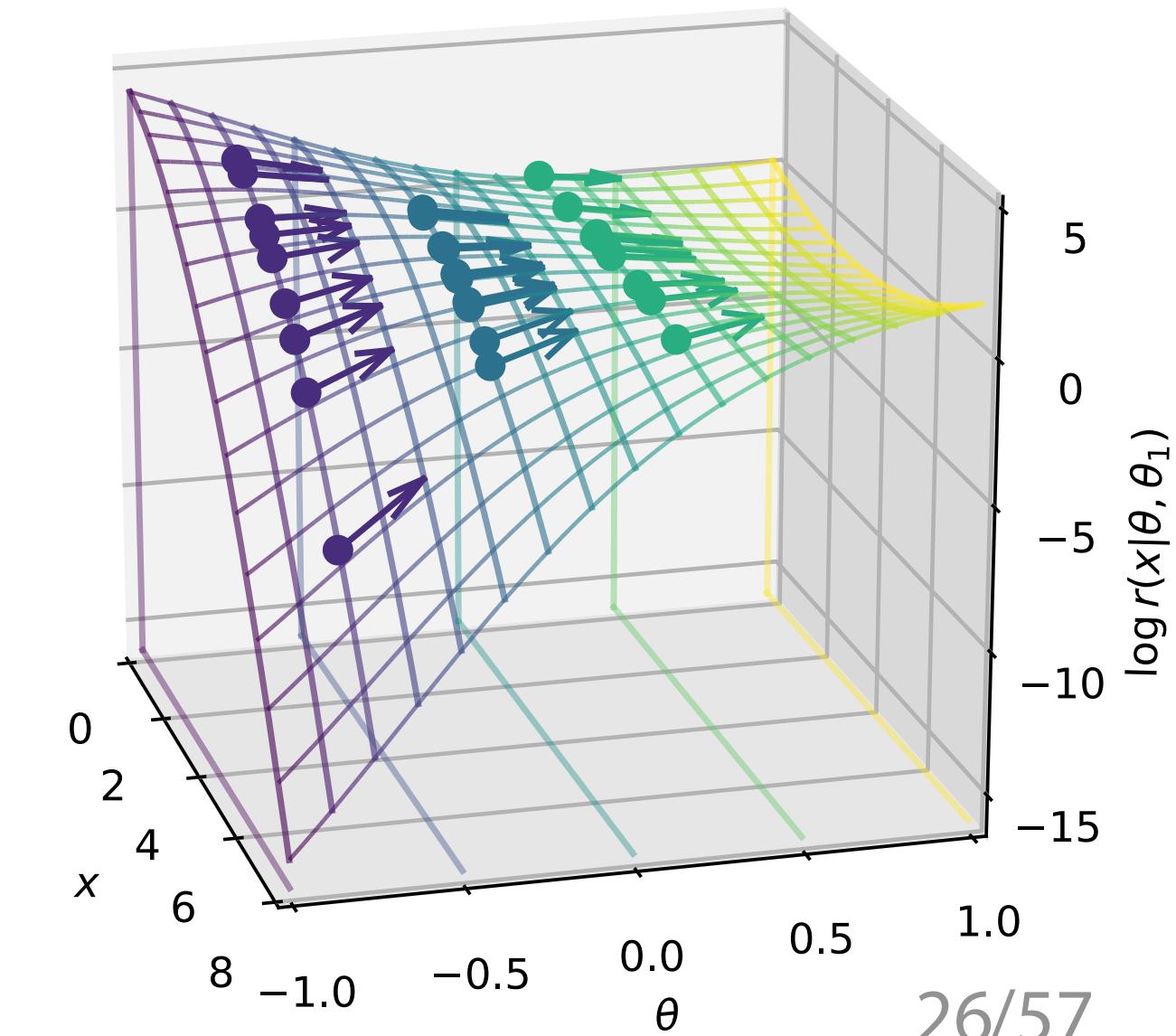
$$L[g] = \frac{1}{N} \sum_i |g(x_i, \theta_i) - r(x_i, z_i|\theta_i)|^2$$

- The network will converge to

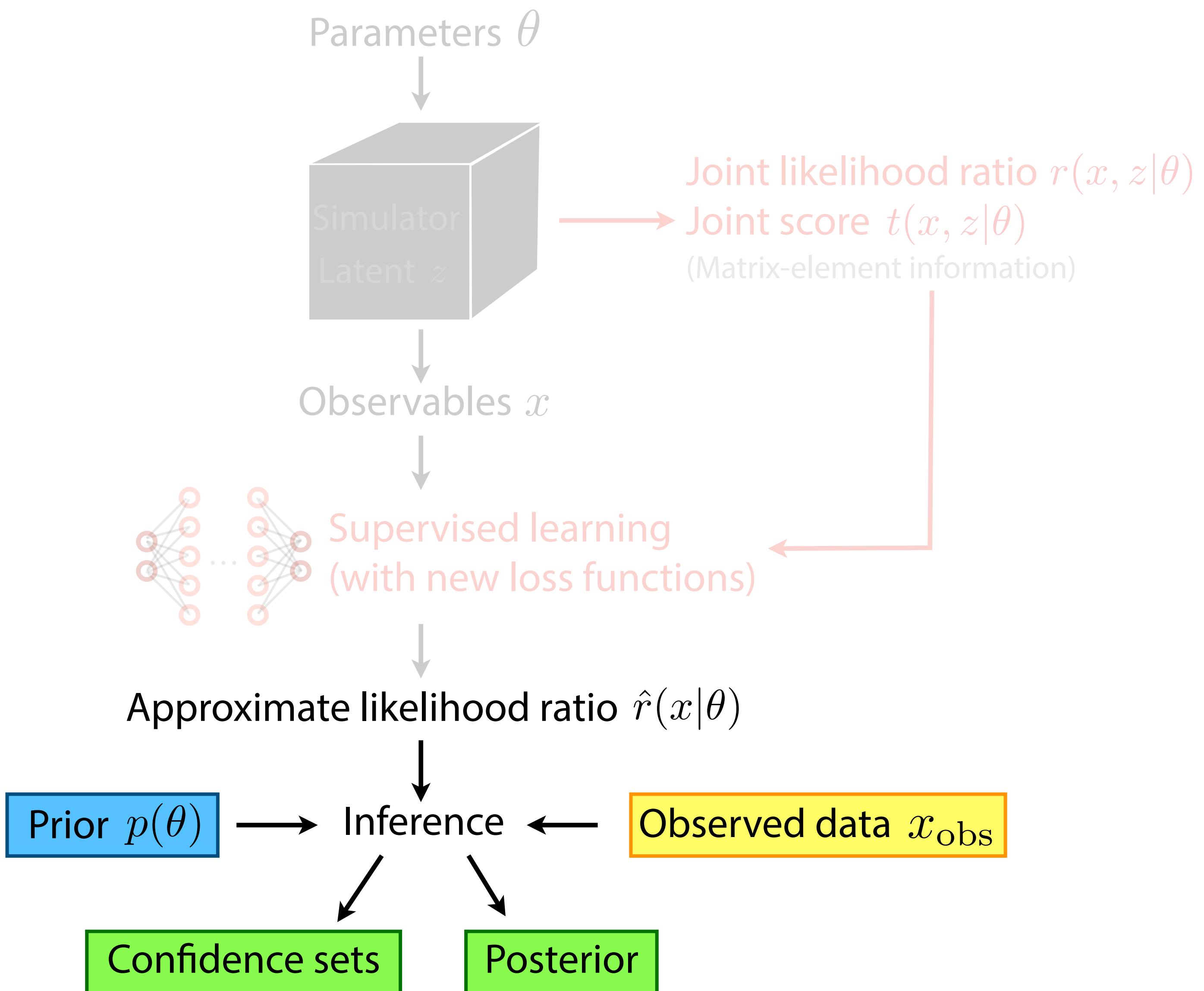
$$g(x, \theta) \rightarrow \arg \min L[g] = r(x|\theta) = \frac{p(x|\theta)}{p_{\text{ref}}(x)} !$$

(for sufficient training data, NN capacity, efficient optimization)

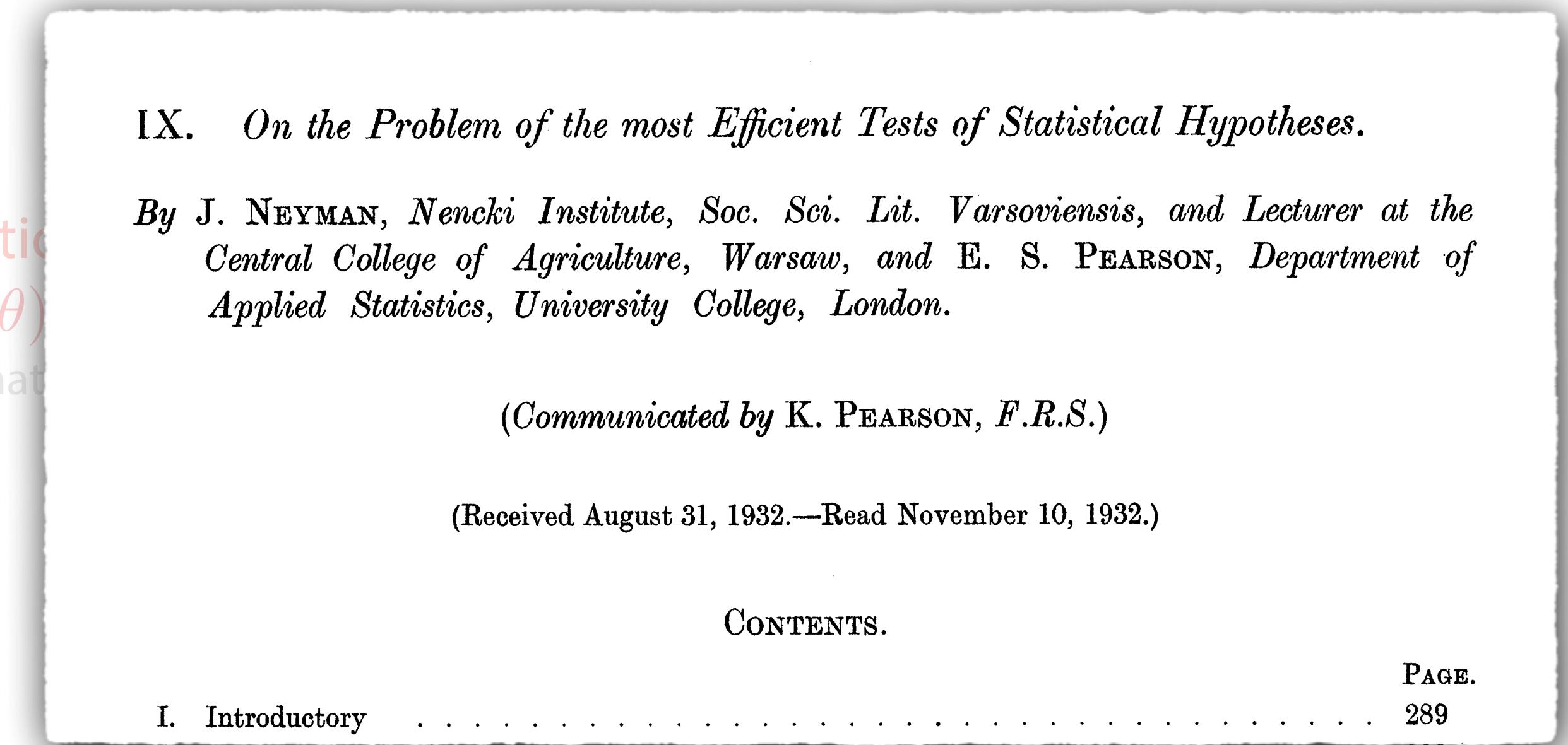
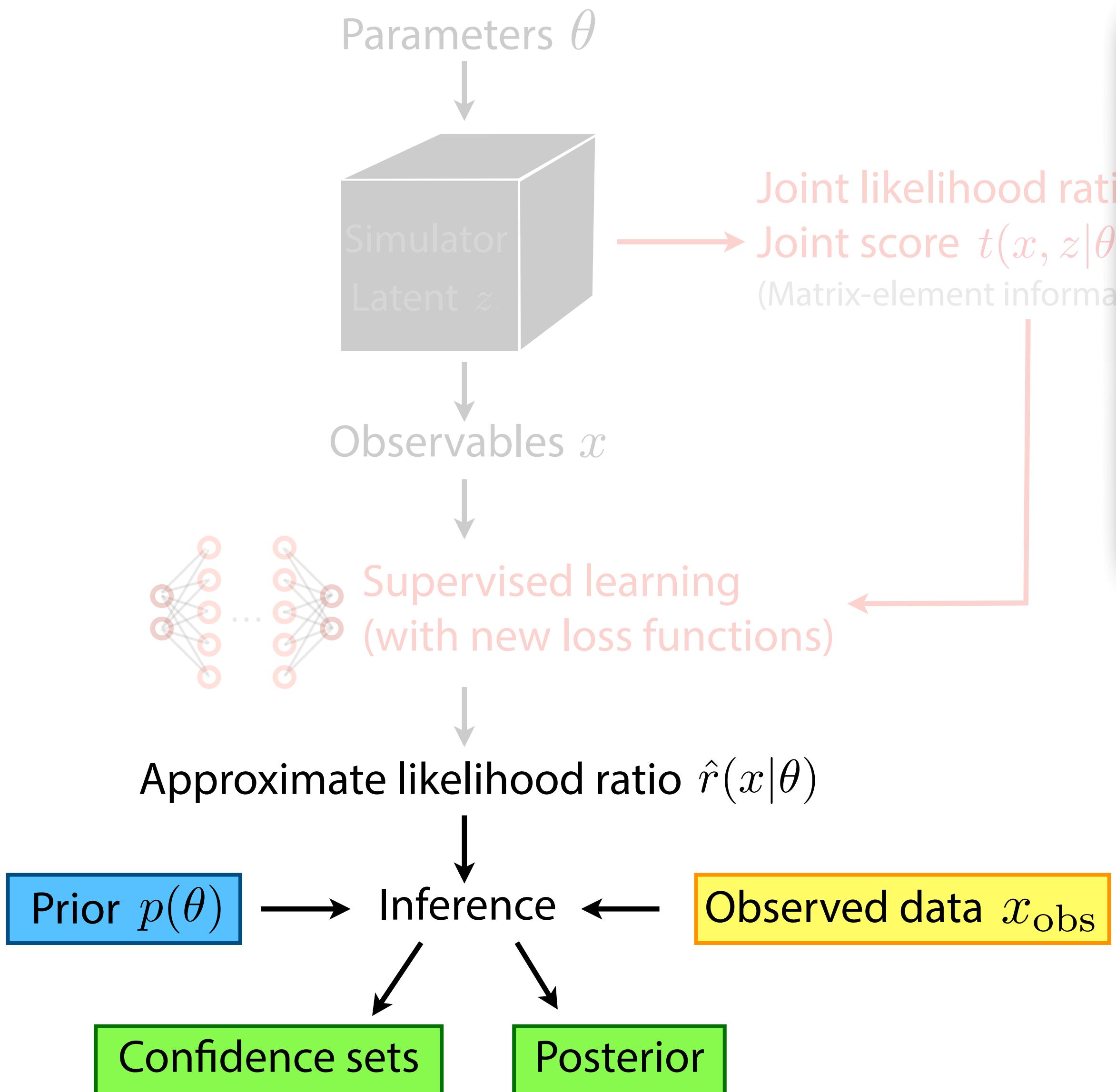
- RASCAL:
Joint score adds gradient information
 \Rightarrow three orthogonal pieces of information



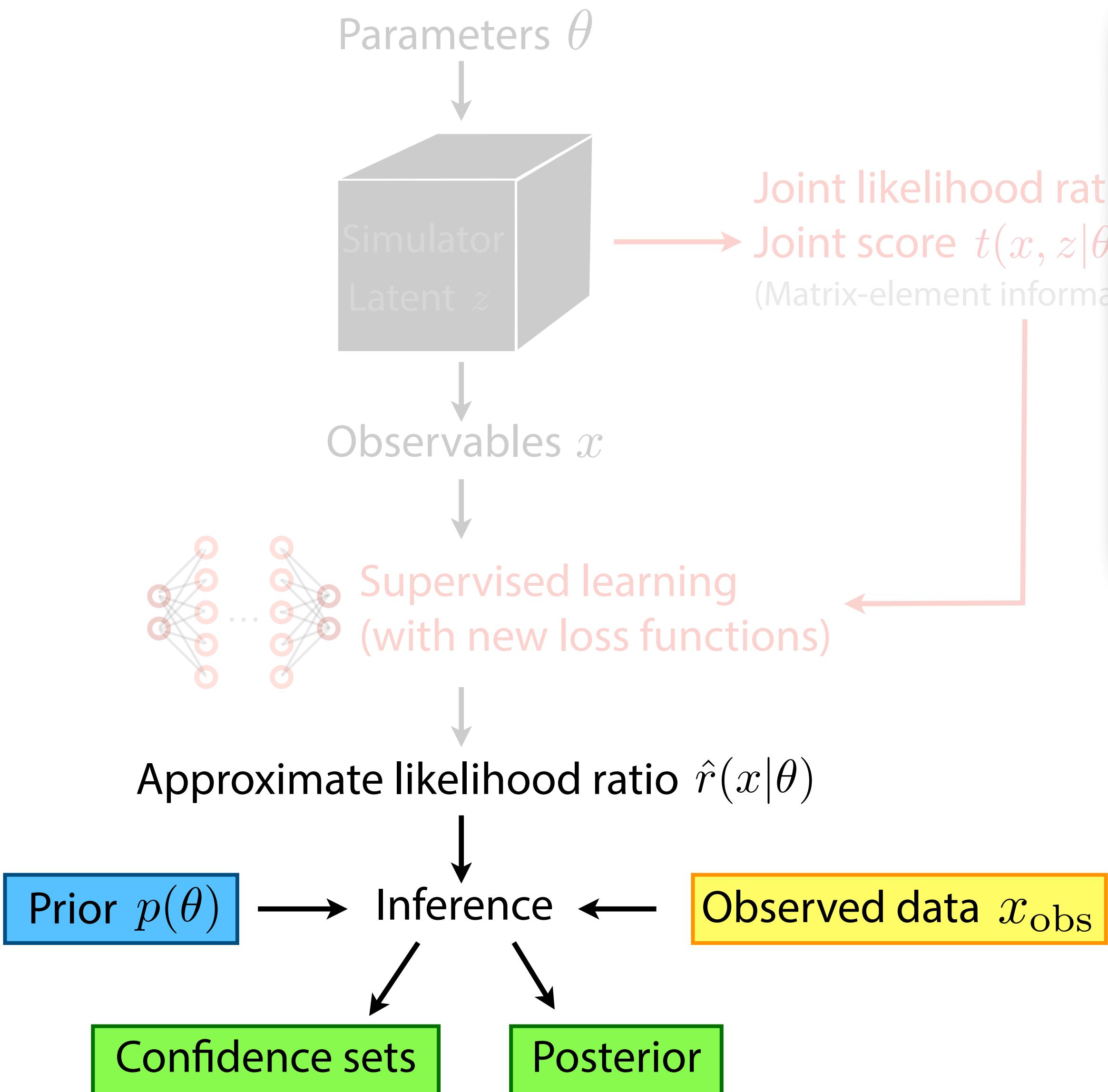
Step 3: Inference



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.)

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 Cowan¹, Kyle Cranmer², Eilam Gross³, Ofer Vitells^{3,a}

¹Physics Department, Royal Holloway, University of London, Egham TW20 0EX, UK

²Physics Department, New York University, New York, NY 10003, USA

³Weizmann Institute of Science, Rehovot 76100, Israel

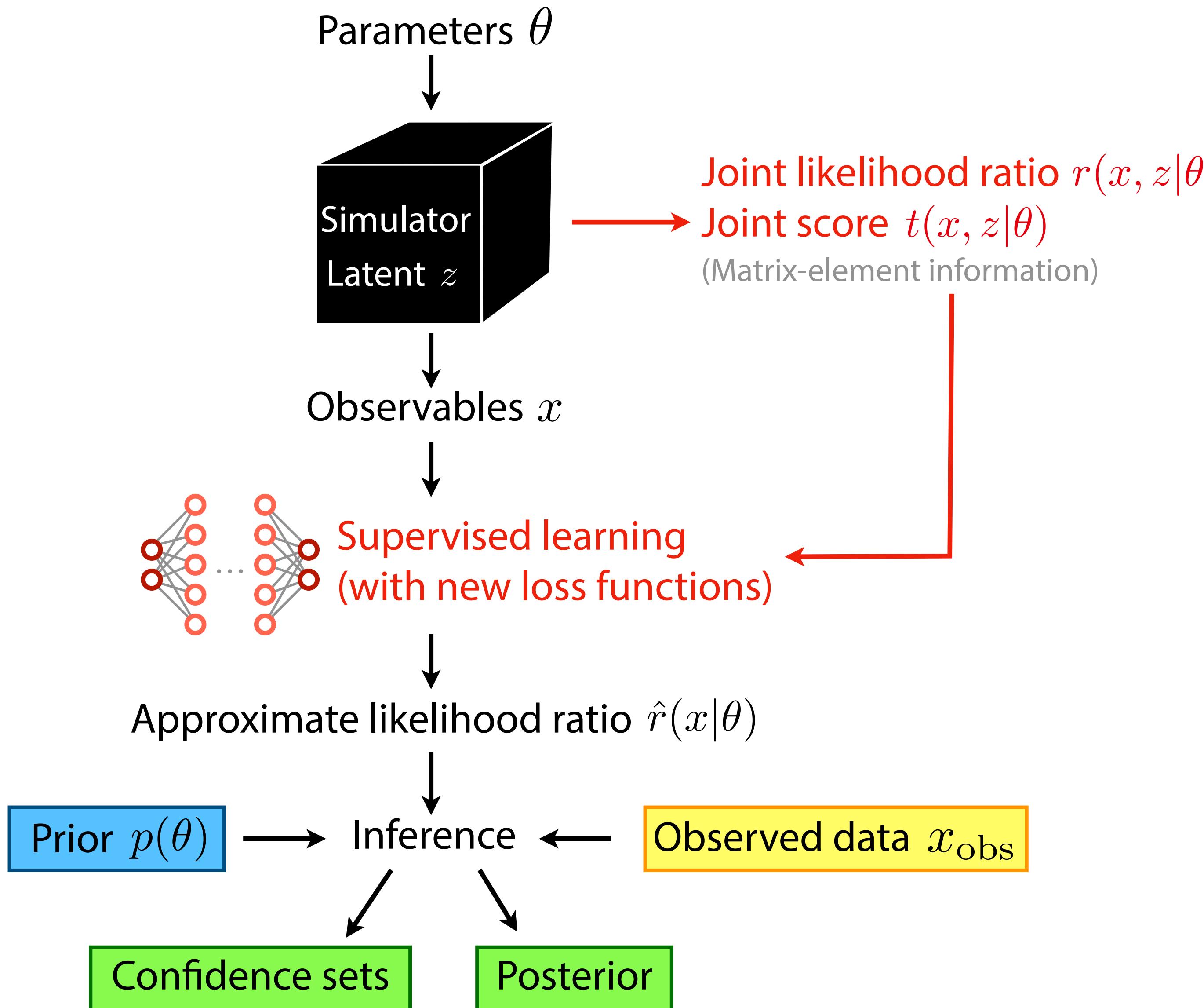
Received: 15 October 2010 / Revised: 6 January 2011 / Published online: 9 February 2011

© The Author(s) 2011. This article is published with open access at Springerlink.com

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

Mining gold



"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)
- Amortized (evaluation in μs)

We can also machine learn optimal observables.

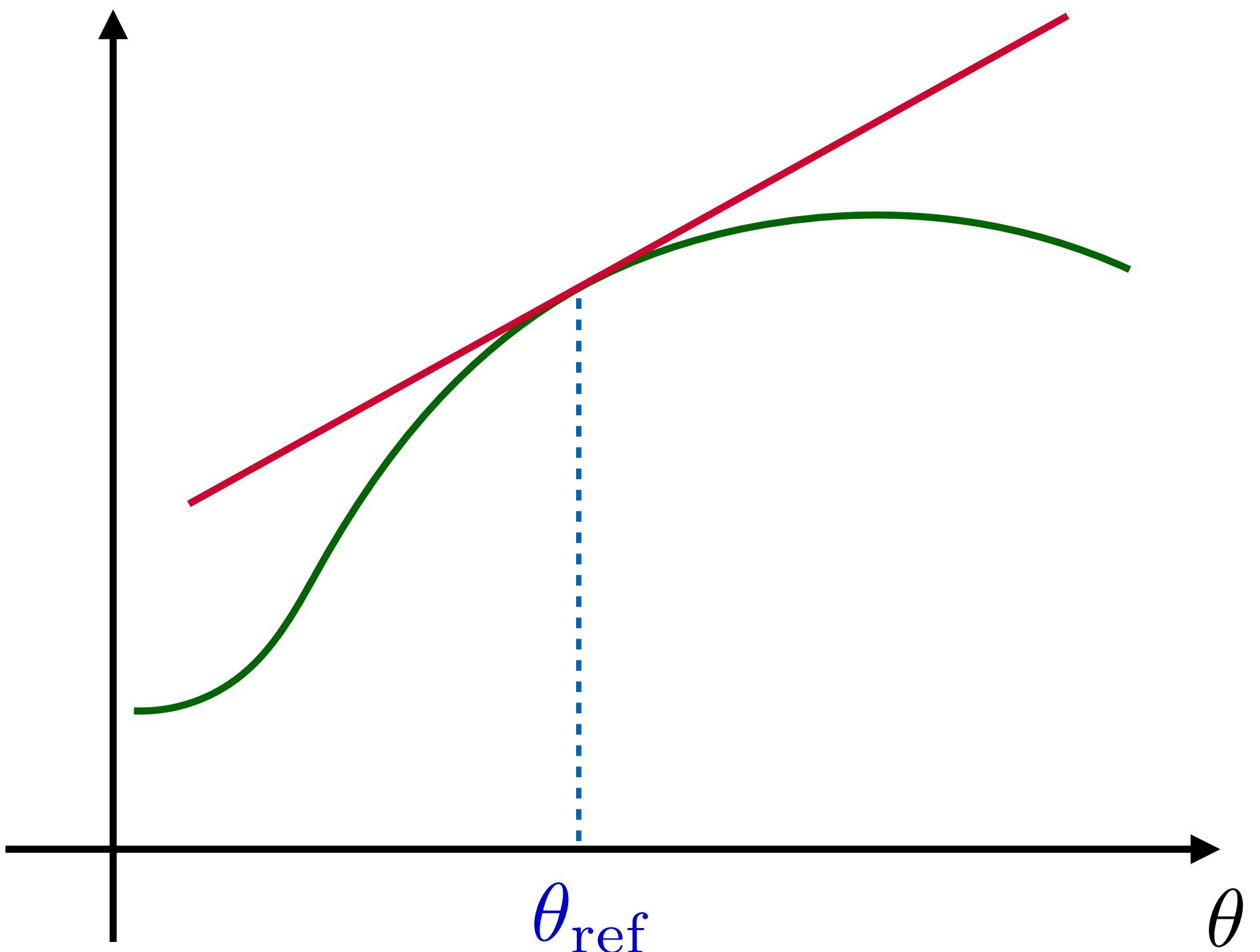
[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013, 1805.00020, 1805.12244]

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 θ_{ref} :

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



The local model

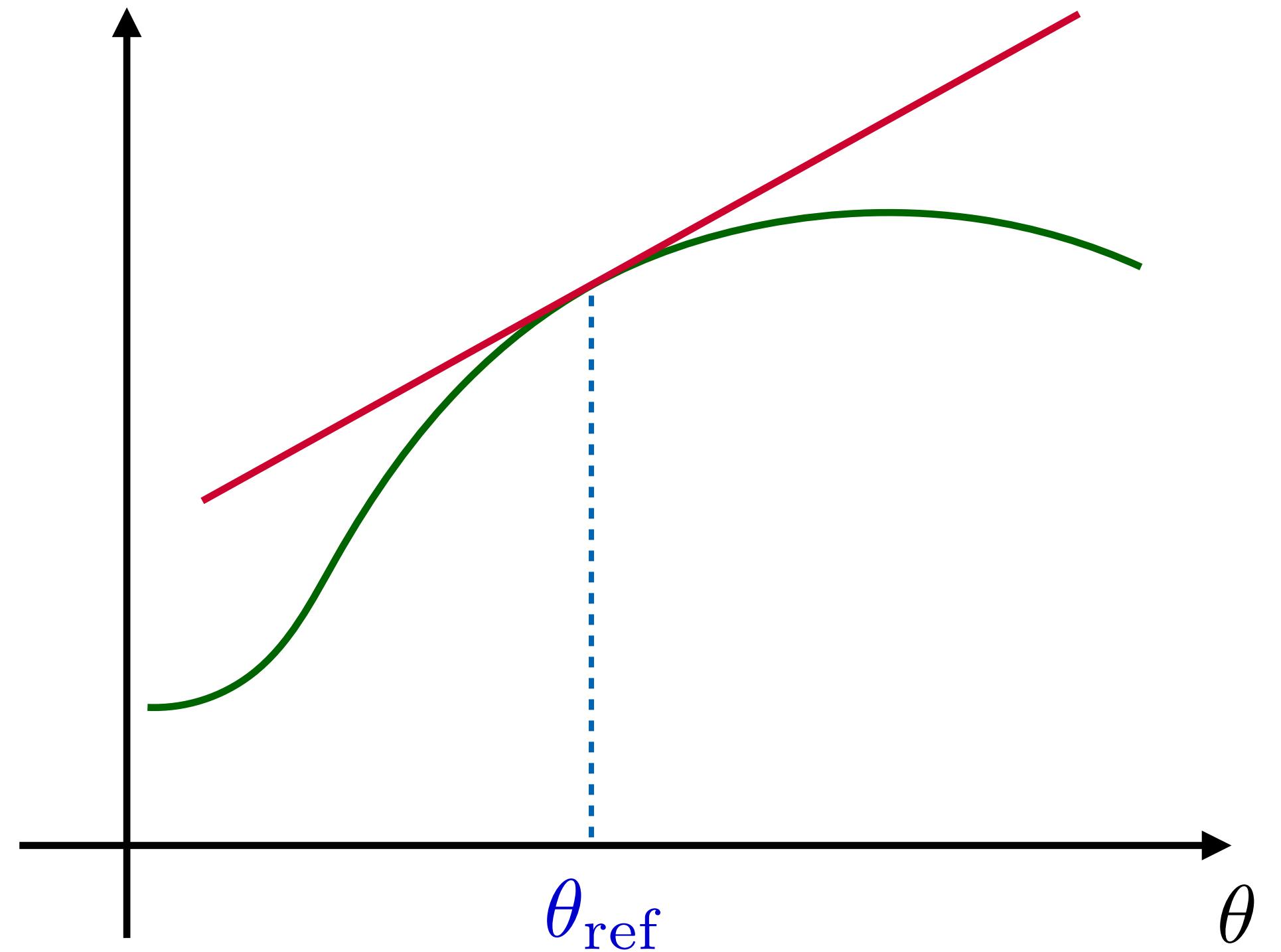
[see also J. Alsing, B. Wandelt 1712.00012; J. Alsing, B. Wandelt, S. Freeney 1801.01497;
P. de Castro, T. Dorigo 1806.04743; J. Alsing, B. Wandelt 1903.01473]

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

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

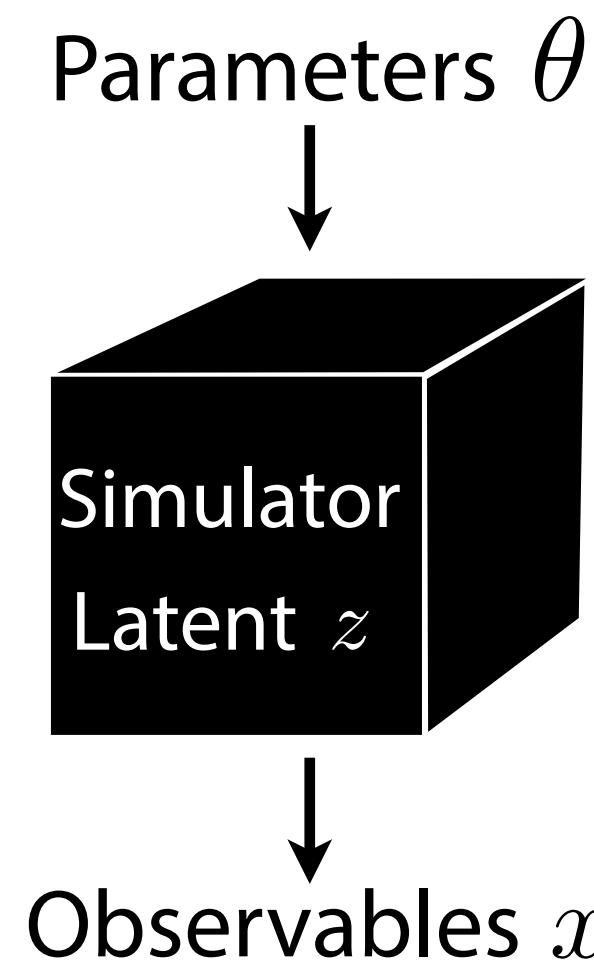
In the neighborhood of θ_{ref} (e.g. close to the SM):

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

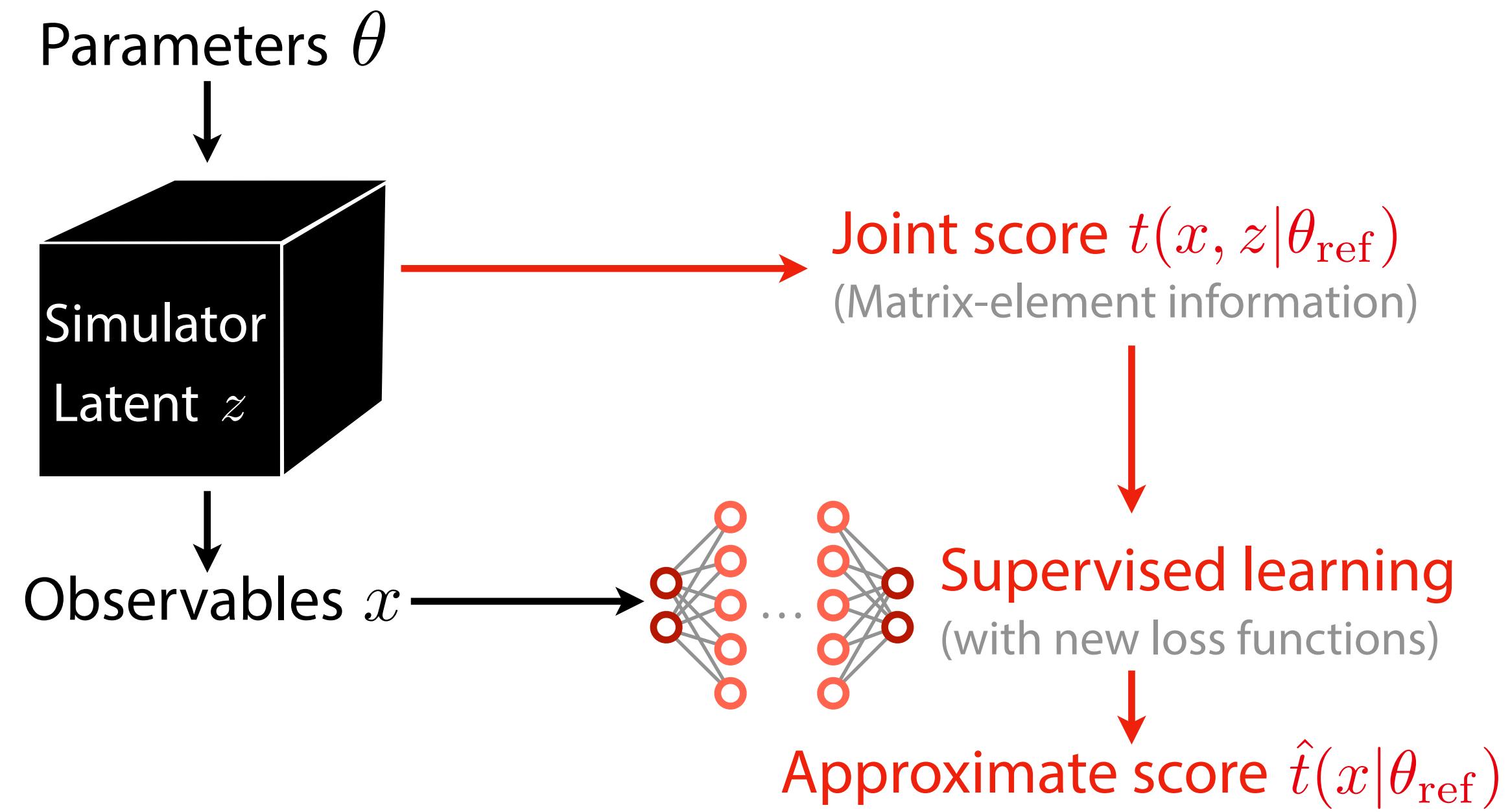


The score itself is intractable. But we can use the same trick as for the likelihood ratio!

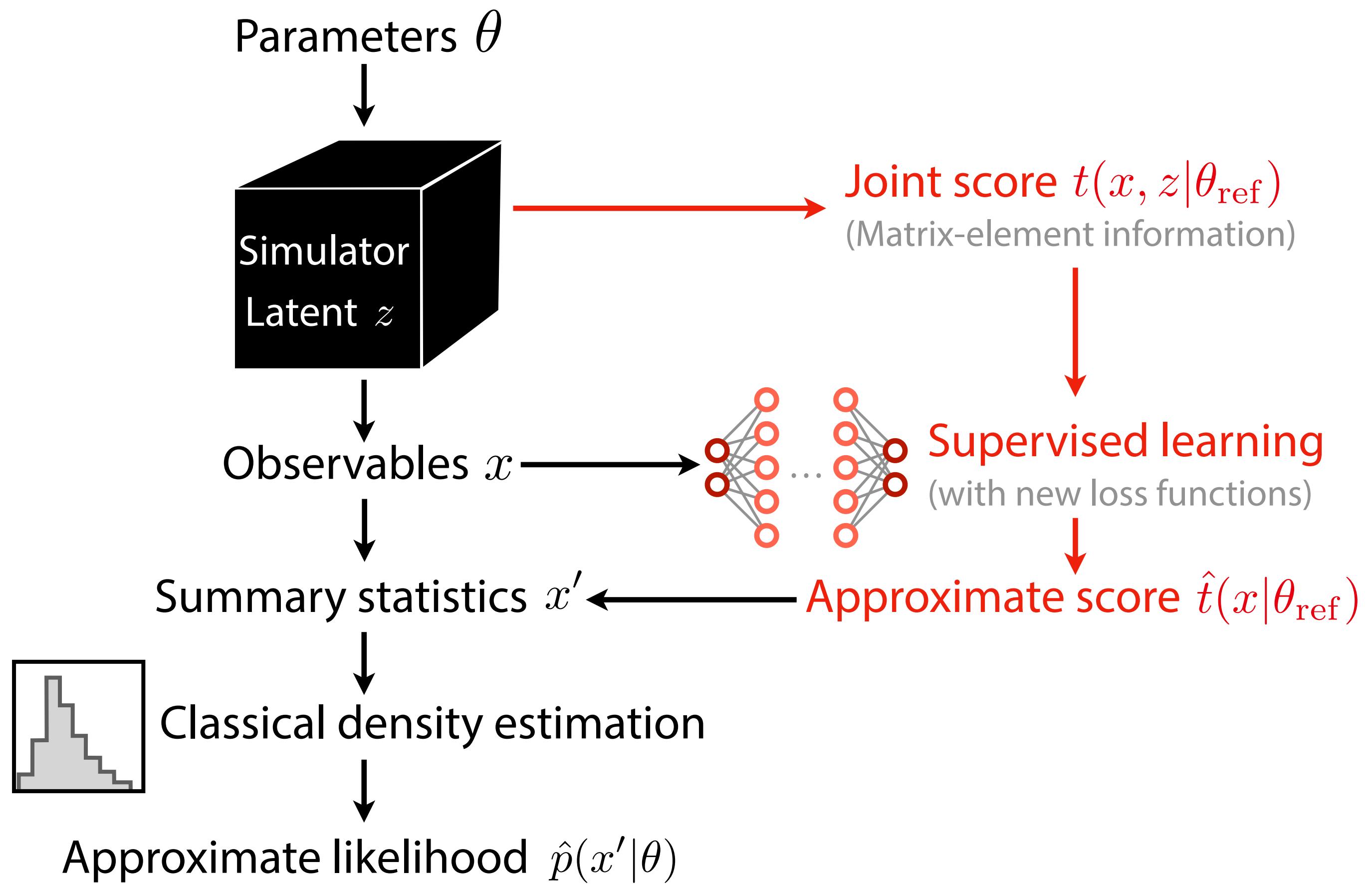
Neural optimal observables (SALLY)



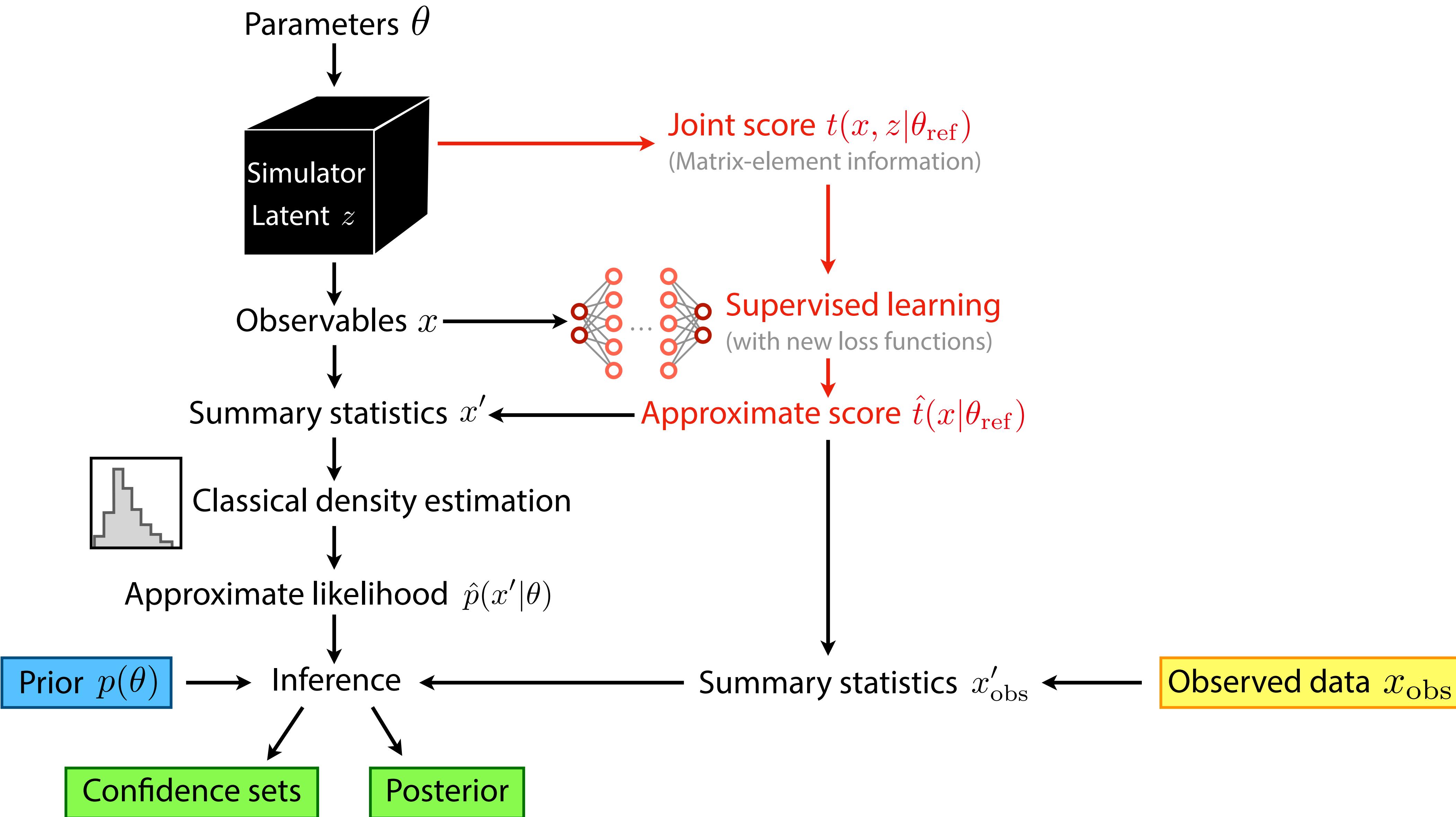
Neural optimal observables (SALLY)



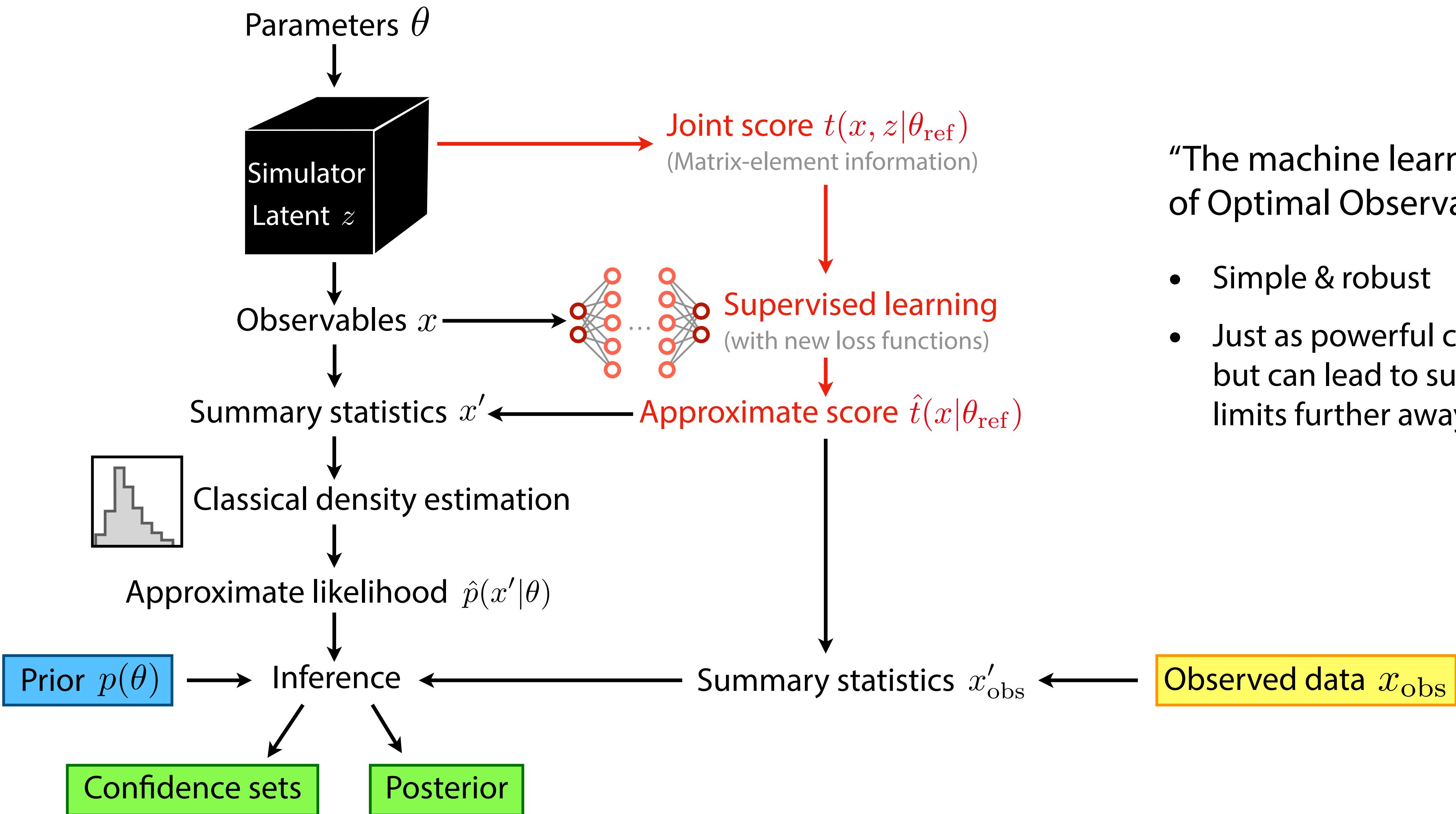
Neural optimal observables (SALLY)



Neural optimal observables (SALLY)



Neural optimal observables (SALLY)



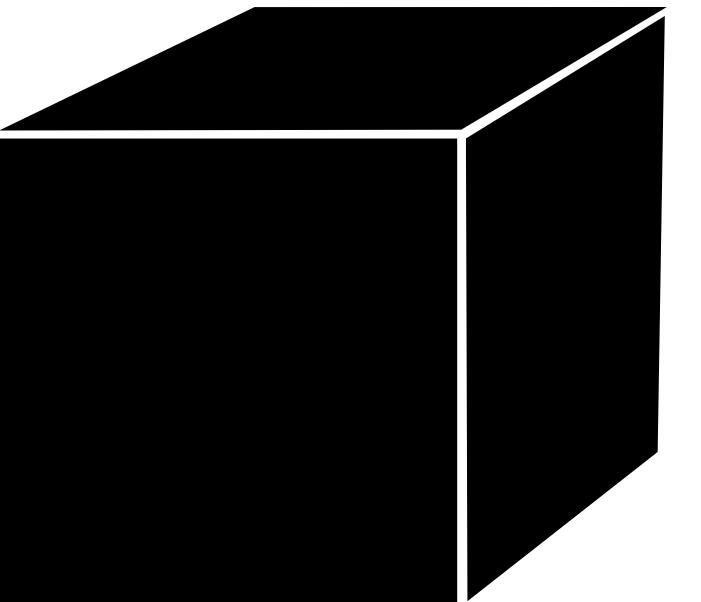
“The machine learning version of Optimal Observables”

- Simple & robust
- Just as powerful close to θ_{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



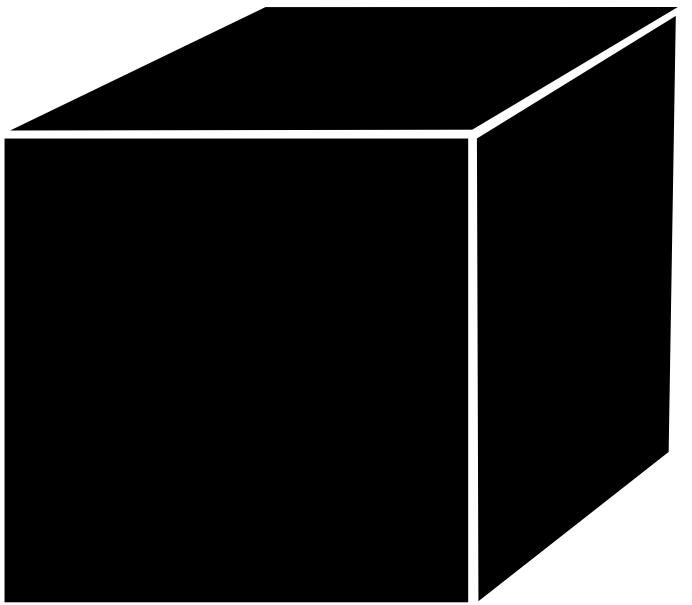
Don't fully trust the simulator?

- 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



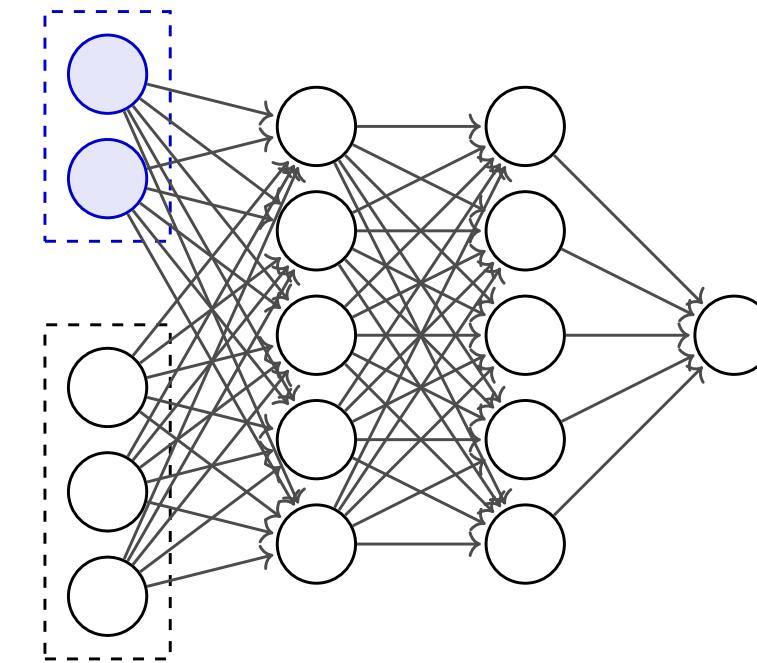
Don't fully trust the simulator?

- 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)

MadMiner automates all of these methods.

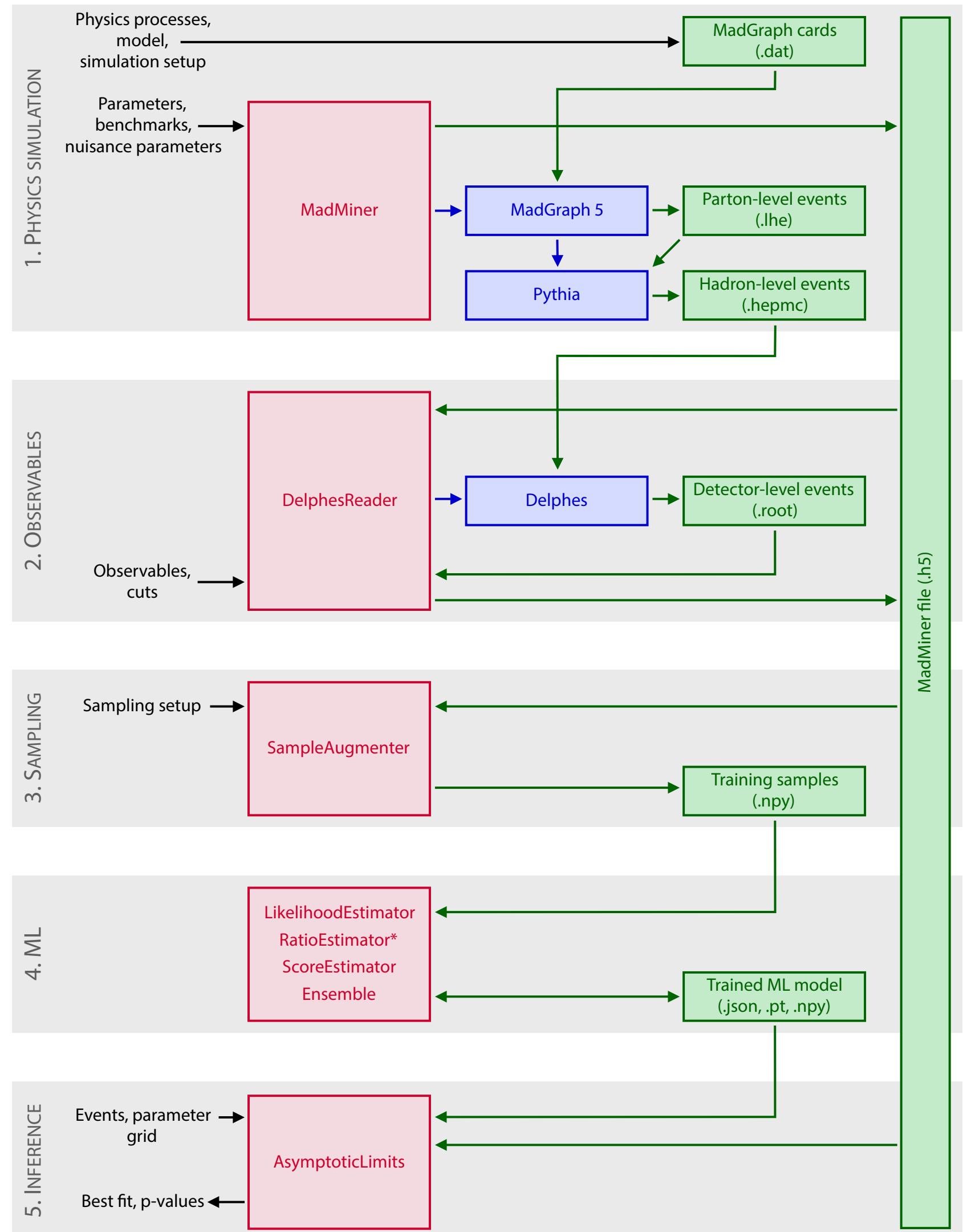
[JB, F. Kling, I. Espejo, K. Cranmer 1907.10621]

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 physicists 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

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

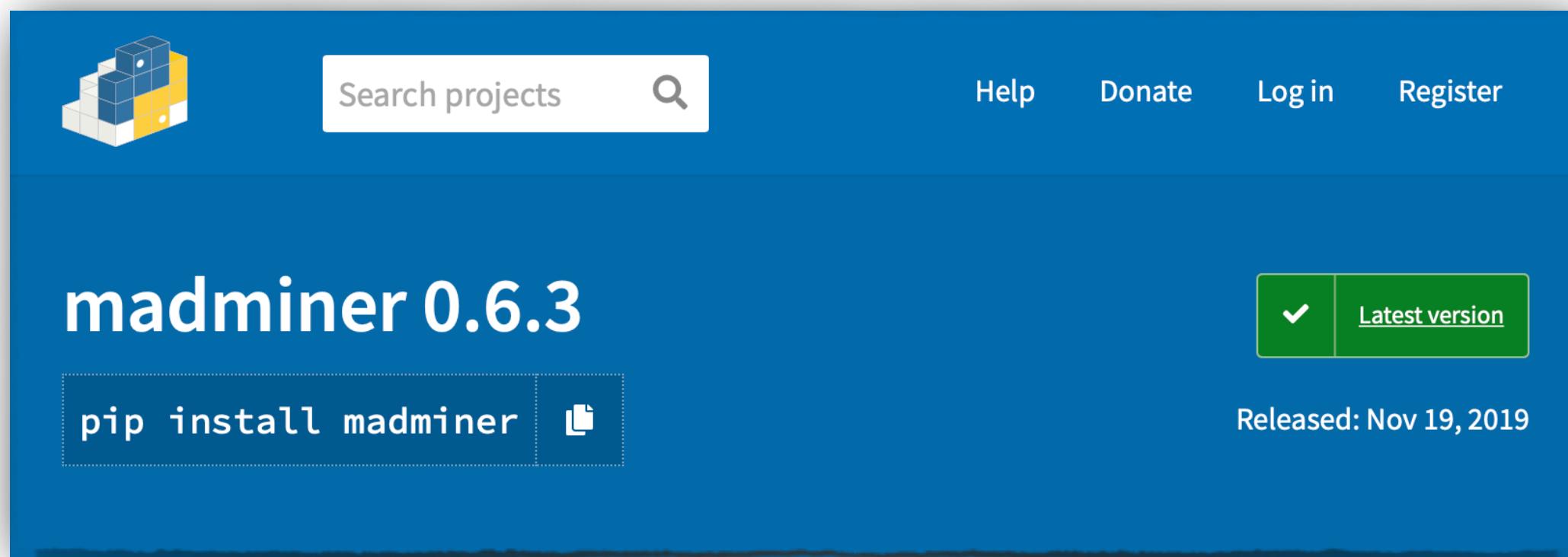
² Department of Physics and Astronomy, University of California, Irvine, CA 92697, USA

³ SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA

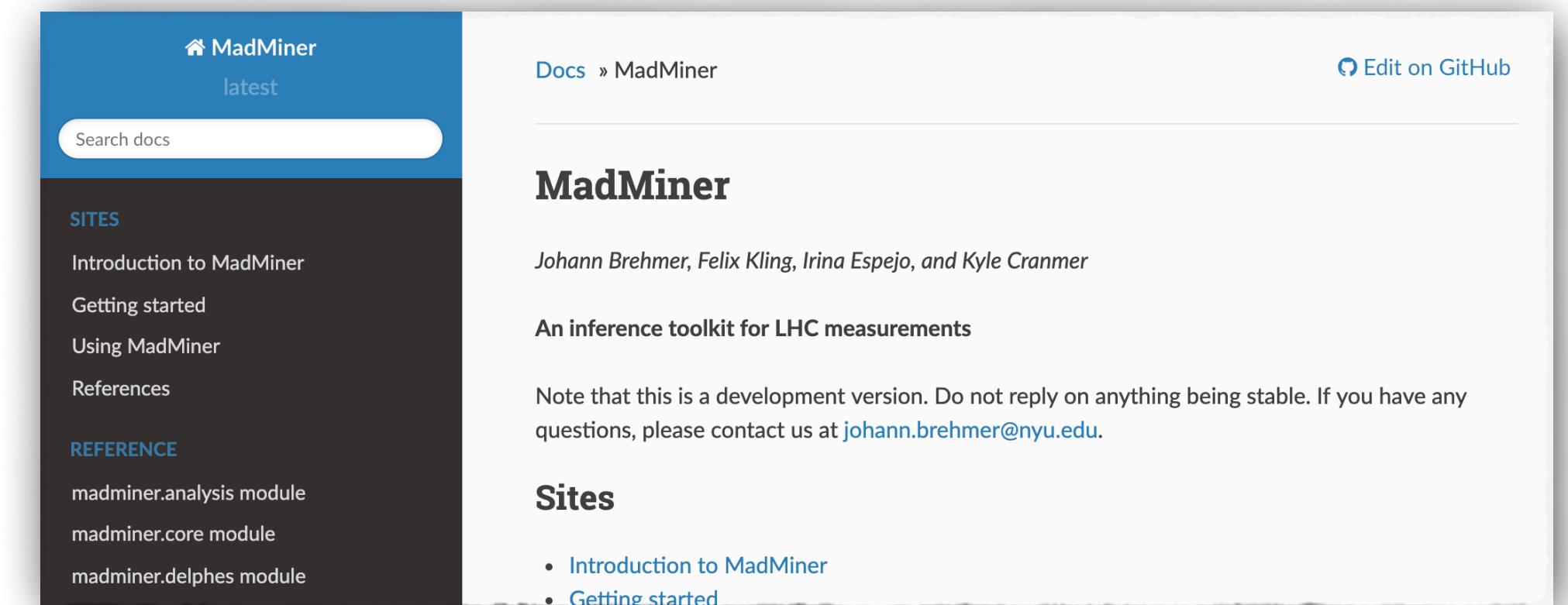
Precision measurements at the LHC often require analyzing high-dimensional event data for subtle kinematic signatures, which is challenging for established analysis methods. Recently, a powerful family of multivariate inference techniques that leverage both matrix element information and machine learning 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

Repository and tutorials:
github.com/johannbrehmer/madminer

Paper with detailed explanations:
[1907.10621](https://arxiv.org/abs/1907.10621)



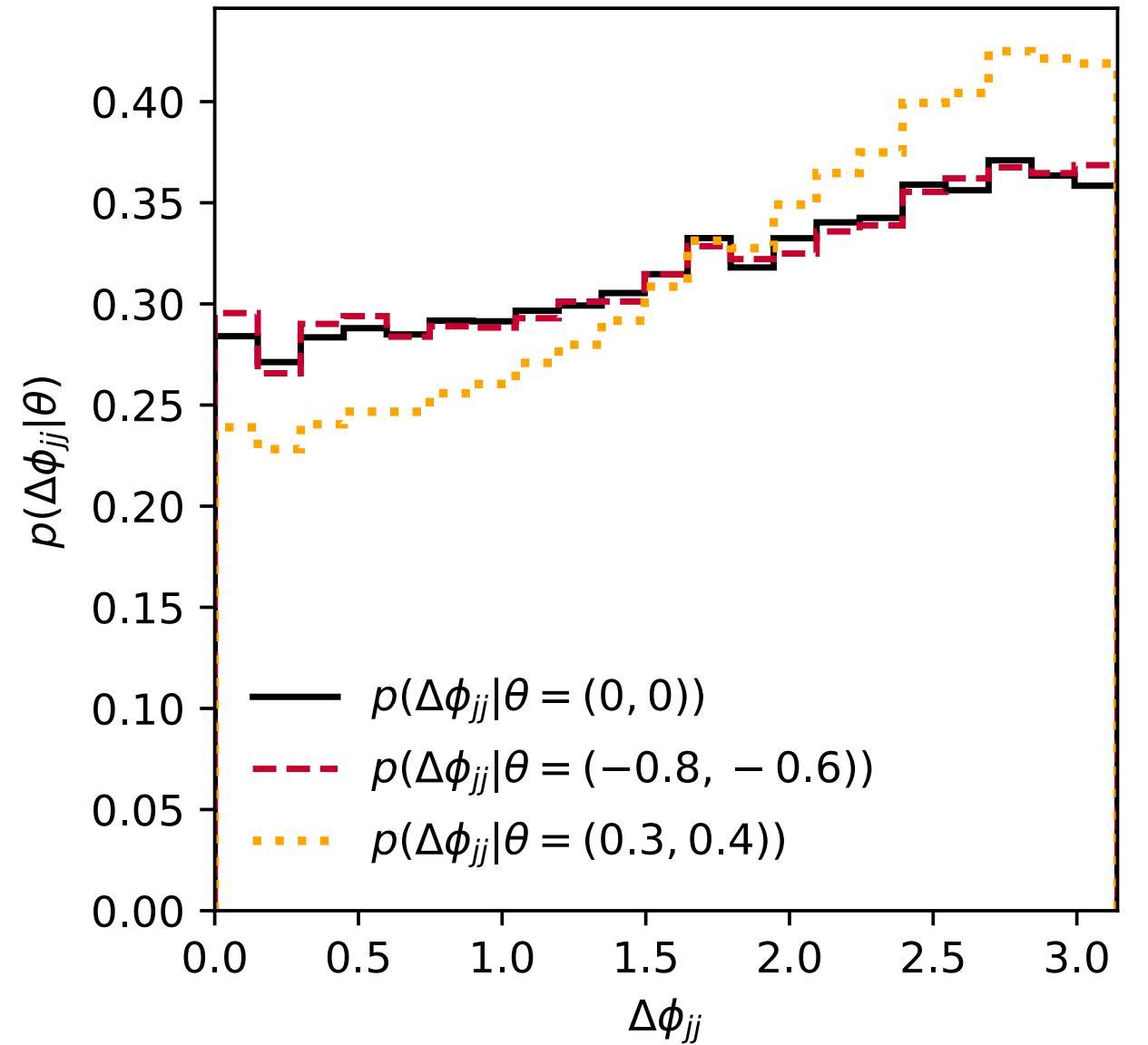
Installation:
`pip install madminer`



API documentation:
madminer.readthedocs.io

These techniques let us constrain
effective theories more effectively.

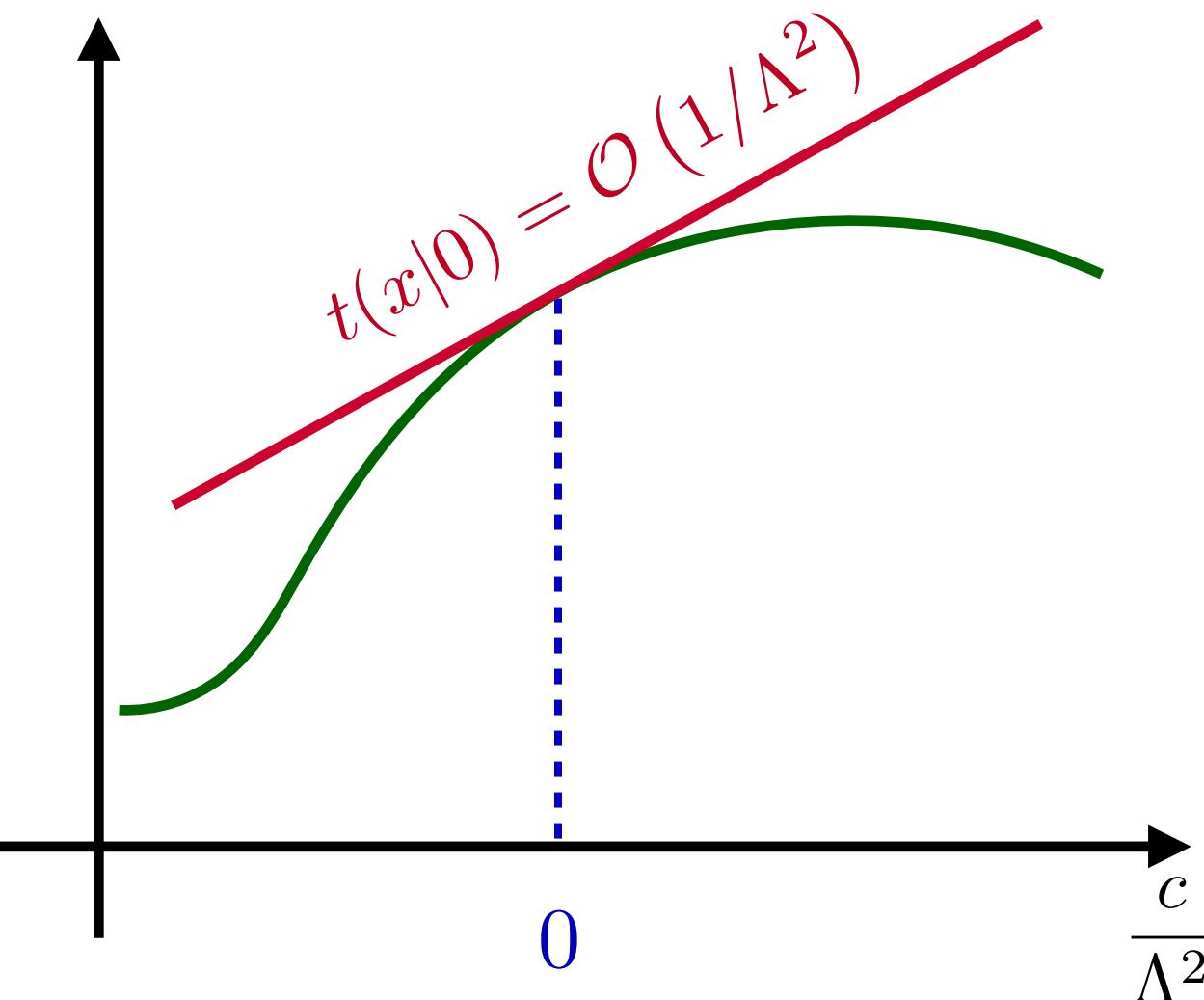
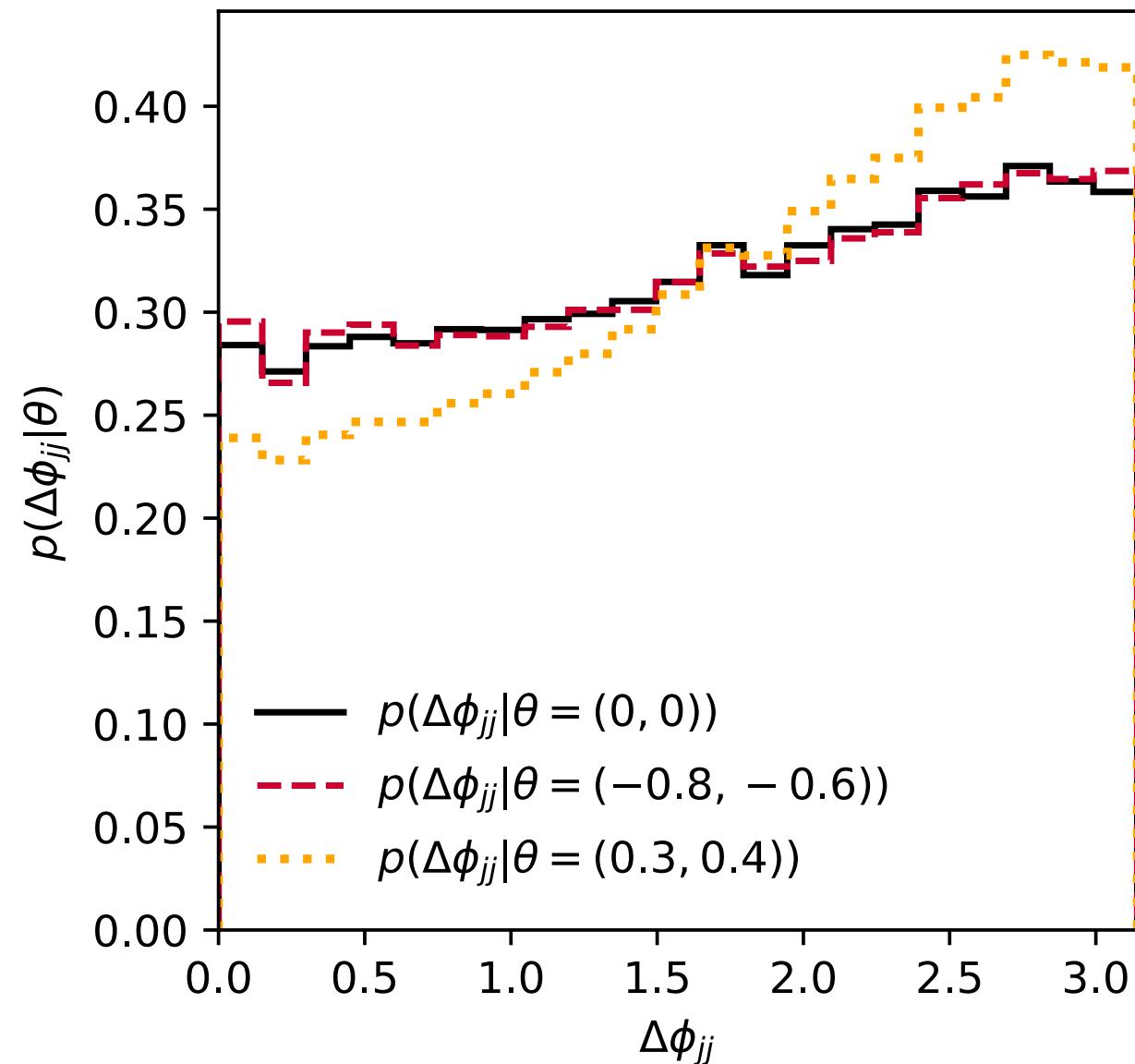
Perfect match for EFT measurements



- Good for subtle kinematic effects

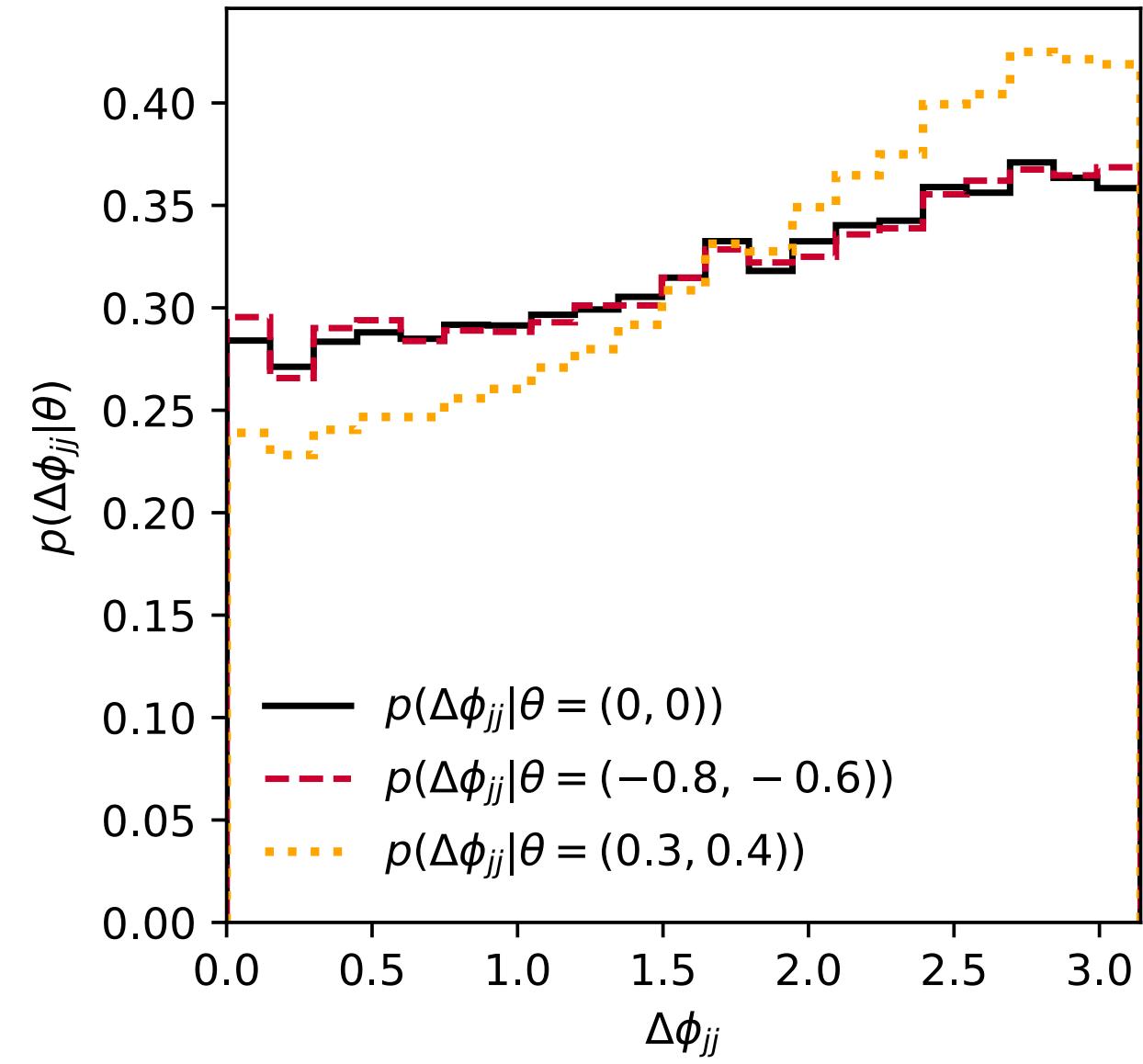
(Subtle point: Large overlap of kinematic distributions reduces variance of joint likelihood ratio / joint score)

Perfect match for EFT measurements

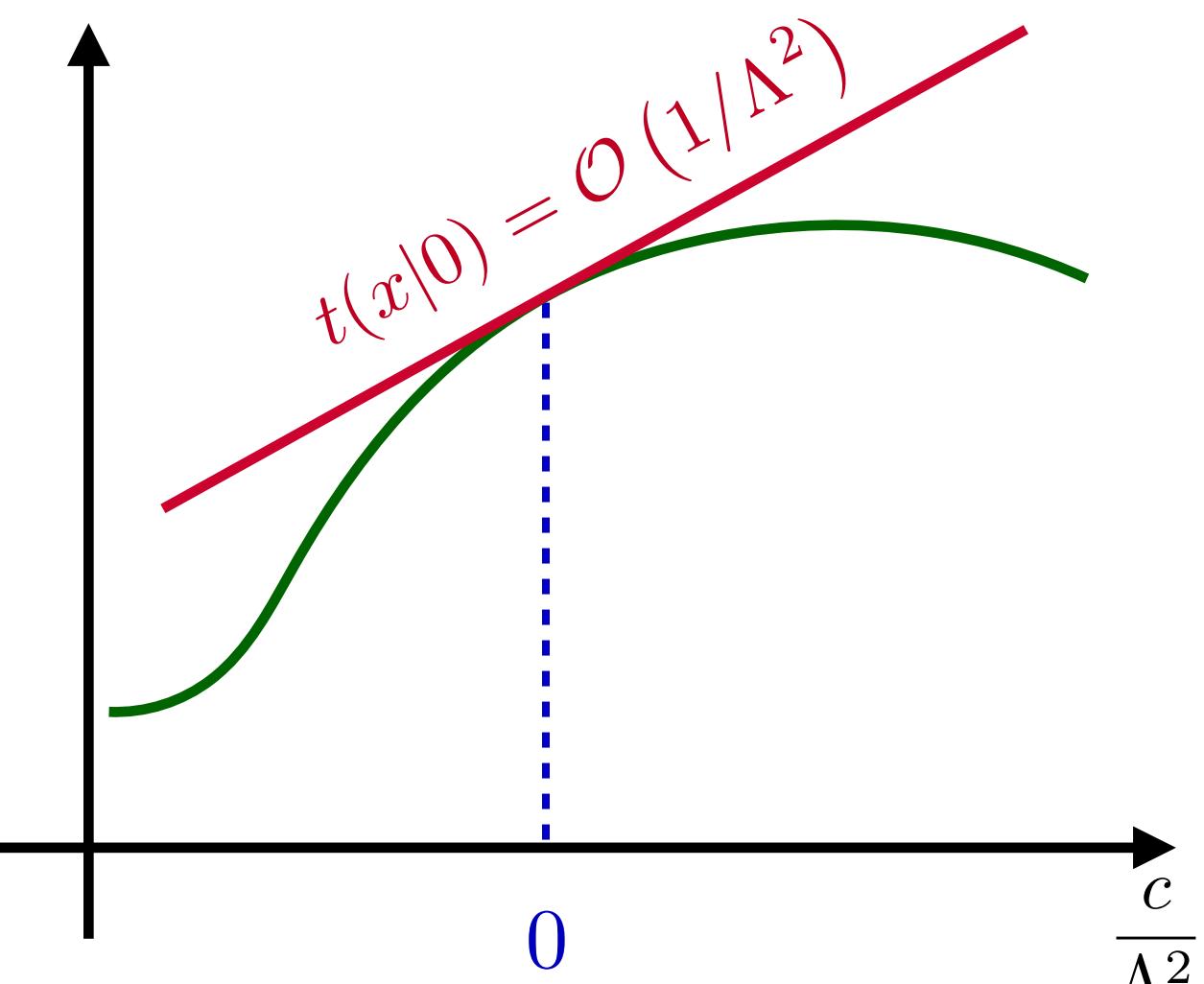


- Good for subtle kinematic effects
(Subtle point: Large overlap of kinematic distributions reduces variance of joint likelihood ratio / joint score)
- Interference effects can be isolated using SALLY at the SM (SMALLY?)

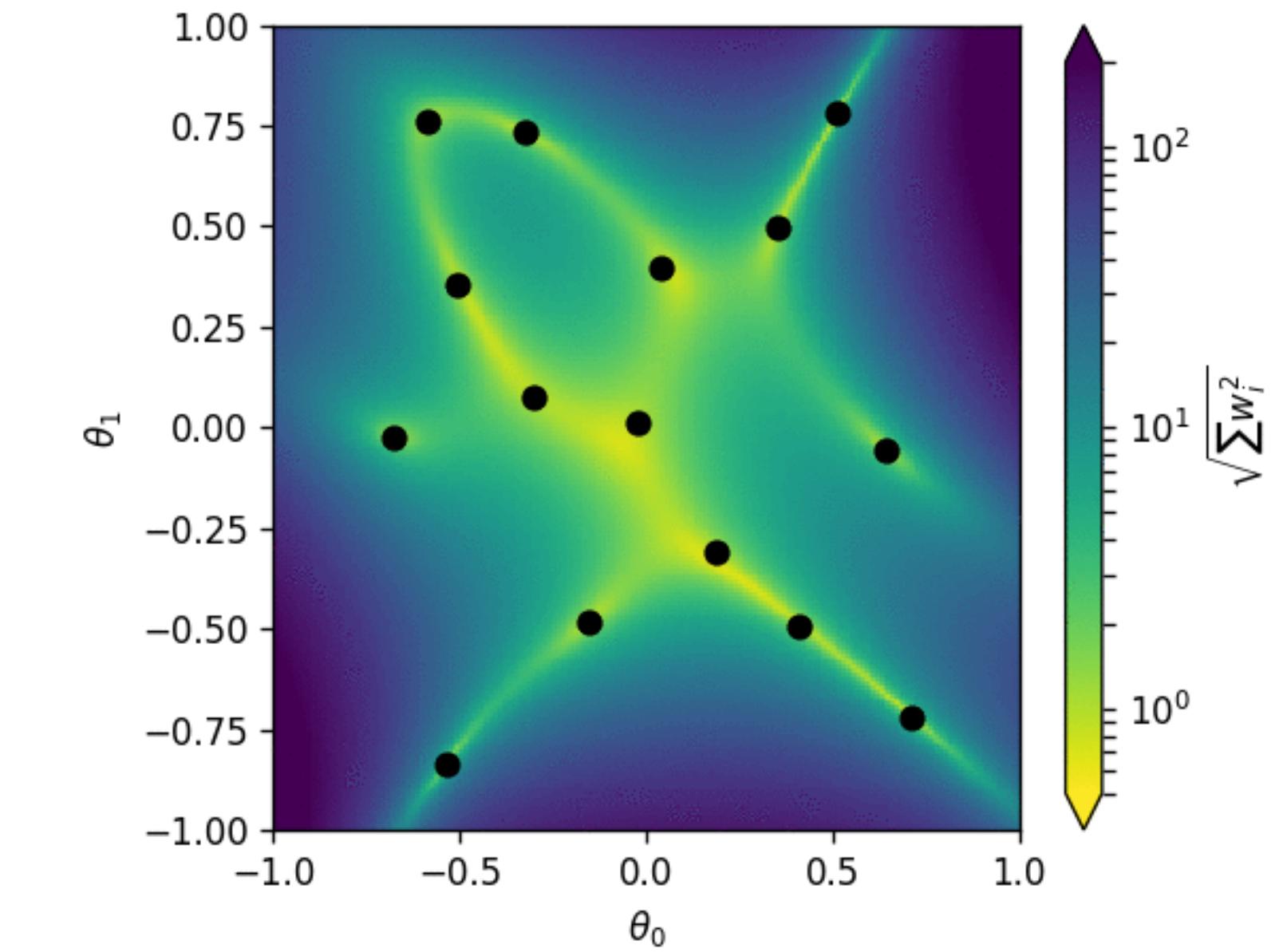
Perfect match for EFT measurements



- Good for subtle kinematic effects
(Subtle point: Large overlap of kinematic distributions reduces variance of joint likelihood ratio / joint score)



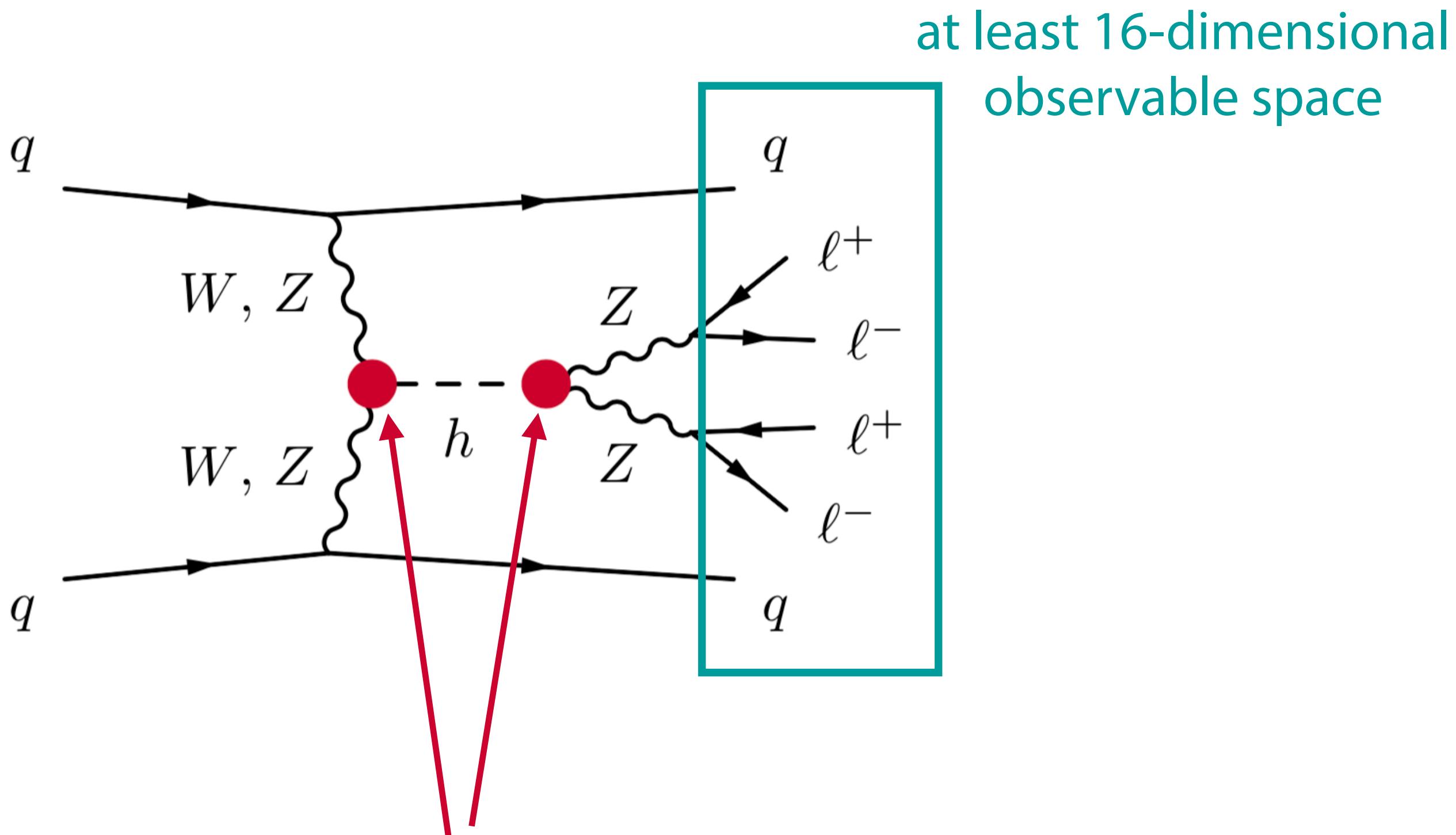
- Interference effects can be isolated using SALLY at the SM (SMALLY?)



- Morphing techniques allow fast reweighting to any parameter points
[e.g. ATL-PHYS-PUB-2015-047]

Proof of concept: Higgs production in weak boson fusion

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



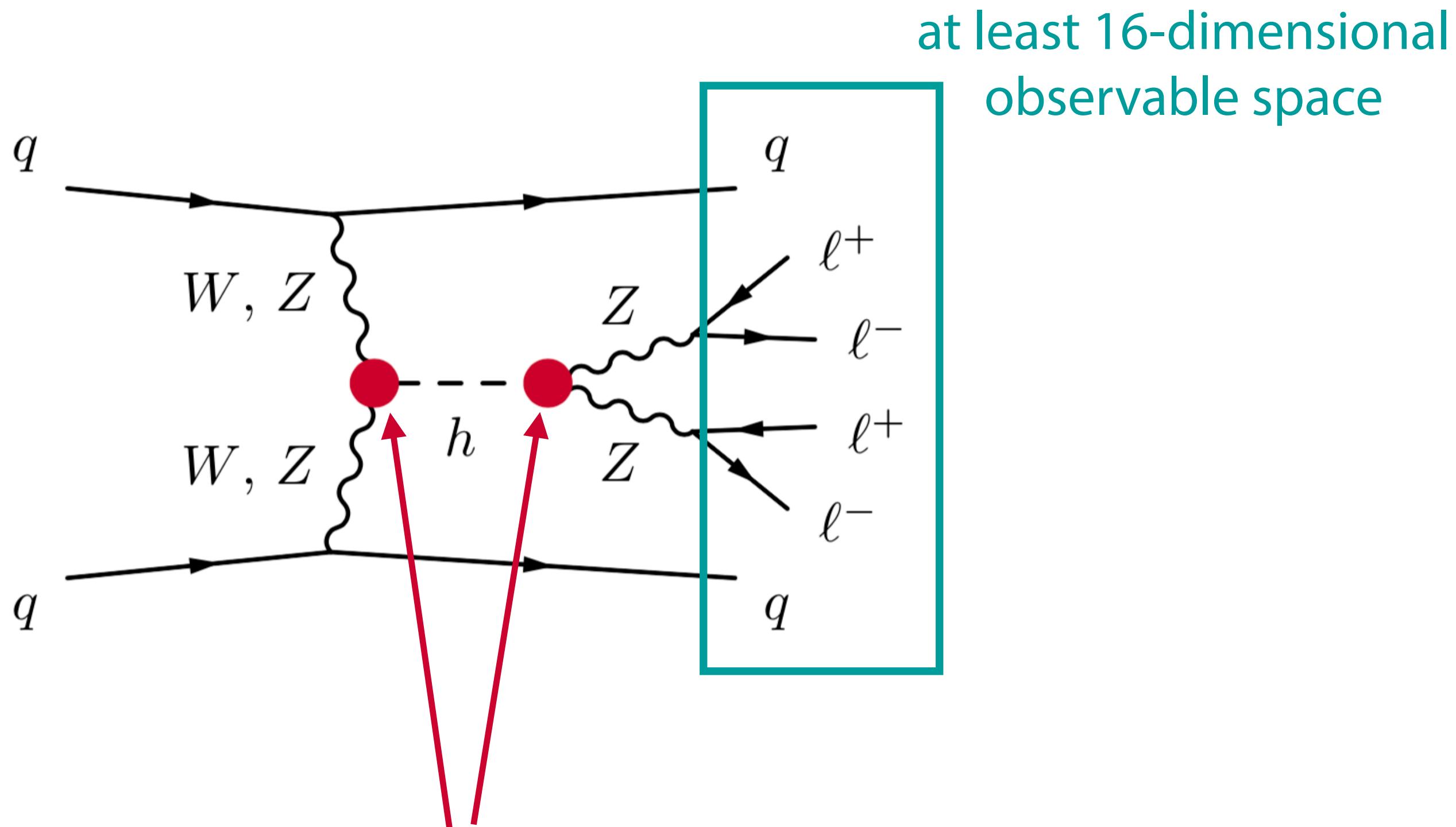
Exciting new physics might hide here!

We parameterize it with two EFT coefficients:

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

Proof of concept: Higgs production in weak boson fusion

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



We parameterize it with two EFT coefficients:

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

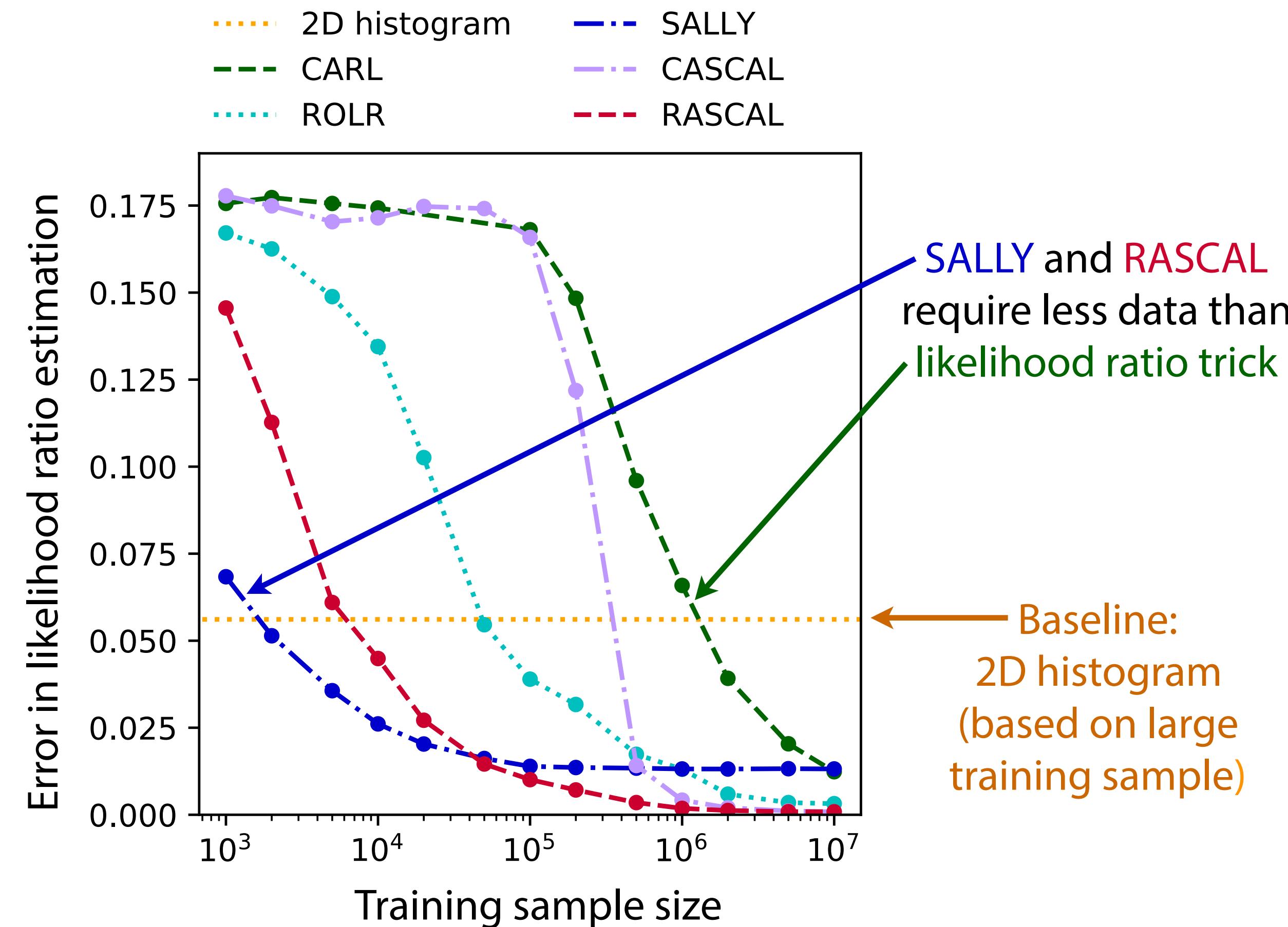
Goal: constrain the two EFT parameters

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

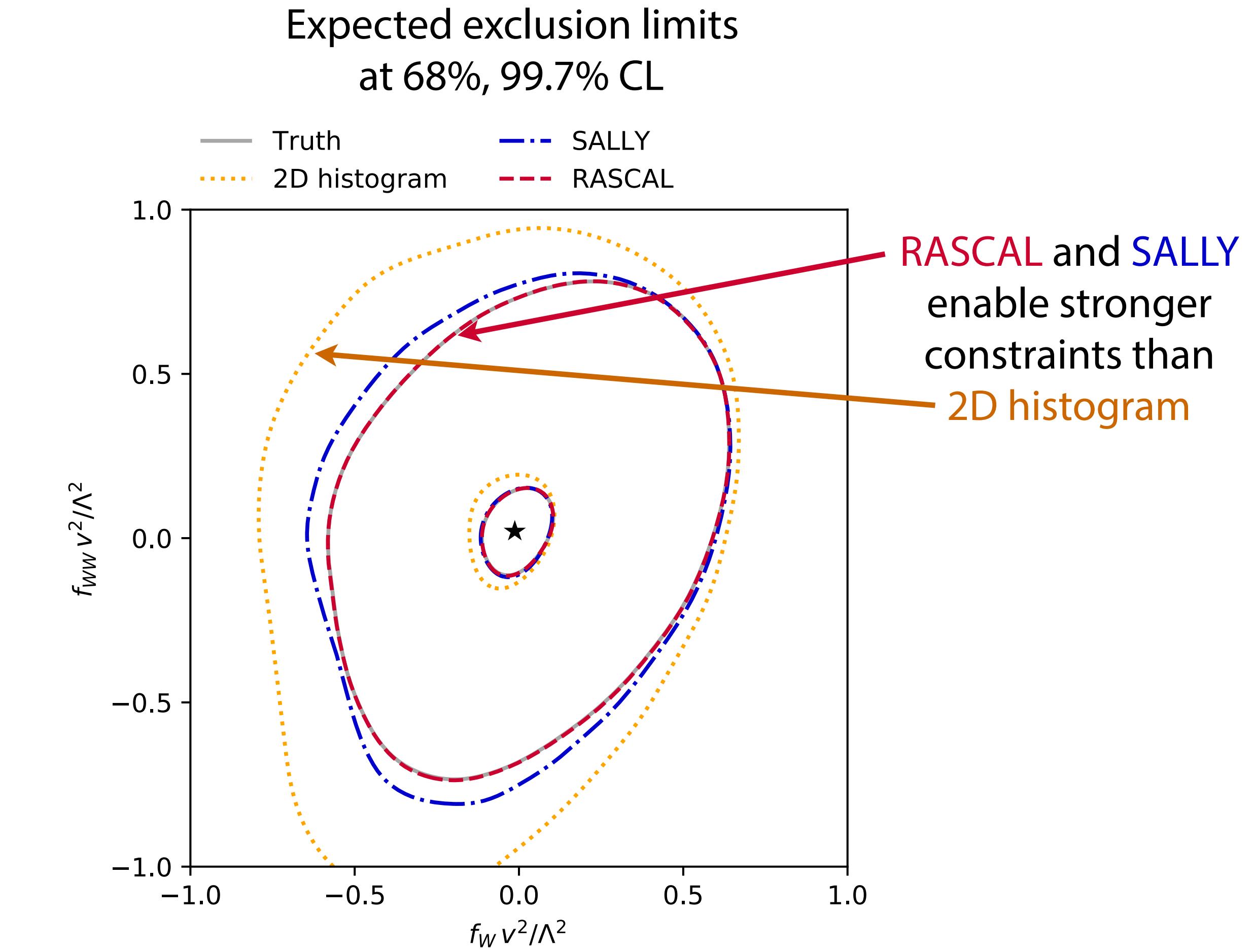
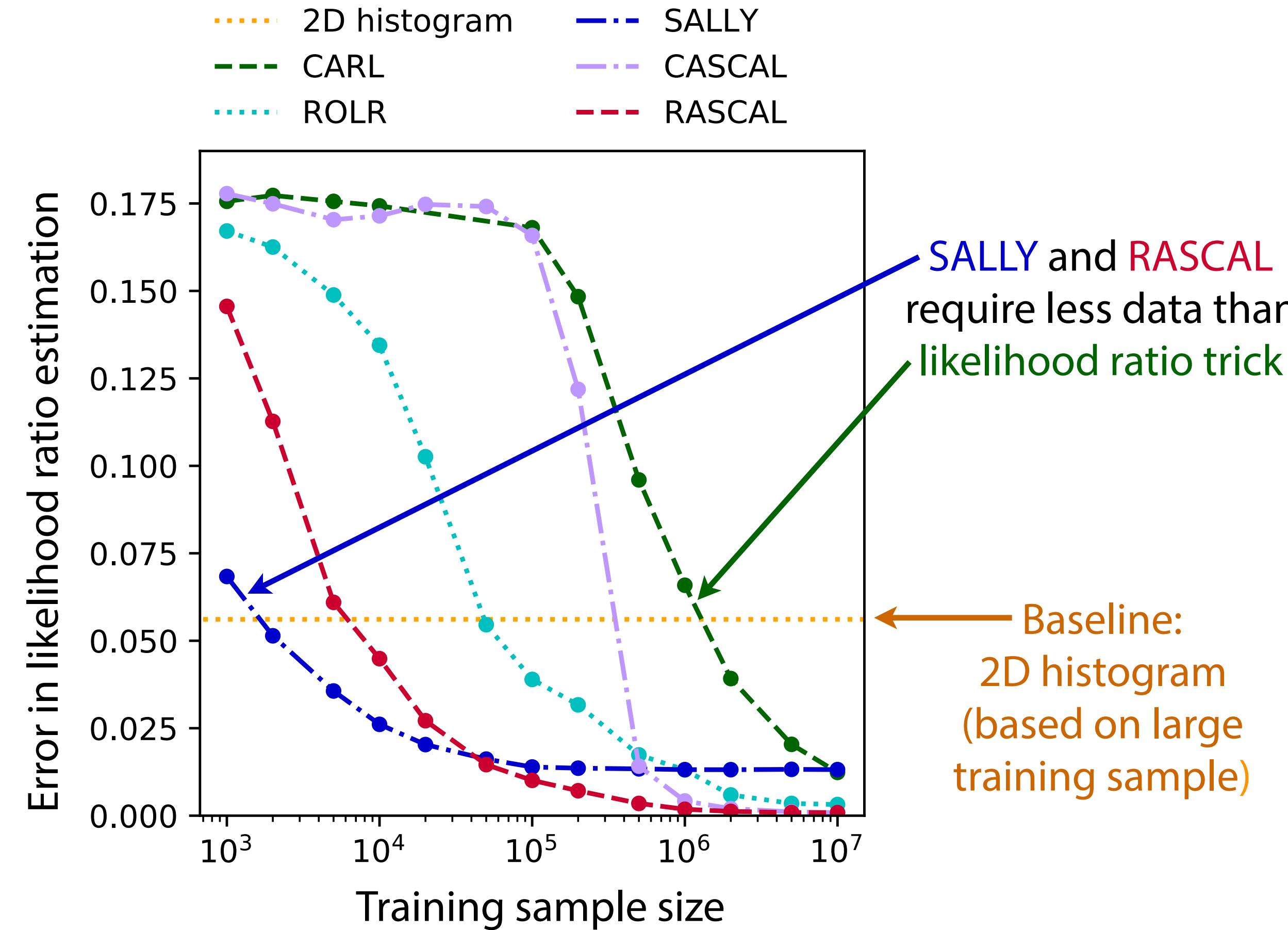
Two scenarios:

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

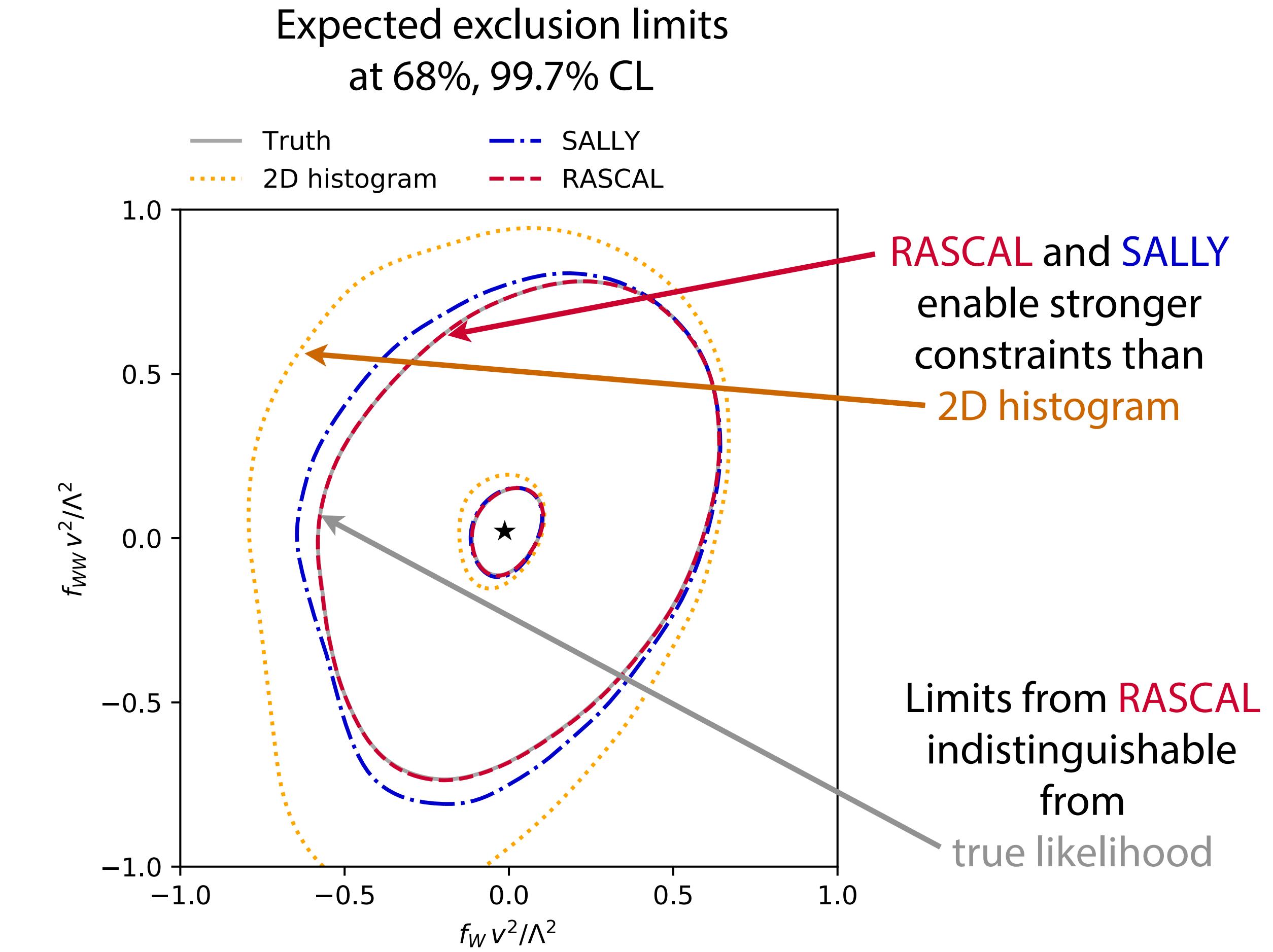
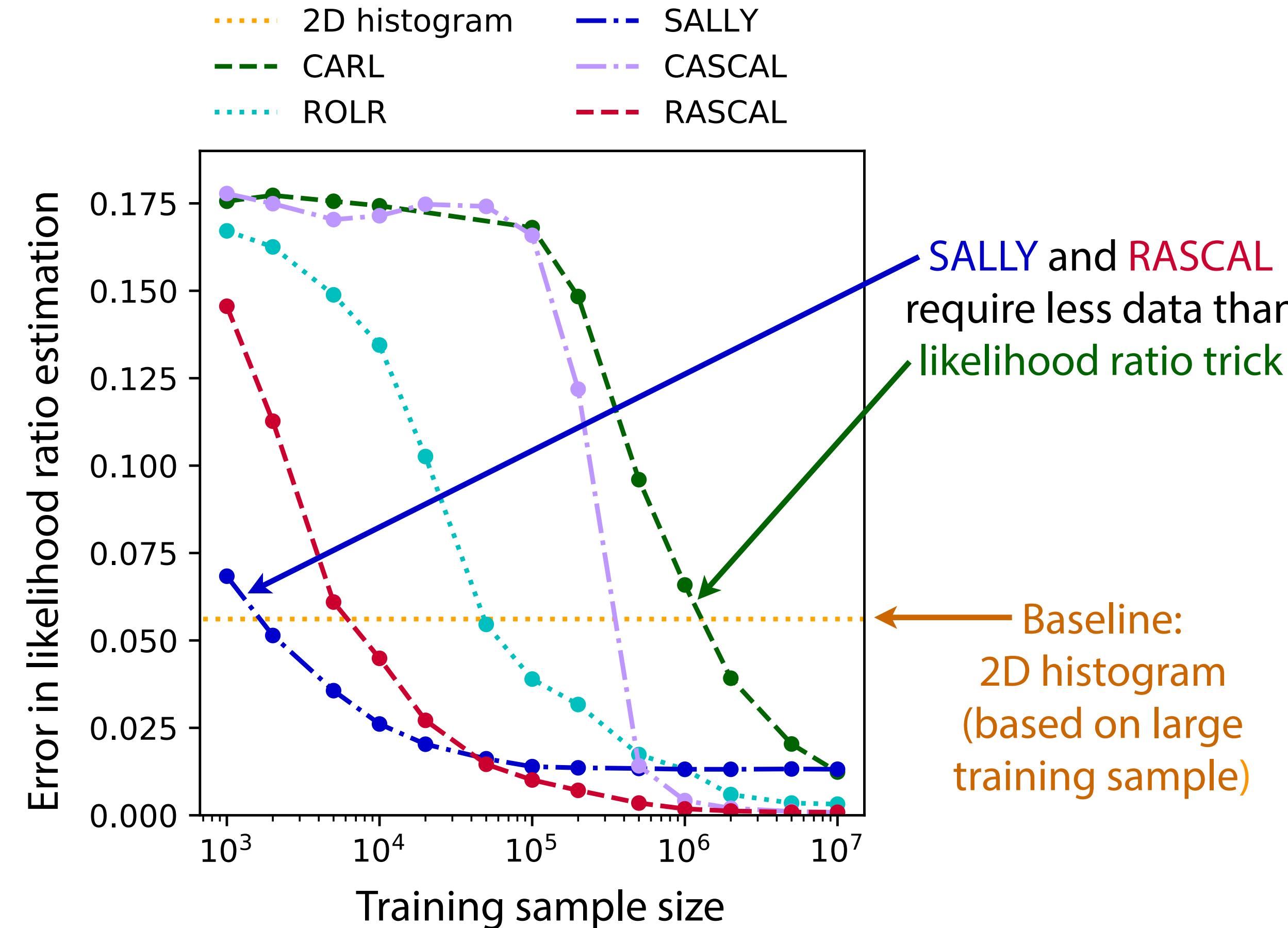
Proof of concept: Stronger constraints with less training data



Proof of concept: Stronger constraints with less training data



Proof of concept: Stronger constraints with less training data



Constraining operators in ttH effectively

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

- Pheno-level analysis of

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

with MadGraph + Pythia + Delphes

- Inference on three EFT operators:

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

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

Constraining operators in ttH effectively

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

- Pheno-level analysis of

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

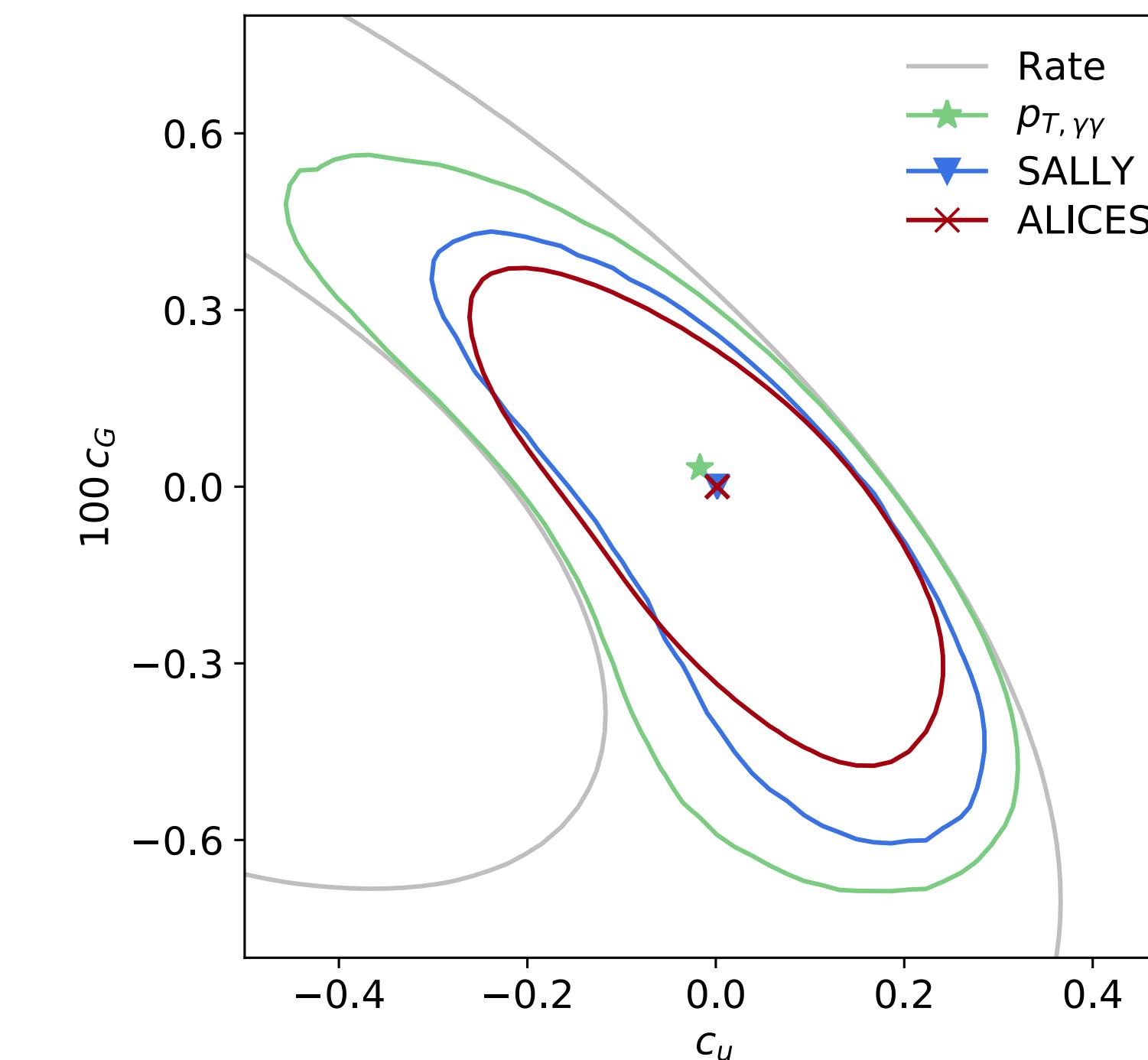
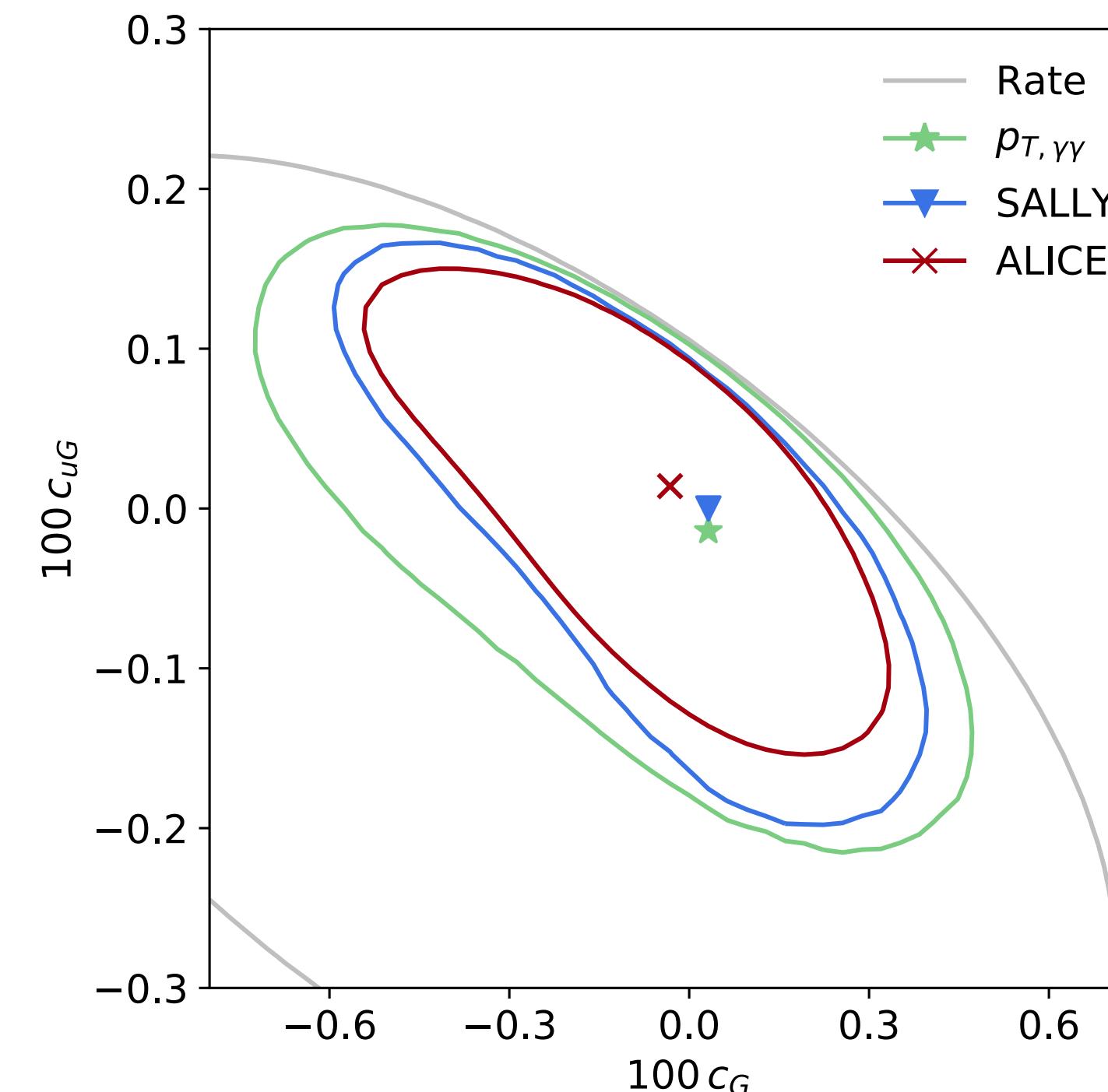
with MadGraph + Pythia + Delphes

- Inference on three EFT operators:

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

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

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

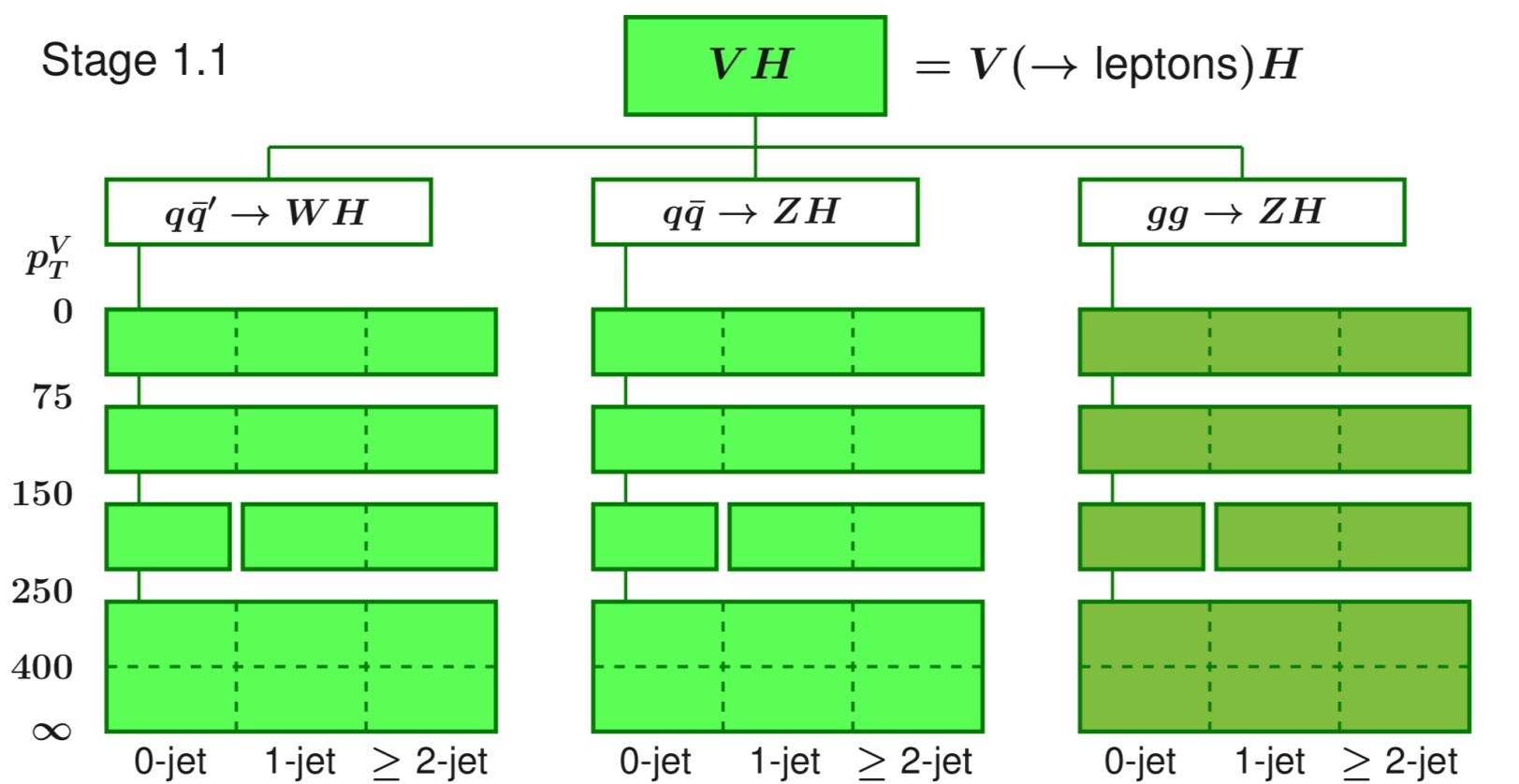


Benchmarking STXS in WH

[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

$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\square} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger \phi) \square (\phi^\dagger \phi) - \frac{1}{4} (\phi^\dagger D^\mu \phi)^* (\phi^\dagger D_\mu \phi)$$

$$\mathcal{O}_{HW} = \phi^\dagger \phi W_{\mu\nu}^a W^{\mu\nu a}$$

$$\mathcal{O}_{Hq}^{(3)} = (\phi^\dagger i \overleftrightarrow{D}_\mu^a \phi) (\overline{Q}_L \sigma^a \gamma^\mu Q_L),$$

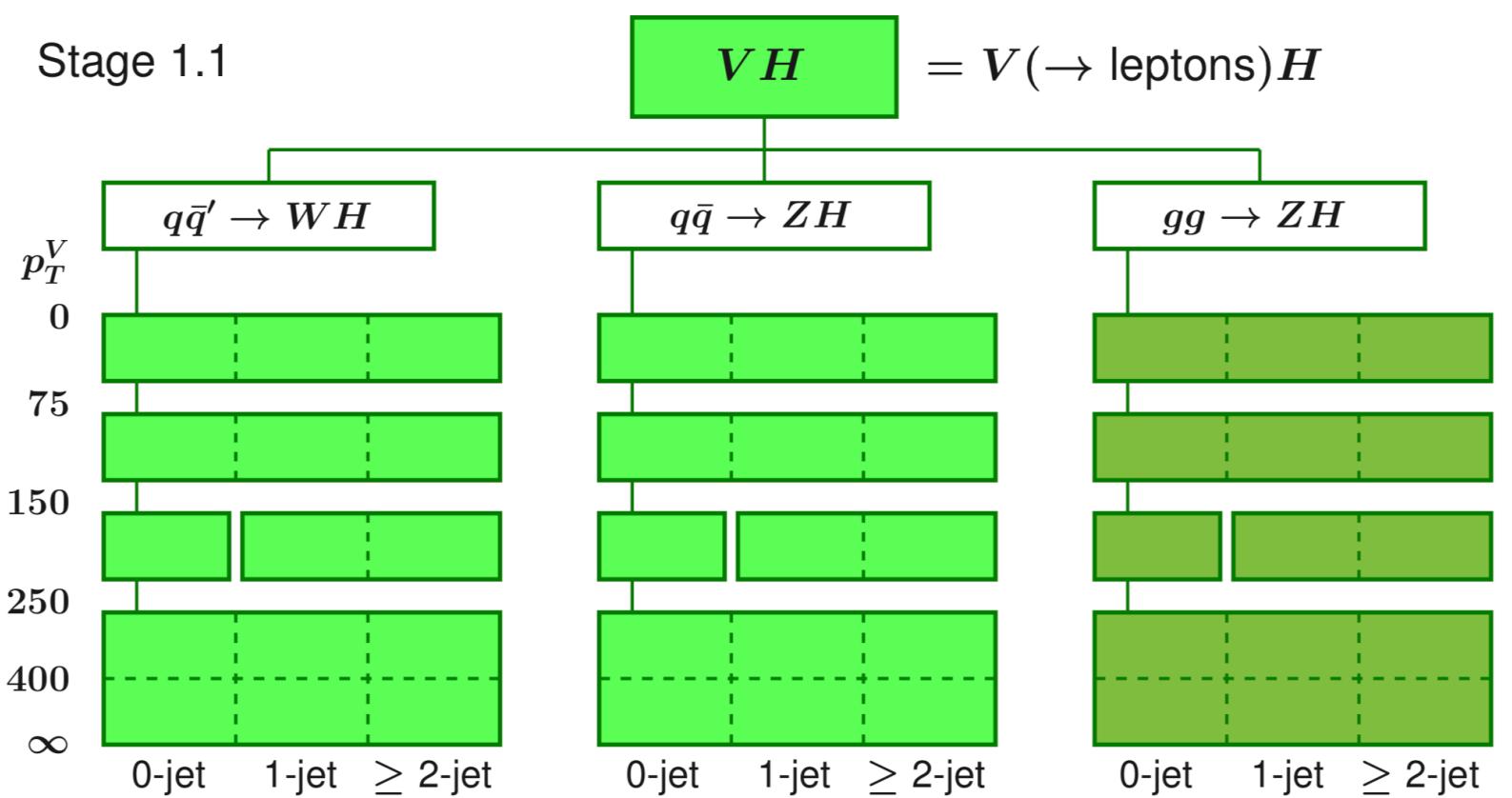
can we extract from $pp \rightarrow WH \rightarrow \ell\nu b\bar{b}$?

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

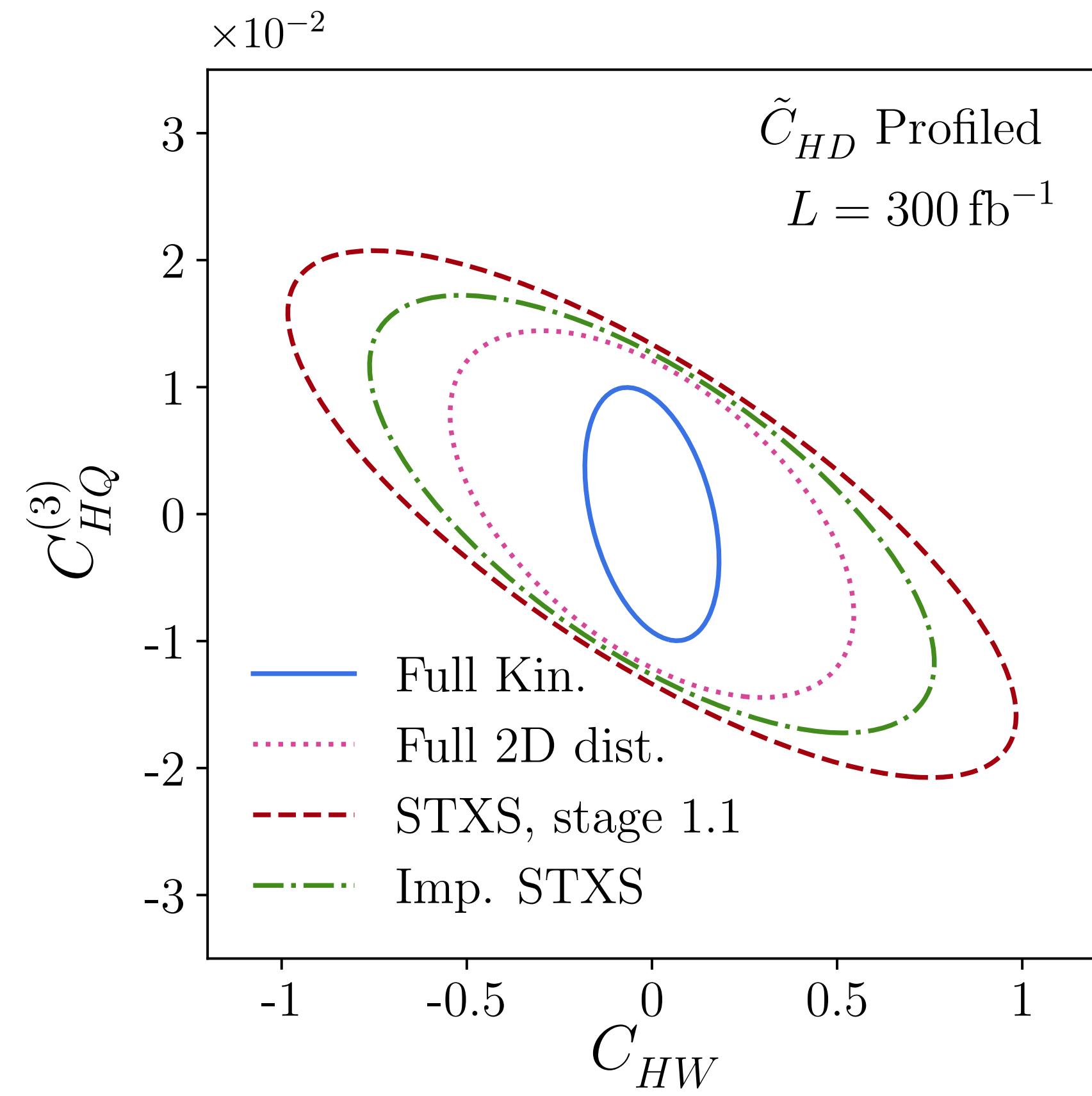
$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\square} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger \phi) \square (\phi^\dagger \phi) - \frac{1}{4} (\phi^\dagger D^\mu \phi)^* (\phi^\dagger D_\mu \phi)$$

$$\mathcal{O}_{HW} = \phi^\dagger \phi W_{\mu\nu}^a W^{\mu\nu a}$$

$$\mathcal{O}_{Hq}^{(3)} = (\phi^\dagger i \overleftrightarrow{D}_\mu^a \phi) (\bar{Q}_L \sigma^a \gamma^\mu Q_L),$$

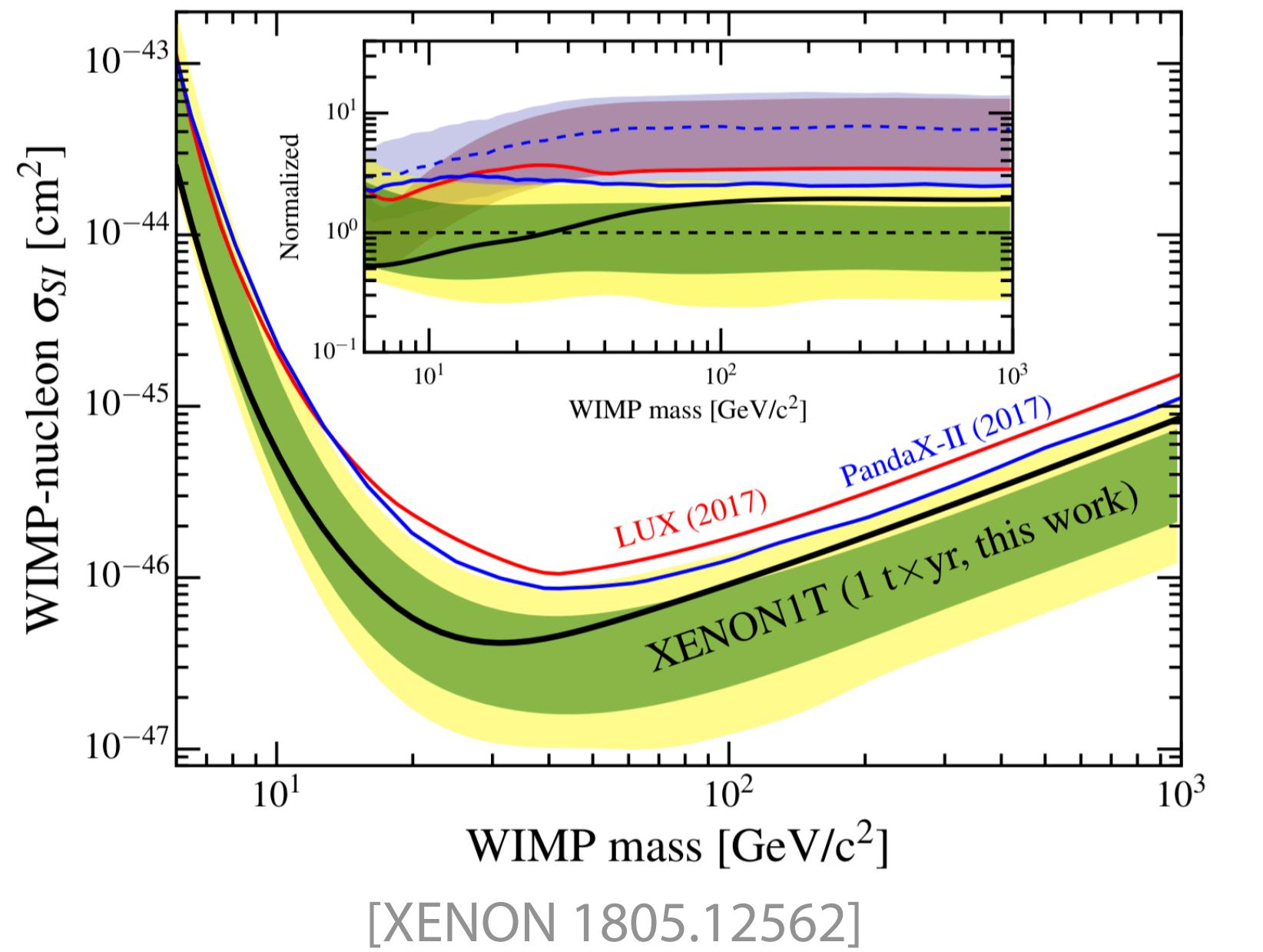
can we extract from $pp \rightarrow WH \rightarrow \ell\nu b\bar{b}$?

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



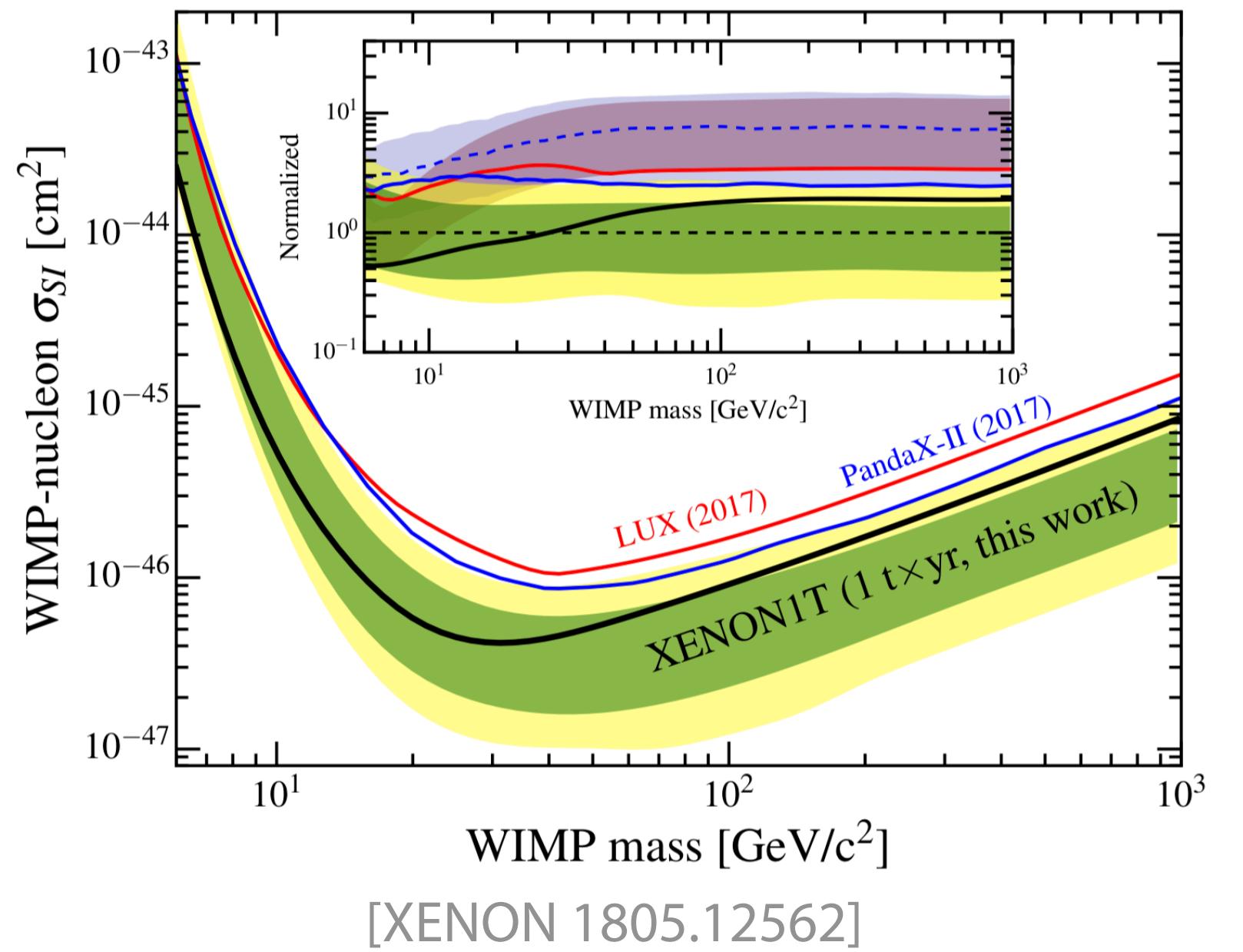
These methods work beyond the LHC.

Dark matter (DM) status



Non-gravitational interactions:
no evidence so far

Dark matter (DM) status

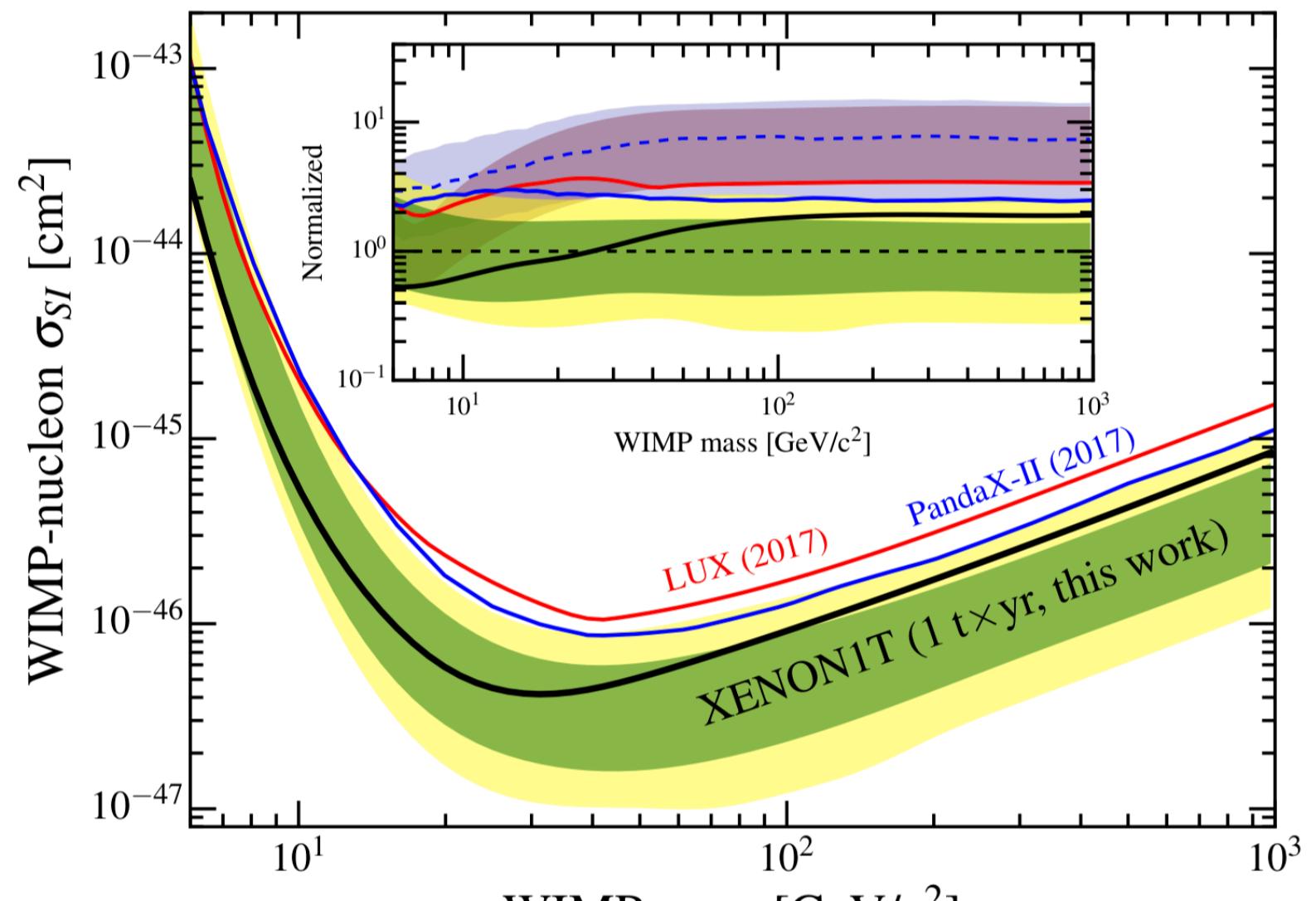


[NASA]

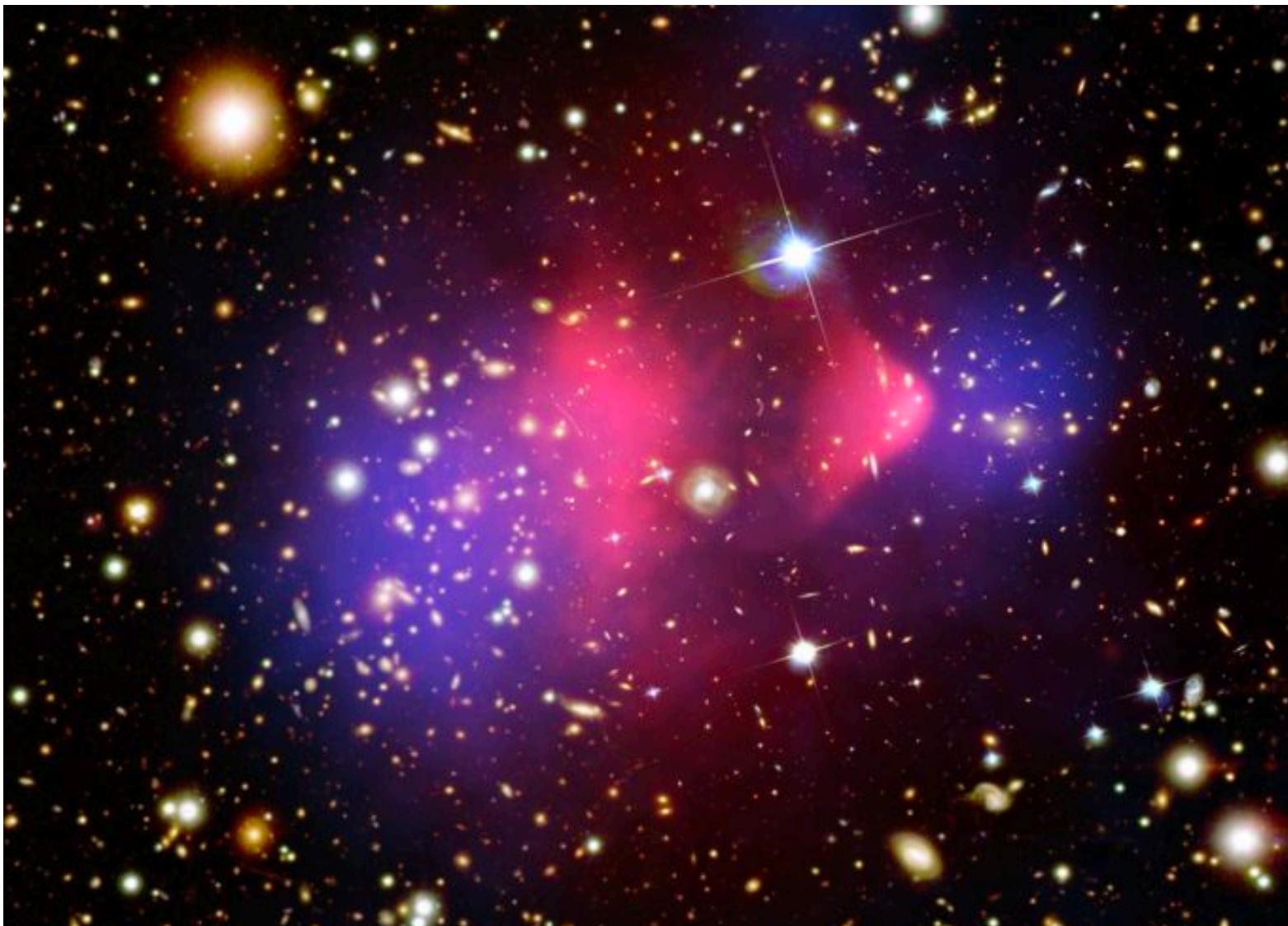
Non-gravitational interactions:
no evidence so far

Gravitational interactions at **large length scales**: plenty of evidence, consistent with Λ CDM

Dark matter (DM) status



[XENON 1805.12562]



[NASA]



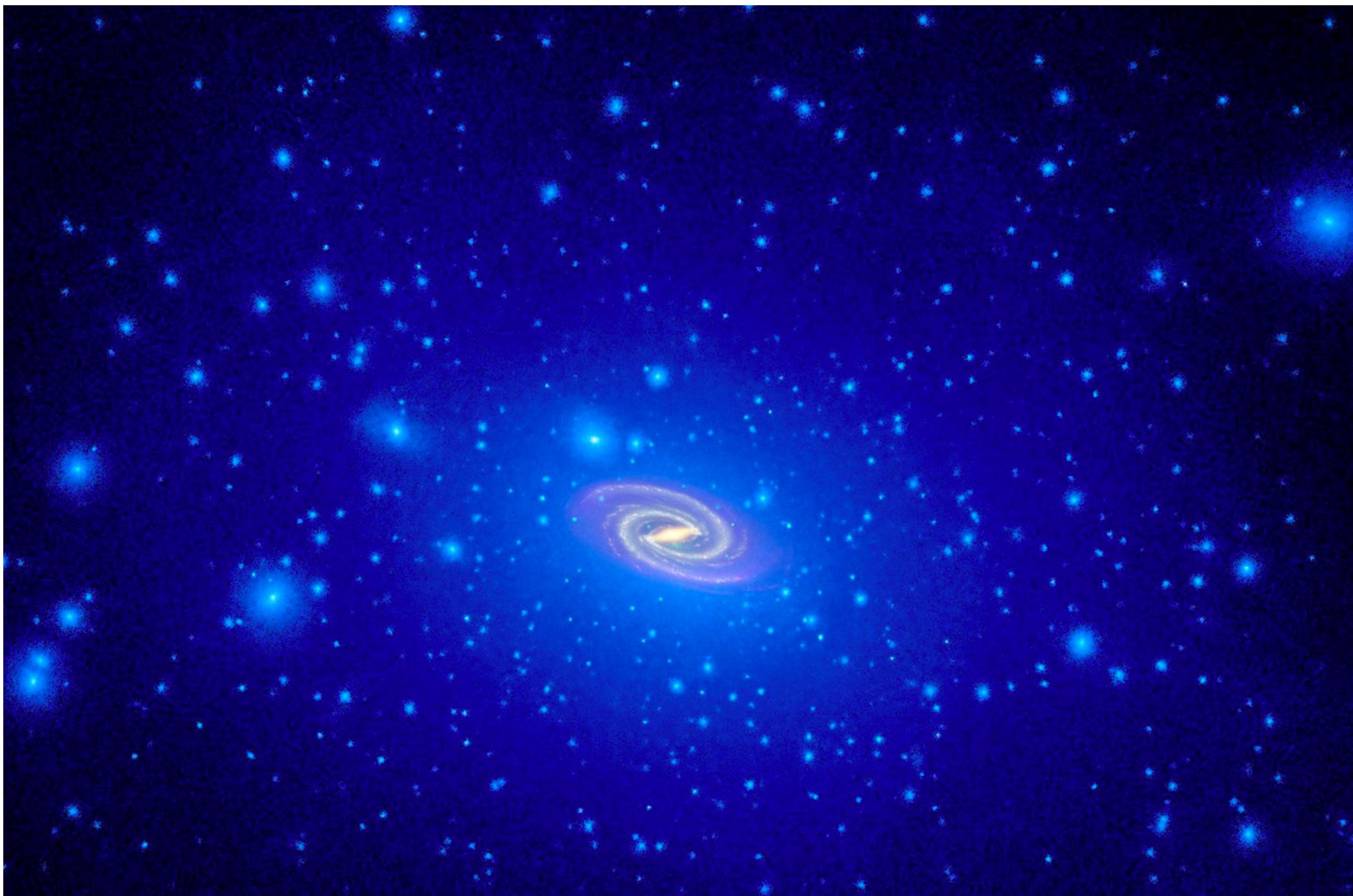
[T. Brown, J.Tumlinson]

Non-gravitational interactions:
no evidence so far

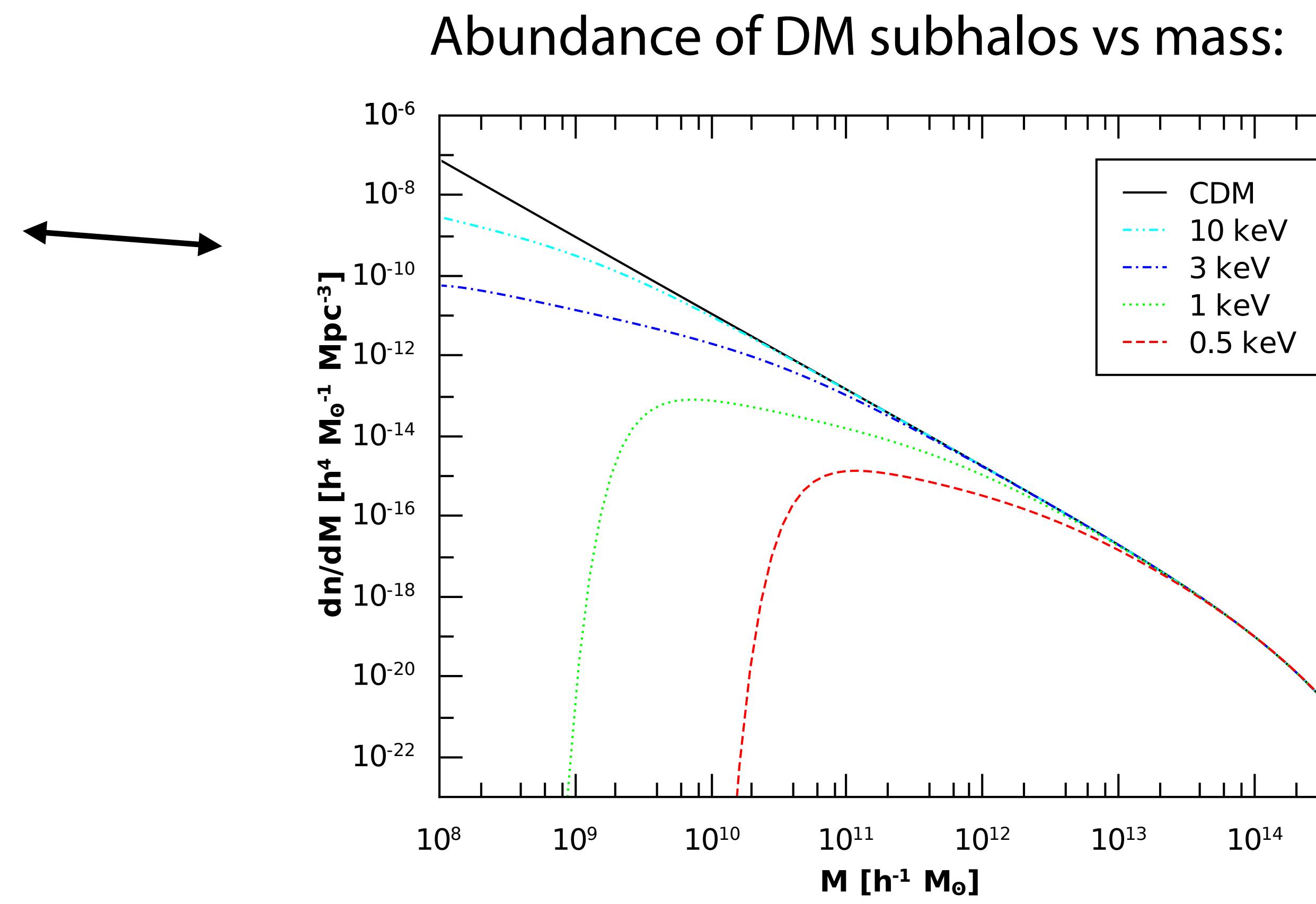
Gravitational interactions at **large length scales**: plenty of evidence, consistent with Λ CDM

Gravitational interactions at **small scales (DM substructure)**: beginning to be probed, many models predict deviations from Λ CDM

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

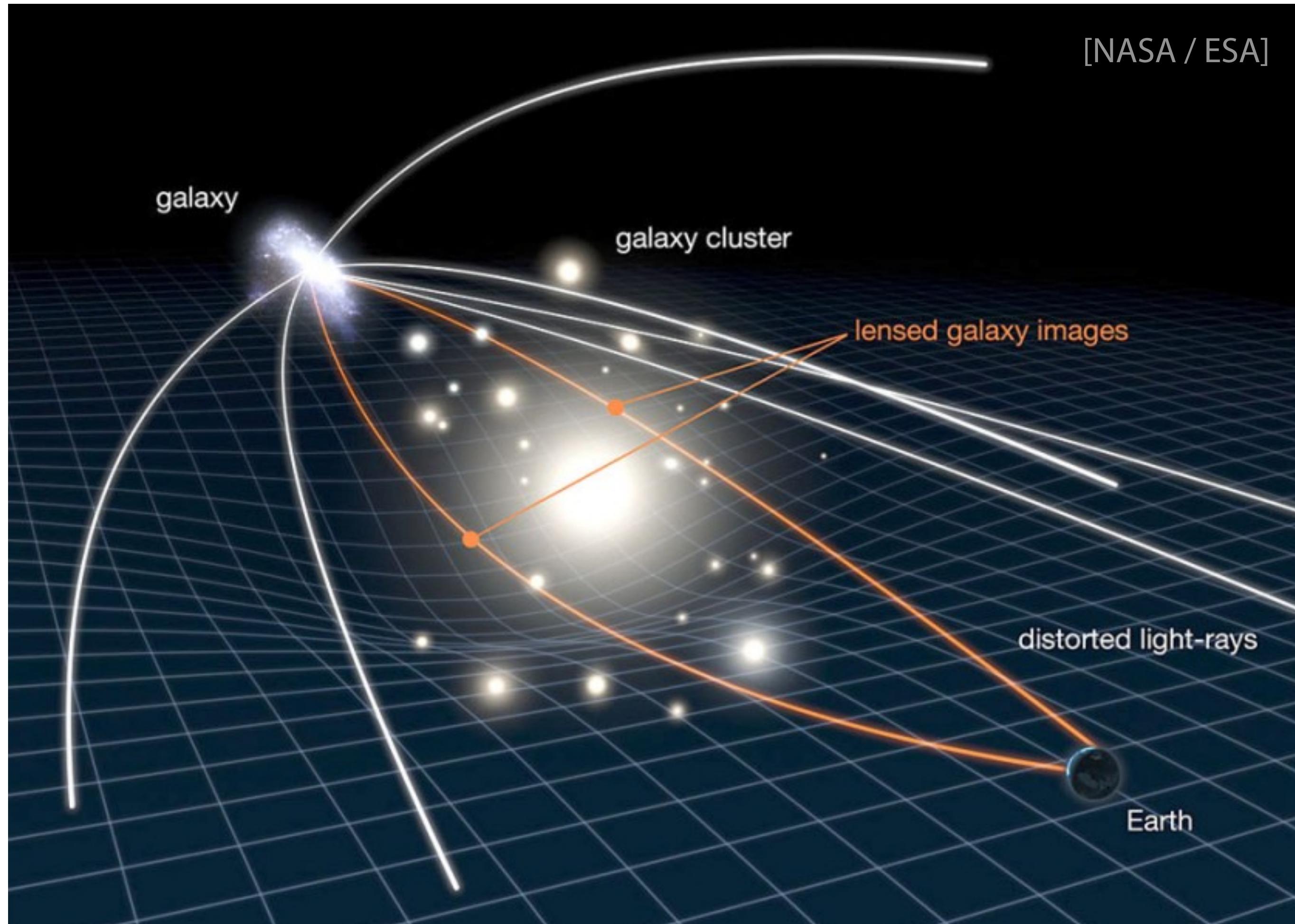


[T. Brown, J.Tumlinson]

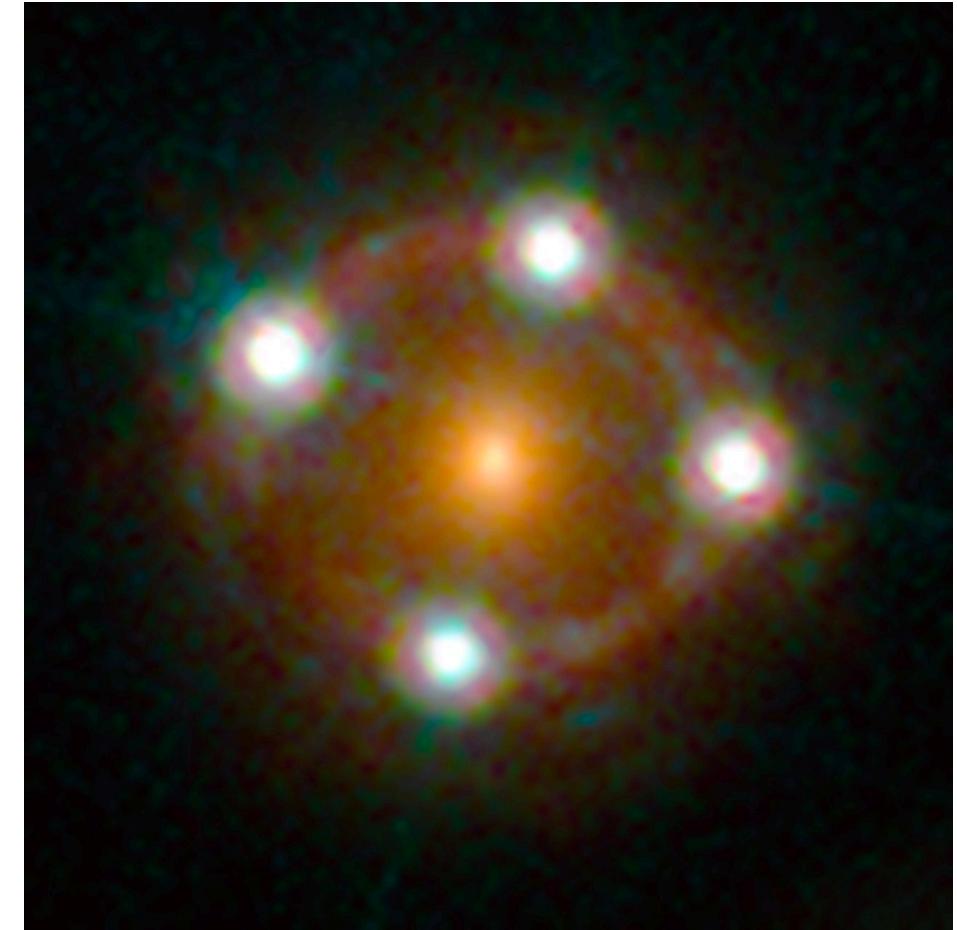


[R. Dunstan et al 1109.6291]

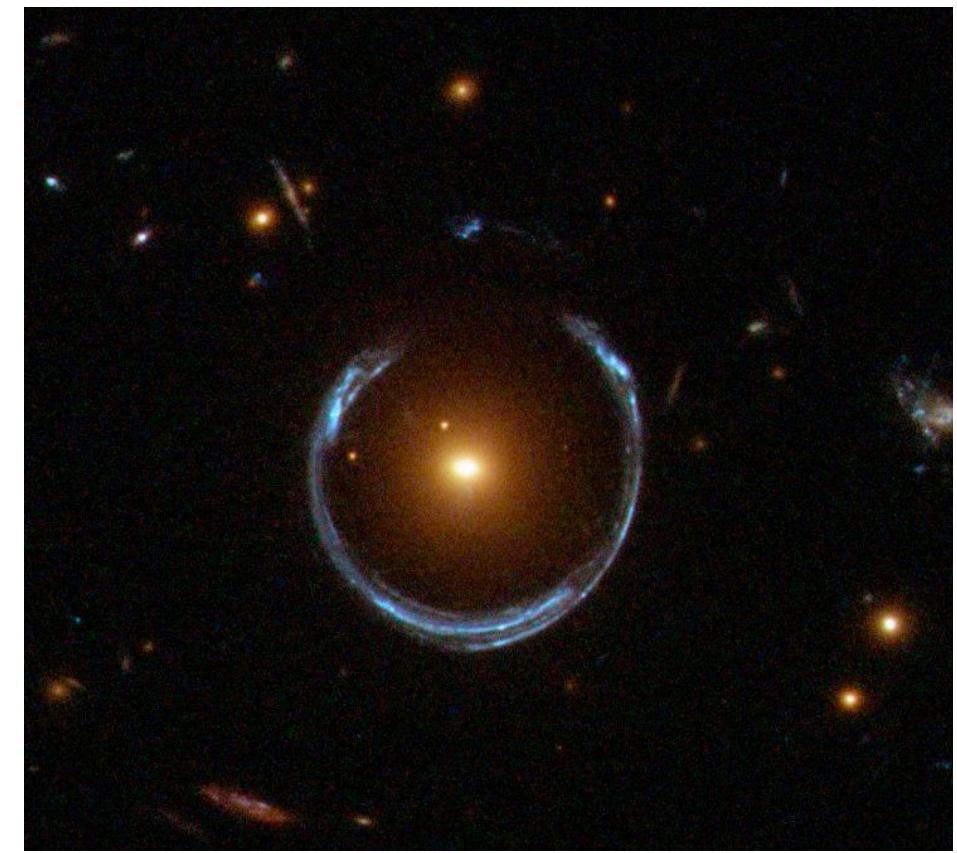
Strong gravitational lensing



Multiple images
of quasars

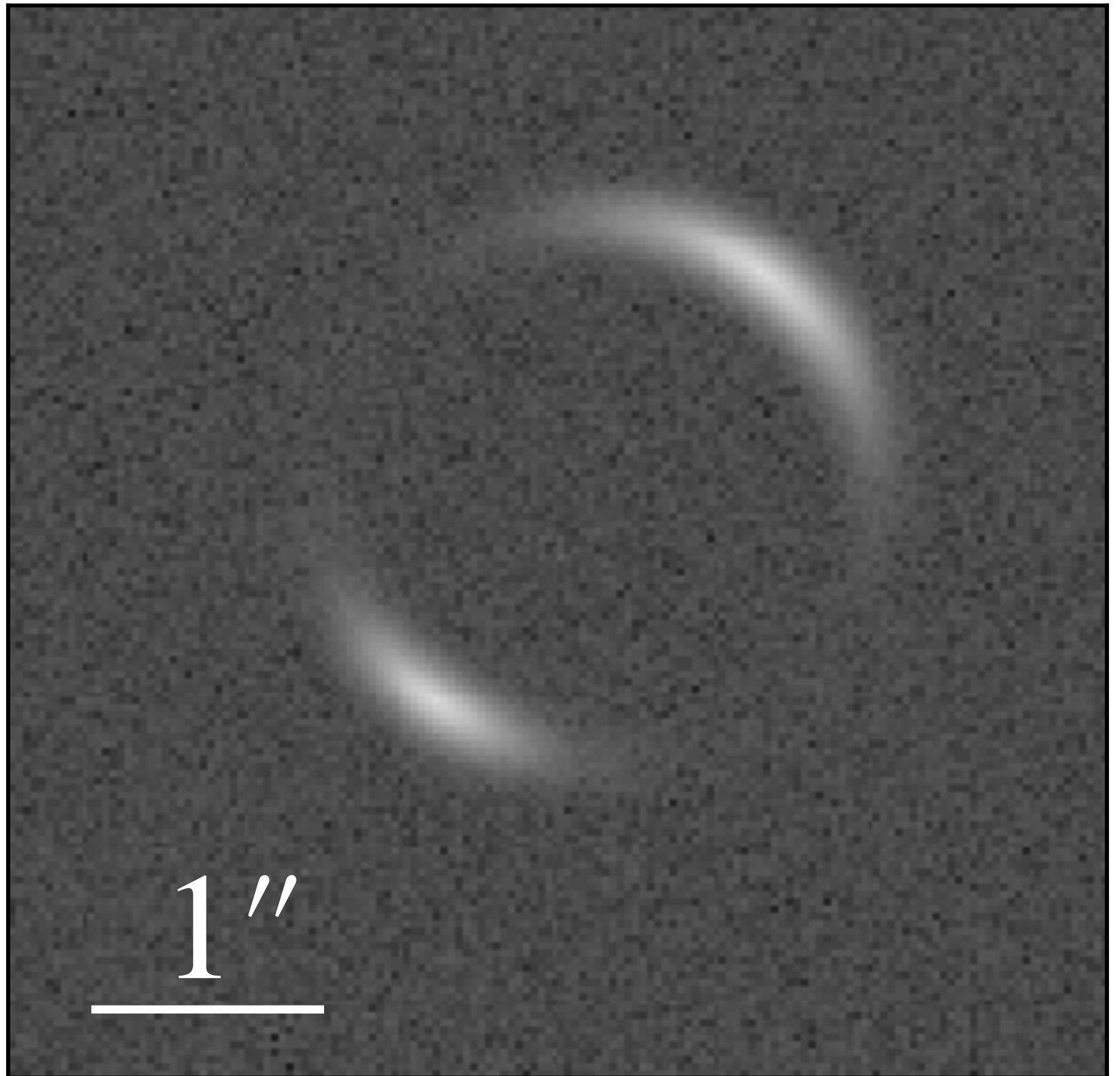


Extended arcs
from galaxies

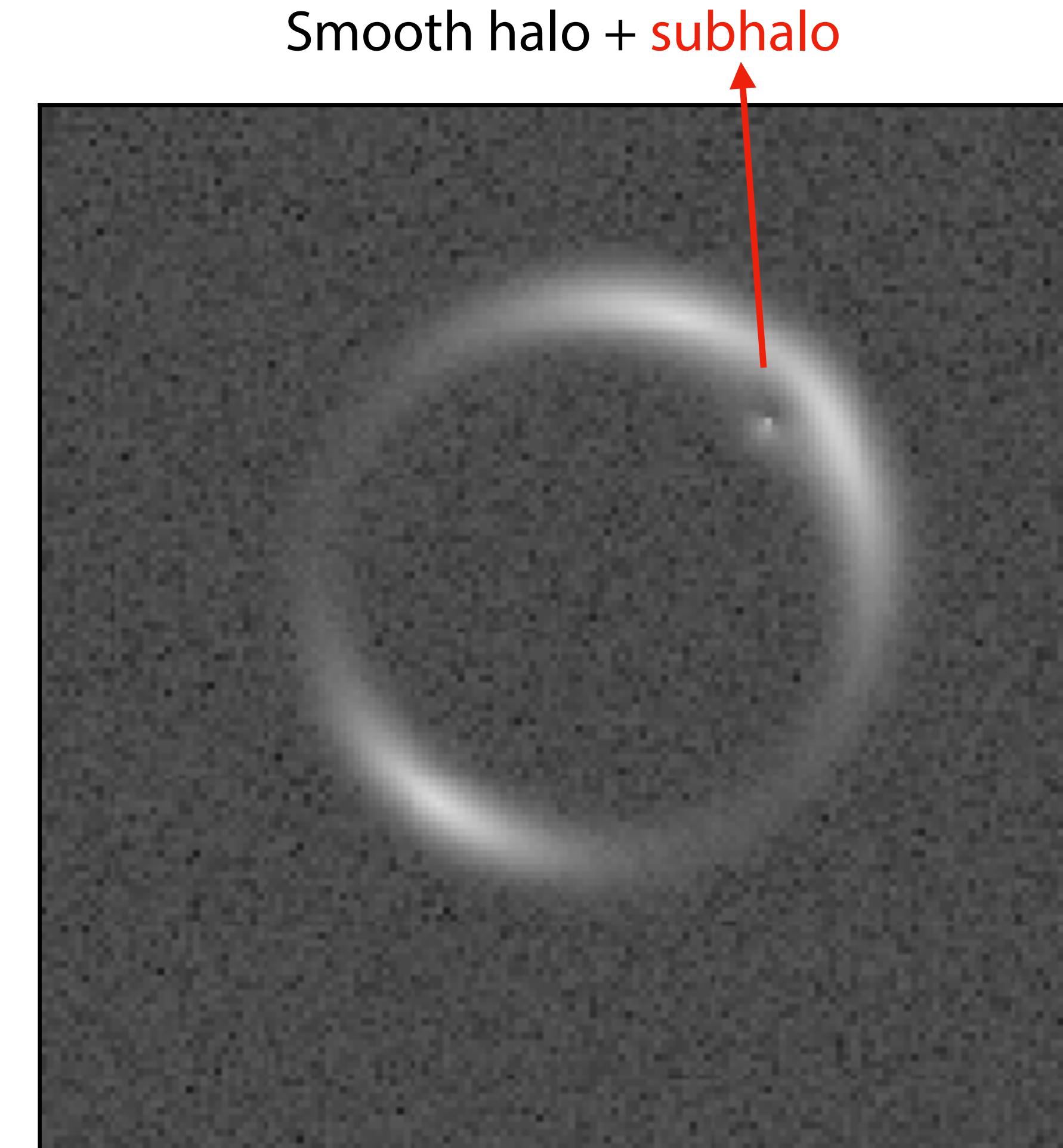
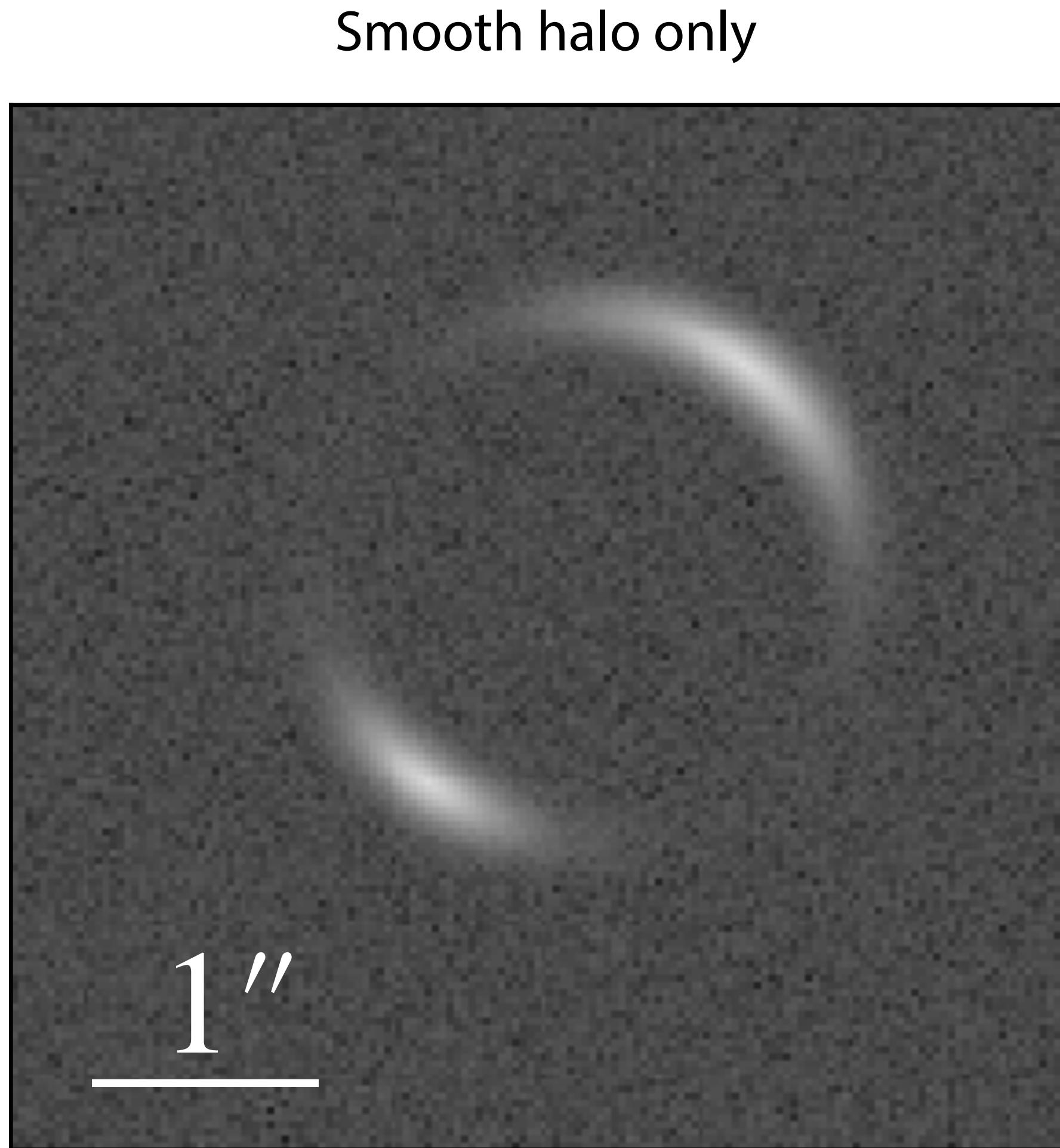


Subhalos affect strong lensing

Smooth halo only

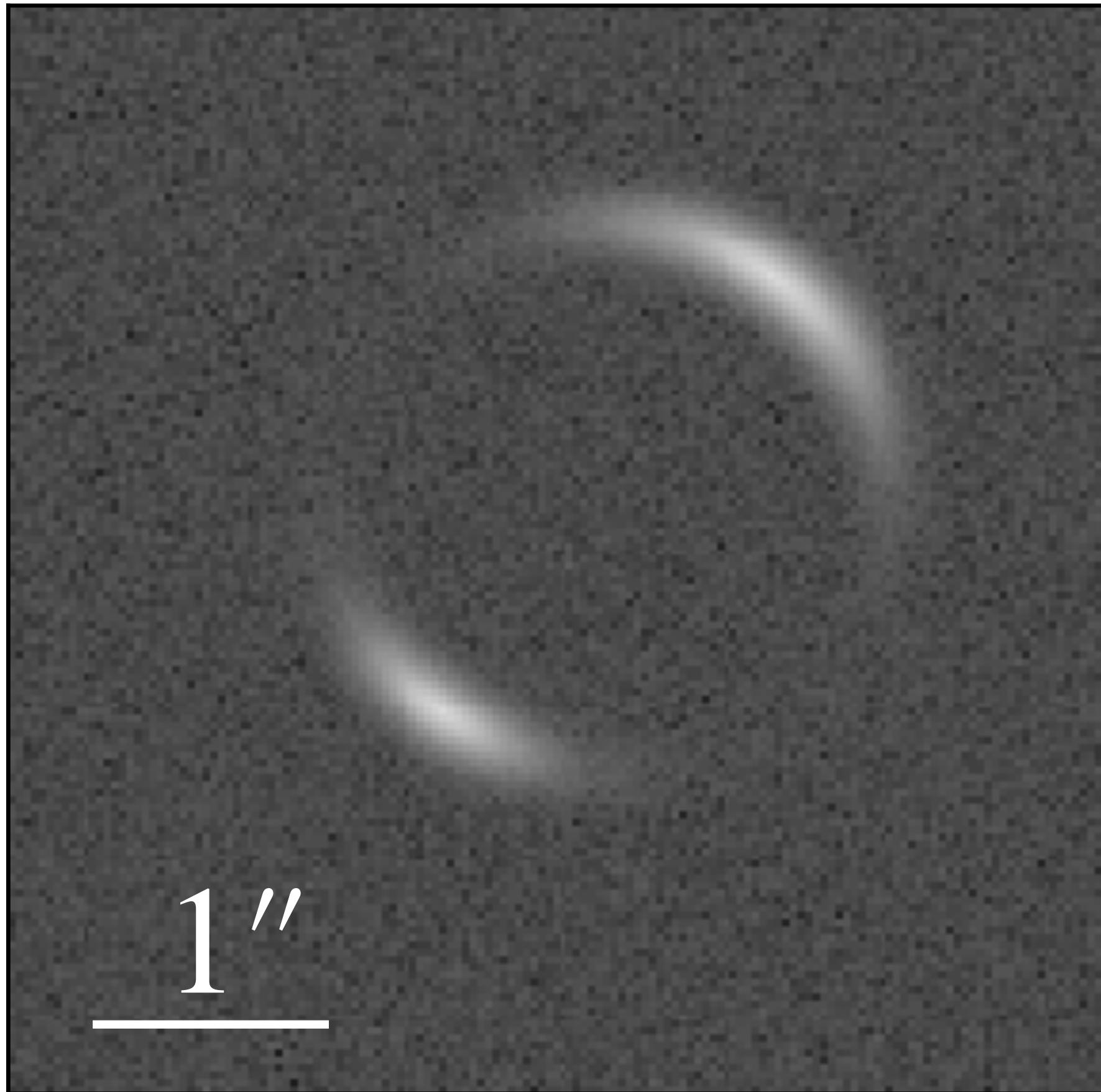


Subhalos affect strong lensing

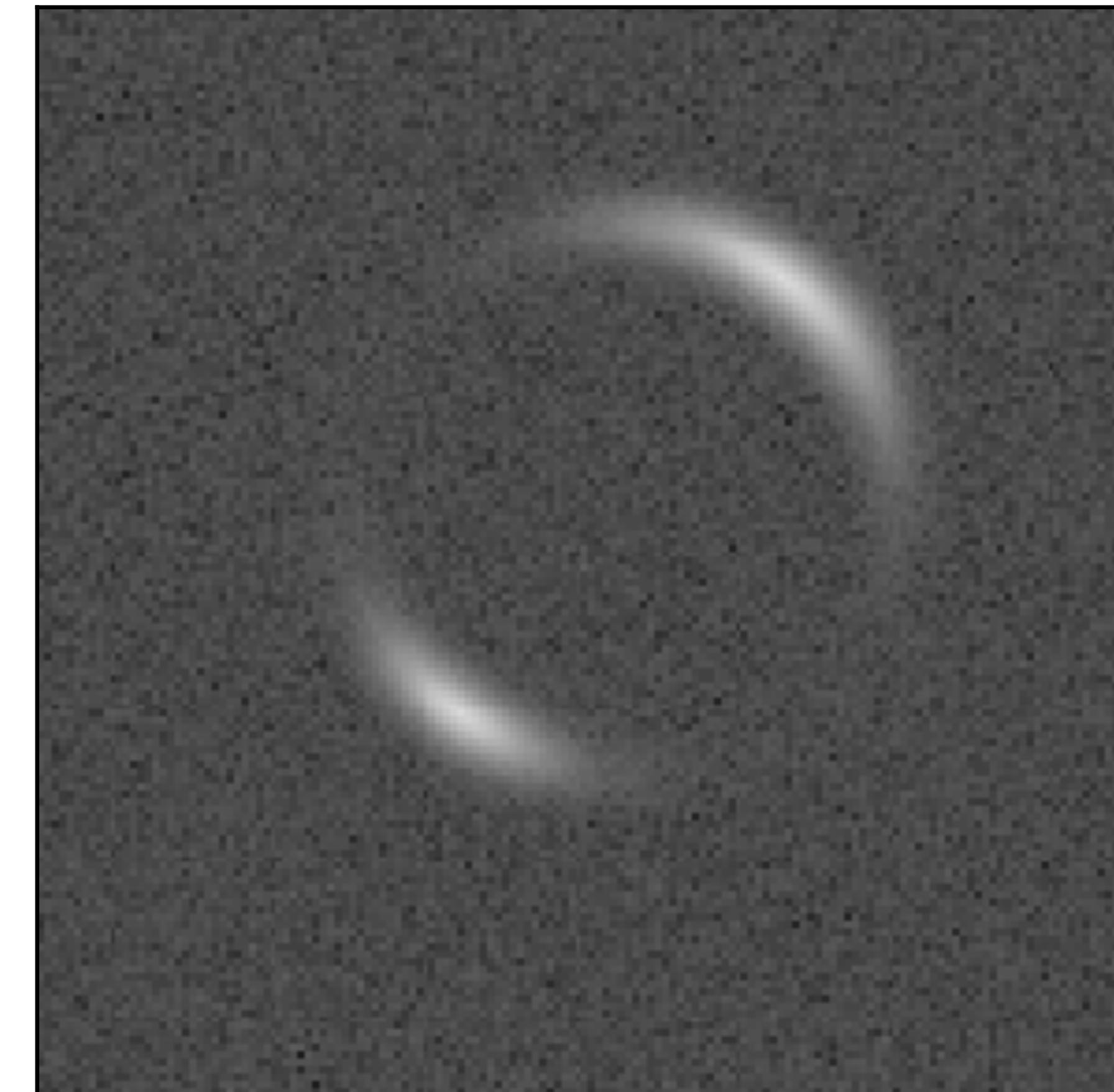


Subhalos affect strong lensing... realistically

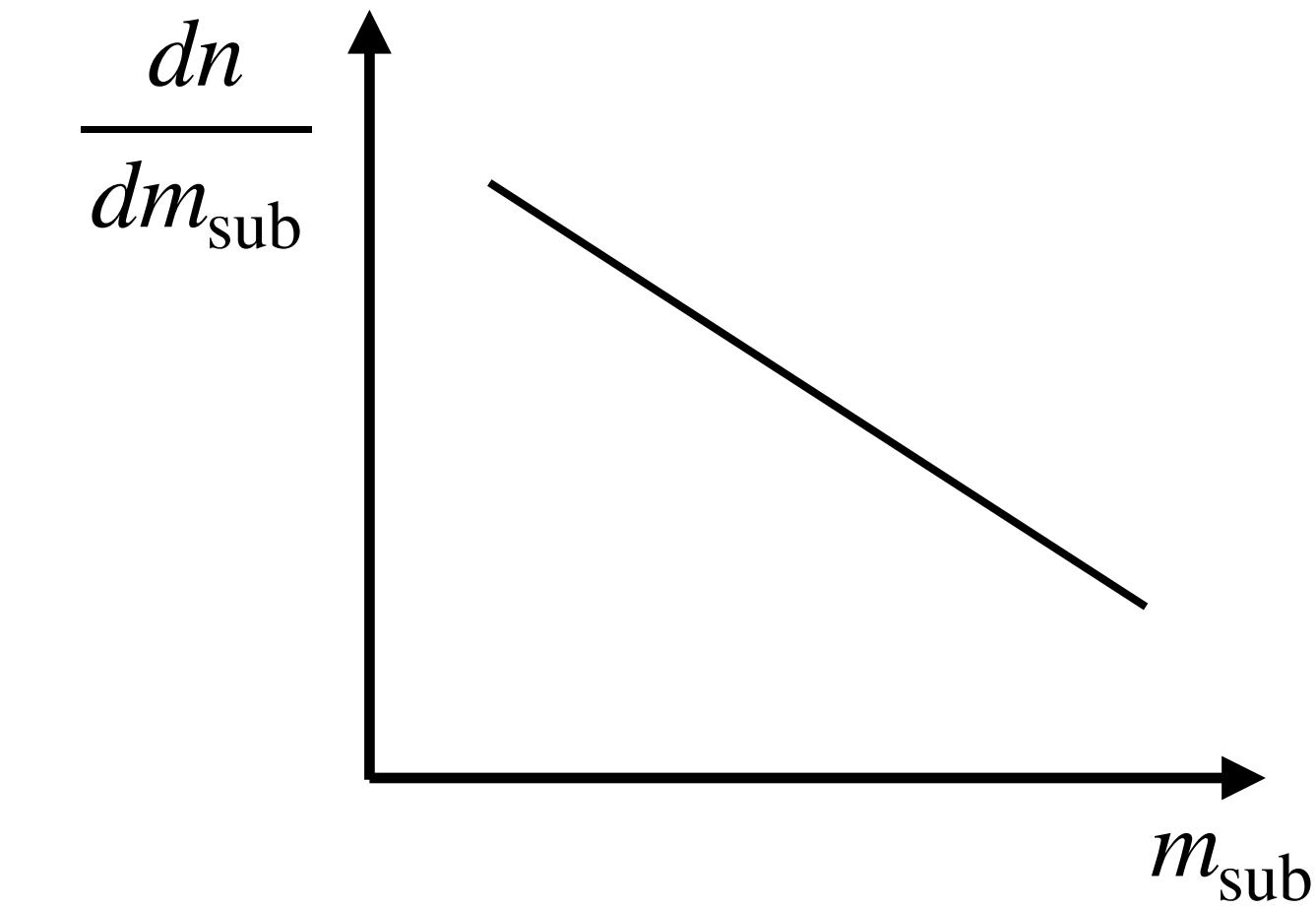
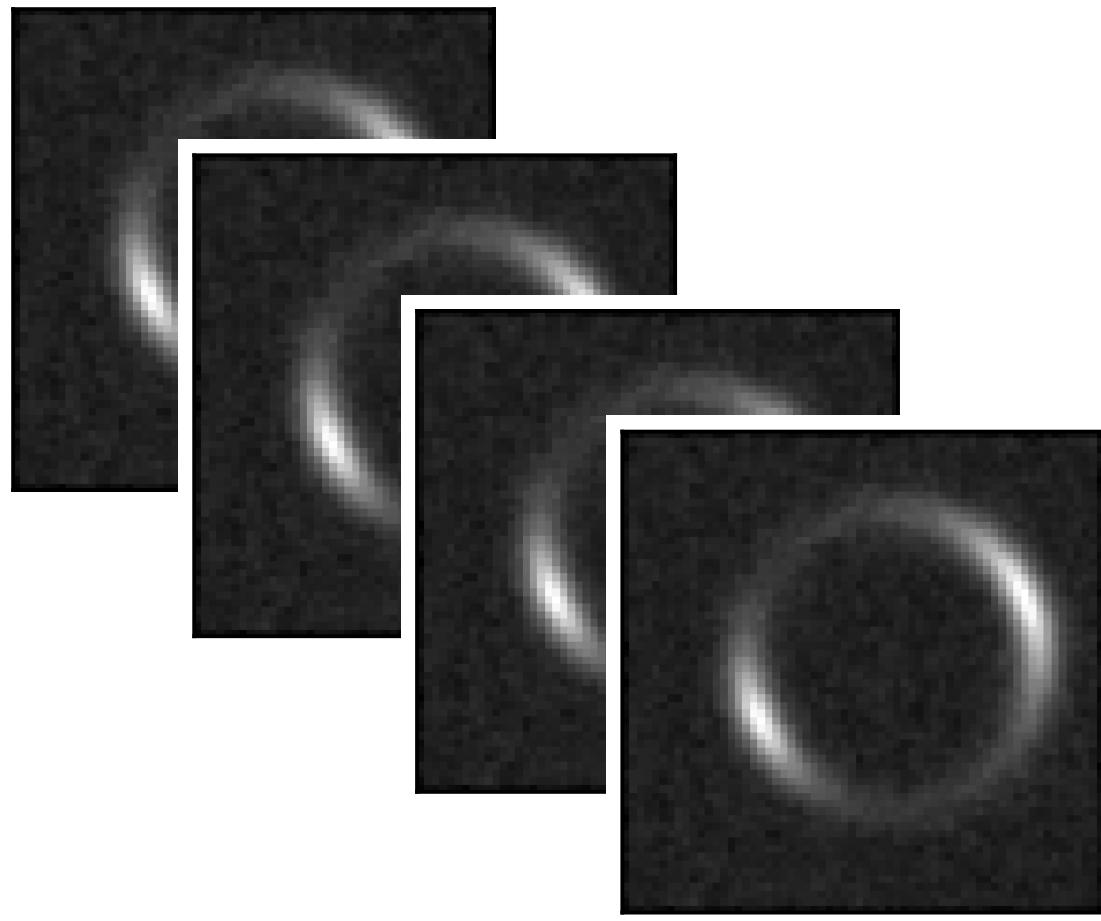
Smooth halo only



Smooth halo + subhalos



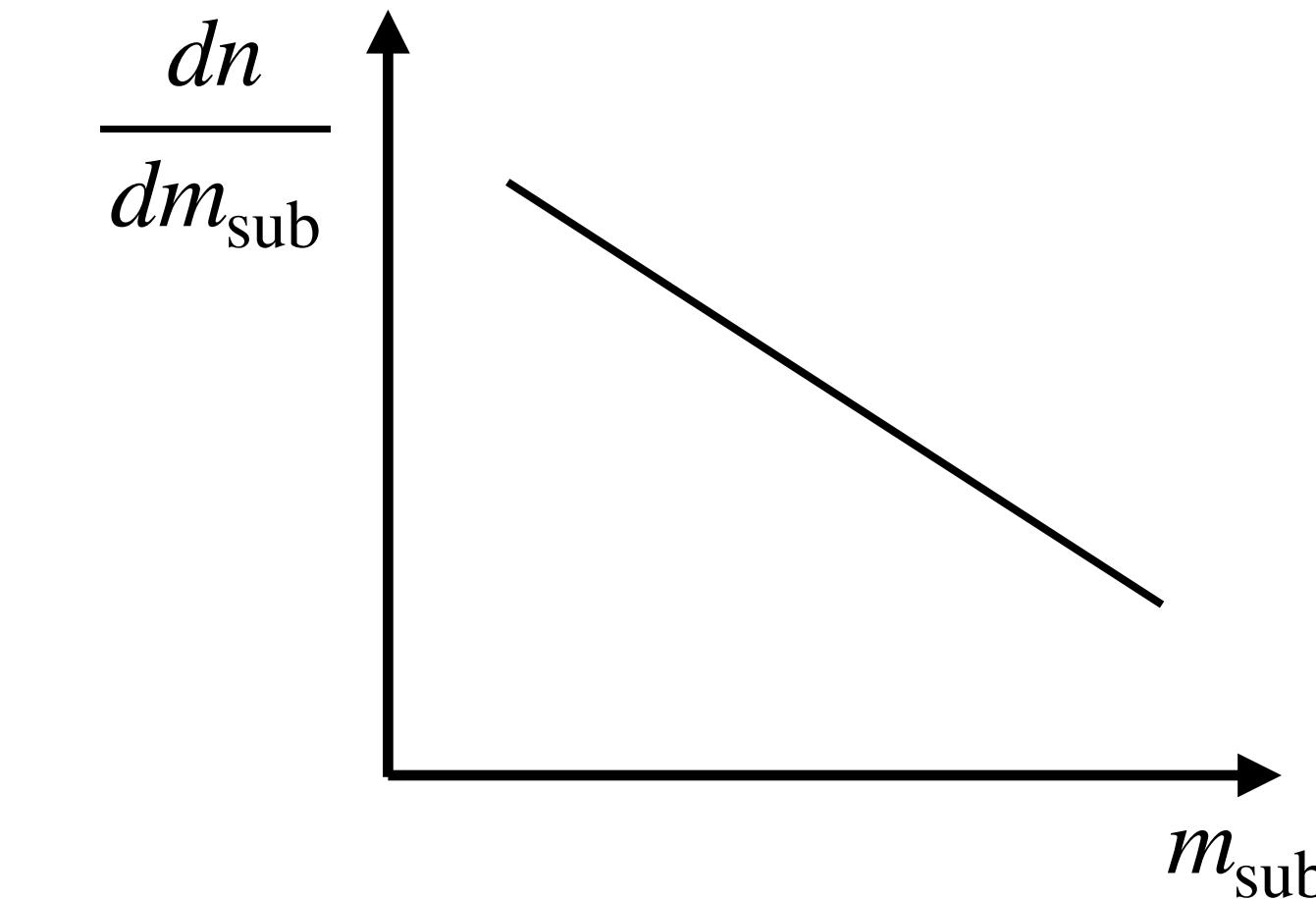
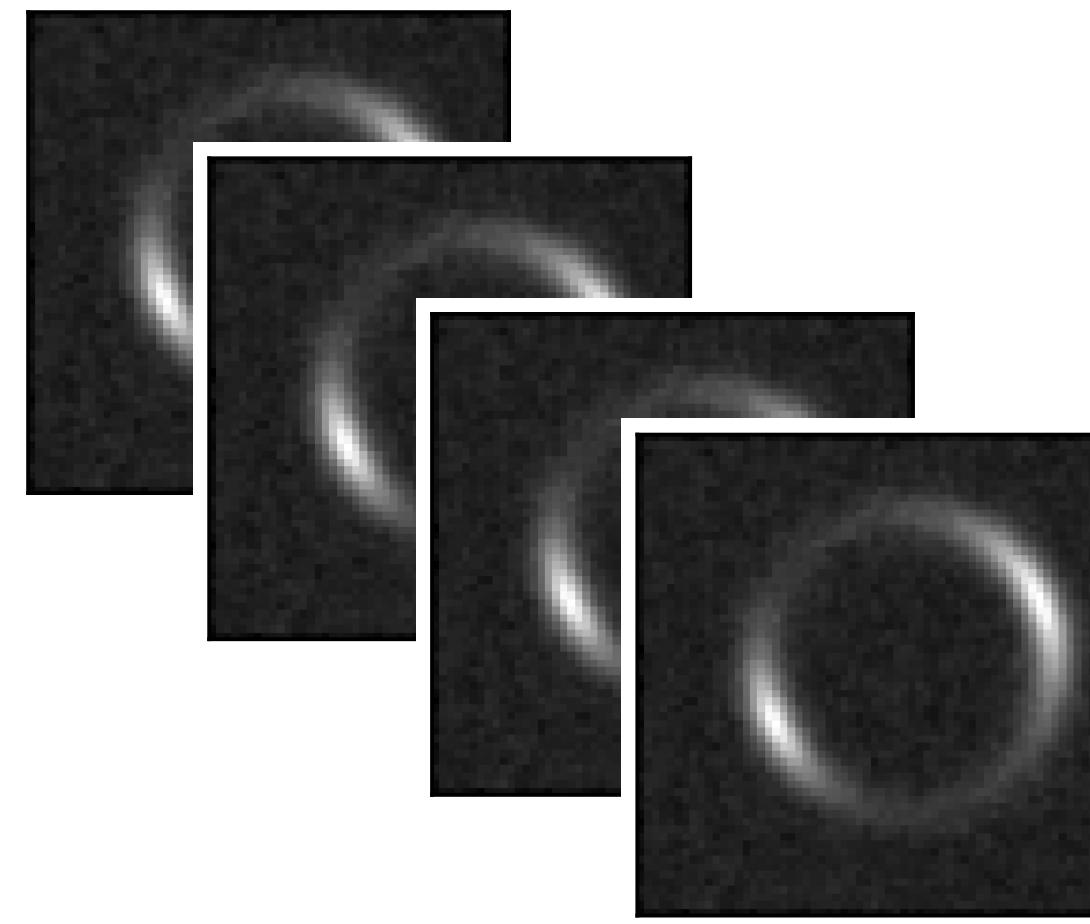
Scalable inference for small subhalos



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

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]

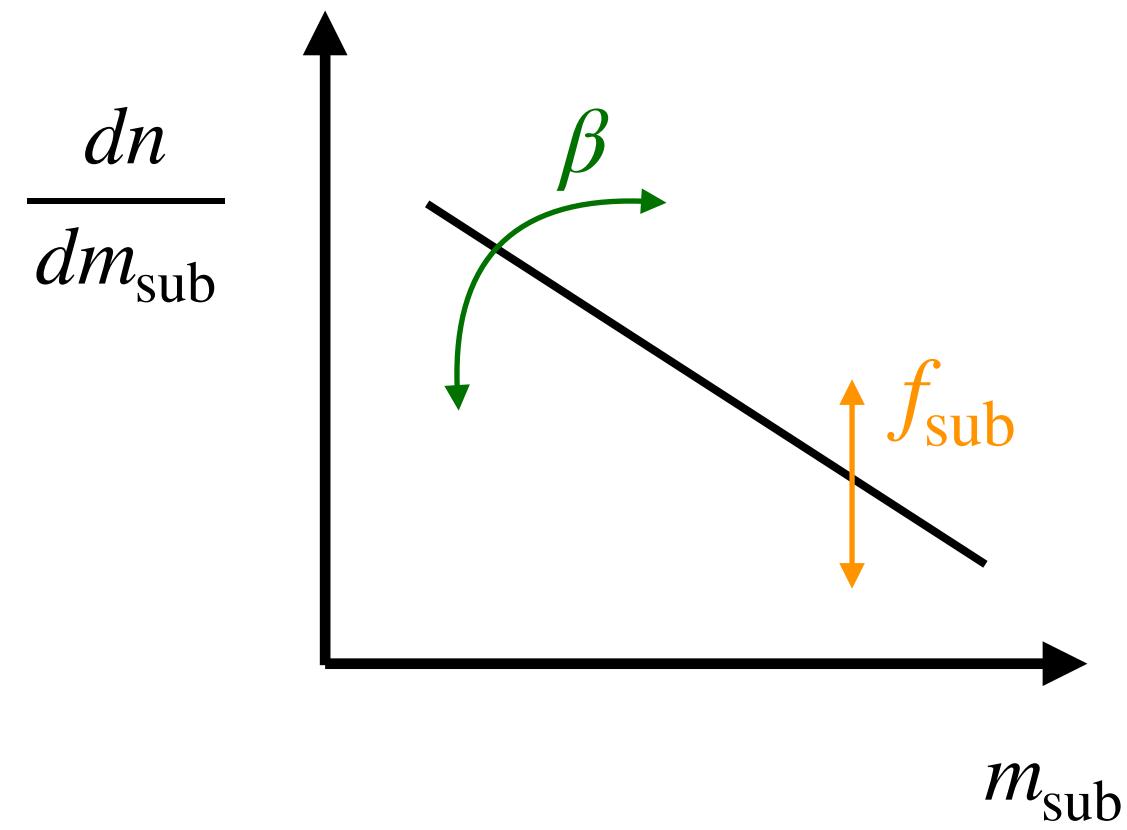
Goal: infer subhalo mass distribution through collective effects of many light subhalos

⇒ 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)

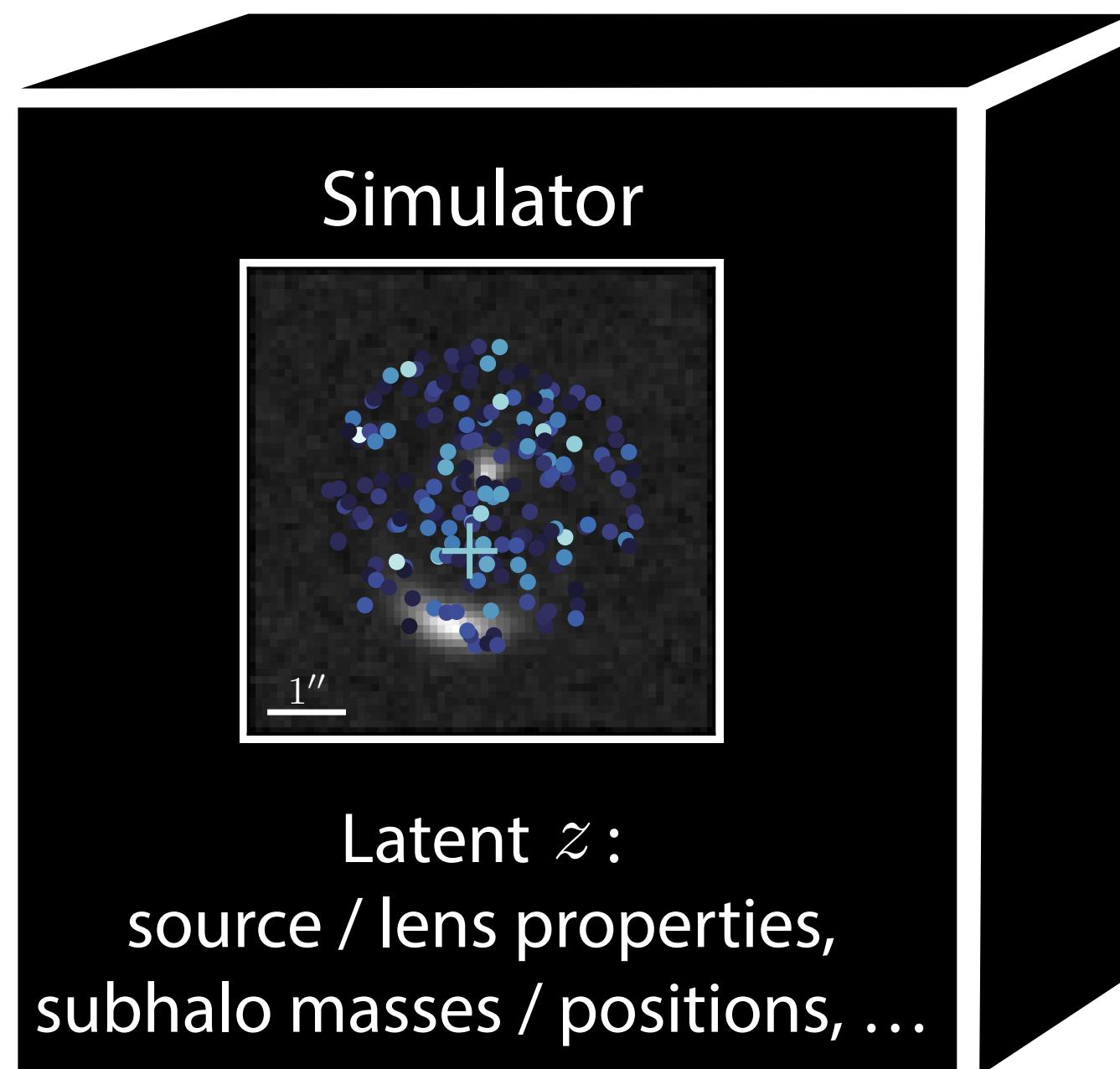
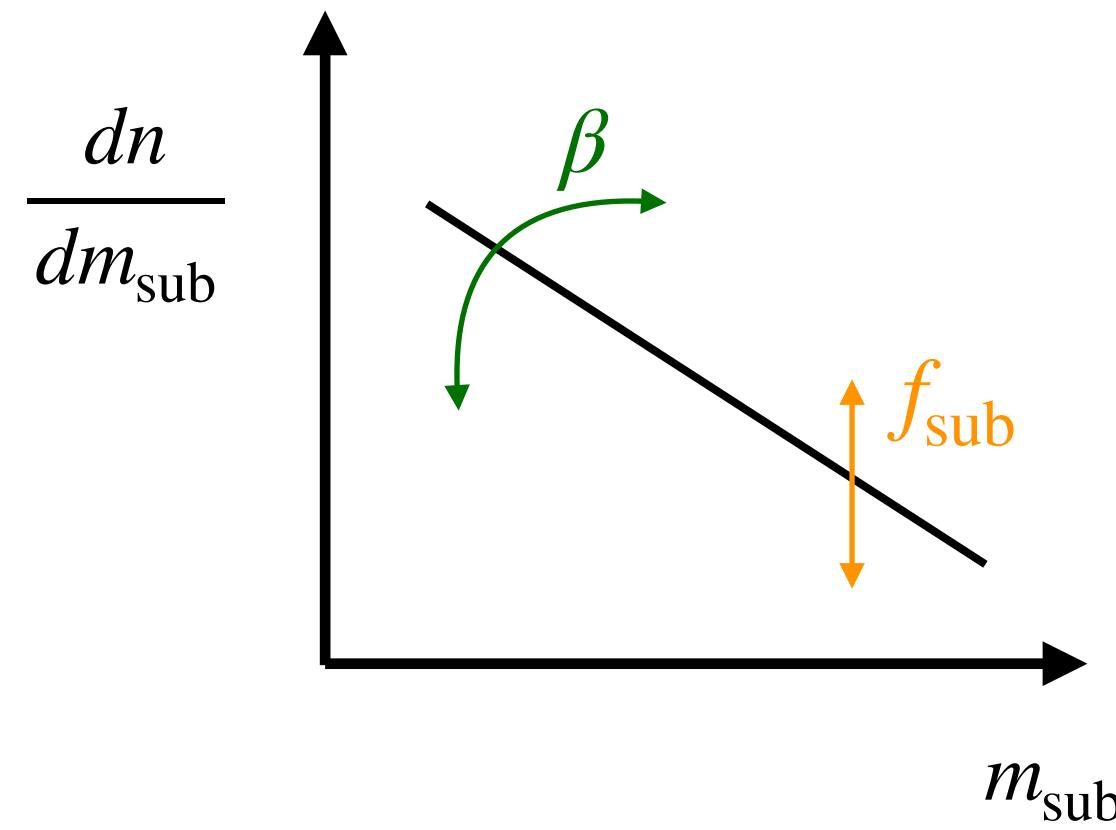
Simulation-based inference for strong lensing

2 parameters $\theta = (\beta, f_{\text{sub}})$



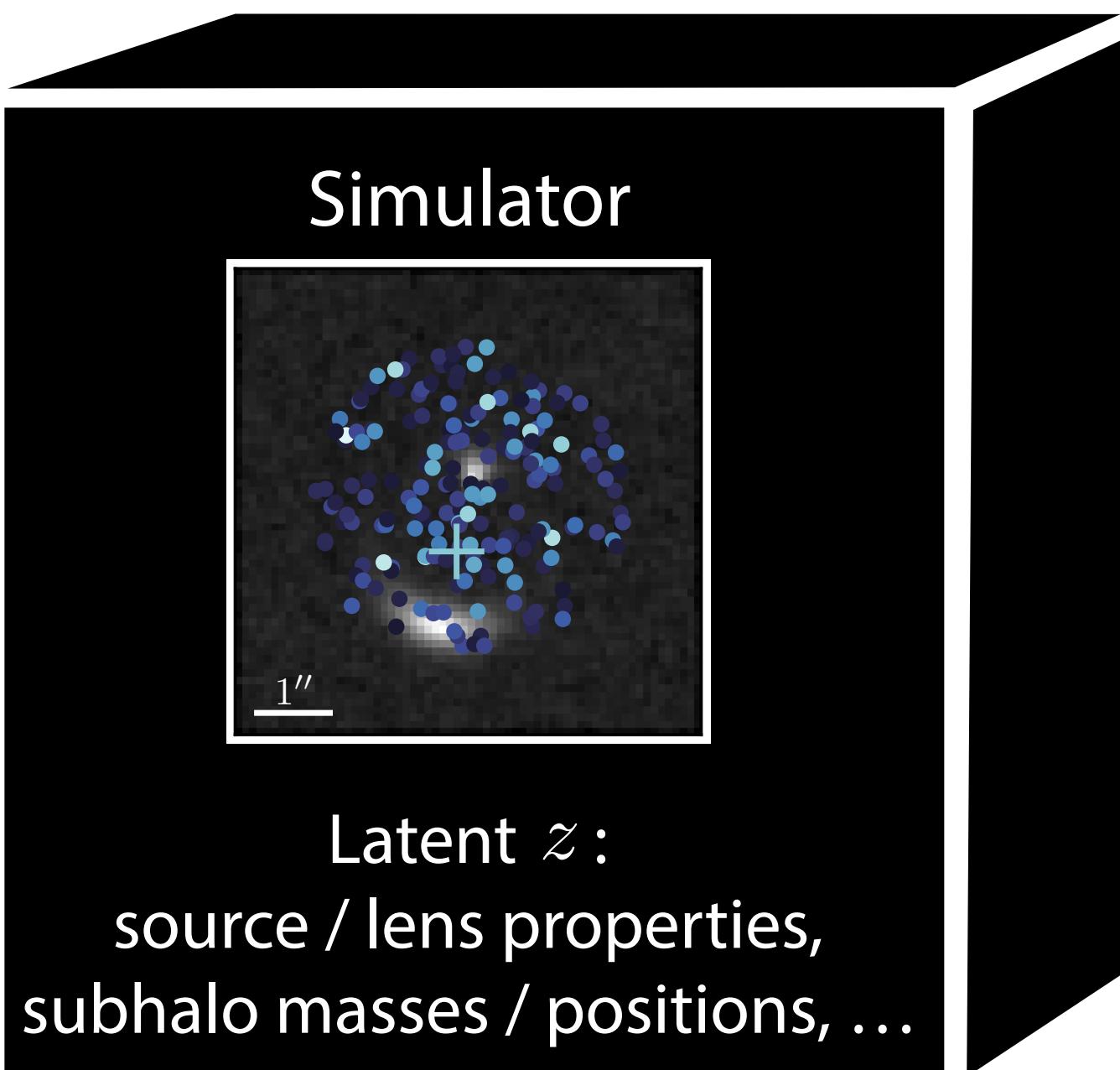
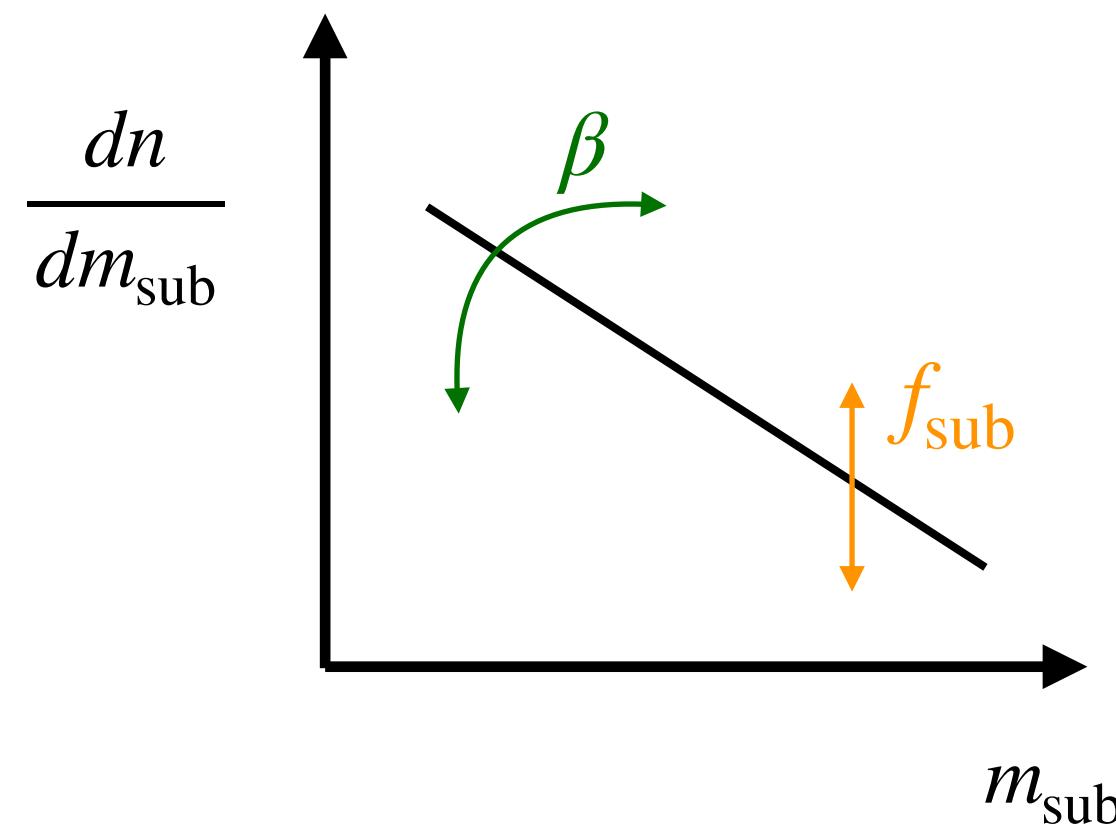
Simulation-based inference for strong lensing

2 parameters $\theta = (\beta, f_{\text{sub}})$

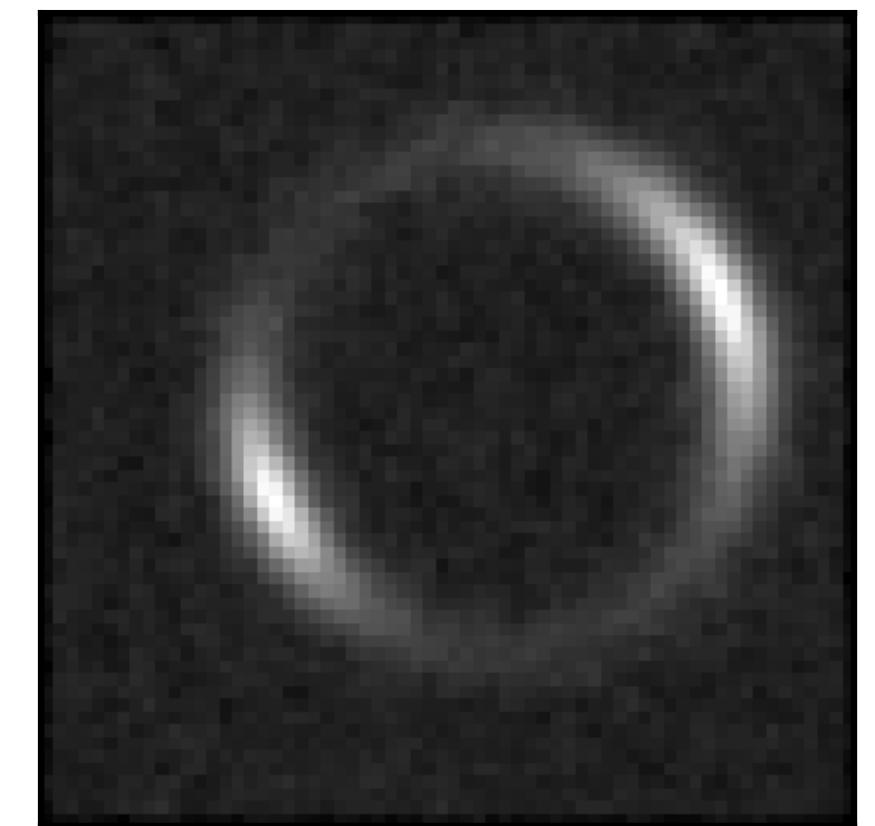


Simulation-based inference for strong lensing

2 parameters $\theta = (\beta, f_{\text{sub}})$

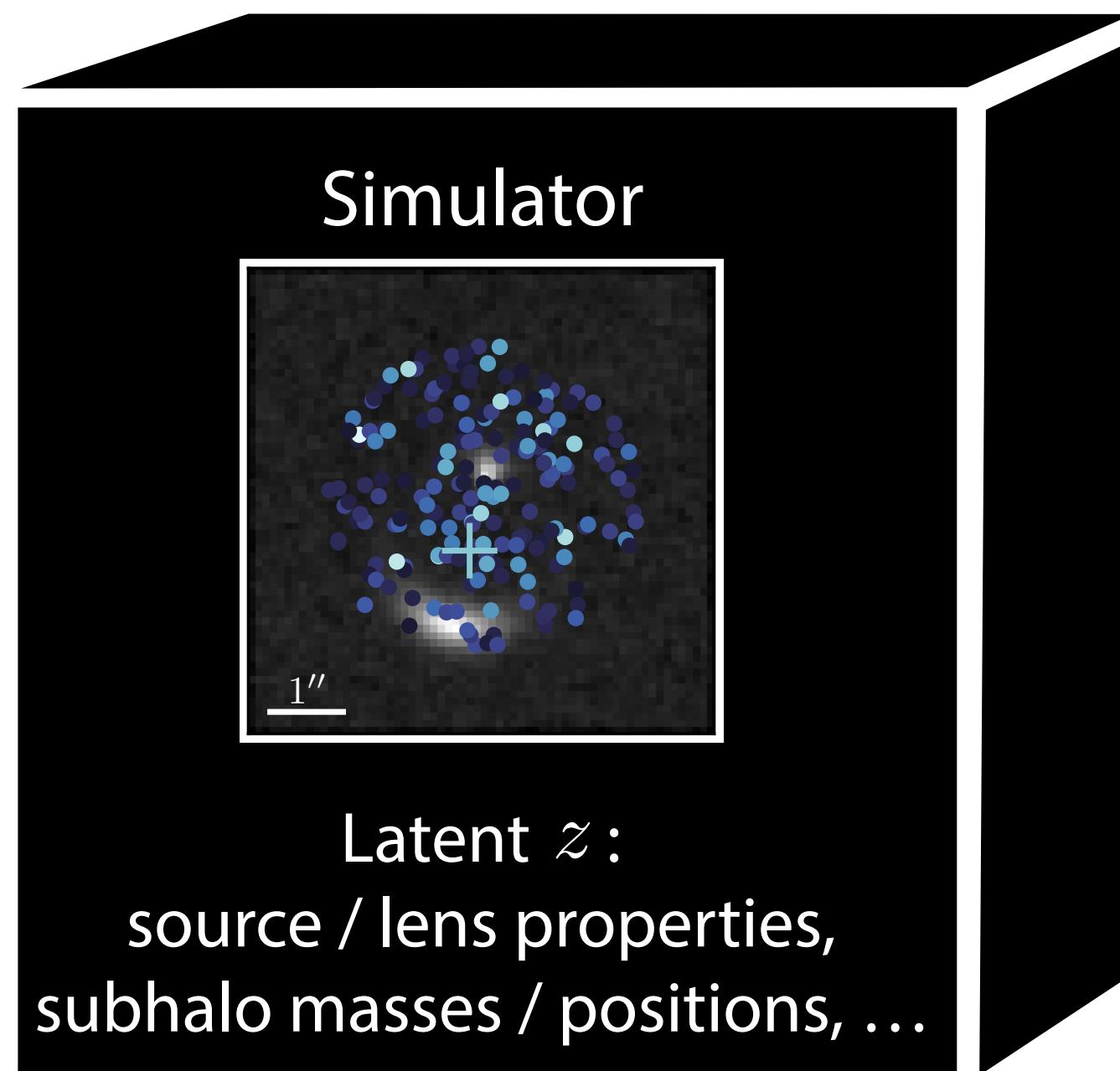
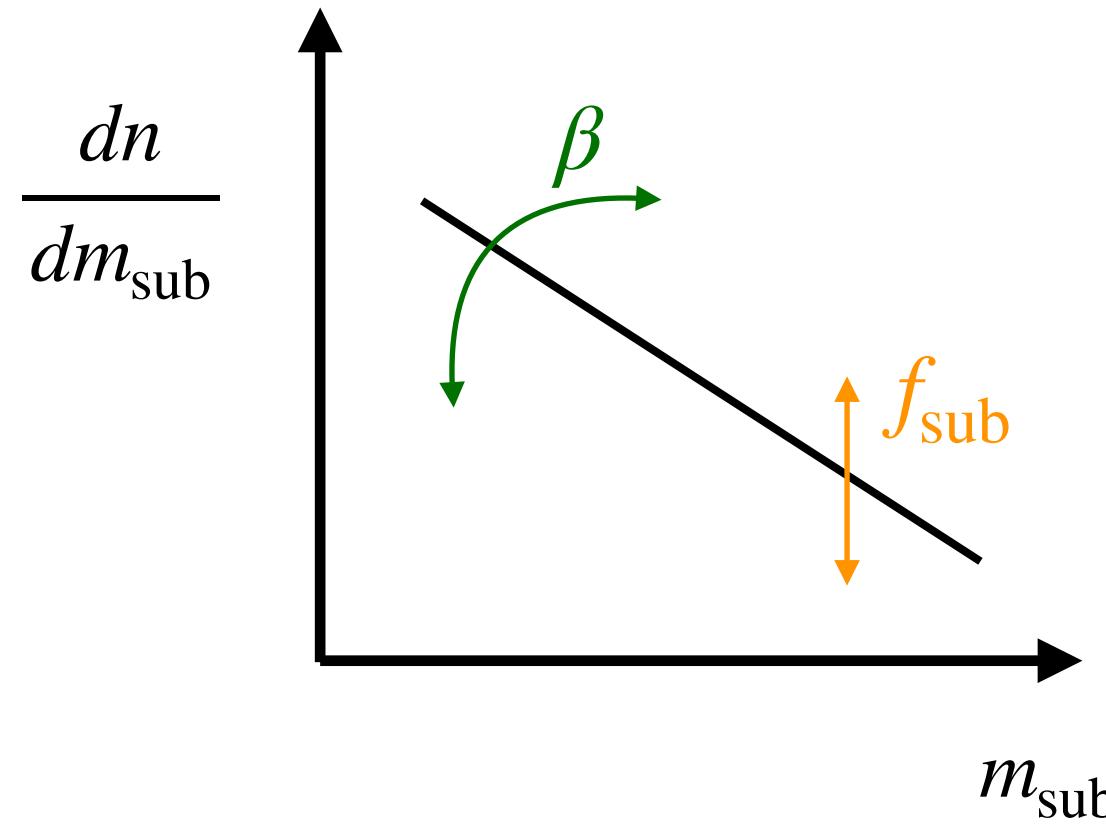


64² observables x



Simulation-based inference for strong lensing

2 parameters $\theta = (\beta, f_{\text{sub}})$



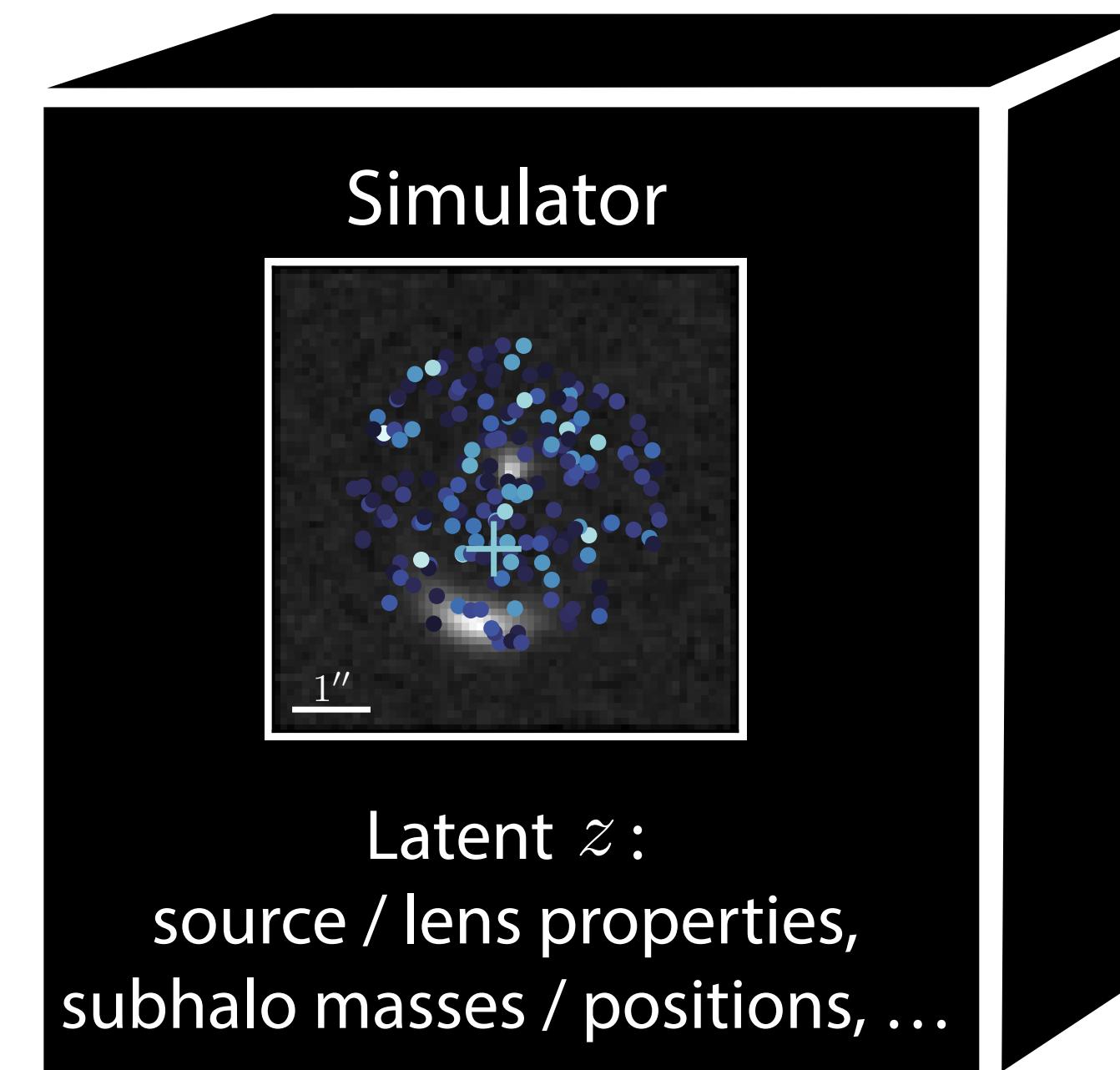
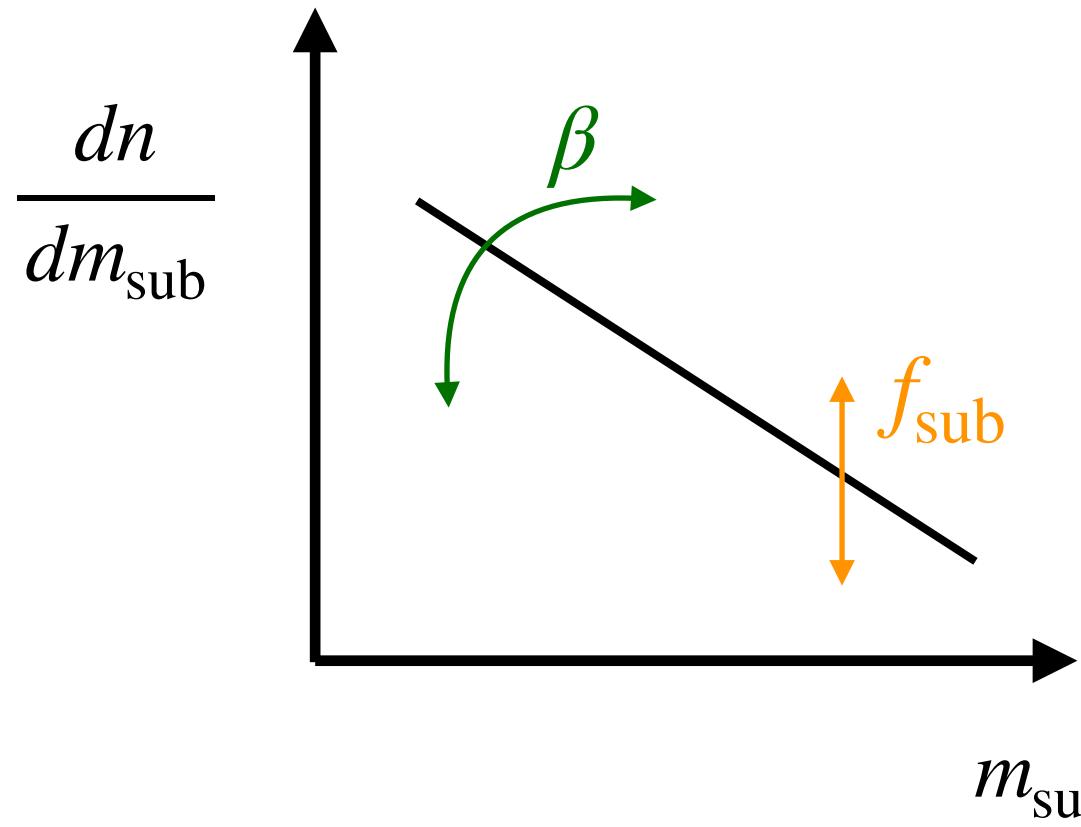
64² observables x



Prediction: We construct a simulator that can sample $x \sim p(x|\theta)$

Simulation-based inference for strong lensing

2 parameters $\theta = (\beta, f_{\text{sub}})$

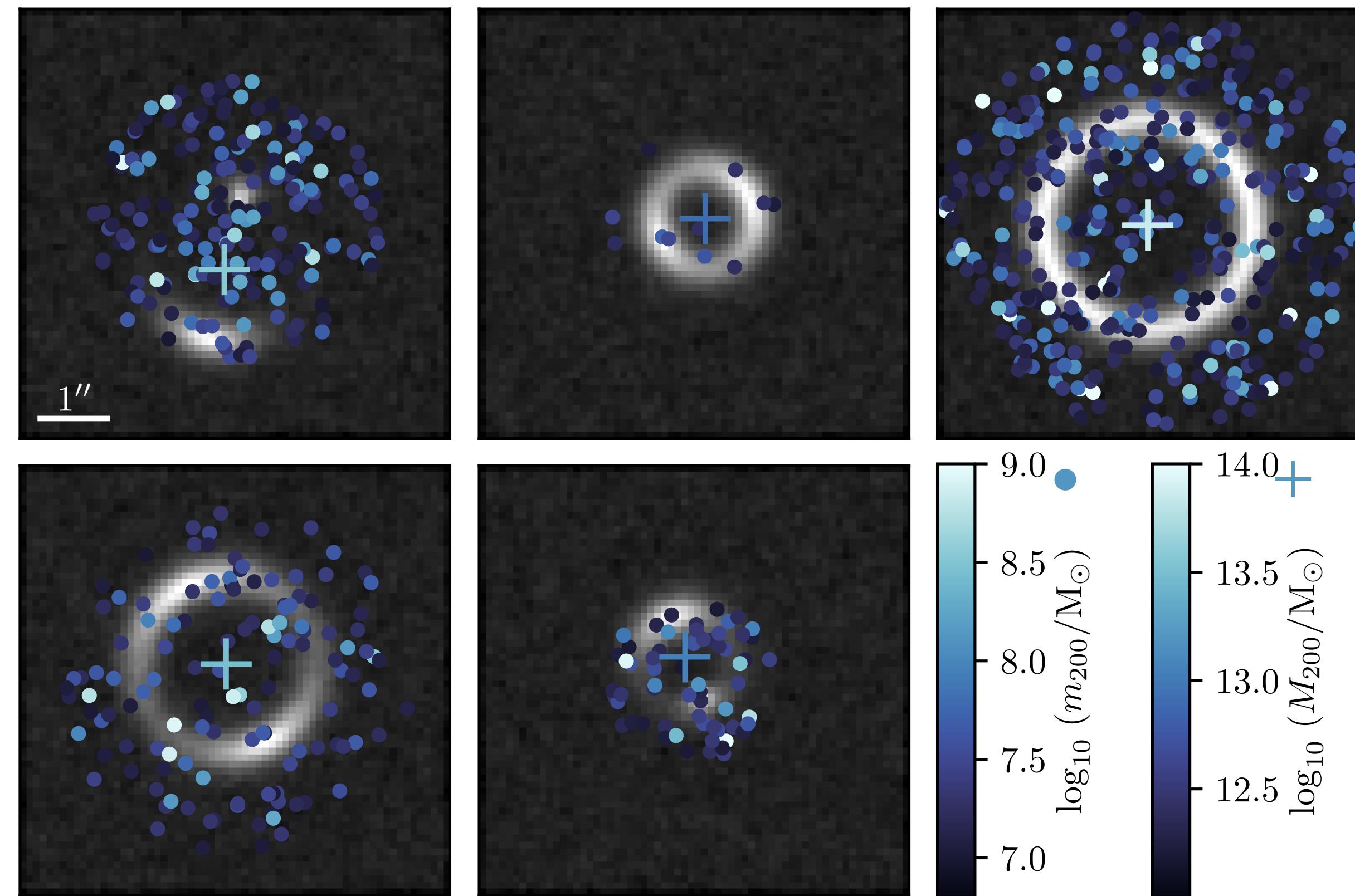


Prediction: We construct a simulator that can sample $x \sim p(x|\theta)$

Inference: We train neural likelihood ratio estimators $\hat{r}(x|\theta)$

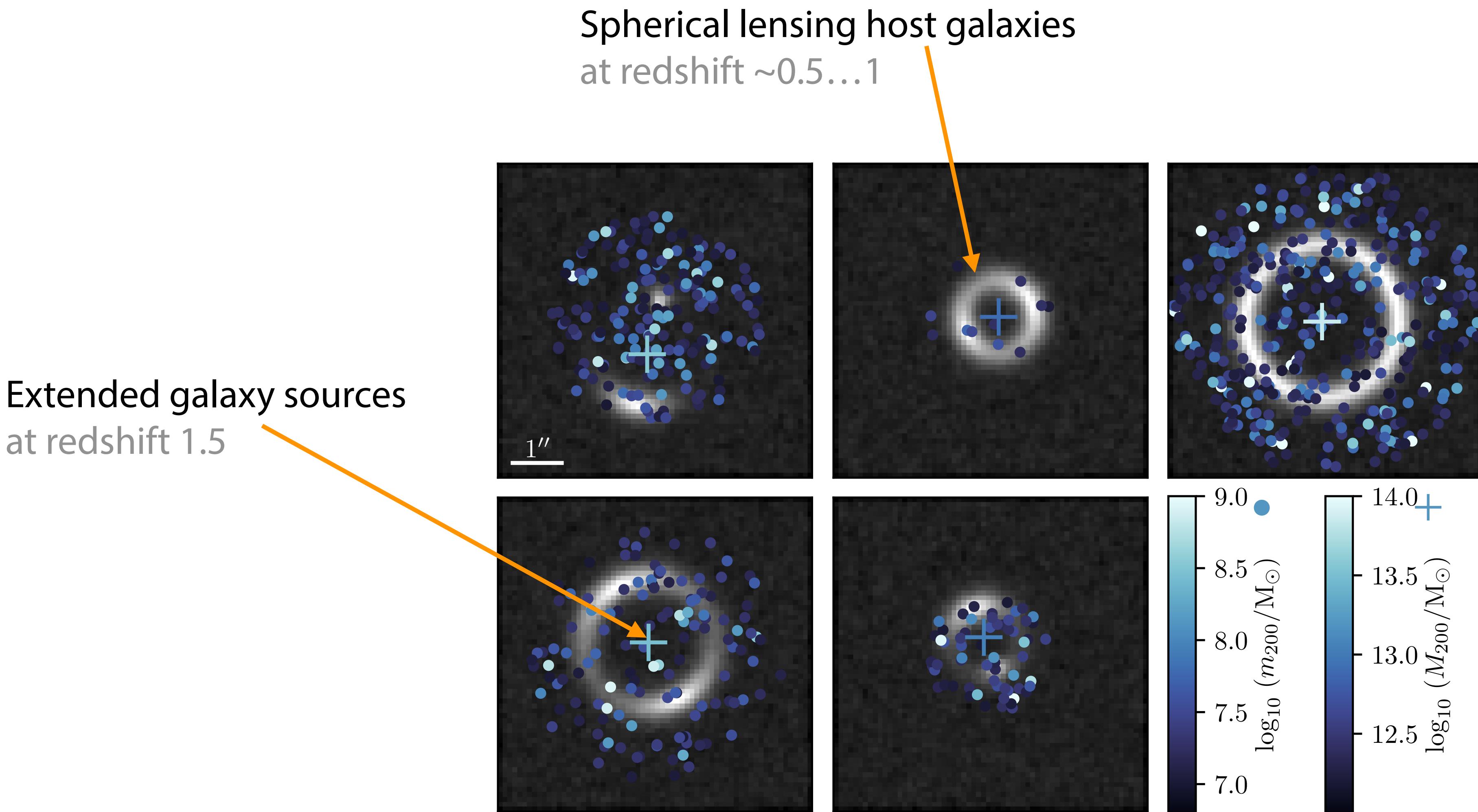
Proof-of-principle simulator

[following T. Collett 1507.02657]



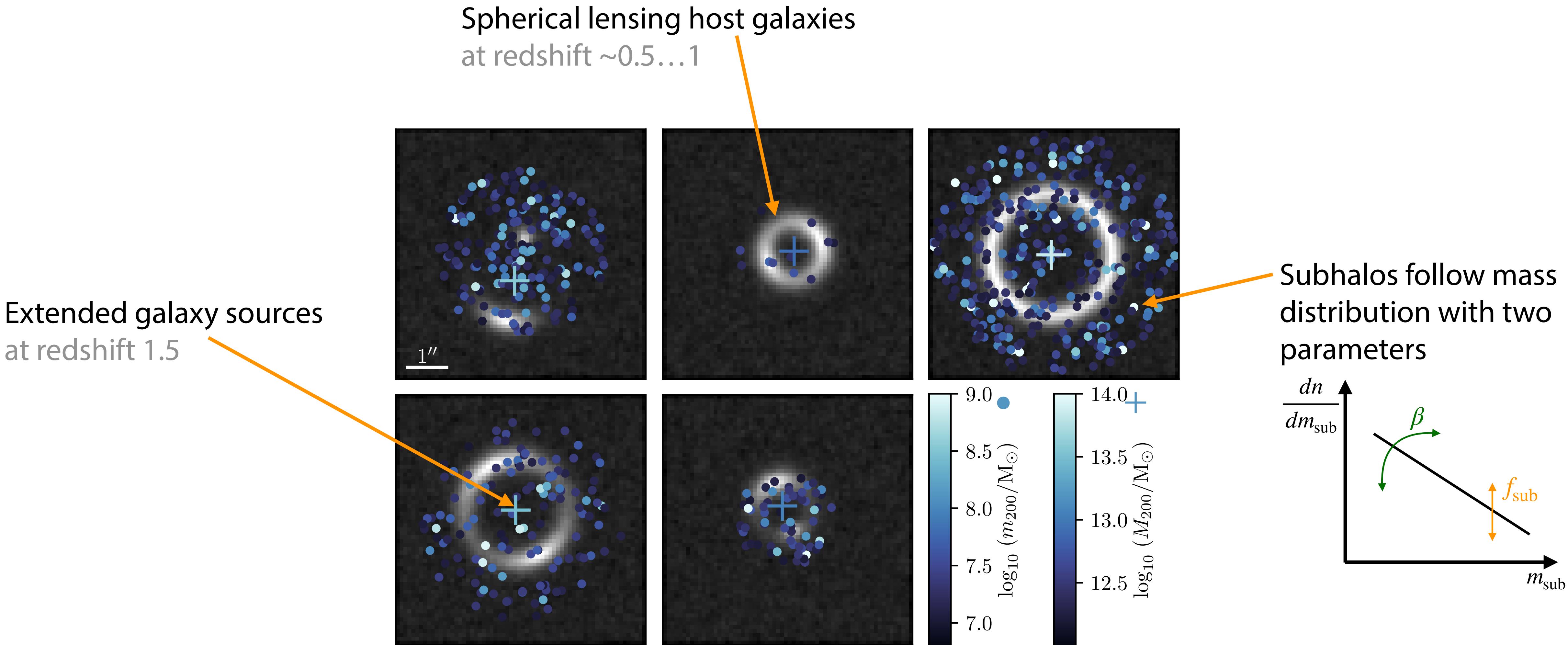
Proof-of-principle simulator

[following T. Collett 1507.02657]



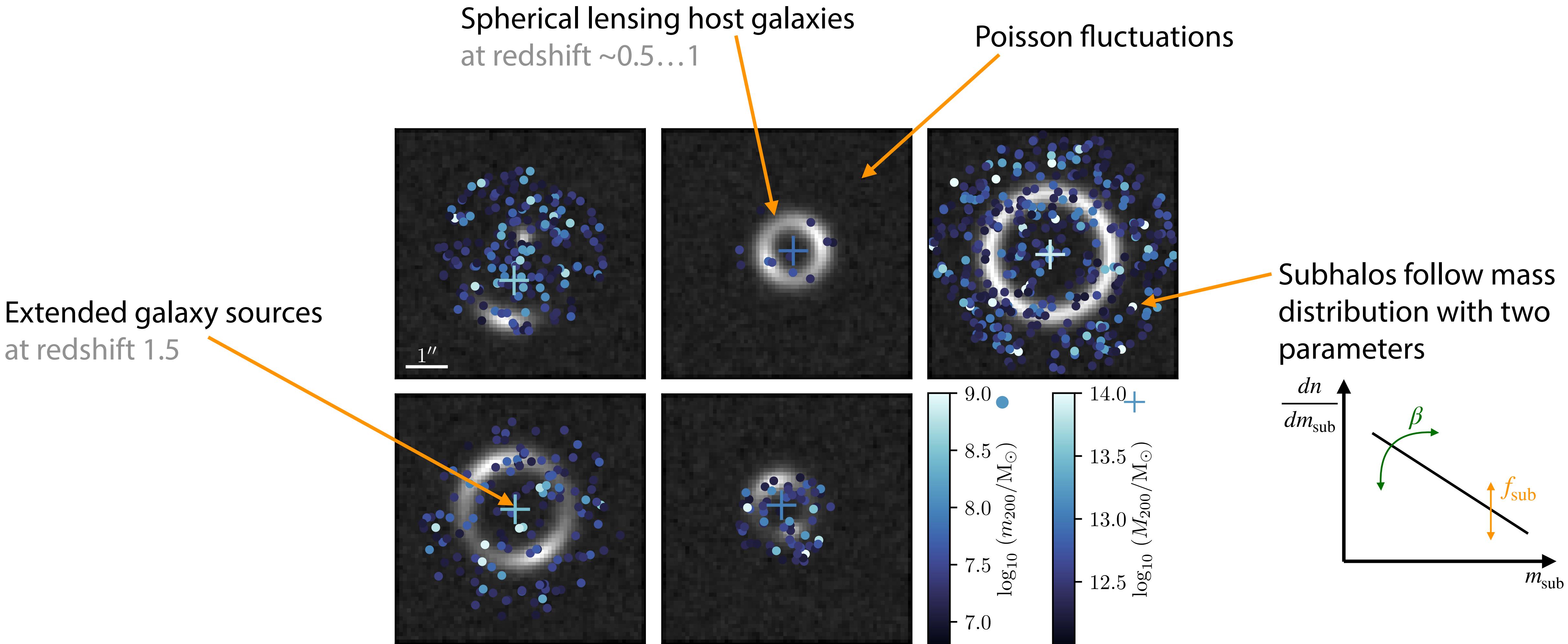
Proof-of-principle simulator

[following T. Collett 1507.02657]



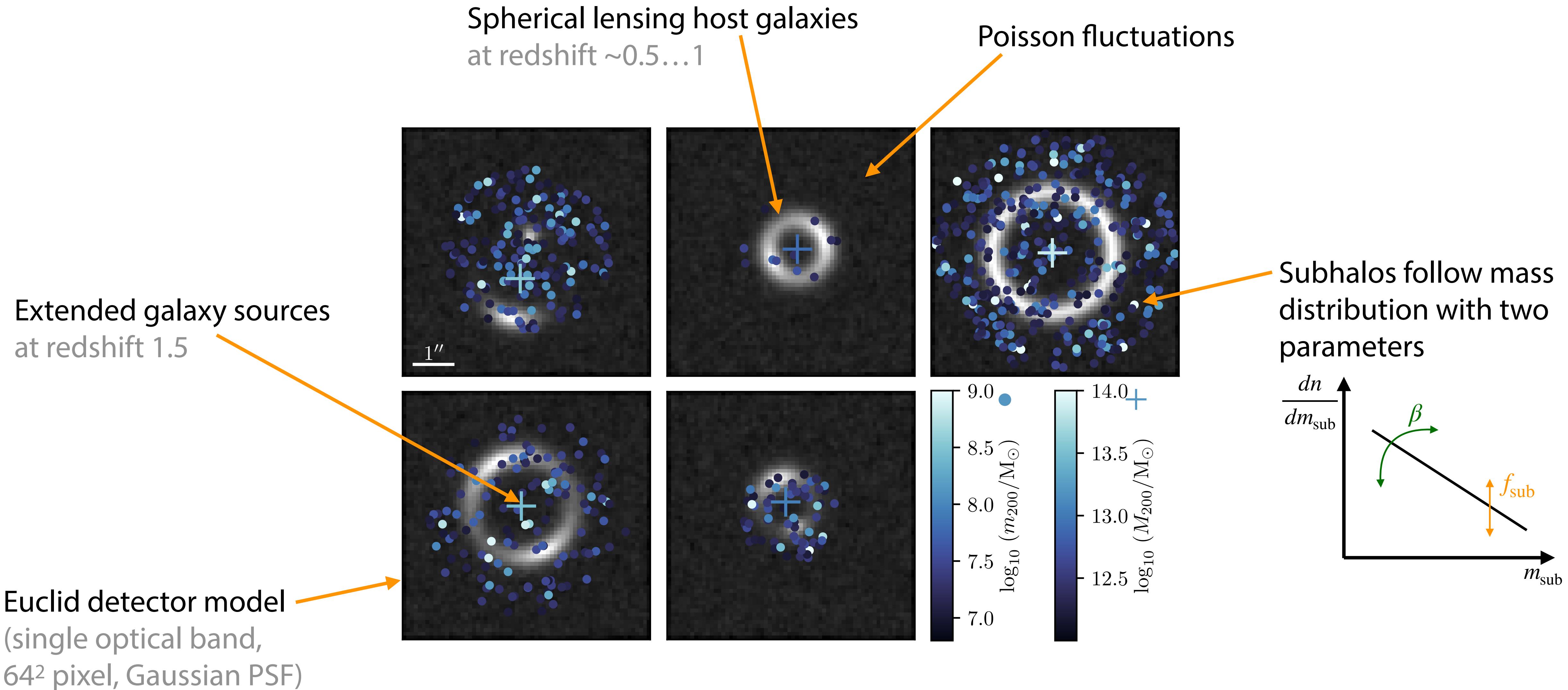
Proof-of-principle simulator

[following T. Collett 1507.02657]

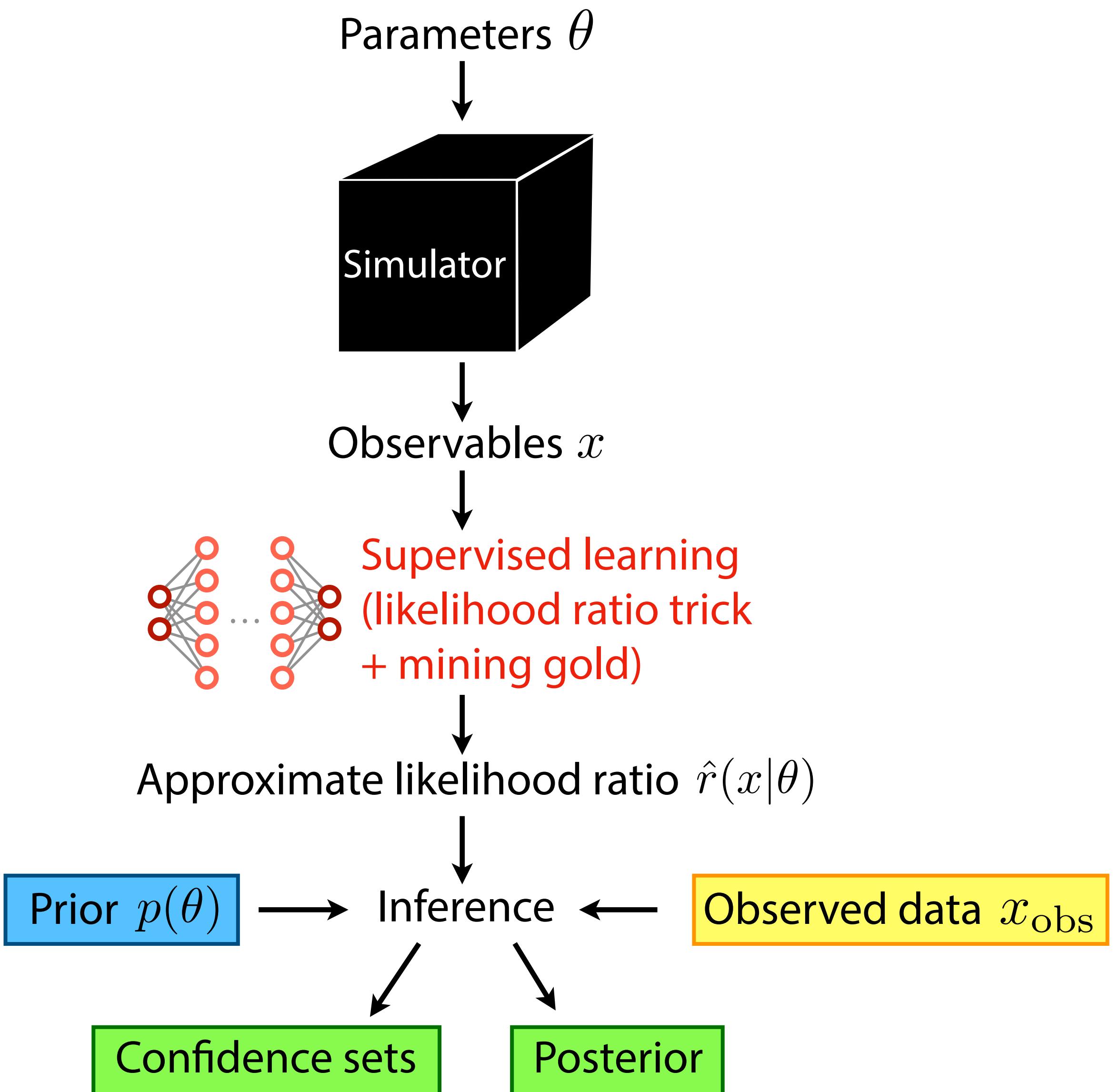


Proof-of-principle simulator

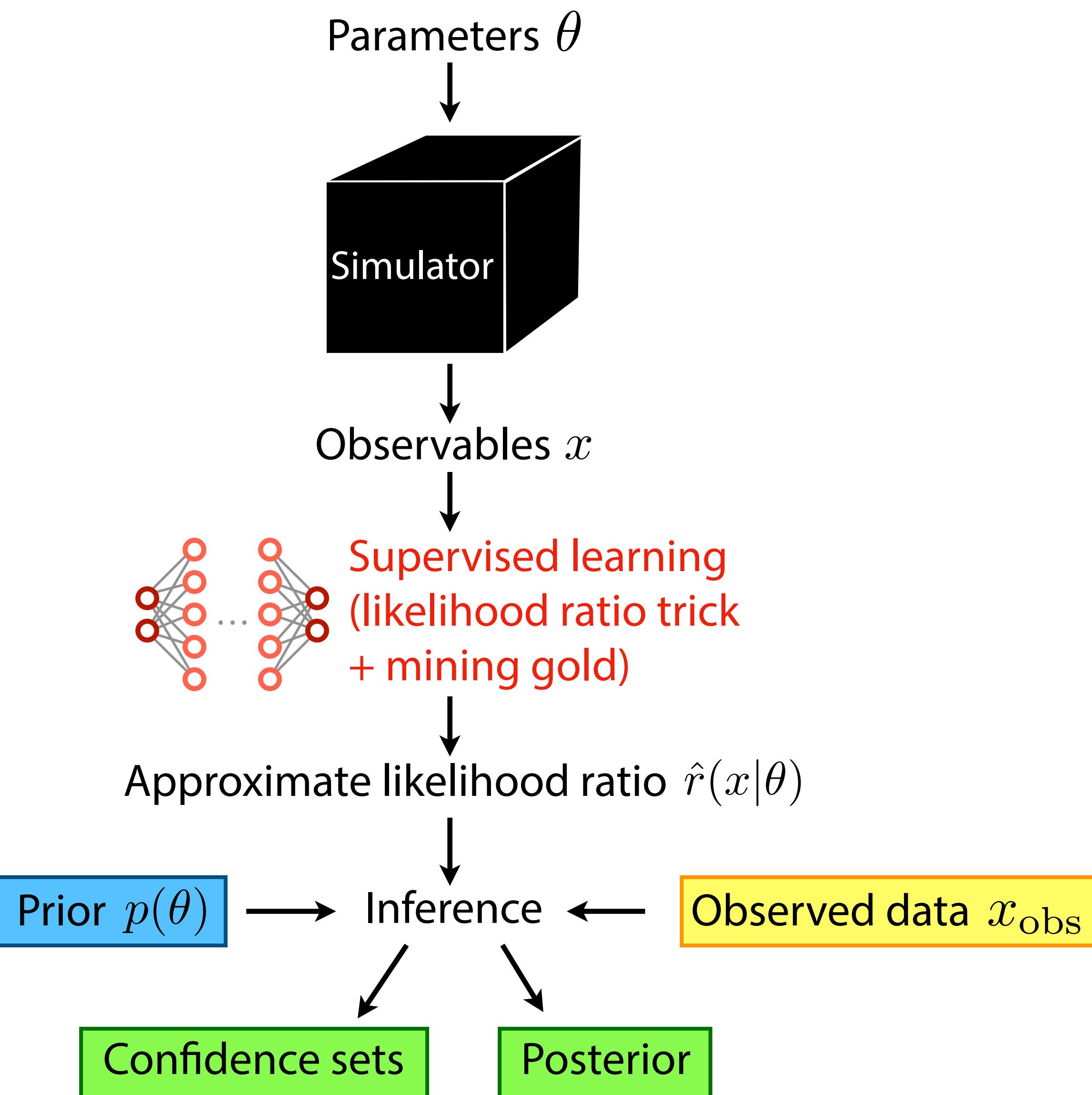
[following T. Collett 1507.02657]



Inference setup

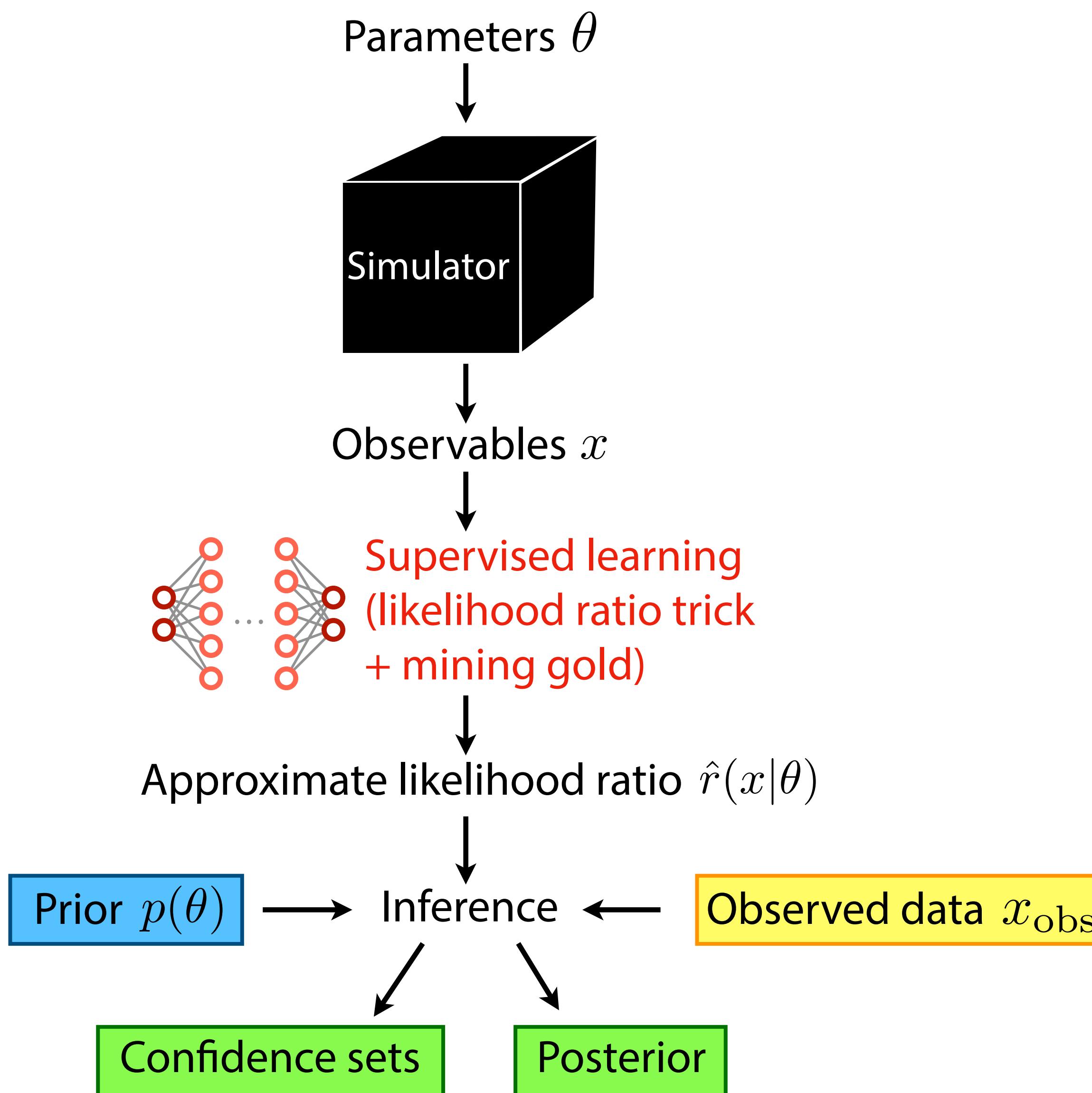


Inference setup



Training data: 10^6 lensed images with
 $0 \leq f_{\text{sub}} \leq 0.2, -1.5 \leq \beta \leq -0.5$

Inference setup

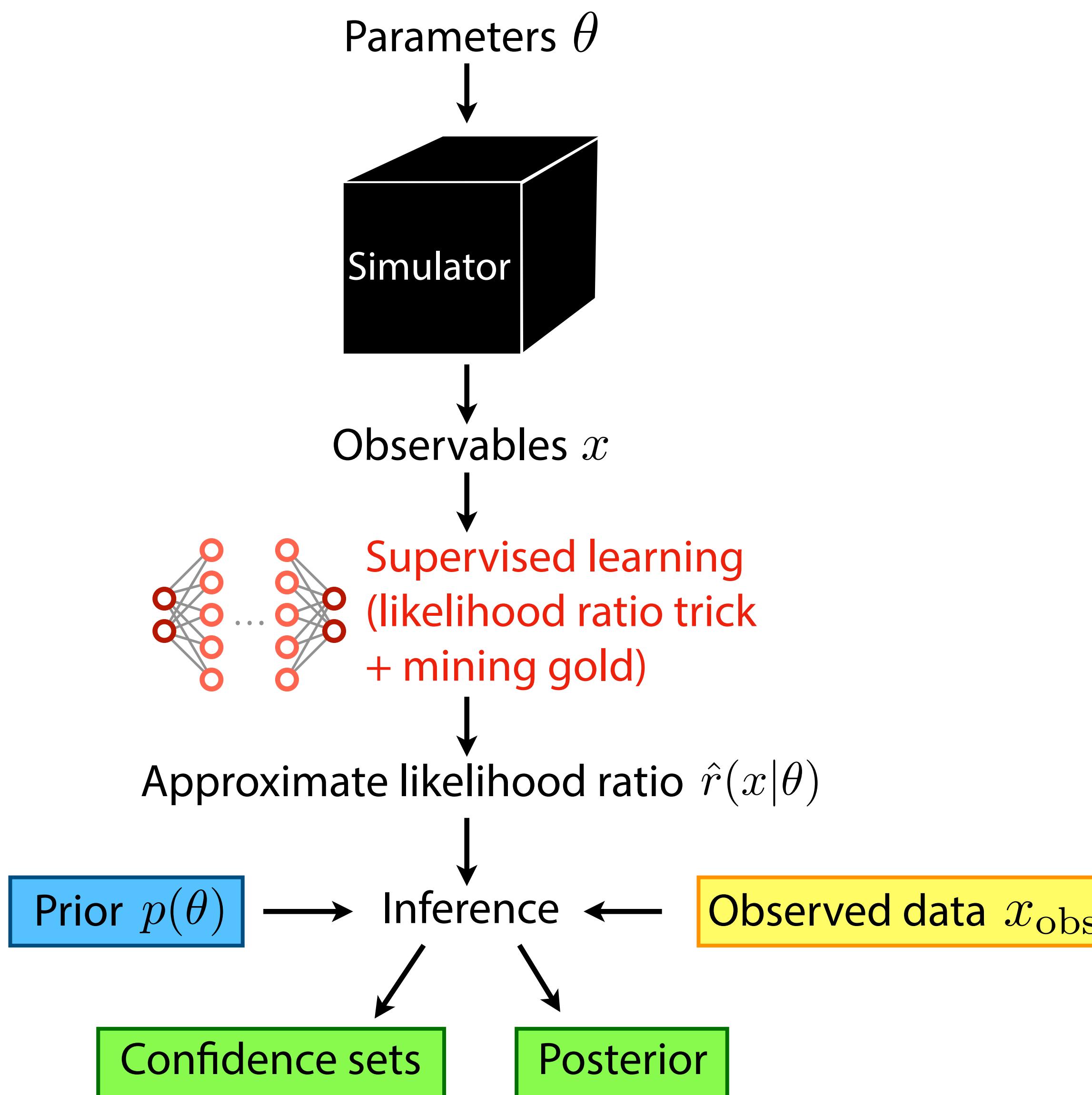


Training data: 10^6 lensed images with
 $0 \leq f_{\text{sub}} \leq 0.2, -1.5 \leq \beta \leq -0.5$

Convolutional neural network (modified ResNet-18)
trained on ALICES loss
[M. Stoye, JB, J. Pavez, G, Louppe, K. Cranmer 1808.00973]

Probabilistic calibration of network output

Inference setup



Training data: 10^6 lensed images with
 $0 \leq f_{\text{sub}} \leq 0.2, -1.5 \leq \beta \leq -0.5$

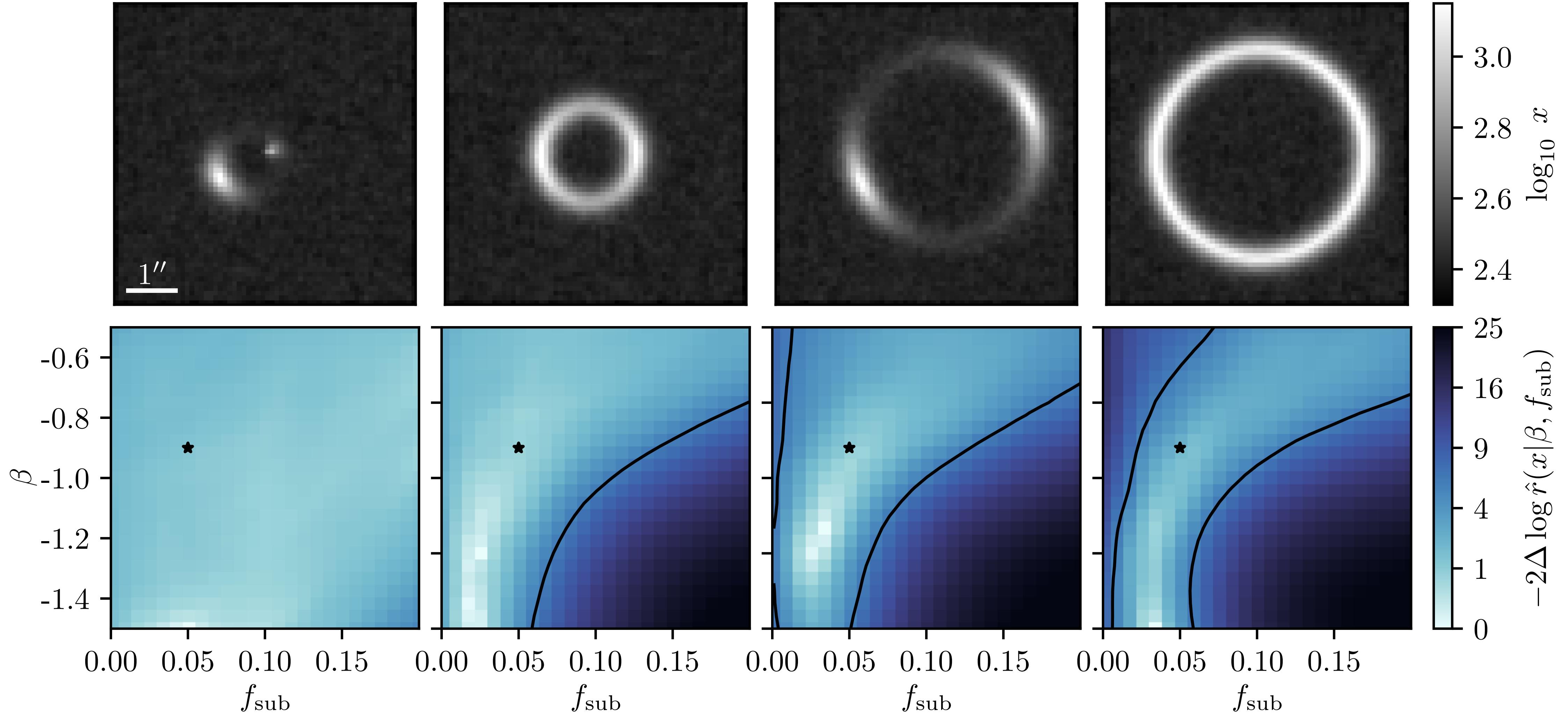
Convolutional neural network (modified ResNet-18)
trained on ALICES loss
[M. Stoye, JB, J. Pavez, G, Louppe, K. Cranmer 1808.00973]

Probabilistic calibration of network output

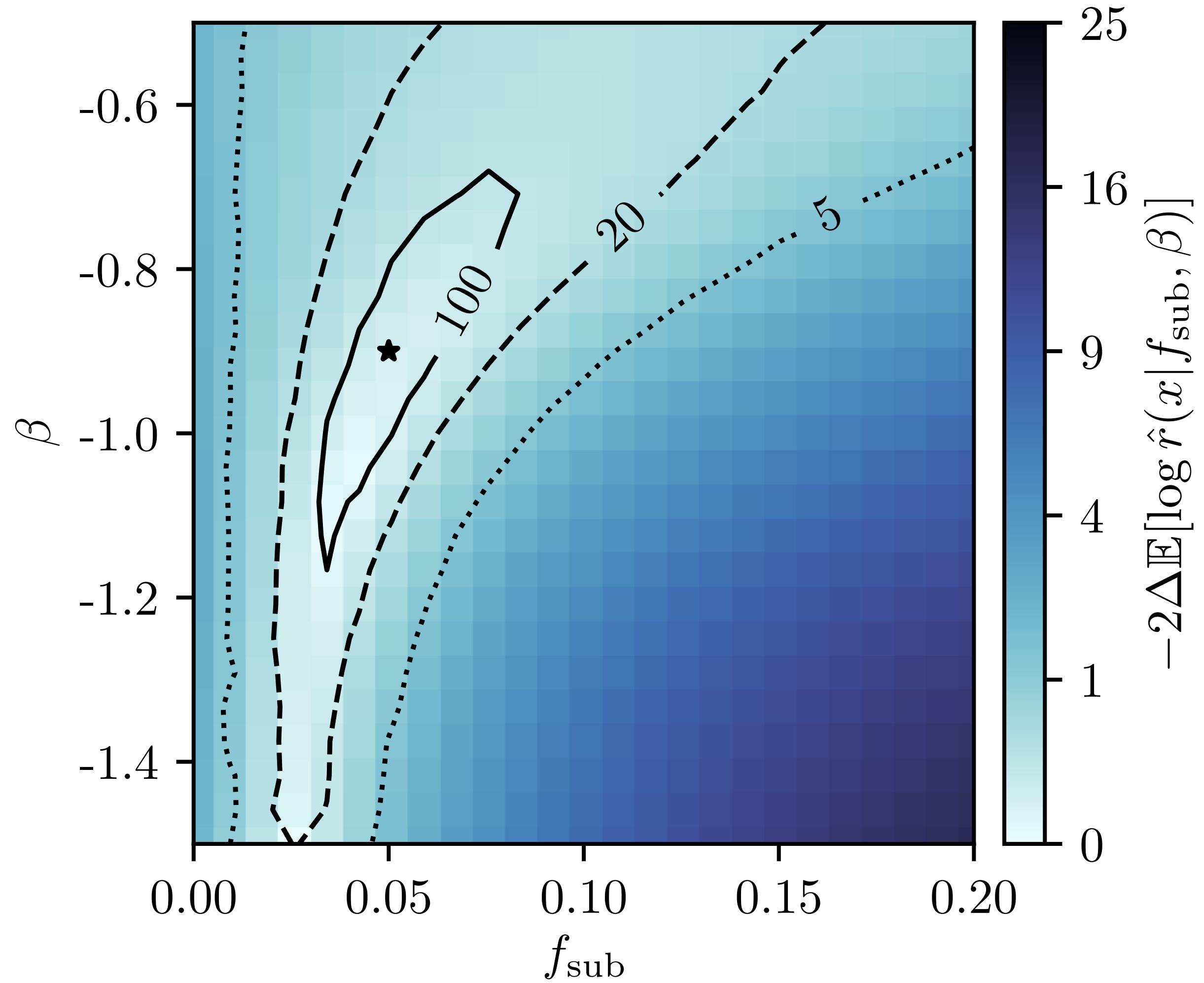
Synthetic "observed" data set simulated for
 $f_{\text{sub}} = 0.05, \beta = -0.9$

Bayesian & frequentist inference

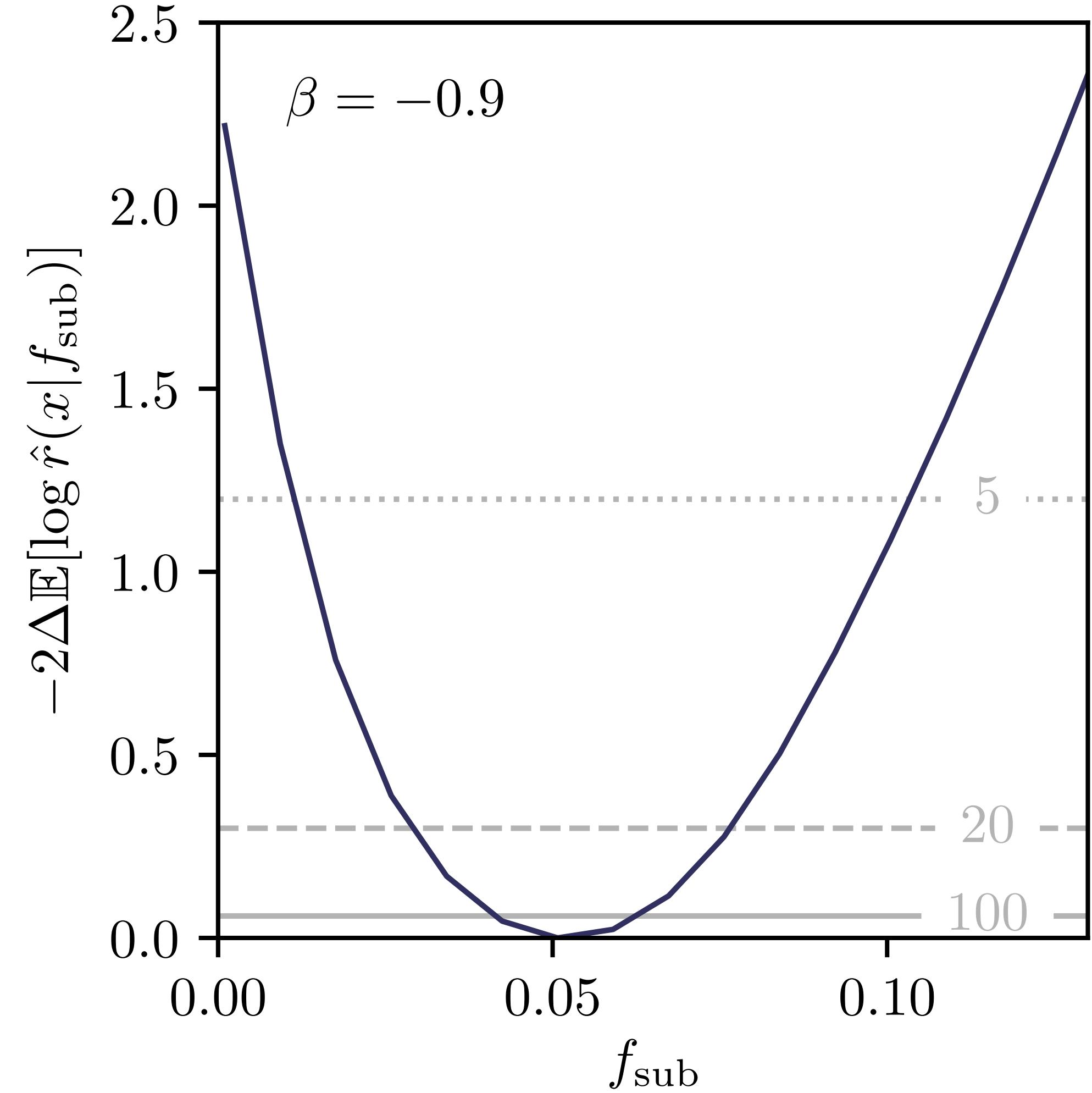
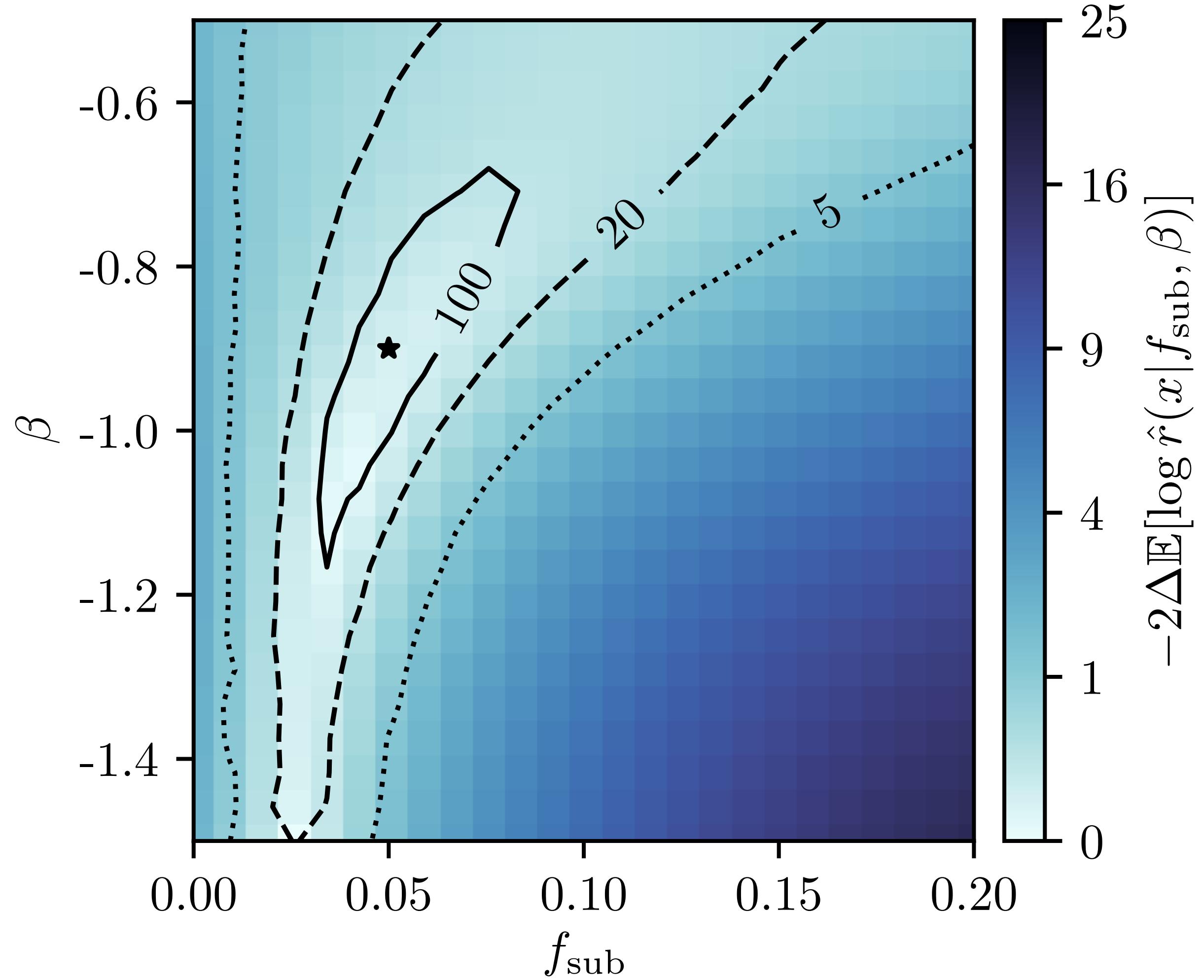
Inferring parameters from individual images



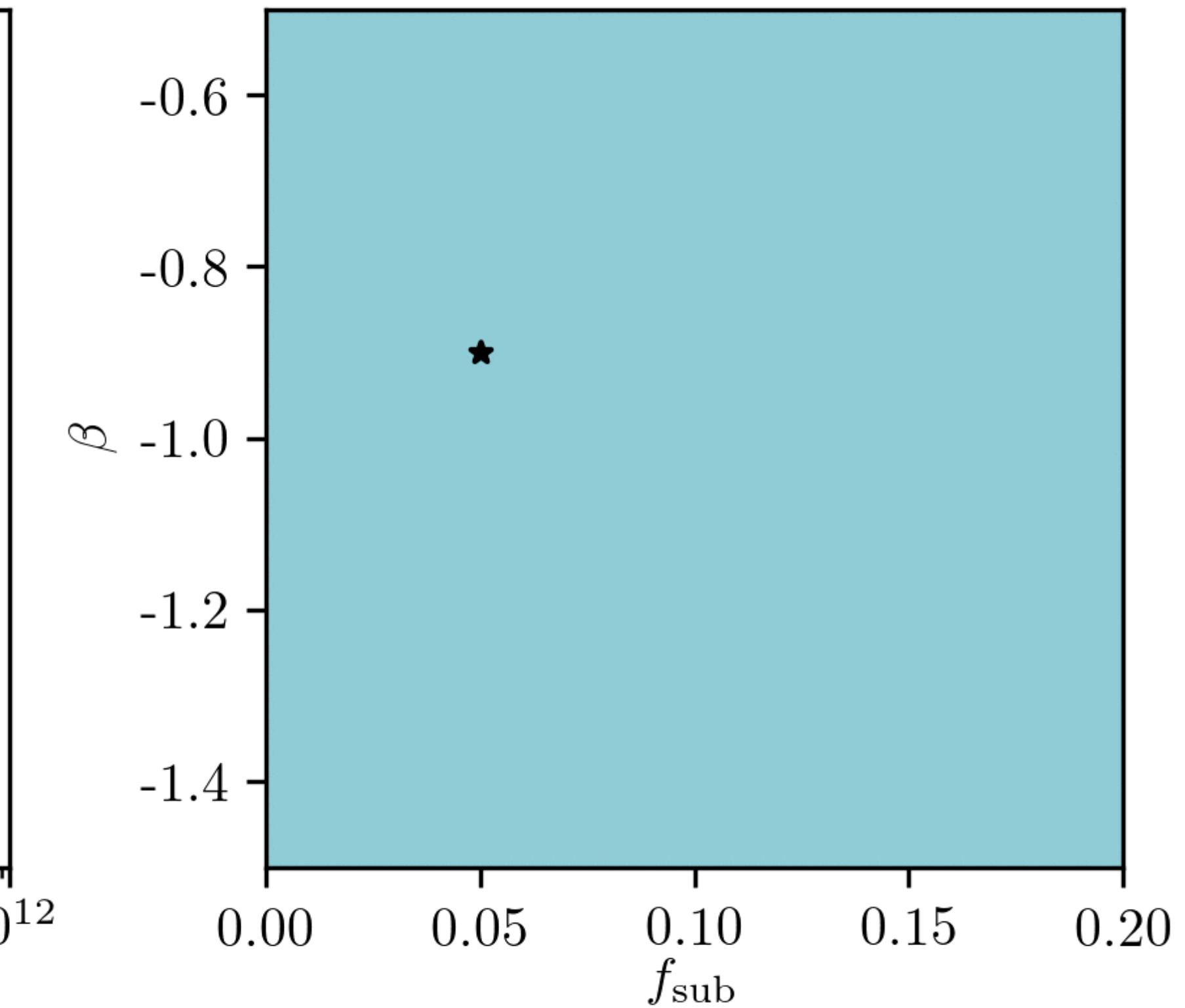
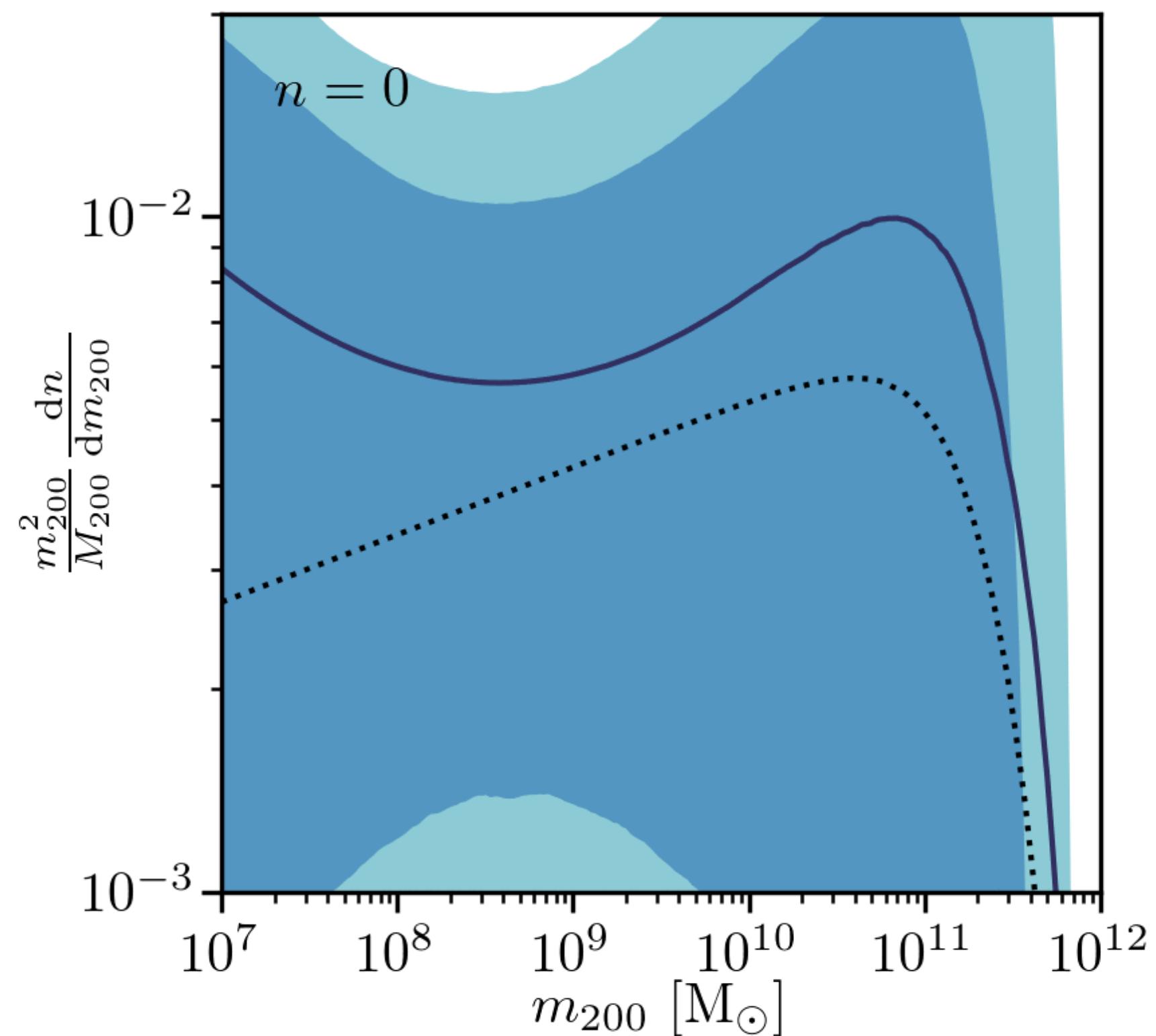
Expected likelihood ratio map

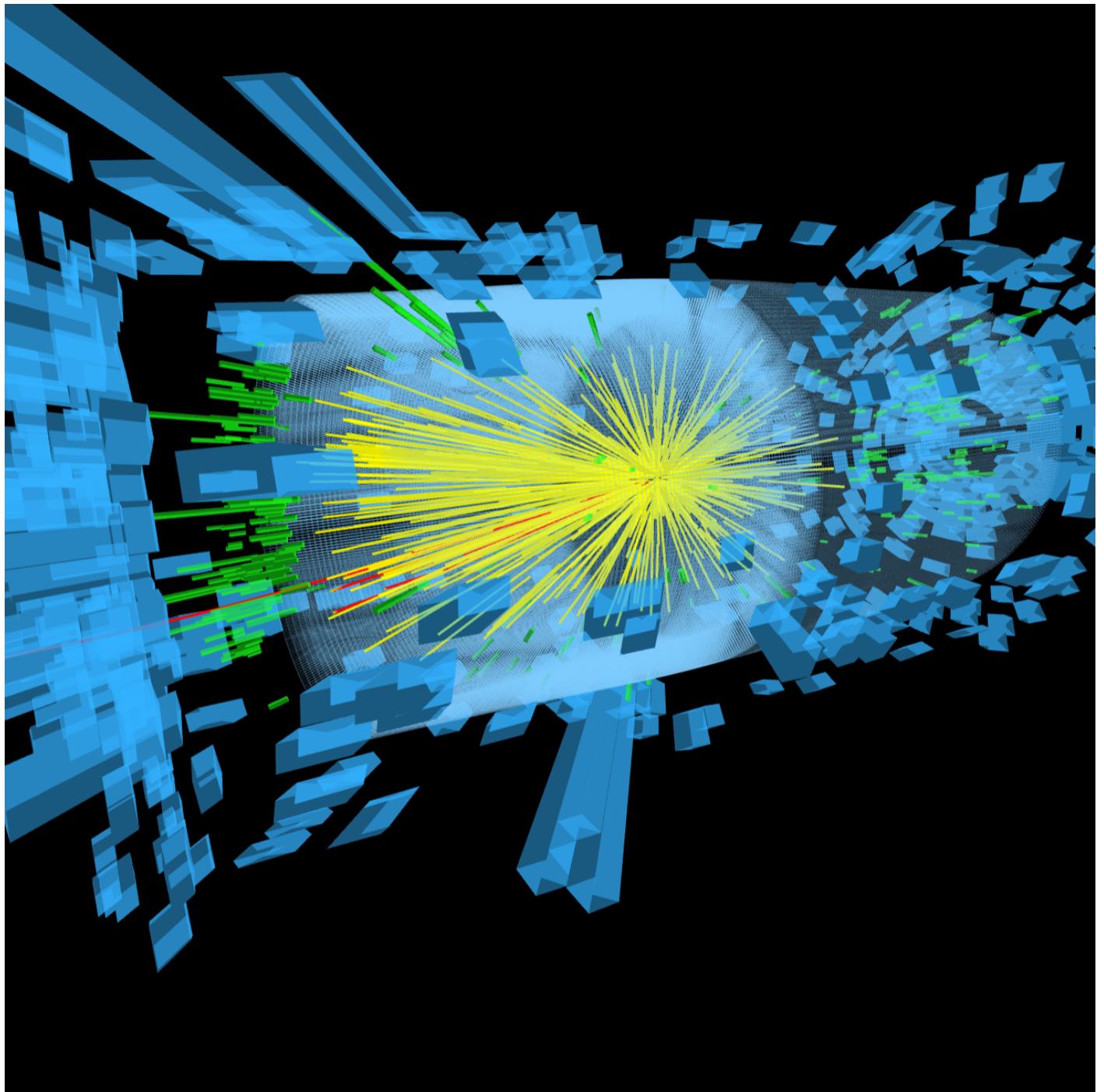


Expected likelihood ratio map

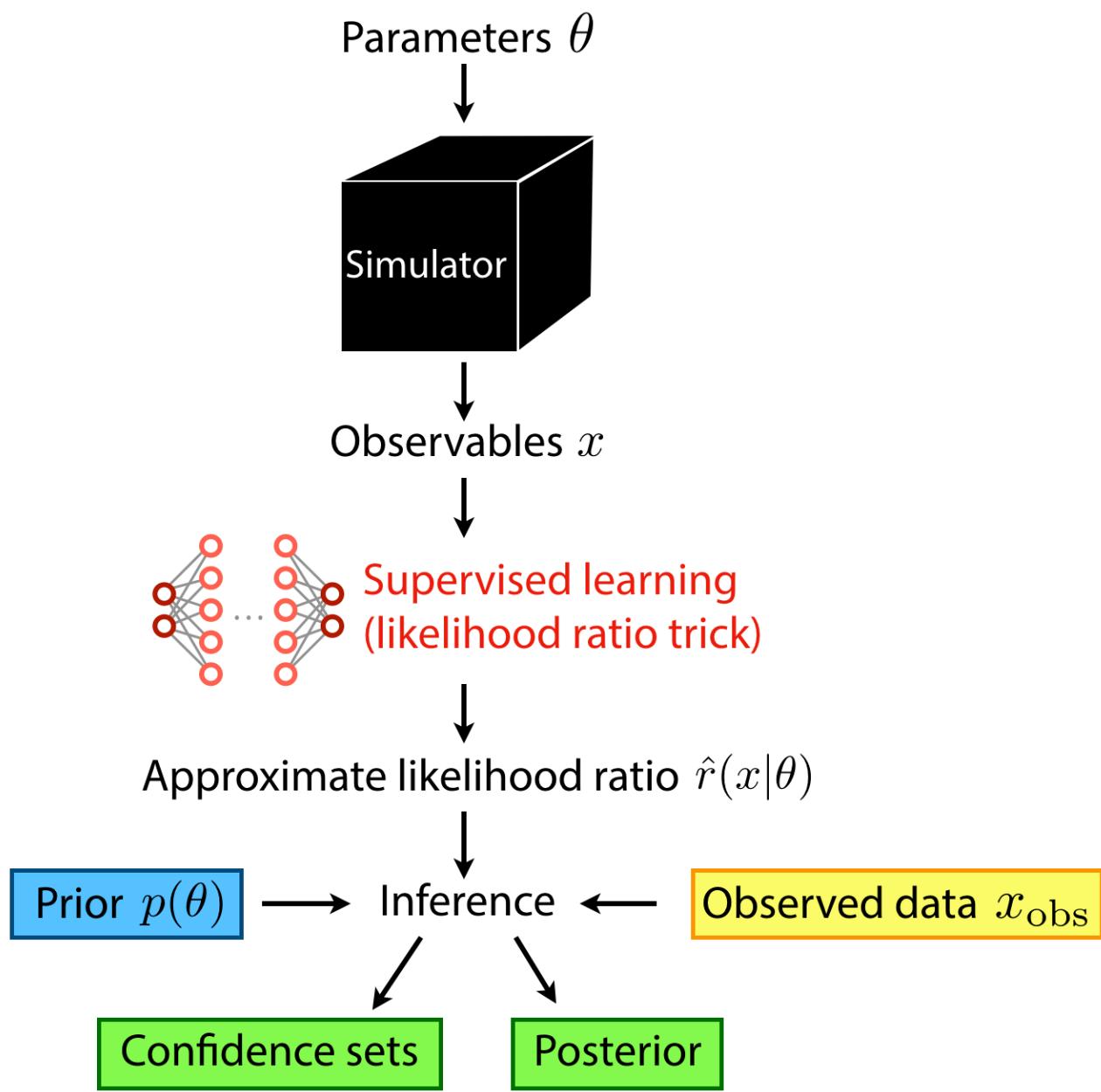


Bayesian inference

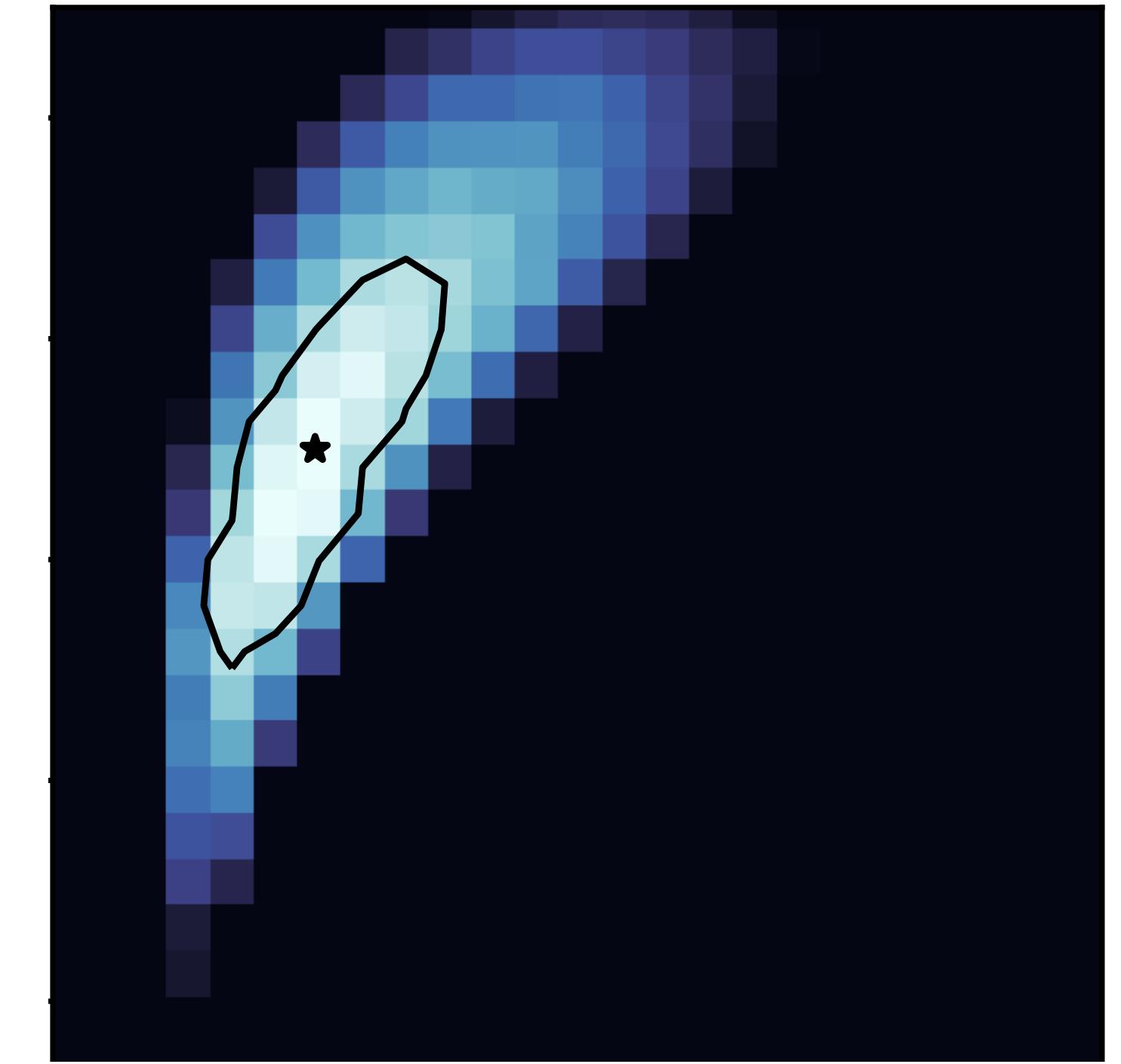




Inference is challenging when phenomena are modeled with computer simulations and the data are high-dimensional



New inference algorithms are based on normalizing flows or the likelihood ratio trick; we can improve them with physics understanding



They let us extract more knowledge e.g. from particle physics experiments

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

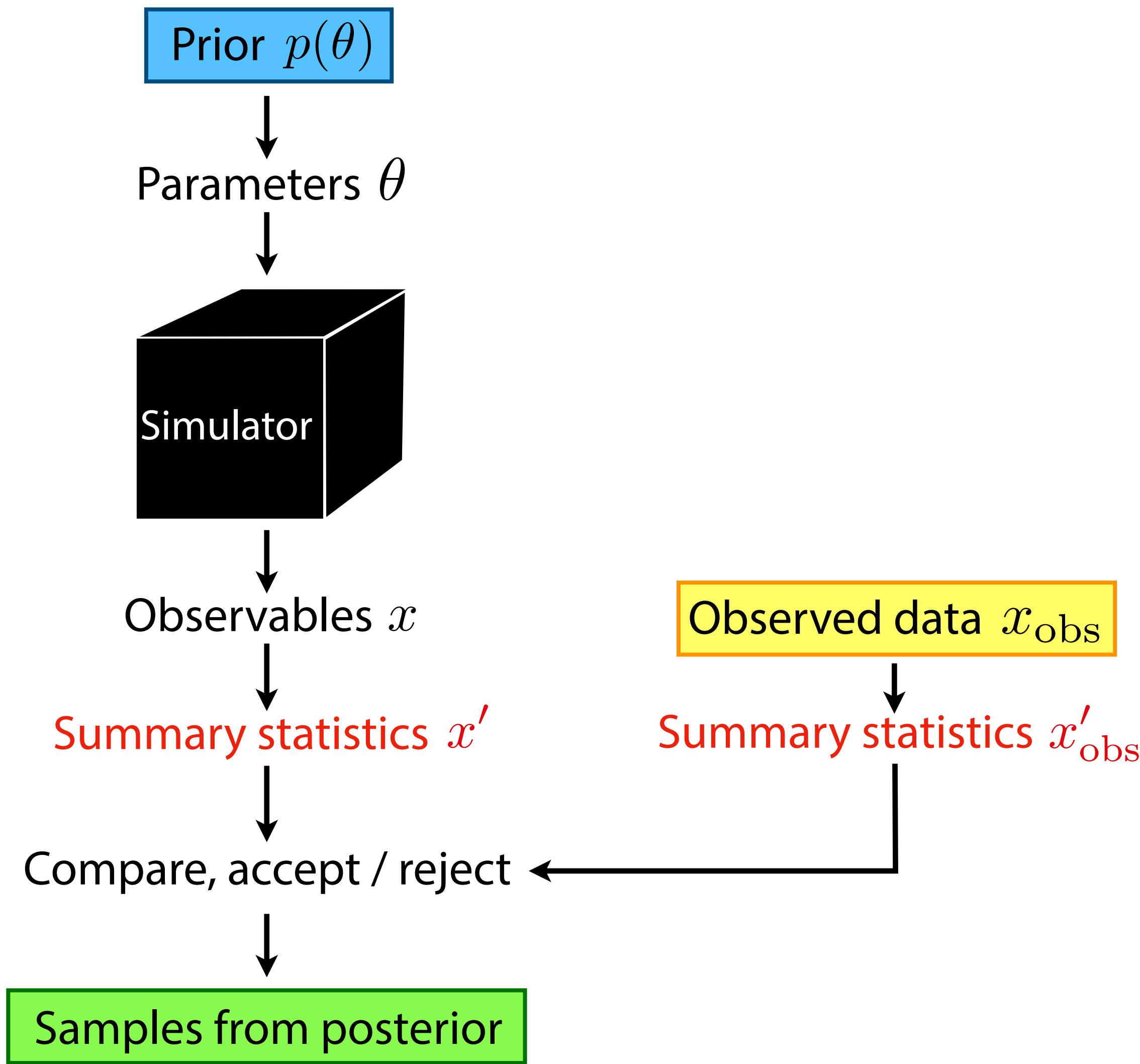
M. Stoye, JB, K. Cranmer, G. Louppe, J. Pavez:
“Likelihood-free inference with an improved
cross-entropy estimator”
[NeurIPS workshop, 1808.00973]

“Hooray! Question mark?”

— Todd Chavez

Approximate Bayesian Computation (ABC)

[D. Rubin 1984]

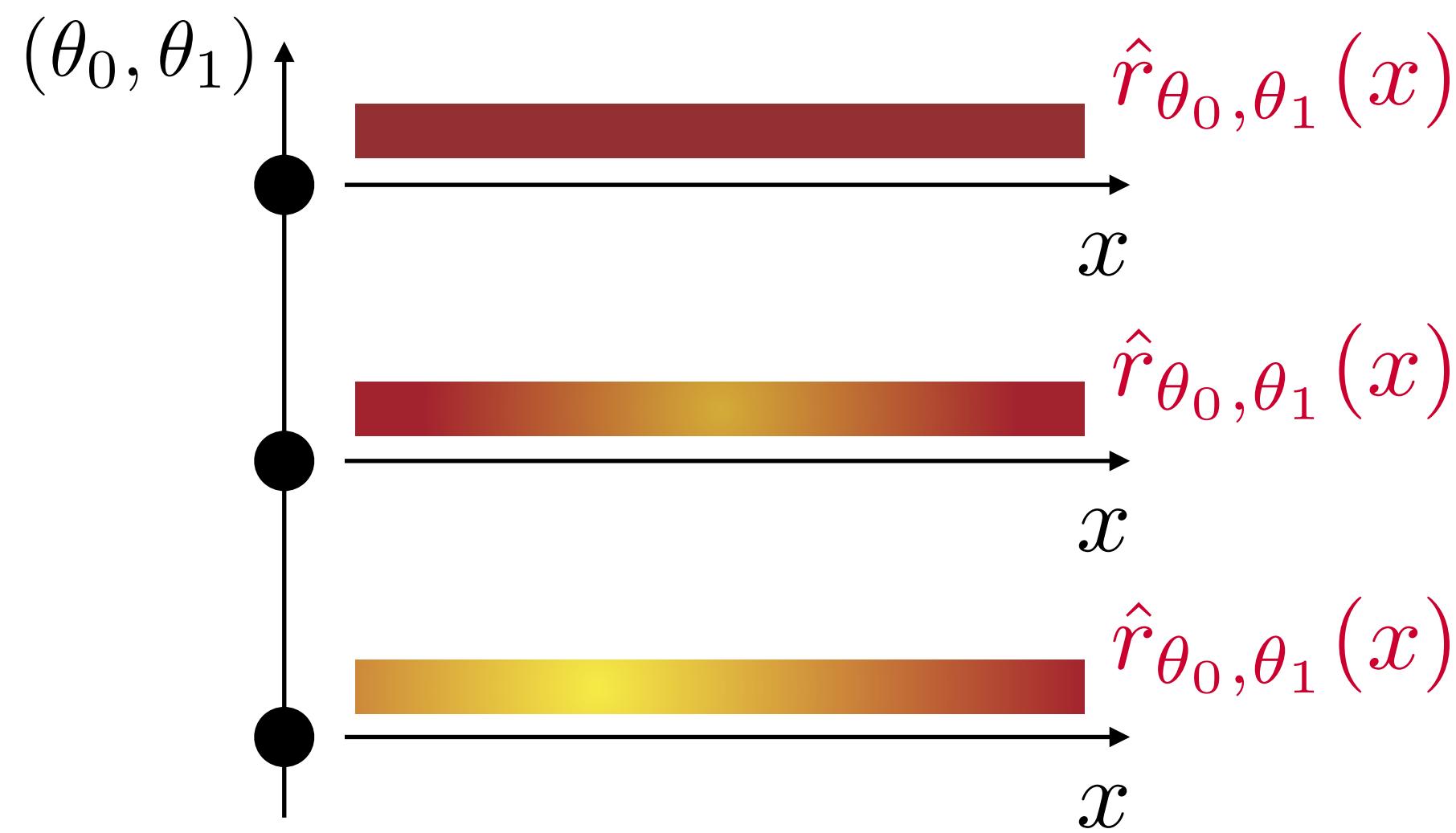


- Compression to summary statistics and acceptance threshold reduce quality of inference
- Rejection algorithm can be very sample inefficient

Two types of likelihood ratio estimators

A) Point by point:

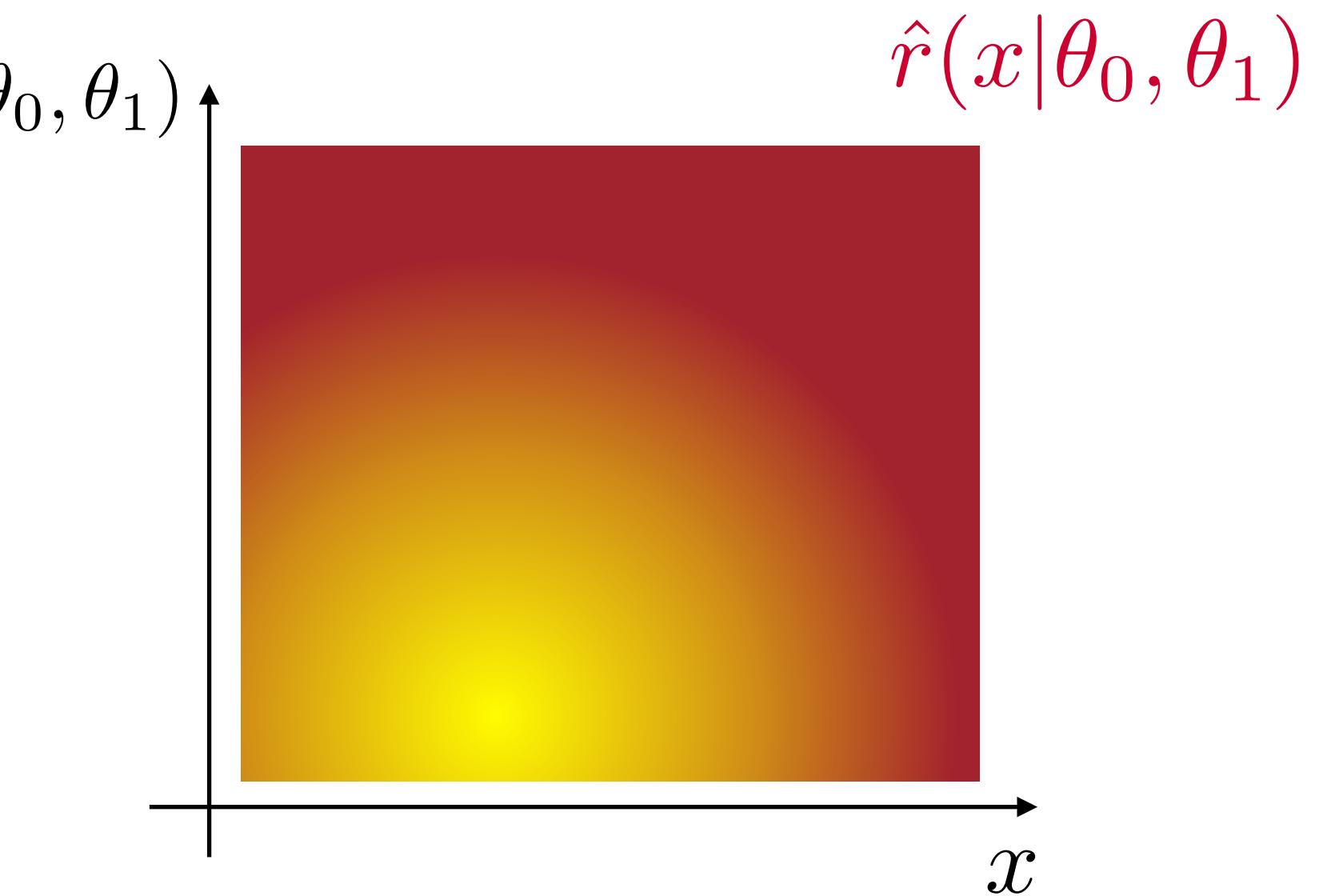
- first, define grid of parameter points $\{(\theta_0, \theta_1)\}$
- for each combination (θ_0, θ_1) ,
create separate estimator $\hat{r}_{\theta_0, \theta_1}(x)$
- final results can be interpolated between grid points



B) Parameterized:

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

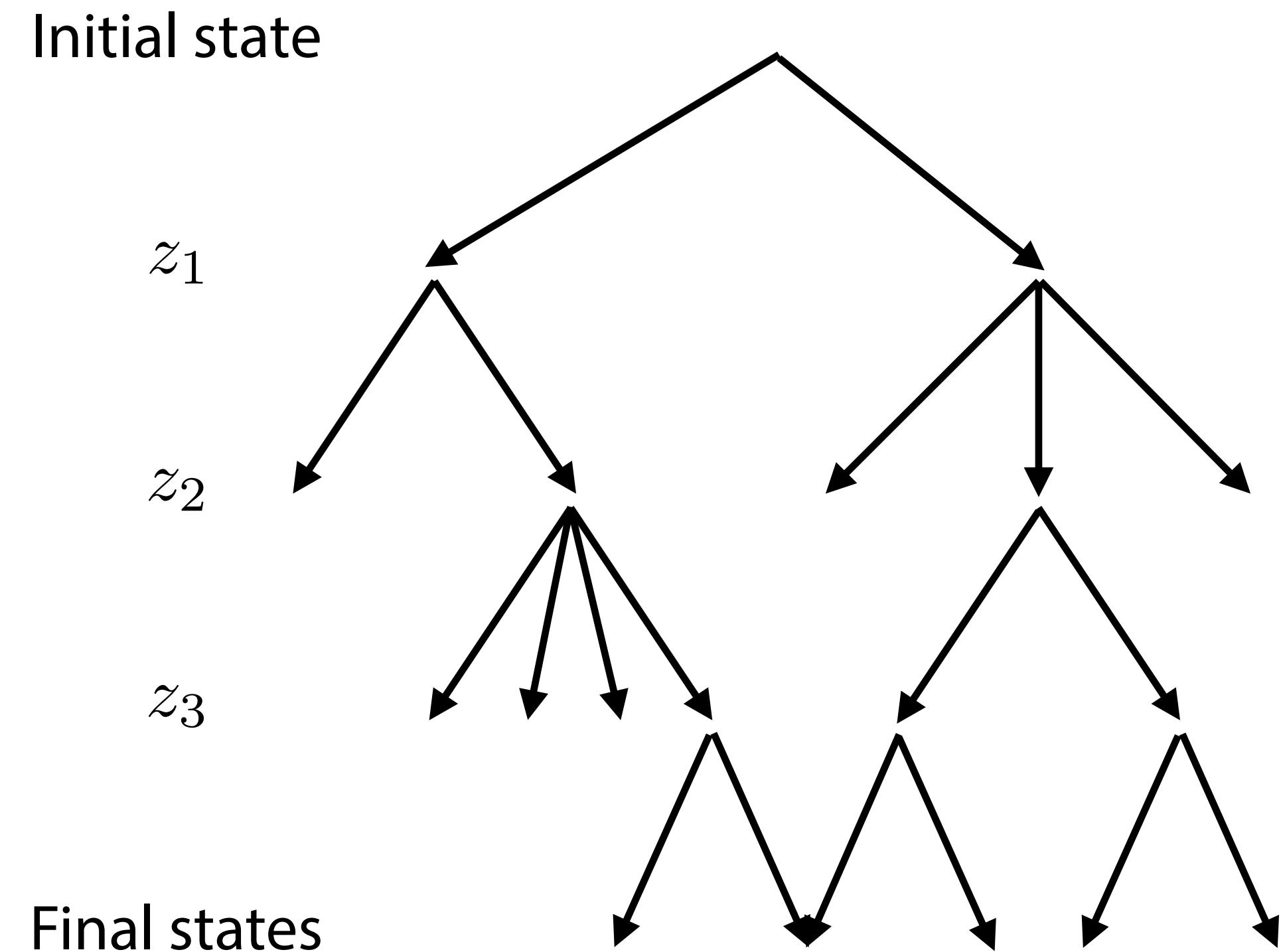
- create one estimator $\hat{r}(x|\theta_0, \theta_1)$
that is a function of θ_0 and θ_1
- no further interpolation necessary
- “borrows information” from close points



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()
```



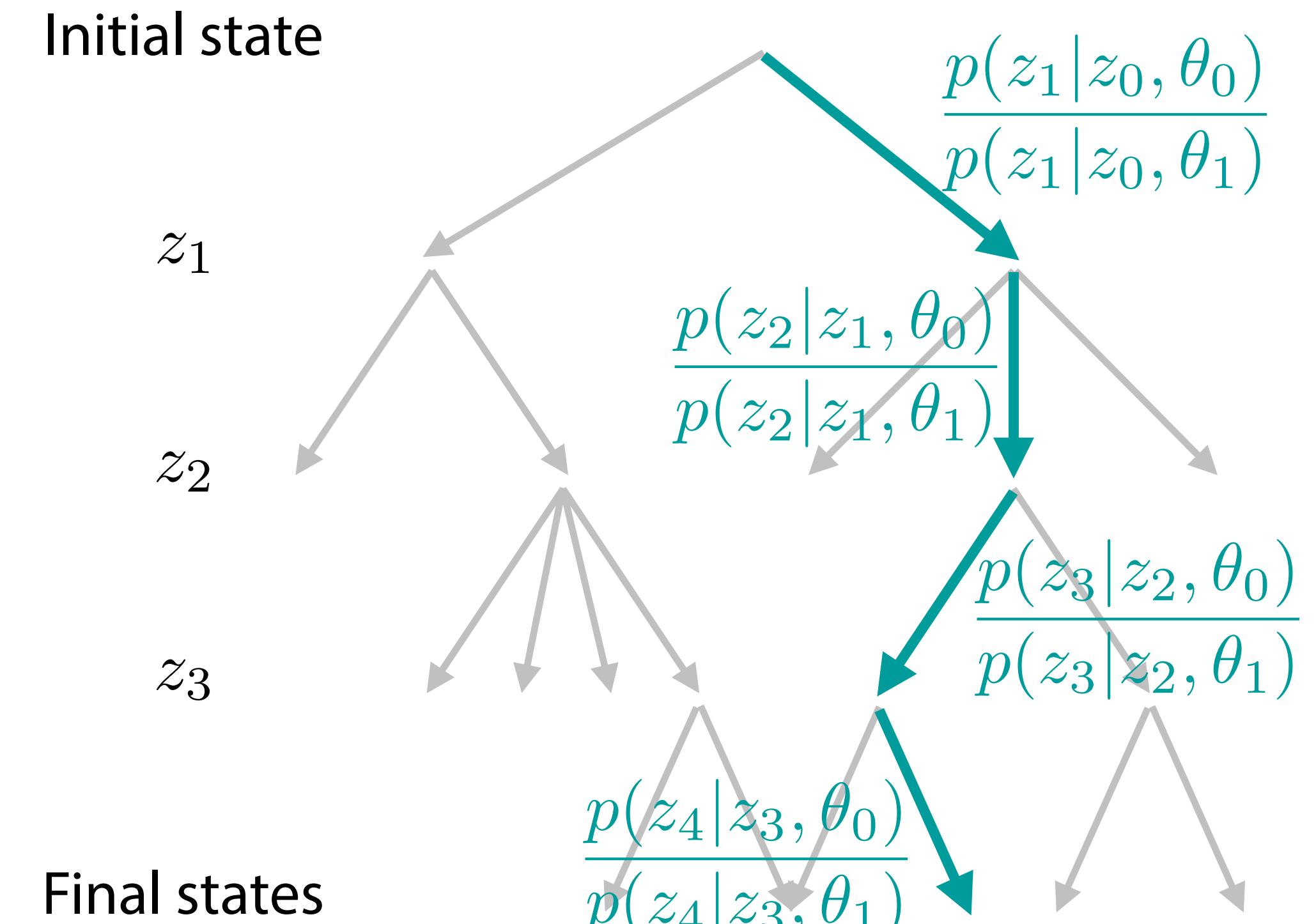
Mining gold from any simulation

- 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		NN estimates	Asympt. exact	Generative
		$r(x, z)$	$t(x, z)$			
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	θ_{ref}		✓	$\hat{t}(x \theta_{\text{ref}})$	in local approx.	
SALLINO	θ_{ref}		✓	$\hat{t}(x \theta_{\text{ref}})$	in local approx.	

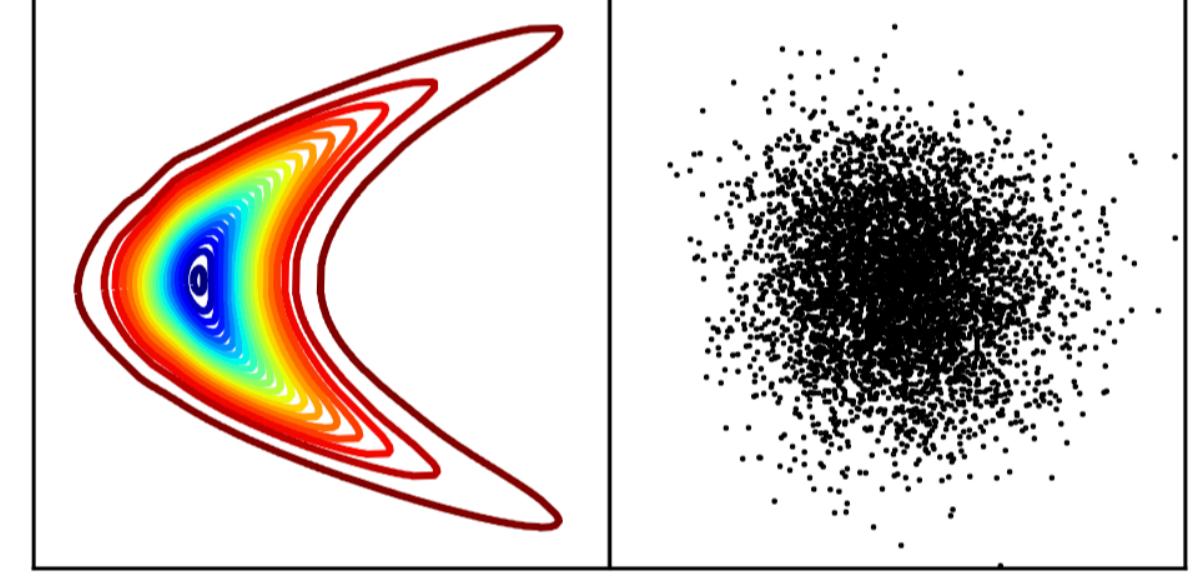
Mining gold: A family of new inference techniques

Method	Simulate	Extract $r(x, z)$	$t(x, z)$	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	θ_{ref}		✓	$\hat{t}(x \theta_{\text{ref}})$	in local approx.	
SALLINO	θ_{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 $r(x, z)$	$t(x, z)$	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	θ_{ref}		✓	$\hat{t}(x \theta_{\text{ref}})$	in local approx.	
SALLINO	θ_{ref}		✓	$\hat{t}(x \theta_{\text{ref}})$	in local approx.	



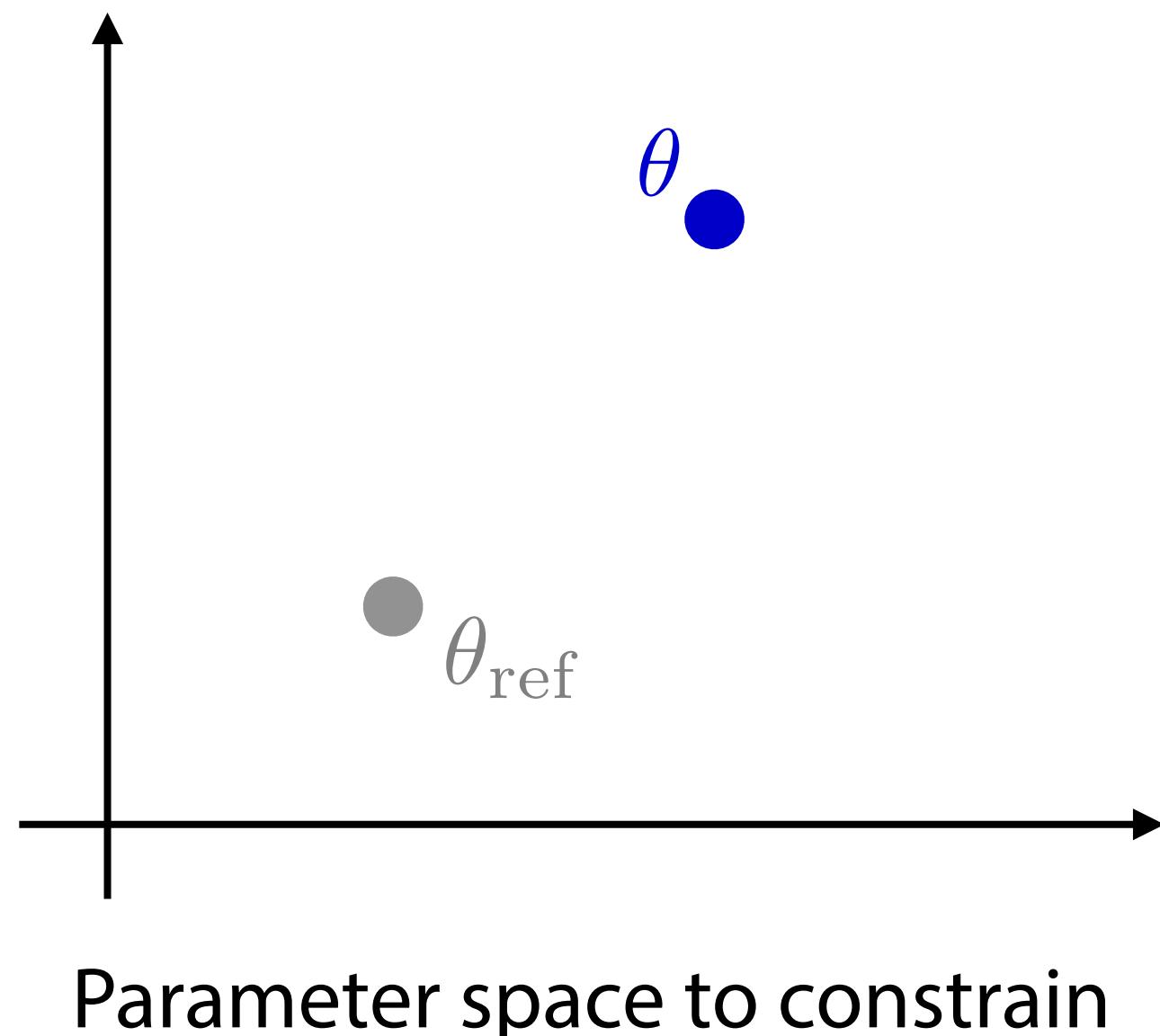
Combination with state-of-the-art conditional neural density estimators, e.g. normalizing flows

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

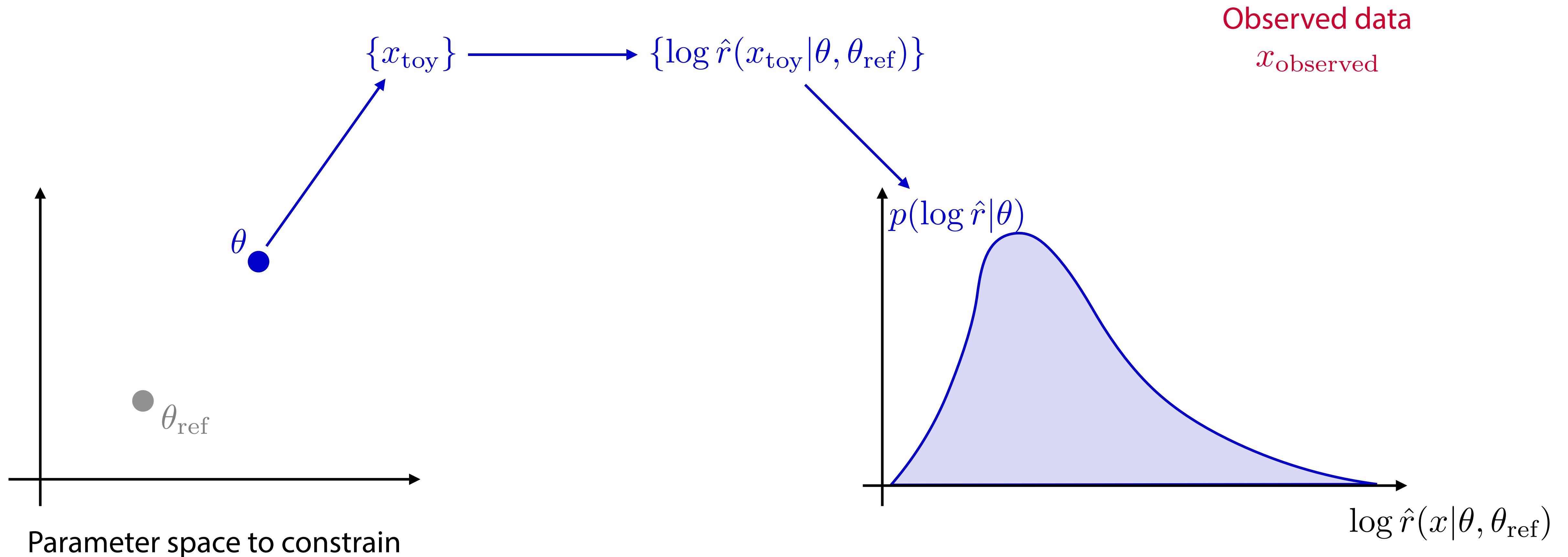
Limit setting (frequentist, standard ATLAS / CMS practice)

Observed data

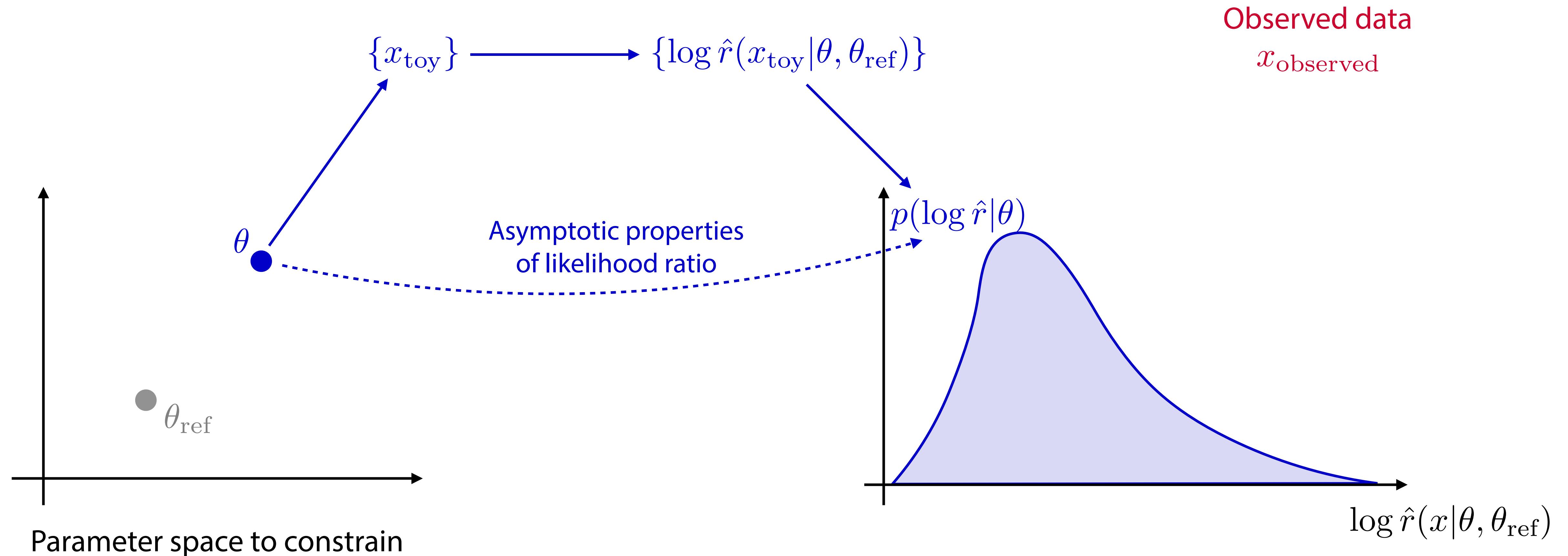
x_{observed}



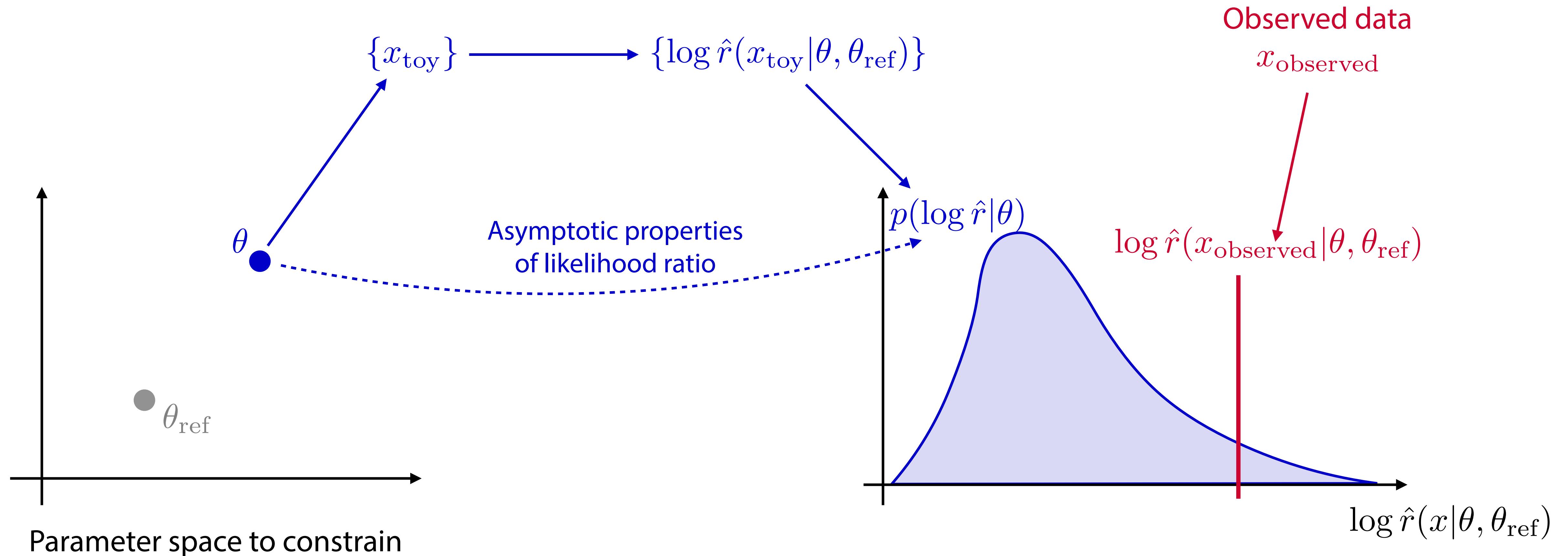
Limit setting (frequentist, standard ATLAS / CMS practice)



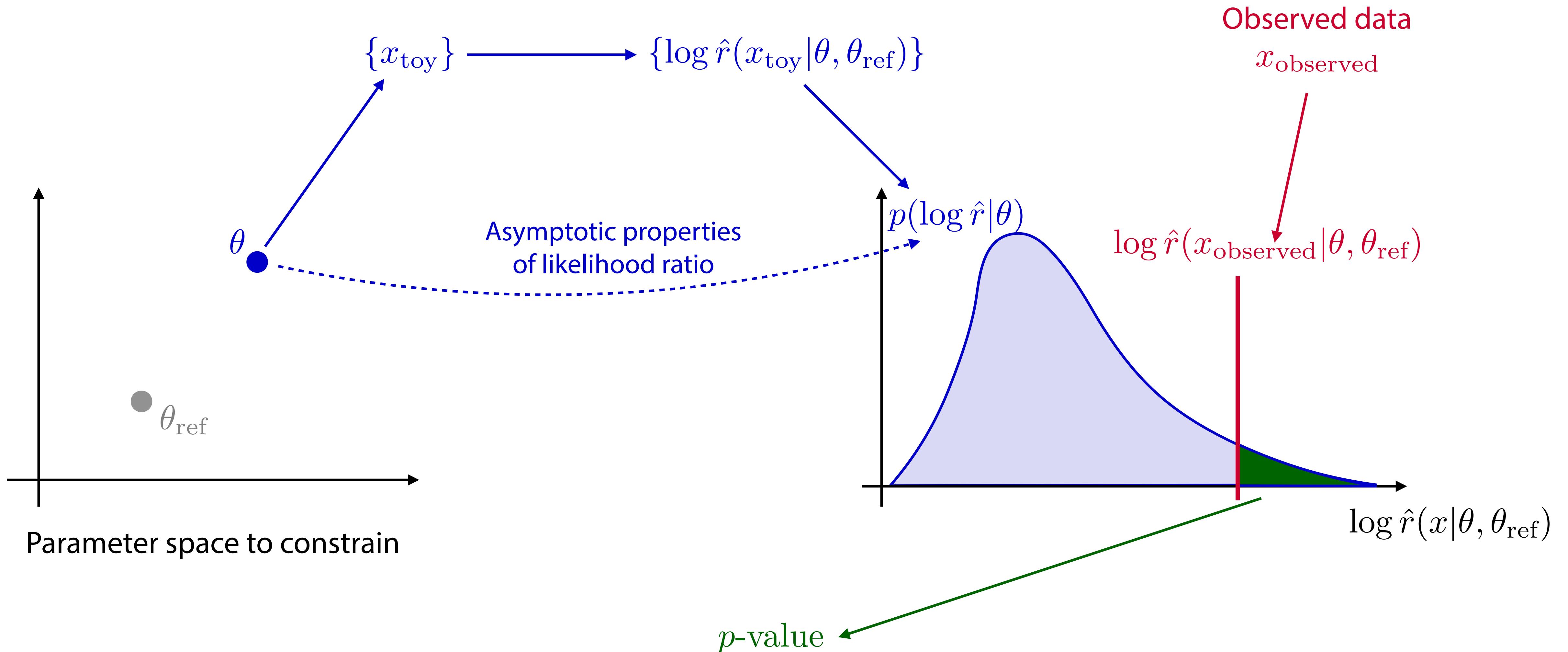
Limit setting (frequentist, standard ATLAS / CMS practice)



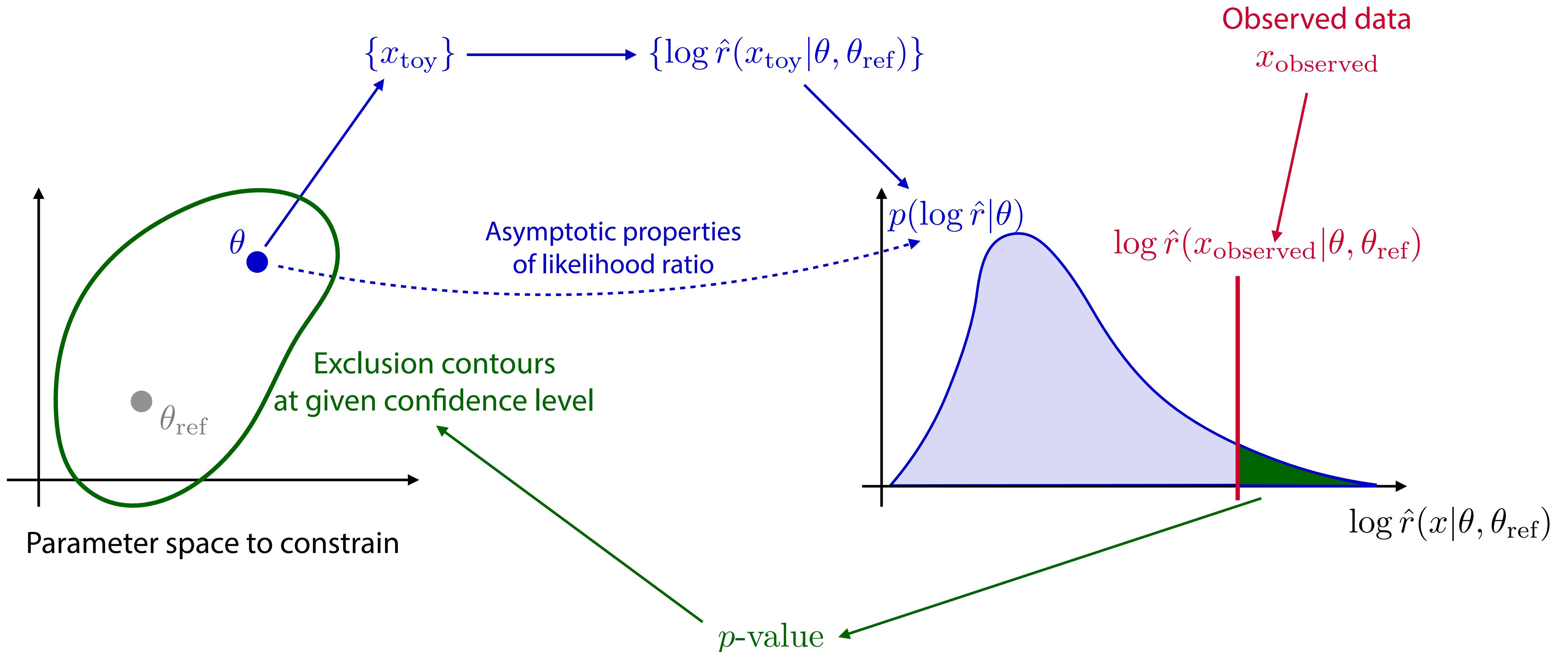
Limit setting (frequentist, standard ATLAS / CMS practice)



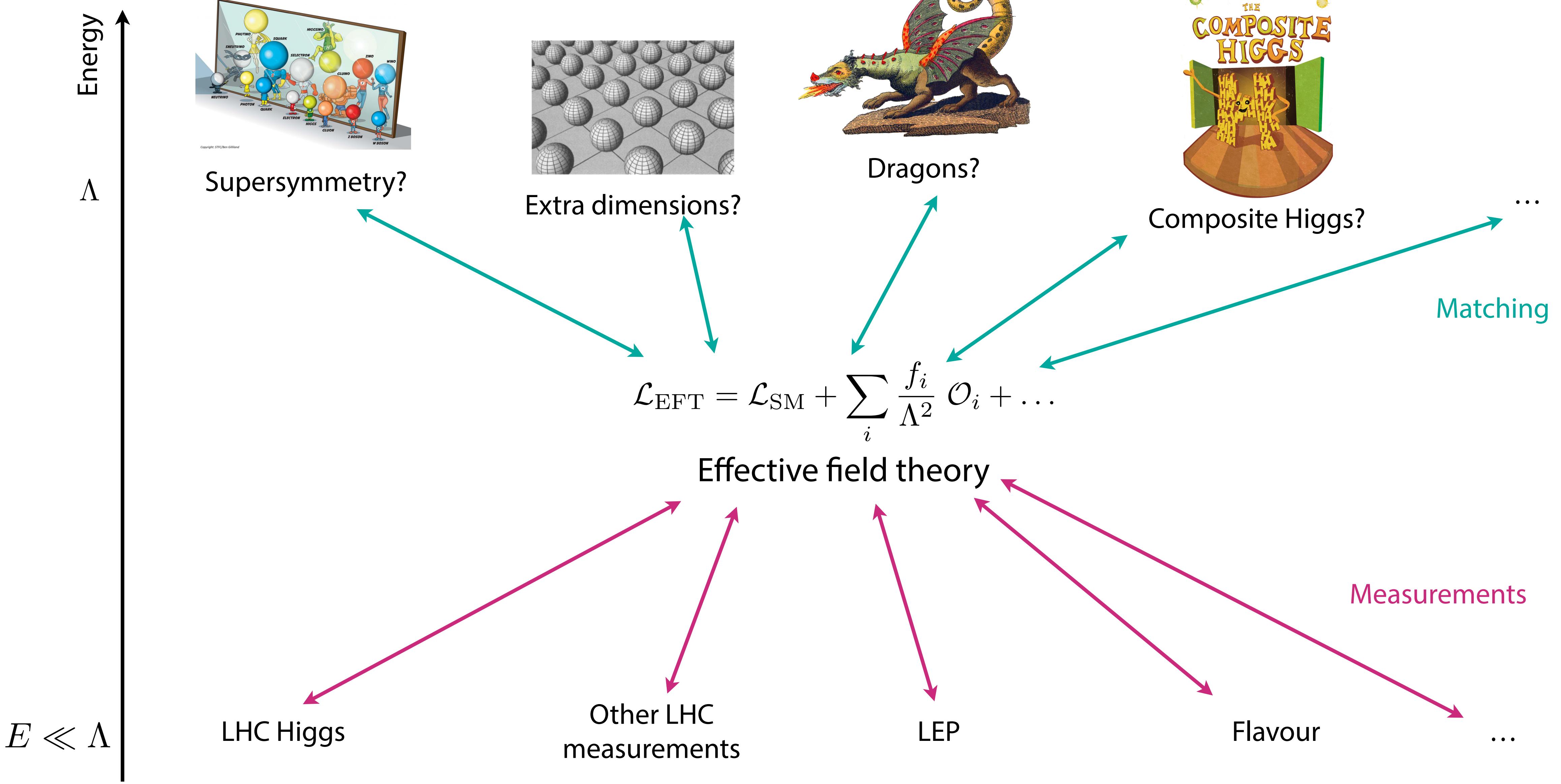
Limit setting (frequentist, standard ATLAS / CMS practice)



Limit setting (frequentist, standard ATLAS / CMS practice)

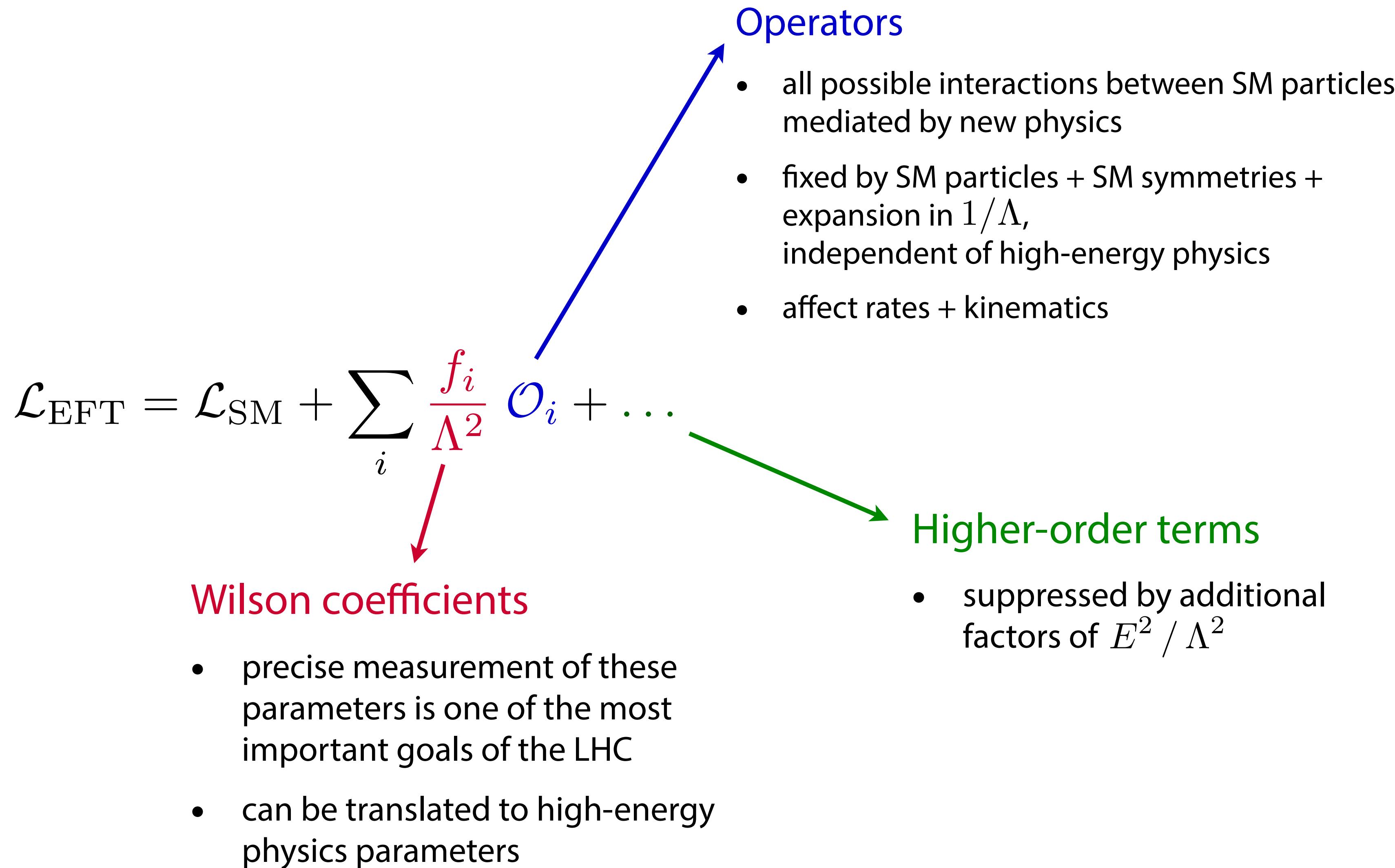


Effective field theory



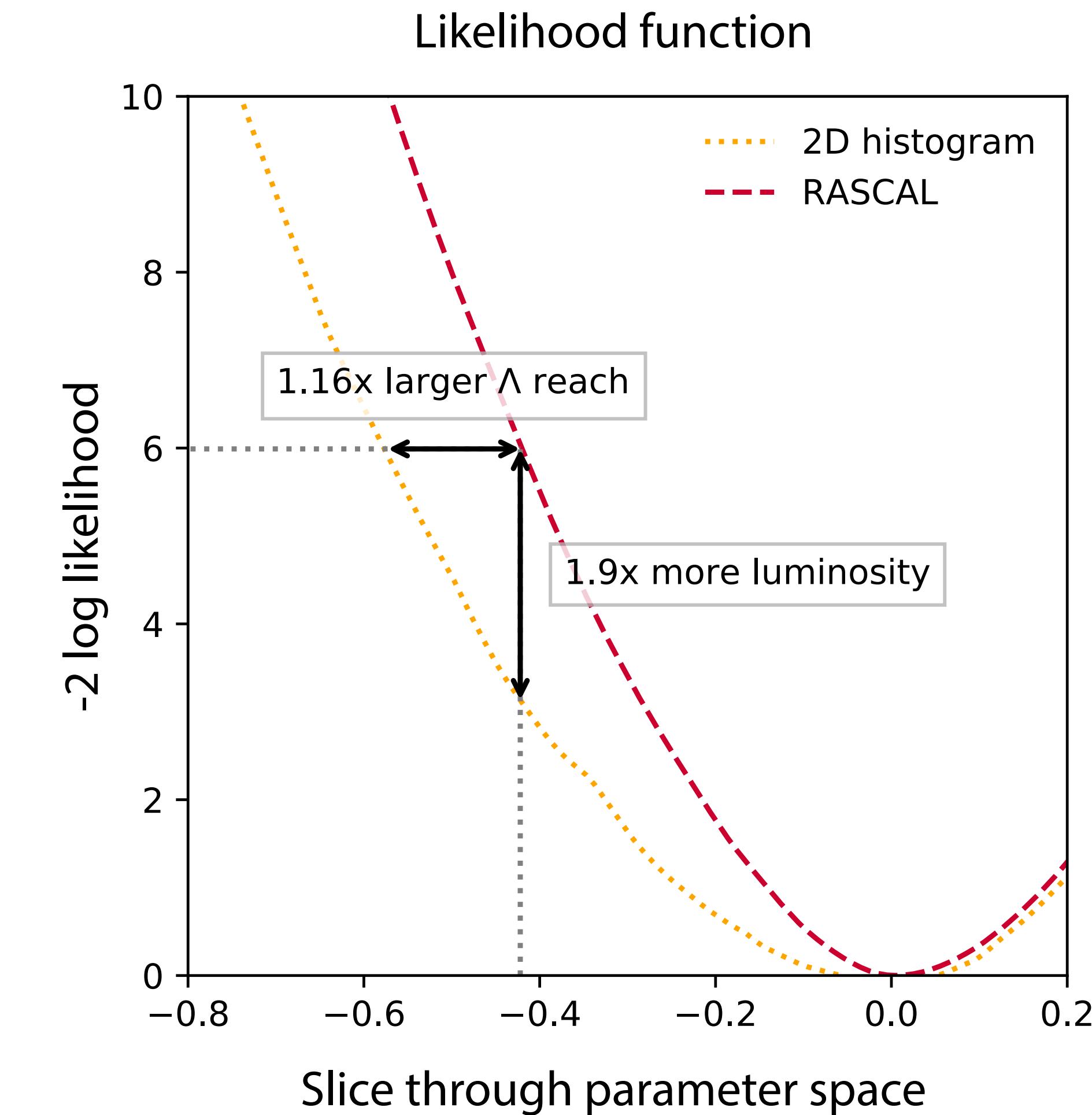
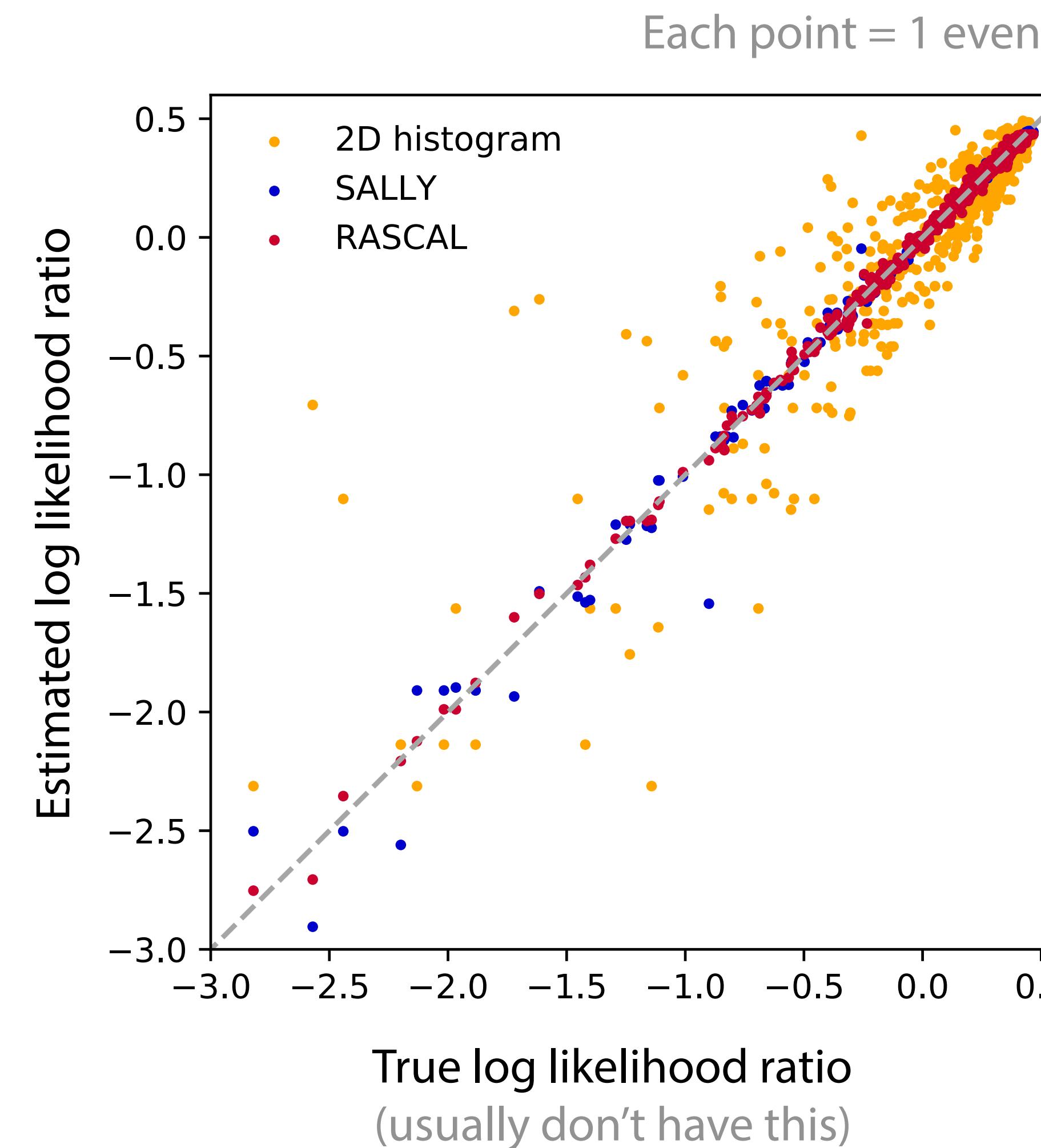
SMEFT (Standard Model Effective Field Theory)

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

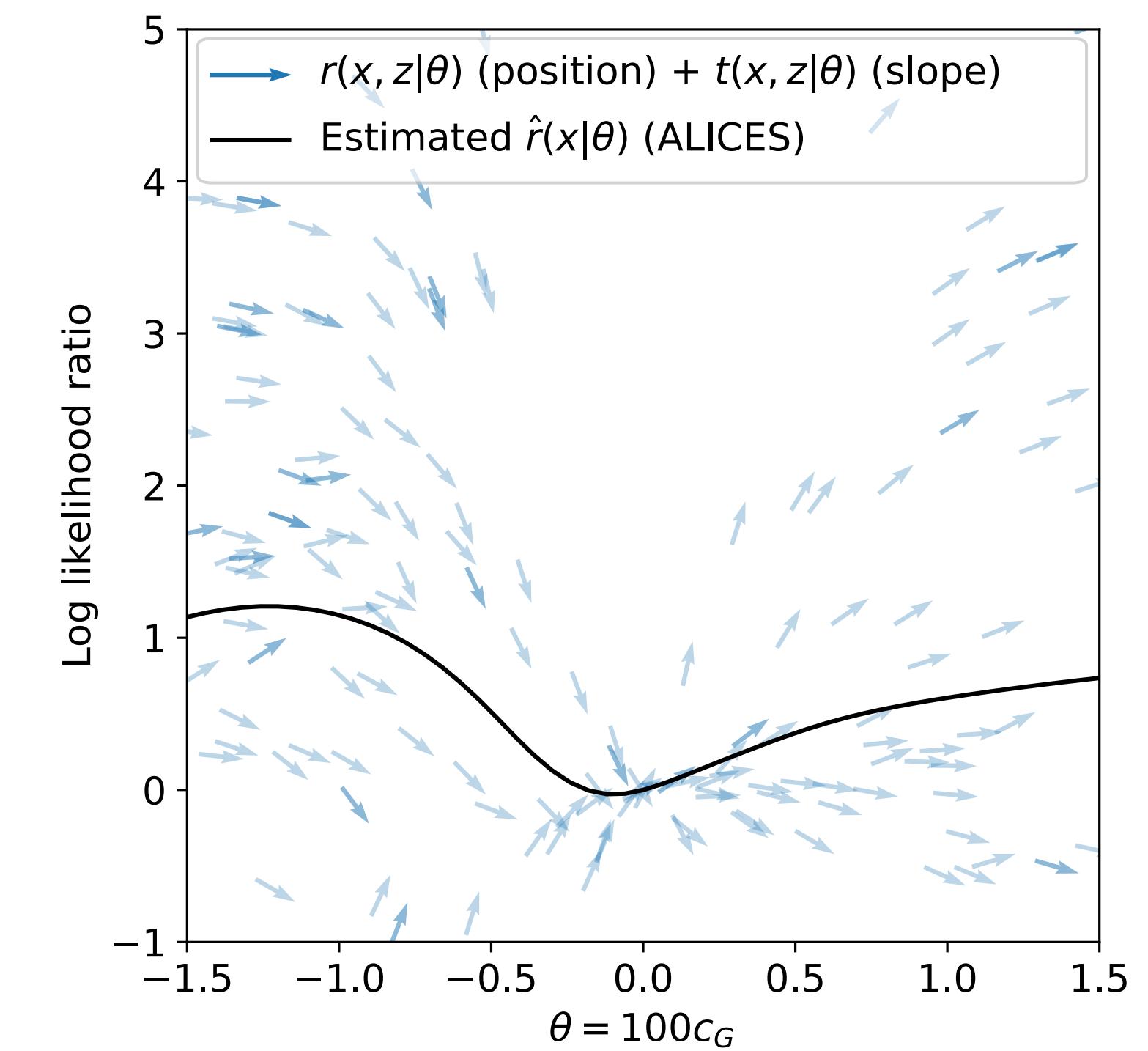
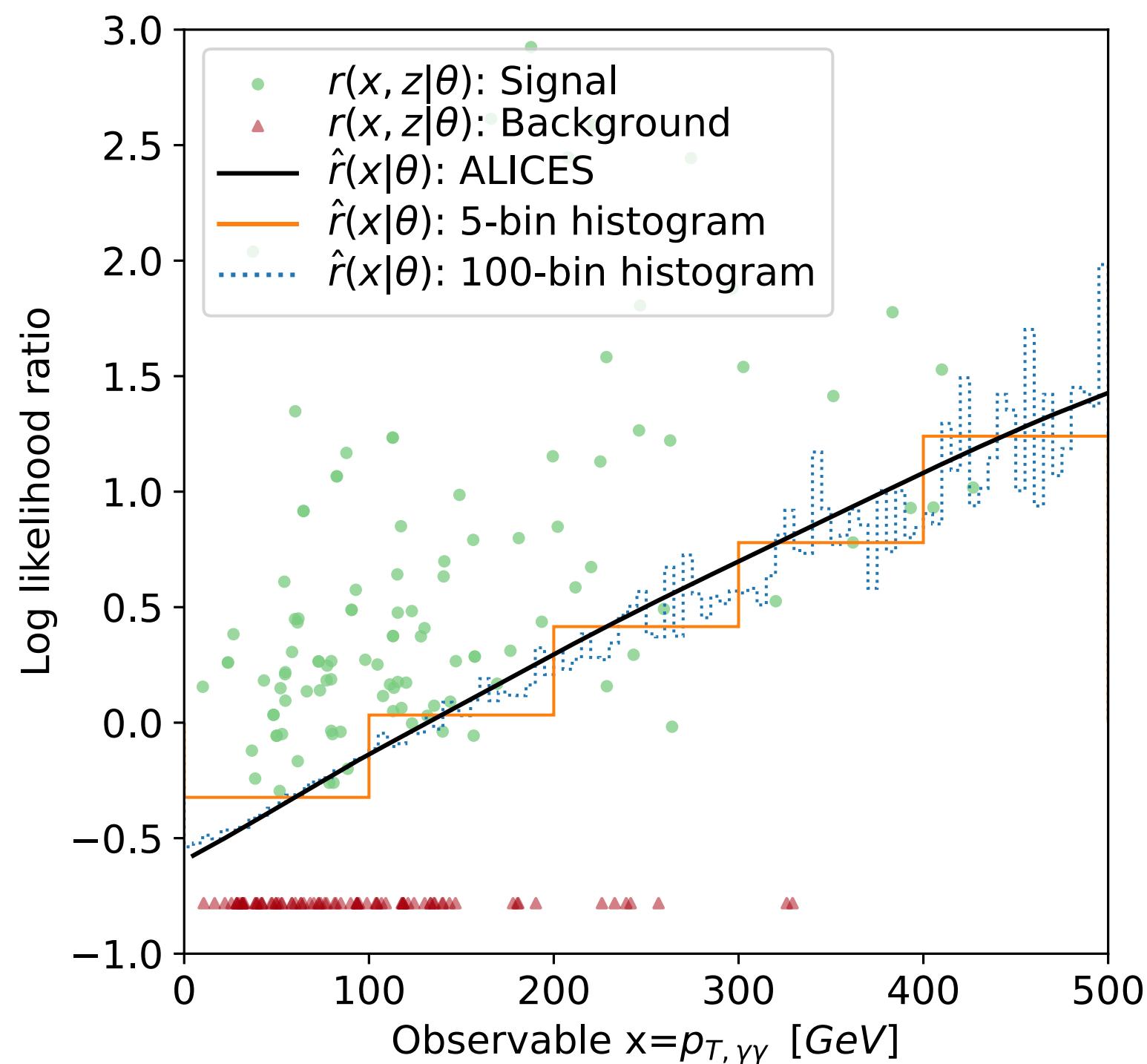
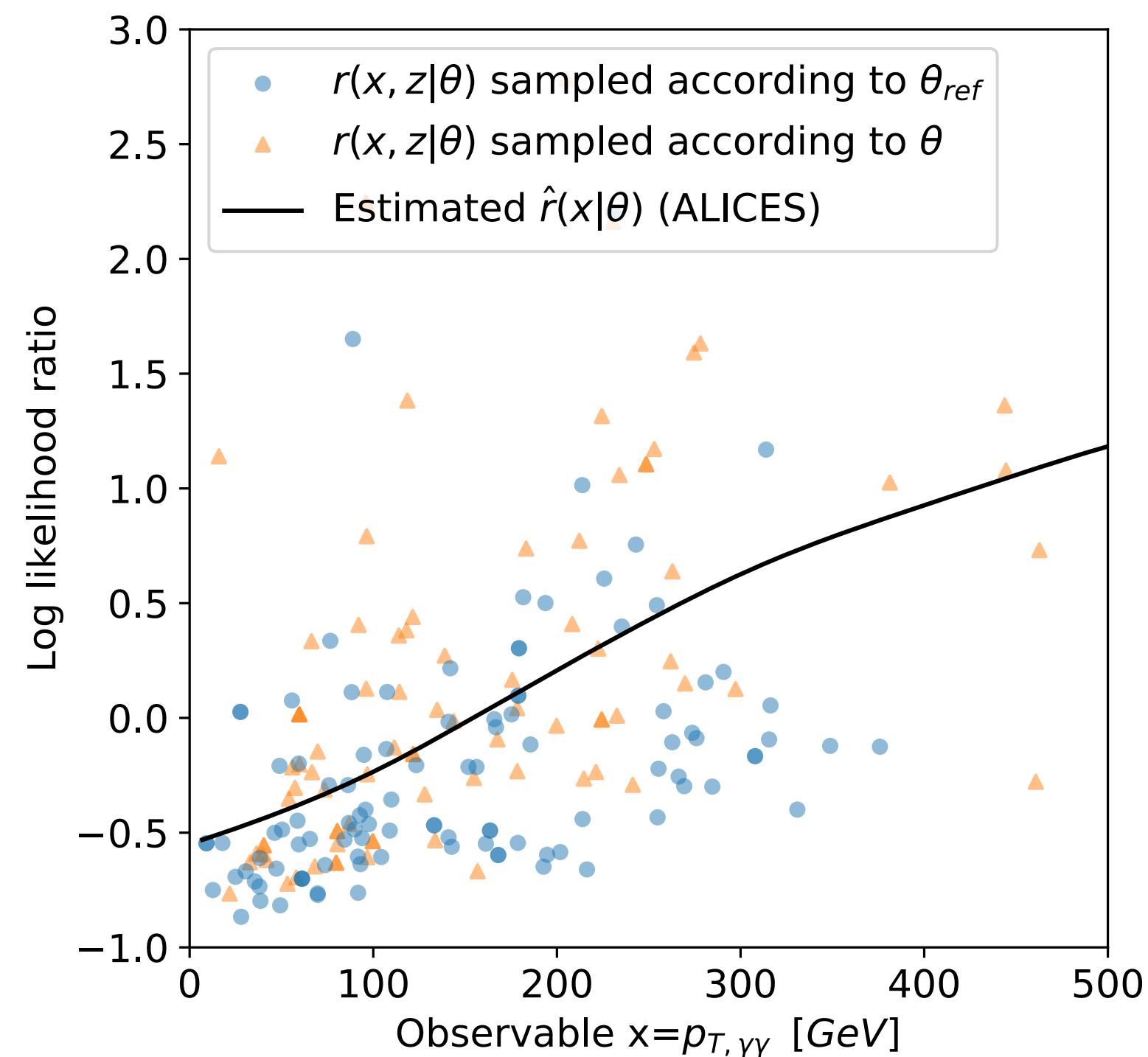


$\mathcal{O}_{\phi,1} = (D_\mu \phi)^\dagger \phi \phi^\dagger D^\mu \phi$	$\mathcal{O}_{GG} = (\phi^\dagger \phi) G_{\mu\nu}^a G^{\mu\nu a}$
$\mathcal{O}_{\phi,2} = \frac{1}{2} \partial_\mu (\phi^\dagger \phi) \partial^\mu (\phi^\dagger \phi)$	$\mathcal{O}_{BB} = -\frac{g'^2}{4} (\phi^\dagger \phi)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

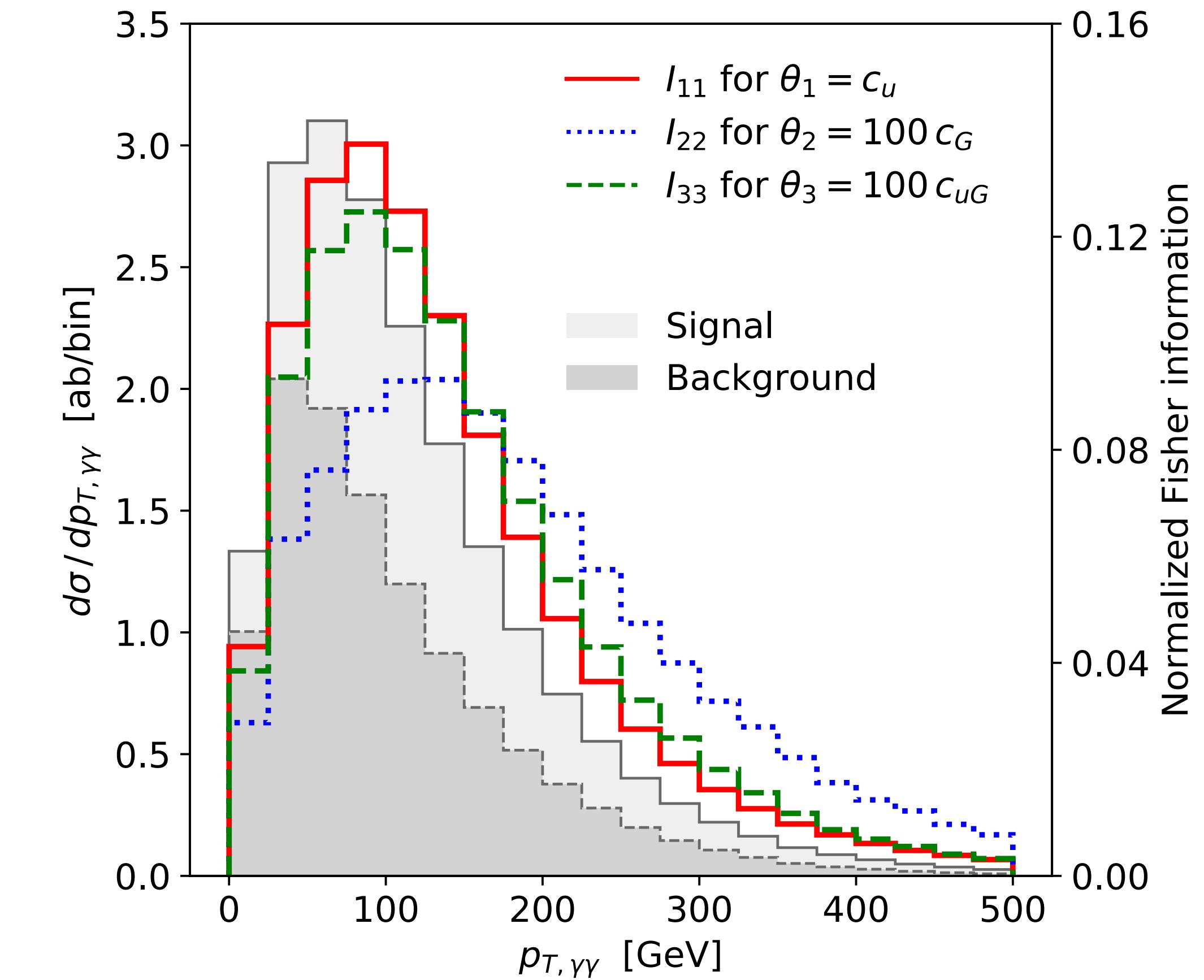
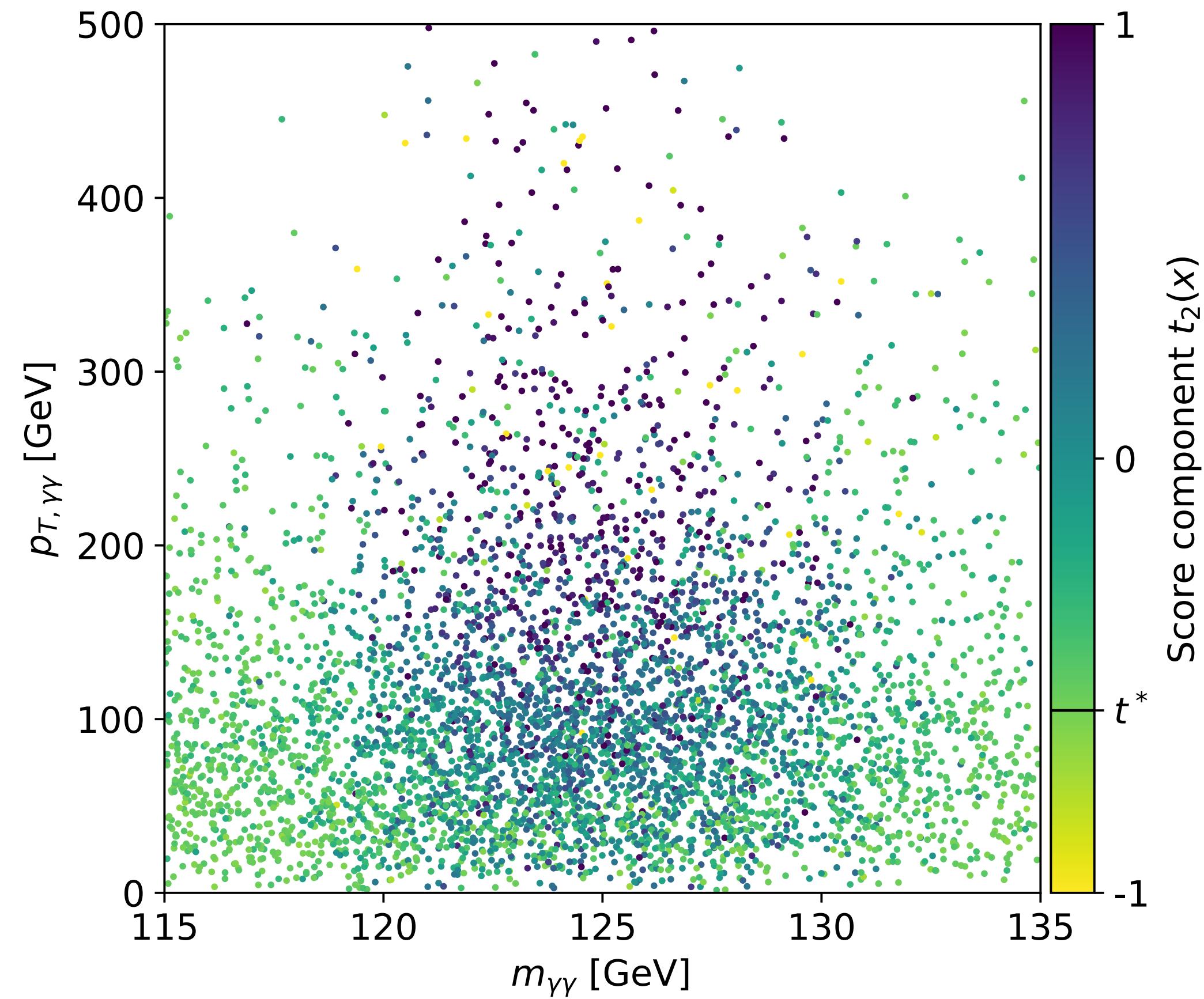
Proof of concept: more results



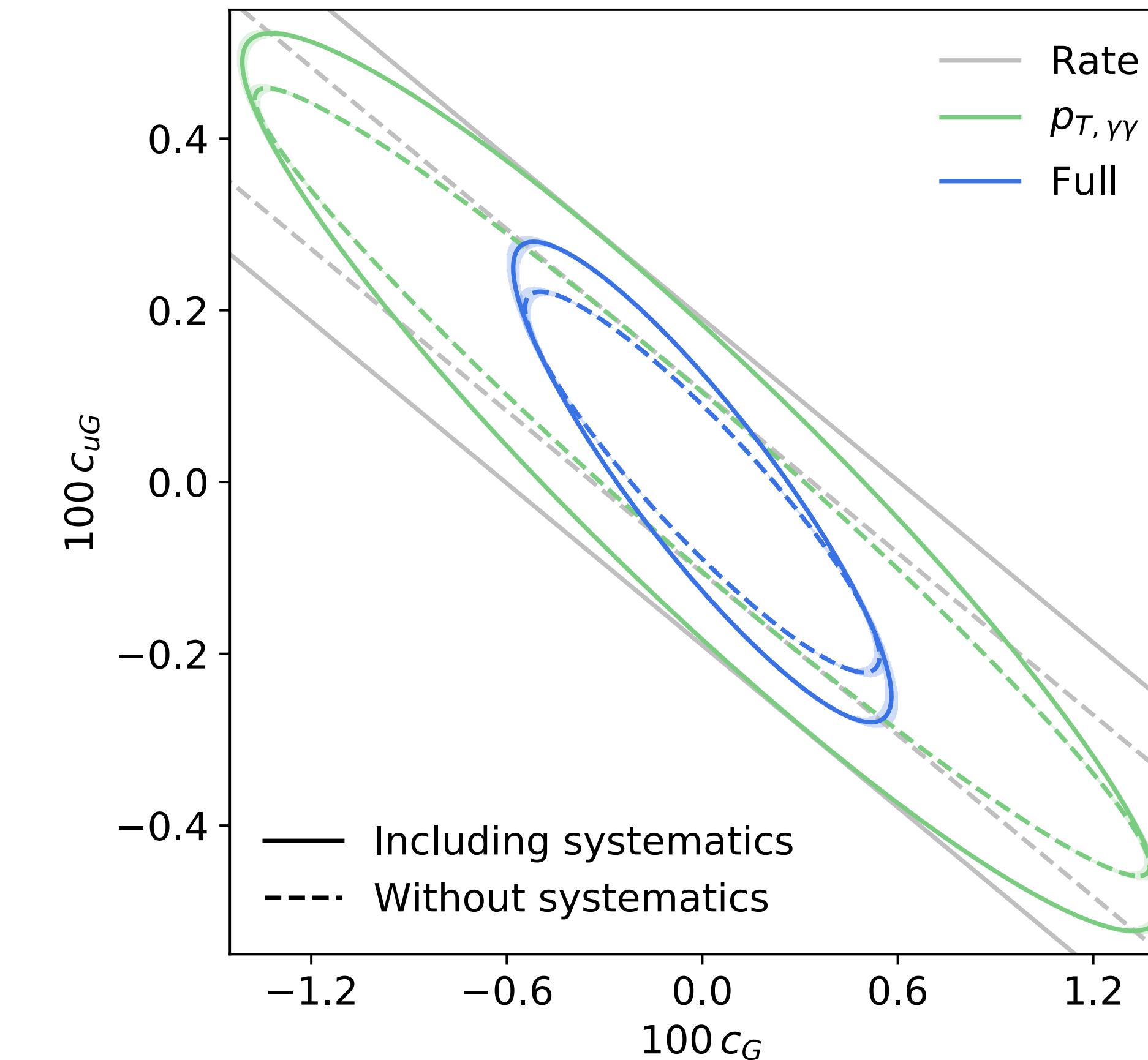
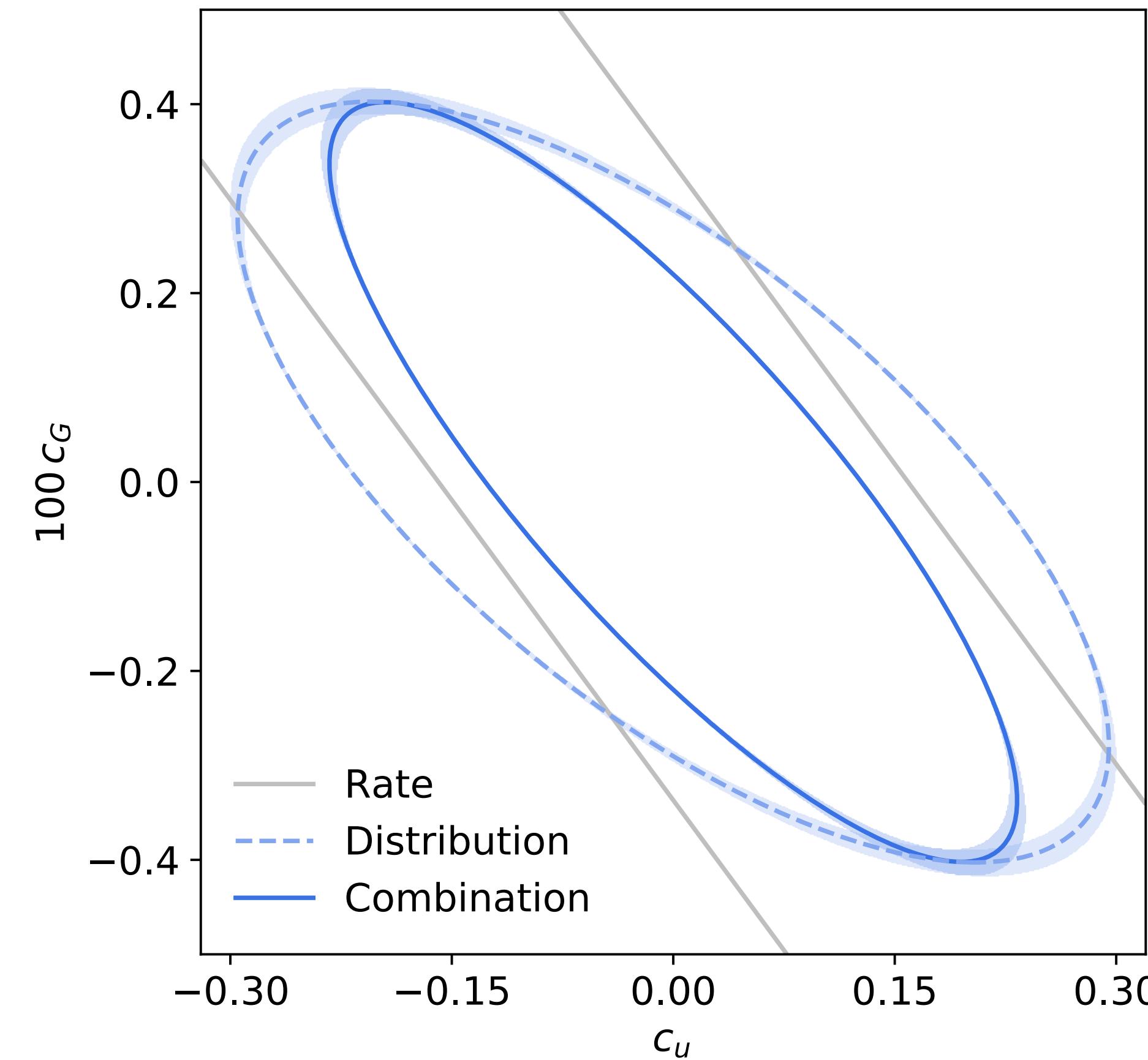
ttH: 1D illustration



ttH: score and information vs pT and m

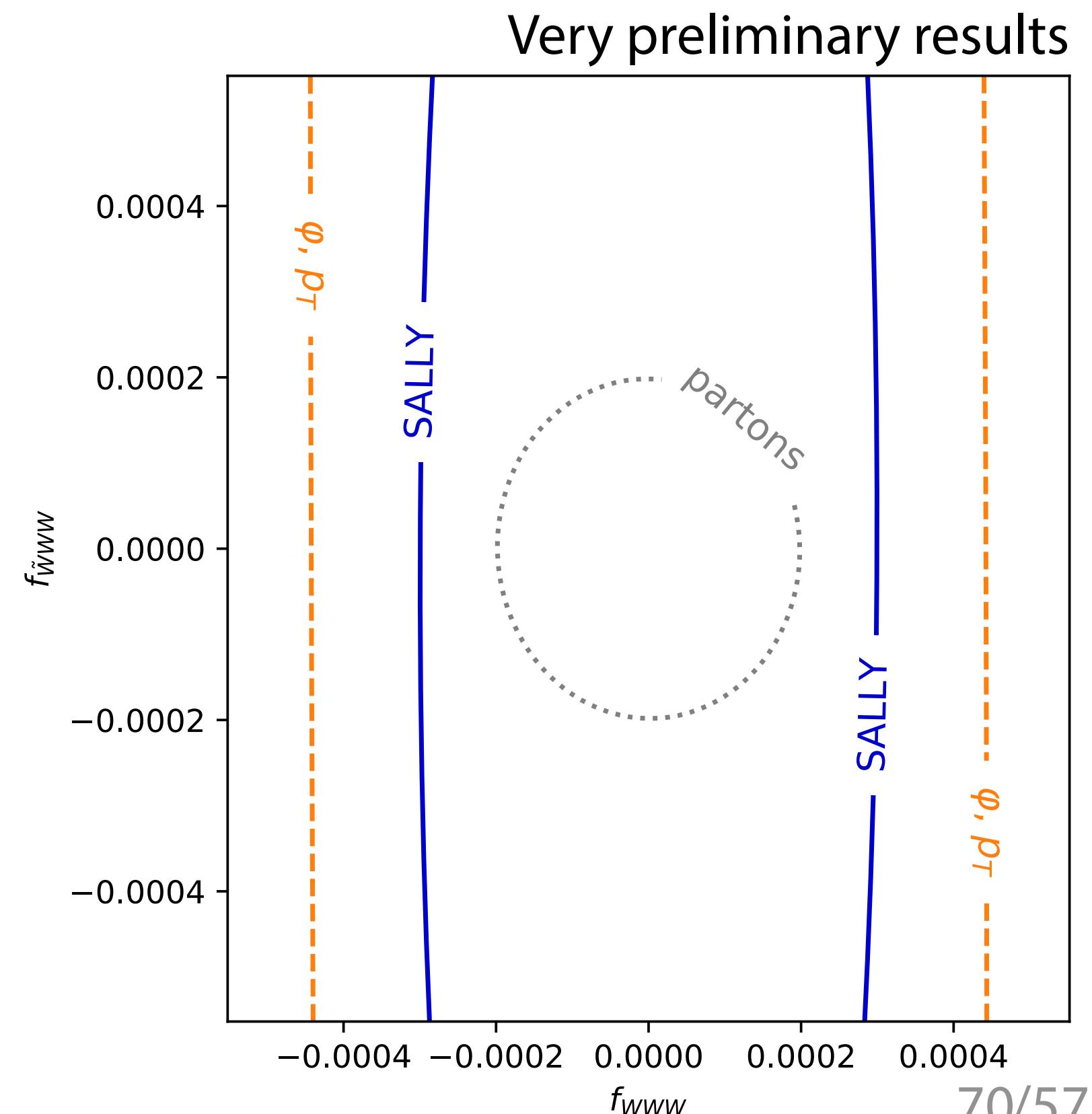
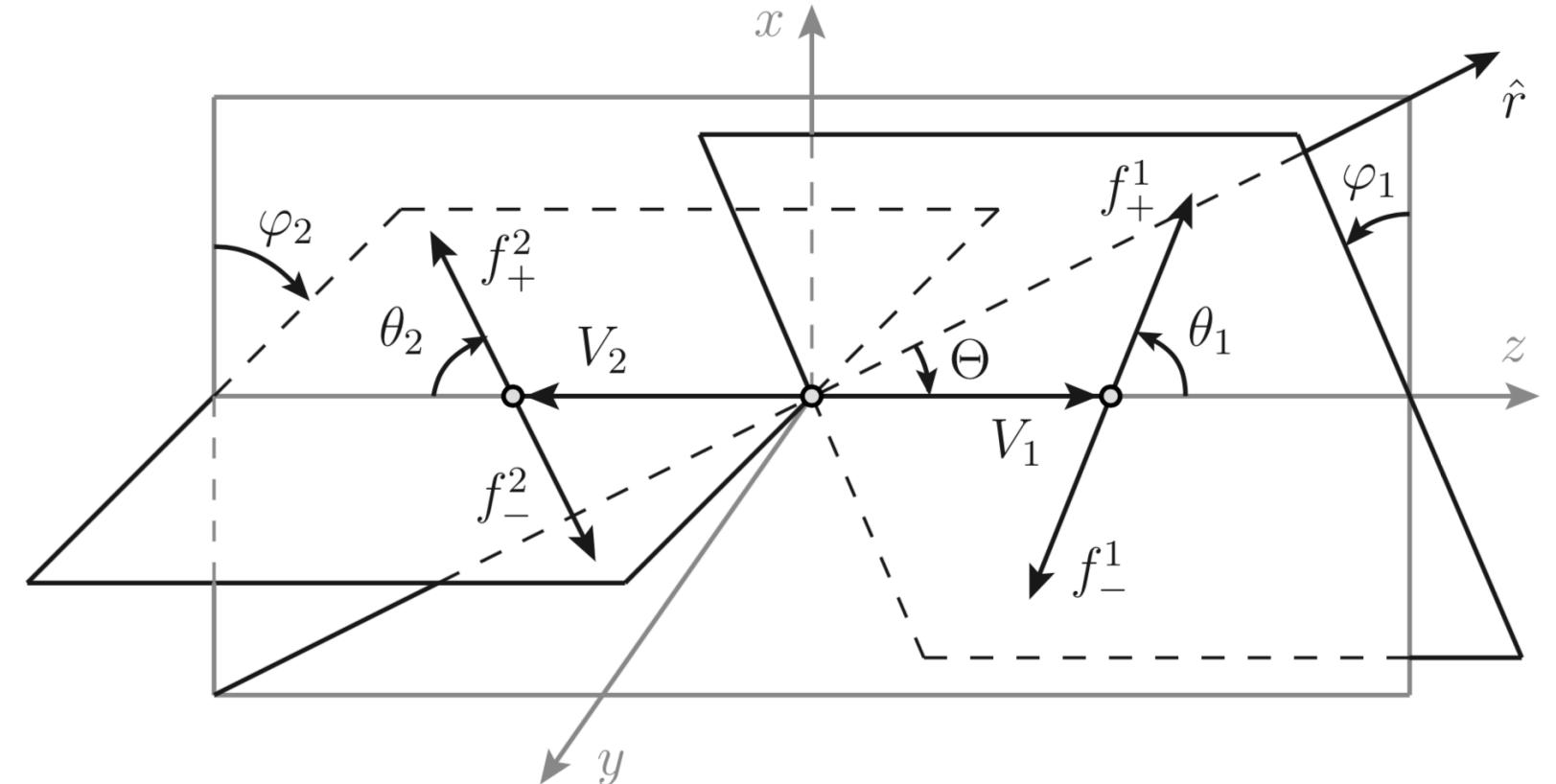


ttH: more information results

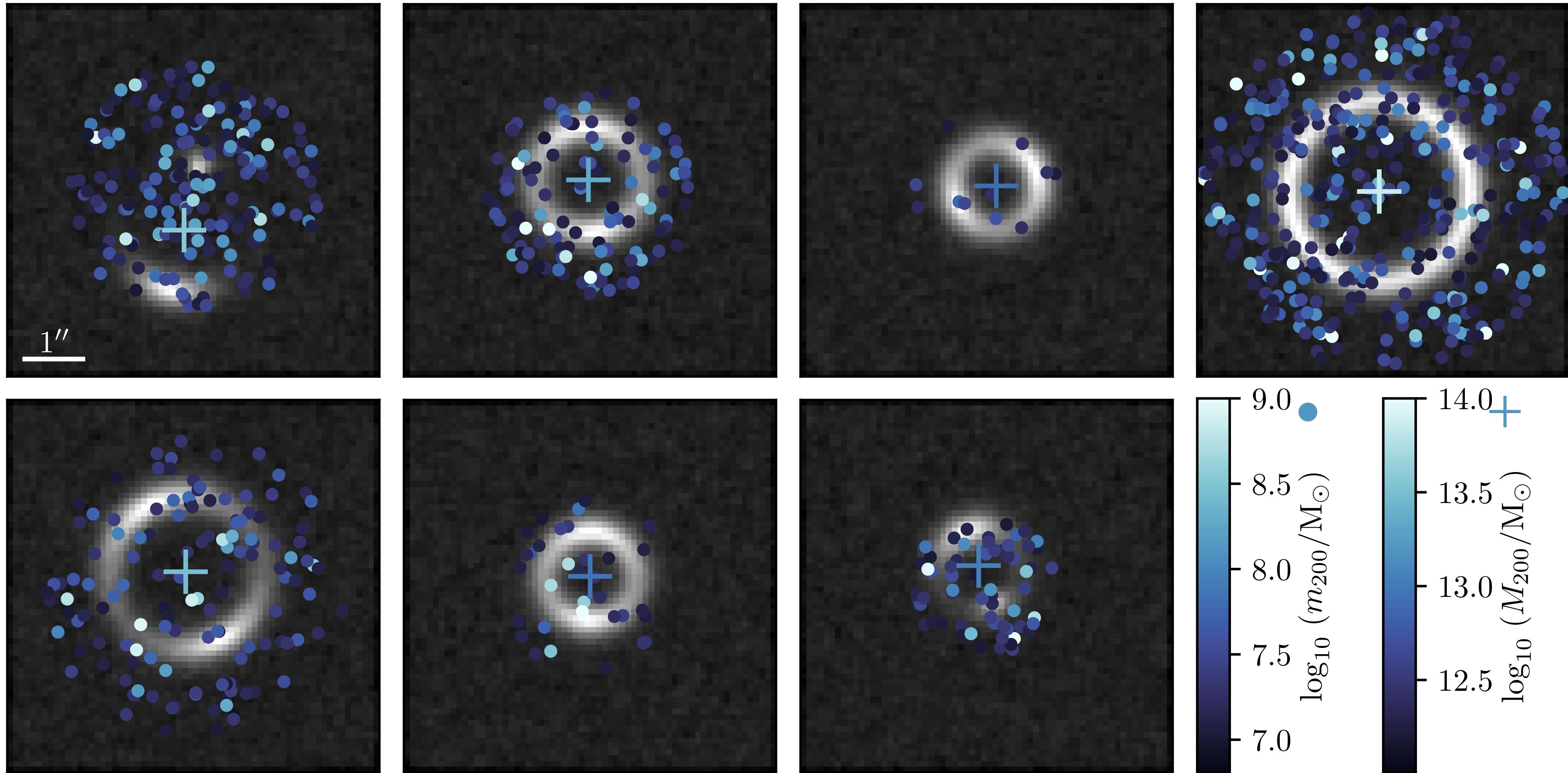


Diboson production

- In inclusive observables, the interference between SM and new physics amplitudes vanishes
⇒ Reduced sensitivity to new physics
- “Diboson interference resurrection”: an **angular variable** φ 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 $W\gamma \rightarrow \ell\nu \gamma$
[JB, K. Cranmer, M. Farina, F. Kling, D. Pappadopulo, J. Ruderman in progress]
- Preliminary results: we can extract more information when we **analyze events with SALLY** than with **histograms of φ** and standard observables



Strong lensing: simulated images



Strong lensing: expected posterior

