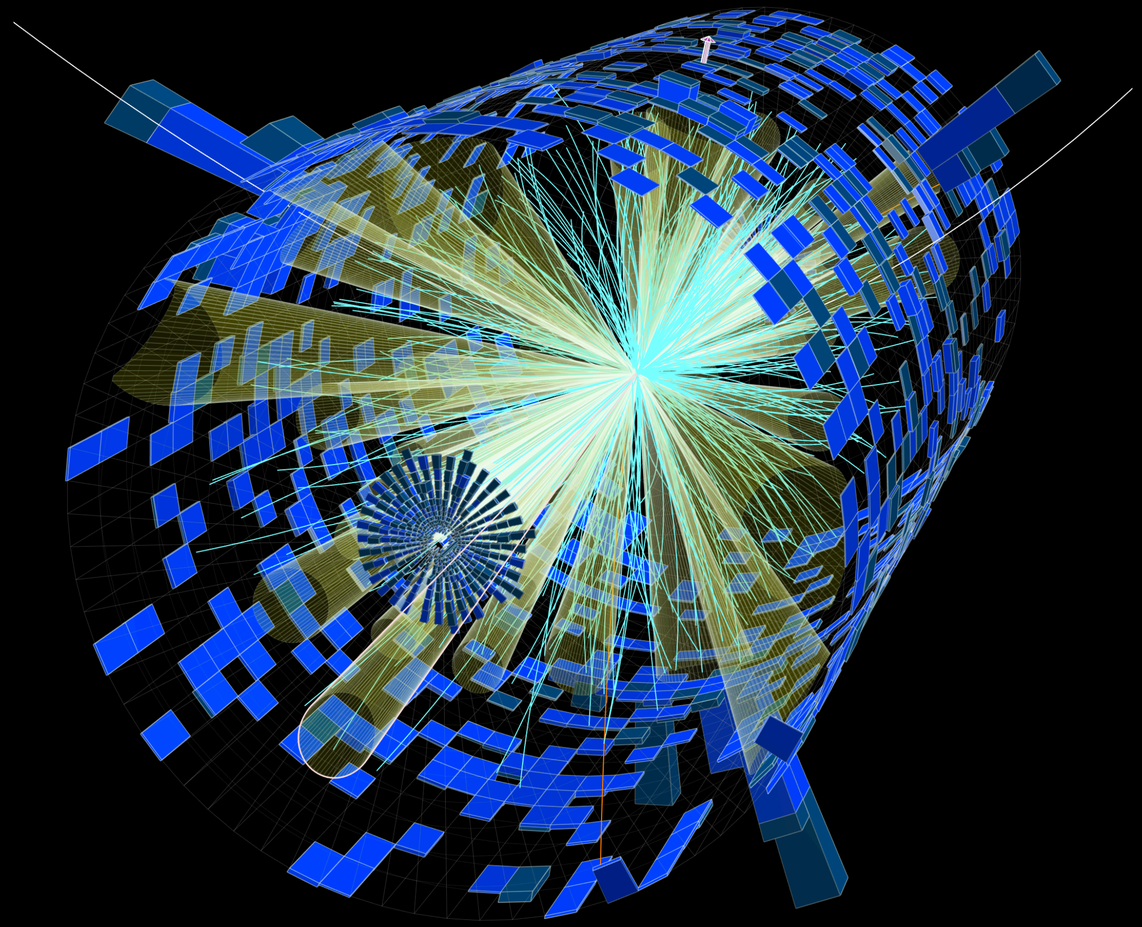
NYU Center for Data Science

Center for Cosmology and particle physics

# CONSTRAINING EFTS WITH MACHINE LEARNING

## LECTURE 2



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

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 😃 )

# The frontier of simulation-based inference

Johann Brehmer

New York University

Machine Learning at LHC workshop, Nagoya University
February 5, 2020

Particle physics processes do not have a tractable likelihood function.

# Modelling particle physics processes

Theory
parameters
$\theta$

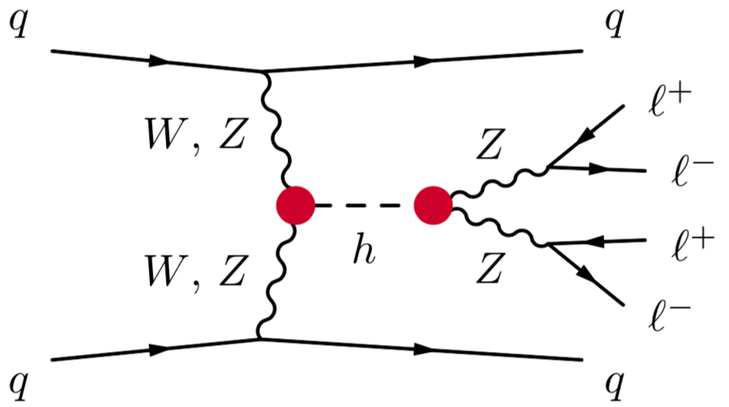
Evolution

# Modelling particle physics processes

Latent variables

Parton-level momenta $z_p$

Theory parameters $\theta$

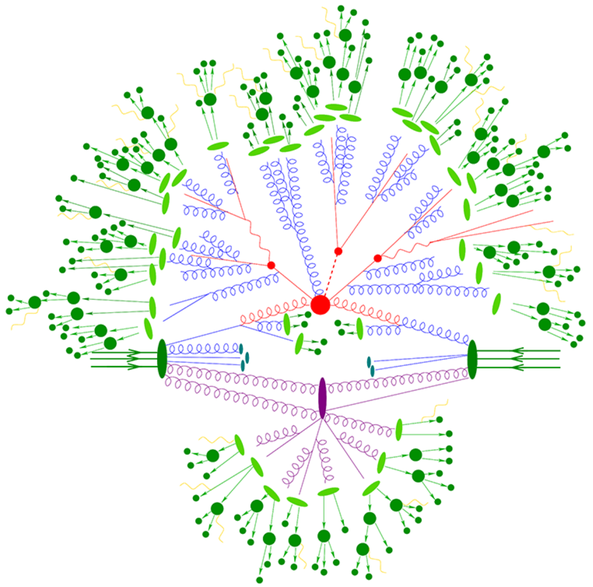



Evolution

# Modelling particle physics processes

Latent variables

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



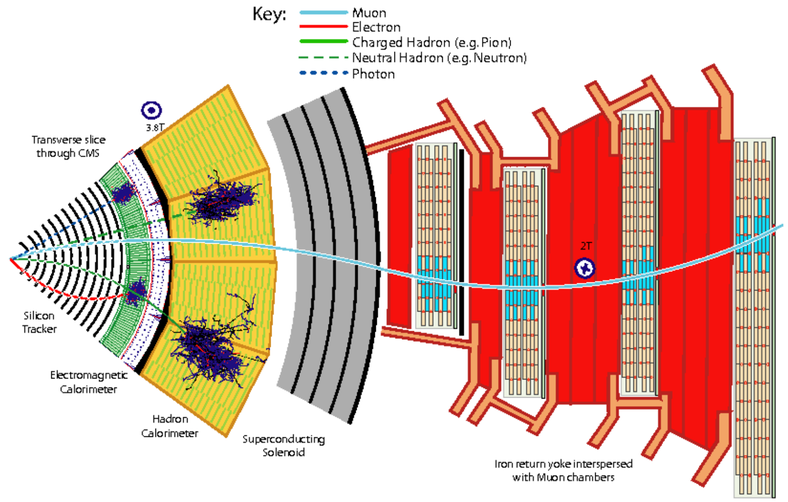
[F. Krauss]

Evolution

# Modelling particle physics processes

Latent variables

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



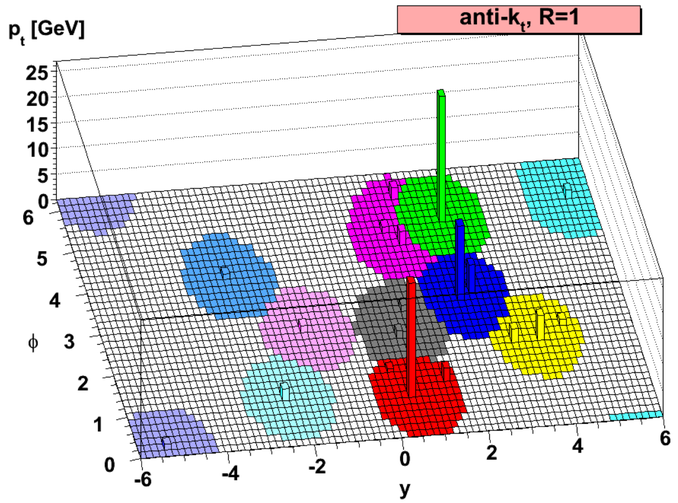

[CMS]

Evolution

# Modelling particle physics processes

Latent variables

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



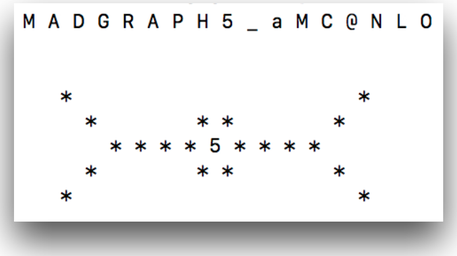

[M. Cacciari, G. Salam, G. Soyez 0802.1189]

Evolution

# Modelling particle physics processes

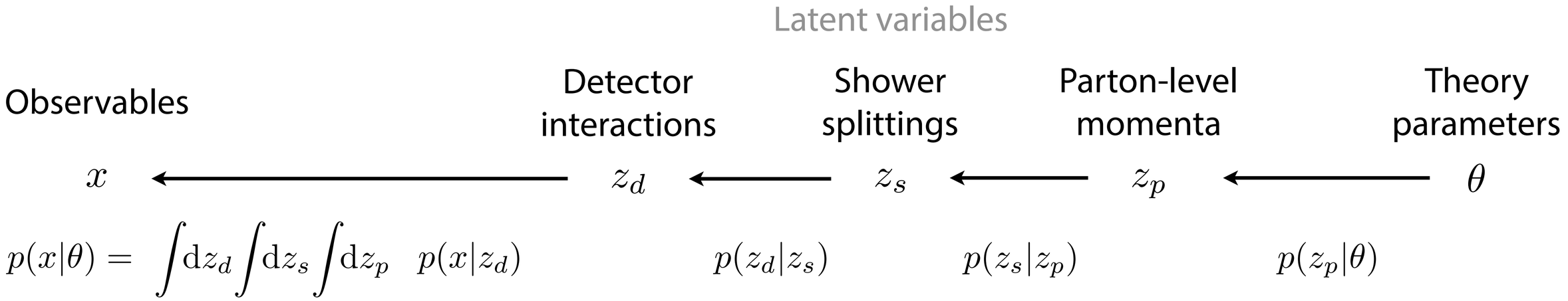
Latent variables

| Observables | Detector interactions | Shower splittings | Parton-level momenta | Theory parameters |
|---|---|---|---|---|
| $x$ | $z_d$ | $z_s$ | $z_p$ | $\theta$ |
| Sample from | $p(x\|z_d)$ | $p(z_d\|z_s)$ | $p(z_s\|z_p)$ | $p(z_p\|\theta)$ |

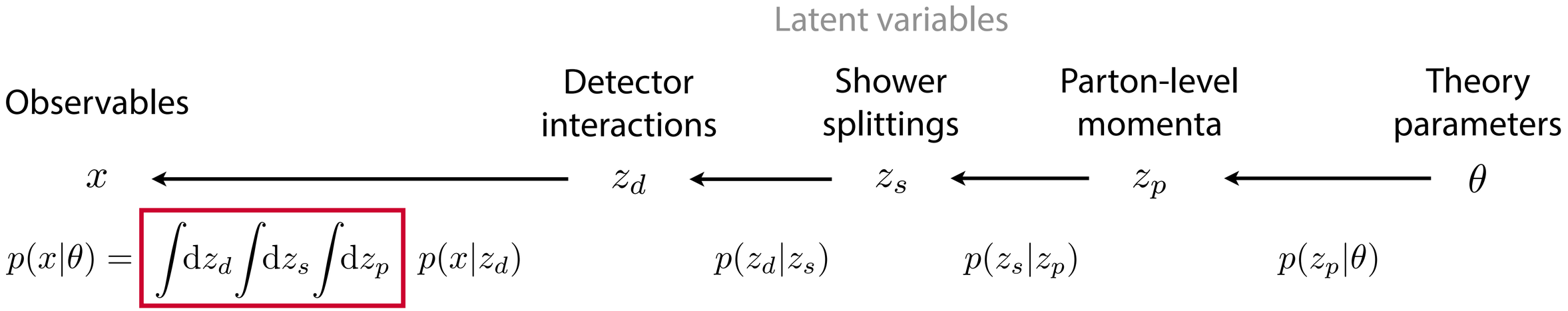

Prediction (simulation)

# Modelling particle physics processes




Inference

# Modelling particle physics processes

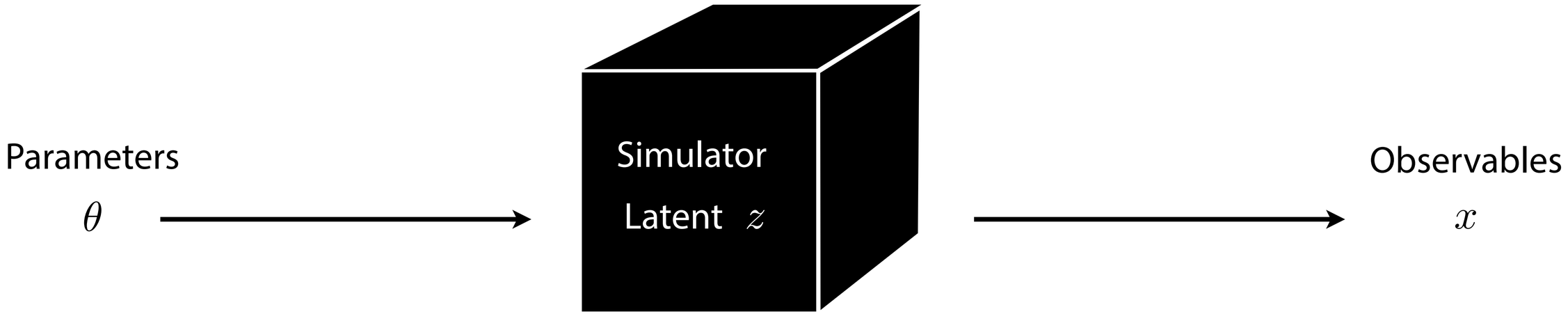



It's infeasible to calculate the integral over this enormous space!

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

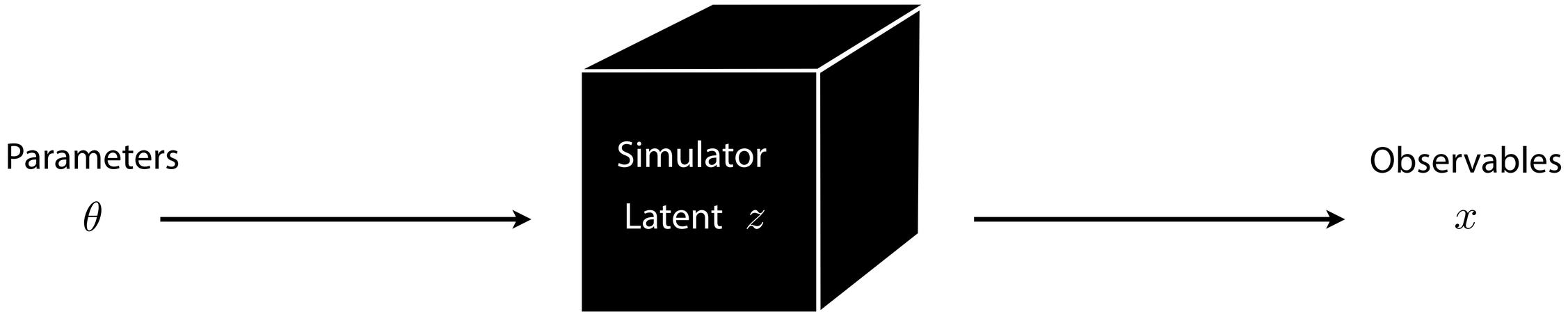
Inference

# Simulation-based (“likelihood-free”) inference




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

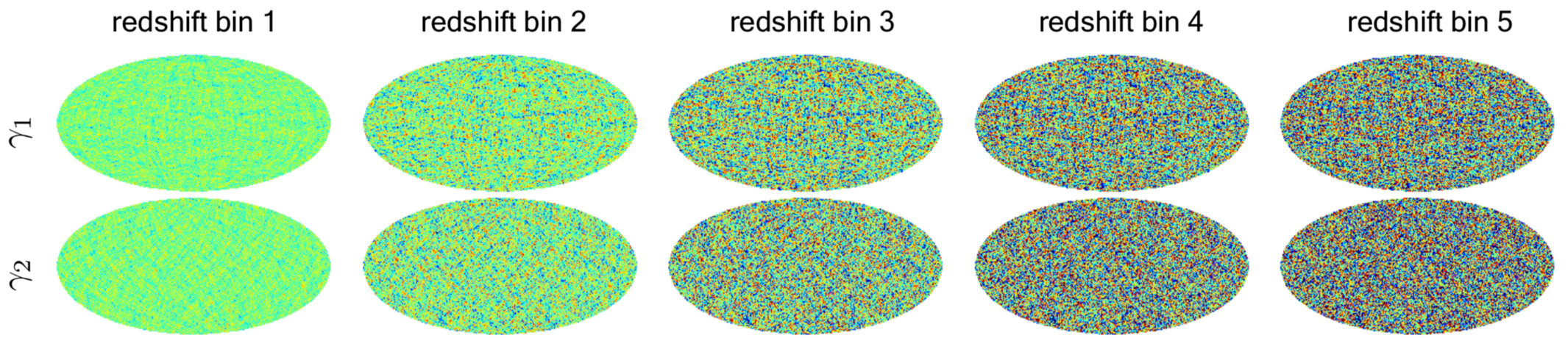
# Simulation-based (“likelihood-free”) inference




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

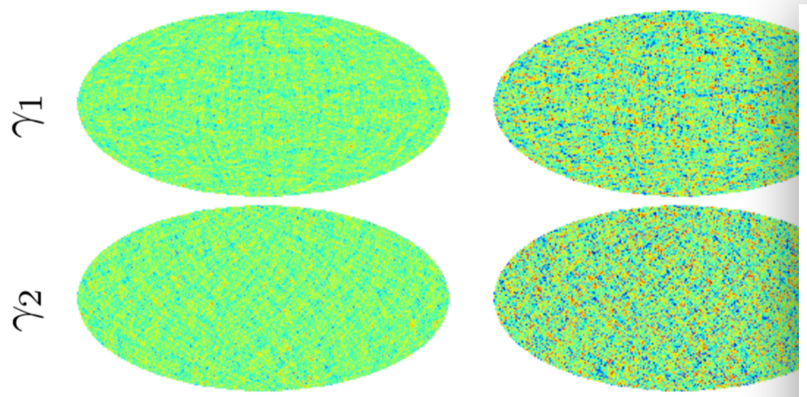
# It’s not just particle physics!

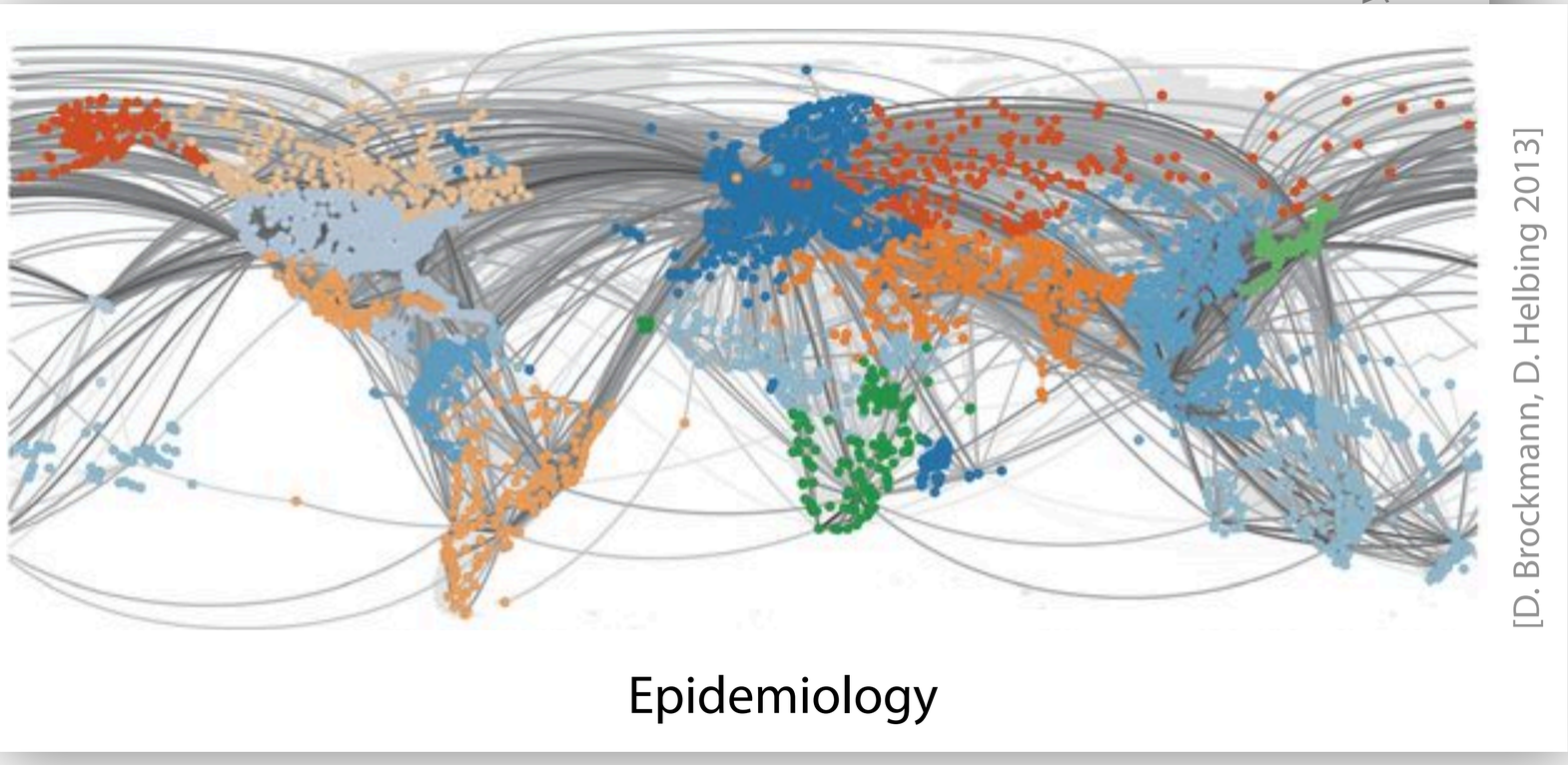



[J. Alsing, T. Charnock, S. Feeney, B. Wandeelt 1903.00007]

Cosmology

# It’s not just particle physics!

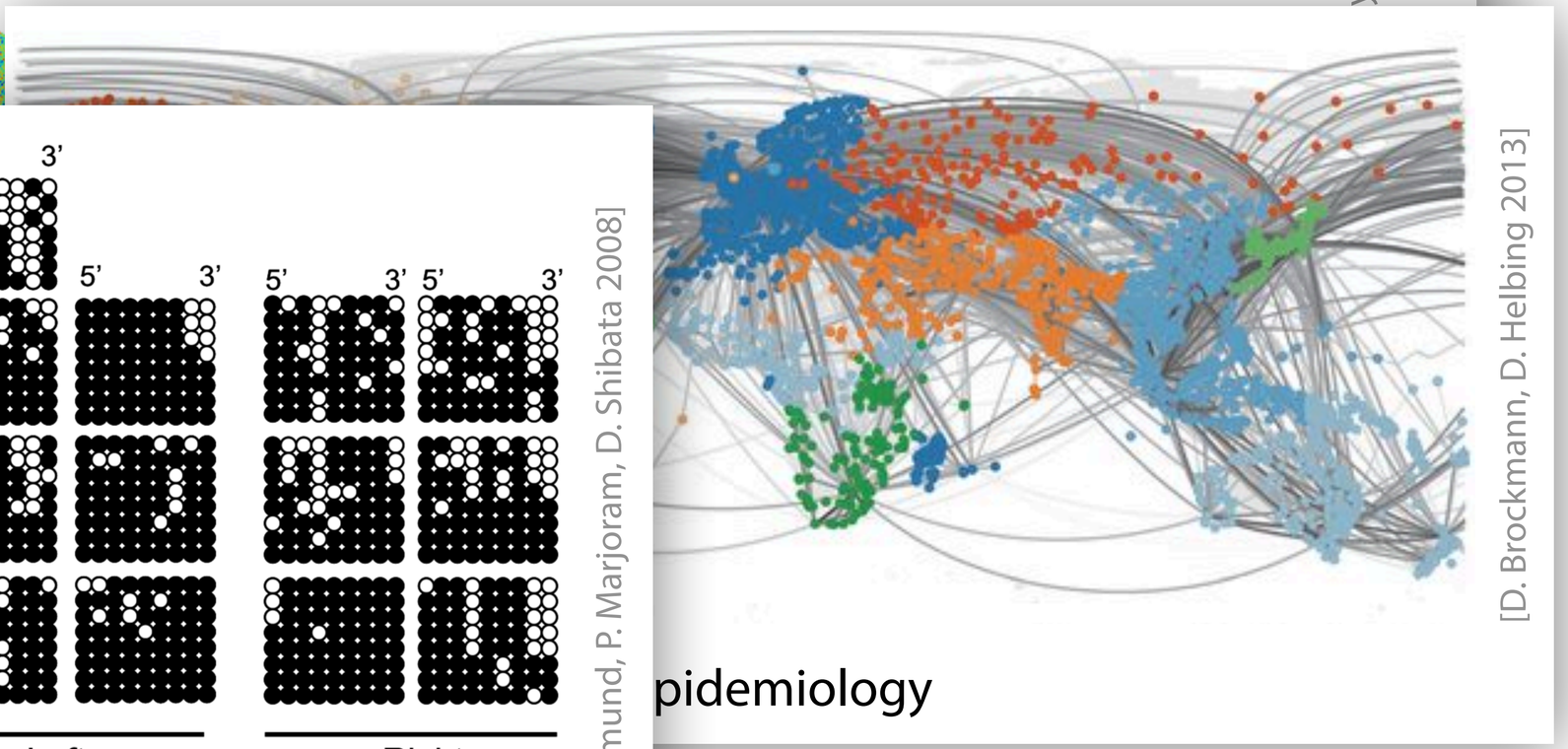


Epidemiology

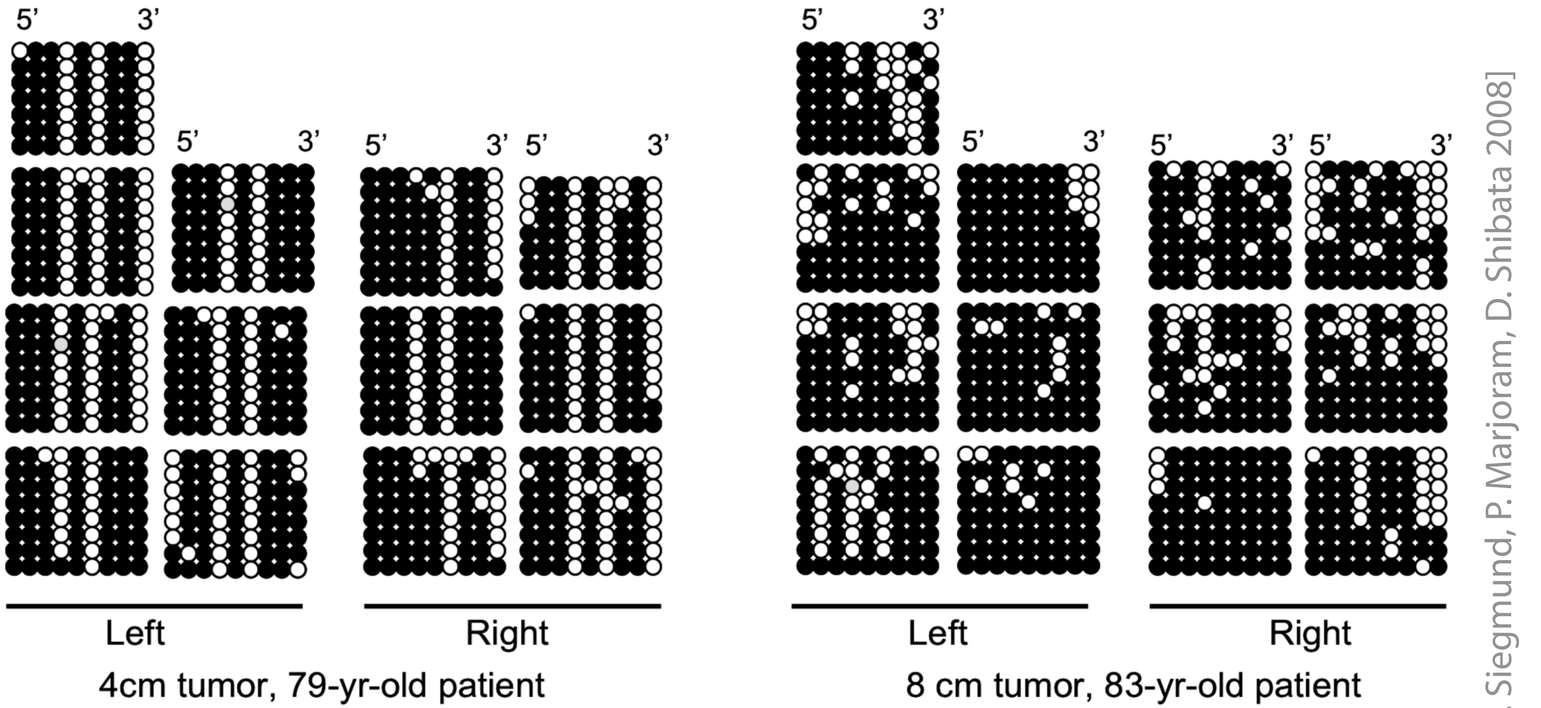
[D. Brockmann, D. Helbing 2013]

# It’s not just particle physics!

[D. Brockmann, D. Helbing 2013]

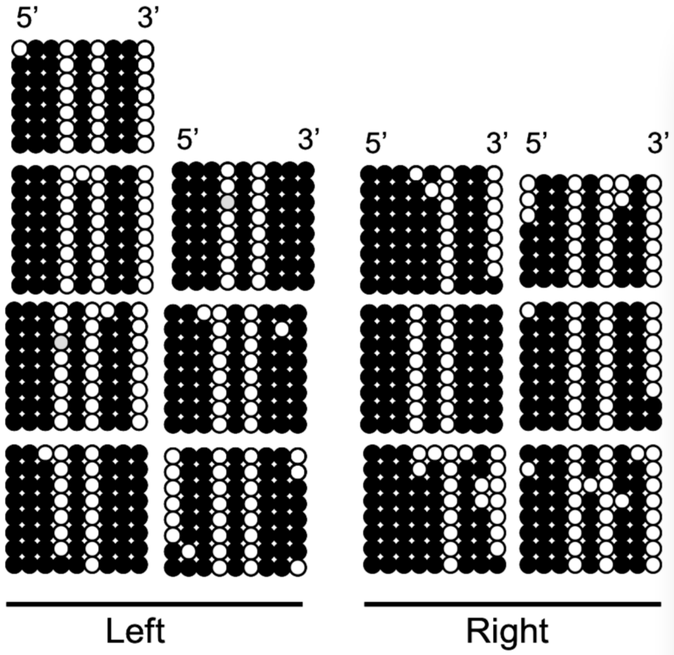
pidemiology

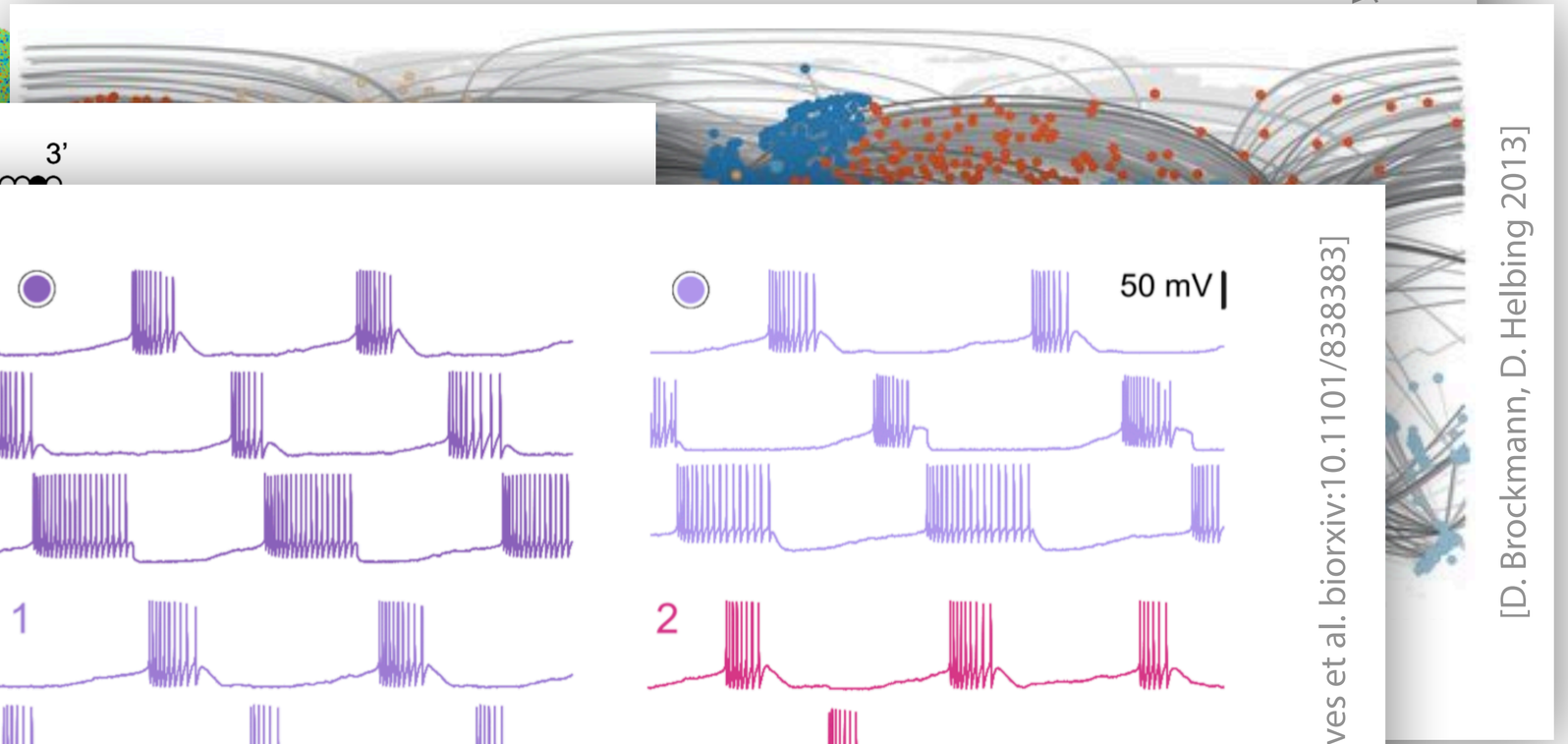



[K. Siegmund, P. Marjoram, D. Shibata 2008]

Genetics

# It’s not just particle physics!

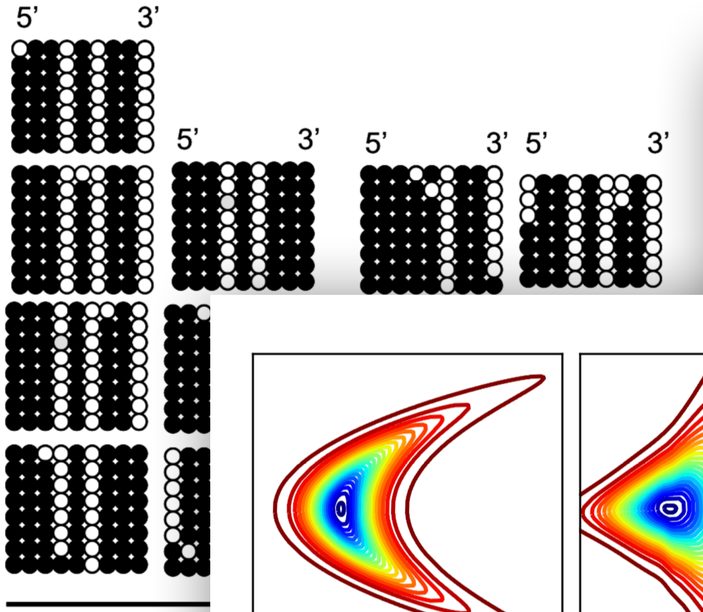


[D. Brockmann, D. Helbing 2013]

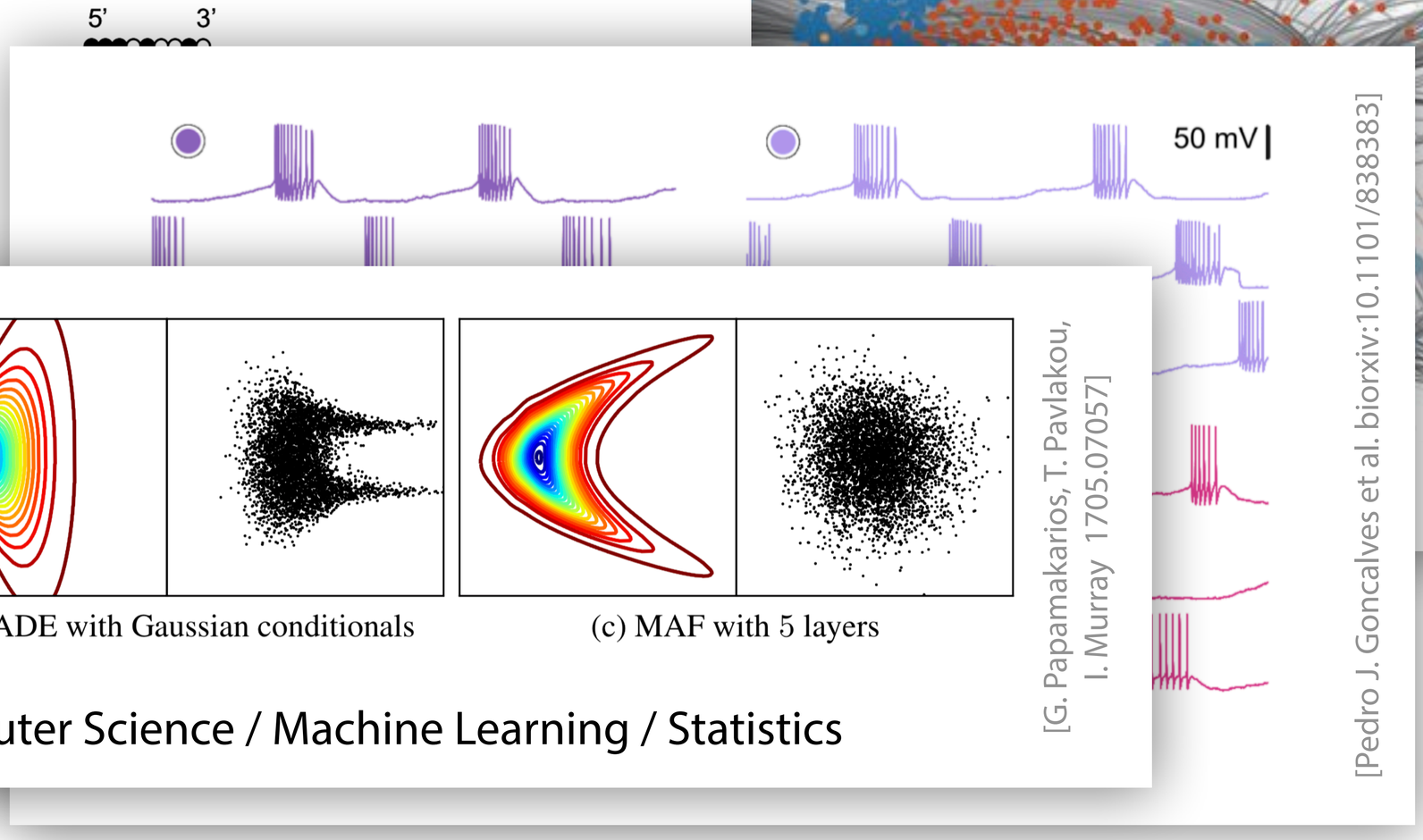



[Pedro J. Goncalves et al. biorxiv:10.1101/838383]

# It’s not just particle physics!


[D. Brockmann, D. Helbing 2013]




[Pedro J. Goncalves et al. biorxiv:10.1101/838383]



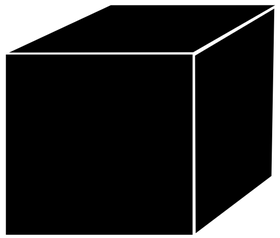

[G. Papamakarios, T. Pavlakou, I. Murray 1705.07057]

Computer Science / Machine Learning / Statistics

# 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 $\theta_i$ of your choice

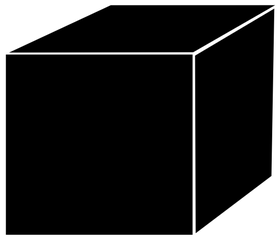


- observed data $x_{\mathrm{obs}} \sim p(x_{\mathrm{obs}}|\theta_{\mathrm{true}})$
- prior belief $p(\theta)$

# 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 $\theta_i$ of your choice



- observed data $x_{\mathrm{obs}} \sim p(x_{\mathrm{obs}}|\theta_{\mathrm{true}})$
- prior belief $p(\theta)$

Goals: estimate either…

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

… 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?
- 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?
- 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?
- ~ normalized differential xsec
- 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?
- For simplicity, we ignore this part in this talk

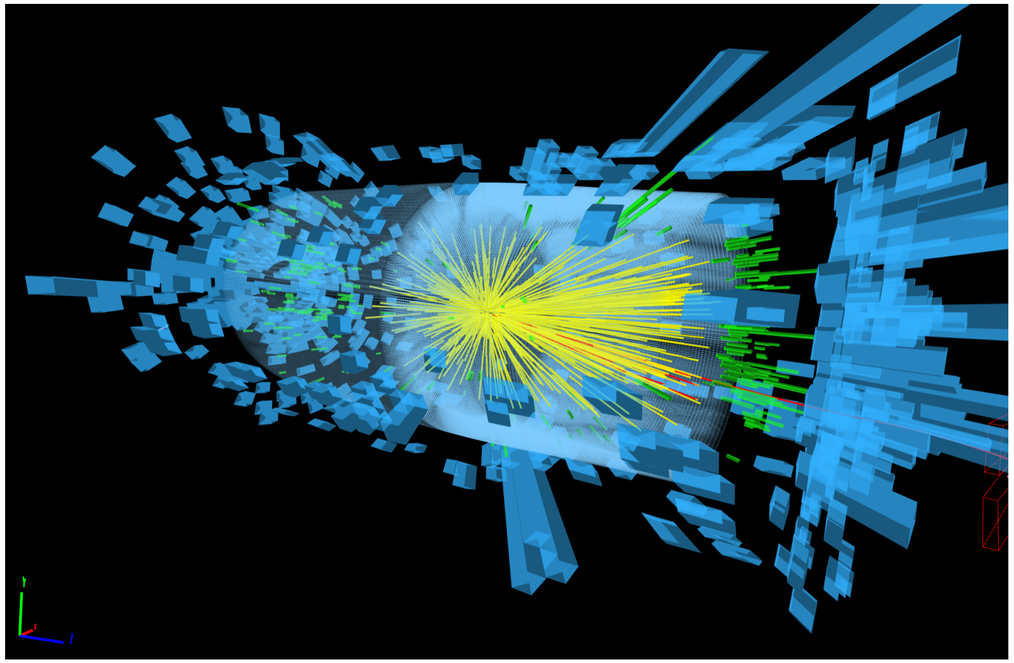
Kinematic term for each event:

- How likely is it that an event looks like it does?
- ~ normalized differential xsec
- 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.

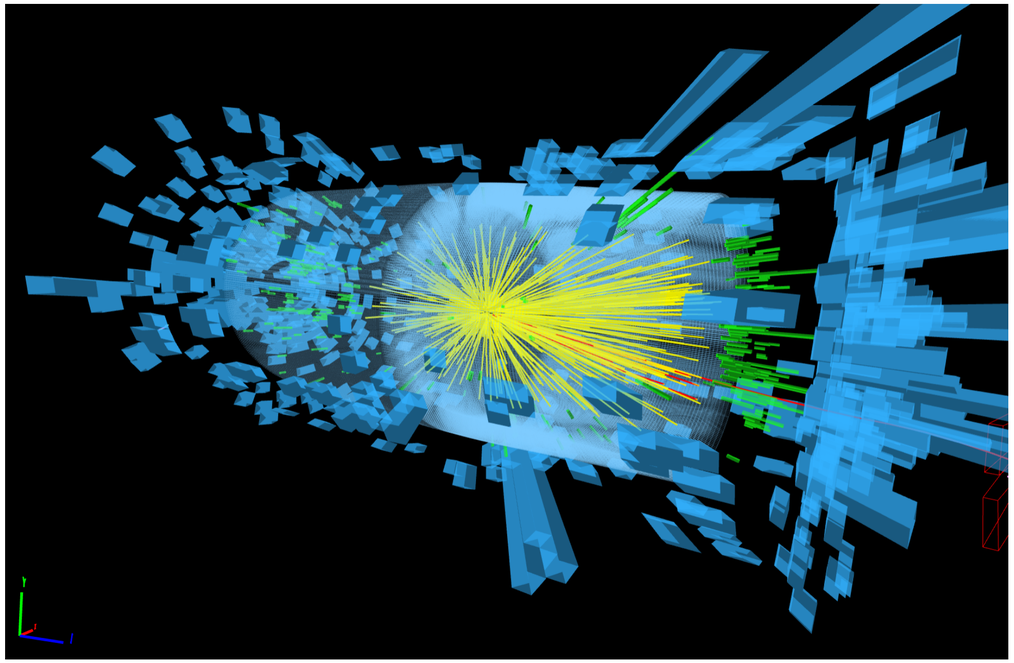
# Solve it with summary statistics

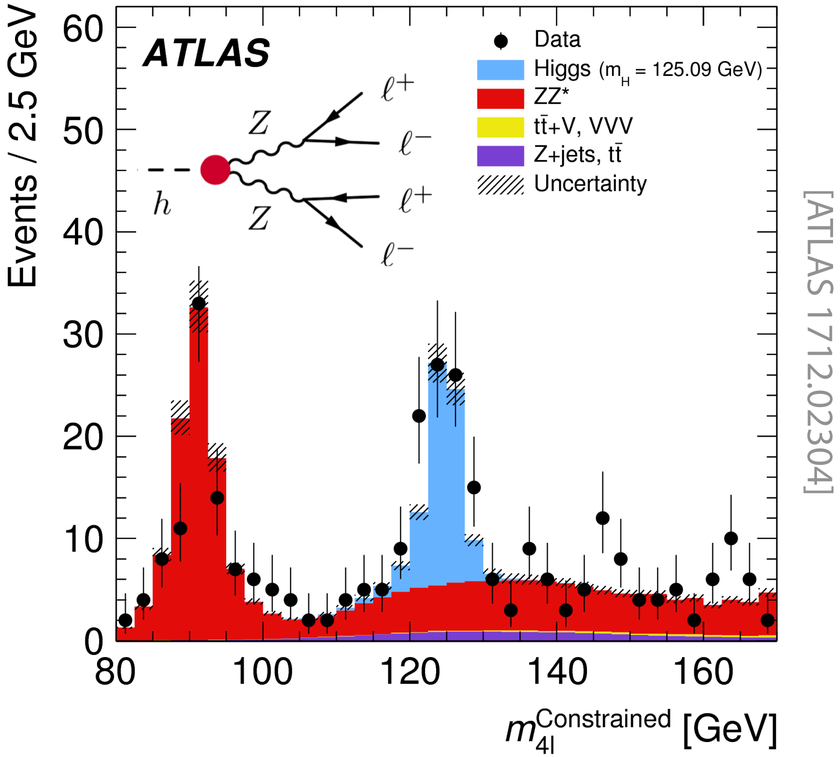


High-dimensional event data $x$

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

# Solve it with summary statistics






High-dimensional event data $x$

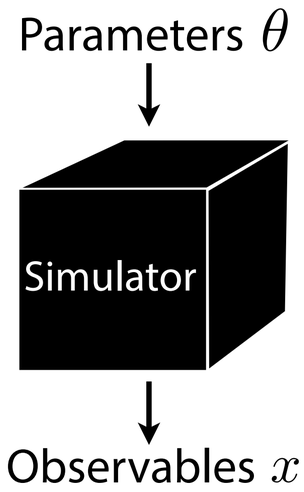
$p(x|\theta)$ cannot be calculated

One or two summary statistics $x'$

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

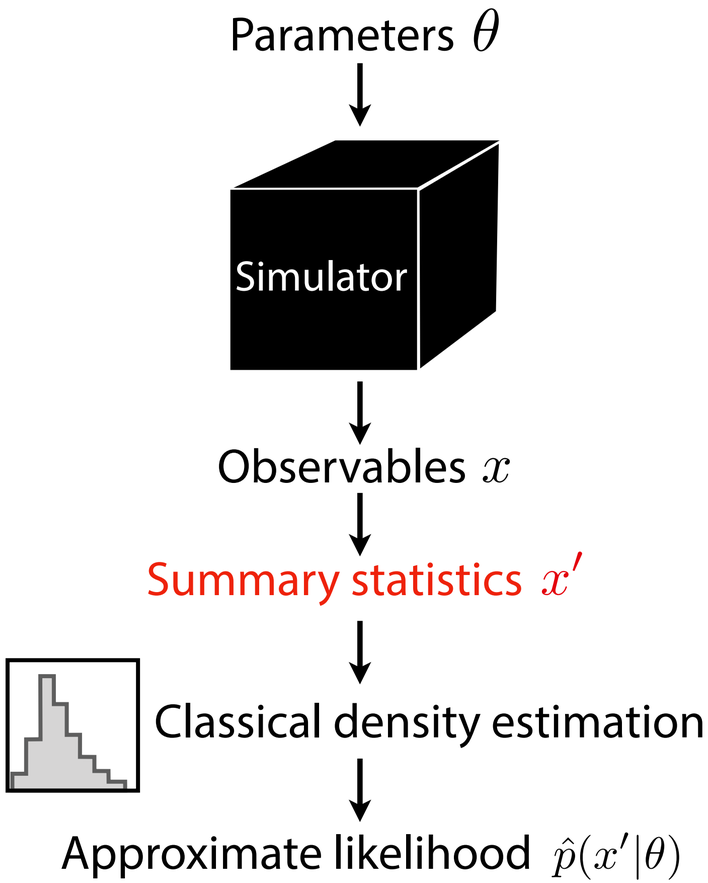
# Inference by estimating the likelihood

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



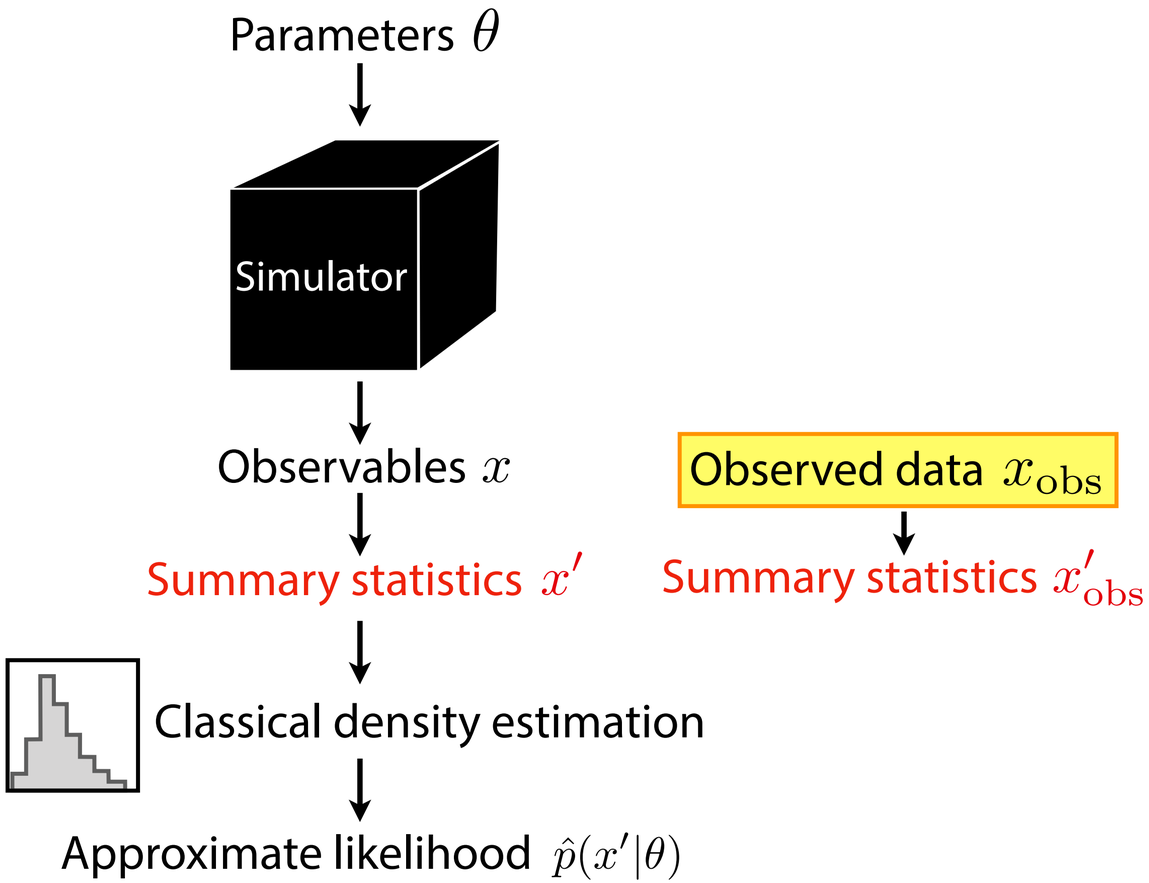
# Inference by estimating the likelihood

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



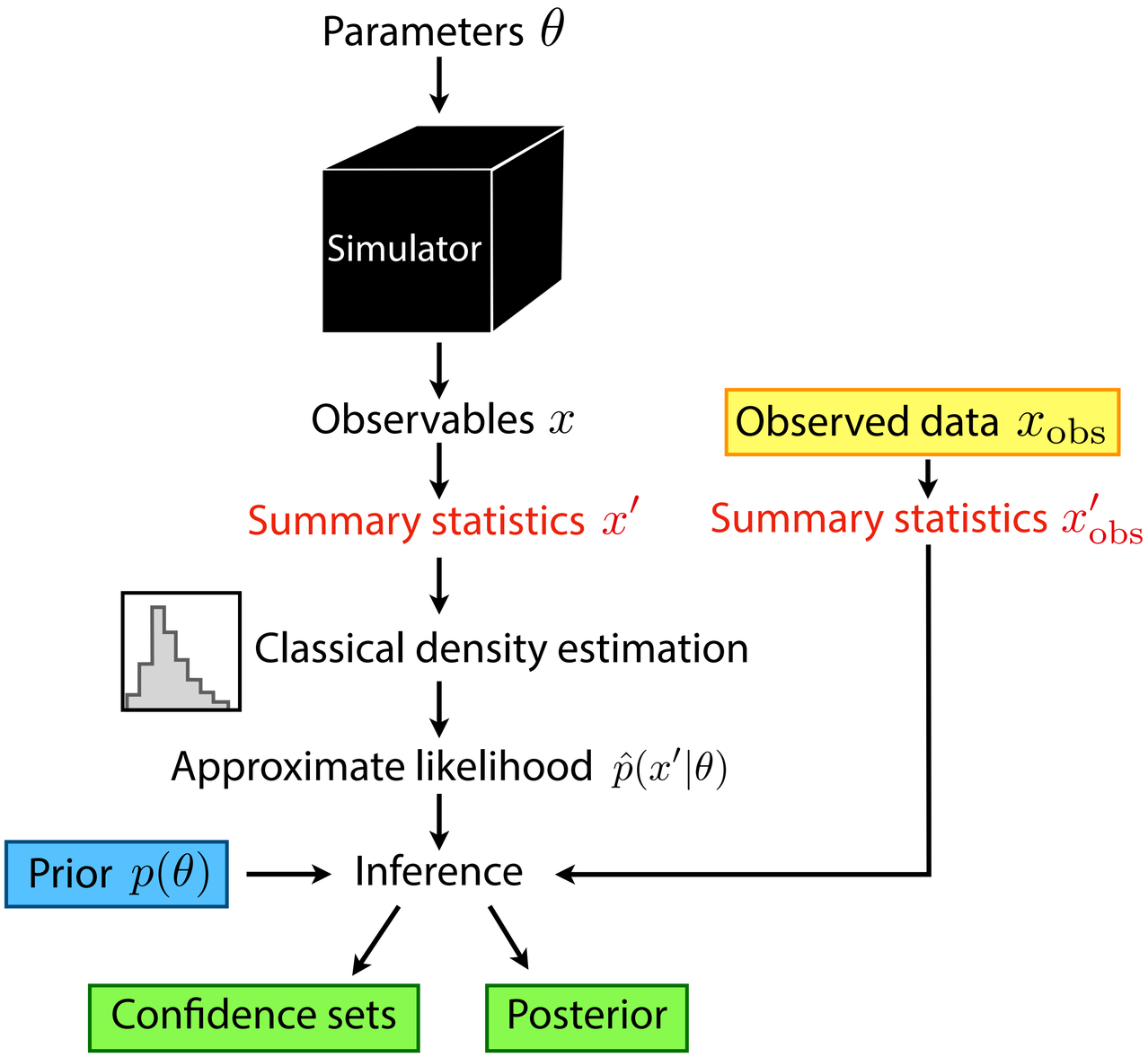
# Inference by estimating the likelihood

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



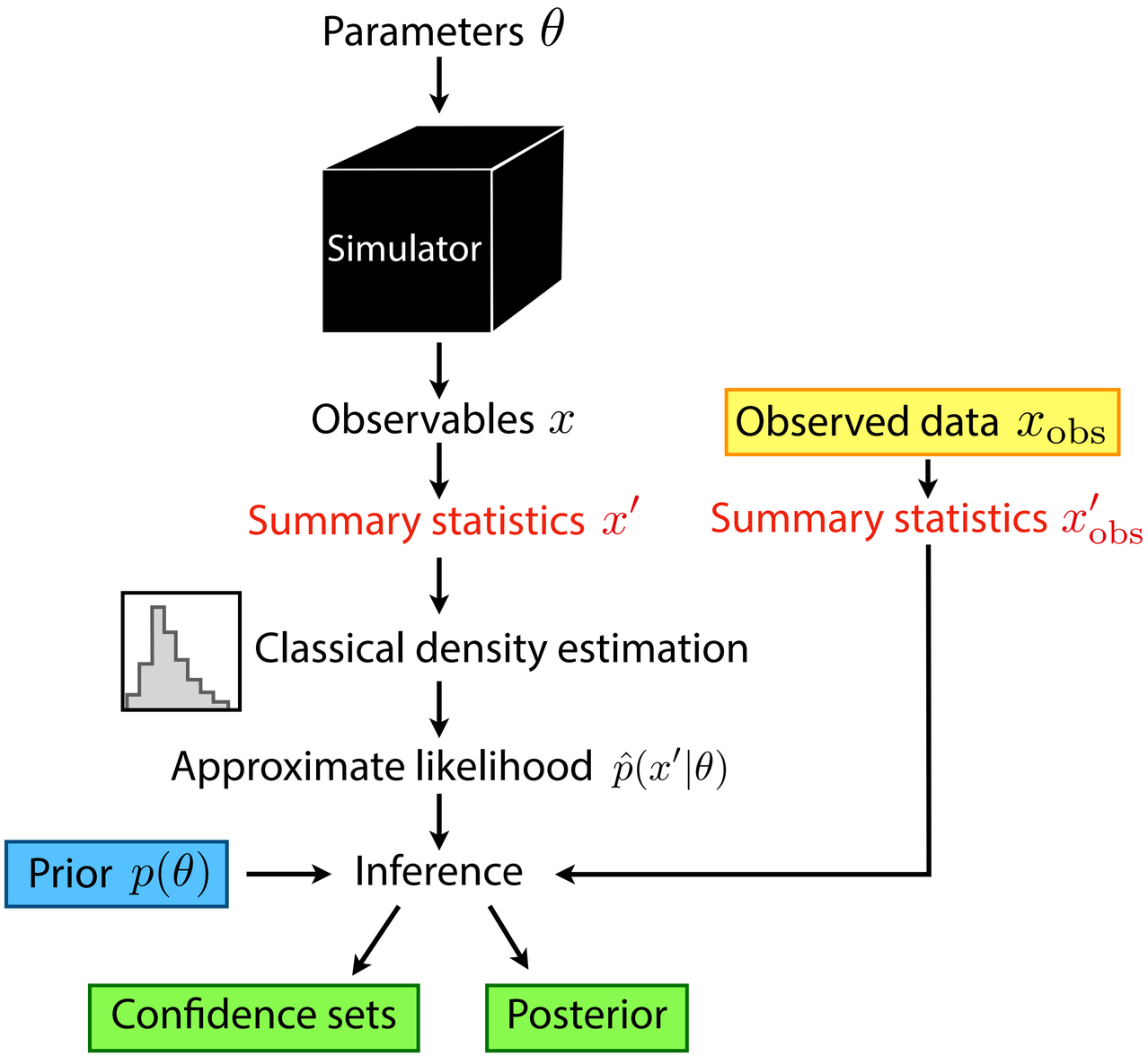
# Inference by estimating the likelihood

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



# Inference by estimating the likelihood

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




- Compression to summary statistics 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")

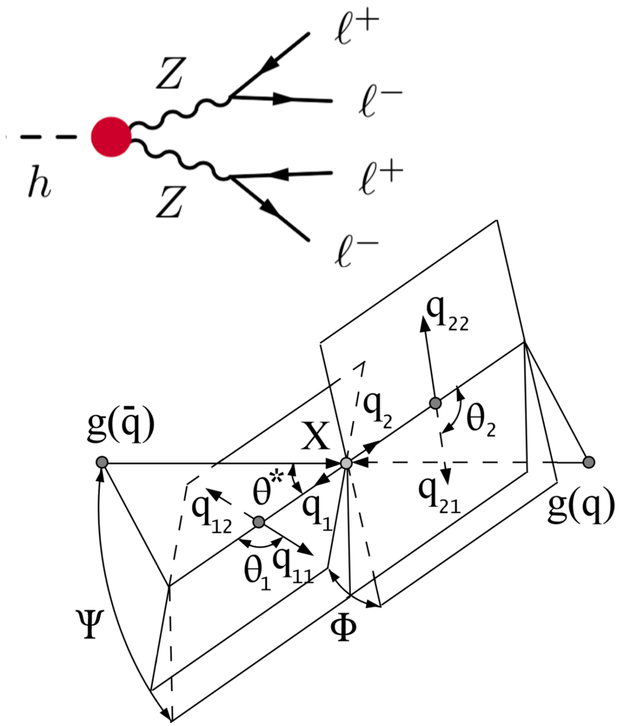
# Summary statistics for LHC measurements?

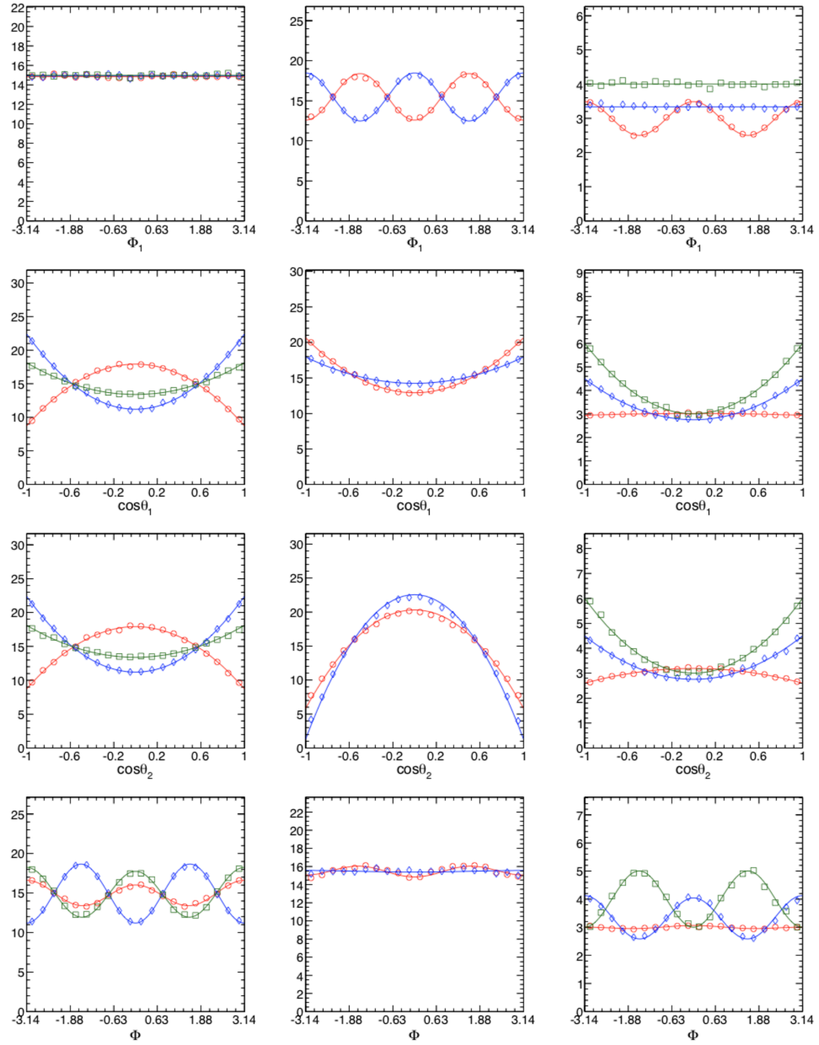
- In many LHC problems there is no single good summary statistic: compressing to any $x'$ loses information!

  [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")



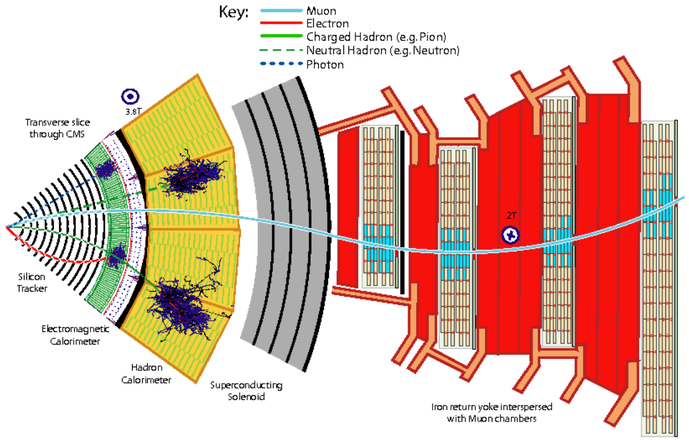




[Bolognesi et al. 1208.4018]

# Solve it by approximating the integral

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

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

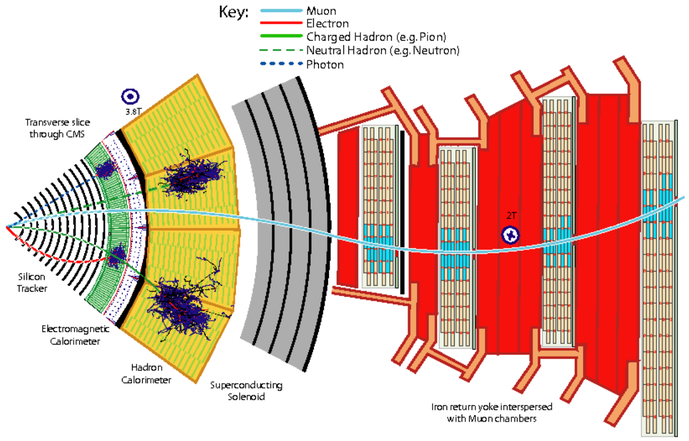



[CMS]

# Solve it by approximating the integral

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

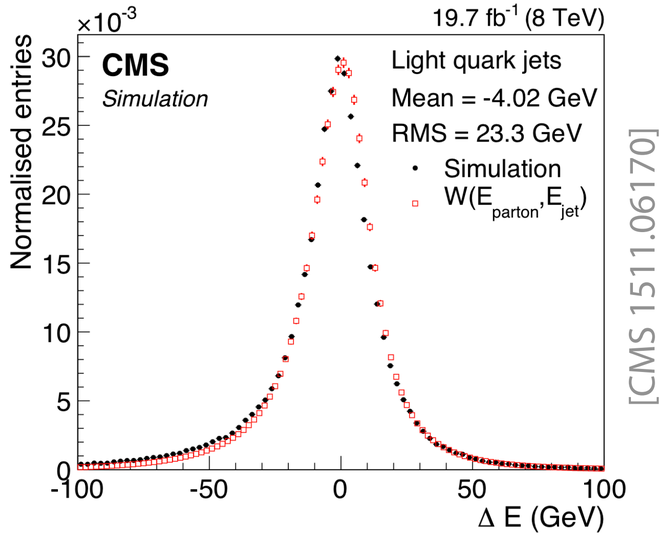
$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \; p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p(z_p|\theta)$$




[CMS]

- 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

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

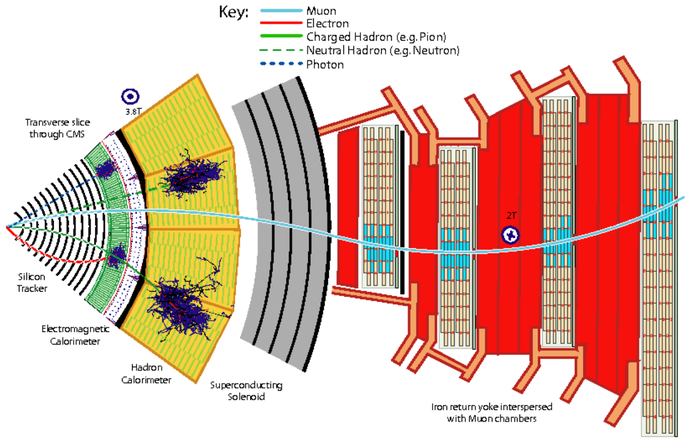



[CMS 1511.06170]

# Solve it by approximating the integral

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

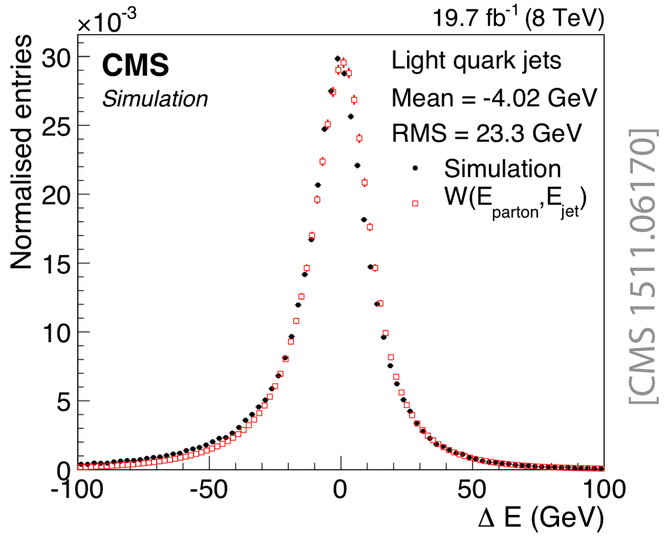
$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \; p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p(z_p|\theta)$$




[CMS]

- 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

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




[CMS 1511.06170]

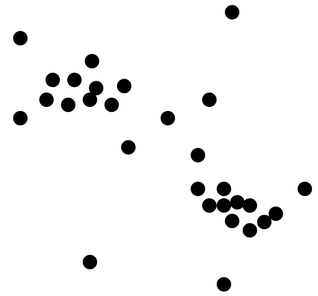
⇒ Uses matrix-element information, no summary statistics necessary, but:

- 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.

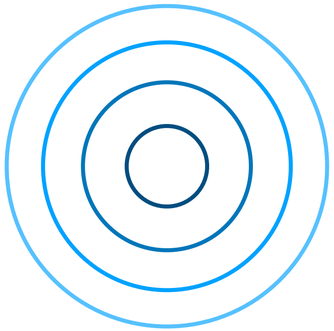
[K. Cranmer, JB, G. Louppe 1911.01429]

# High-dimensional density estimation with normalizing flows



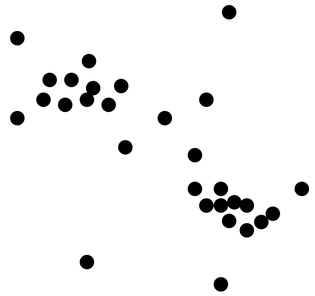
[lots of great work recently; see G. Papamakarios et al 1912.02762 for a review]

# High-dimensional density estimation with normalizing flows



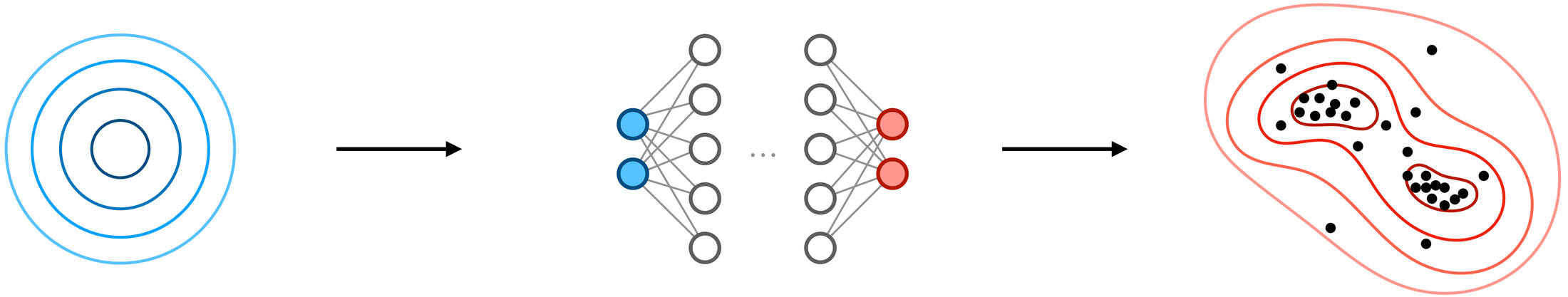
Simple base density

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



[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 $x = f(u)$

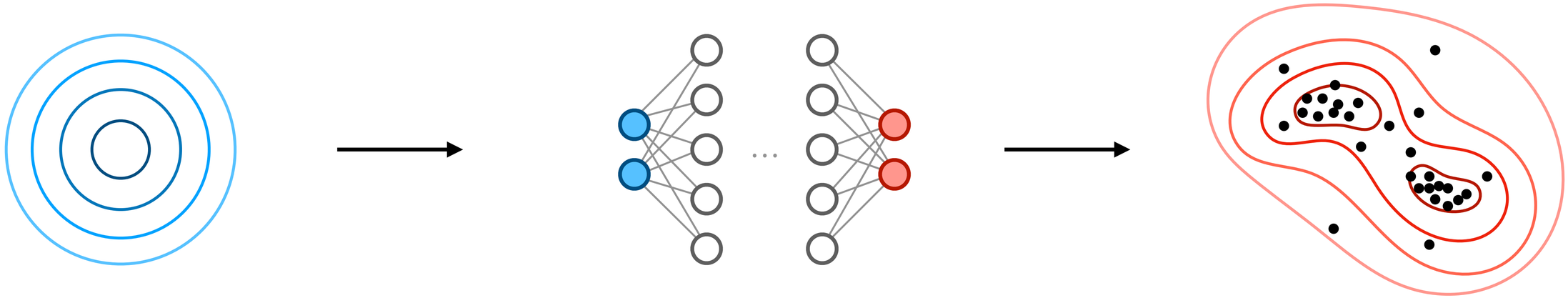
- one-to-one and invertible
- differentiable
- $f^{-1}$ and $\det \nabla f$ are tractable

Target density is given by

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

[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

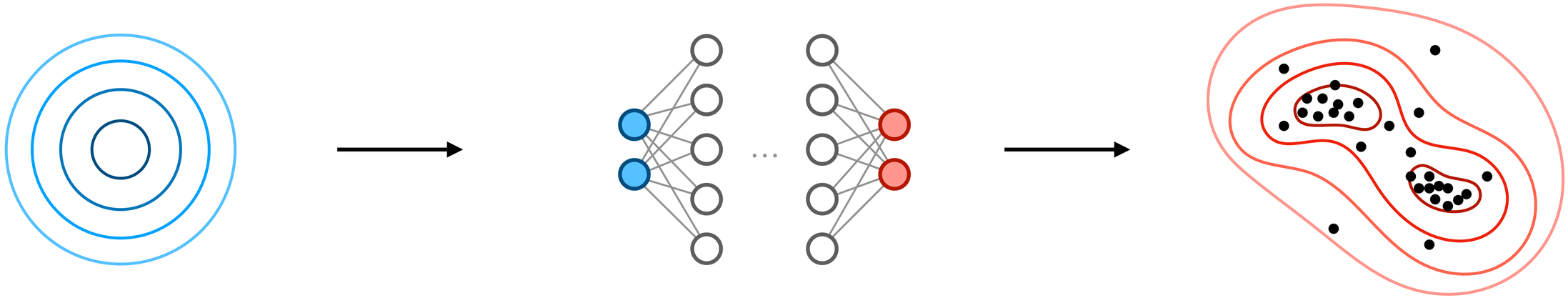
Target density is given by

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

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

[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

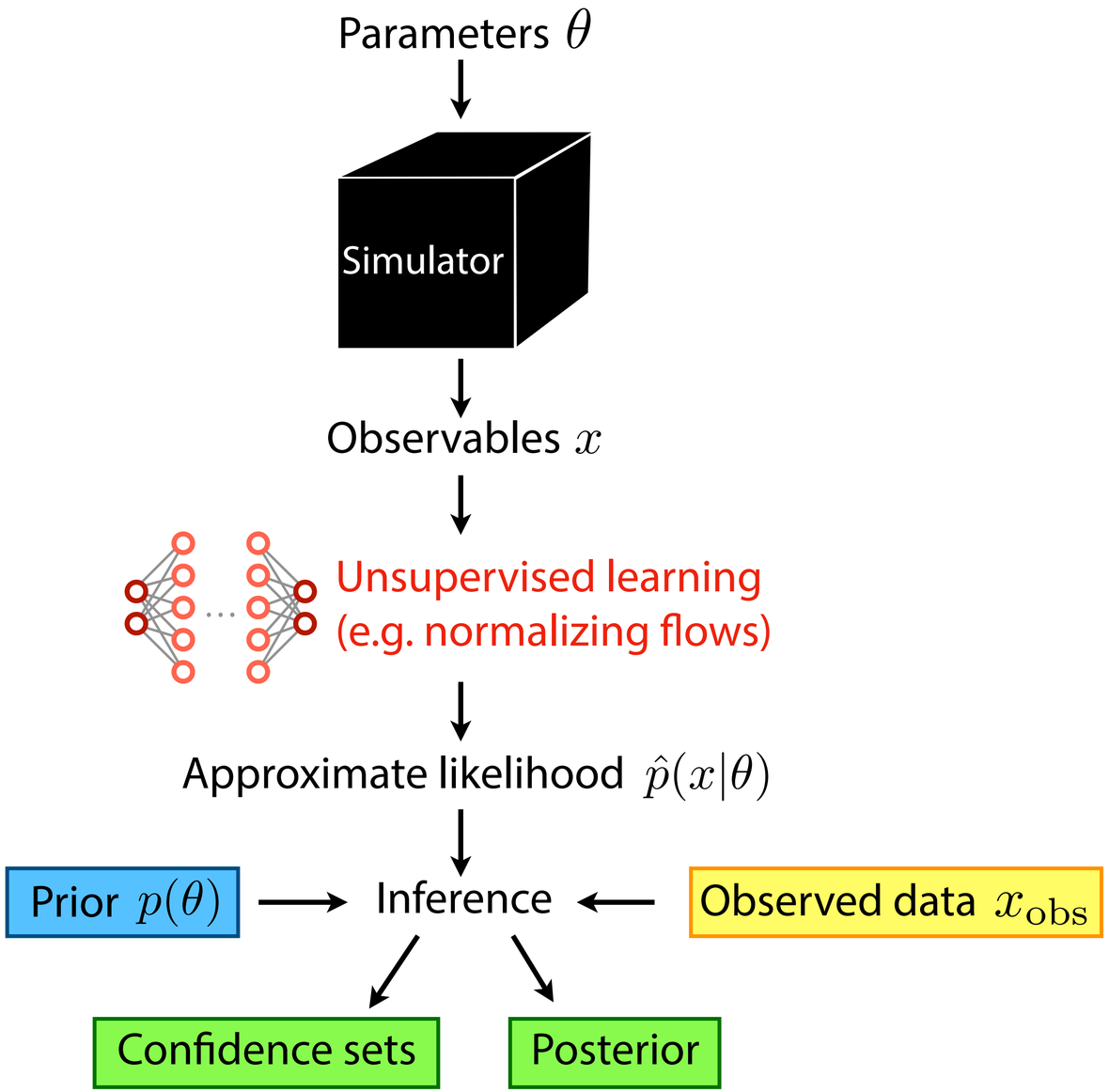
$\hat{p}(x) = \pi(f^{-1}(x)) \, |\det \nabla f|^{-1}$

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

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

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

# Inference with neural likelihood estimation




[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 $\hat{p}(\theta|x_{\mathrm{obs}})$

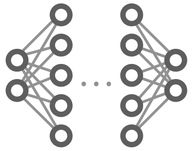
[G. Papamakarios et al 1605.06376;
J.-M. Lueckmann et al 1711.01861]

# The likelihood ratio trick

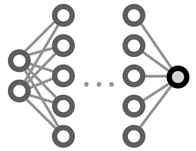
- Generative Adversarial Networks (GANs):

[I. Goodfellow et al. 1406.2661]

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



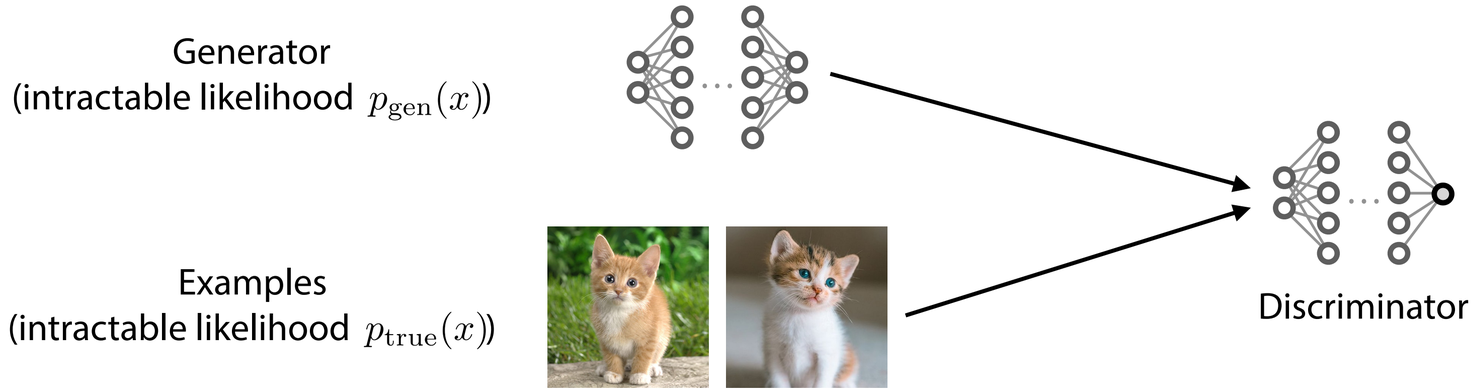
Examples
(intractable likelihood $p_{\text{true}}(x)$)



Discriminator

# The likelihood ratio trick

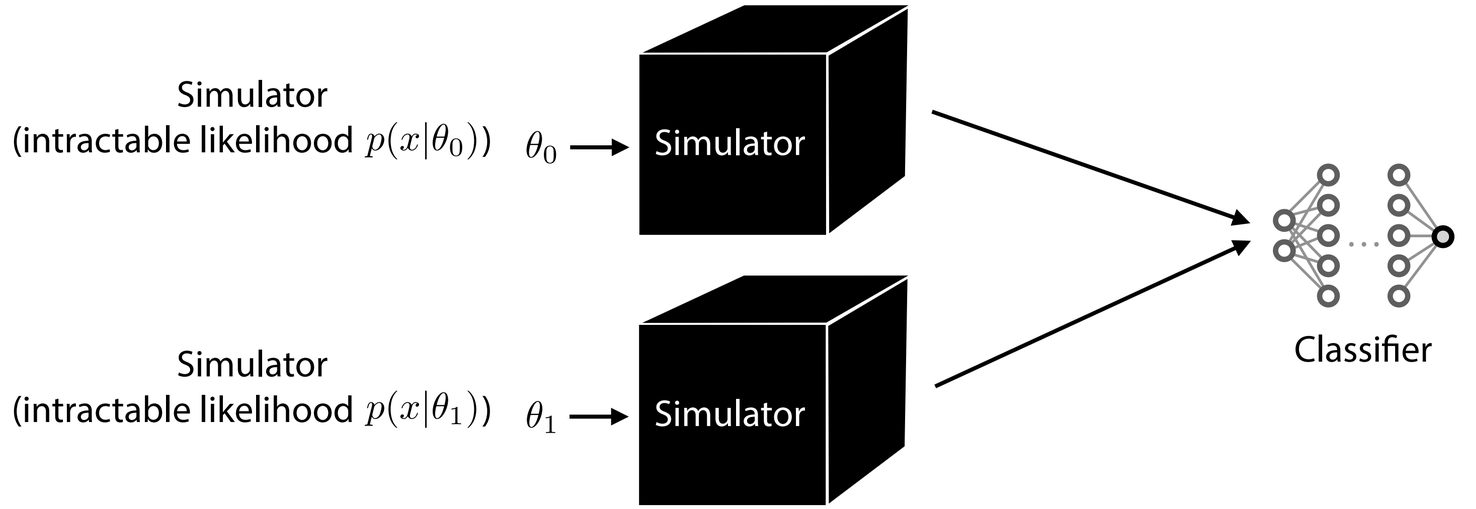
- Generative Adversarial Networks (GANs): [I. Goodfellow et al. 1406.2661]




Discriminator learns decision function

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

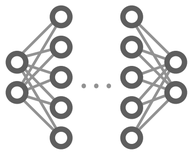
- Similarly, we can train a classifier between two sets of simulated samples [K. Cranmer, J. Pavez, G. Louppe 1506.02169]



# The likelihood ratio trick

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

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



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

Discriminator

Discriminator learns decision function

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

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

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

$\theta_0$ → Simulator

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

$\theta_1$ → Simulator

Classifier

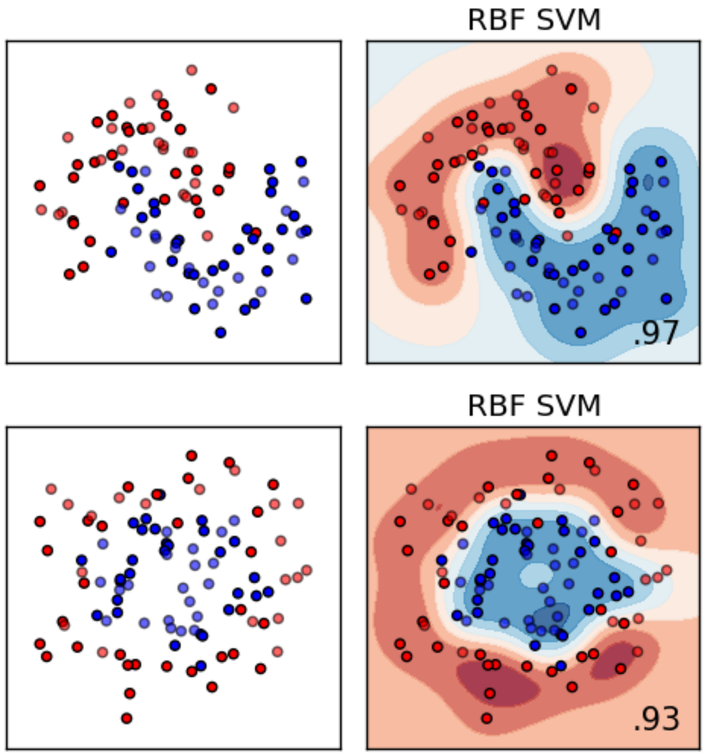
Classifier learns decision function

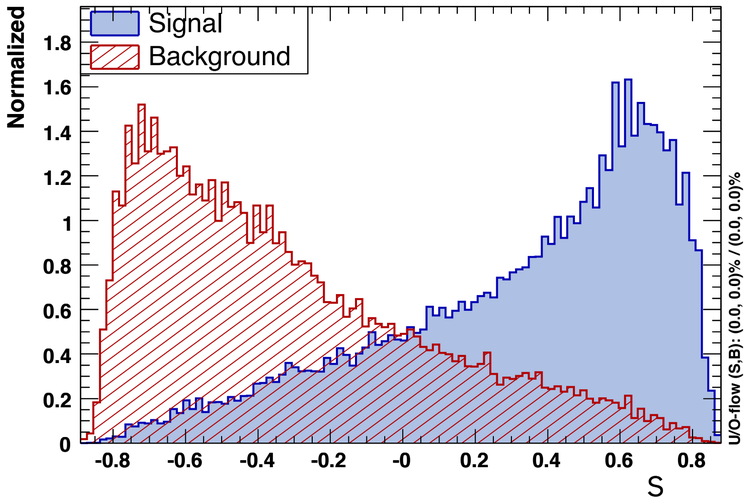
$$s(x) \to \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)}$$

⇒ Estimator for likelihood ratio

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

# LIKELIHOOD RATIO TRICK




- **binary classifier**: find function s(x) that minimizes ***loss***:

$$L[s] = \int p(x|H_0)\,(0 - s(x))^2\,dx + \int p(x|H_1)\,(1 - s(x))^2 dx$$

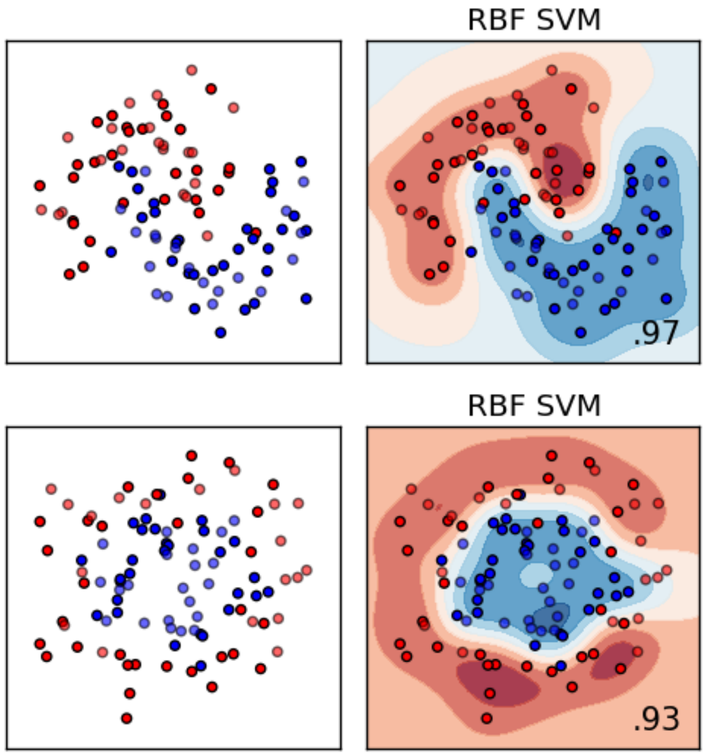
- i.e. approximate the optimal classifier

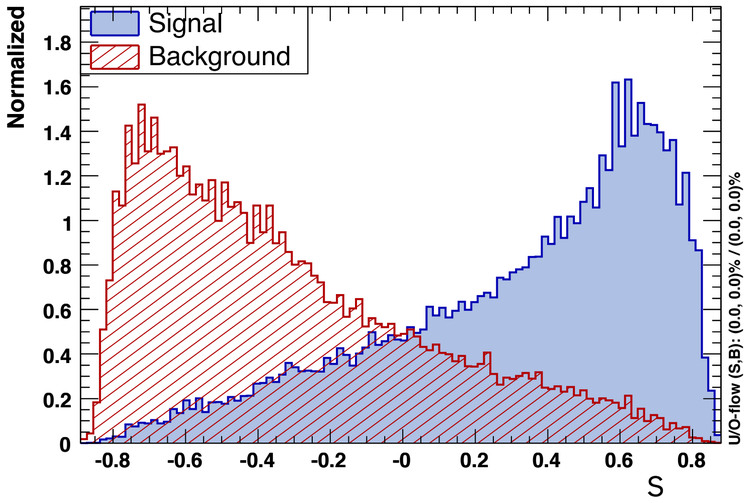
$$s(x) = \frac{p(x|H_1)}{p(x|H_0) + p(x|H_1)}$$

- which is 1-to-1 with the likelihood ratio

$$\frac{p(x|H_1)}{p(x|H_0)}$$

# LIKELIHOOD RATIO TRICK




- **binary classifier**: find function s(x) that minimizes ***loss***:

$$L[s] = \int p(x|H_0)\,(0 - s(x))^2\,dx + \int p(x|H_1)\,(1 - s(x))^2 dx$$

$$\approx \frac{1}{N}\sum_{i=1}^{N}(y_i - s(x_i))^2$$

- i.e. approximate the optimal classifier

$$s(x) = \frac{p(x|H_1)}{p(x|H_0) + p(x|H_1)}$$

- which is 1-to-1 with the likelihood ratio

$$\frac{p(x|H_1)}{p(x|H_0)}$$

# EXTENDING THE LIKELIHOOD RATIO TRICK

A binary classifier approximates

$$s(x) = \frac{p(x|H_1)}{p(x|H_0) + p(x|H_1)}$$

Which is one-to-one with the likelihood ratio

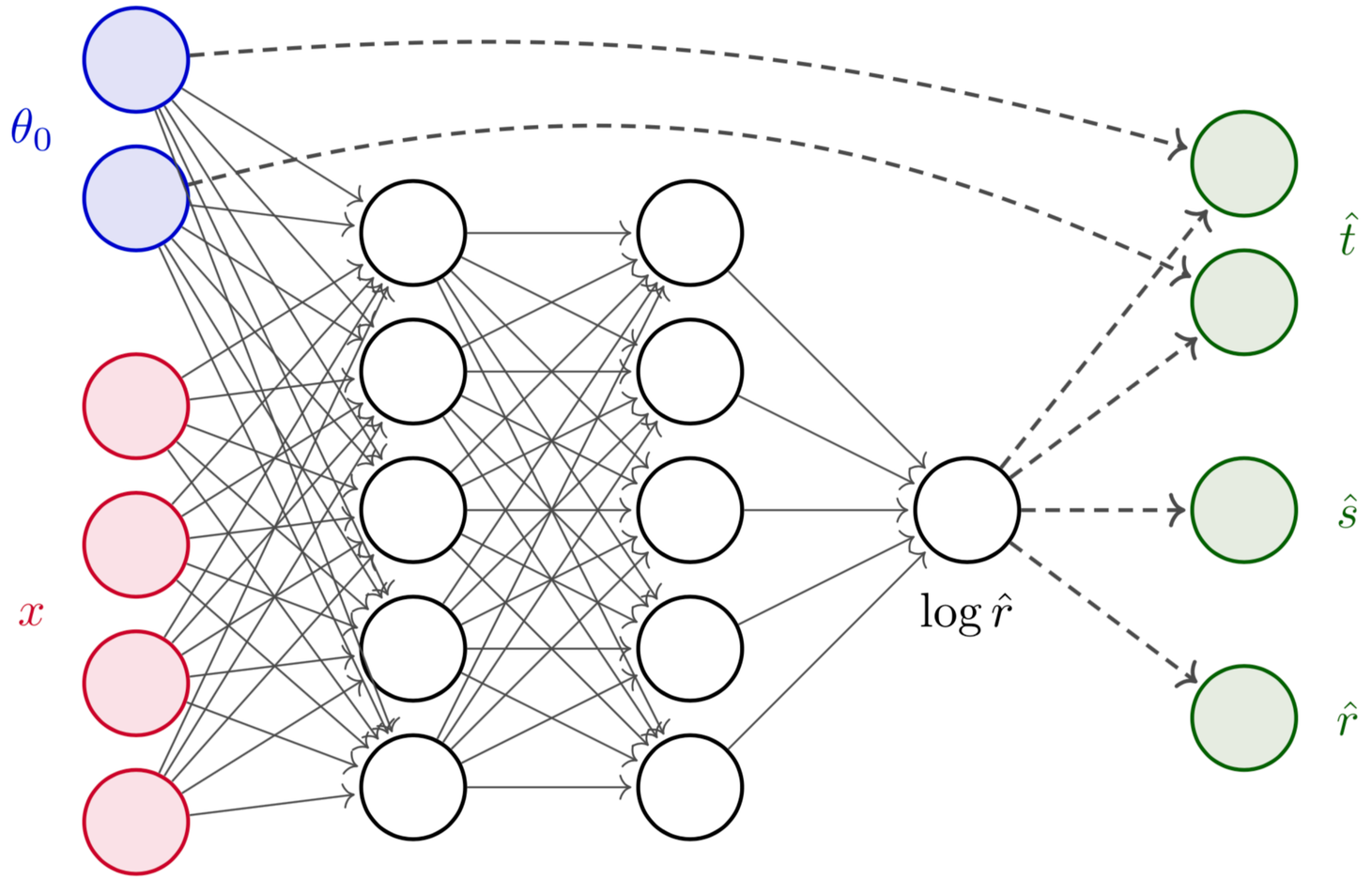
$$\frac{p(x|H_1)}{p(x|H_0)} = 1 - \frac{1}{s(x)}$$

Can do the same thing for any two points $\theta_0$ & $\theta_1$ in parameter space $\Theta$. I call this a **parametrized classifier**

$$s(x; \theta_0, \theta_1) = \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)}$$

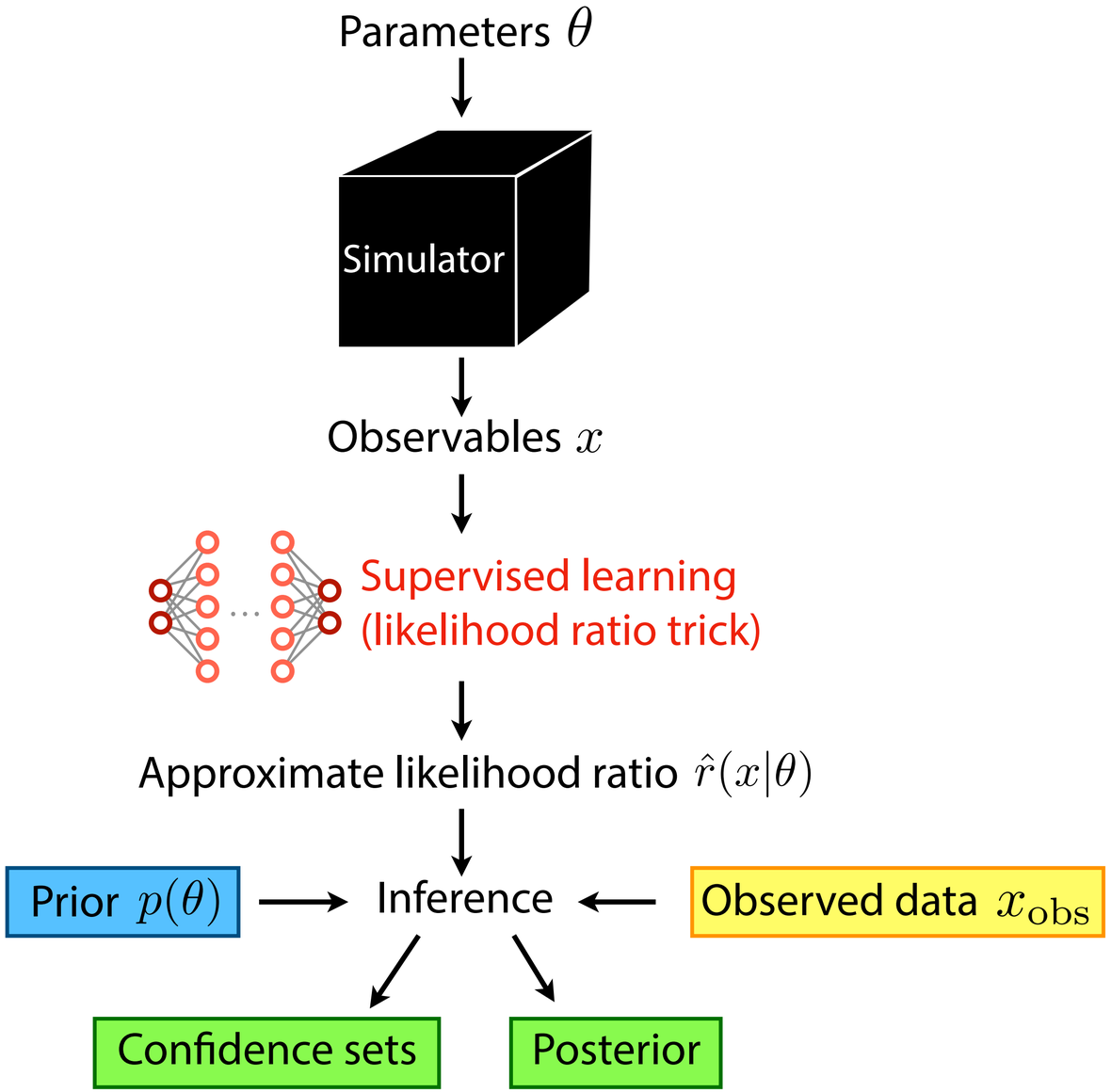
K.C., G. Louppe, J. Pavez: Approximating Likelihood Ratios with Calibrated Discriminative Classifiers [arXiv:1506.02169]

# PARAMETERIZED CLASSIFIERS



# Inference by likelihood ratio trick

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



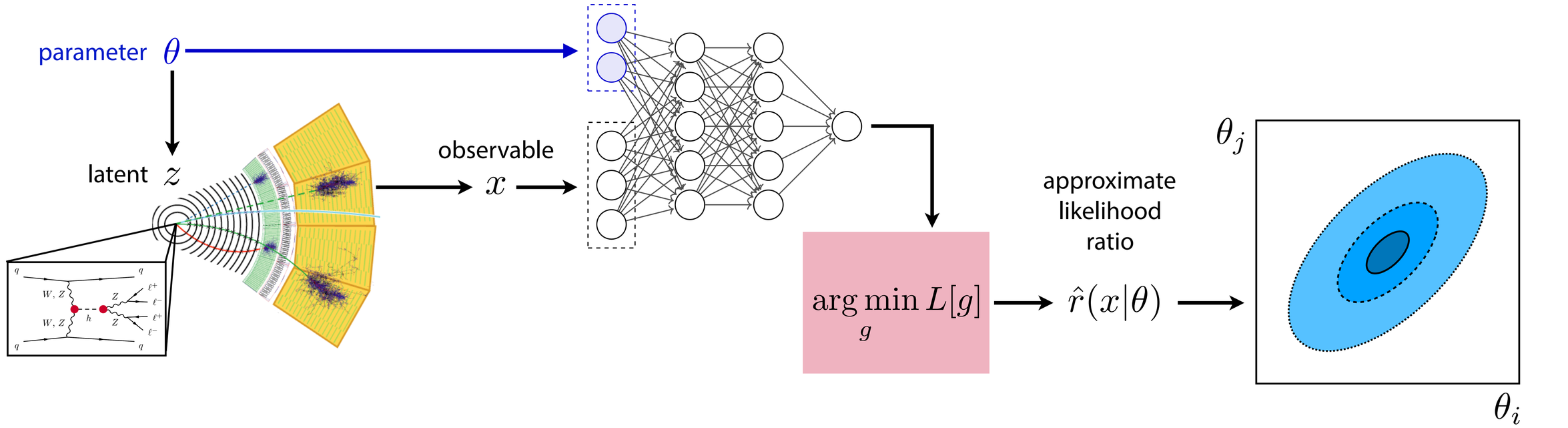

- For inference, likelihood and likelihood ratio are interchangeable
- Advantage: Learning the likelihood ratio can be a simpler task than learning the likelihood
- Disadvantage: Cannot sample from likelihood ratio

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

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

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

# Learning with Simulated Data




## Simulation

"Mining gold": Extract additional information from simulator

## Machine Learning

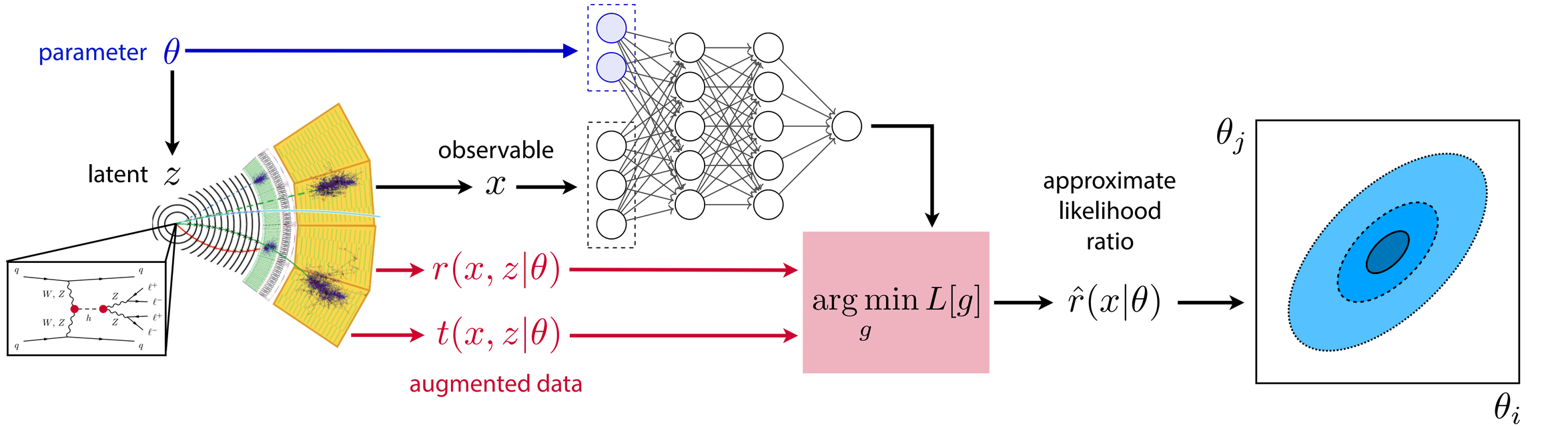
Use this information to train estimator for likelihood ratio

## Inference

Limit setting with standard hypothesis tests

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

# Learning with Augmented Data




## Simulation

"Mining gold": Extract additional information from simulator

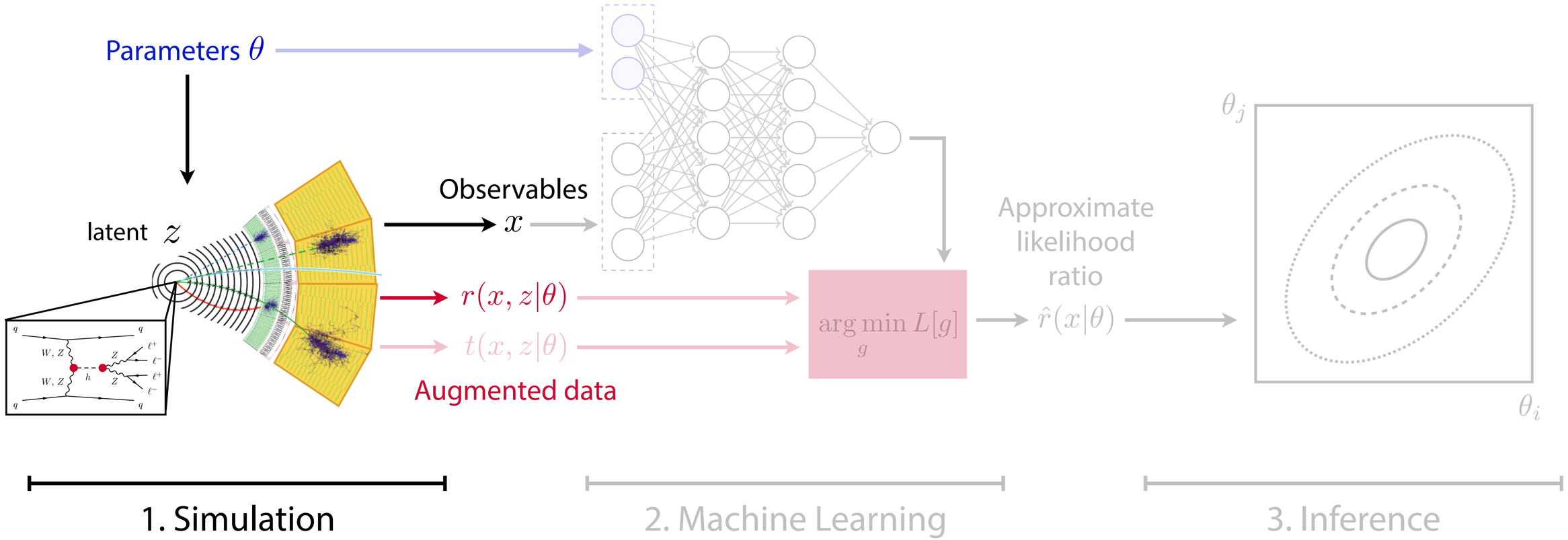
## Machine Learning

Use this information to train estimator for likelihood ratio

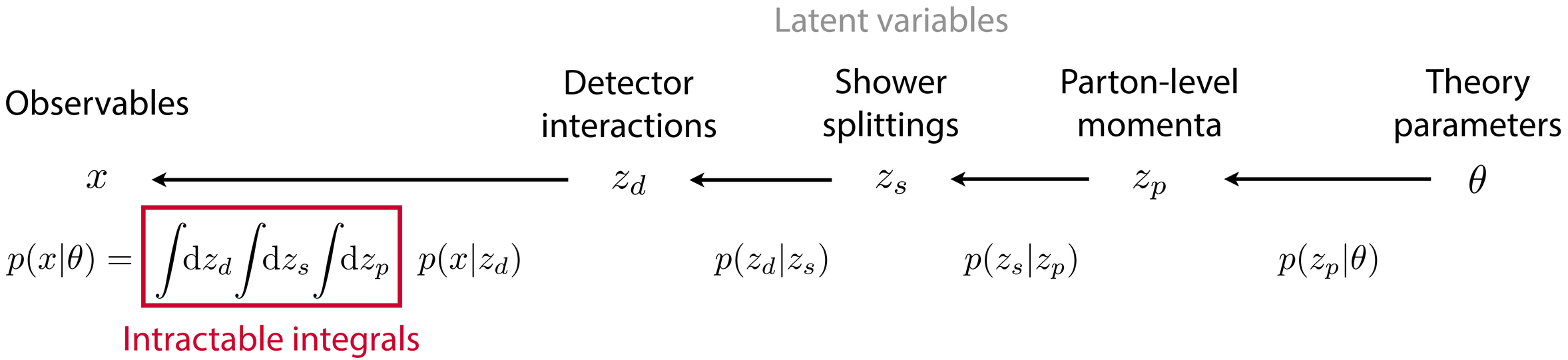
## Inference

Limit setting with standard hypothesis tests

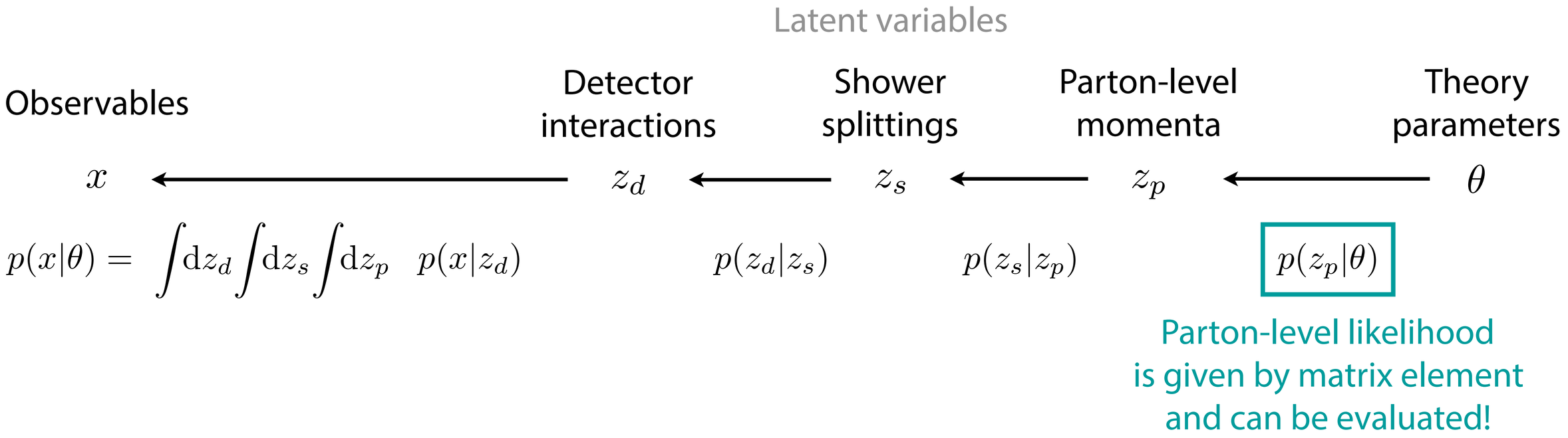
# Learning with Augmented Data



# Mining gold from the simulator



# Mining gold from the simulator




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

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

# 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

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

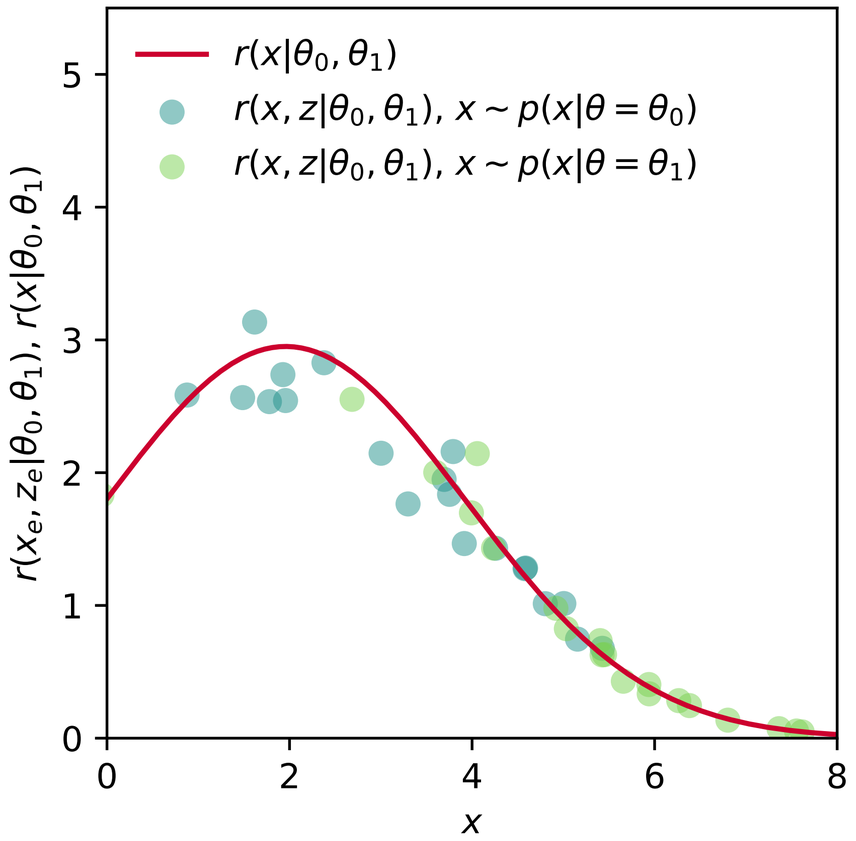
("How much more likely is the observation $x$ for $\theta_0$ compared to $\theta_1$?")

# 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

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

(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

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

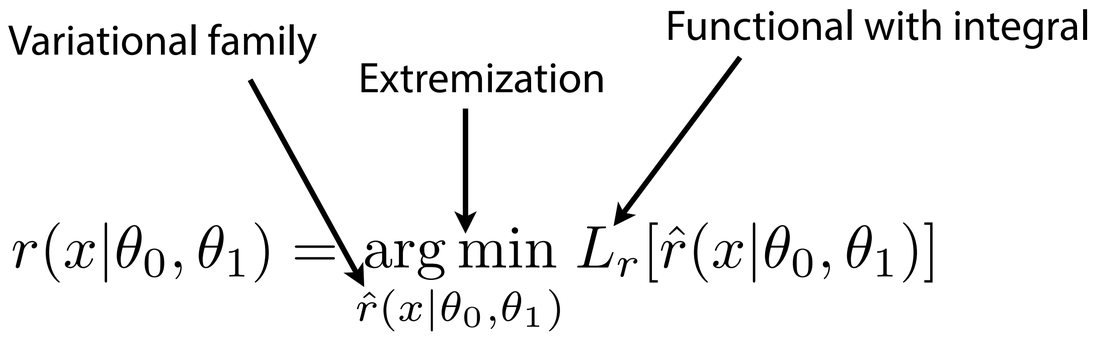
(And we can sample from $p(x, z|\theta)$ by running the simulator.)

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

$$\begin{aligned}\mathbb{E}_{z\sim p(z|x,\theta_1)}\left[r(x, z|\theta_0, \theta_1)\right] &= \int \mathrm{d}z\, p(z|x, \theta_1)\, \frac{p(x, z|\theta_0)}{p(x, z|\theta_1)} \\ &= \int \mathrm{d}z\, \frac{p(x, z|\theta_1)}{p(x|\theta_1)}\, \frac{p(x, z|\theta_0)}{p(x, z|\theta_1)} \\ &= r(x|\theta_0, \theta_1)\ \textbf{!}\end{aligned}$$

# Machine learning = applied calculus of variatios

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



# 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

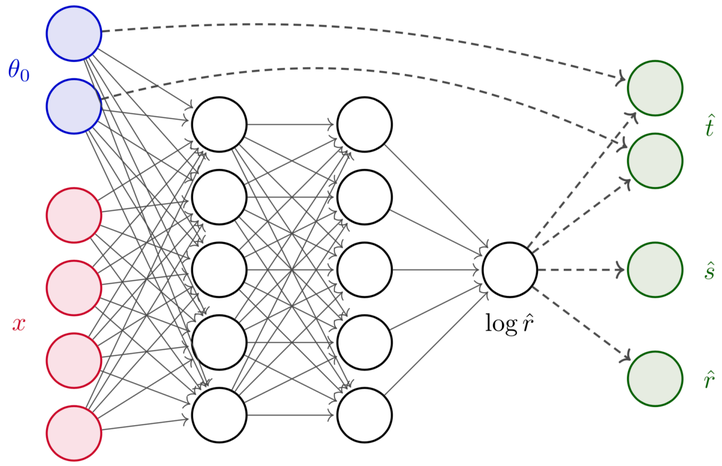
$$r(x|\theta_0,\theta_1) = \underset{\hat{r}(x|\theta_0,\theta_1)}{\arg\min}\ L_r[\hat{r}(x|\theta_0,\theta_1)]$$

This is where machine learning comes in!

Neural network

Stochastic gradient descent

Loss function with
finite sum over samples



# 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

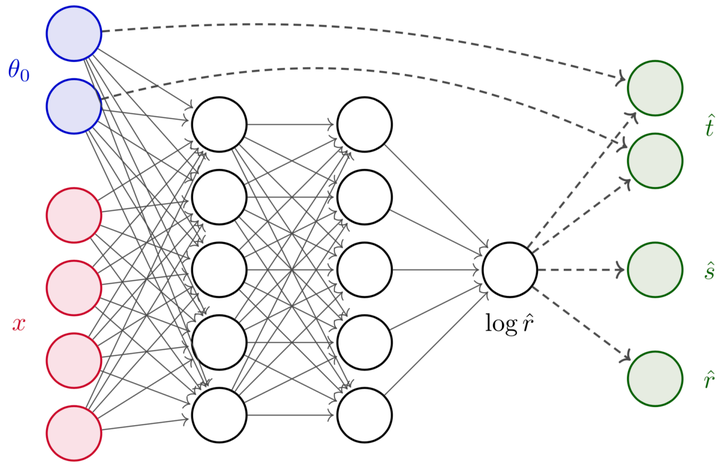
$$r(x|\theta_0,\theta_1) = \underset{\hat{r}(x|\theta_0,\theta_1)}{\arg\min}\ L_r[\hat{r}(x|\theta_0,\theta_1)]$$

This is where machine learning comes in!

Neural network

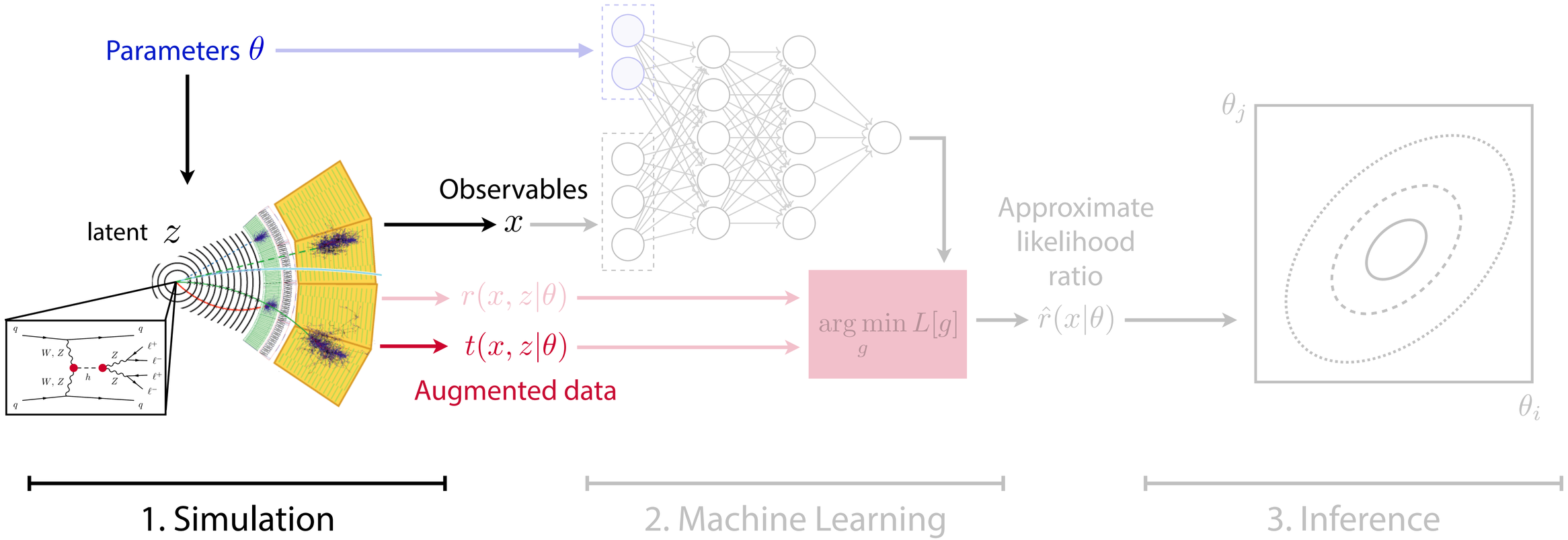
Loss function with
finite sum over samples

Stochastic gradient descent



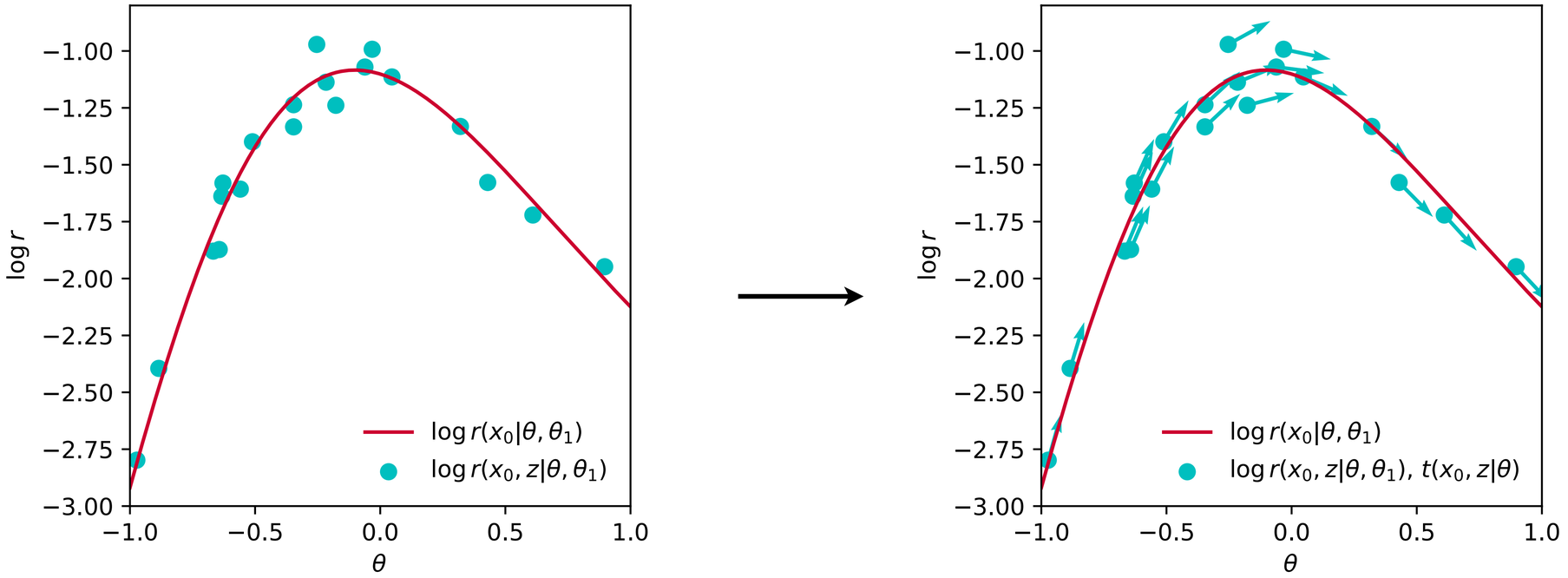

A sufficiently expressive neural network
efficiently trained in this way
with enough data will learn
the likelihood ratio function $r(x|\theta_0,\theta_1)$!

# Learning with Augmented Data



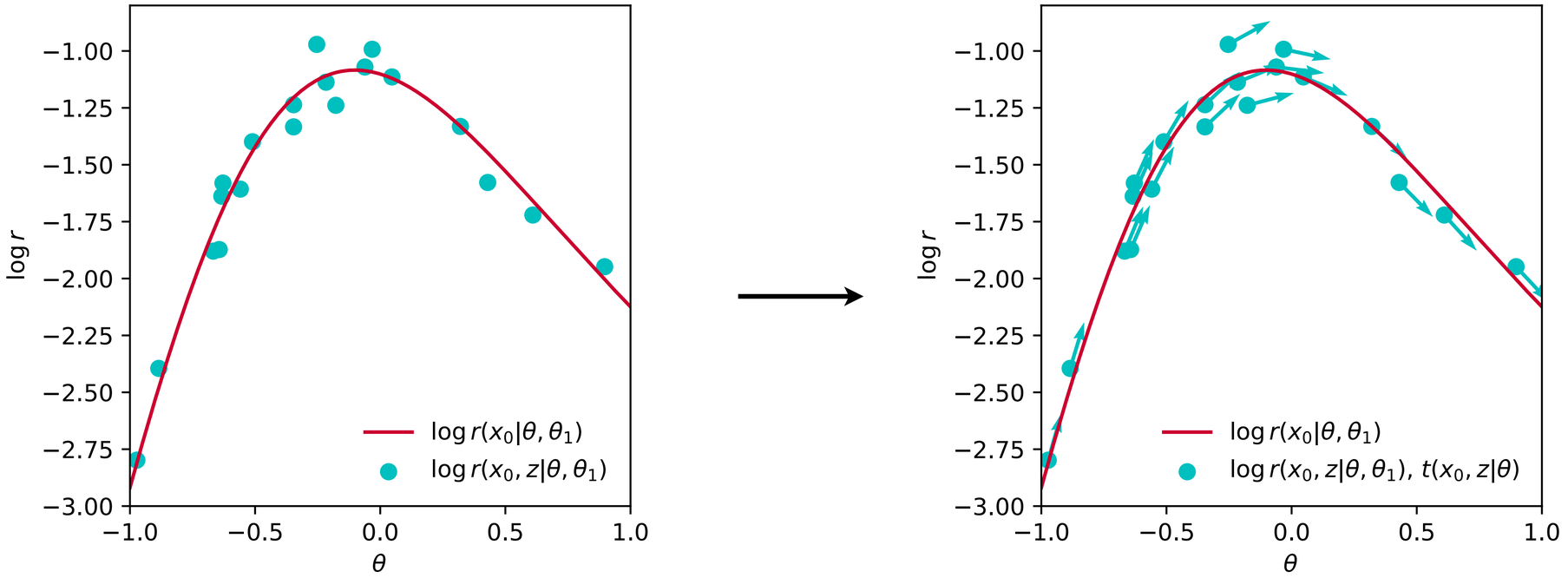
# One more piece: the score

- 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

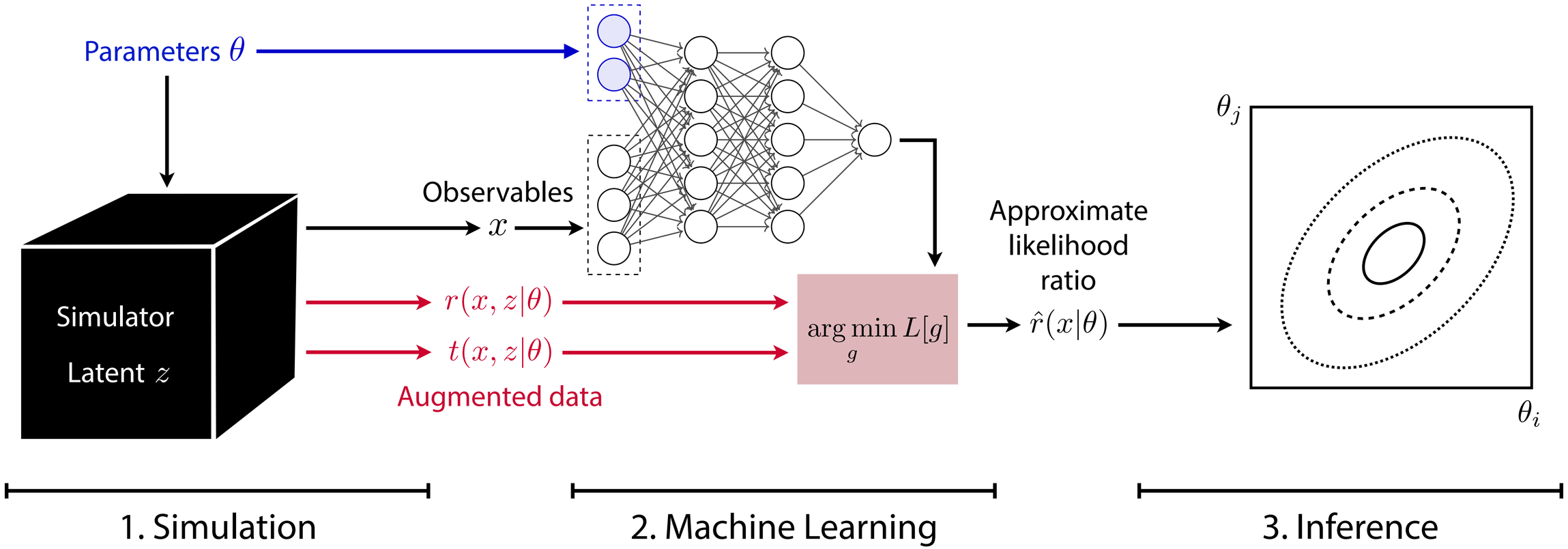
$$L_t[\hat{t}(x|\theta_0)] = \int \mathrm{d}x \int \mathrm{d}z \;\; p(x, z|\theta_0) \;\; \Big[\big(\hat{t}(x|\theta_0) - t(x, z|\theta_0)\big)^2\Big].$$

One can show it is minimized by

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

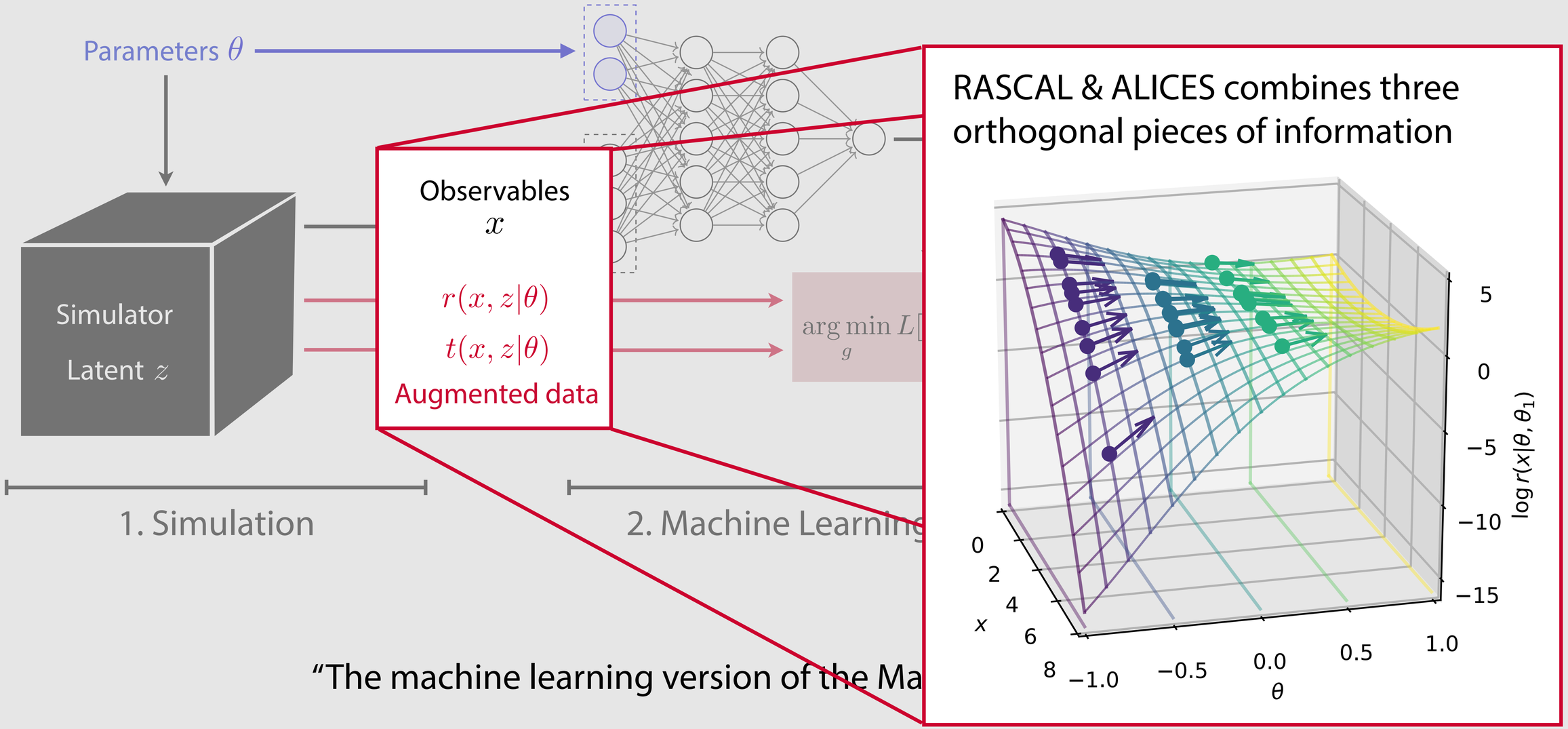
Again, we implement this minimization
through machine learning.

# Putting the pieces together: RASCAL & ALICES



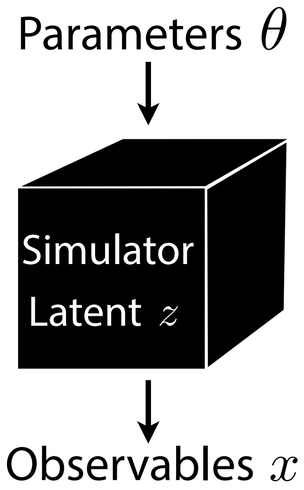

"The machine learning version of the Matrix Element Method"

# Putting the pieces together: RASCAL & ALICES

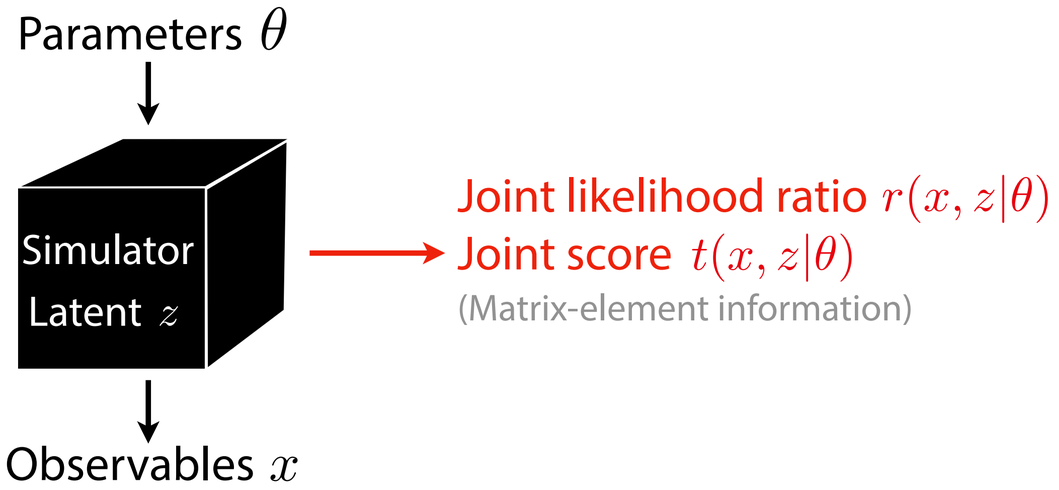



"The machine learning version of the Ma

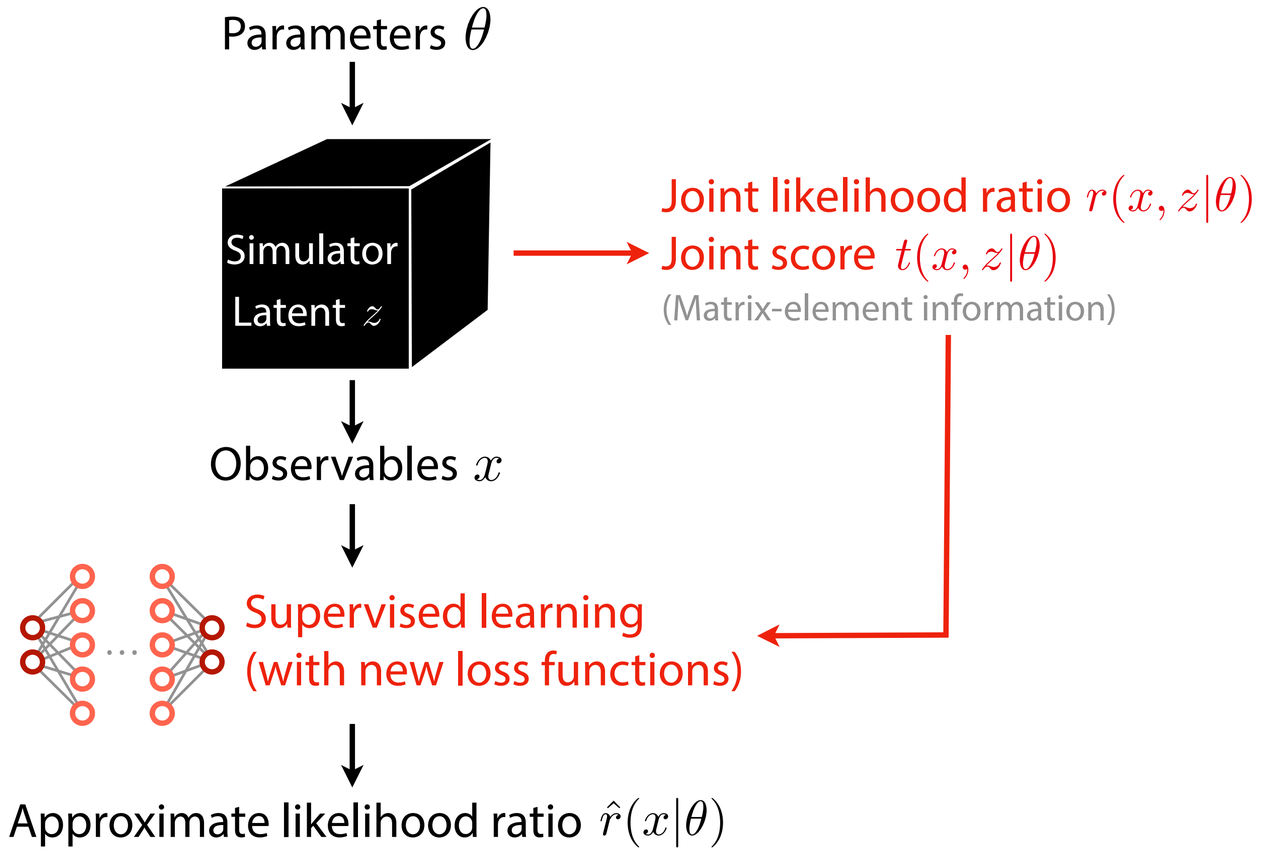
# Mining gold: Recap



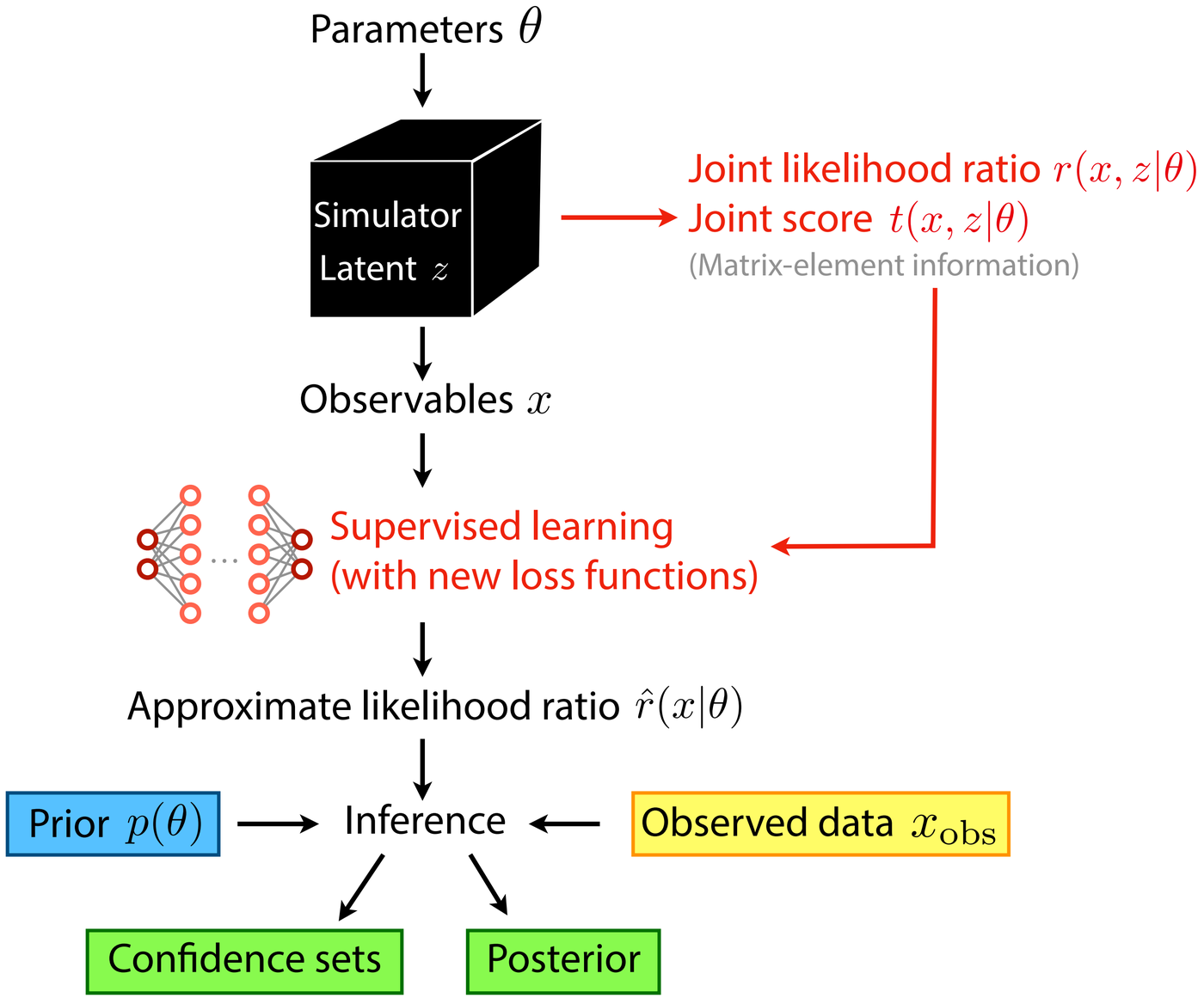
# Mining gold: Recap



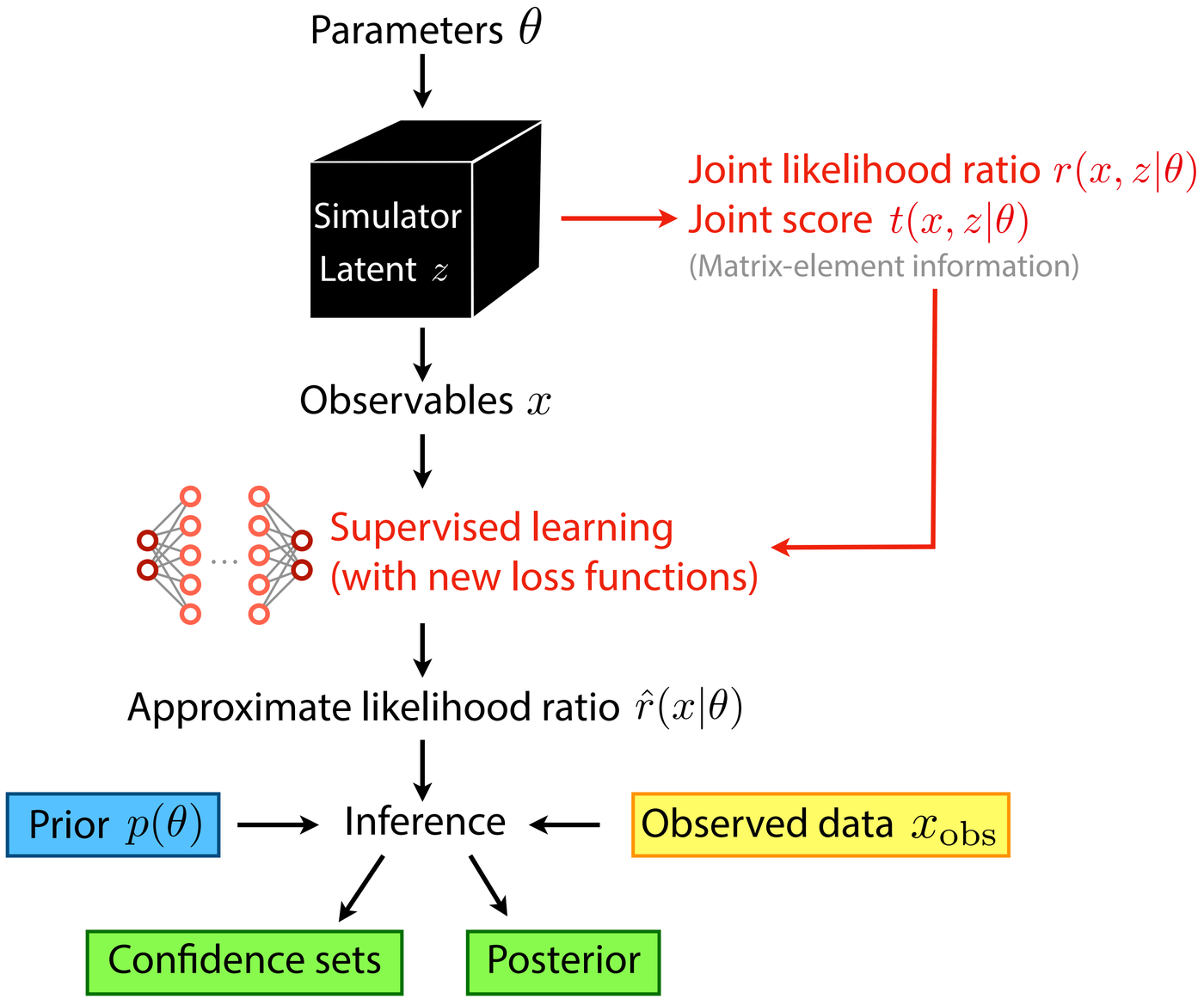
# Mining gold: Recap



# Mining gold: Recap



# 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)

# We can also machine learn optimal observables.

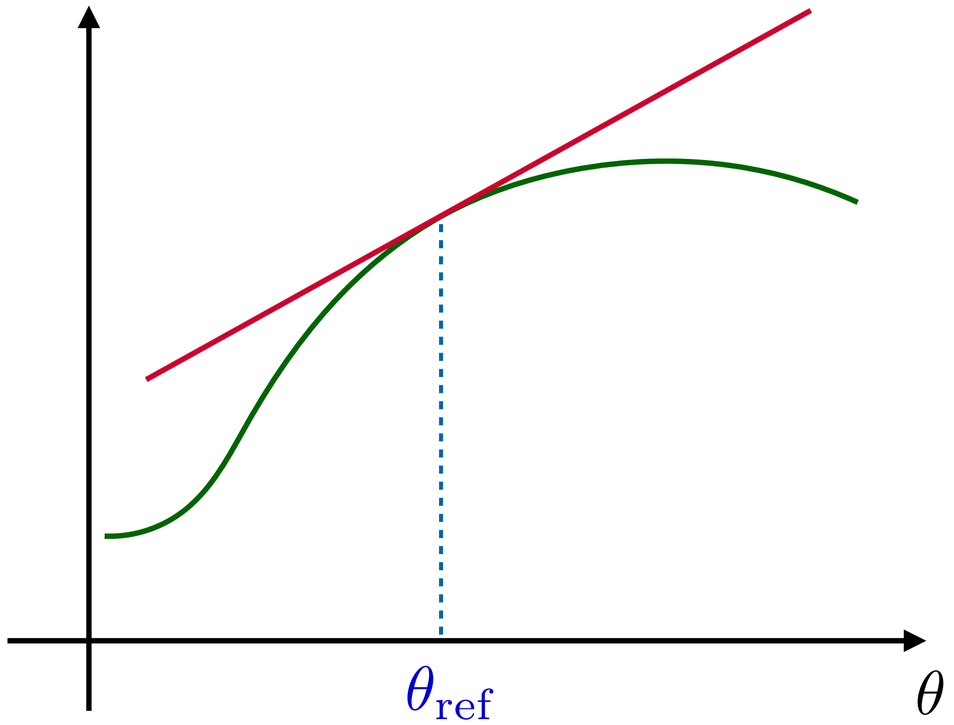
[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013, 1805.00020, 1805.12244]

# 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 $\theta_{\rm ref}$:

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

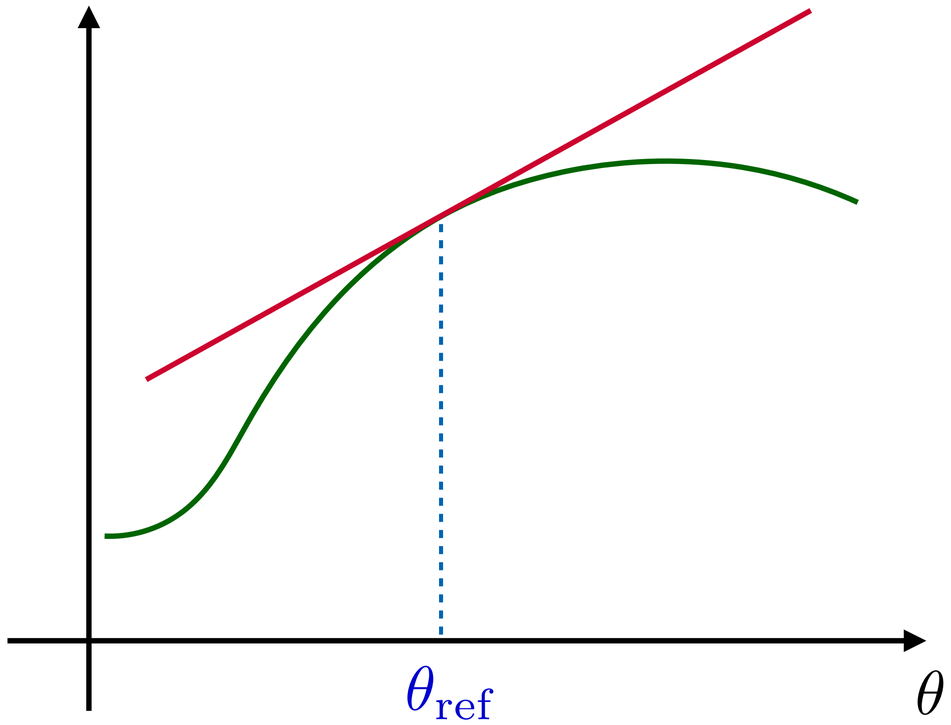


# The local model

[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 $\theta_{\rm ref}$:

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

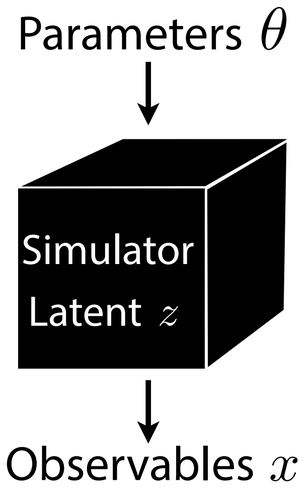



In the neighborhood of $\theta_{\rm ref}$ (e.g. close to the SM):

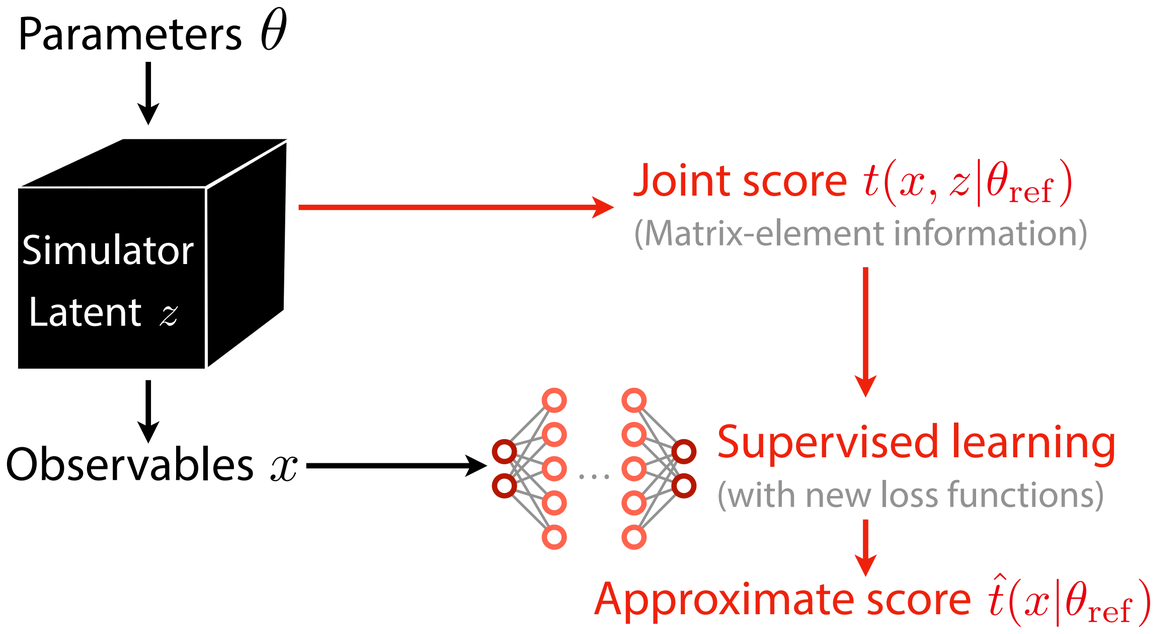
- the score vector $t(x|\theta_{\rm ref})$ components are sufficient statistics
- knowing $t(x|\theta_{\rm ref})$ is just as powerful as knowing the full function $\log p(x|\theta)$
- $t(x|\theta_{\rm ref})$ are the most powerful observables

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

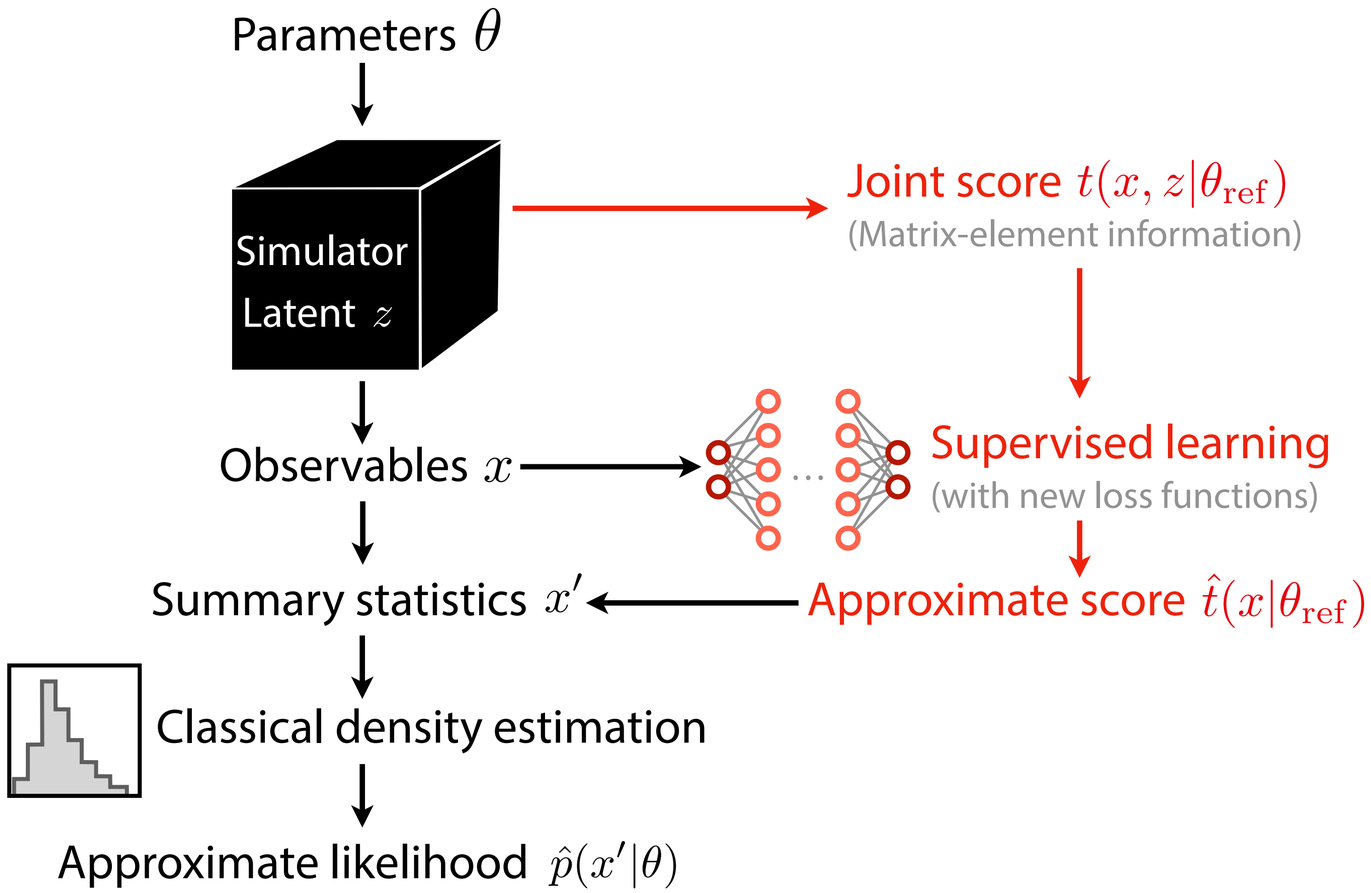
# Neural optimal observables (SALLY)



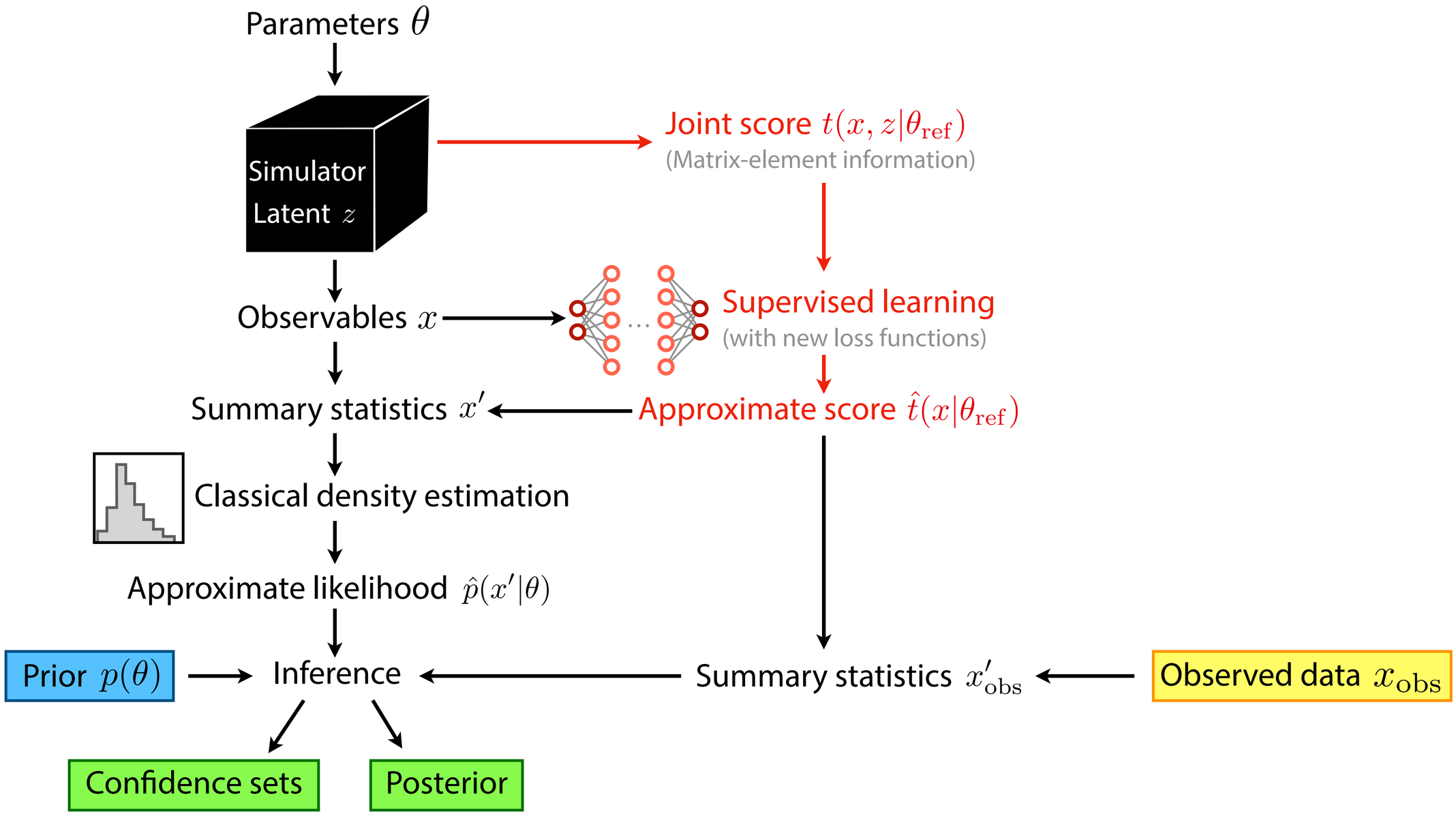
# Neural optimal observables (SALLY)



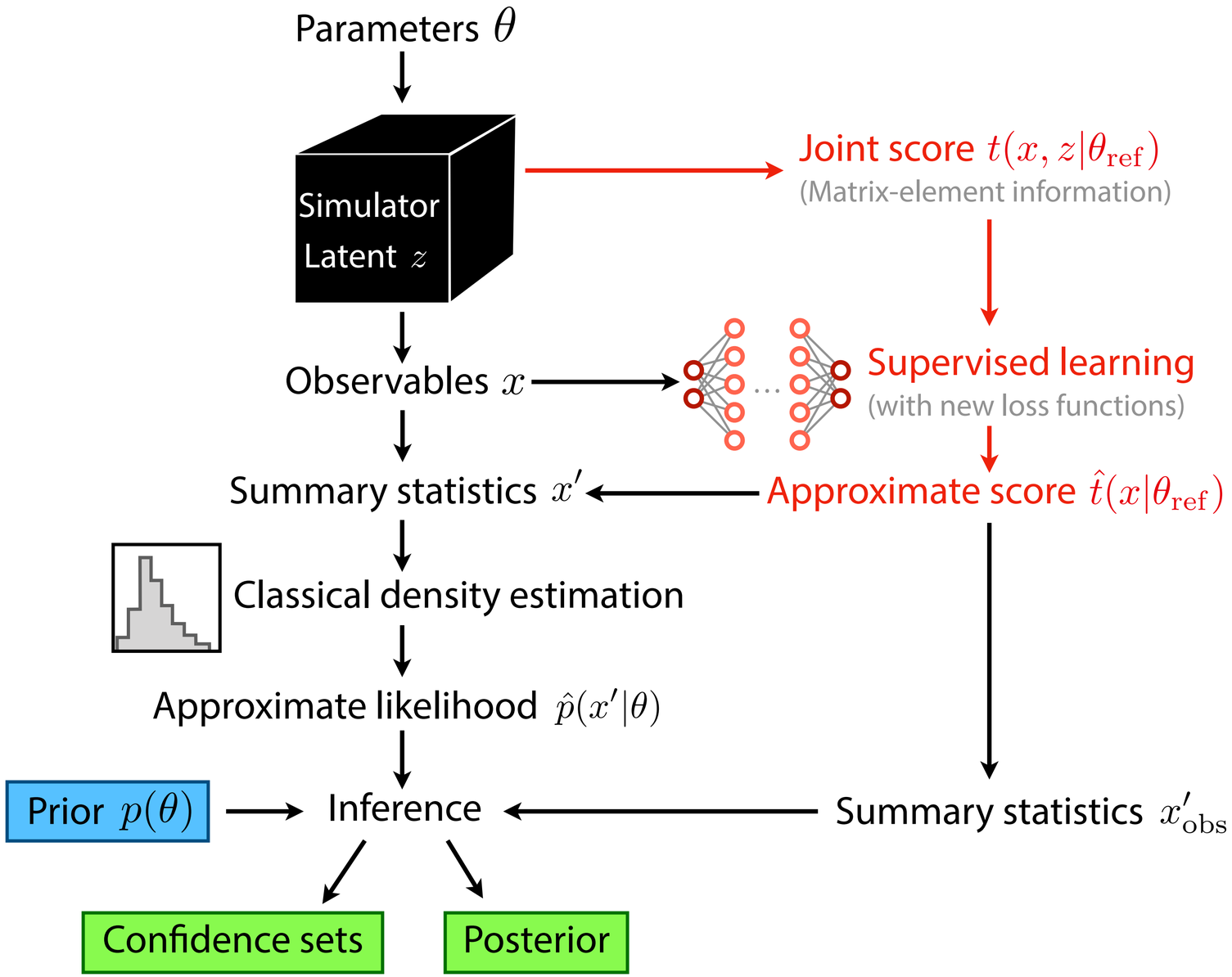
# Neural optimal observables (SALLY)



# Neural optimal observables (SALLY)



# Neural optimal observables (SALLY)



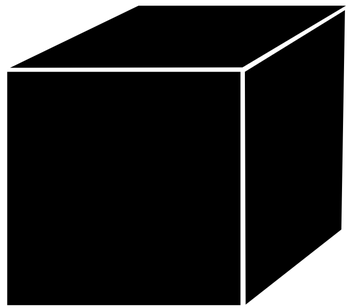

"The machine learning version of Optimal Observables"

- Simple & robust
- Just as powerful close to $\theta_{\rm ref}$, 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 |

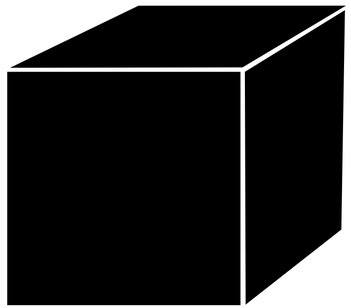
# Systematics

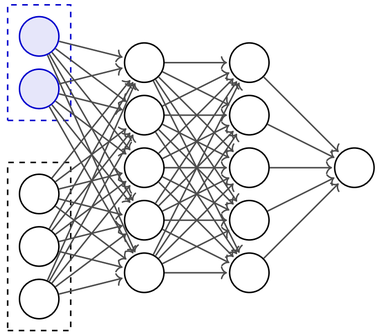


Don’t fully trust the simulator?

- 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]

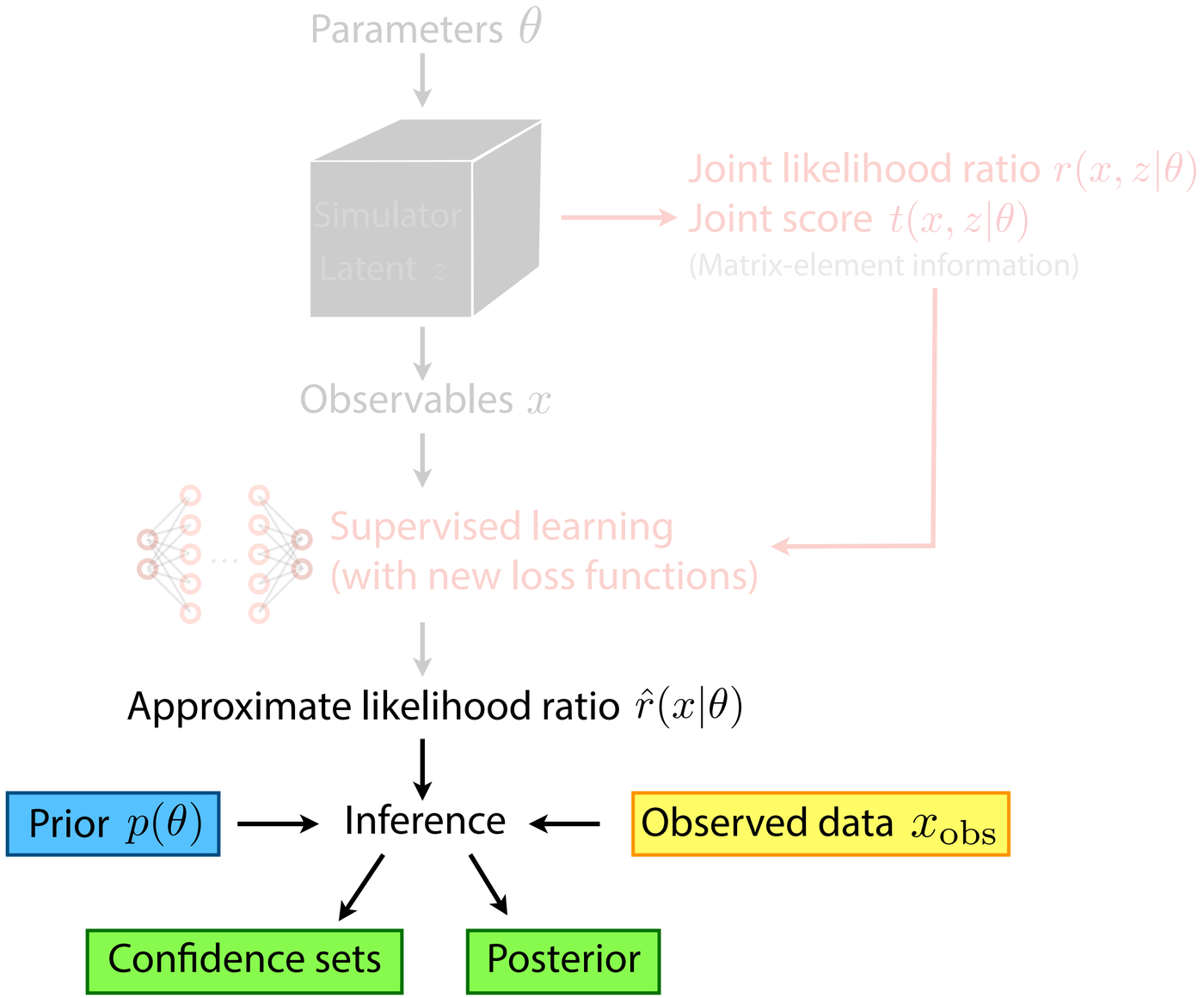
Don’t blindly trust the neural network?

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

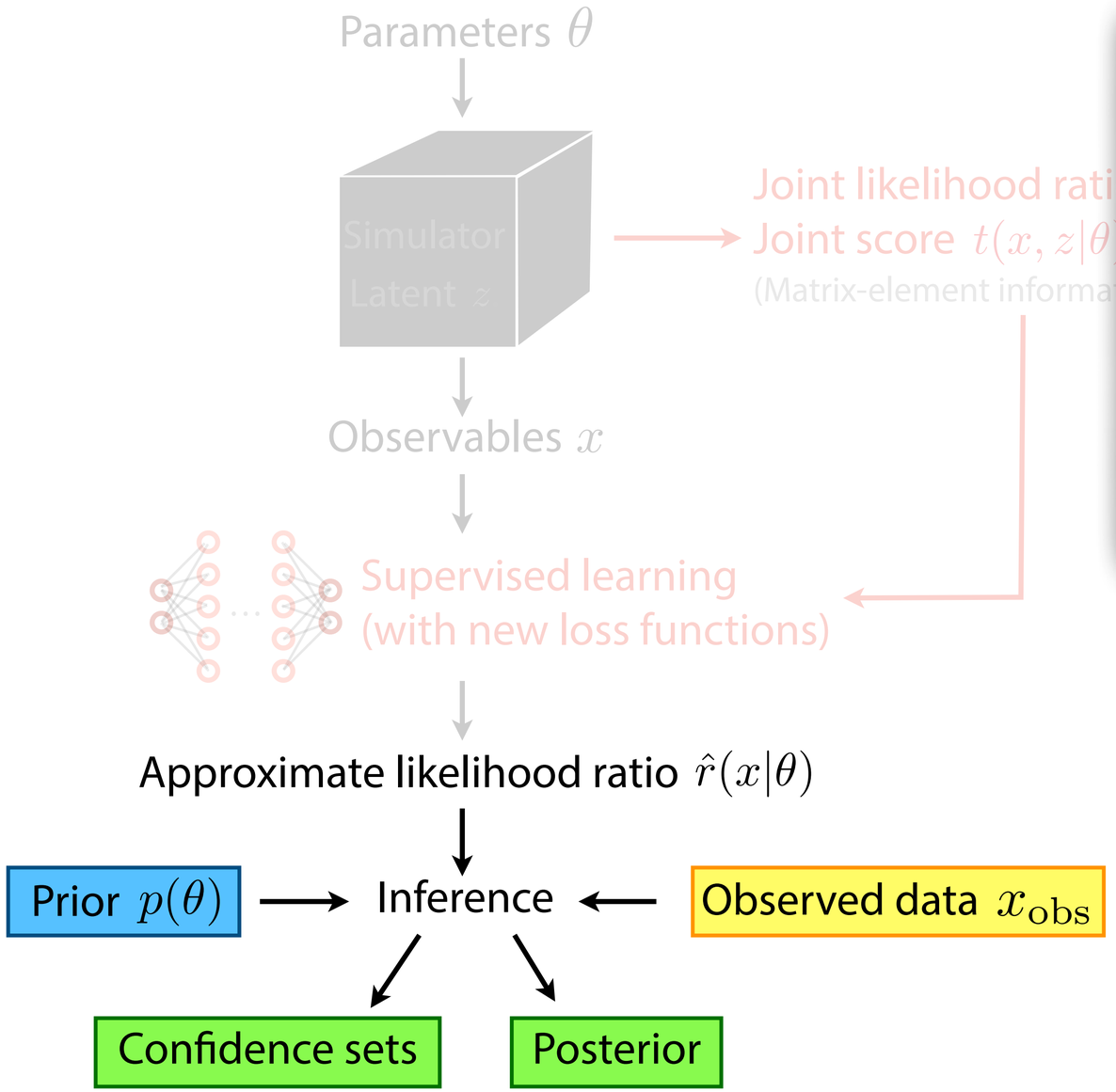
# Inference, Limits, & Fisher Information Geometry

[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.*

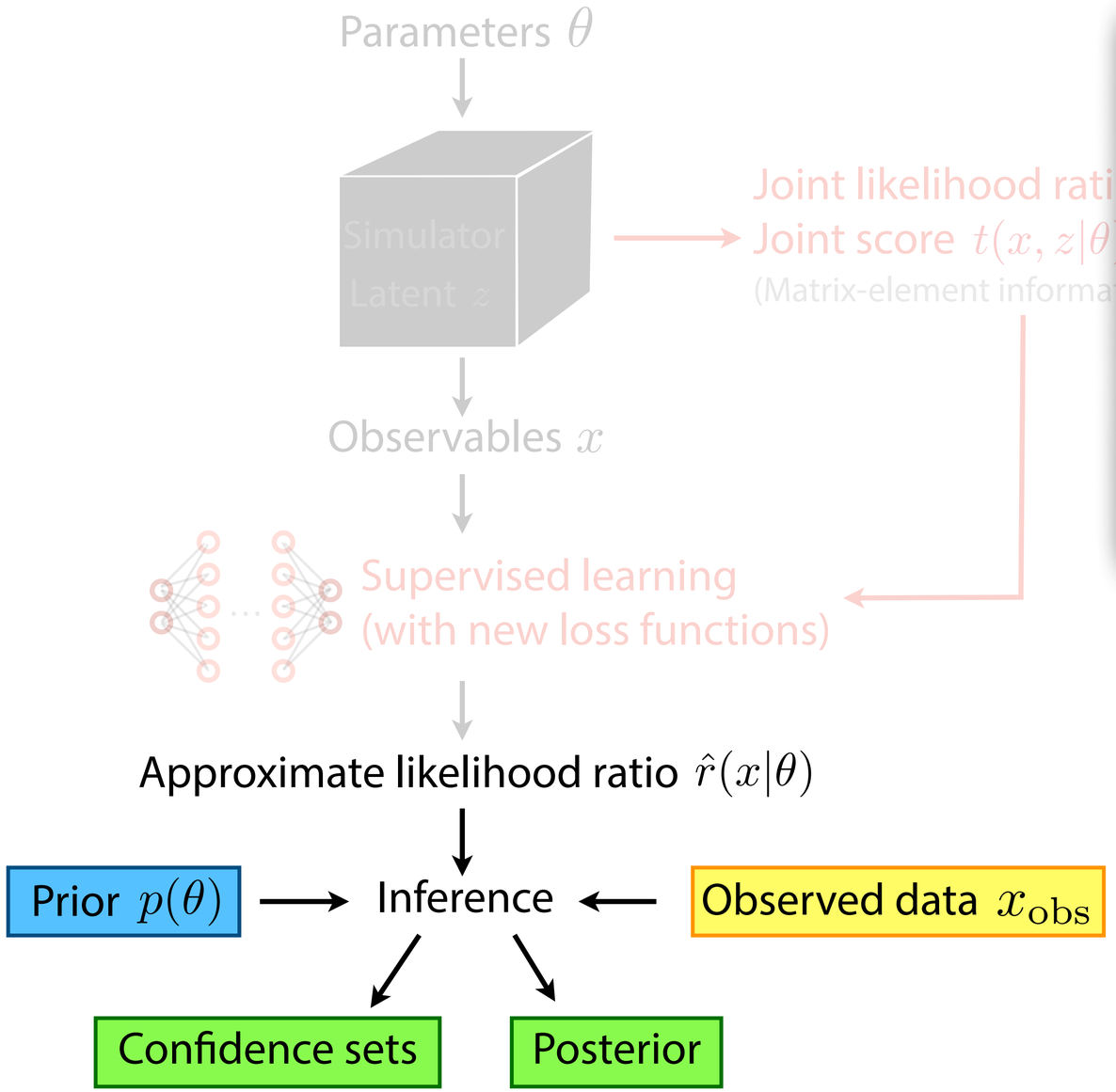
*By* J. NEYMAN, *Nencki Institute, Soc. Sci. Lit. Varsoviensis, and Lecturer at the Central College of Agriculture, Warsaw, and* E. S. PEARSON, *Department of Applied Statistics, University College, London.*

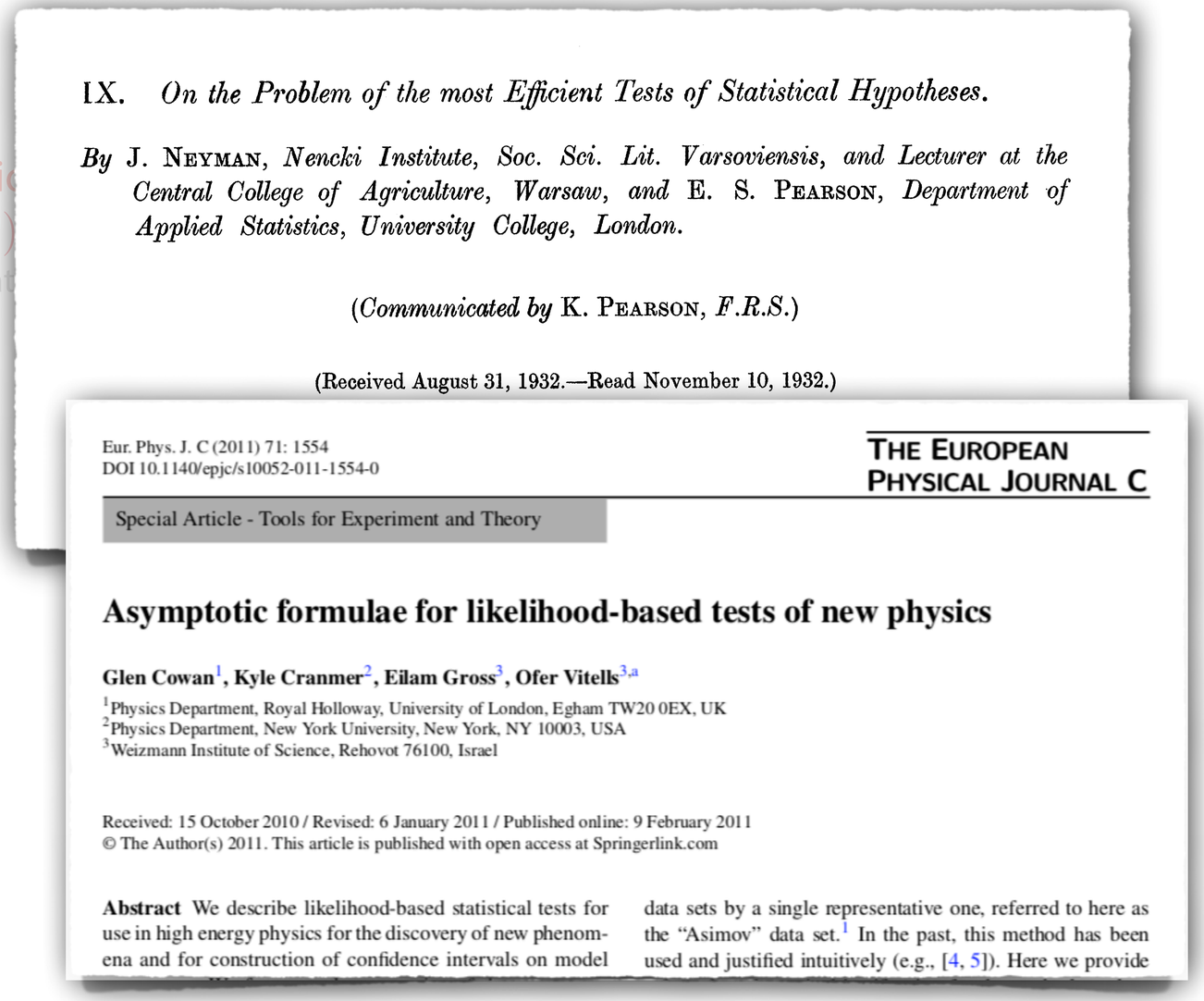
(*Communicated by* K. PEARSON, *F.R.S.*)

(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

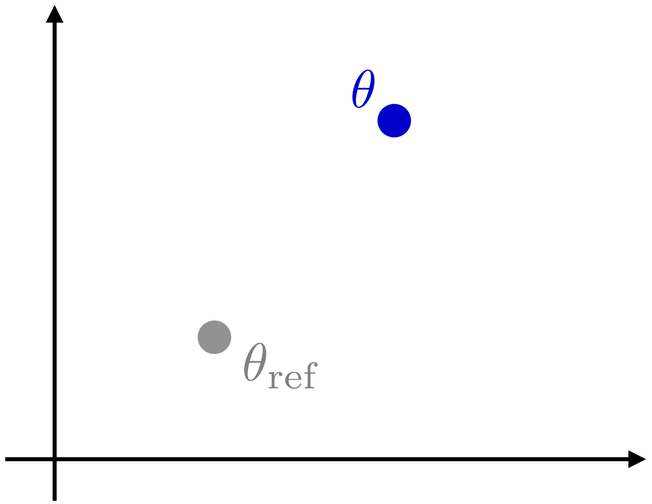
Received: 15 October 2010 / Revised: 6 January 2011 / Published online: 9 February 2011


**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)

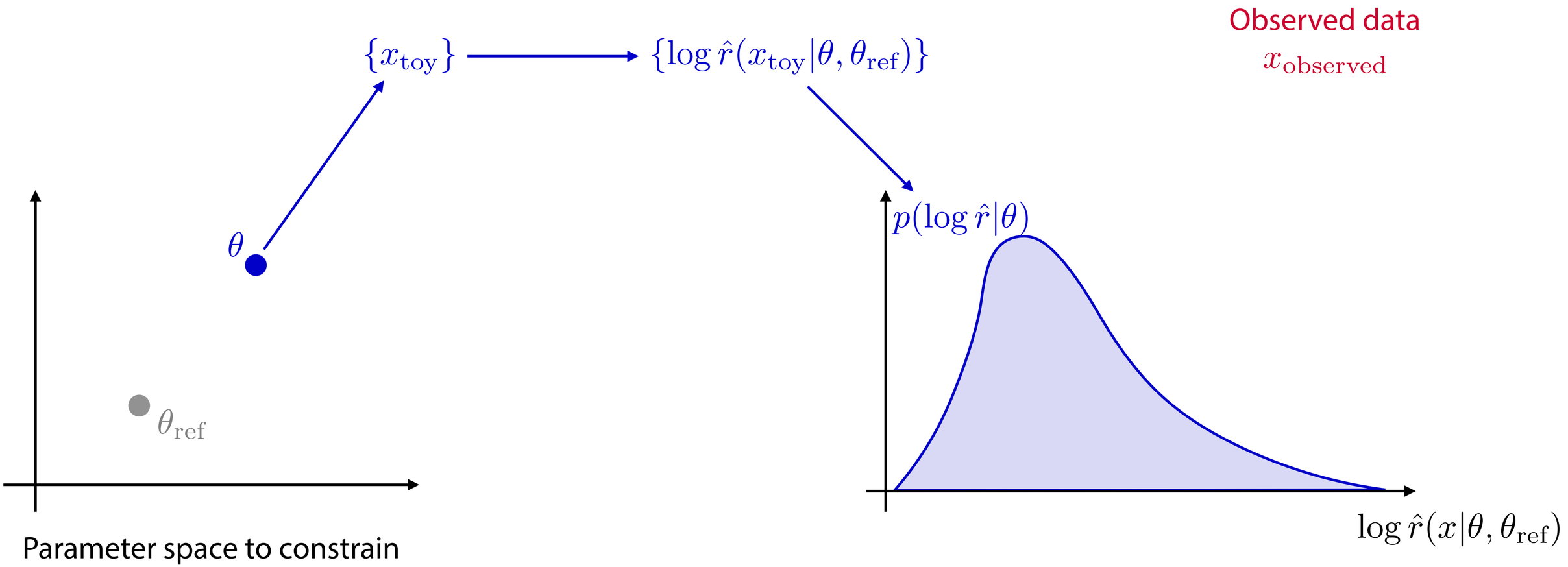
Observed data

$x_{\text{observed}}$

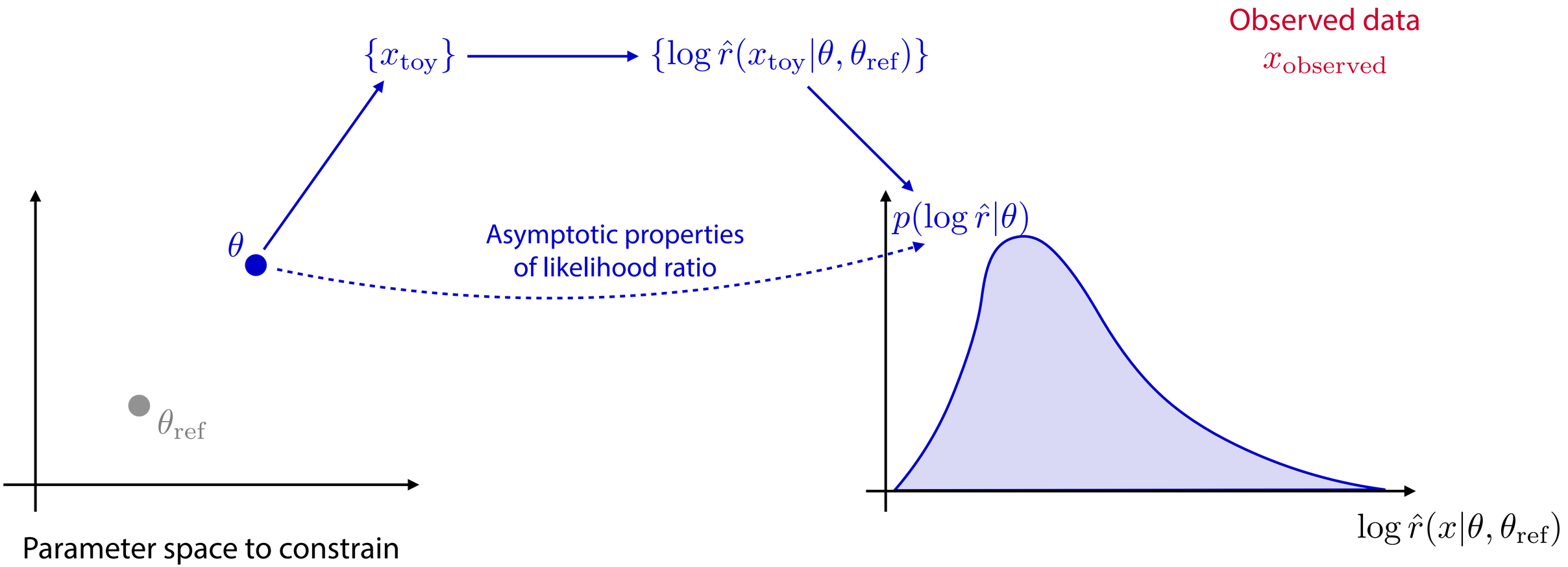



Parameter space to constrain

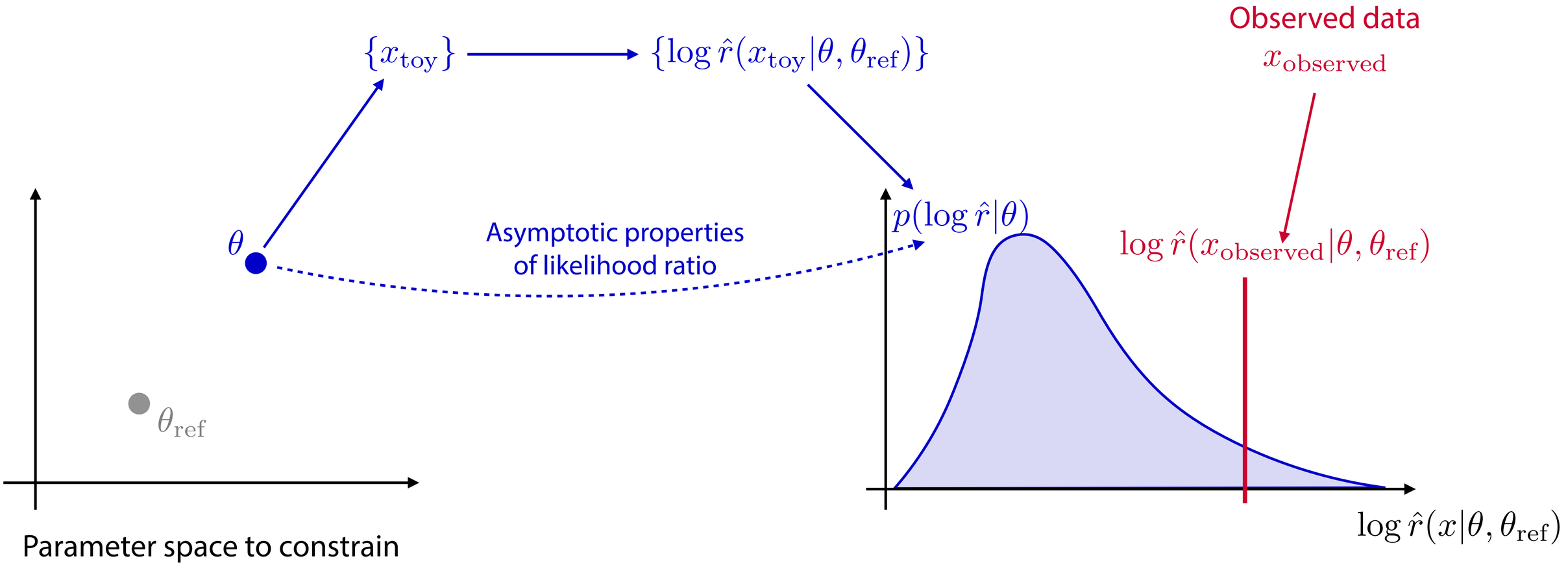
# Limit setting (frequentist, standard ATLAS / CMS practice)



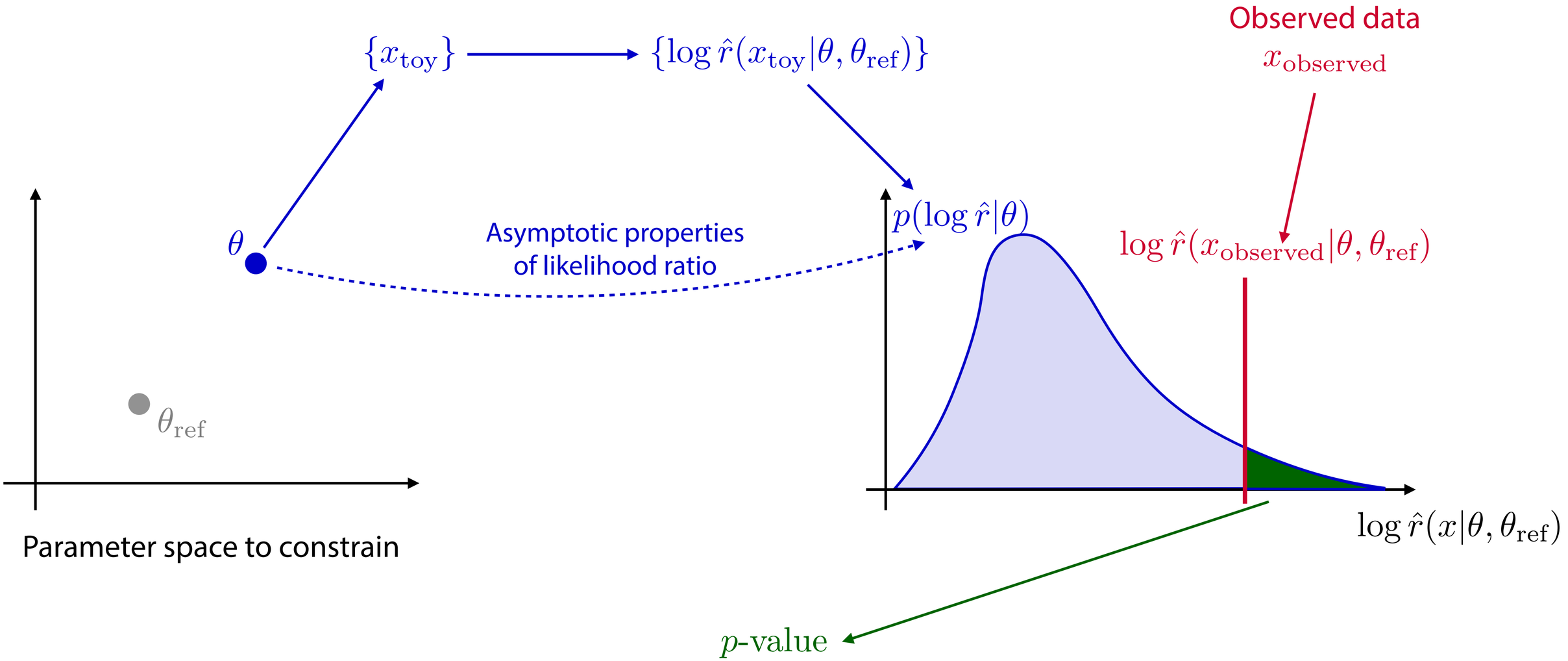
# Limit setting (frequentist, standard ATLAS / CMS practice)



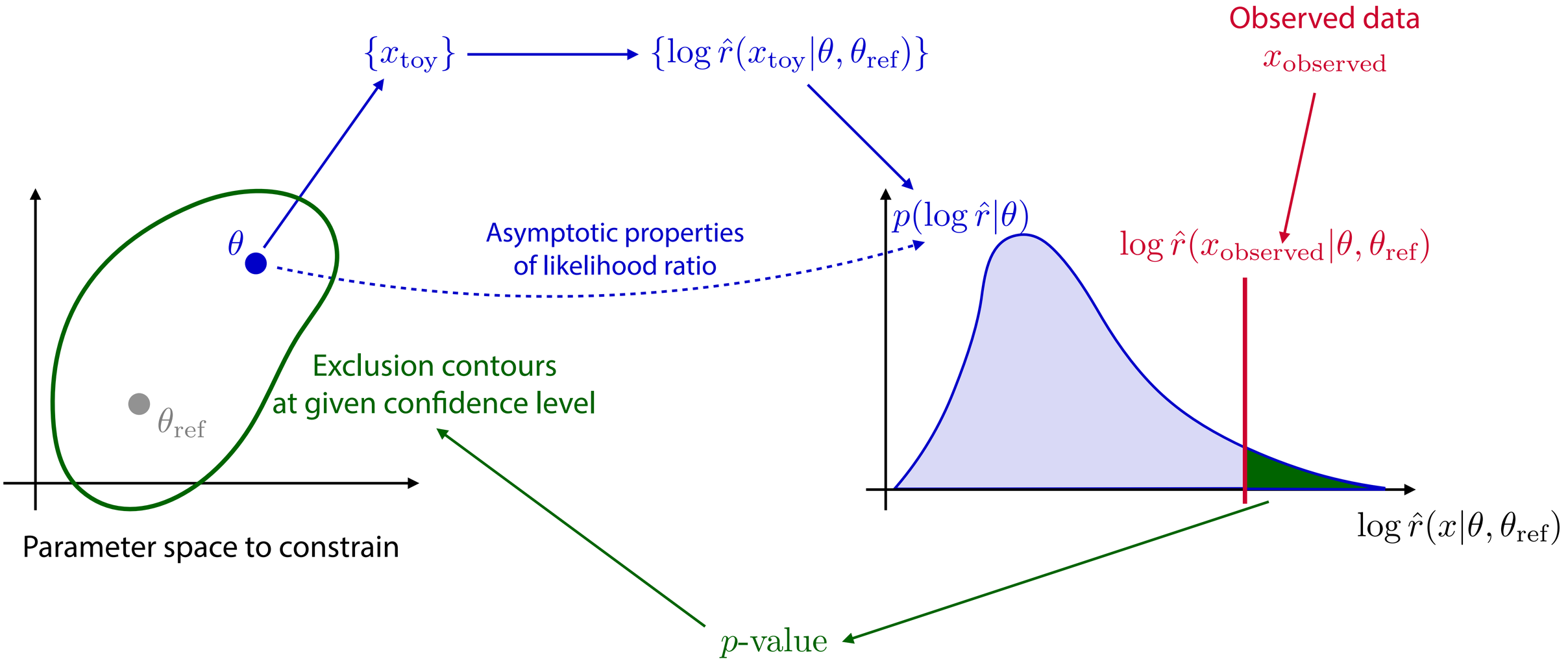
# Limit setting (frequentist, standard ATLAS / CMS practice)



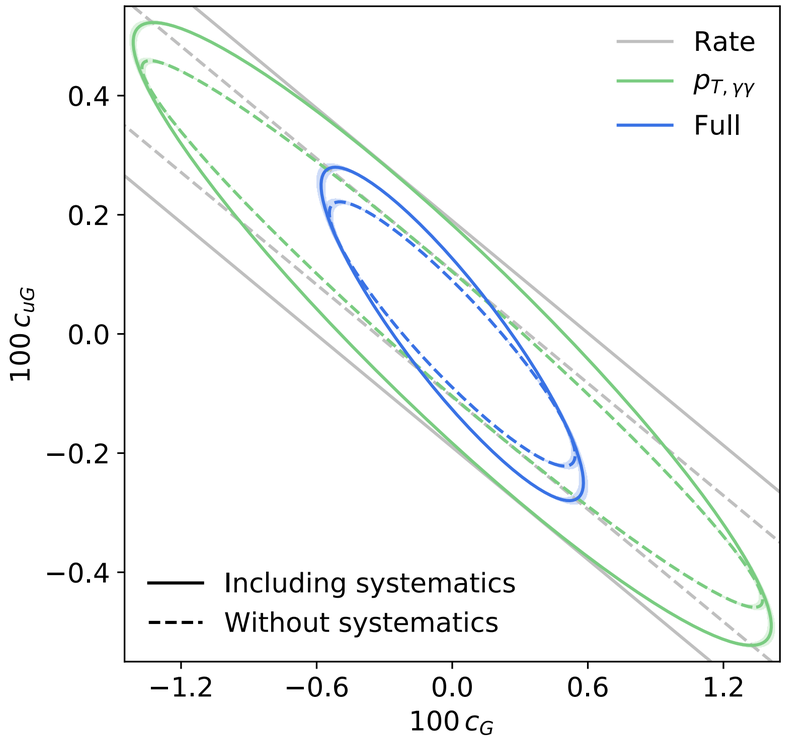
# Limit setting (frequentist, standard ATLAS / CMS practice)



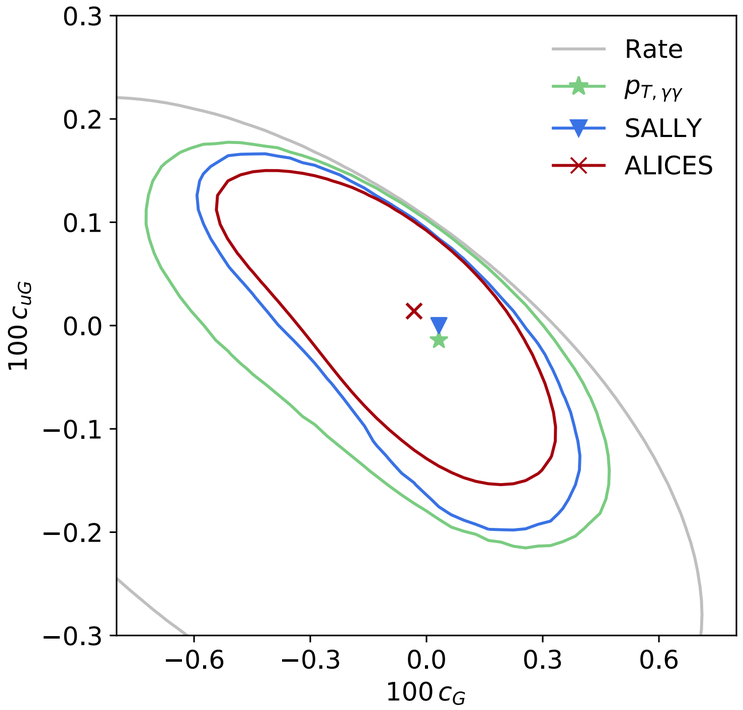
# Limit setting (frequentist, standard ATLAS / CMS practice)



# Types of Inference Results & Sensitivity Summaries



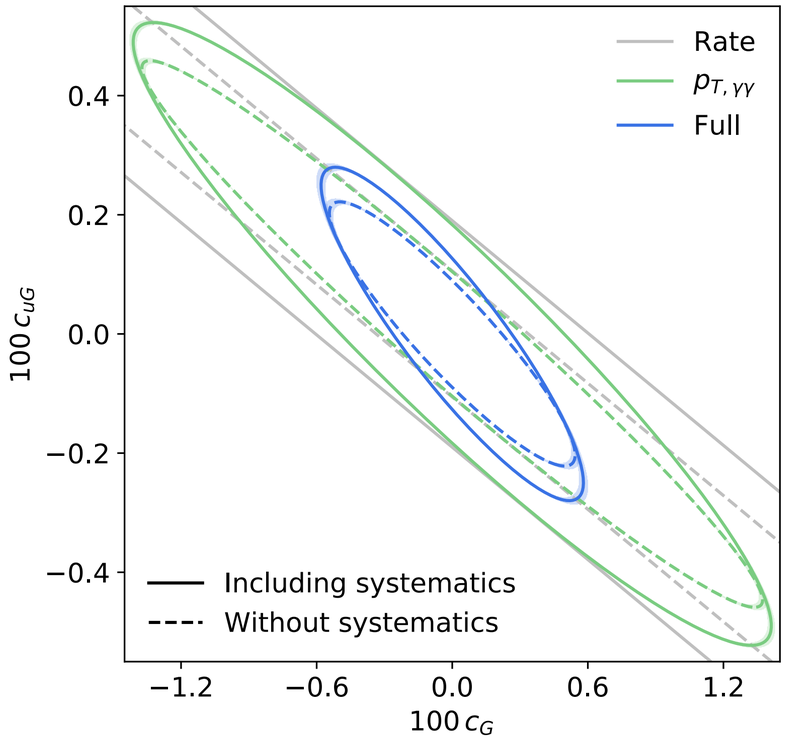

Elliptical Contours
(with / without systematics)



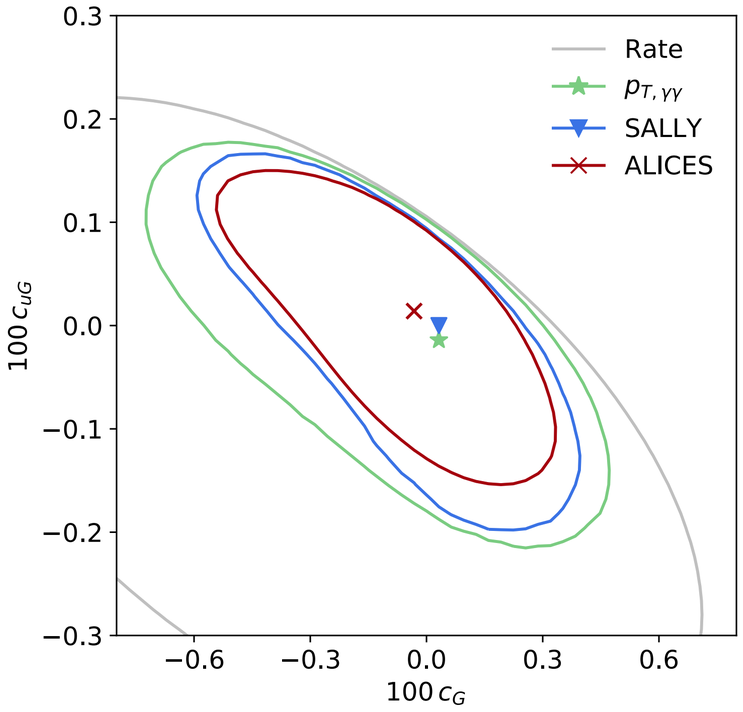

Non-Eliptical Contours

Fisher Information
&
Information Geometry

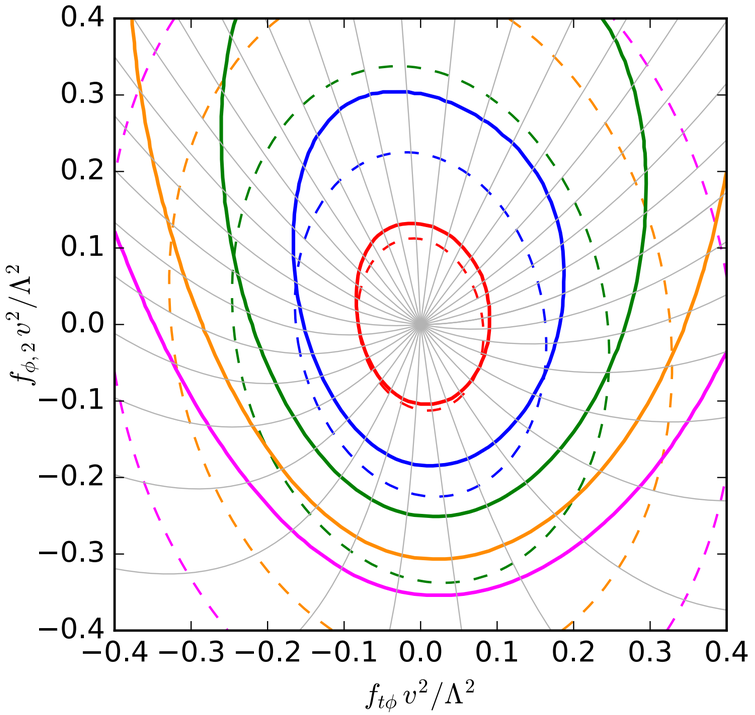
# Types of Inference Results & Sensitivity Summaries




Elliptical Contours
(with / without systematics)




Non-Eliptical Contours




Fisher Information
&
Information Geometry

# MadMiner automates all of these methods.

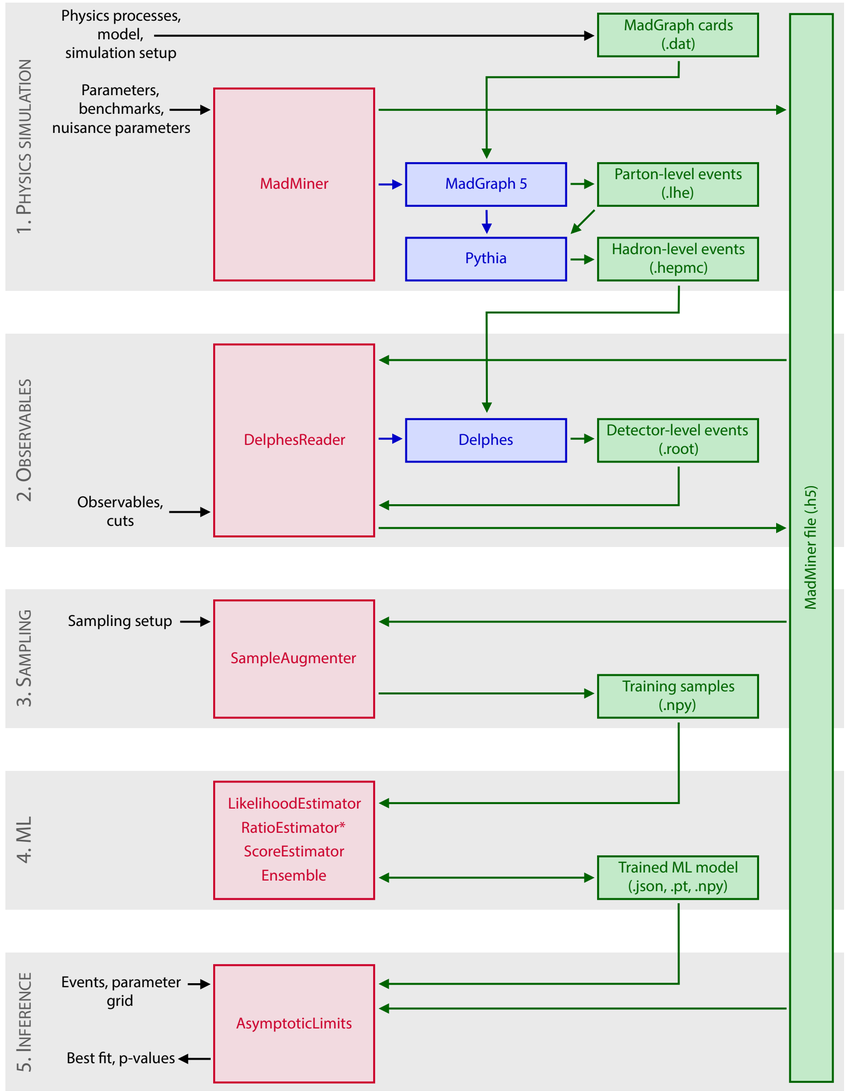
[JB, F. Kling, I. Espejo, K. Cranmer 1907.10621]

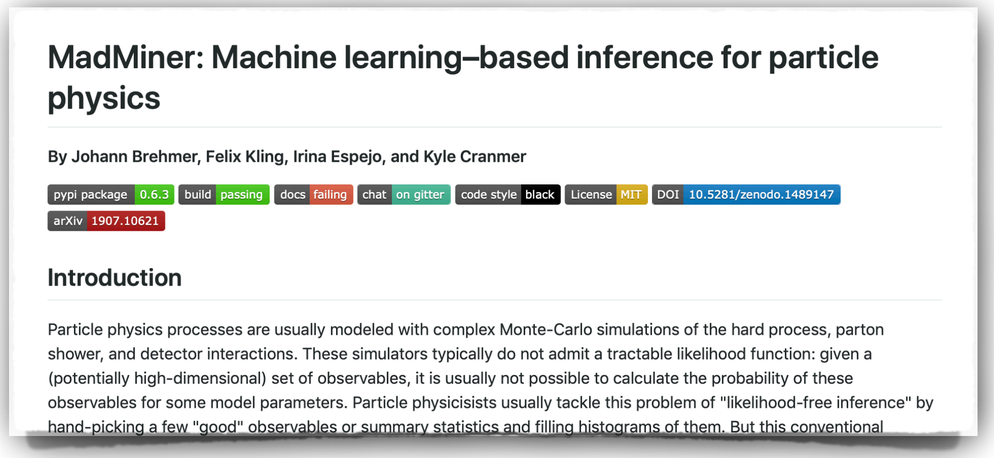
# Automization

[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

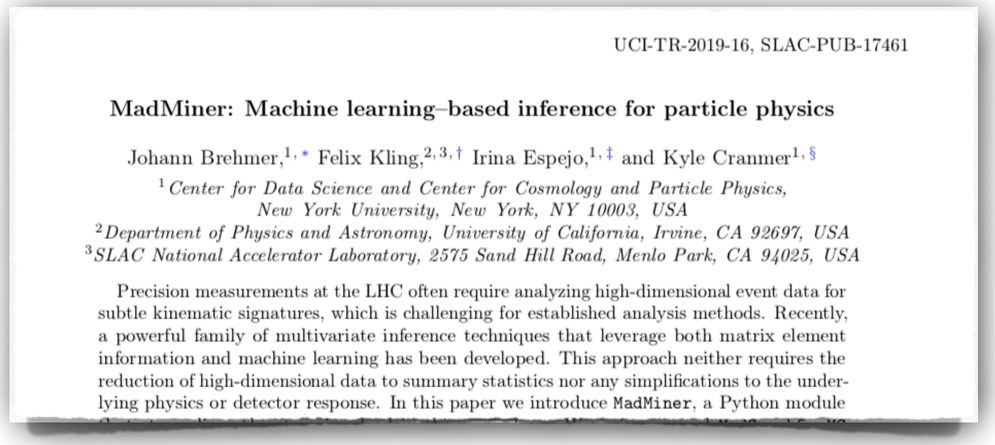
By Johann Brehmer, Felix Kling, Irina Espejo, and Kyle Cranmer

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

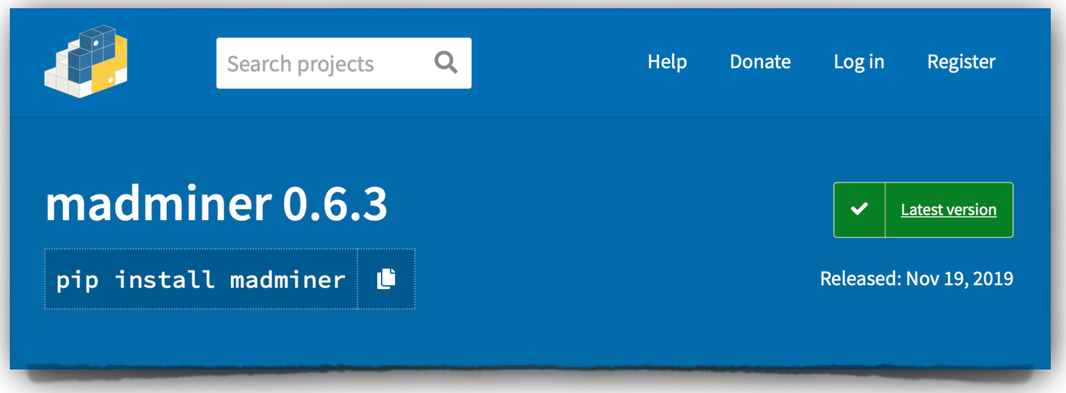
# MadMiner: Machine learning–based inference for particle physics

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



Search projects

Help Donate Log in Register

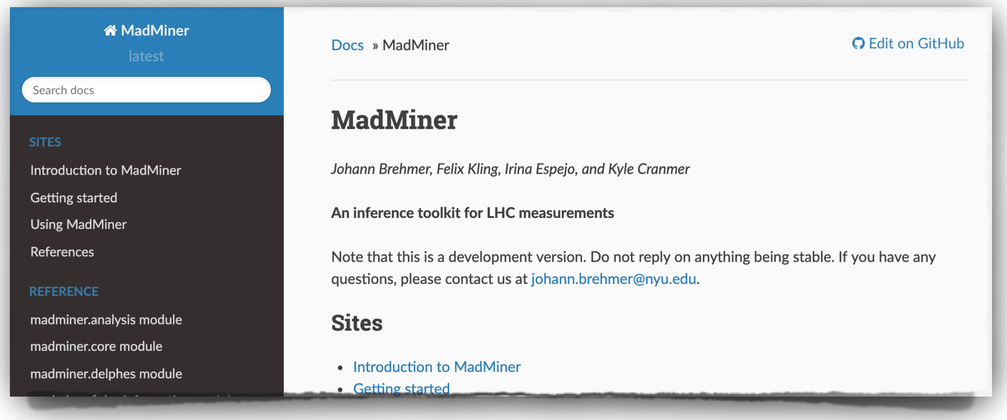
madminer 0.6.3

pip install madminer

Latest version

Released: Nov 19, 2019

Installation:
`pip install madminer`



MadMiner
latest

Search docs

SITES
Introduction to MadMiner
Getting started
Using MadMiner
References

REFERENCE
madminer.analysis module
madminer.core module
madminer.delphes module

Docs » MadMiner

Edit on GitHub

# MadMiner

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

**An inference toolkit for LHC measurements**

Note that this is a development version. Do not reply on anything being stable. If you have any questions, please contact us at johann.brehmer@nyu.edu.

## Sites

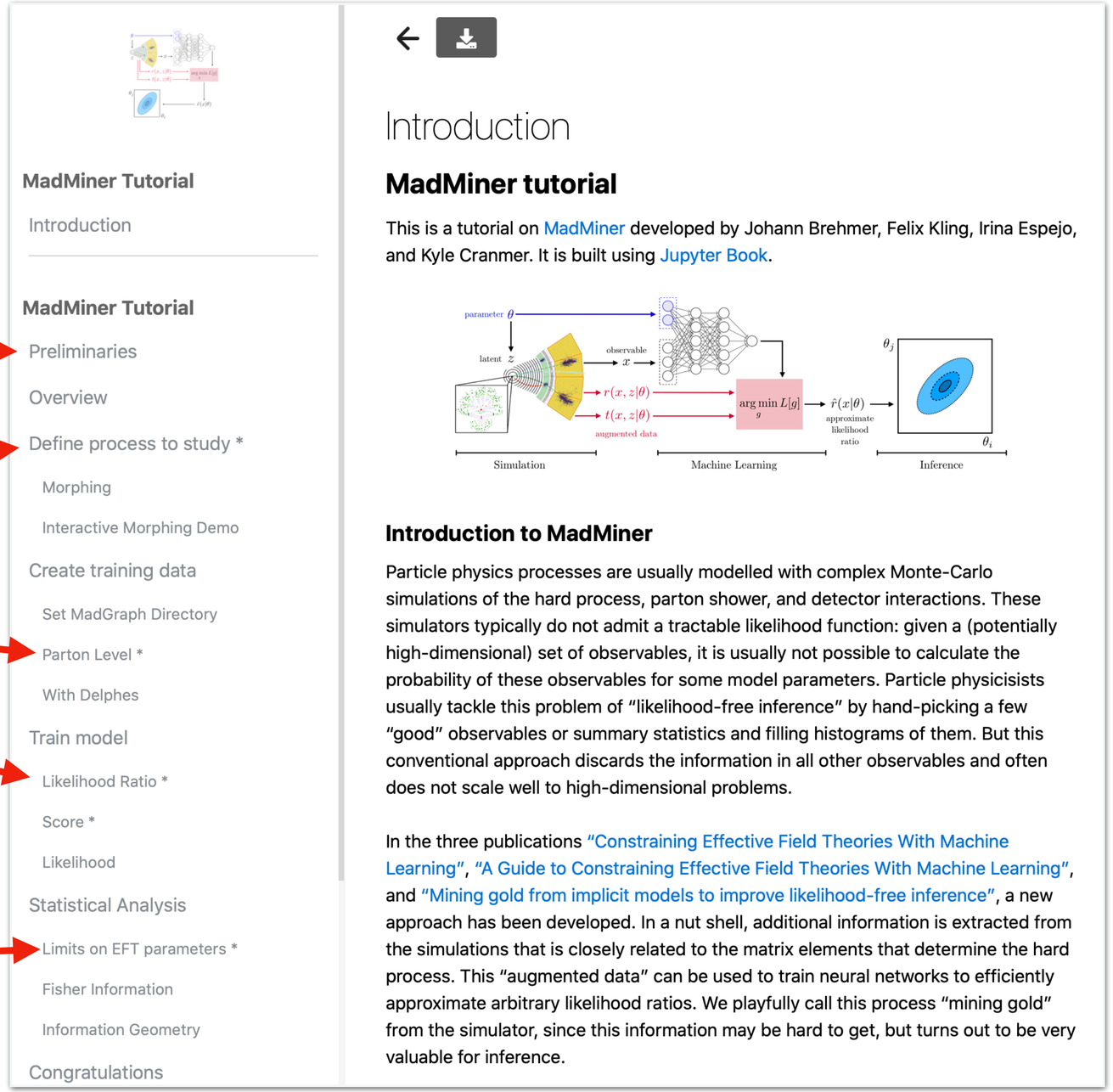
- Introduction to MadMiner
- Getting started

API documentation:
madminer.readthedocs.io

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

# Hands-on Tutorial

- Step-by-step instructions
- Do Preliminaries
  - Need to install Docker
  - And then start Jupyter
- Step 1 is fast
- Step 2 takes ~25 min
- Step 3 takes ~20 min
- While they are running I will lecture
- Then we will finish with results

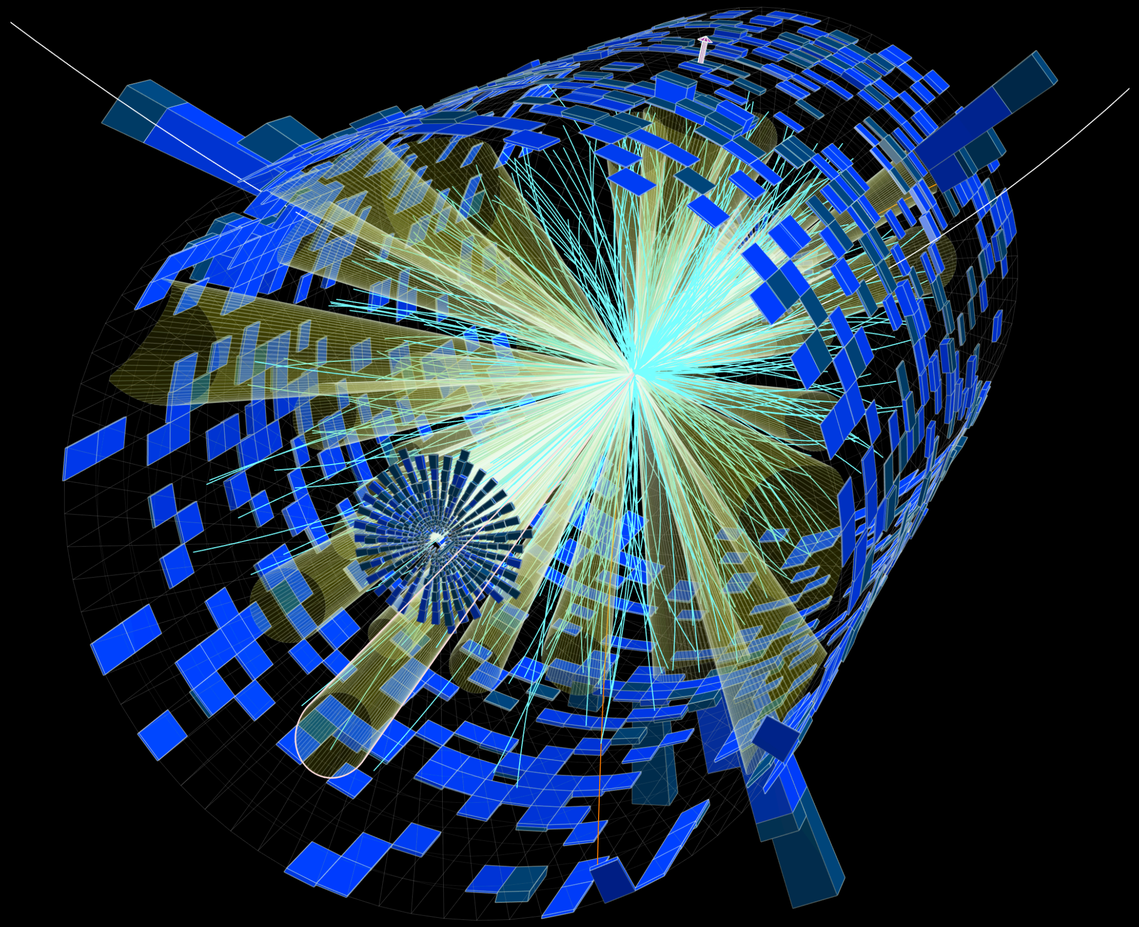


NYU Center for Data Science

Center for Cosmology and particle physics

# MACHINE LEARNING

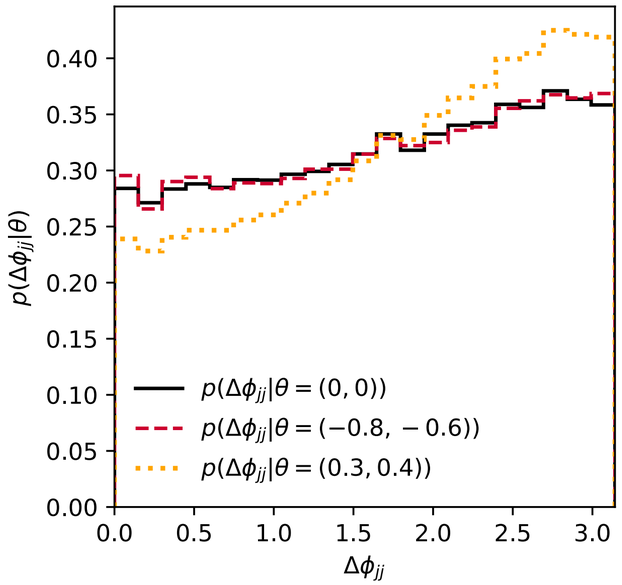
## LECTURE 4



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

These techniques let us constrain effective theories more effectively.

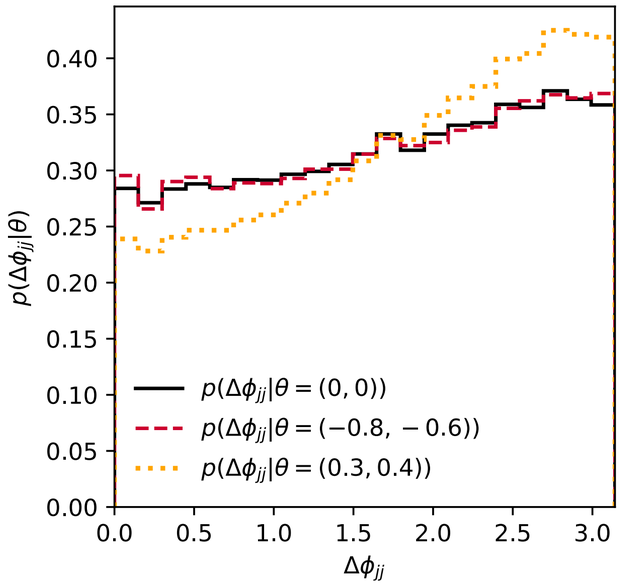
# Perfect match for EFT measurements

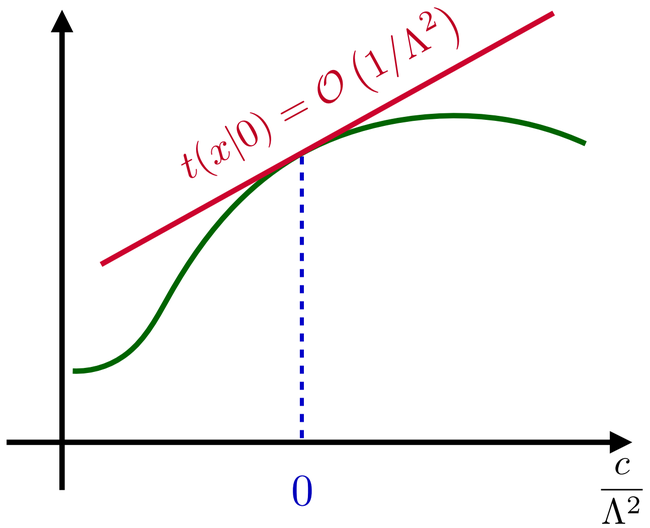



- Good for subtle kinematic effects

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

# Perfect match for EFT measurements



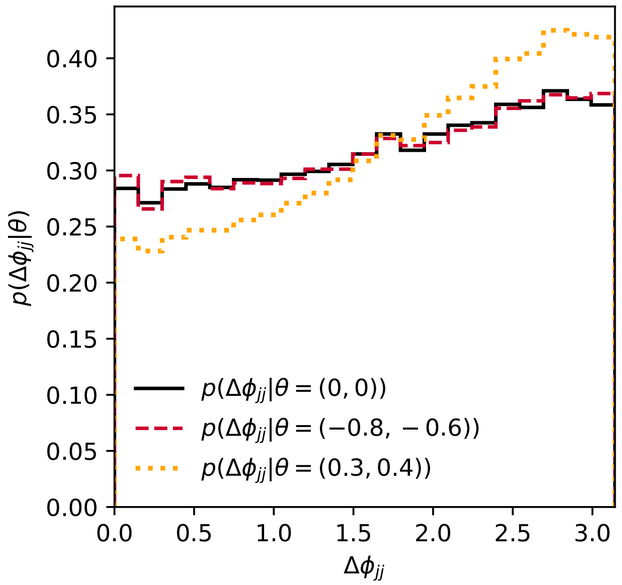

- Good for subtle kinematic effects

  (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

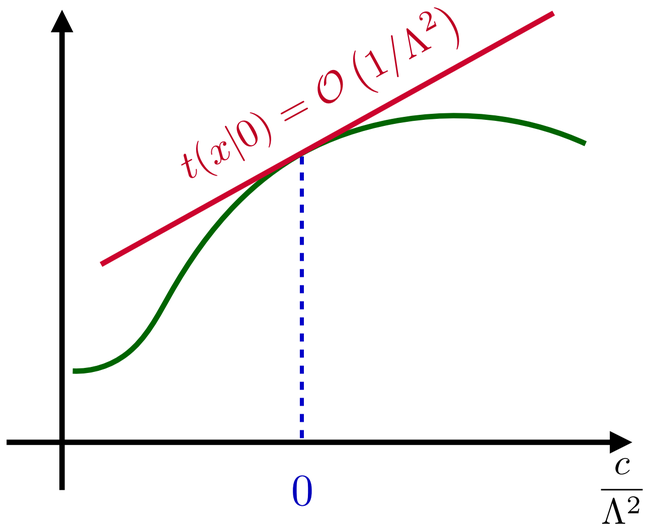
  (SMALLY?)

# Perfect match for EFT measurements



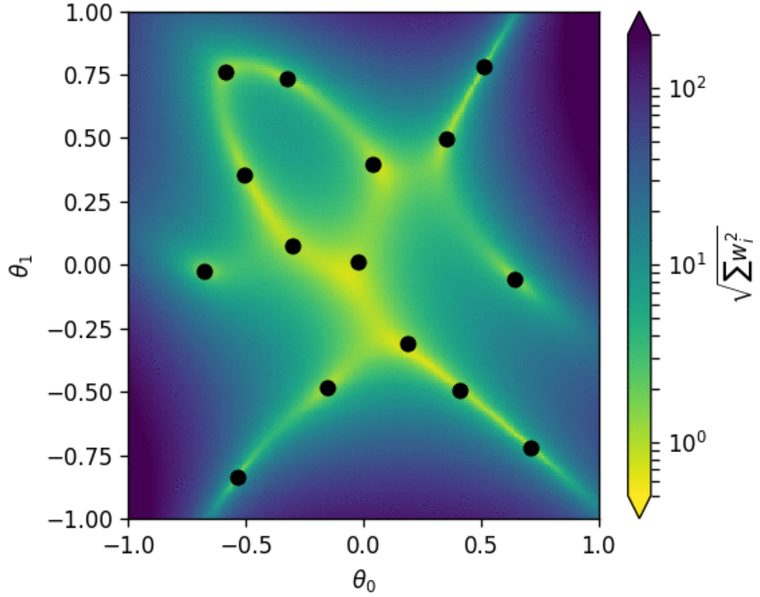

- Good for subtle kinematic effects

  (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?)



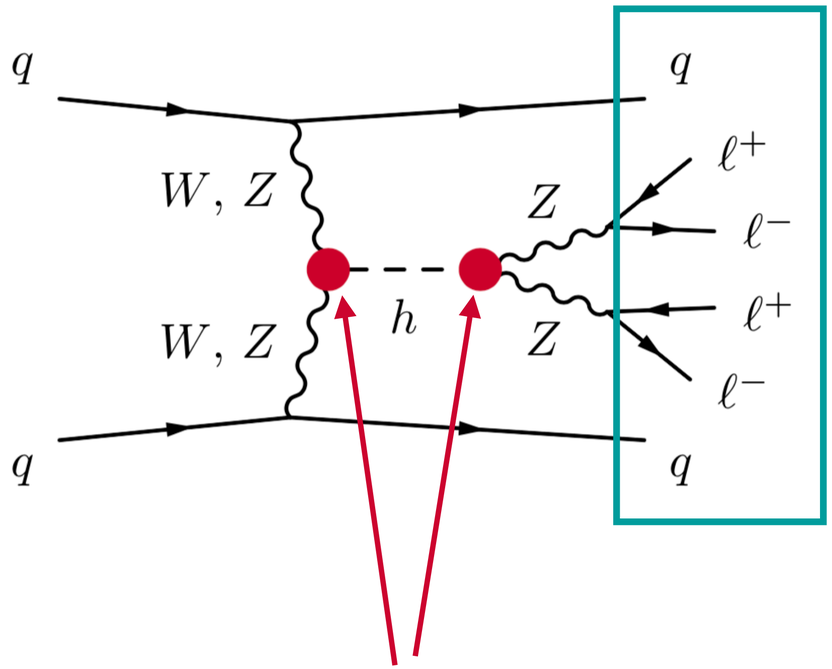

- Morphing techniques allow fast reweighting to any parameter points

  [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]

at least 16-dimensional
observable space



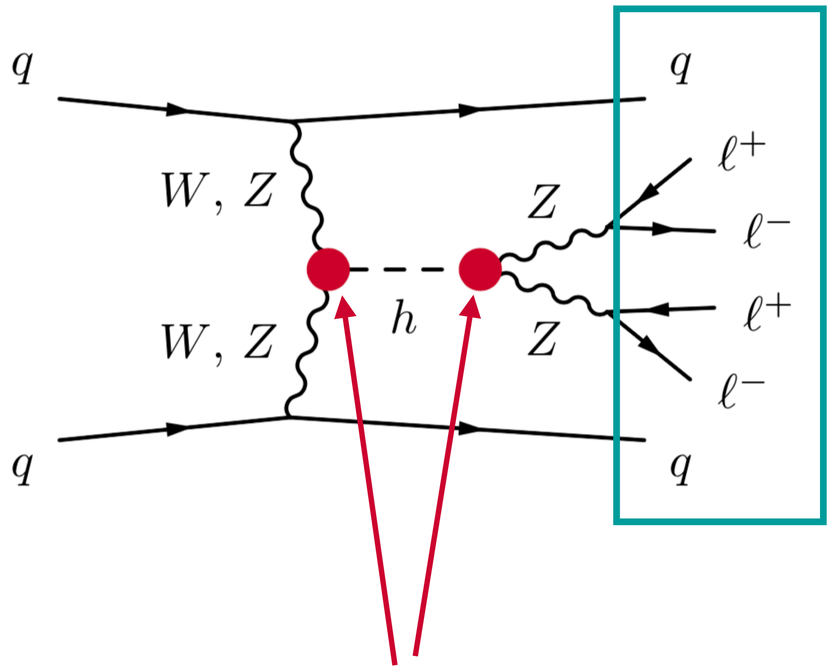

Exciting new physics might hide here!
We parameterize it with two EFT coefficients:

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

# Proof of concept: Higgs production in weak boson fusion

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

at least 16-dimensional observable space




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

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

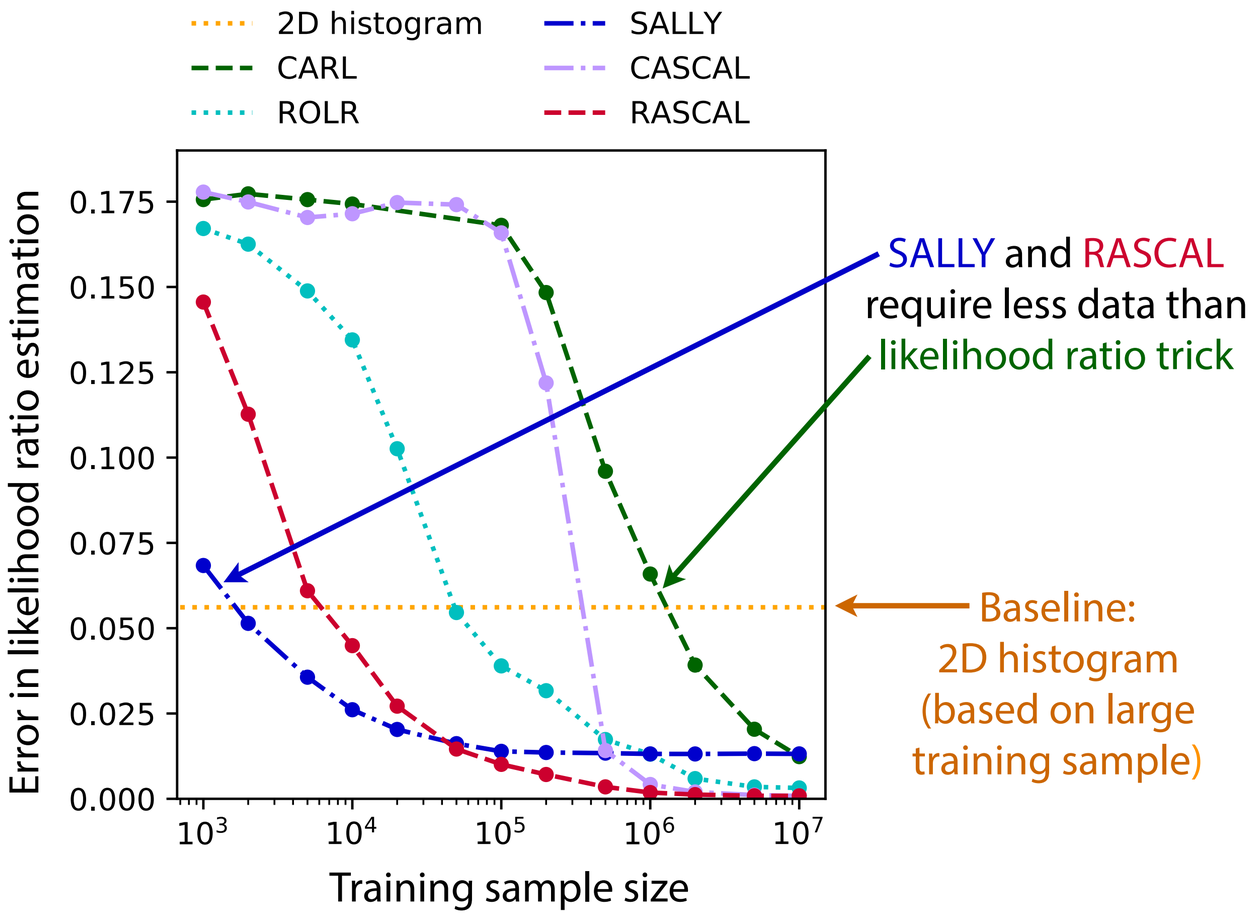
Goal: constrain the two EFT parameters

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

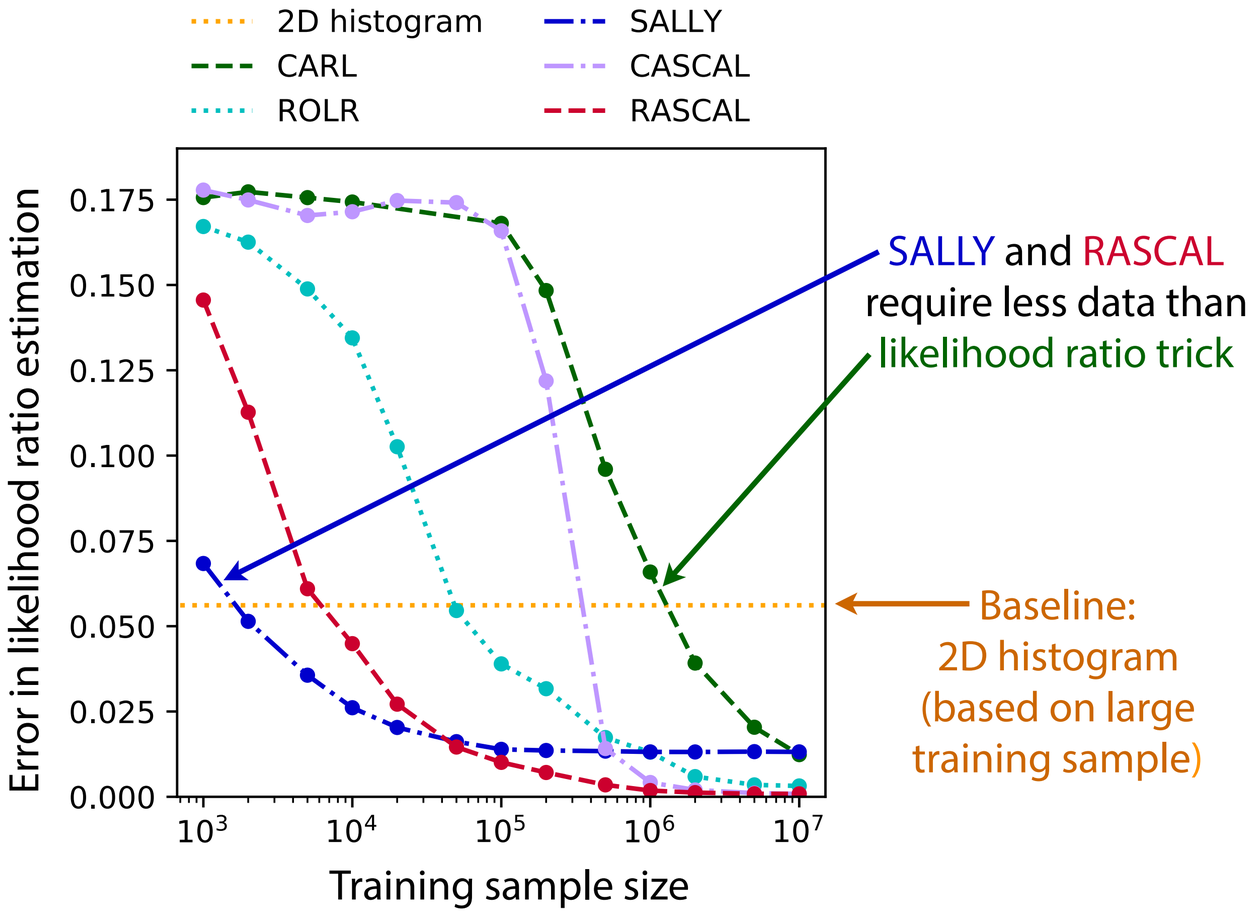
Two scenarios:

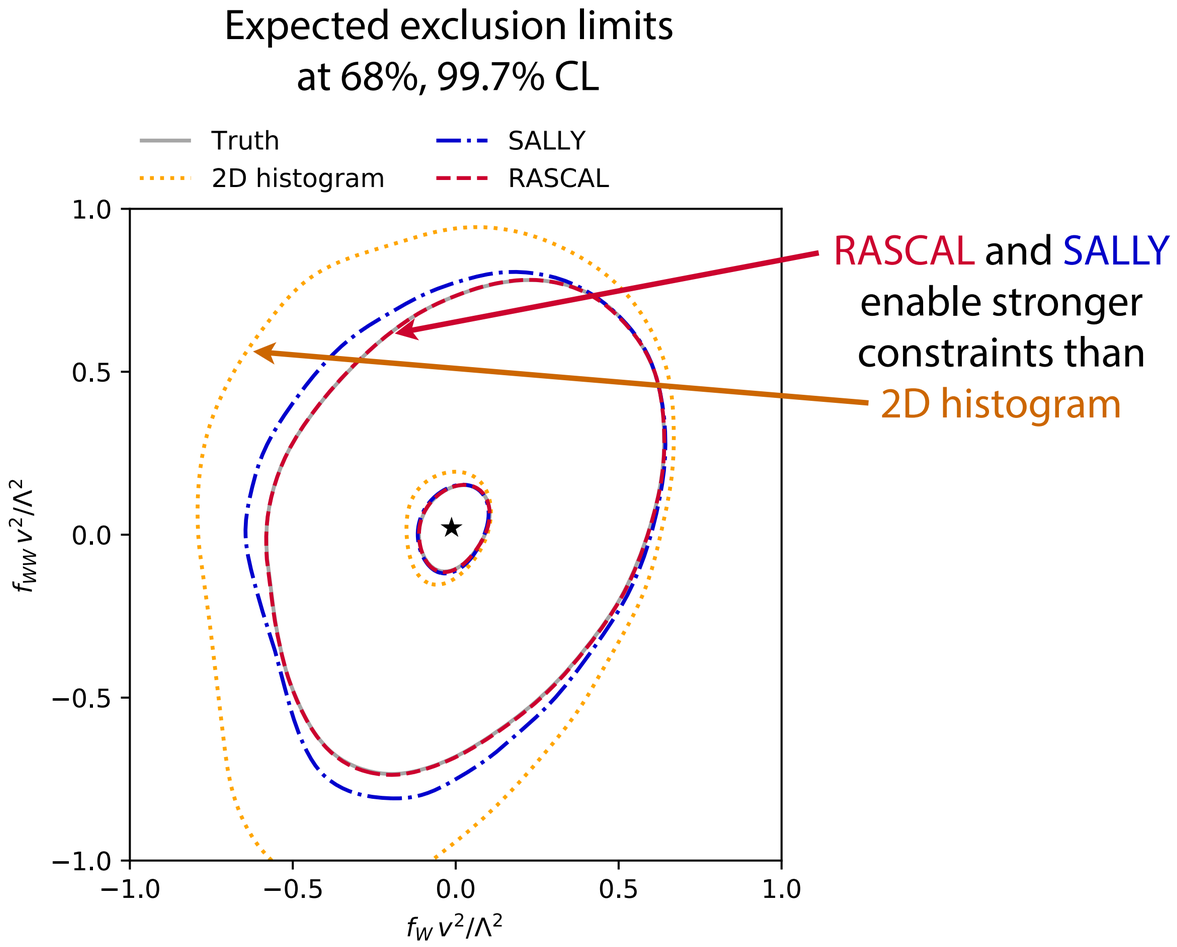
- Simplified setup in which we can compare to true likelihood
- "Realistic" simulation with approximate detector effects

# Proof of concept: Stronger constraints with less training data

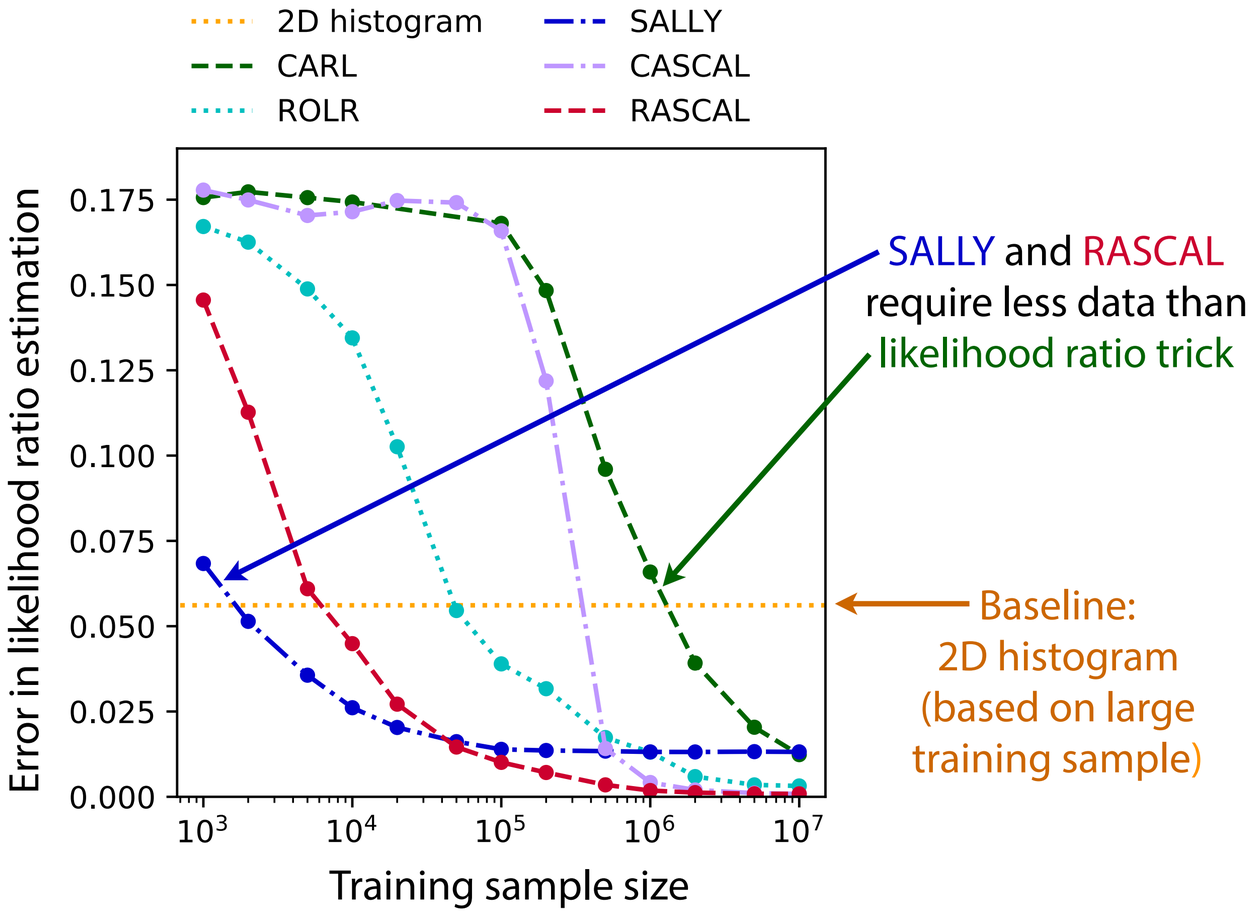


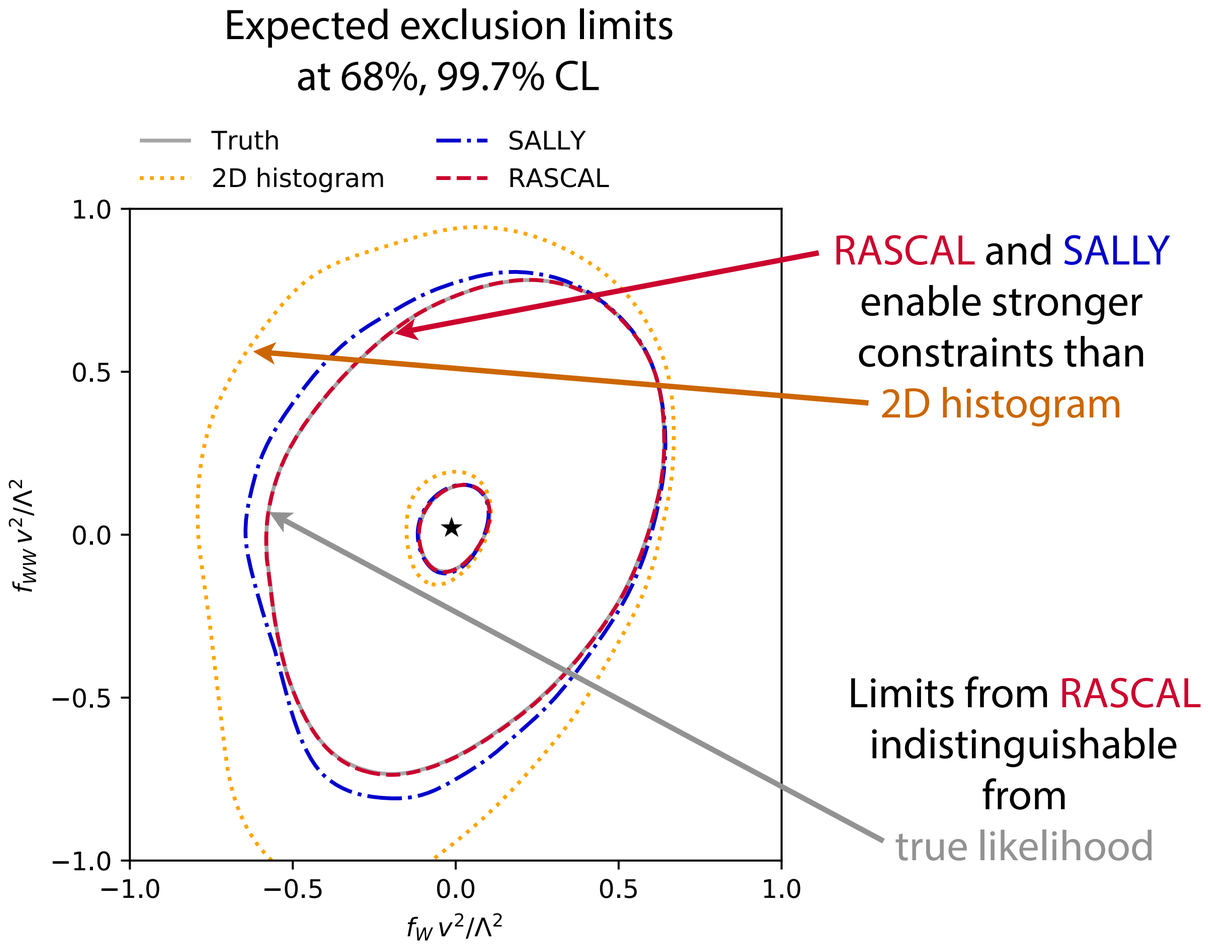
# Proof of concept: Stronger constraints with less training data



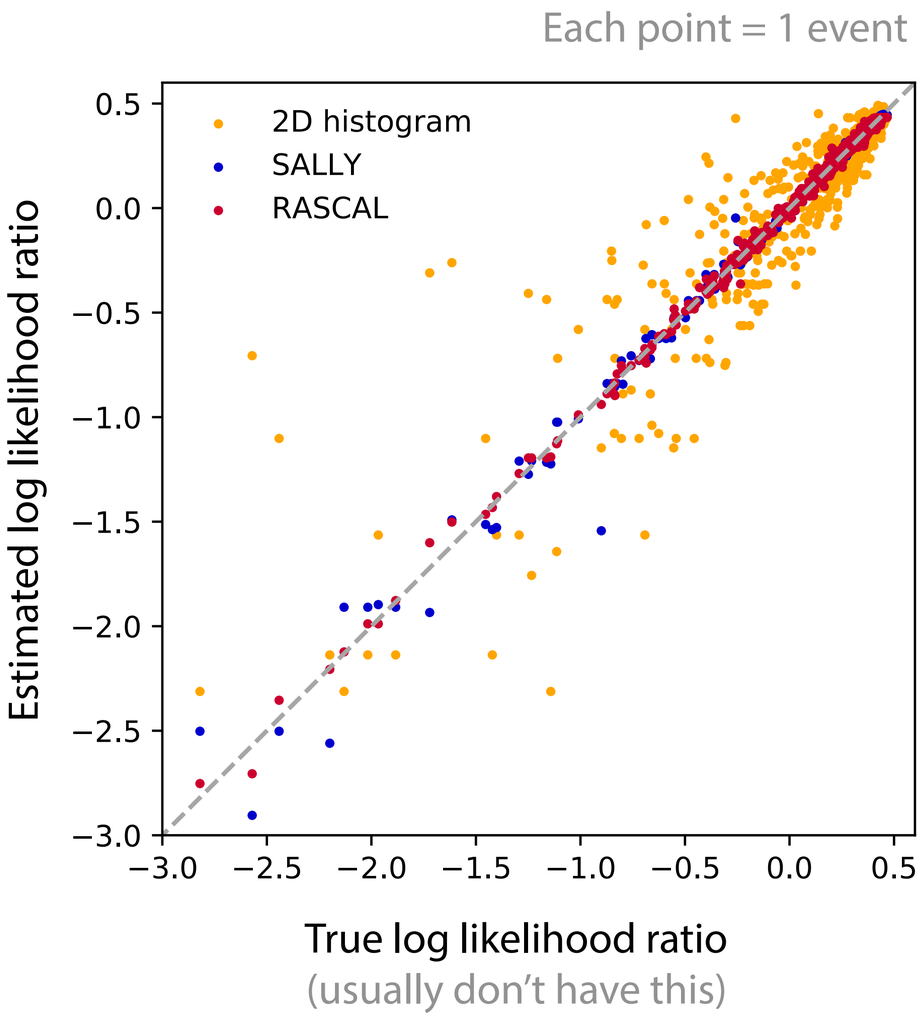



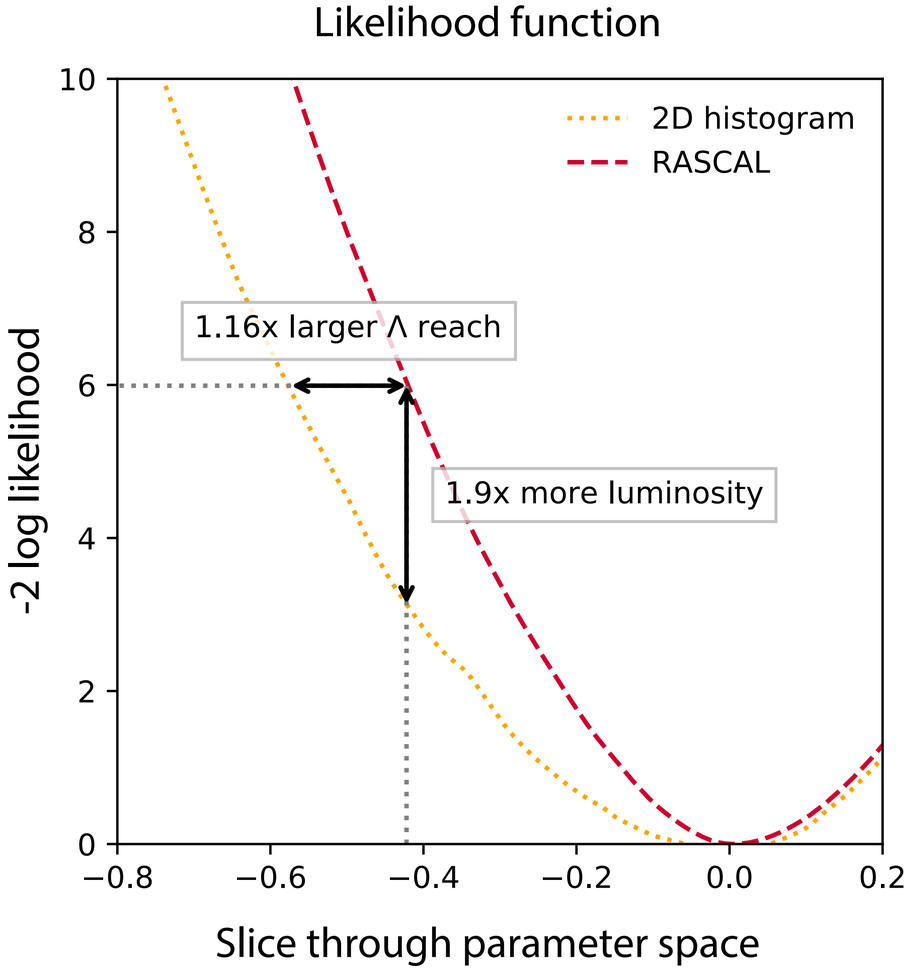
# Proof of concept: Stronger constraints with less training data



# 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

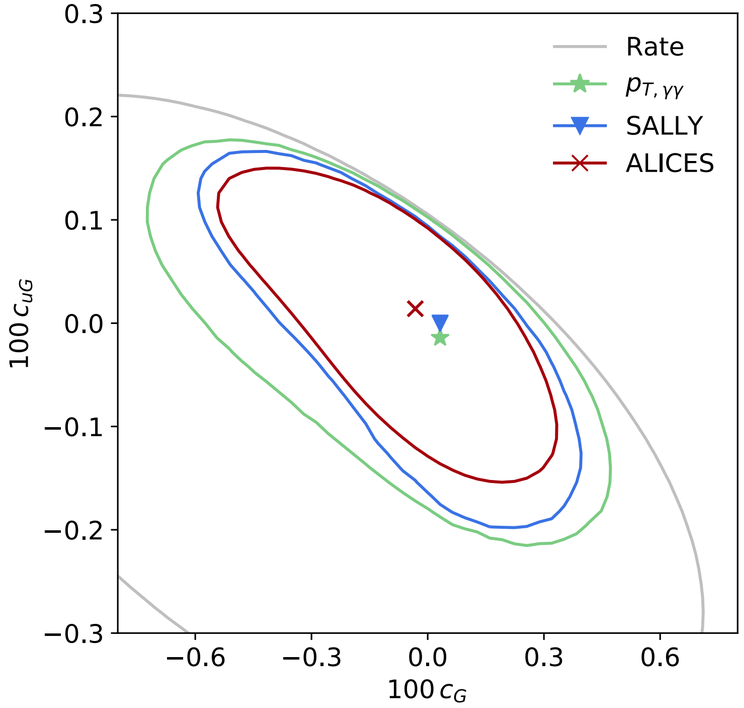
$$pp \to t\bar{t}\,h \to (b\ell^+)\,(\bar{b}\ell^-)\,(\gamma\gamma)\,E_T^{\text{miss}}$$

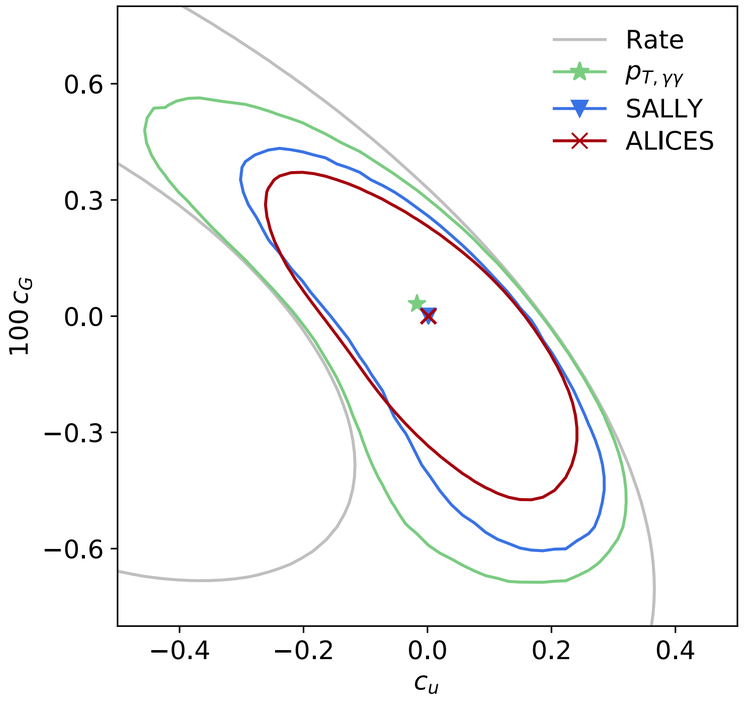
with MadGraph + Pythia + Delphes

- 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:

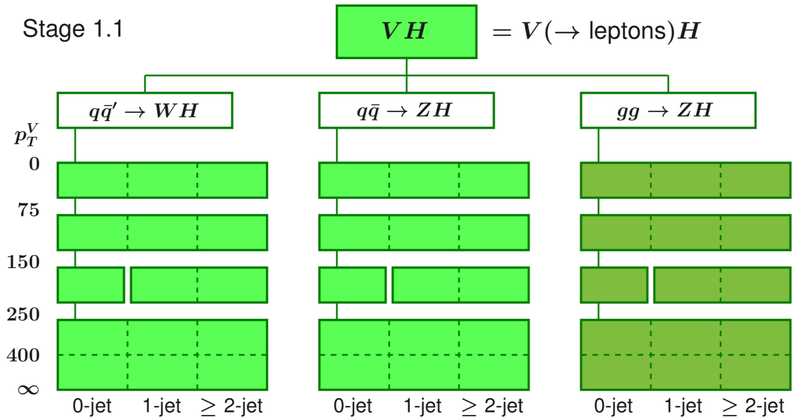


# Benchmarking STXS in WH

[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

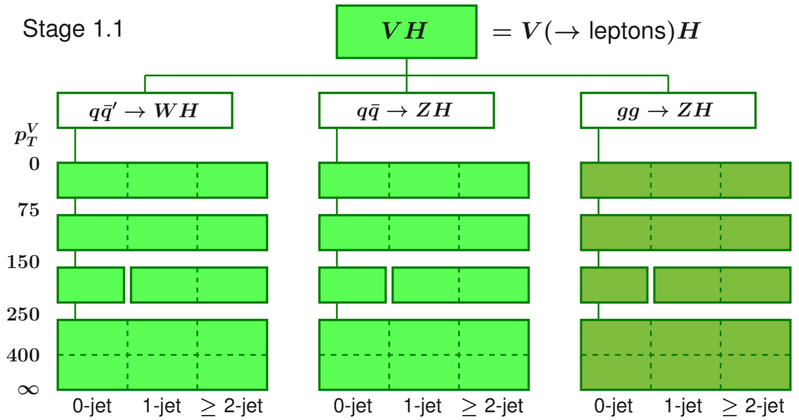
$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\Box} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger\phi)\Box(\phi^\dagger\phi) - \frac{1}{4}(\phi^\dagger D^\mu\phi)^*(\phi^\dagger D_\mu\phi)$$
$$\mathcal{O}_{HW} = \phi^\dagger\phi W^a_{\mu\nu}W^{\mu\nu a}$$
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu\phi)(\overline{Q}_L\sigma^a\gamma^\mu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

  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]

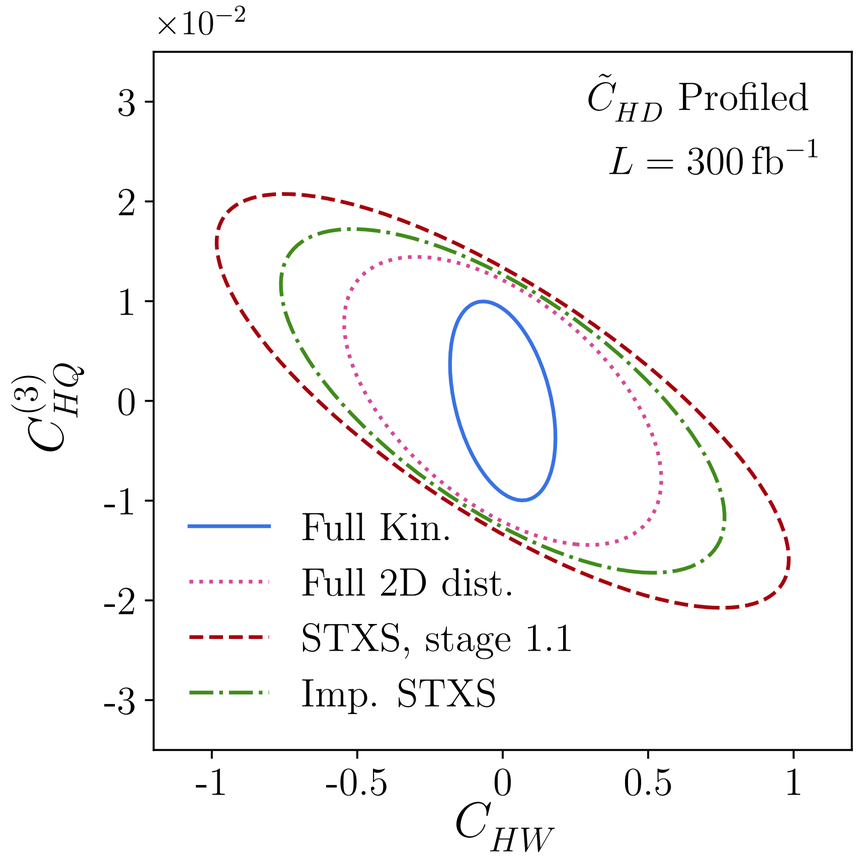



- Let's check! How much information on

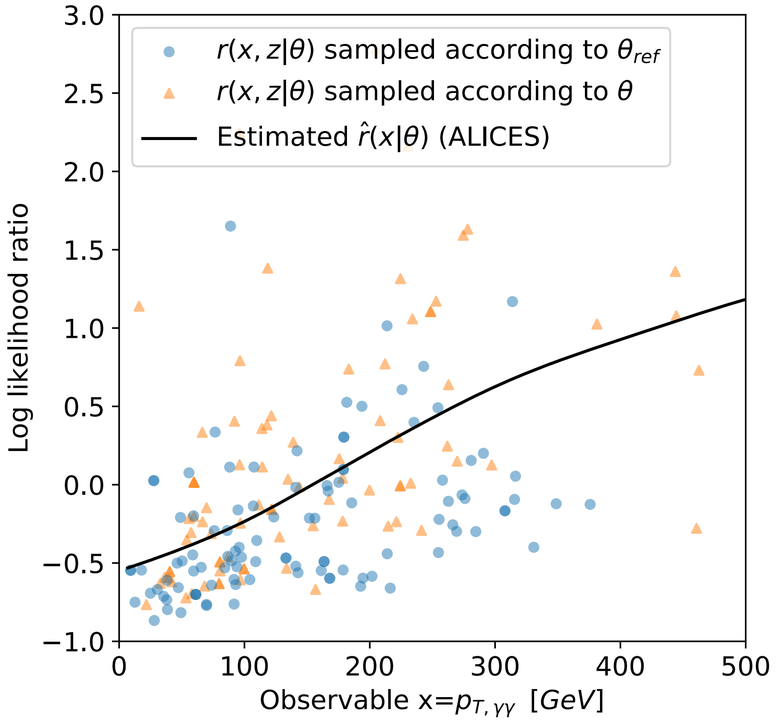
$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\Box} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger\phi)\Box(\phi^\dagger\phi) - \frac{1}{4}(\phi^\dagger D^\mu\phi)^*(\phi^\dagger D_\mu\phi)$$
$$\mathcal{O}_{HW} = \phi^\dagger\phi W^a_{\mu\nu}W^{\mu\nu a}$$
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu\phi)(\overline{Q}_L\sigma^a\gamma^\mu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

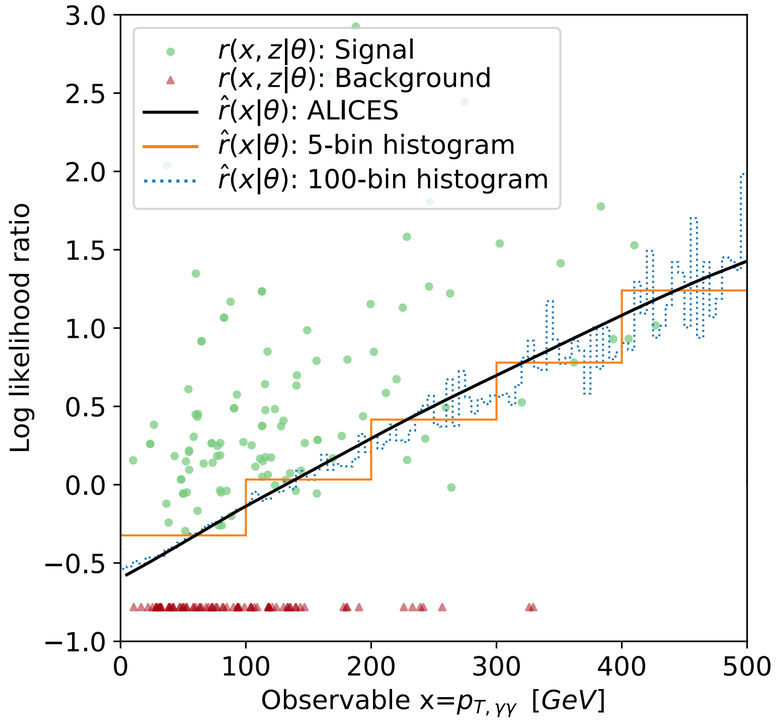
  can we extract from $pp \to WH \to \ell\nu\ b\bar{b}$ ?

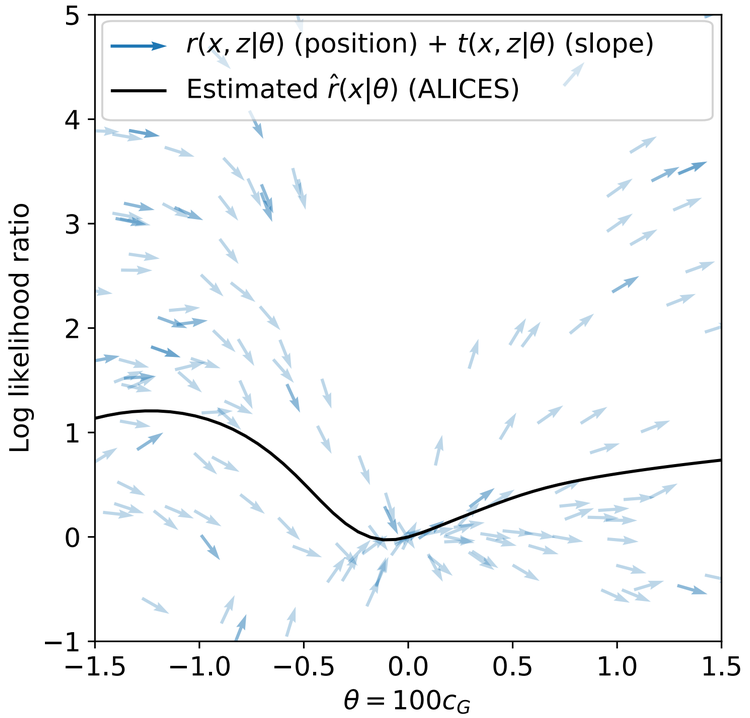
- Results: STXS are indeed sensitive to operators, adding a few more bins improve them, but a multivariate analysis is still stronger



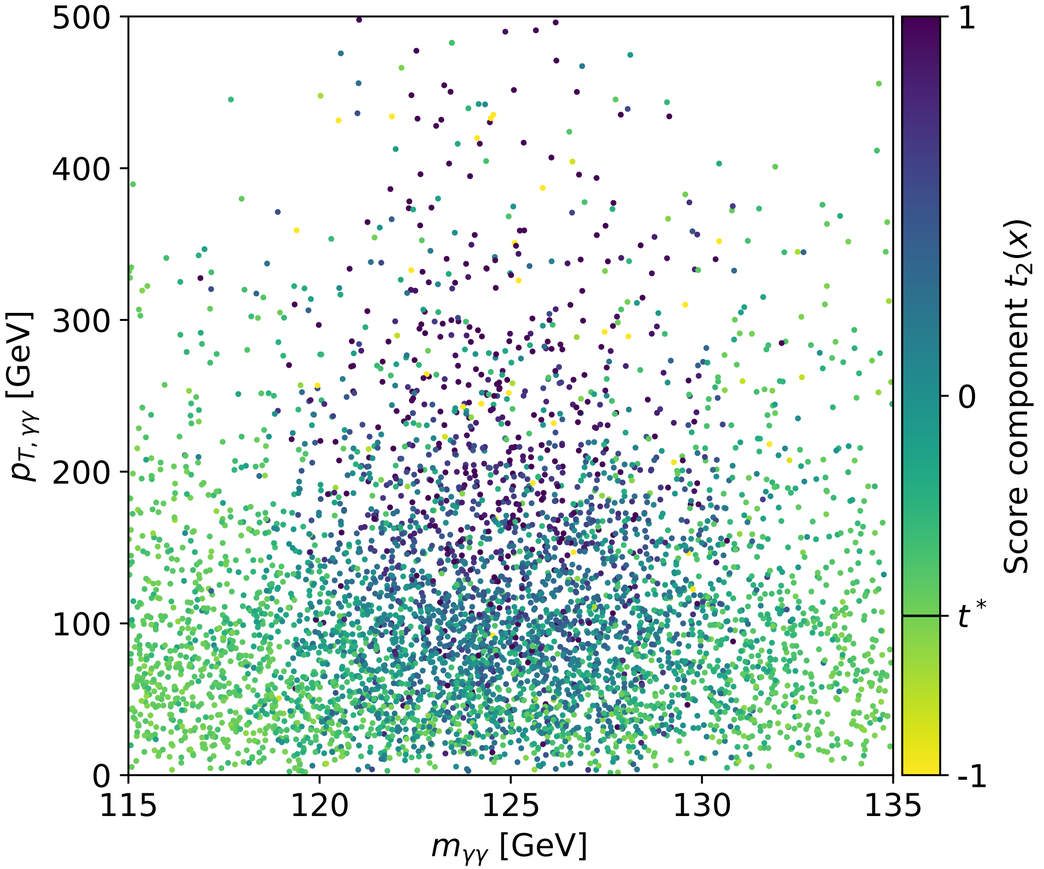
# ttH: 1D illustration

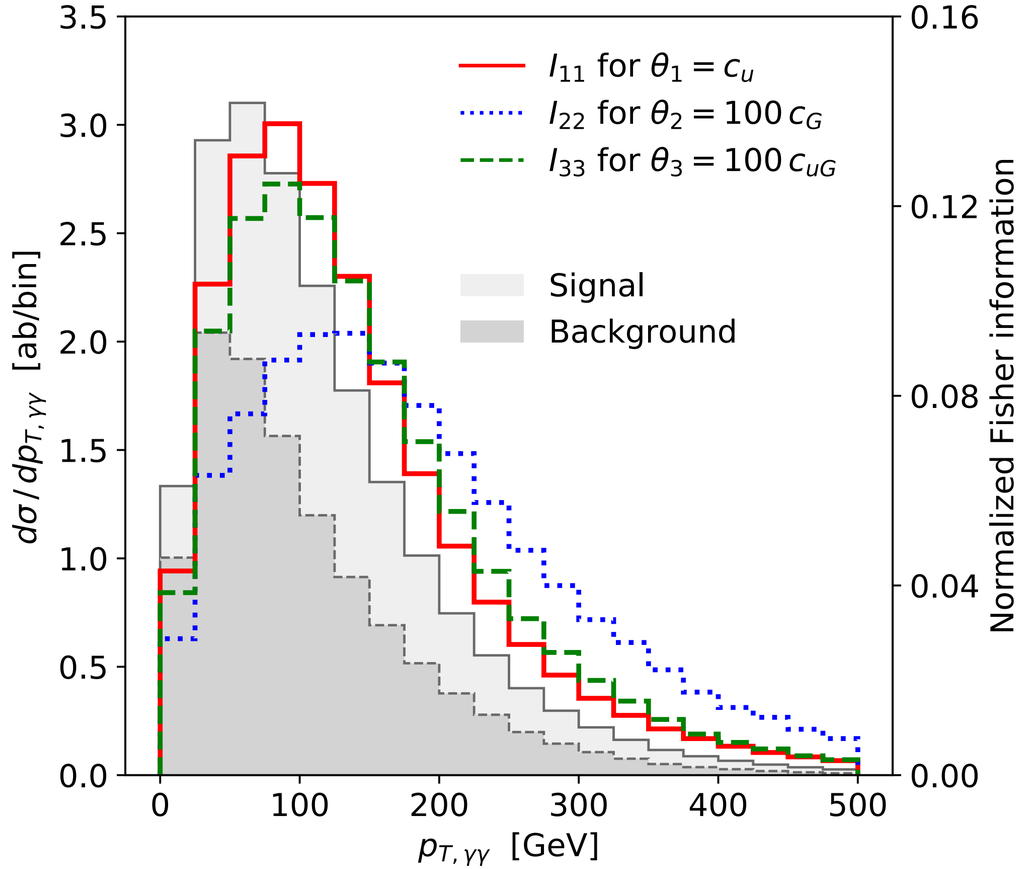


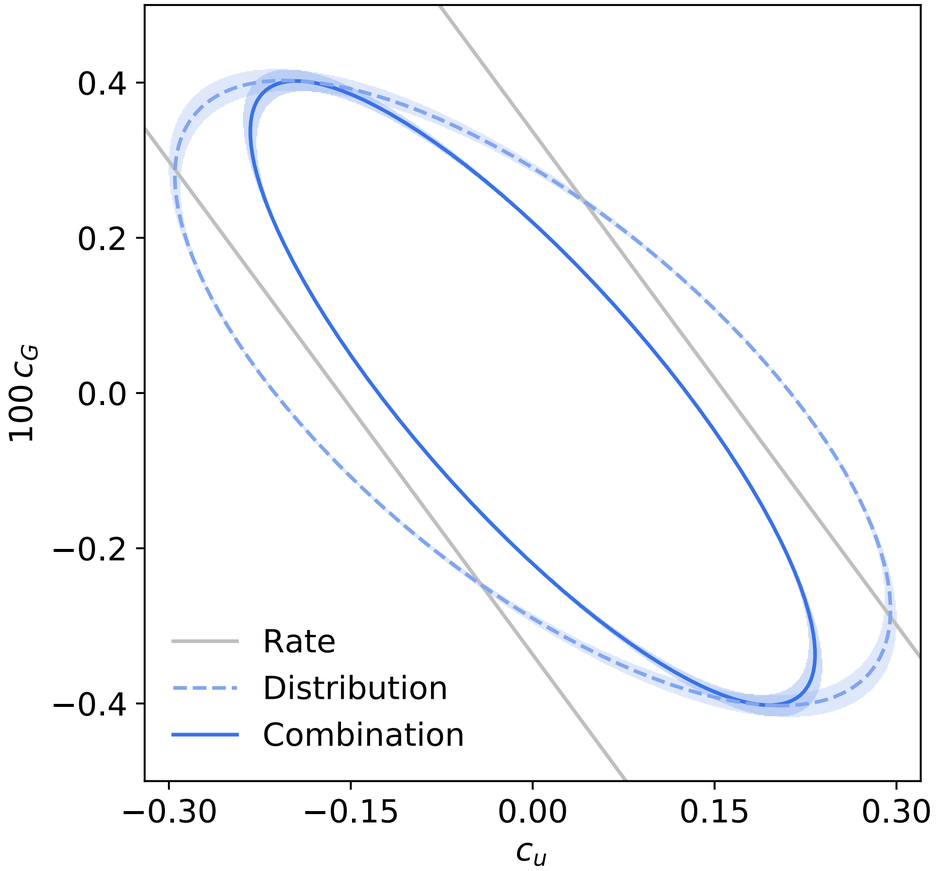



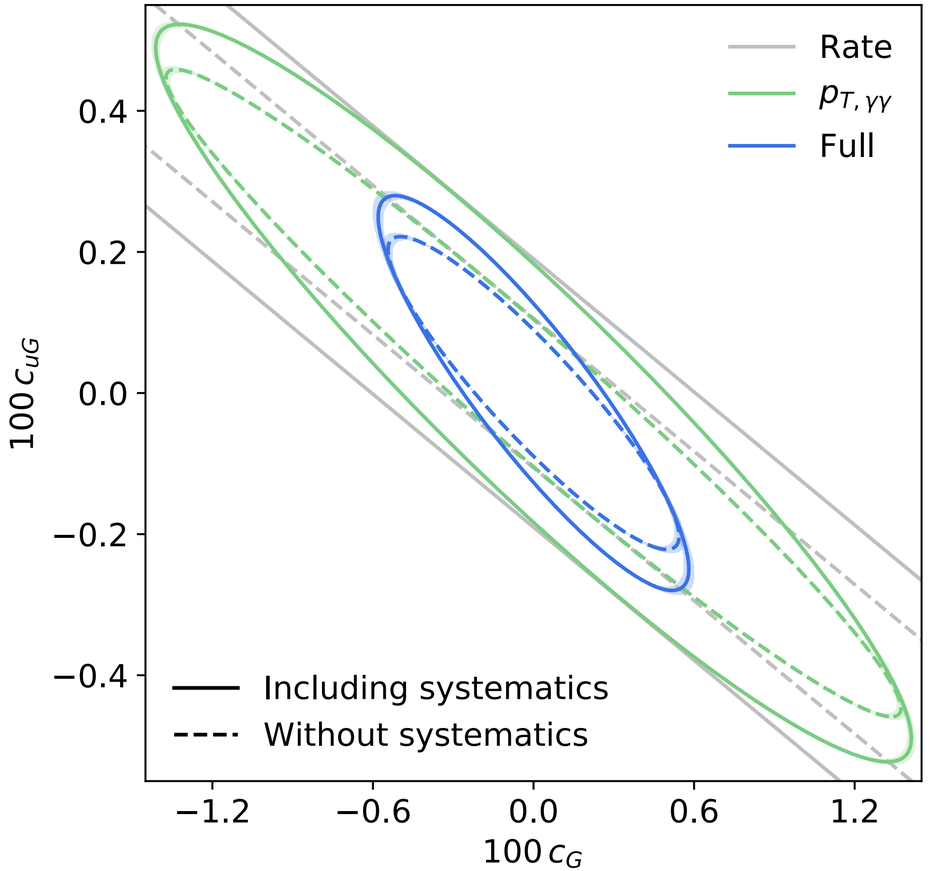
# ttH: score and information vs pT and m



# ttH: more information results

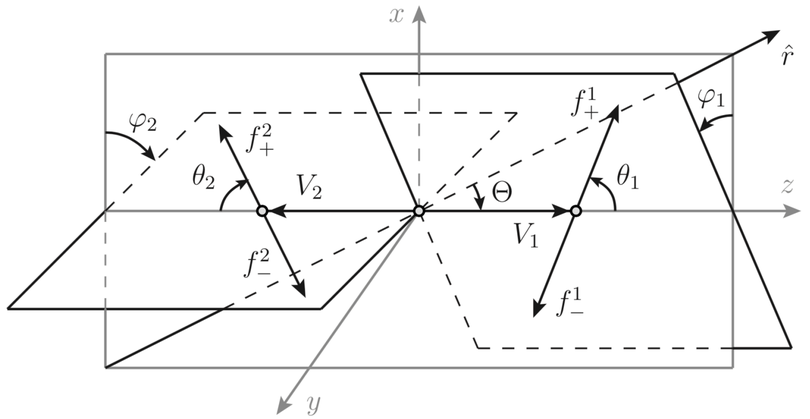


# Diboson production

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

  ⇒ Reduced sensitivity to new physics

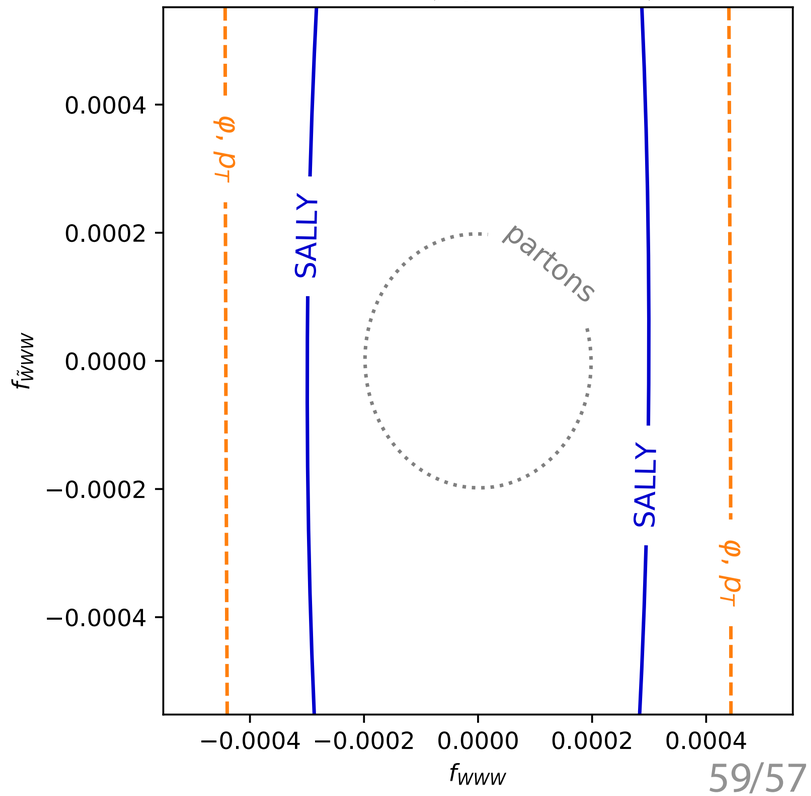



- "Diboson interference resurrection":
  an angular variable $\varphi$ can be constructed to be sensitive to this interference
  [G. Panico, F. Riva, A. Wulzer 1708.07823;
  A. Azatov, D. Barducci, E. Venturini 1901.04821]

- 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]

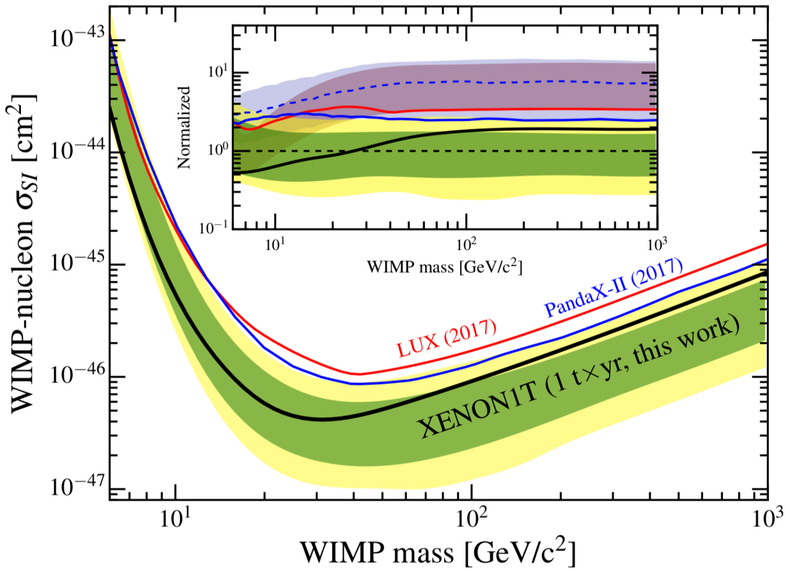
- Preliminary results: we can extract more information when we analyze events with SALLY than with histograms of $\varphi$ and standard observables

Very preliminary results



These methods work beyond the LHC.

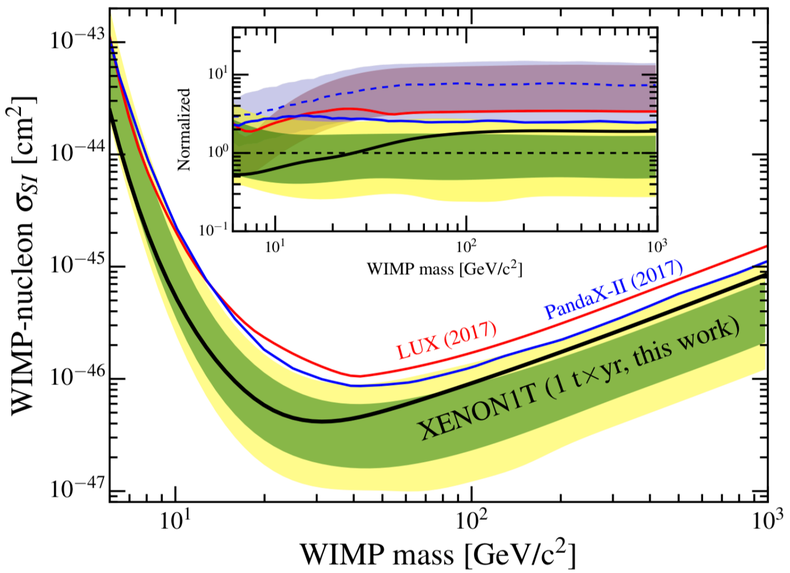
# Dark matter (DM) status



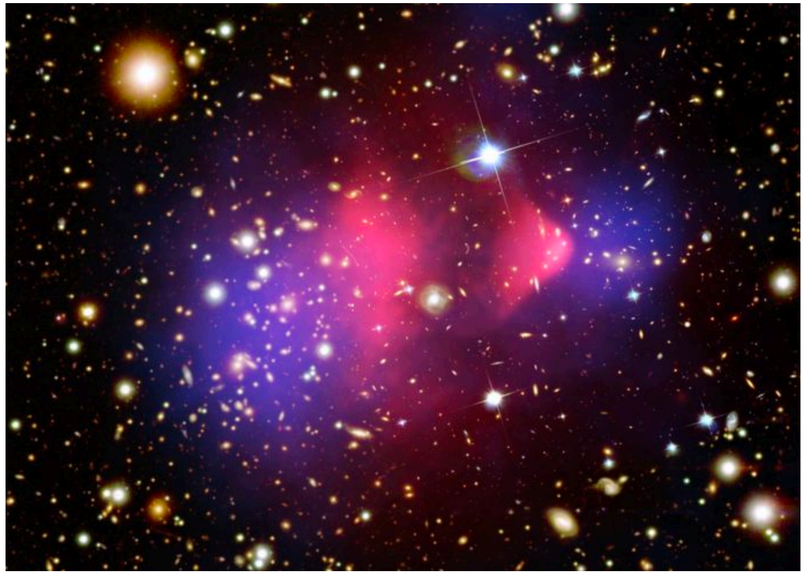

[XENON 1805.12562]

Non-gravitational interactions:
no evidence so far

# Dark matter (DM) status




[XENON 1805.12562]

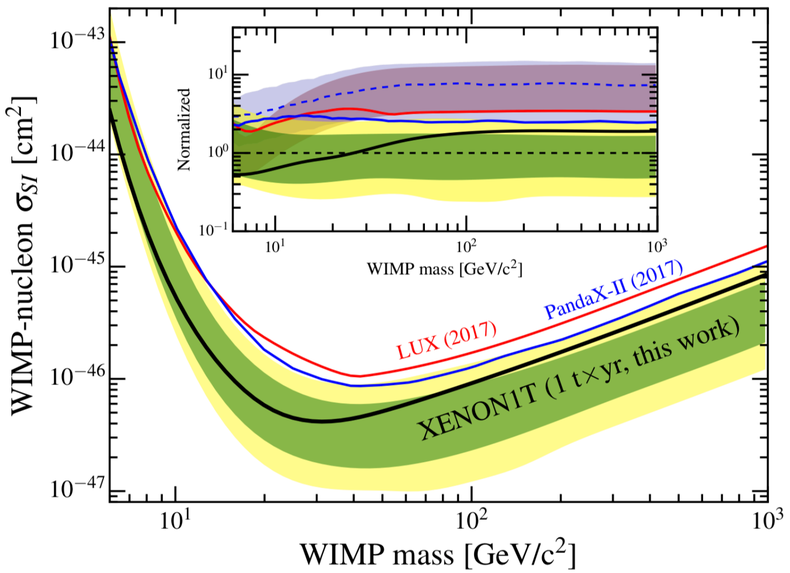


[NASA]

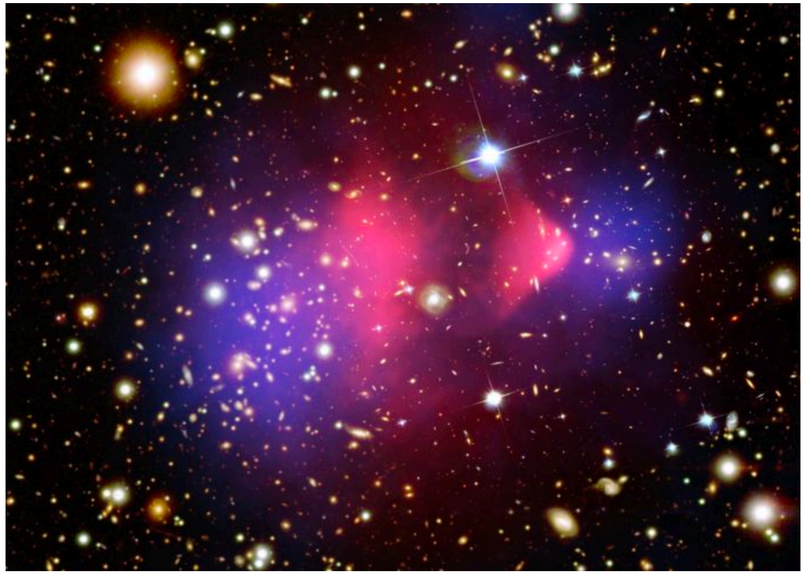
Non-gravitational interactions: no evidence so far

Gravitational interactions at large length scales: plenty of evidence, consistent with ΛCDM

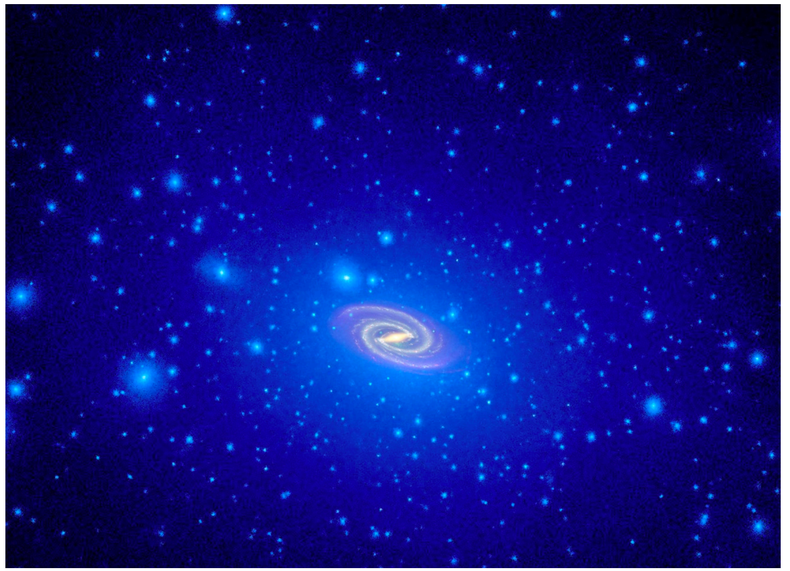
# Dark matter (DM) status




[XENON 1805.12562]



[NASA]



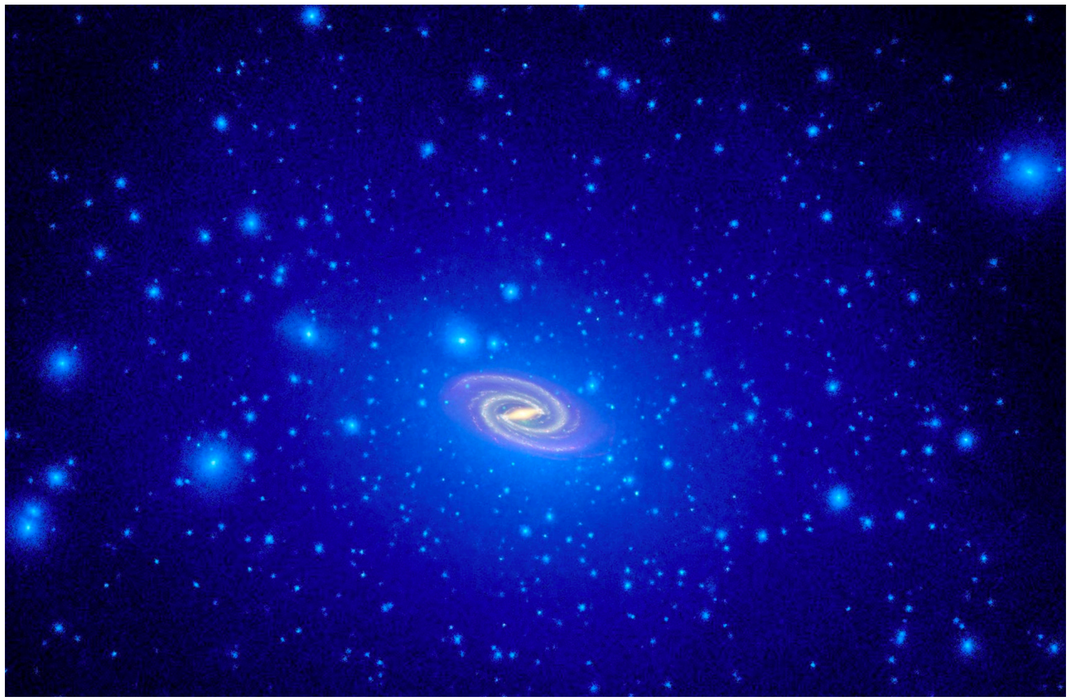
[T. Brown, J.Tumlinson]

Non-gravitational interactions: no evidence so far

Gravitational interactions at large length scales: plenty of evidence, consistent with ΛCDM

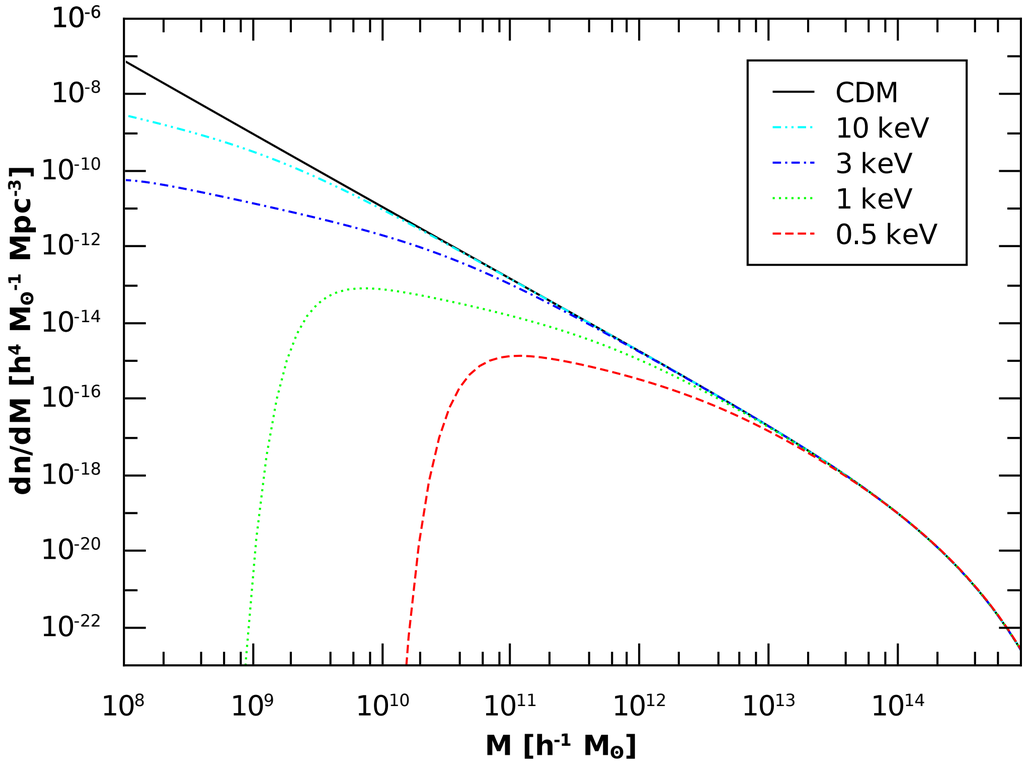
Gravitational interactions at small scales (DM substructure): beginning to be probed, many models predict deviations from ΛCDM

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



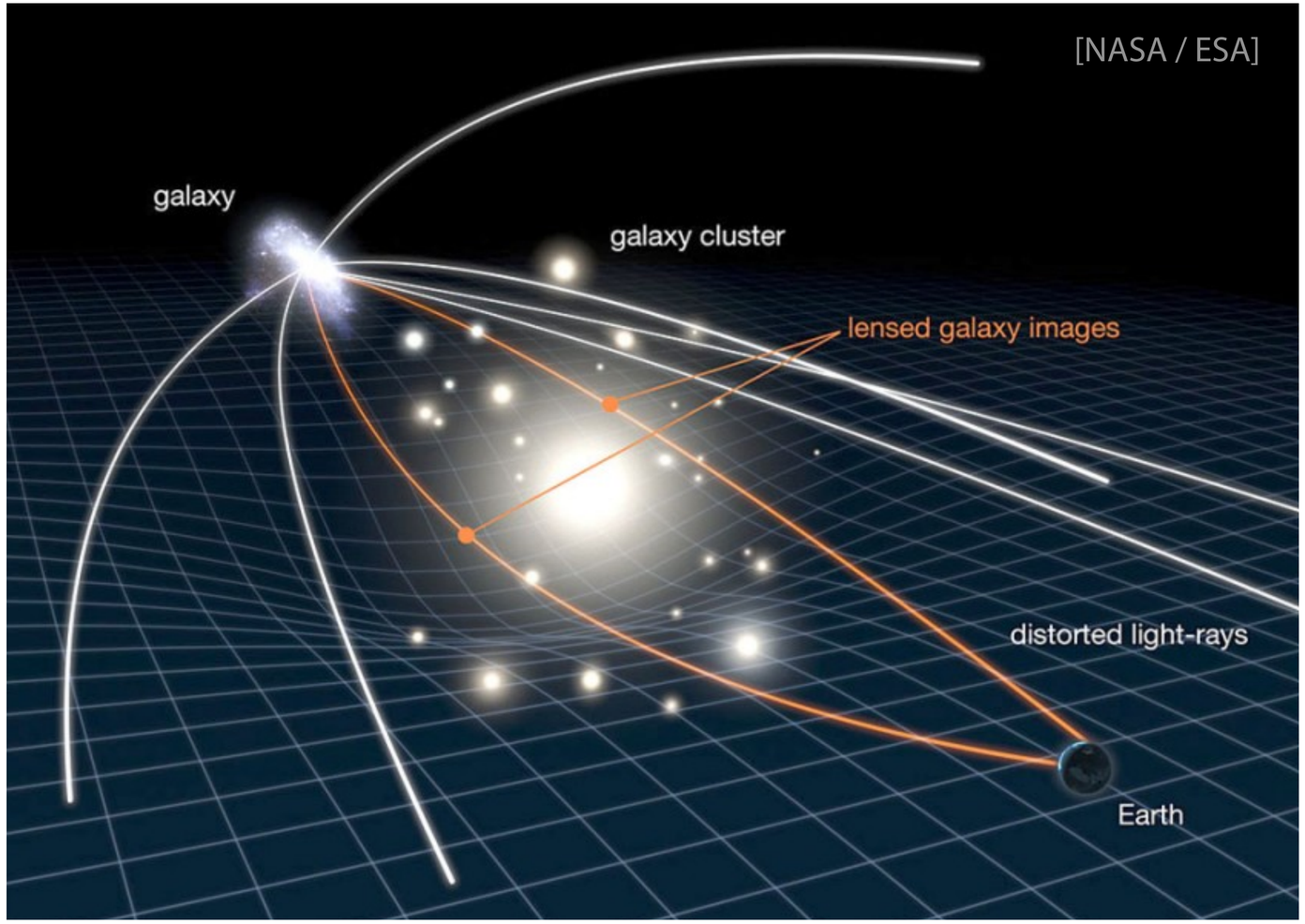
[T. Brown, J.Tumlinson]

Abundance of DM subhalos vs mass:



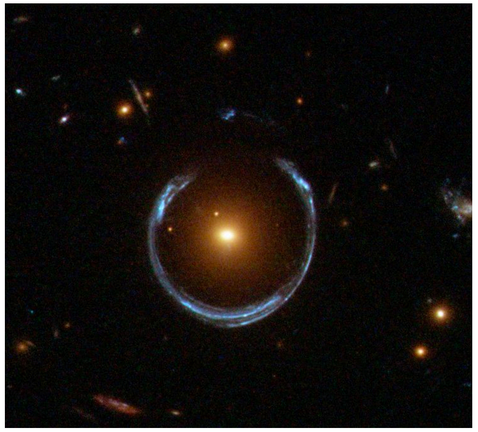

[R. *Dunstan et al* 1109.6291]

# Strong gravitational lensing



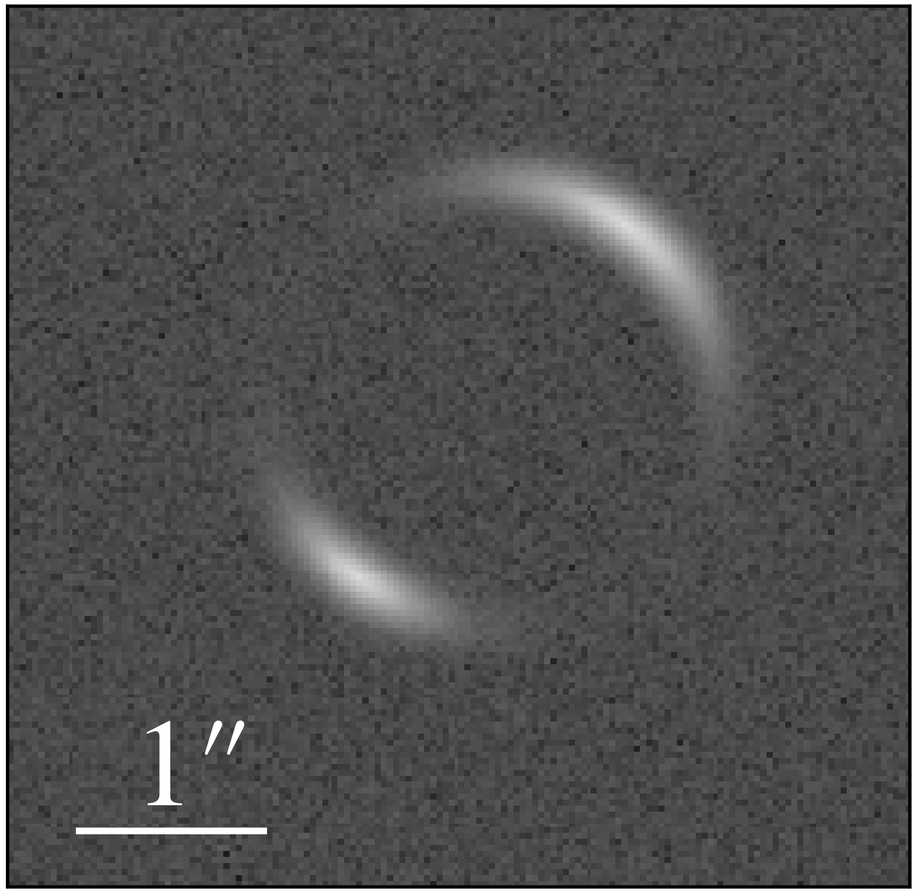

Multiple images of quasars

Extended arcs from galaxies



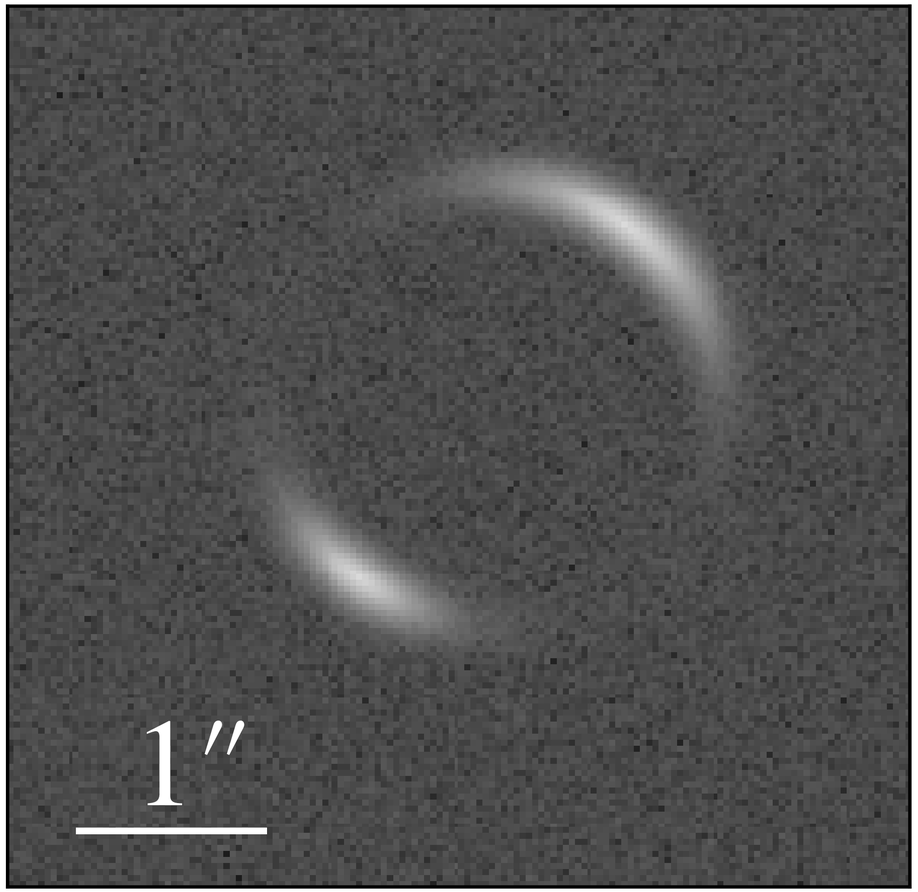
# Subhalos affect strong lensing

Smooth halo only

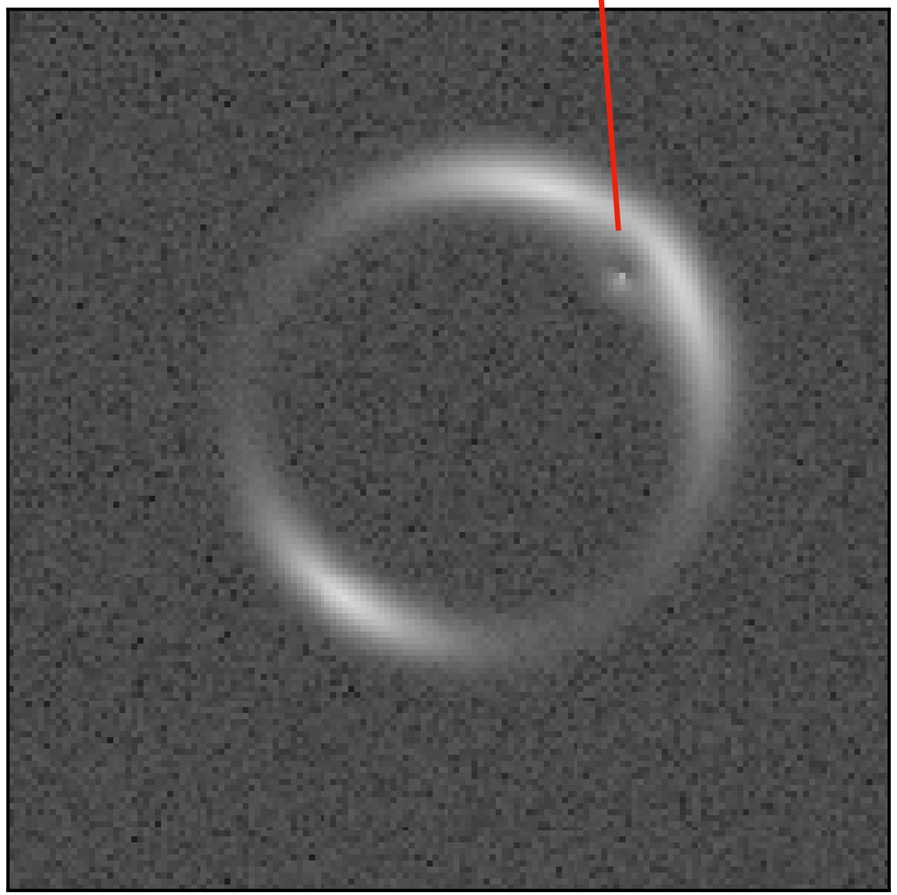


# Subhalos affect strong lensing

Smooth halo only

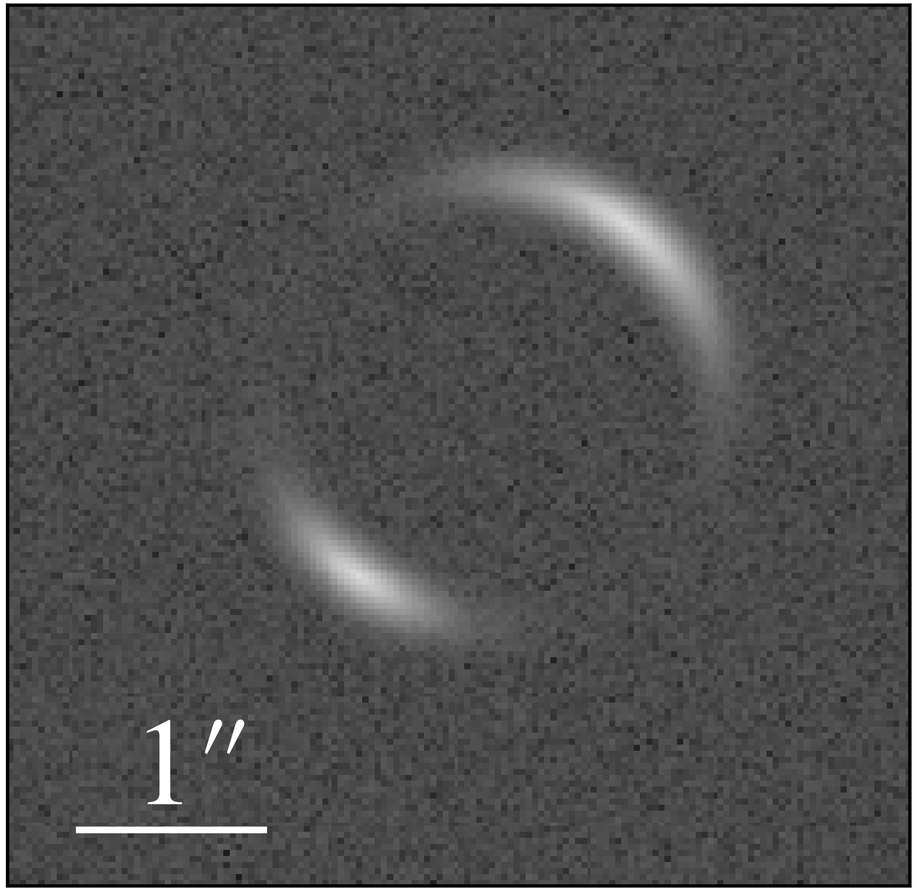



Smooth halo + subhalo

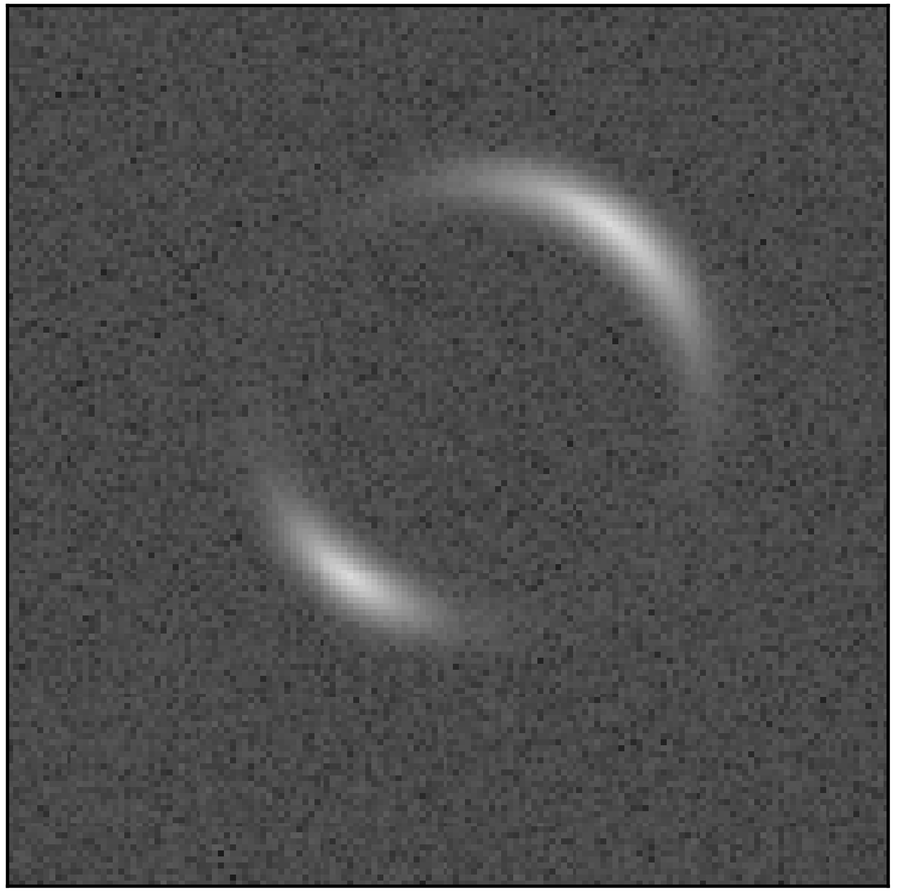


# Subhalos affect strong lensing… realistically

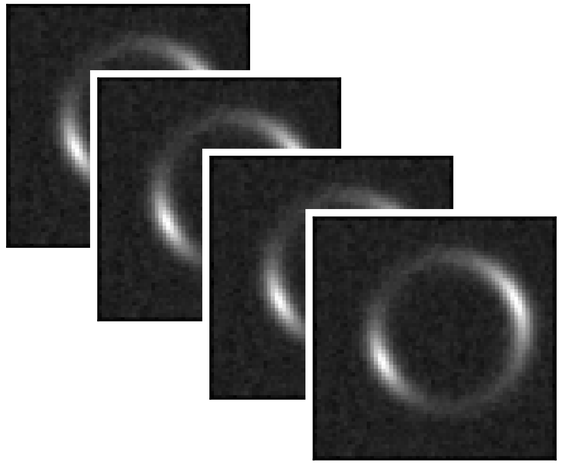
Smooth halo only

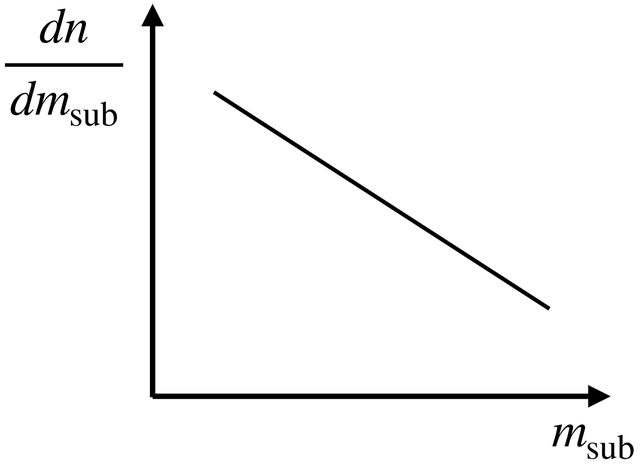



Smooth halo + subhalos



# Scalable inference for small subhalos

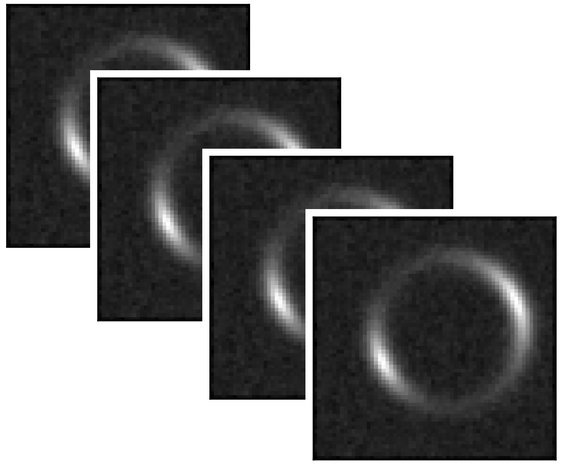


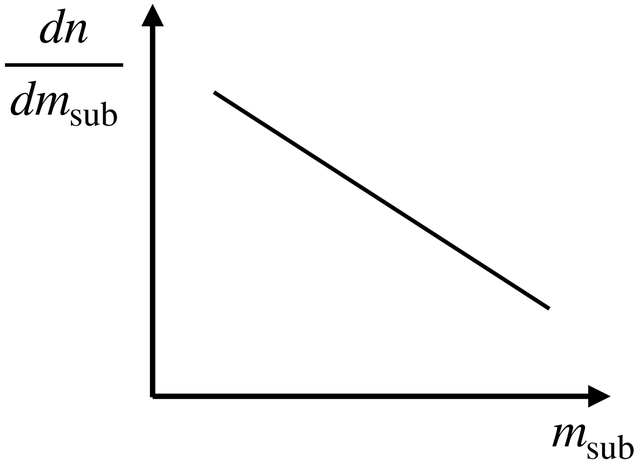



Future surveys (LSST, Euclid) are expected to deliver large samples of galaxy-galaxy strong lenses [Collett et al 1507.02657]

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]

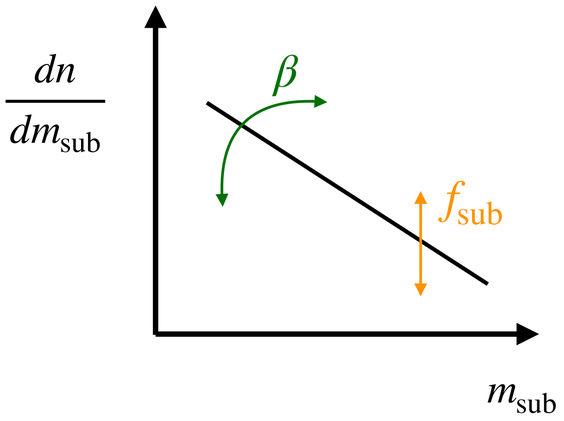
Goal: infer subhalo mass distribution through collective effects of many light subhalos

⇒ 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)

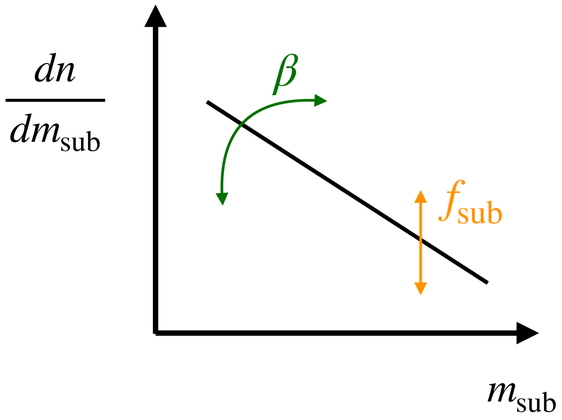
# Simulation-based inference for strong lensing

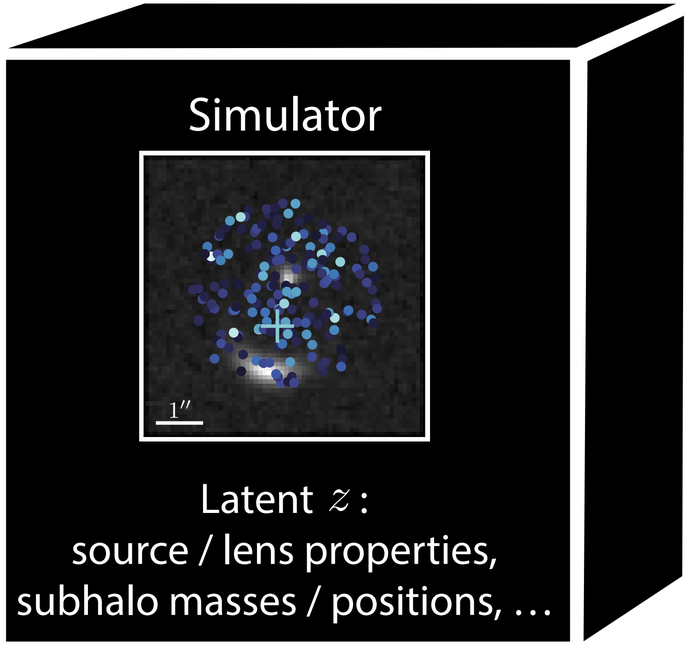
2 parameters $\theta = (\beta, f_{\rm sub})$



# Simulation-based inference for strong lensing

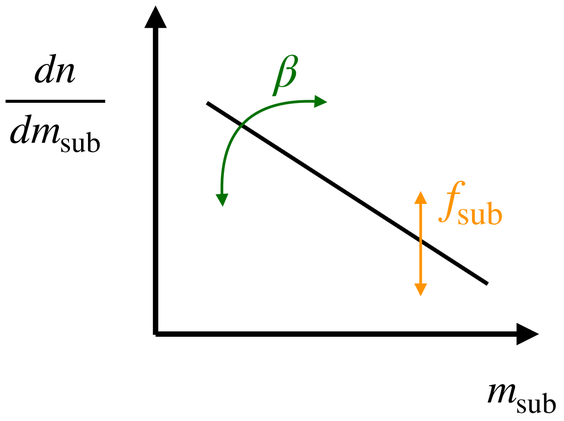
2 parameters $\theta = (\beta, f_{\rm sub})$

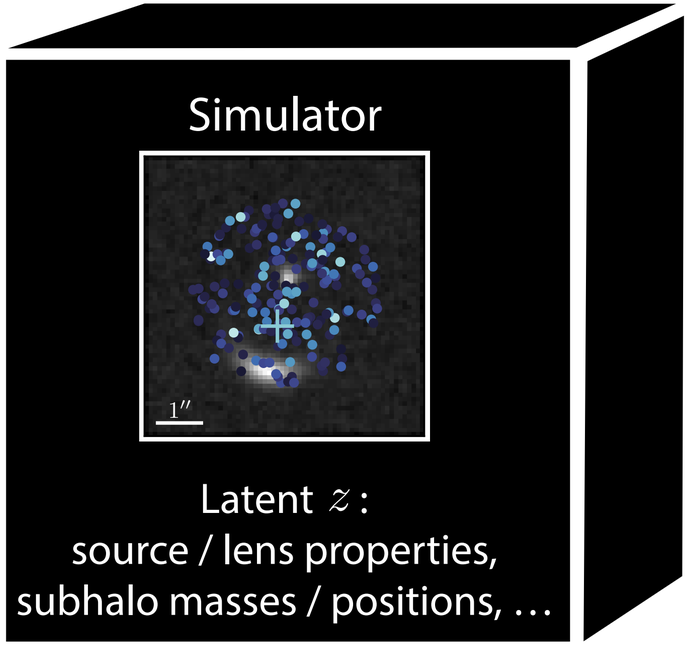


# Simulation-based inference for strong lensing

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

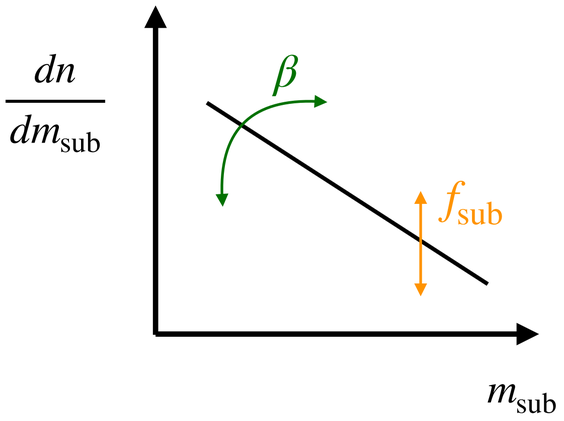


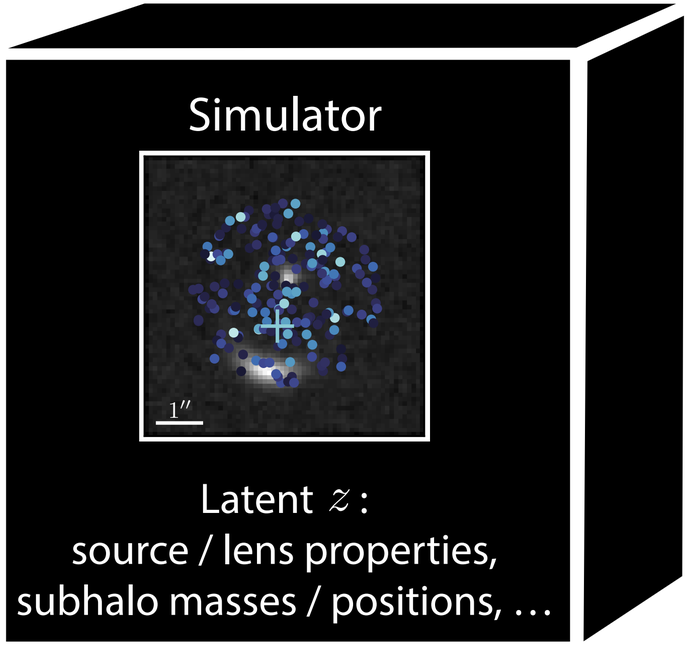




$64^2$ observables $x$

# Simulation-based inference for strong lensing

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



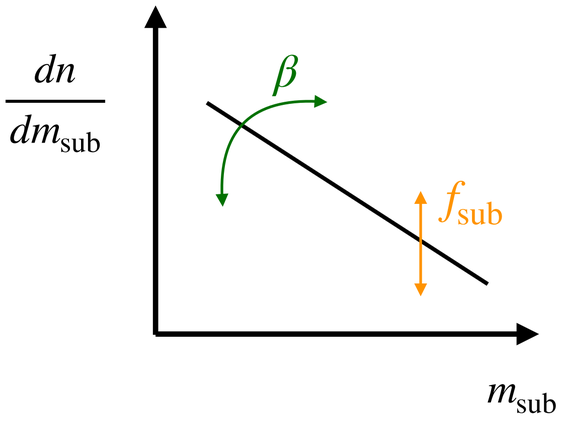



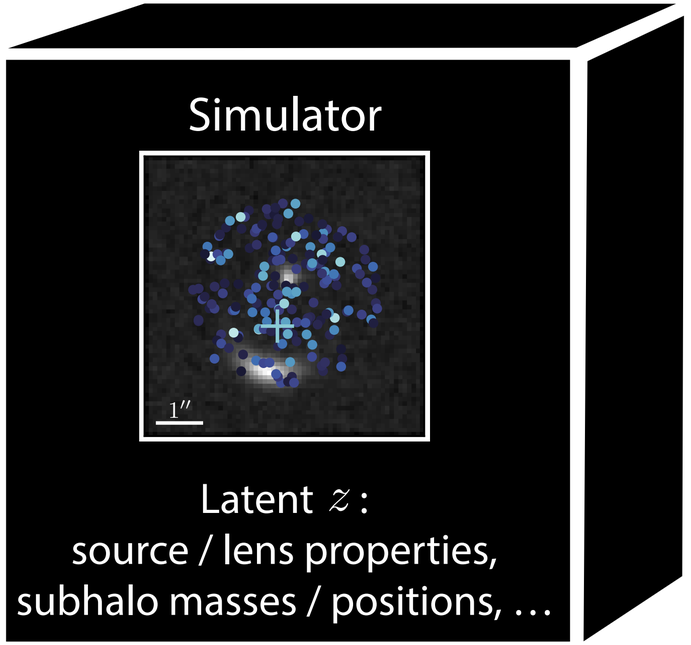

$64^2$ observables $x$

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

# Simulation-based inference for strong lensing

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



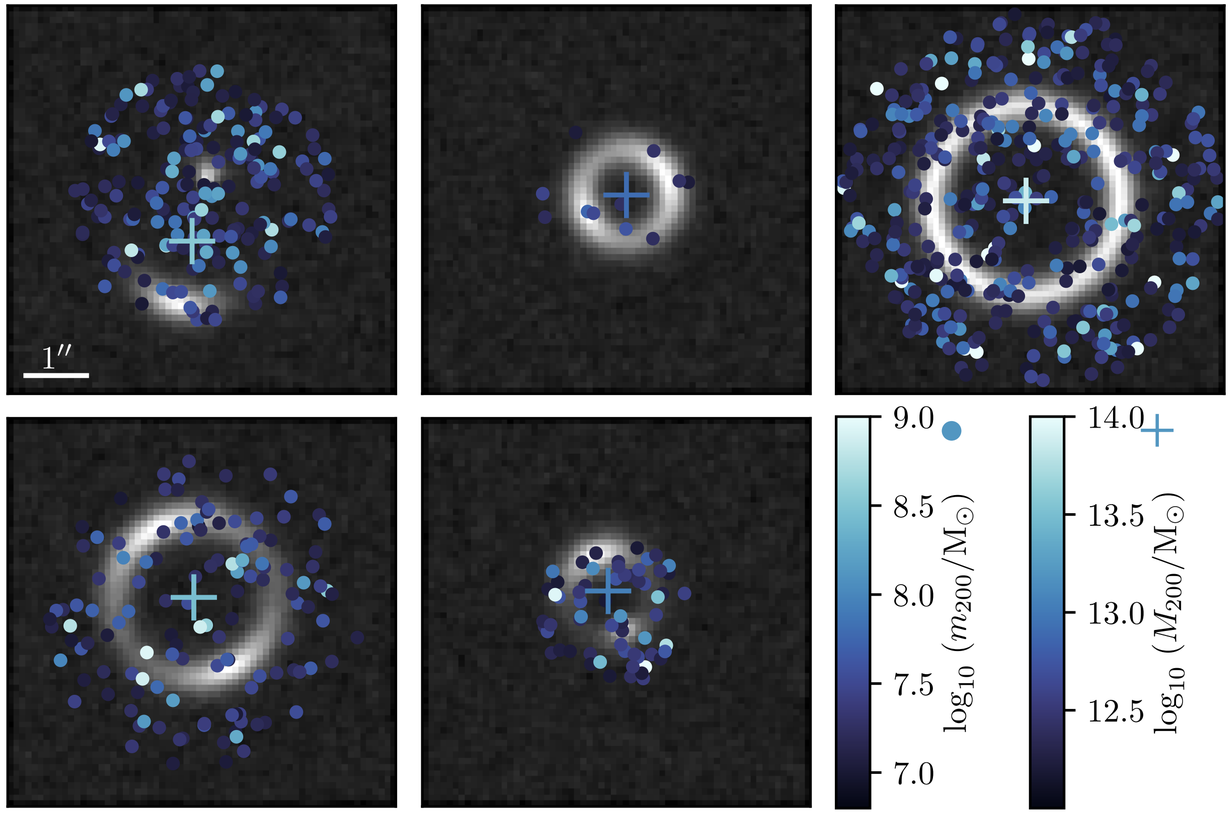

$64^2$ observables $x$

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]

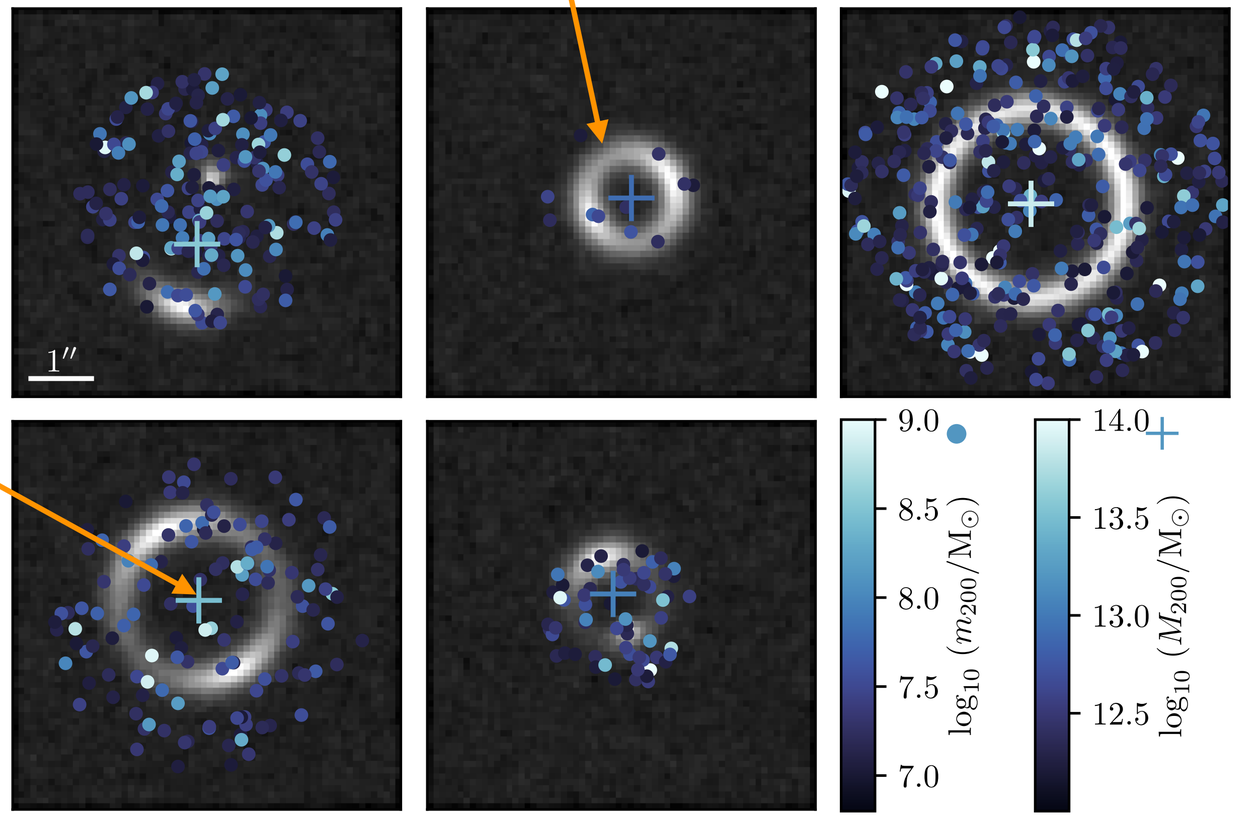


# Proof-of-principle simulator

[following T. Collett 1507.02657]

Spherical lensing host galaxies
at redshift ~0.5…1

Extended galaxy sources
at redshift 1.5

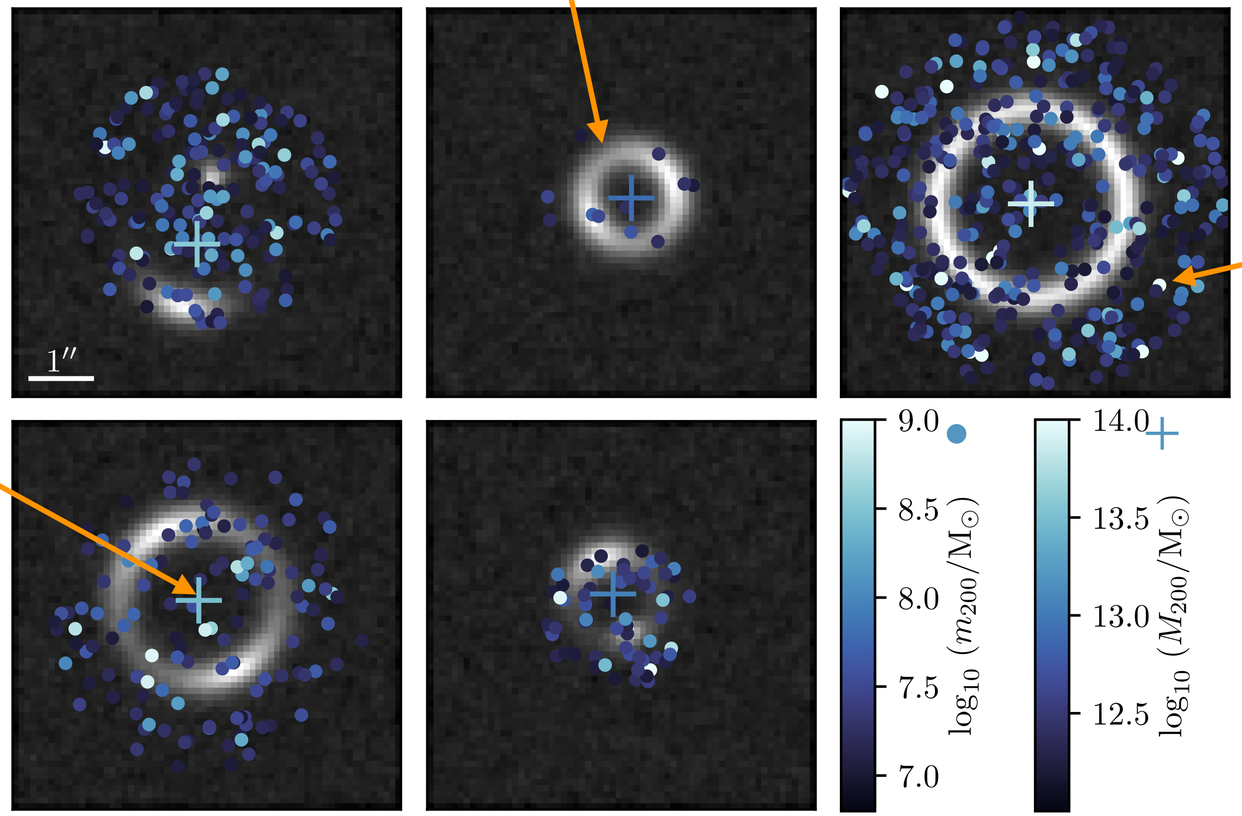


# Proof-of-principle simulator

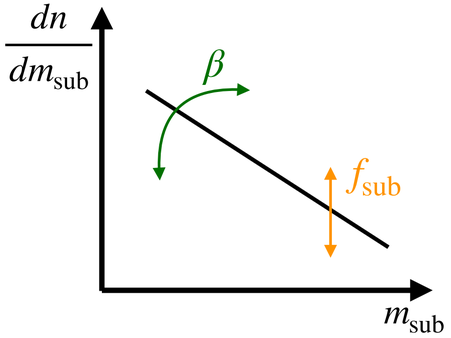
[following T. Collett 1507.02657]

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

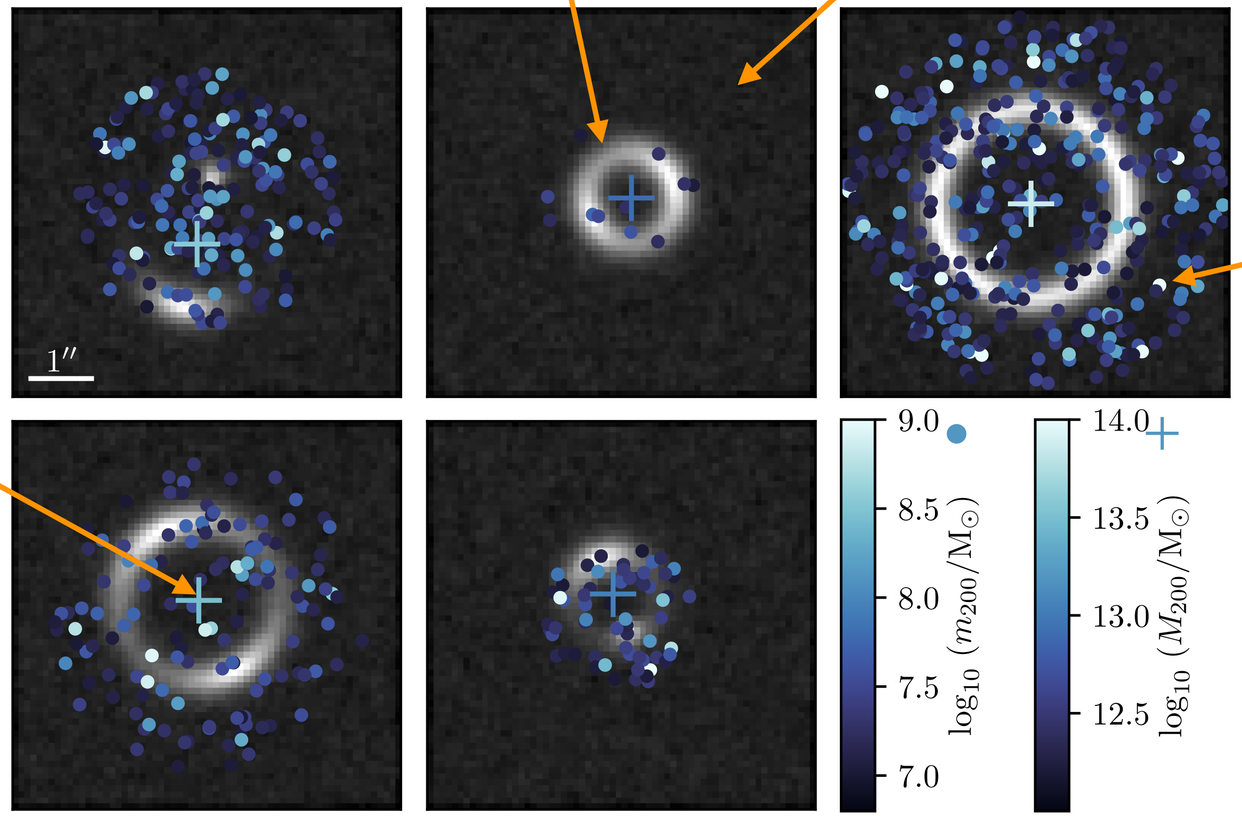
[following T. Collett 1507.02657]

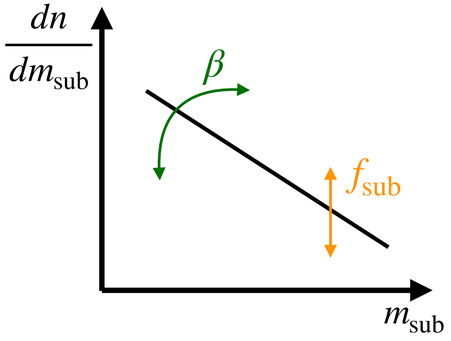
Spherical lensing host galaxies
at redshift ~0.5…1

Poisson fluctuations

Extended galaxy sources
at redshift 1.5

Subhalos follow mass distribution with two parameters



# Proof-of-principle simulator

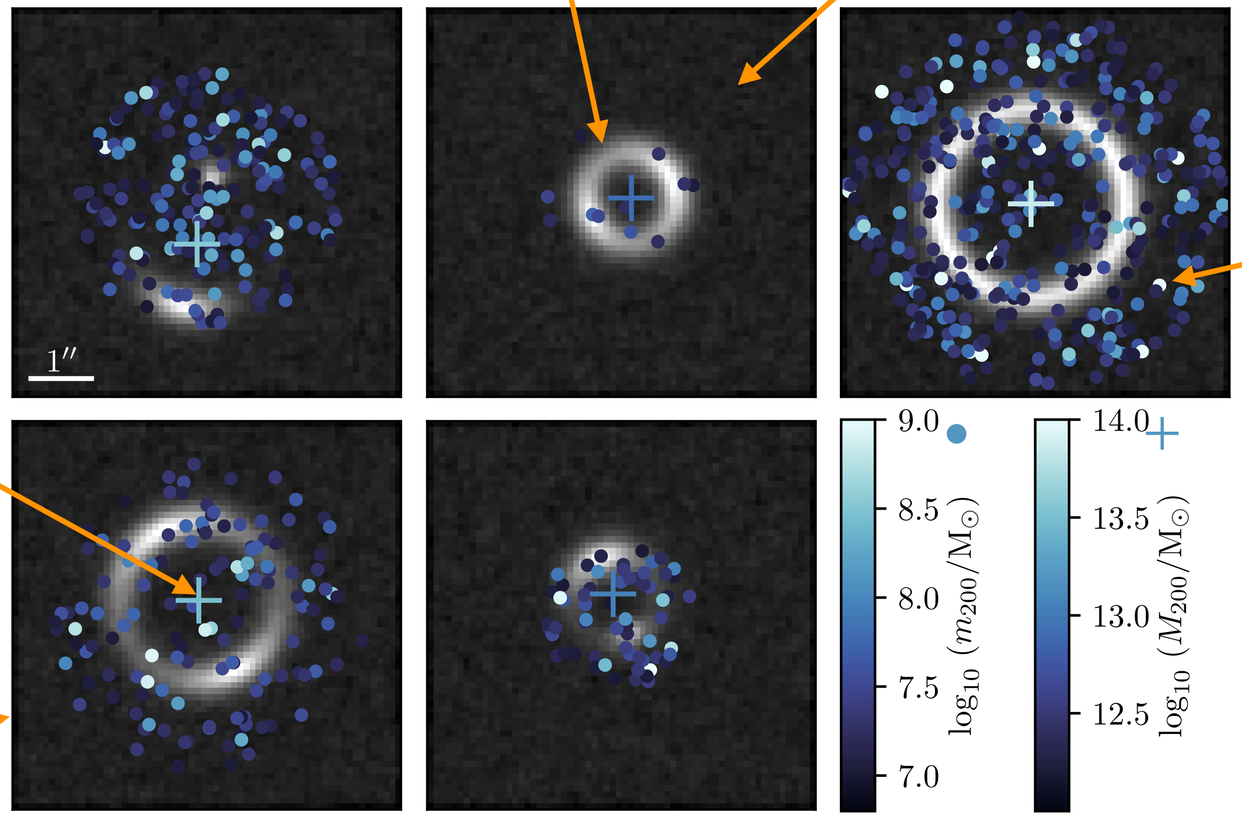
[following T. Collett 1507.02657]

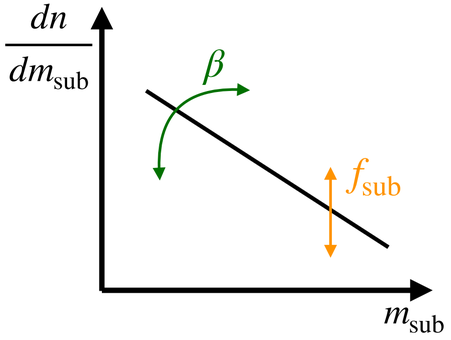
Spherical lensing host galaxies
at redshift ~0.5…1

Poisson fluctuations

Extended galaxy sources
at redshift 1.5

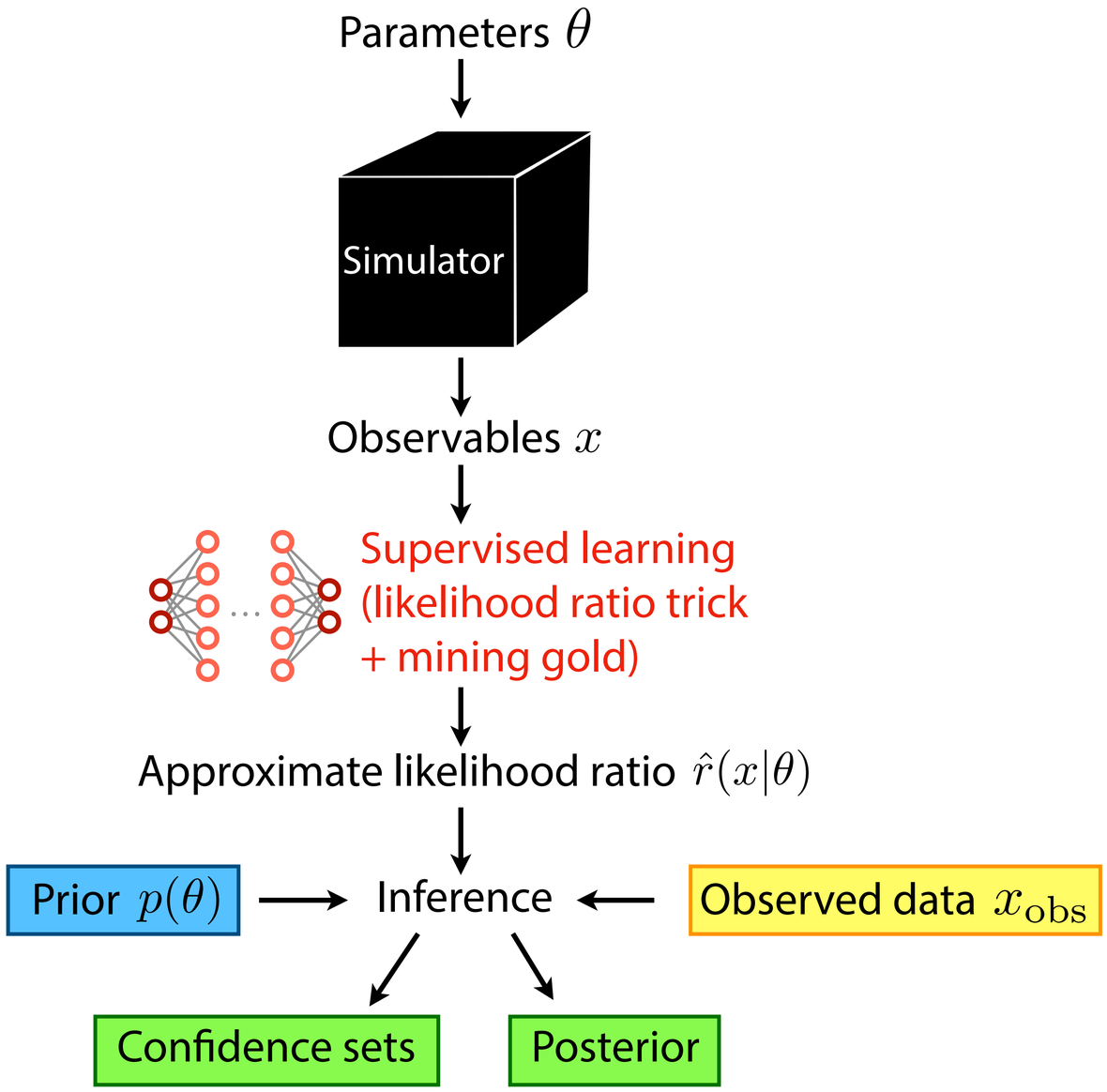
Subhalos follow mass distribution with two parameters



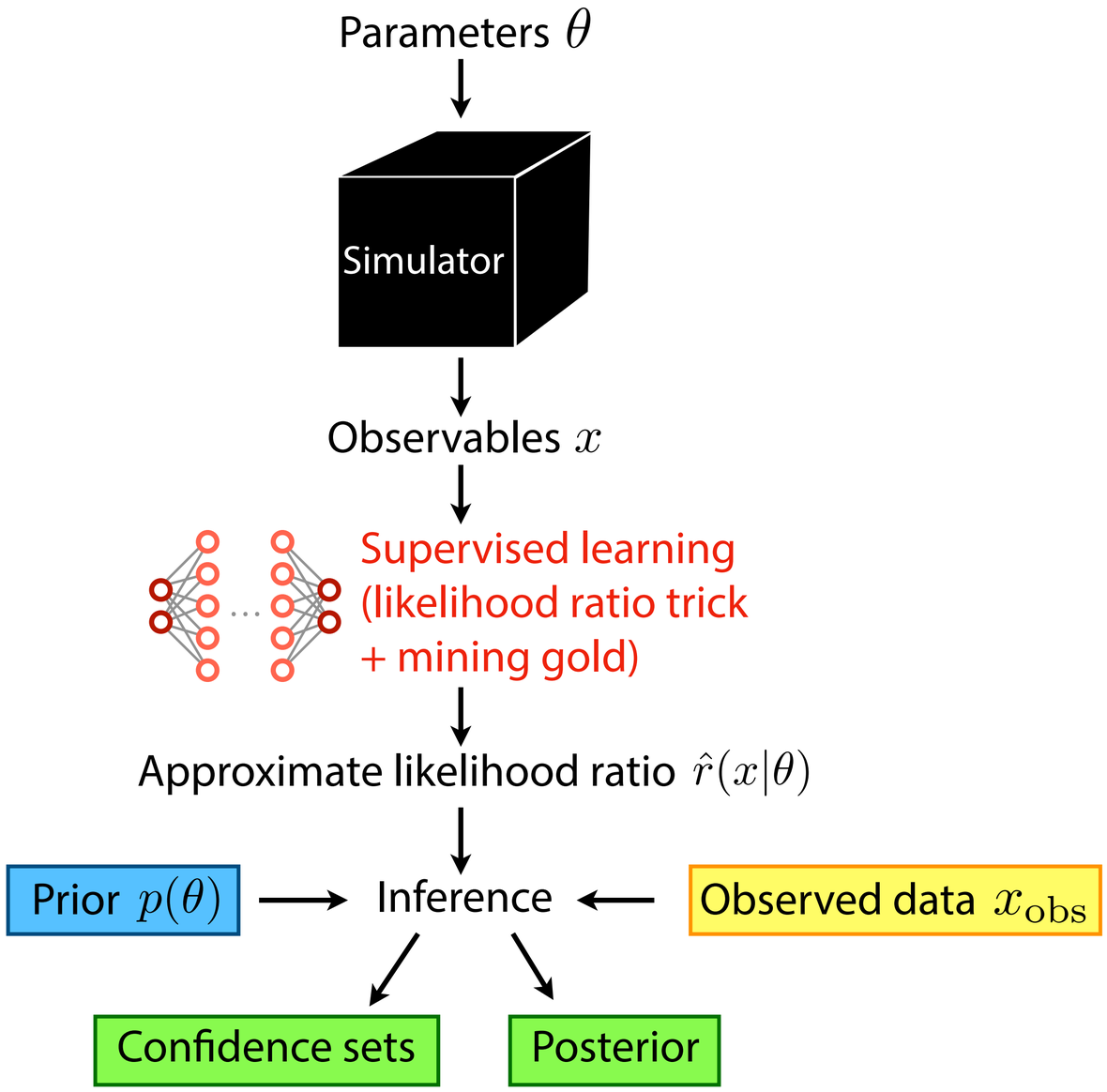




Euclid detector model
(single optical band,
$64^2$ pixel, Gaussian PSF)

# Inference setup

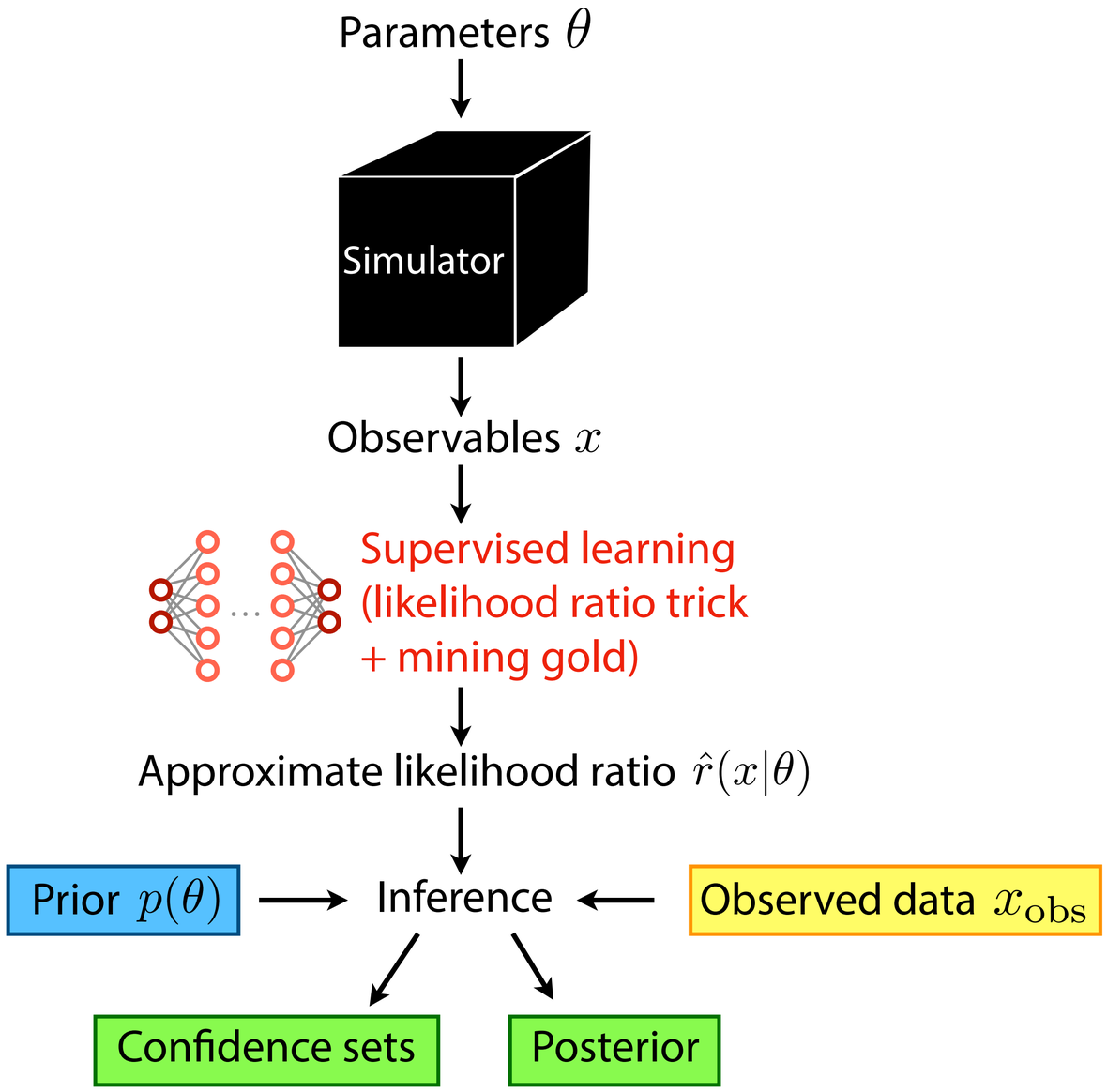


# Inference setup




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

# Inference setup



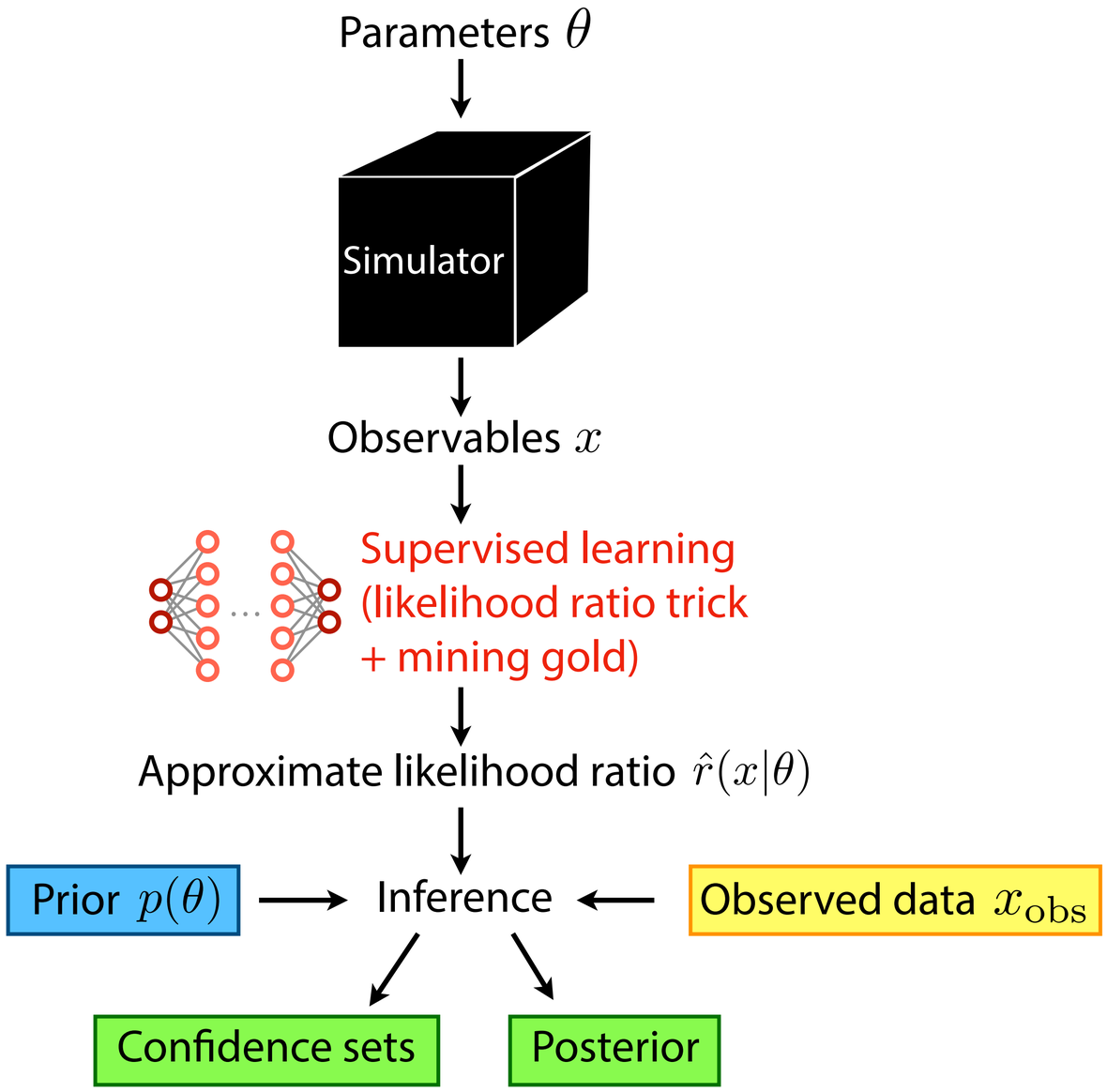

Training data: $10^6$ lensed images with
$0 \leq f_{\rm sub} \leq 0.2, -1.5 \leq \beta \leq -0.5$

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
$0 \leq f_{\rm sub} \leq 0.2, -1.5 \leq \beta \leq -0.5$

Convolutional neural network (modified ResNet-18) trained on ALICES loss

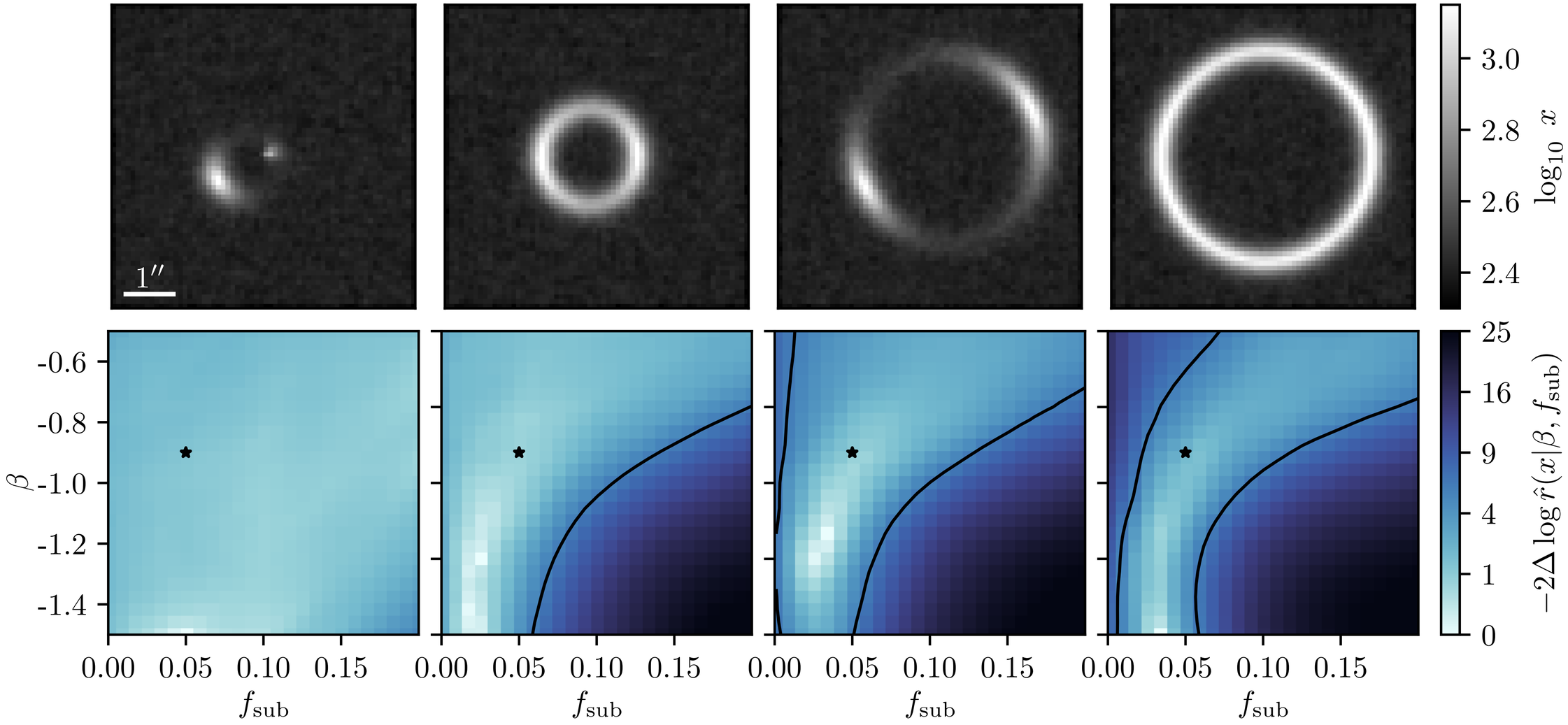
[M. Stoye, JB, J. Pavez, G, Louppe, K. Cranmer 1808.00973]

Probabilistic calibration of network output

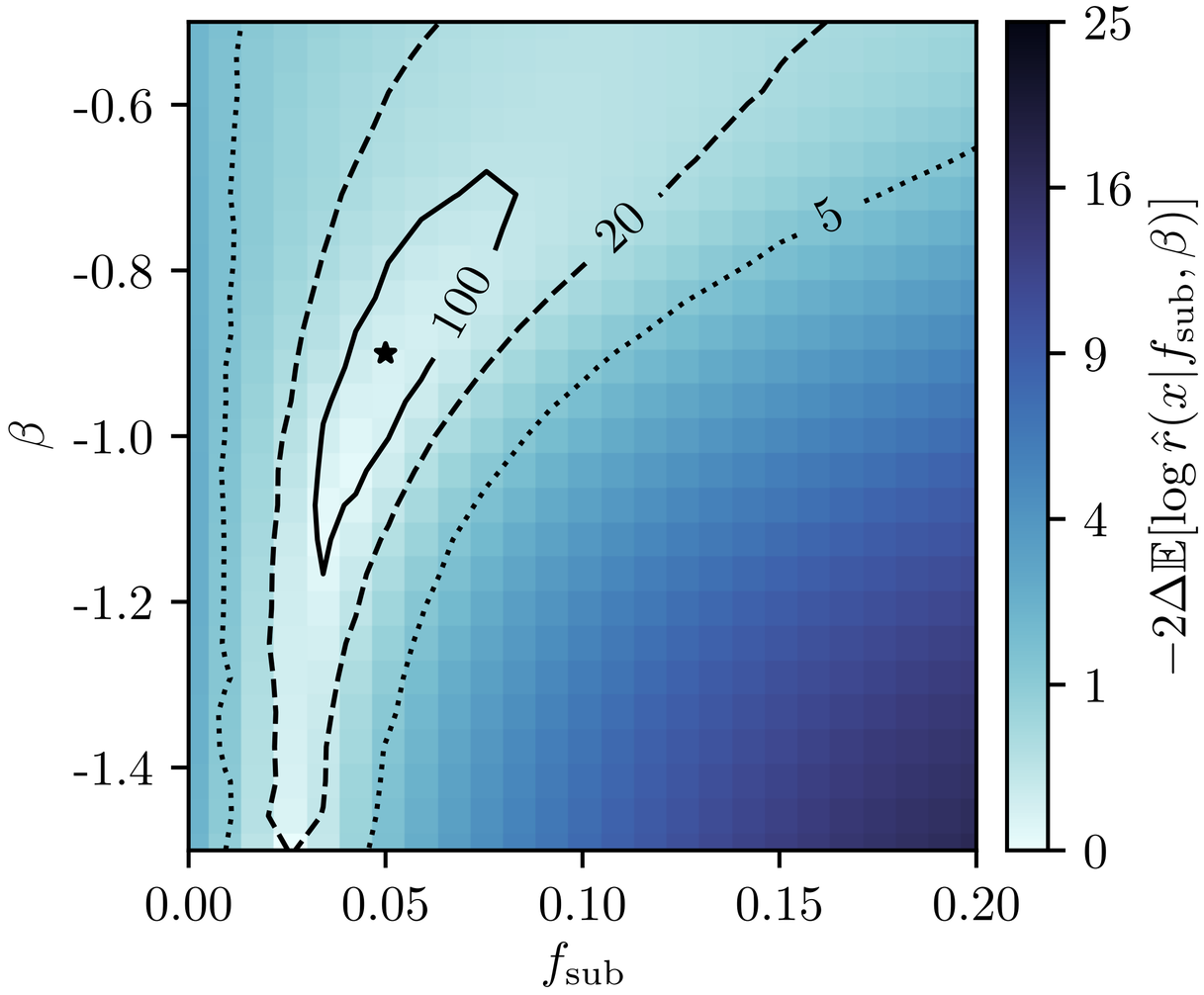
Synthetic "observed" data set simulated for
$f_{\rm sub} = 0.05, \beta = -0.9$

Bayesian & frequentist inference

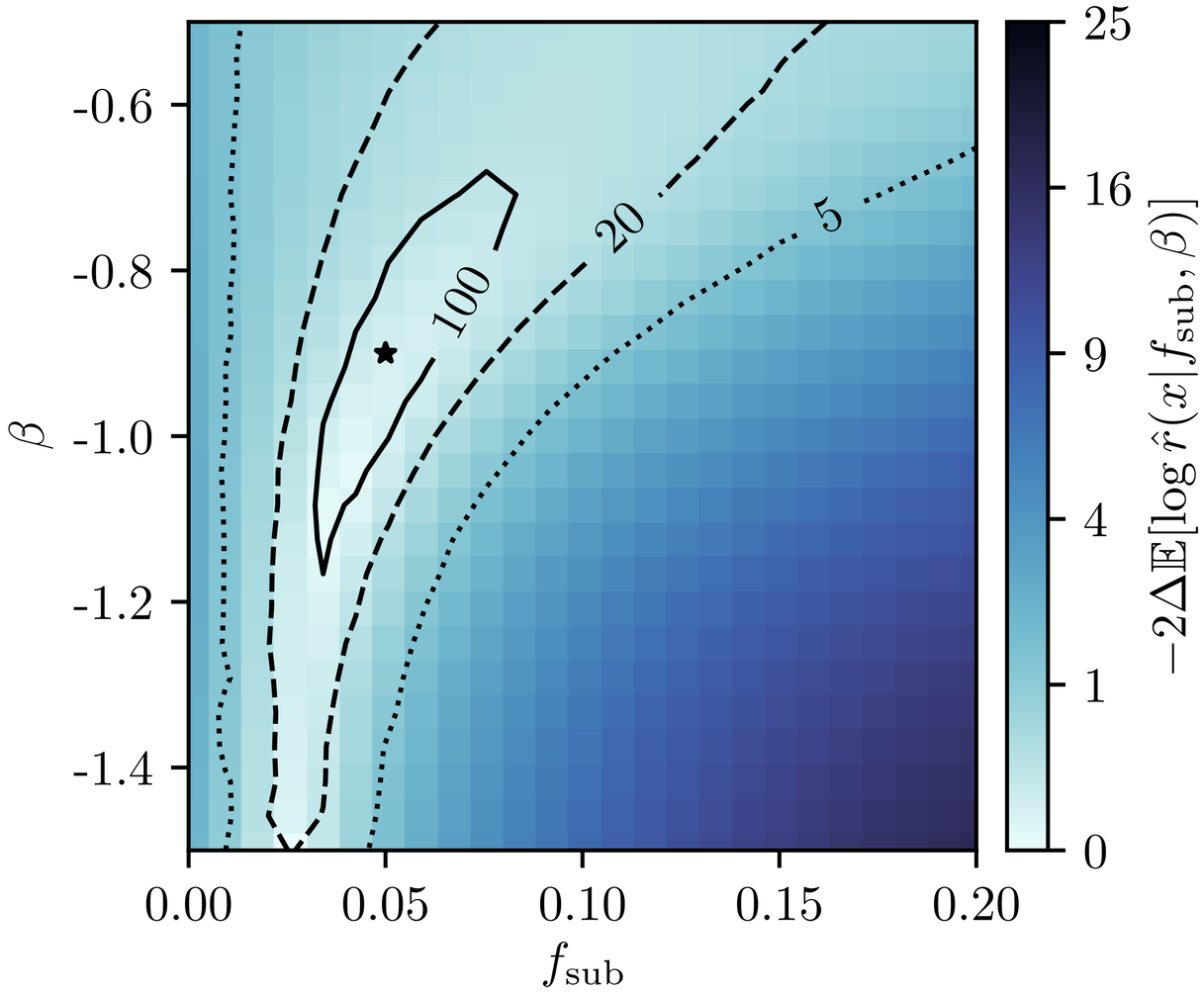
# Inferring parameters from individual images

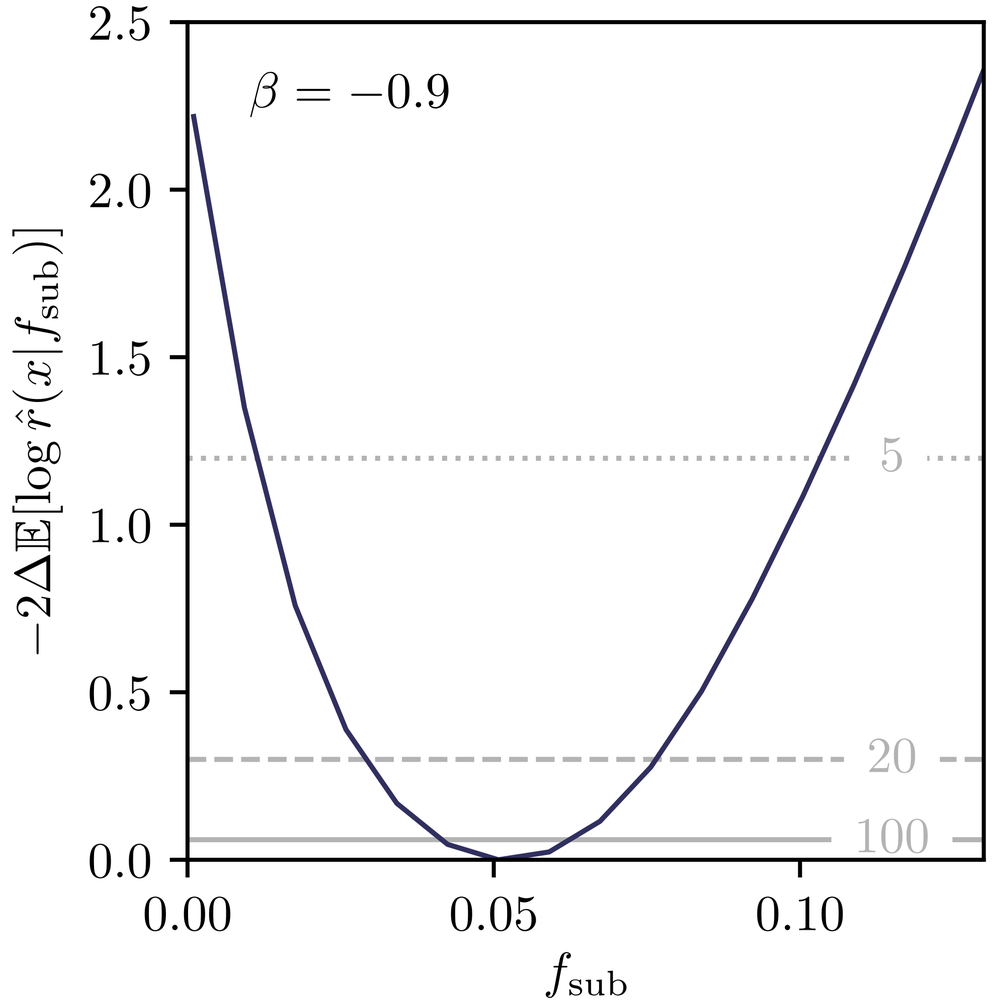


# Expected likelihood ratio map

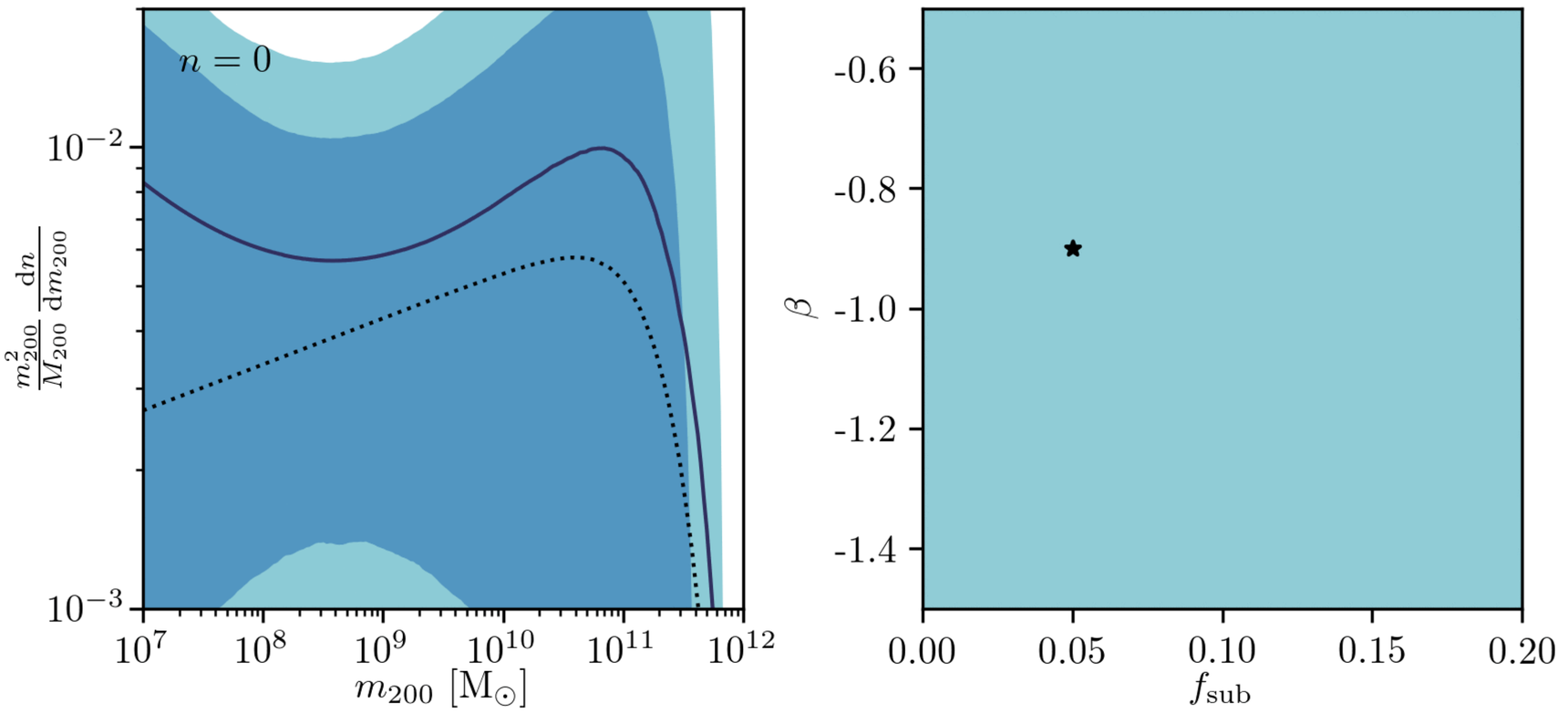


# Expected likelihood ratio map

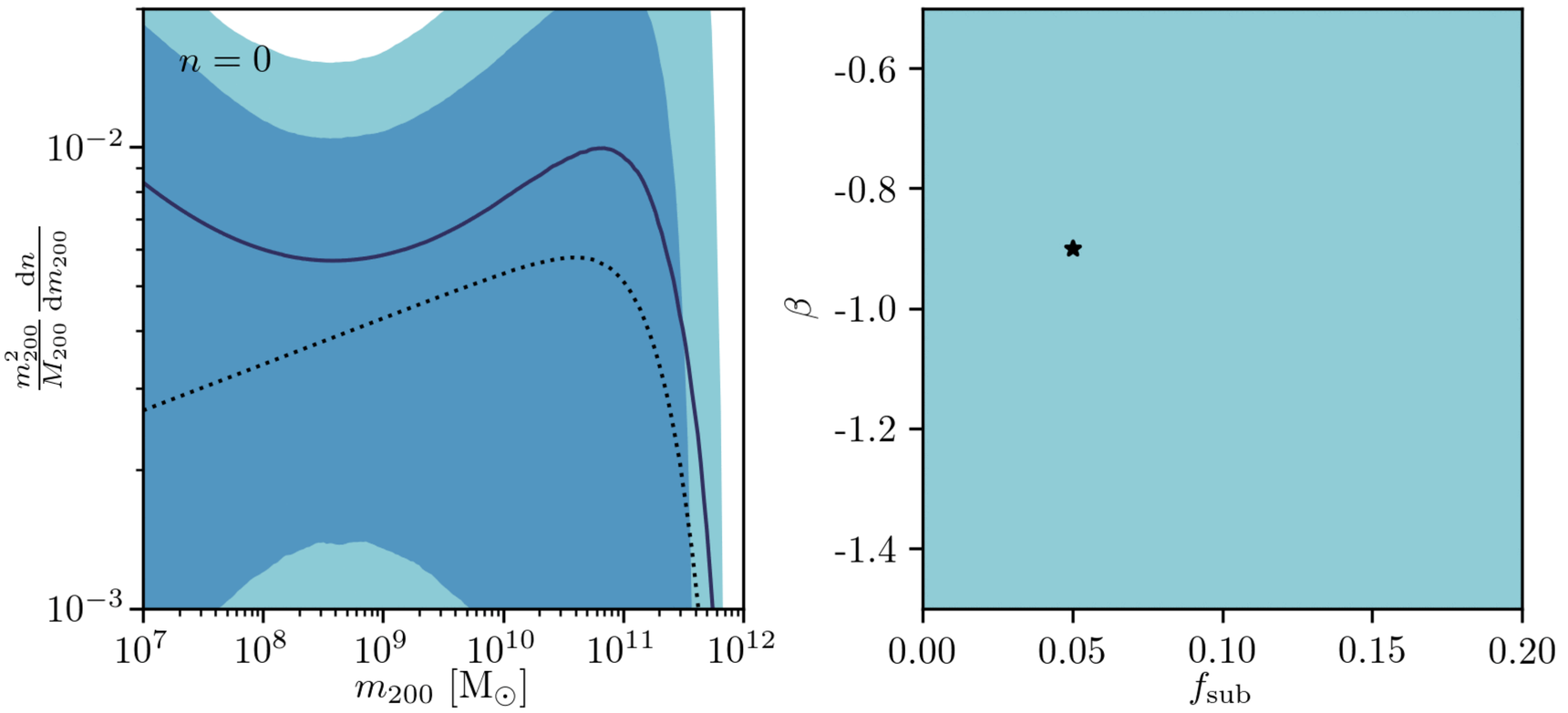


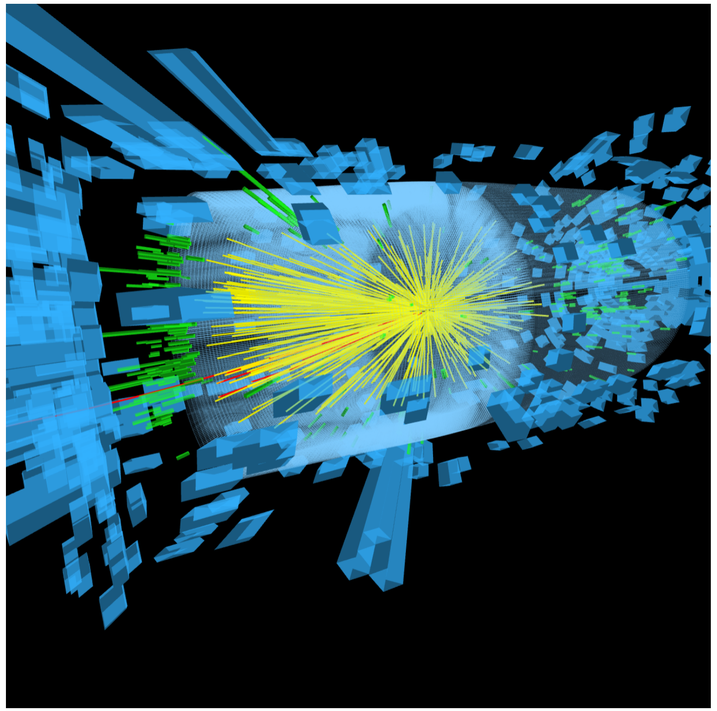



# Bayesian inference

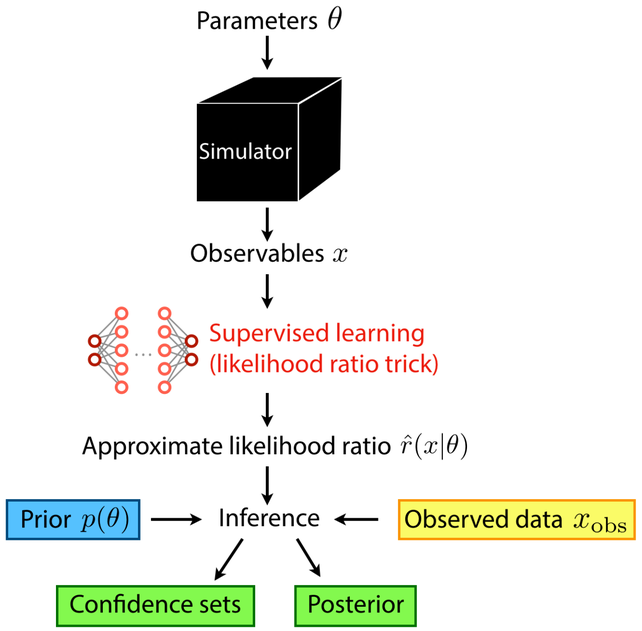


# Bayesian inference

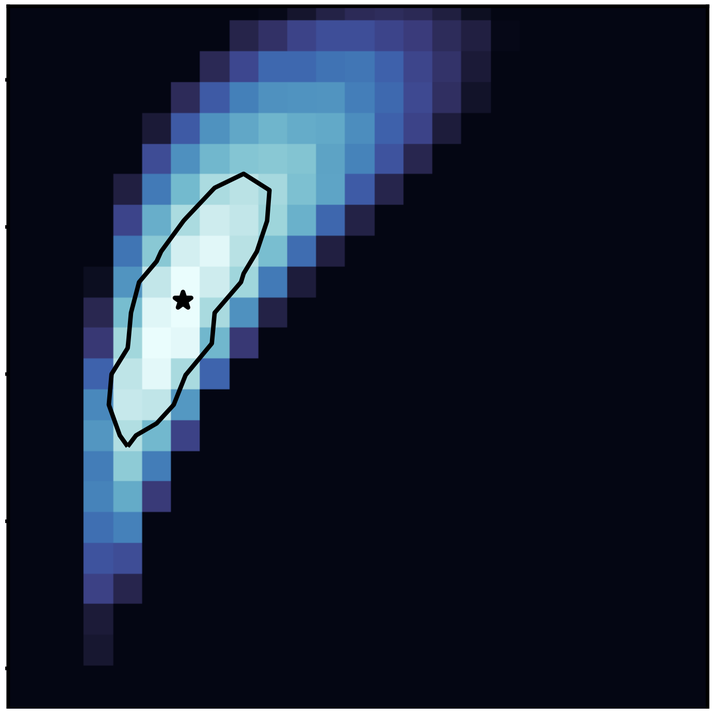




Inference is challenging when and phenomena are modeled with computer simulations and the data are high-dimensional



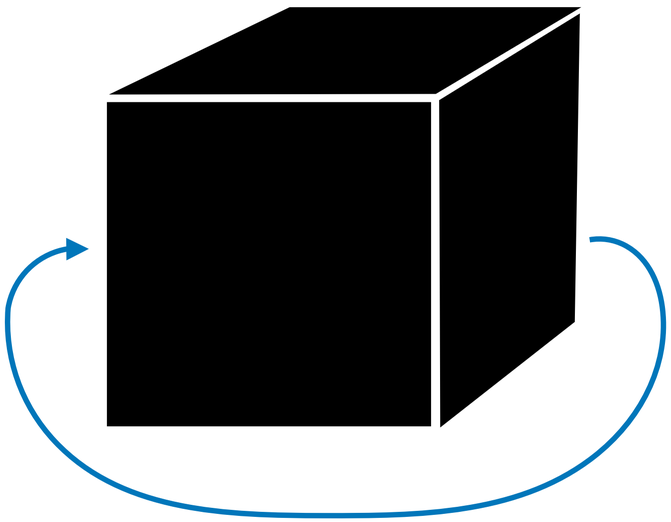

New inference algorithms are based on normalizing flows or the likelihood ratio trick;
we can improve them with physics understanding



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

# Frontiers of simulation-based inference

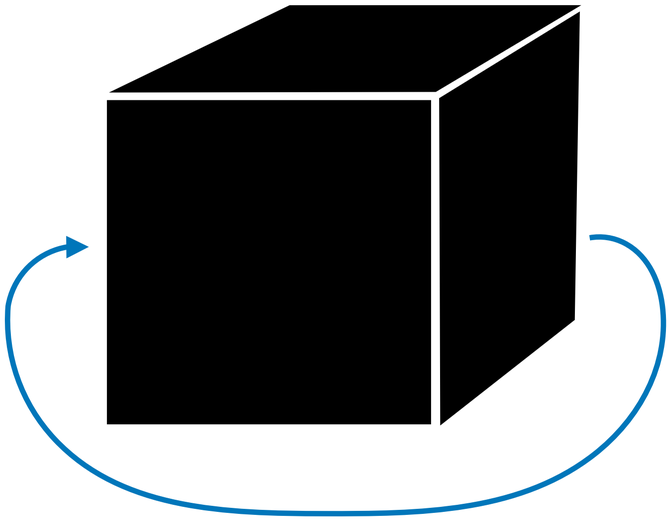
[K. Cranmer, JB, G. Louppe 1911.01429]



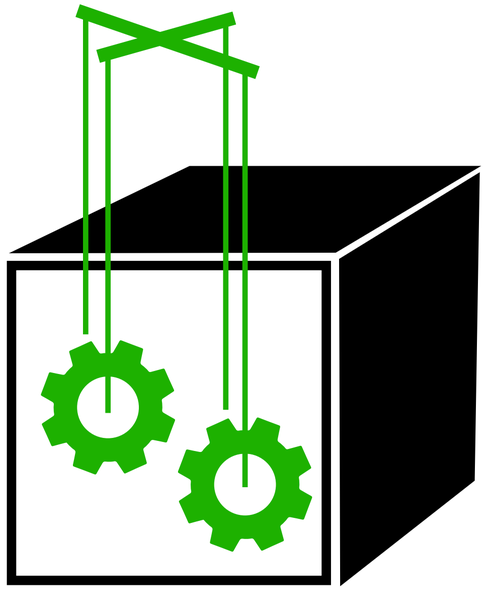
Active learning:
Iteratively guide simulator to important parameter regions based on past results

# Frontiers of simulation-based inference

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



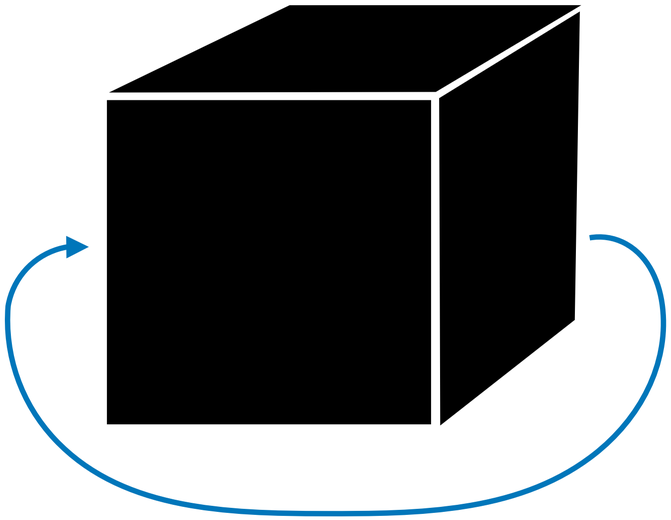
Active learning:
Iteratively guide simulator to important parameter regions based on past results



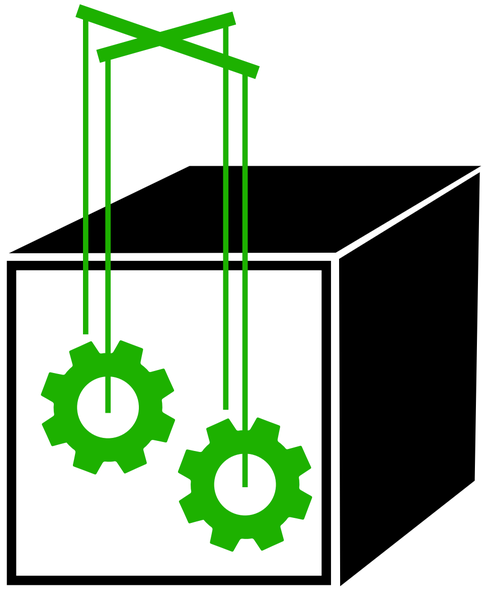
Probabilistic programming:
Explicitly write simulator as probabilistic program, with ability to condition execution trace on observations

# 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

# 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]

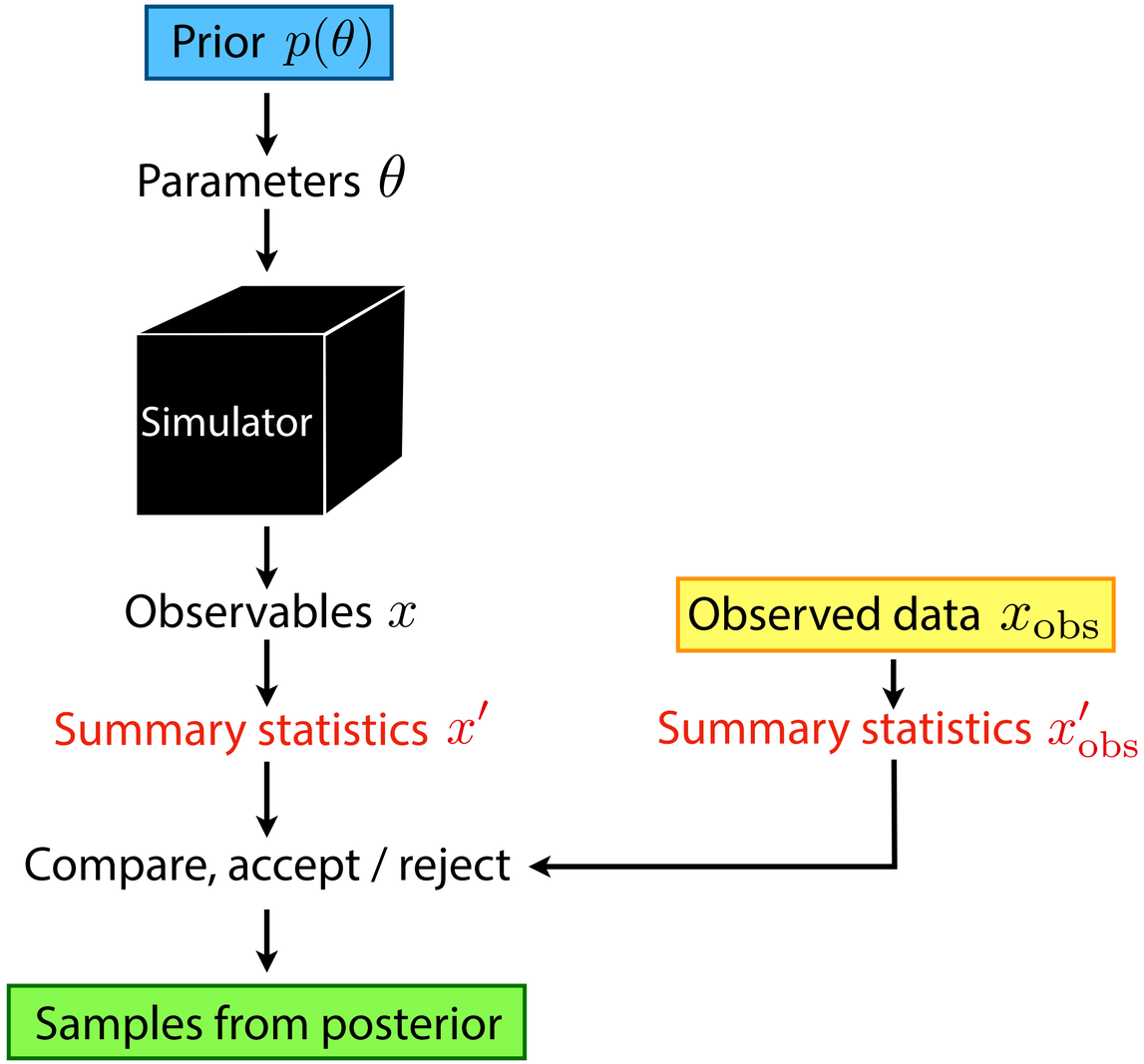
**Follow-up with incremental improvements**
M. Stoye, JB, K. Cranmer, G. Louppe, J. Pavez:
“Likelihood-free inference with an improved cross-entropy estimator”
[NeurIPS workshop, 1808.00973]

“Hooray! Question mark?”

— Todd Chavez

# Approximate Bayesian Computation (ABC)

[D. Rubin 1984]

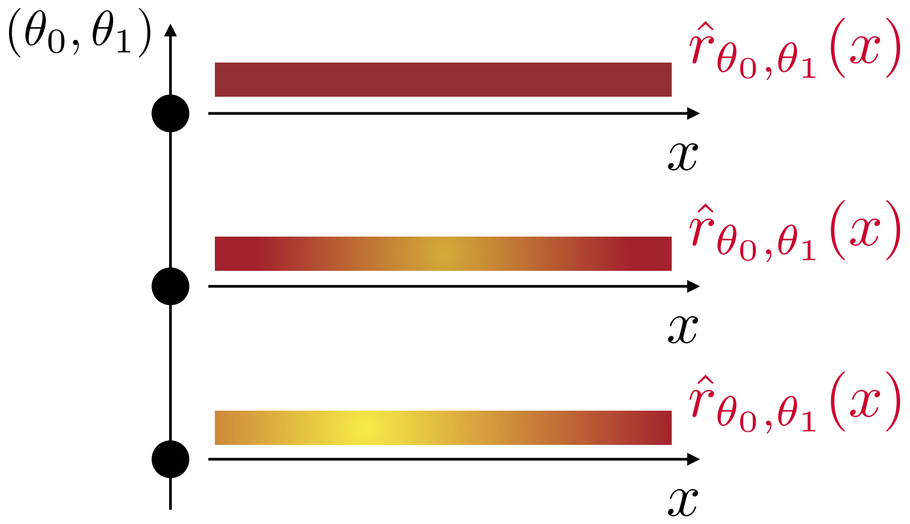



- Compression to summary statistics and acceptance threshold reduce quality of inference
- Rejection algorithm can be very sample inefficient

# Two types of likelihood ratio estimators

A) Point by point:

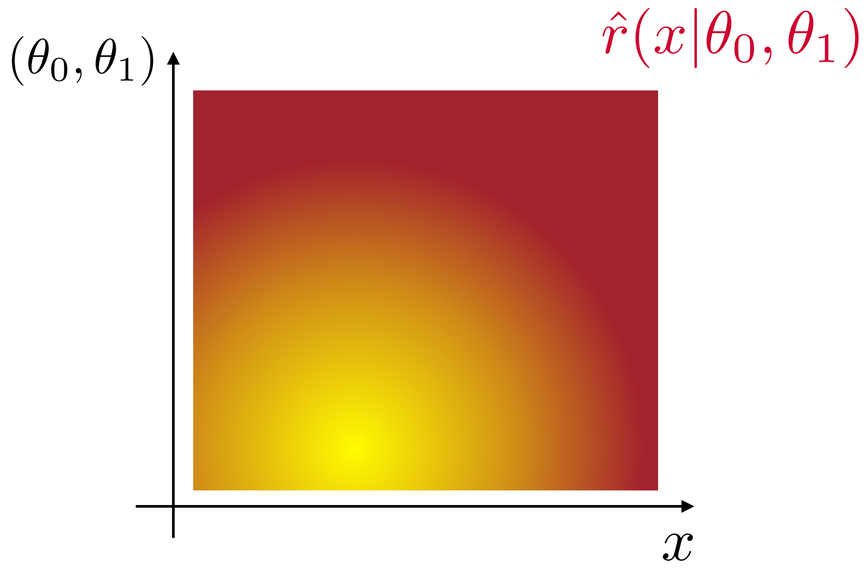
- first, define grid of parameter points $\{(\theta_0, \theta_1)\}$
- for each combination $(\theta_0, \theta_1)$ ,
  create separate estimator $\hat{r}_{\theta_0,\theta_1}(x)$
- final results can be interpolated between grid points



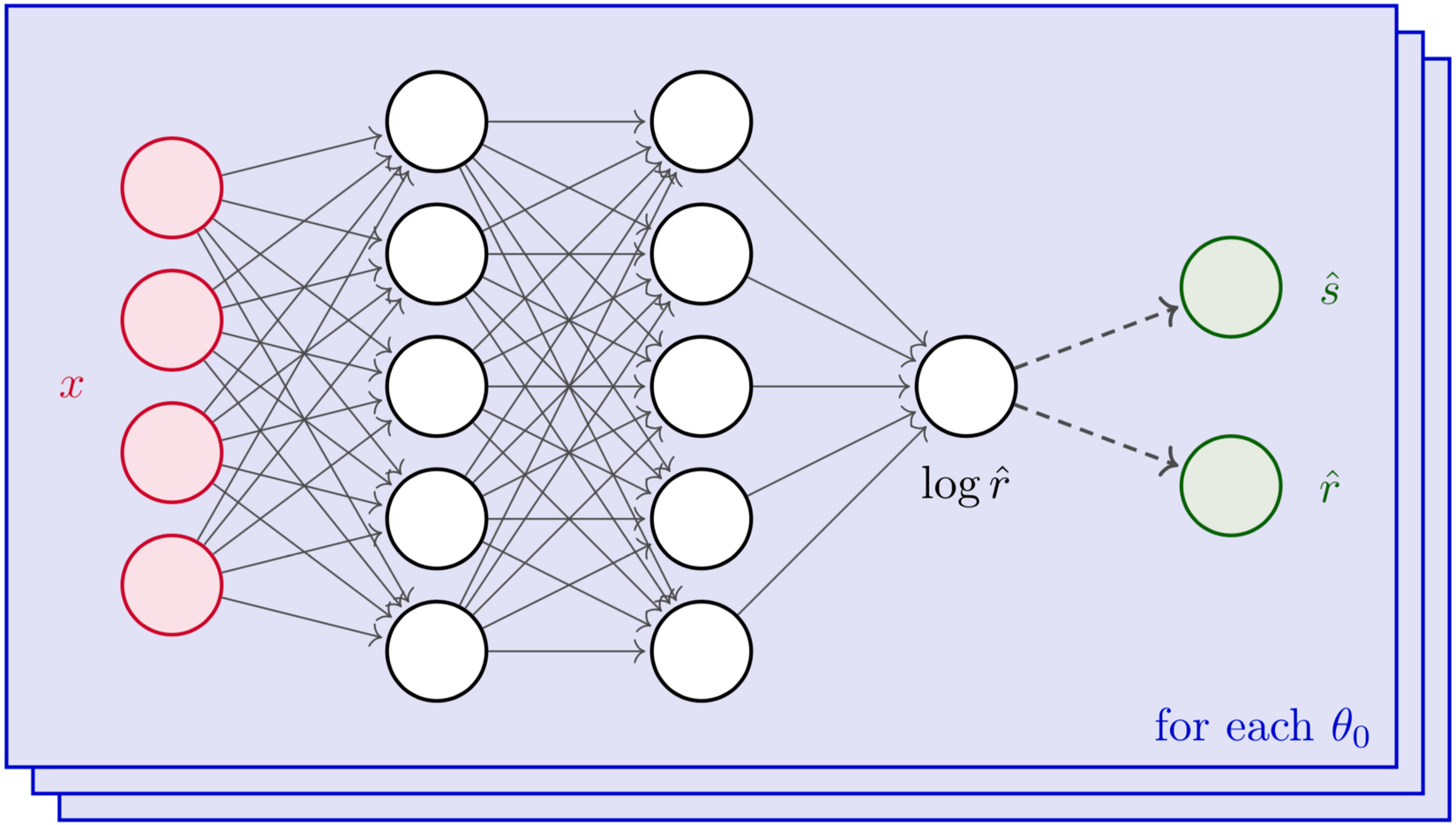

B) Parameterized:

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

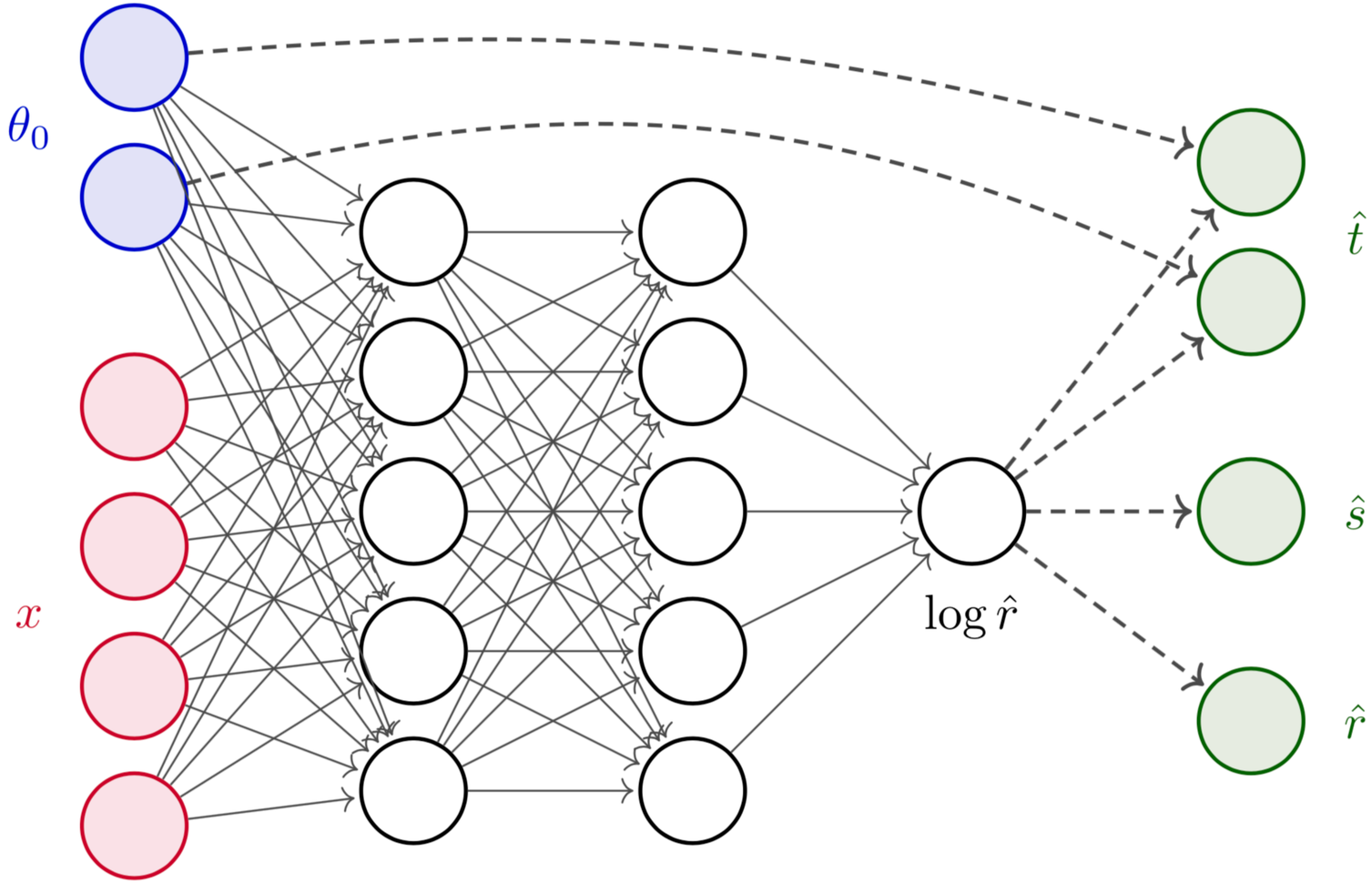
- create one estimator $\hat{r}(x|\theta_0, \theta_1)$
  that is a function of $\theta_0$ and $\theta_1$
- no further interpolation necessary
- "borrows information" from close points



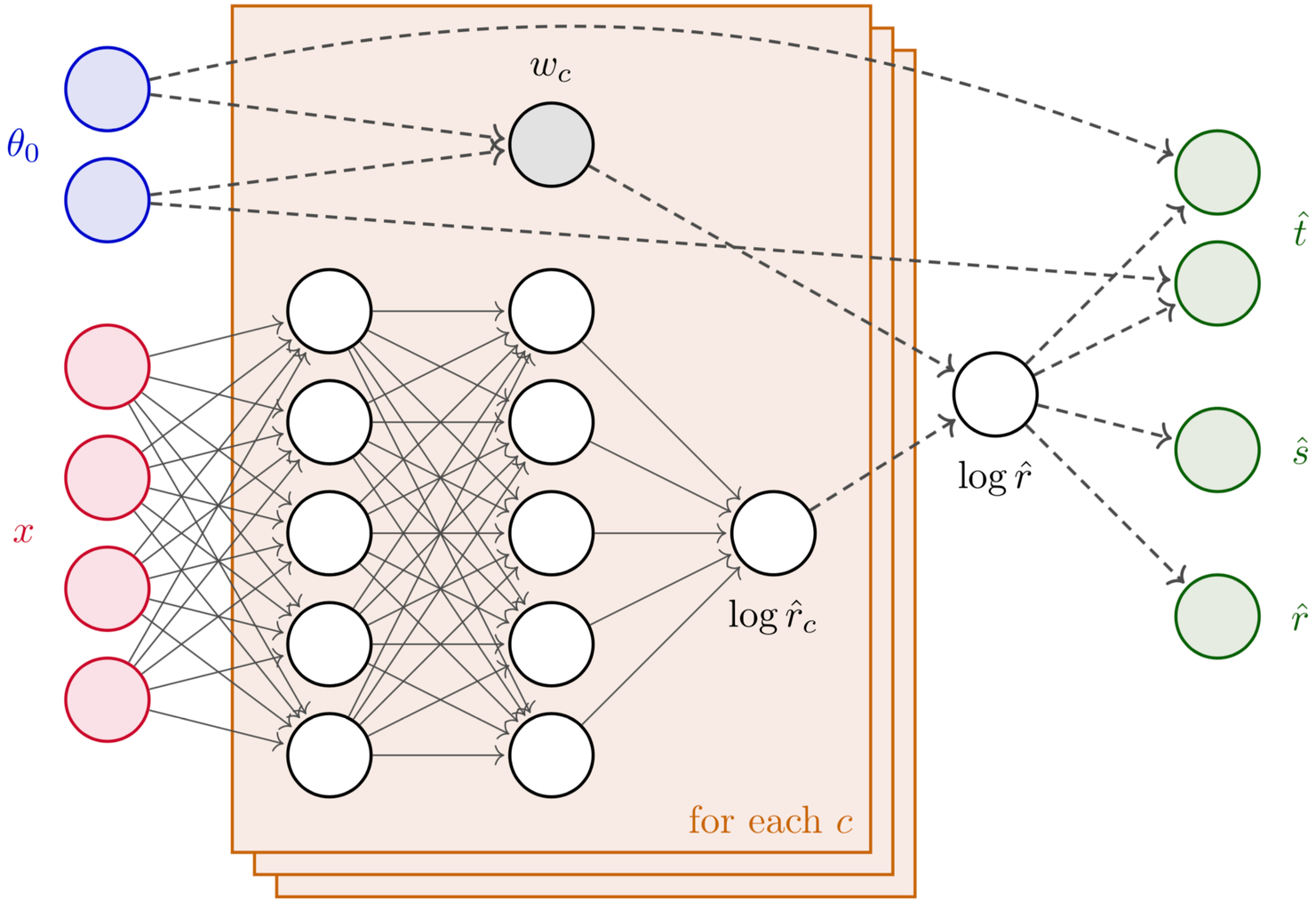
# Point by point



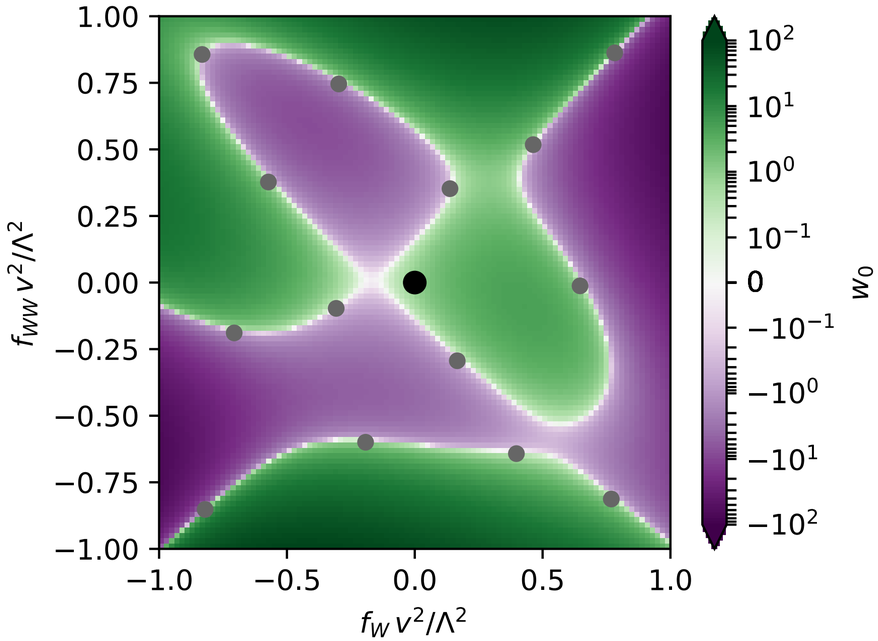
# (Physics-agnostic) parameterized estimators



# Morphing-aware parameterized estimators



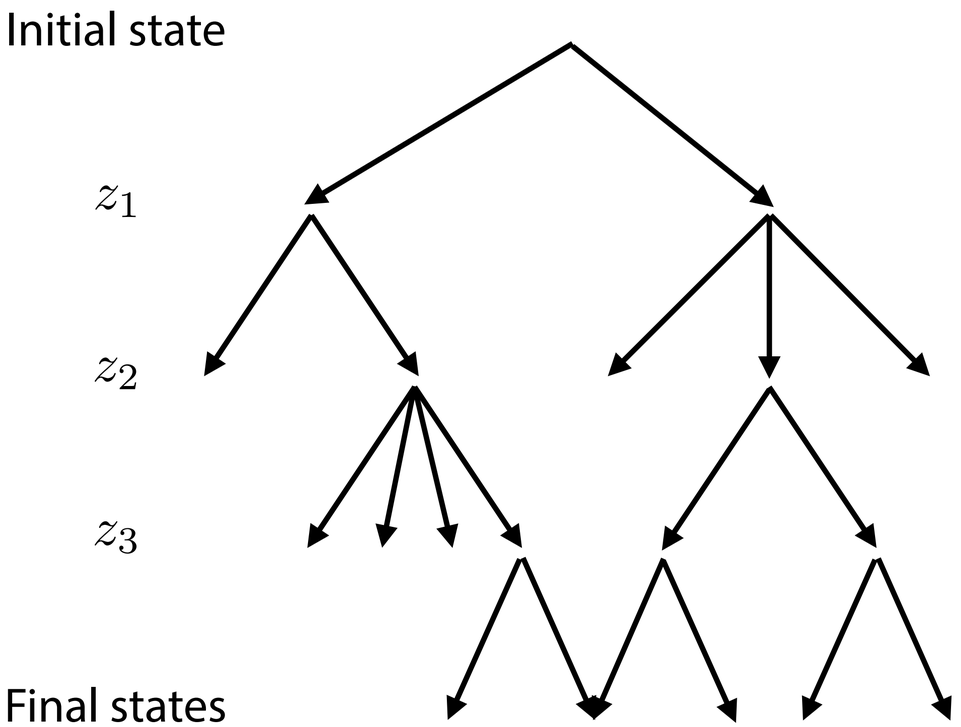

$$\hat{r}(x|\theta_0,\theta_1) = \sum_c w_c(\theta_0)\,\hat{r}_c(x)$$



# 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()
```



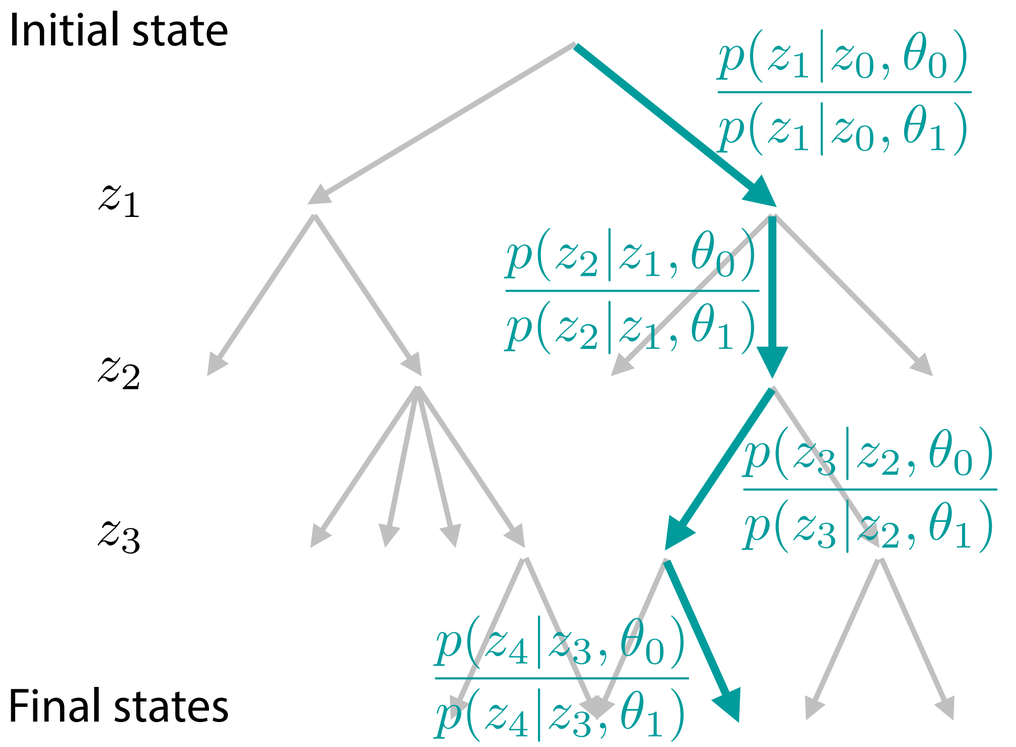
# Mining gold from any simulation

- 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

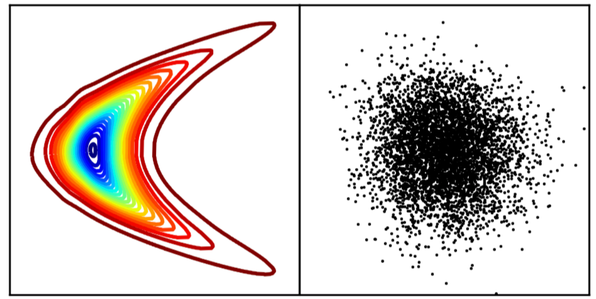
| Method | Simulate | Extract $r(x,z)$ | Extract $t(x,z)$ | NN estimates | Asympt. exact | Generative |
|---|---|---|---|---|---|---|
| ROLR | $\theta_0 \sim \pi(\theta), \theta_1$ | ✓ | | $\hat{r}(x\|\theta_0,\theta_1)$ | ✓ | |
| CASCAL | $\theta_0 \sim \pi(\theta), \theta_1$ | | ✓ | $\hat{r}(x\|\theta_0,\theta_1)$ | ✓ | |
| 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. | |



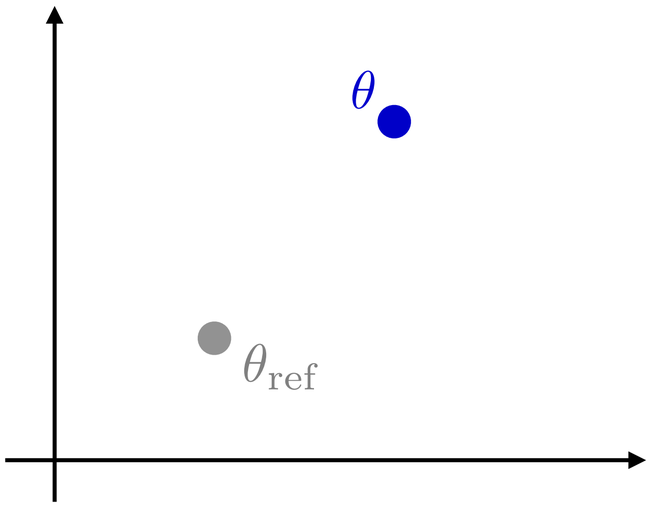
Combination with state-of-the-art conditional neural density estimators, e.g. normalizing flows

[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)

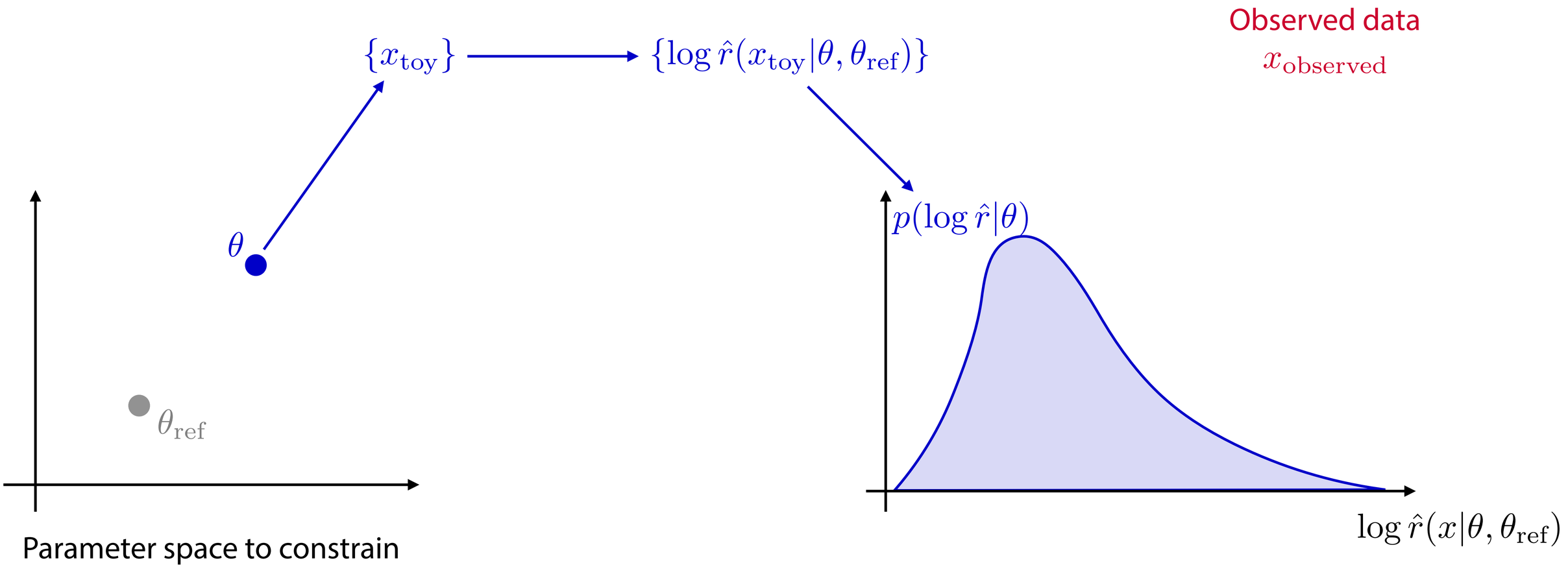
Observed data

$x_{\text{observed}}$

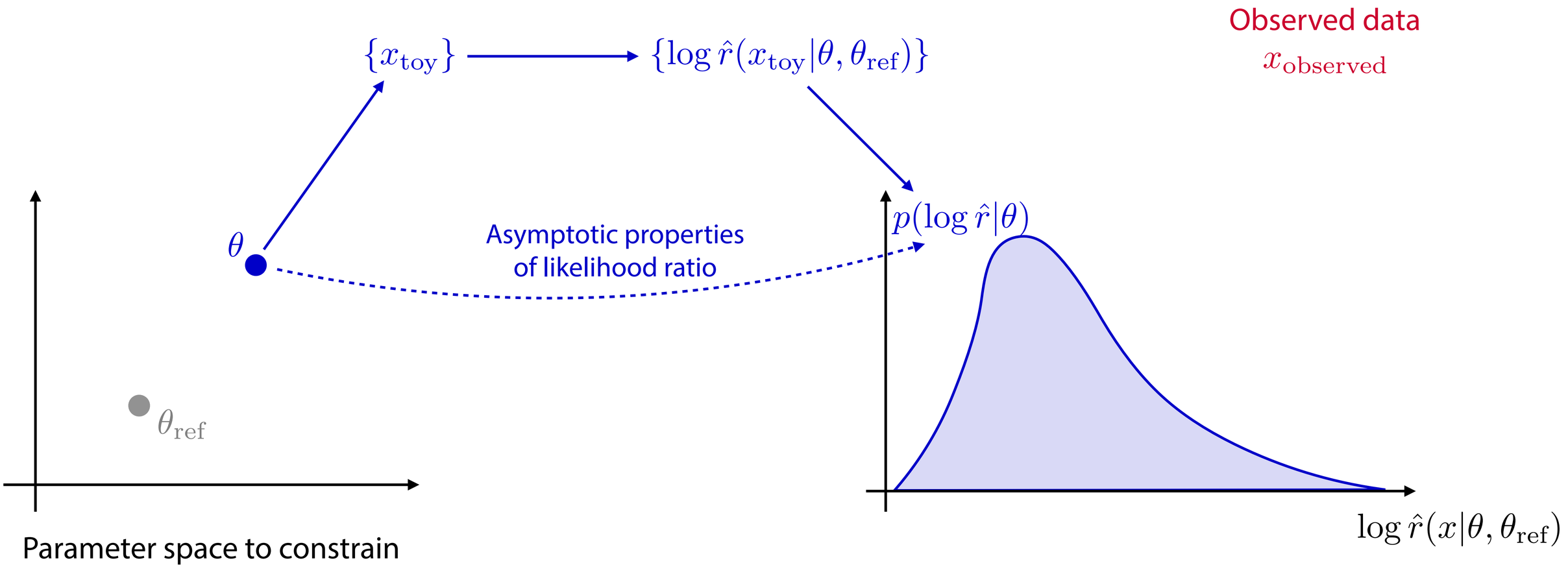



Parameter space to constrain

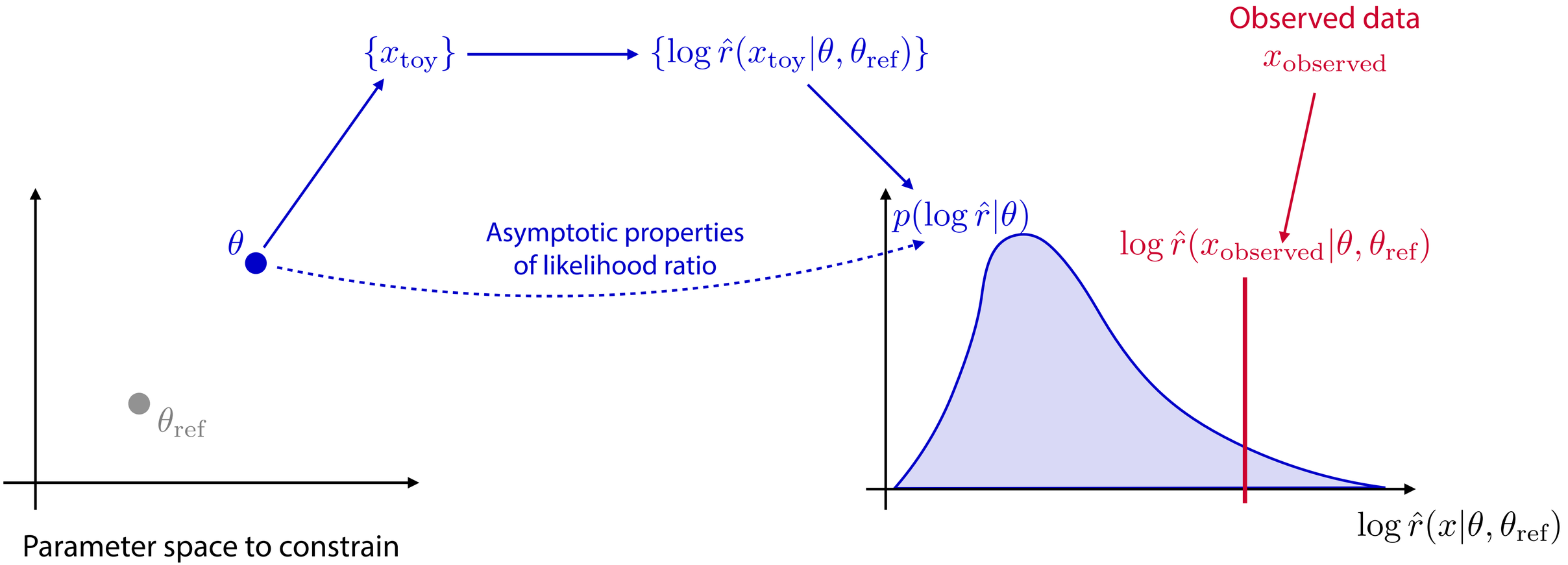
# Limit setting (frequentist, standard ATLAS / CMS practice)



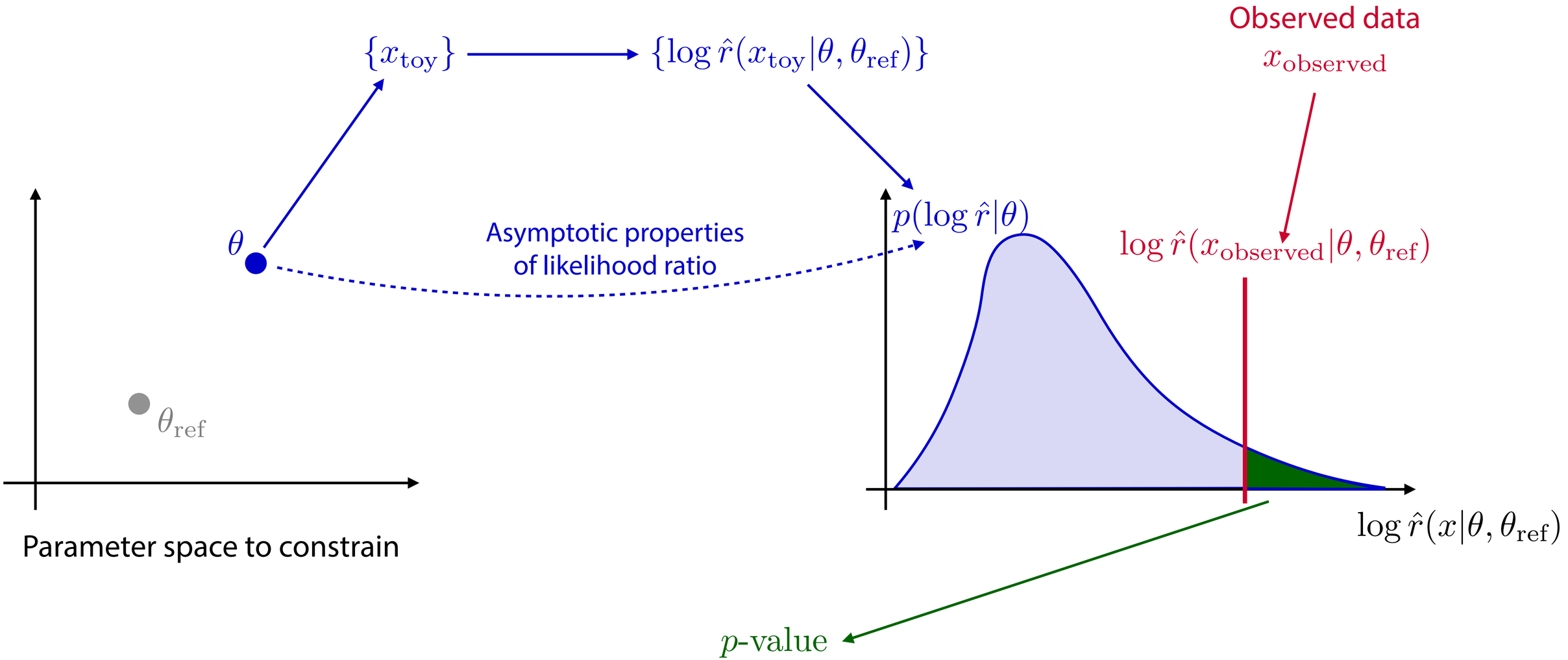
# Limit setting (frequentist, standard ATLAS / CMS practice)



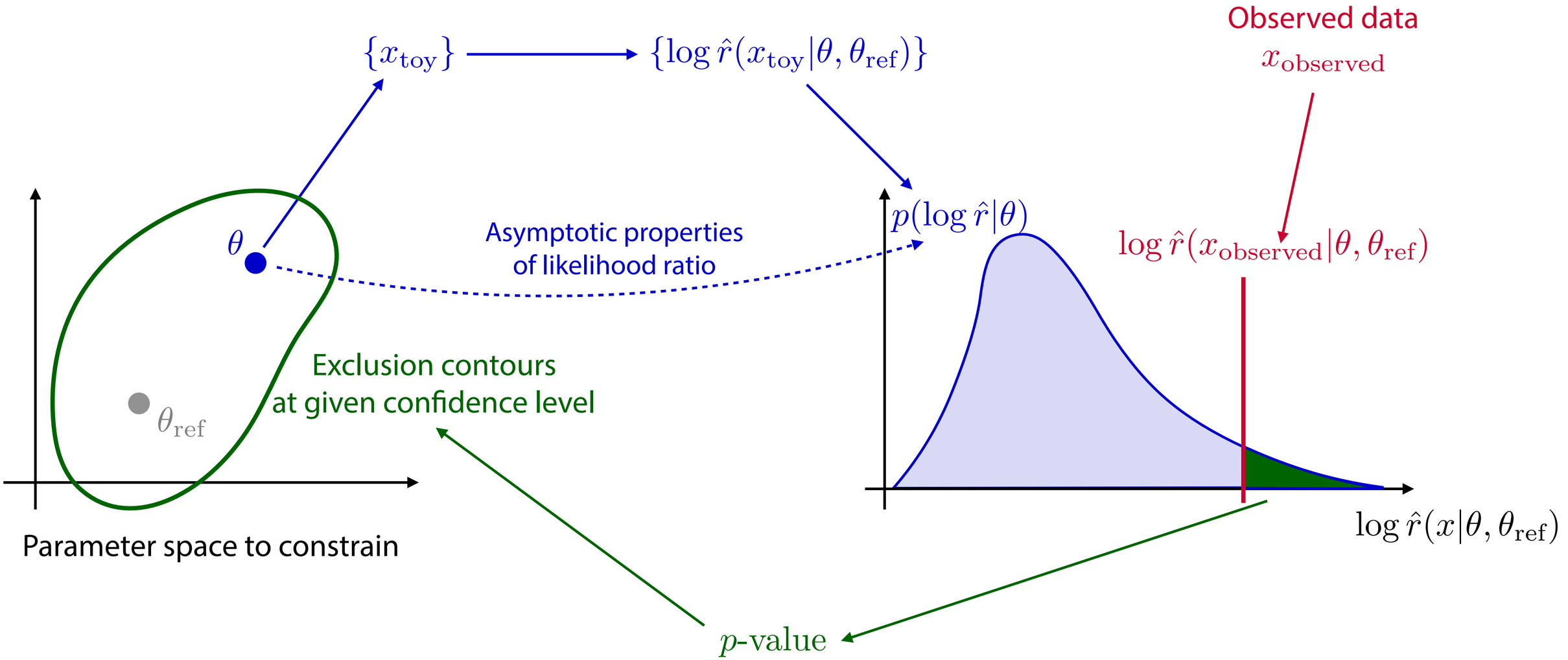
# Limit setting (frequentist, standard ATLAS / CMS practice)



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



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

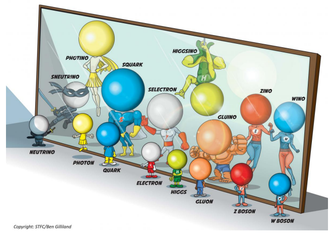


[STFC / Ben Gililand, Sean Carroll, Friedrich Justin Bertuch 1806, symmetry]

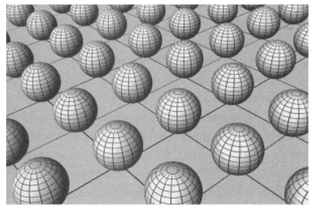
# Effective field theory

Energy

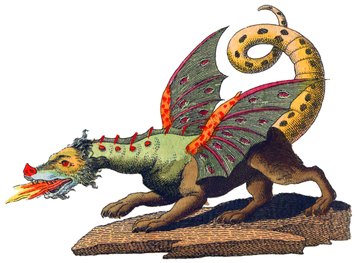
$\Lambda$




Supersymmetry?



Extra dimensions?



Dragons?


Composite Higgs?

…

Matching

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

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; …]

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

## 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

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

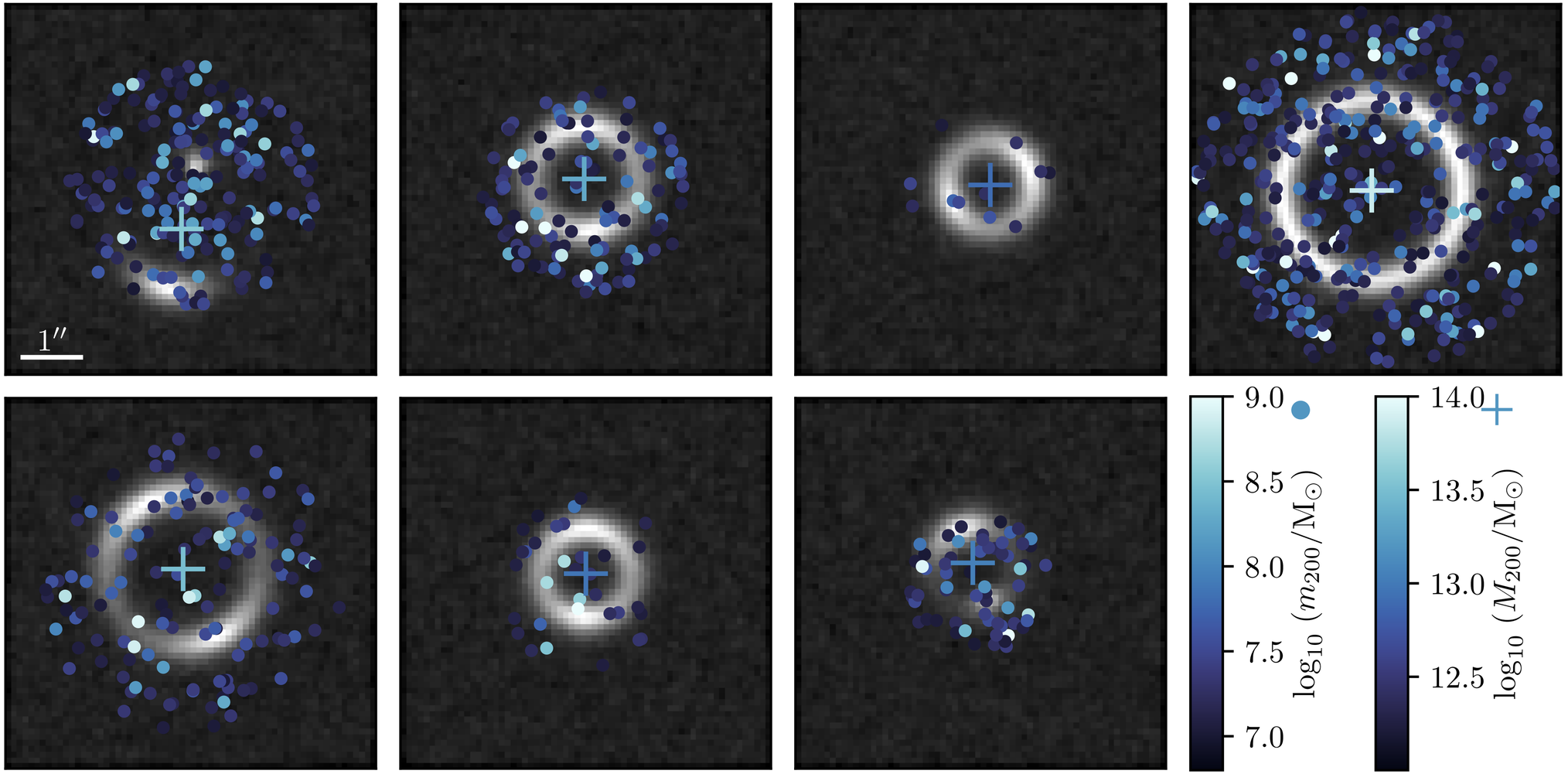
## Wilson coefficients

- 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