Normalizing flows and the likelihood ratio trick in particle physics

Johann Brehmer

New York University

Deep Learning Seminar, Uni Bremen
January 6, 2020

Collaborators

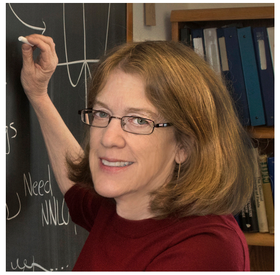
Kyle Cranmer

Gilles Louppe










NYU

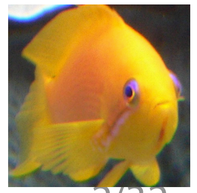
MSDSE
Moore - Sloan
Data Science Environments

dianahep

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

The SCAILFIN Project

scailfin.github.io

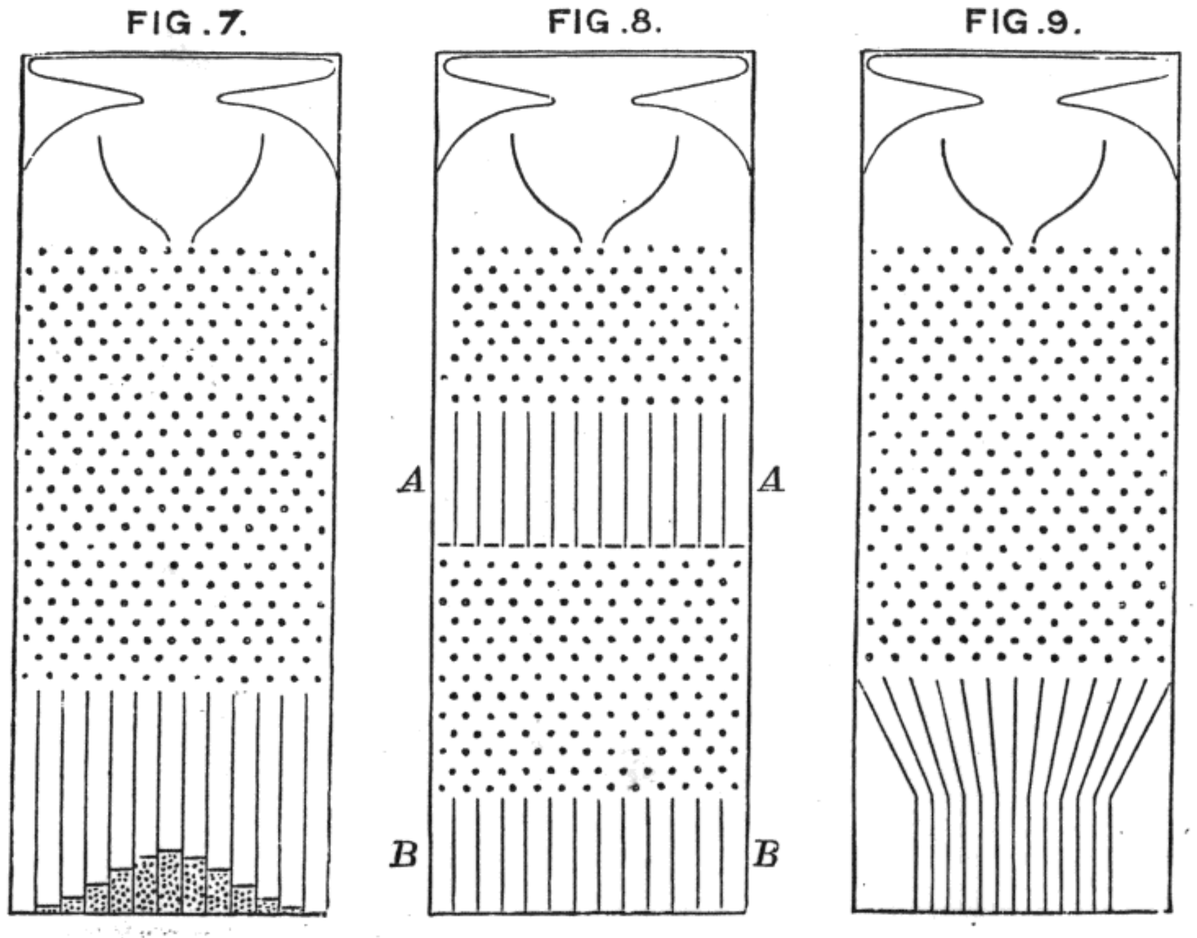


Scientists use simulations to model Nature,
but that leads to a problem for inference.

The Galton board

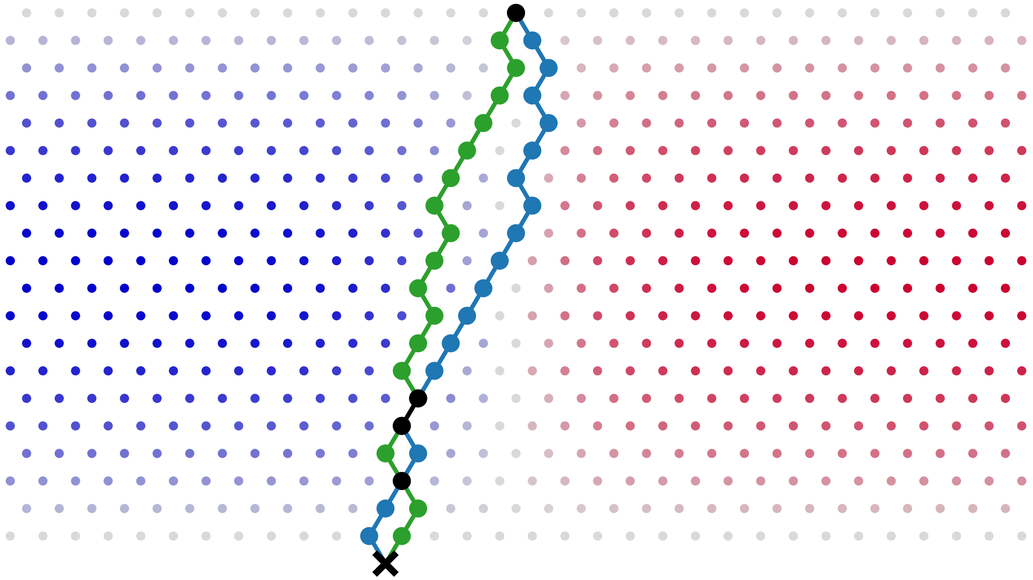
[galtonboard.com]

The Galton board



[F. Galton 1889]

Probabilities from integrating trajectories



Probability of ending in bin x : $p(x) = \int \mathrm{d}z \; p(x, z)$

Sum over all trajectories ("latent variables")

Probability of each path z from start to x

The generalized Galton board

What if probability to go left at a nail is not always 0.5, but some (known) function of some parameters θ ?

- Prediction: given θ, generate samples of observations $\{x_i\}$.

  Simple: just drop balls!

The generalized Galton board

What if probability to go left at a nail is not always 0.5, but some (known) function of some parameters θ ?

- Prediction: given θ, generate samples of observations $\{x_i\}$.

  Simple: just drop balls!

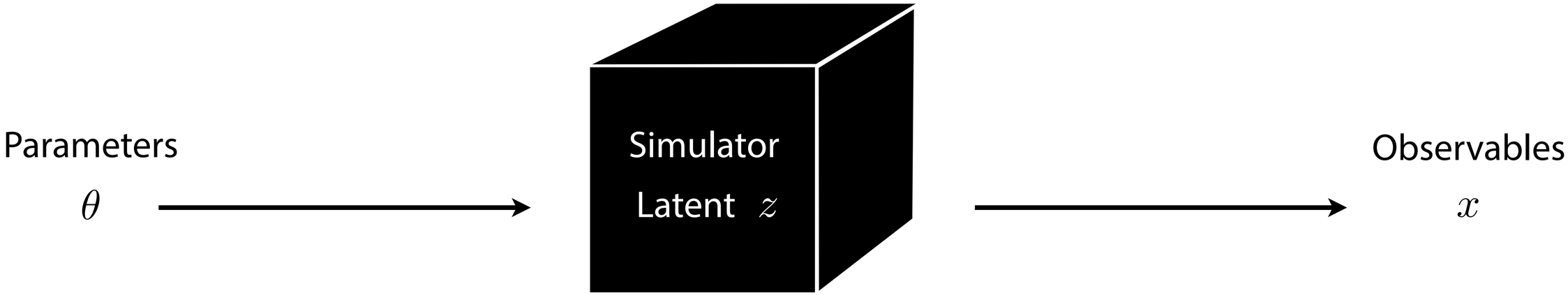
- Inference: given observations $\{x_i\}$, what are the most likely values for θ ?

  Usually we solve this with the likelihood

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

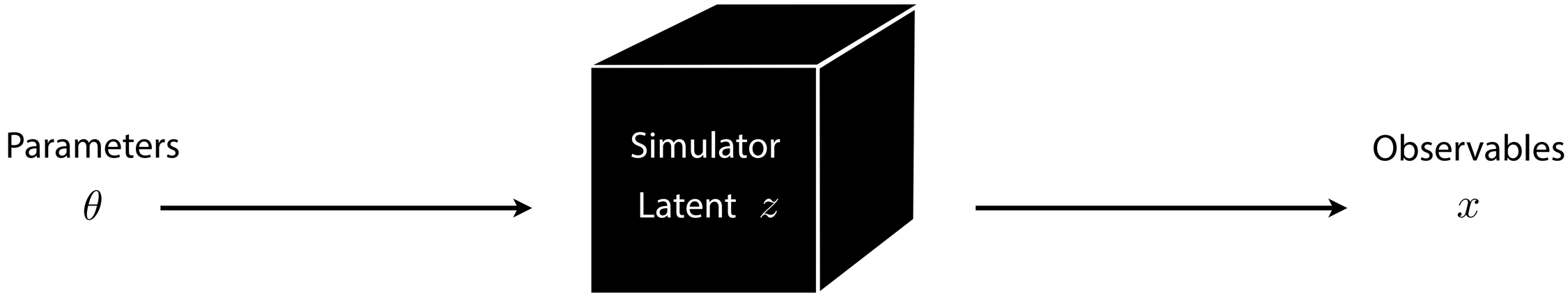
  But the number of possible paths z can be huge, and it becomes impossible to calculate the integral!

Simulation-based (“likelihood-free”) inference



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

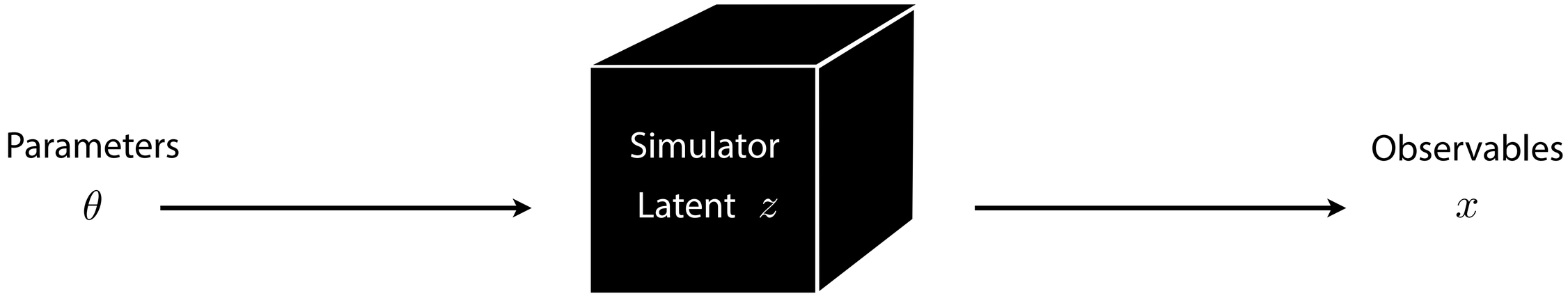
Simulation-based (“likelihood-free”) inference



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

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

Simulation-based (“likelihood-free”) inference

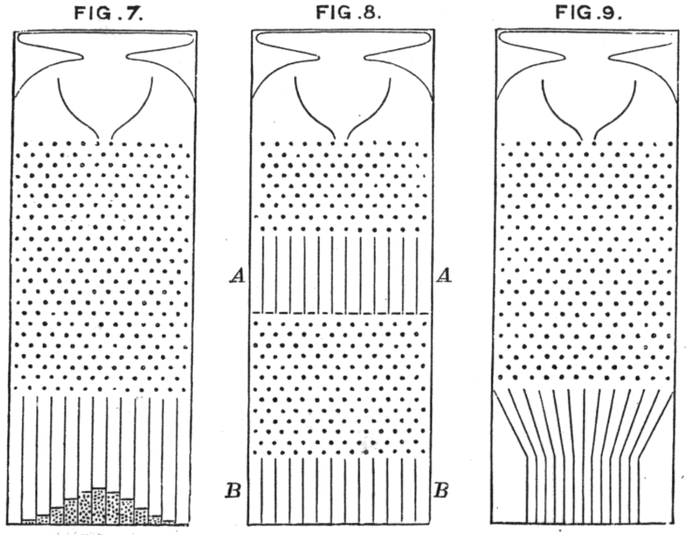


$$p(x|\theta) = \int \mathrm{d}z \; p(x, z|\theta)$$

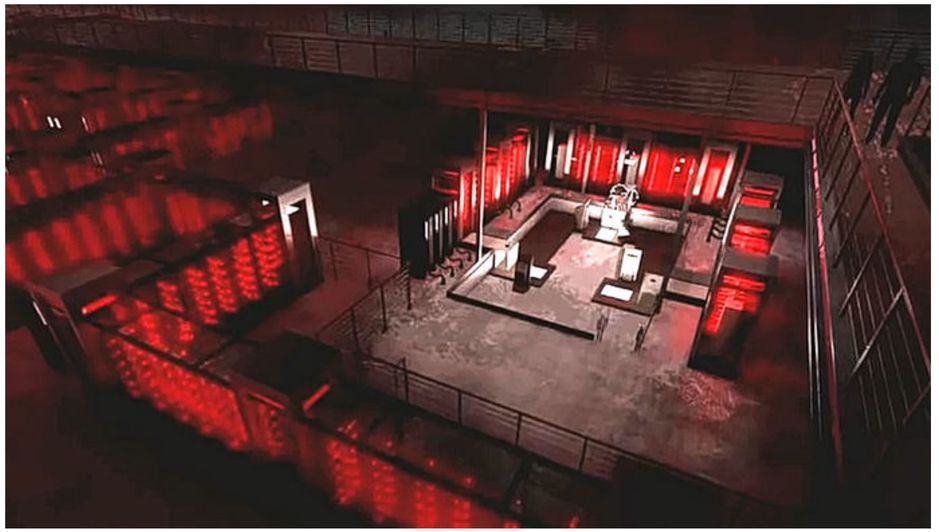
Prediction:
- Well-understood mechanistic model
- Simulator can sample $x \sim p(x|\theta)$

Inference:
- Likelihood $p(x|\theta)$ is intractable
- Inference challenging, needs e.g. estimator $\hat{p}(x|\theta)$

Galton board: metaphor for simulation-based science



[F. Galton 1889]



[HBO 2018]

Galton board device	$\longrightarrow$	Computer simulation
Parameters θ	$\longrightarrow$	Model parameters θ
Bins x	$\longrightarrow$	Observables x
Path z	$\longrightarrow$	Latent variables z (stochastic execution trace through simulator)

This inverse problem appears for instance in particle physics.

5% visible matter:

The Standard Model passes virtually every test, up to 10^{-10} precision.

But it seems fine-tuned.

Why is there more matter than antimatter?

How does gravity fit in?

[Das Universum]

5% visible matter:

The Standard Model passes virtually every test, up to 10^{-10} precision.

But it seems fine-tuned.

Why is there more matter than antimatter?

How does gravity fit in?

27% Dark Matter

What is its (particle) nature?

[Das Universum]

5% visible matter:

The Standard Model passes virtually every test, up to 10^{-10} precision.

But it seems fine-tuned.

Why is there more matter than antimatter?

How does gravity fit in?

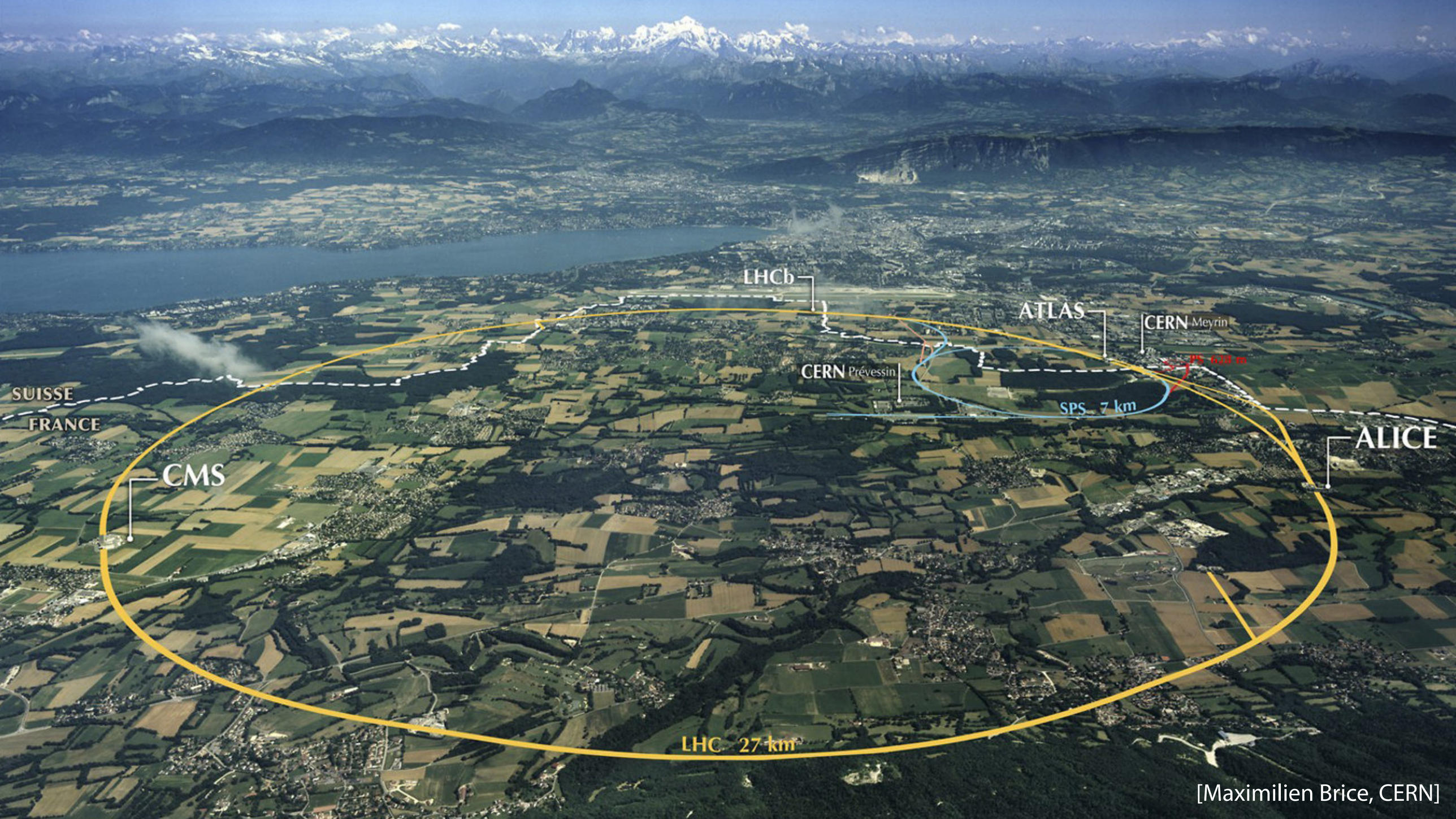
27% Dark Matter

What is its (particle) nature?

68% Dark Energy:

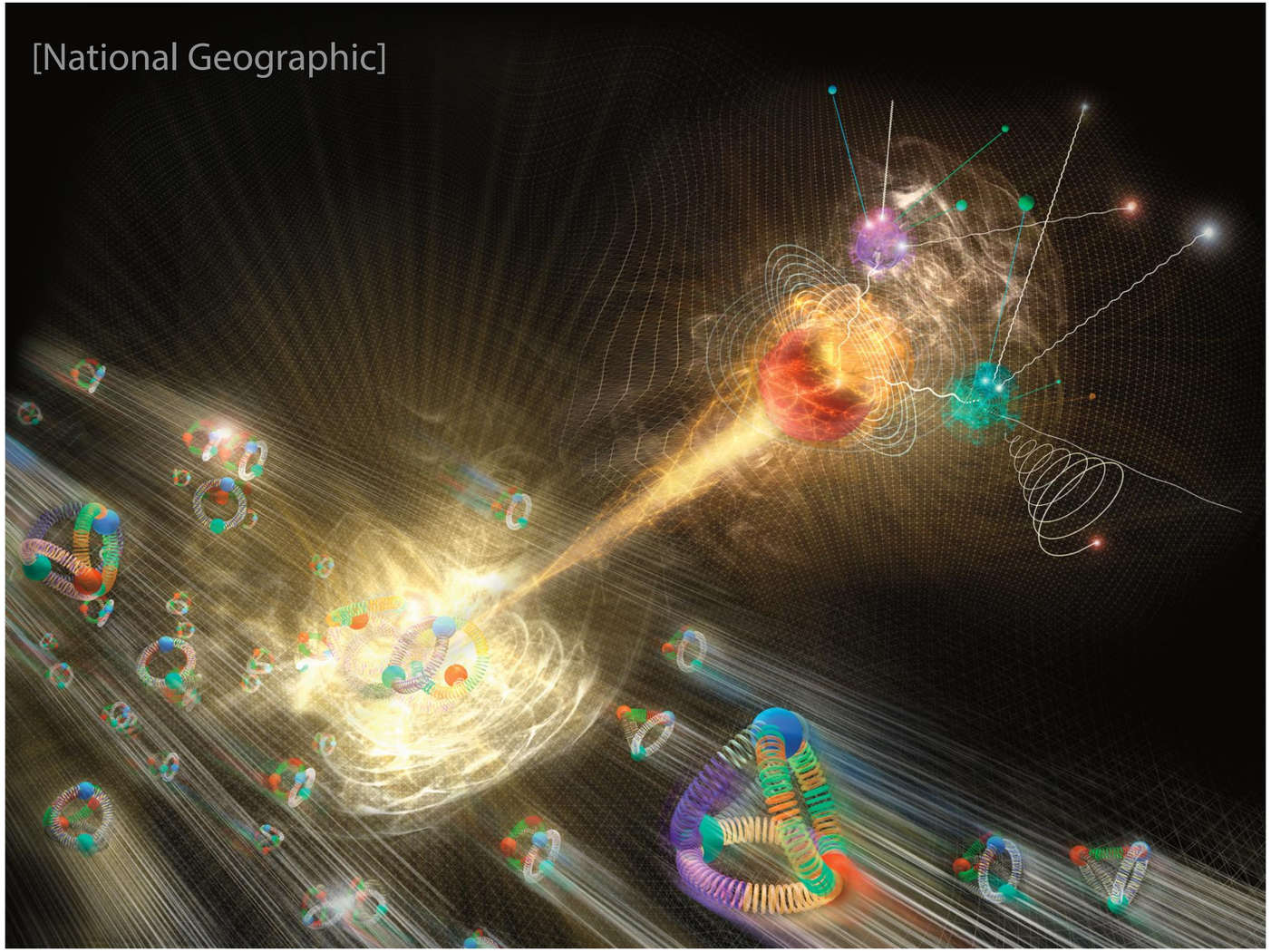
What?

[Das Universum]



[Maximilien Brice, CERN]

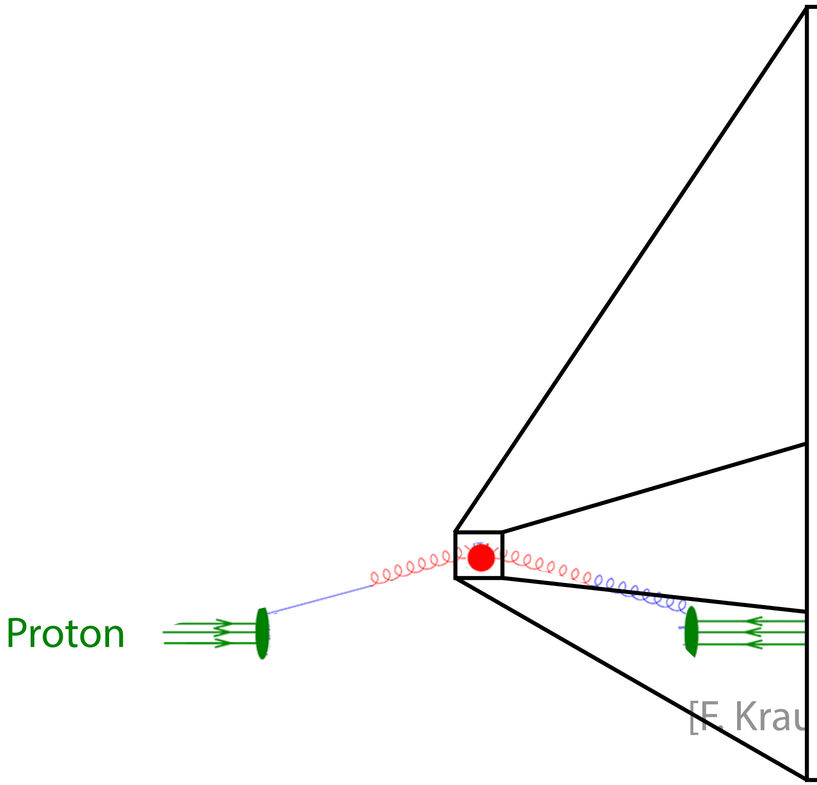
When protons meet



When protons meet

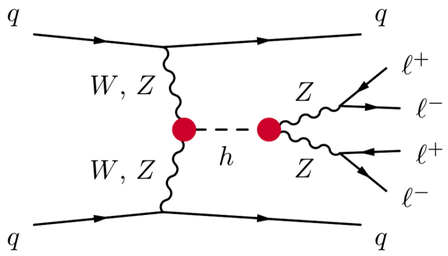
[F. Krauss]

When protons meet



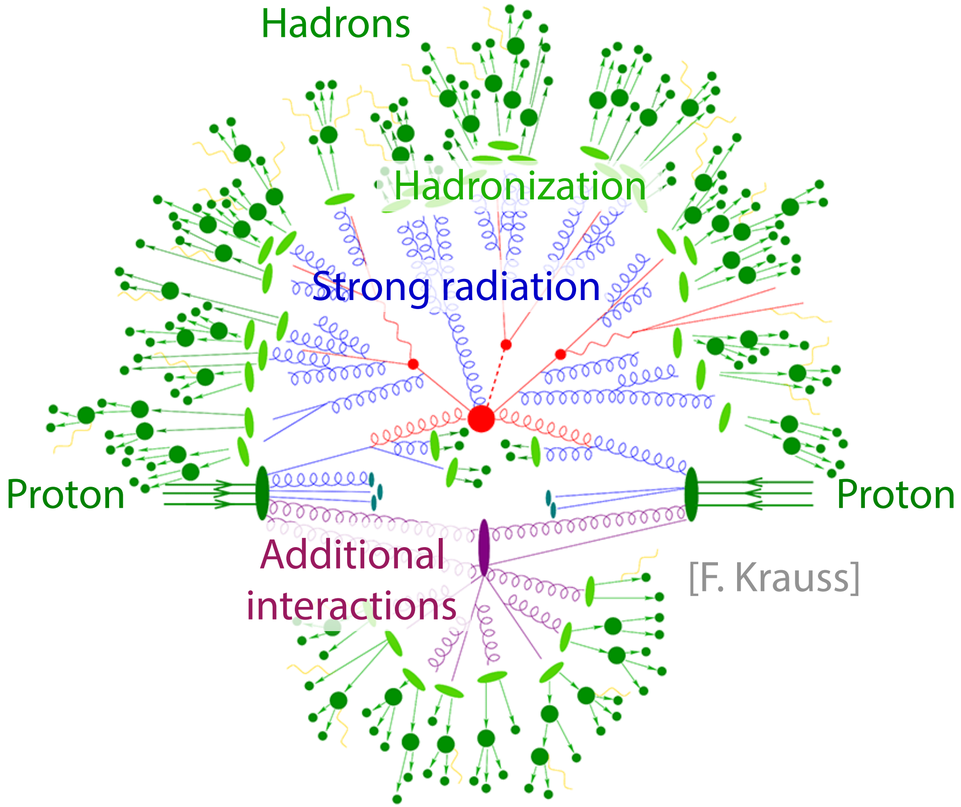
High-energy interactions:

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

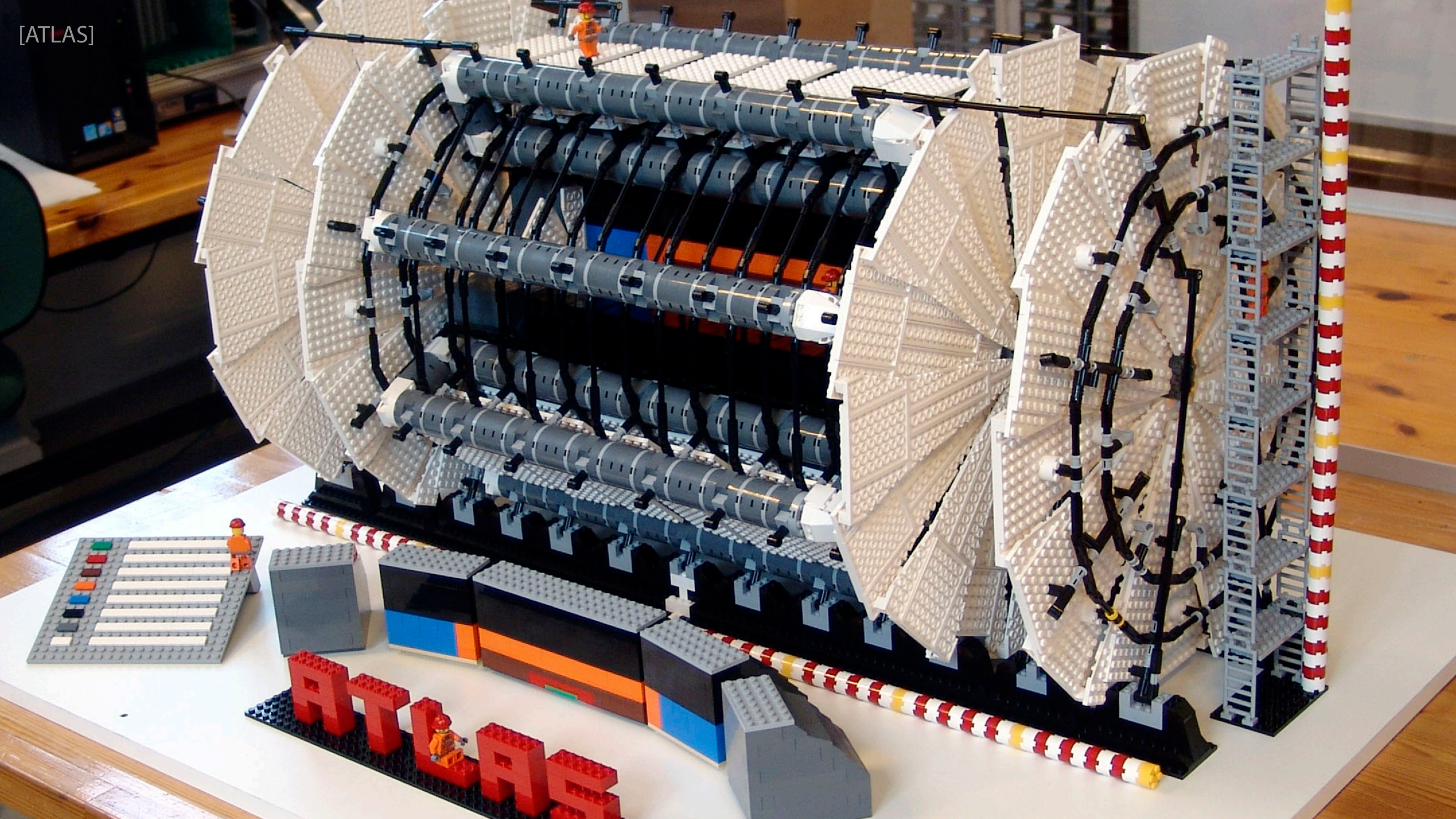


$$
\begin{aligned}
S = \int \mathrm{d}^4x \Bigg[& \mathcal{L}_{\mathrm{SM}} + \frac{f_{\phi,2}}{\Lambda^2}\,\frac{1}{2}\,\partial_\mu(\phi^\dagger\phi)\,\partial^\mu(\phi^\dagger\phi) + \frac{f_{\phi,3}}{\Lambda^2}\,\frac{1}{3}(\phi^\dagger\phi)^3 \\
& + \frac{f_{GG}}{\Lambda^2}\,(\phi^\dagger\phi)\,G^a_{\mu\nu}\,G^{\mu\nu\,a}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

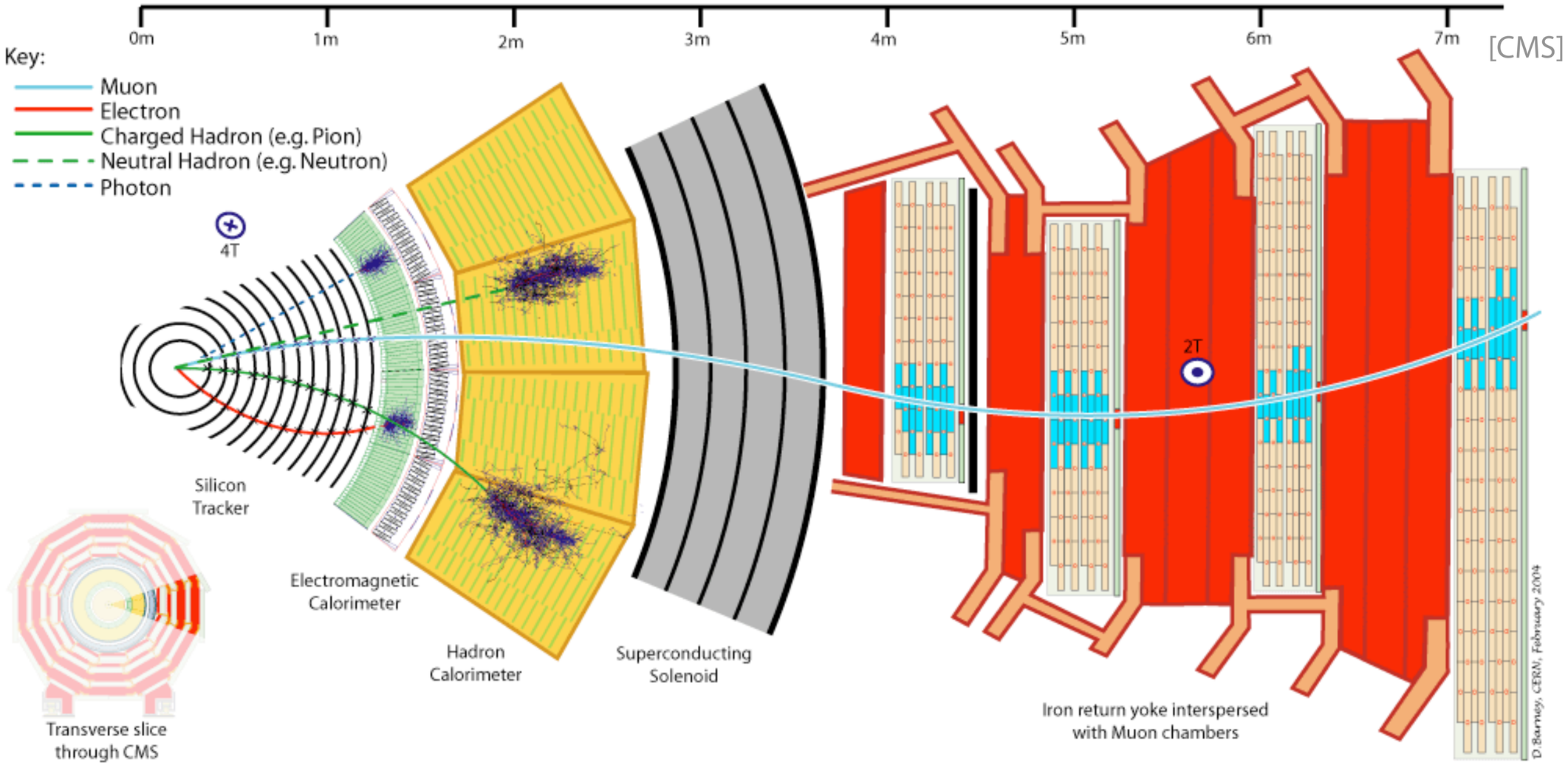
When protons meet

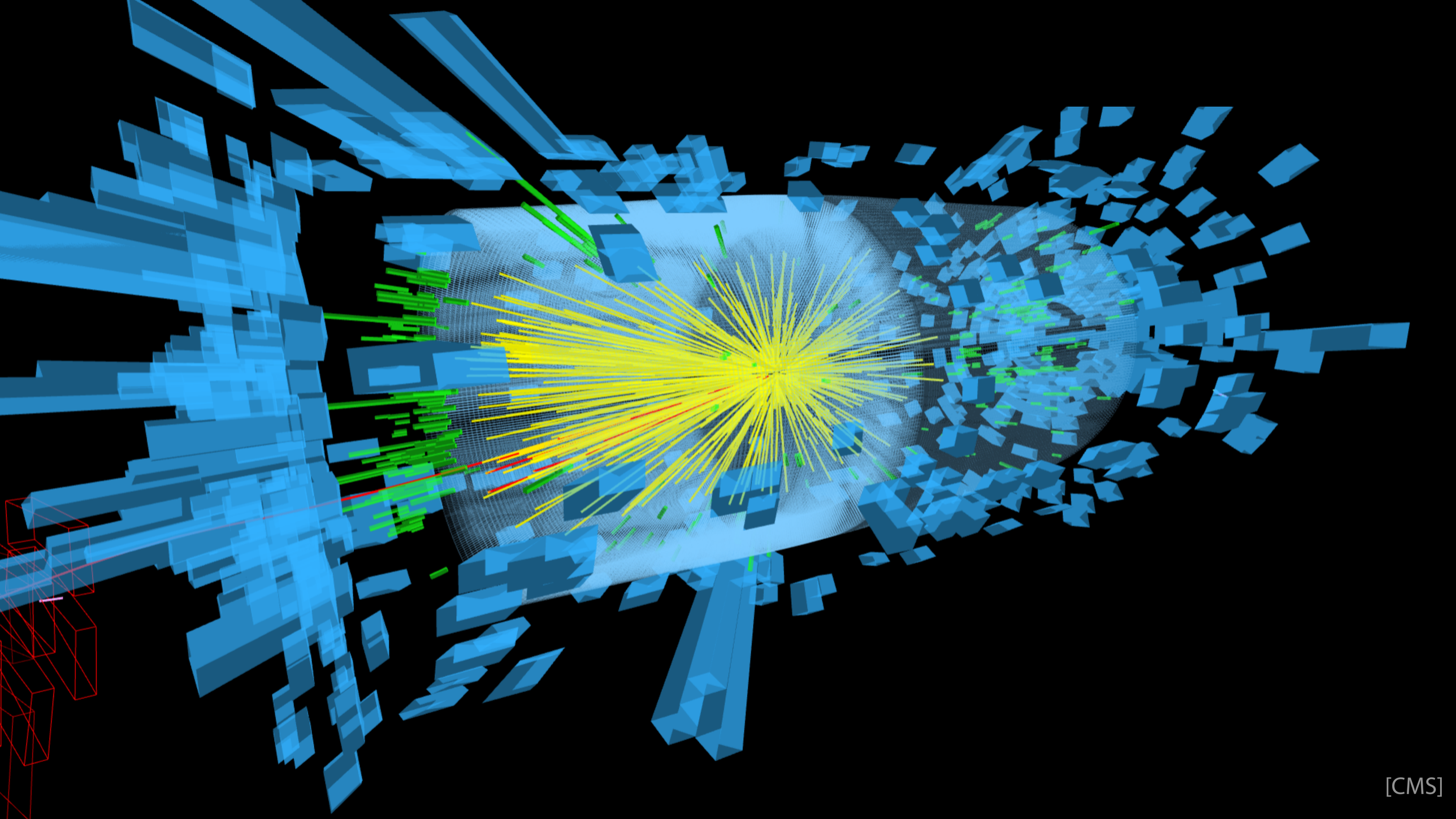


[F. Krauss]


ATLAS

Particle, meet detector




[CMS]

Our model of particle physics is “likelihood-free”

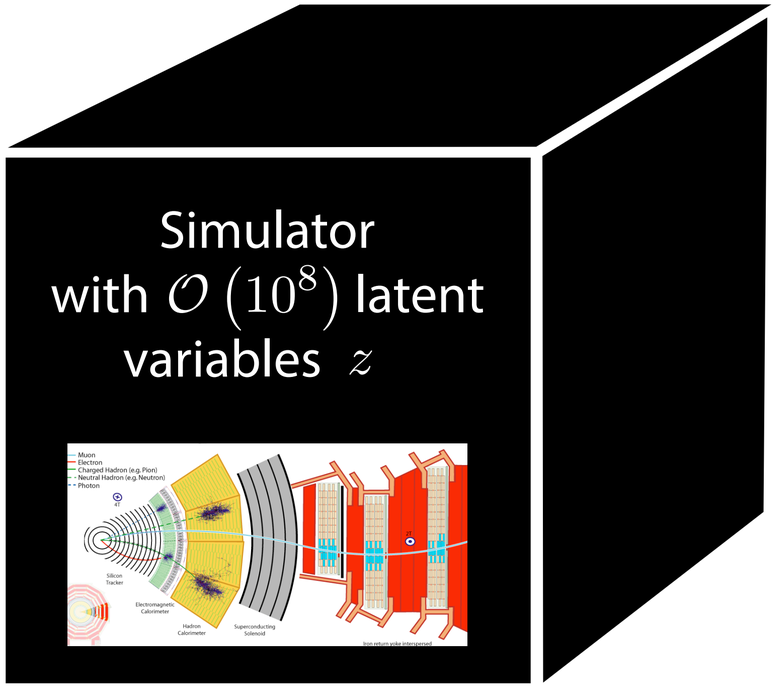
$\mathcal{O}\left(10\right)$ parameters θ

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

Our model of particle physics is “likelihood-free”

$\mathcal{O}(10)$ parameters θ

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

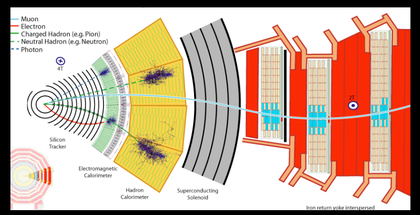


Our model of particle physics is “likelihood-free”

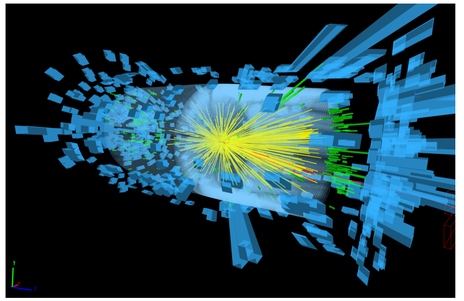
$\mathcal{O}(10)$ parameters θ

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

Simulator with $\mathcal{O}(10^8)$ latent variables z



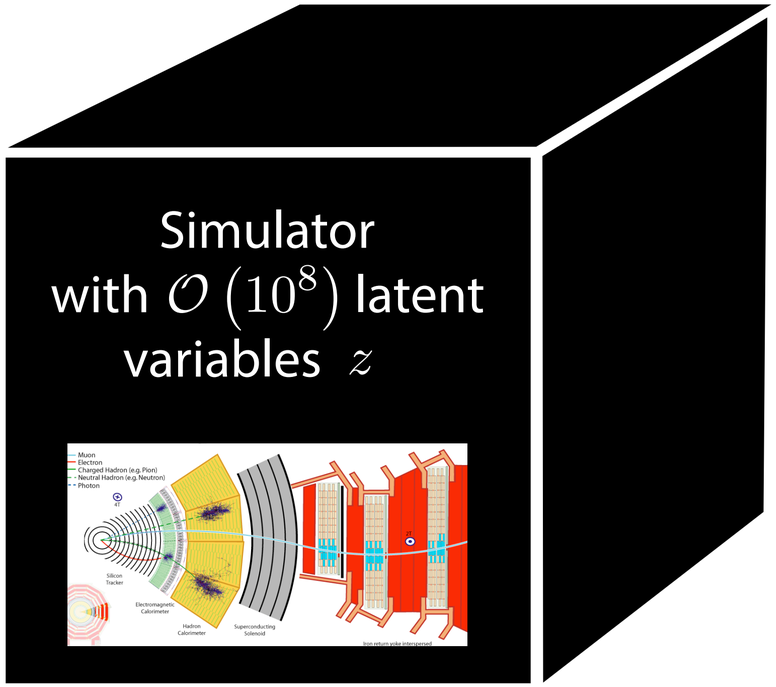
$\mathcal{O}(10 \ldots 1000)$ observables x



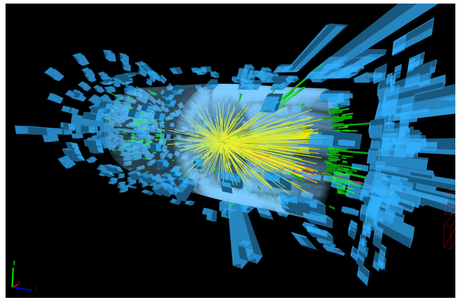
Our model of particle physics is “likelihood-free”

$\mathcal{O}(10)$ parameters θ

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



$\mathcal{O}(10 \ldots 1000)$
observables x



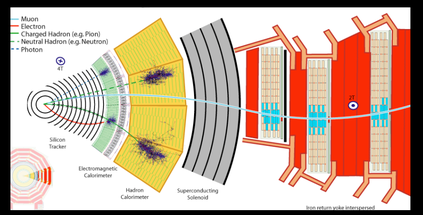
Prediction: Simulator can sample $x \sim p(x|\theta)$

Our model of particle physics is “likelihood-free”

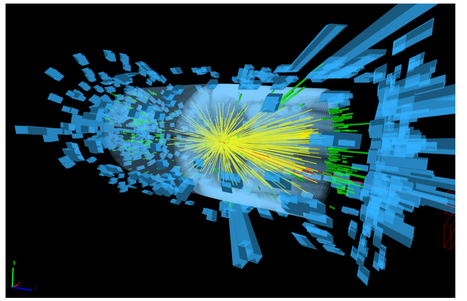
$\mathcal{O}(10)$ parameters θ

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

Simulator with $\mathcal{O}(10^8)$ latent variables z



$\mathcal{O}(10 \ldots 1000)$ observables x



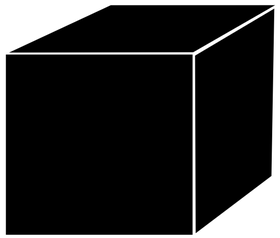
Prediction: Simulator can sample $x \sim p(x|\theta)$

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

Problem statement(s)

You are given

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

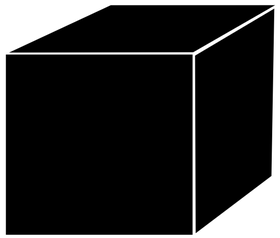


- observed data $x_{\mathrm{obs}} \sim p(x_{\mathrm{obs}}|\theta_{\mathrm{true}})$
- prior belief $p(\theta)$

Problem statement(s)

You are given

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



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

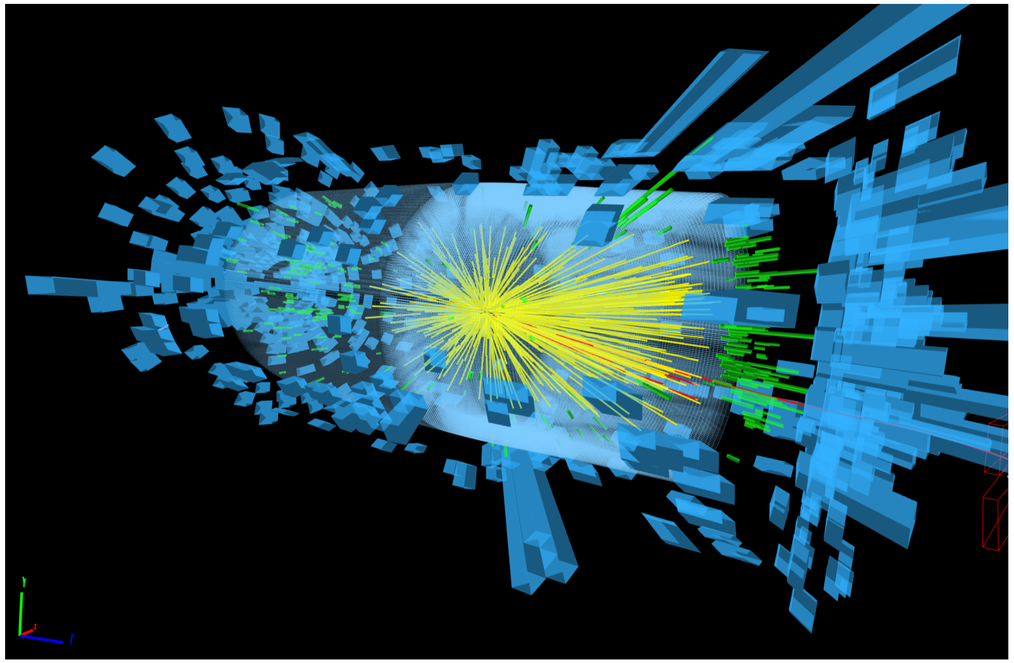
Goals: estimate either…

- 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

We can do inference
by compressing the data to summary statistics.

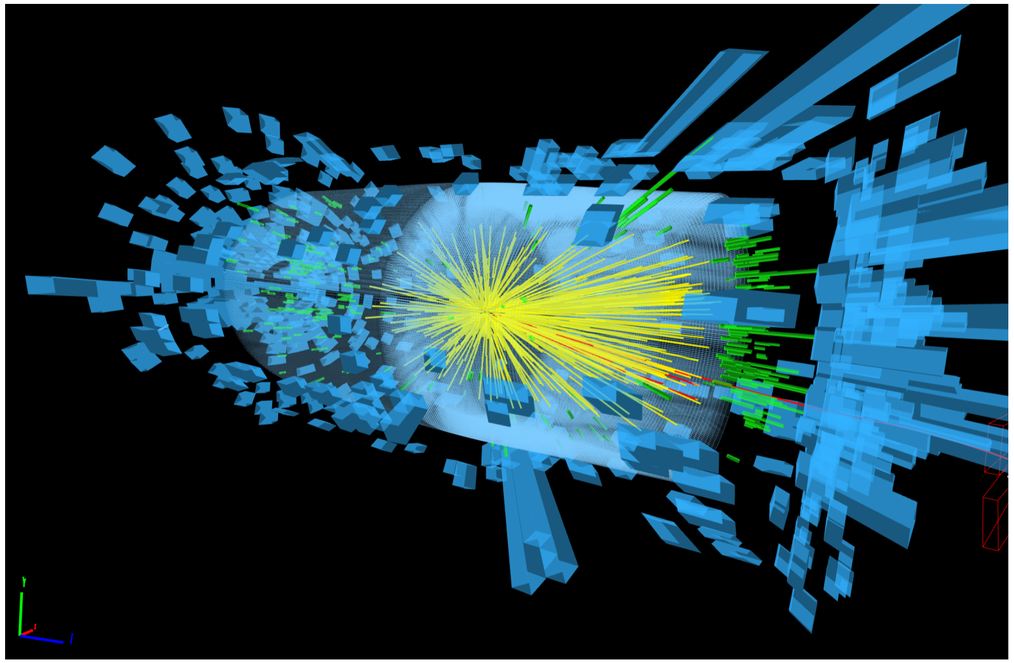
Summary statistics

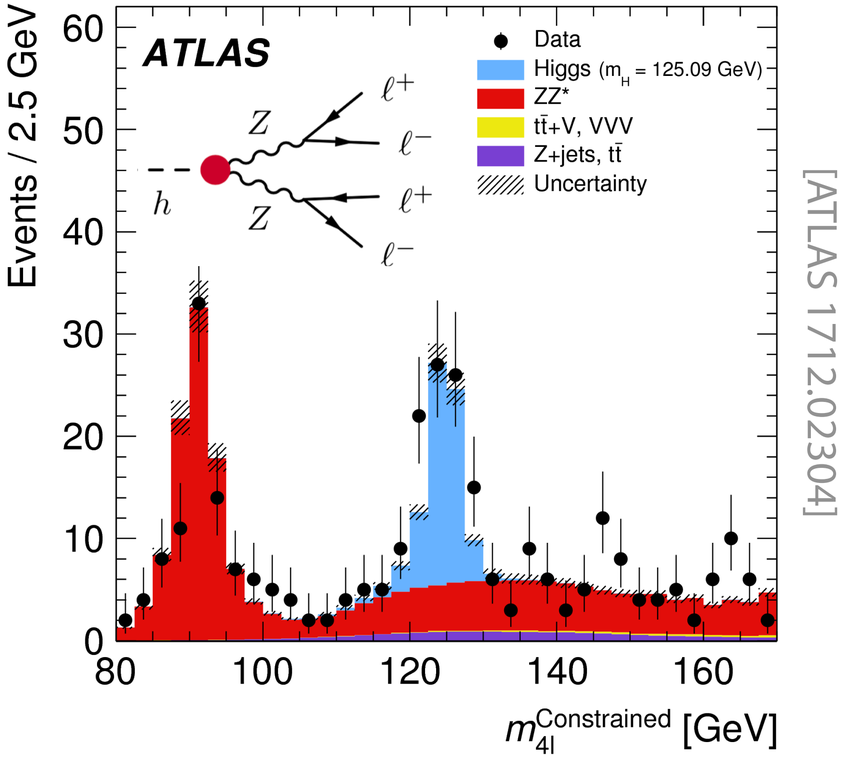


High-dimensional observable data x

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

Summary statistics





High-dimensional observable data x

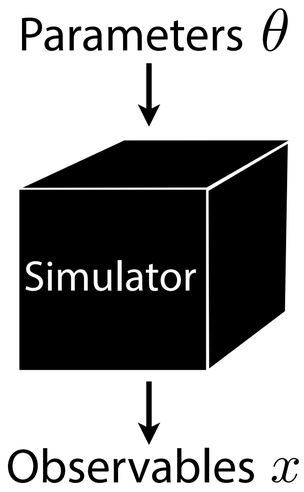
$p(x|\theta)$ cannot be calculated

One or a few summary statistics x'

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

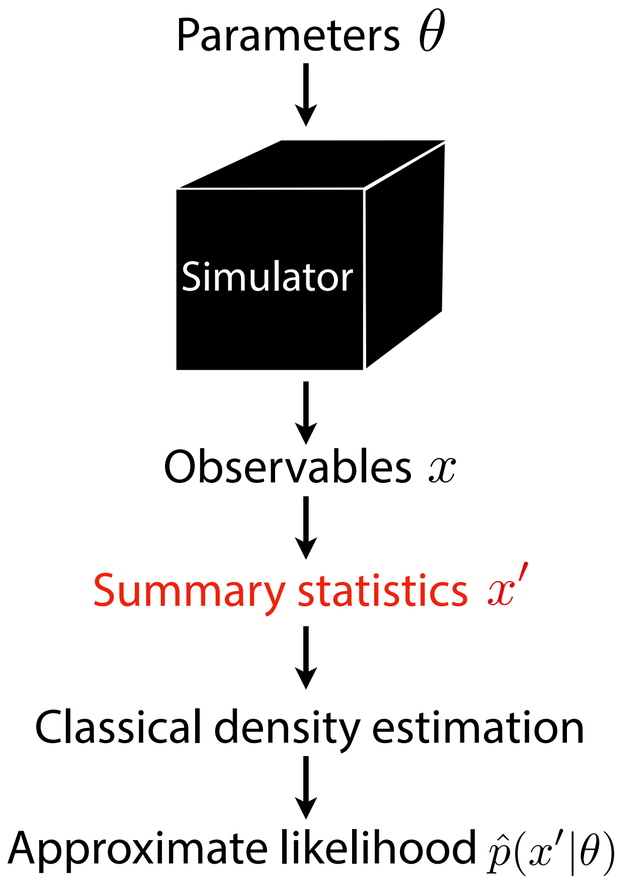
Inference by estimating the likelihood

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



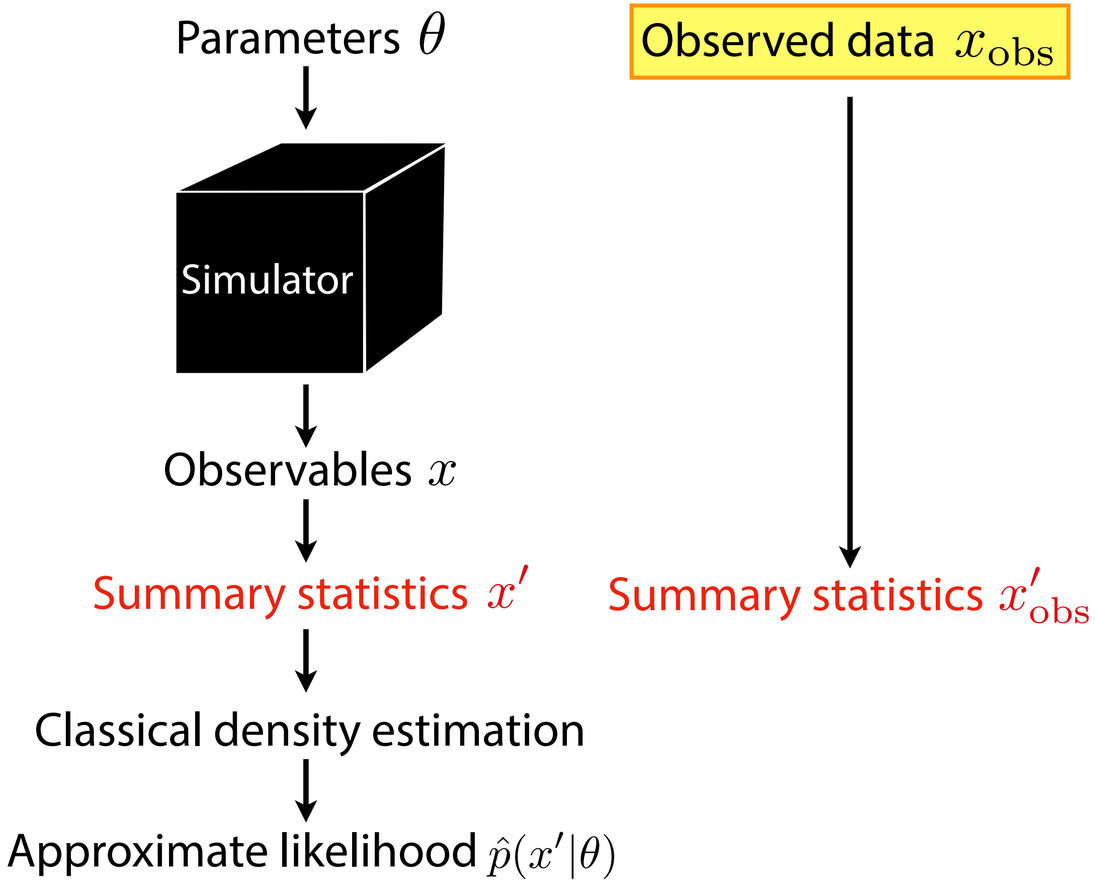
Inference by estimating the likelihood

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



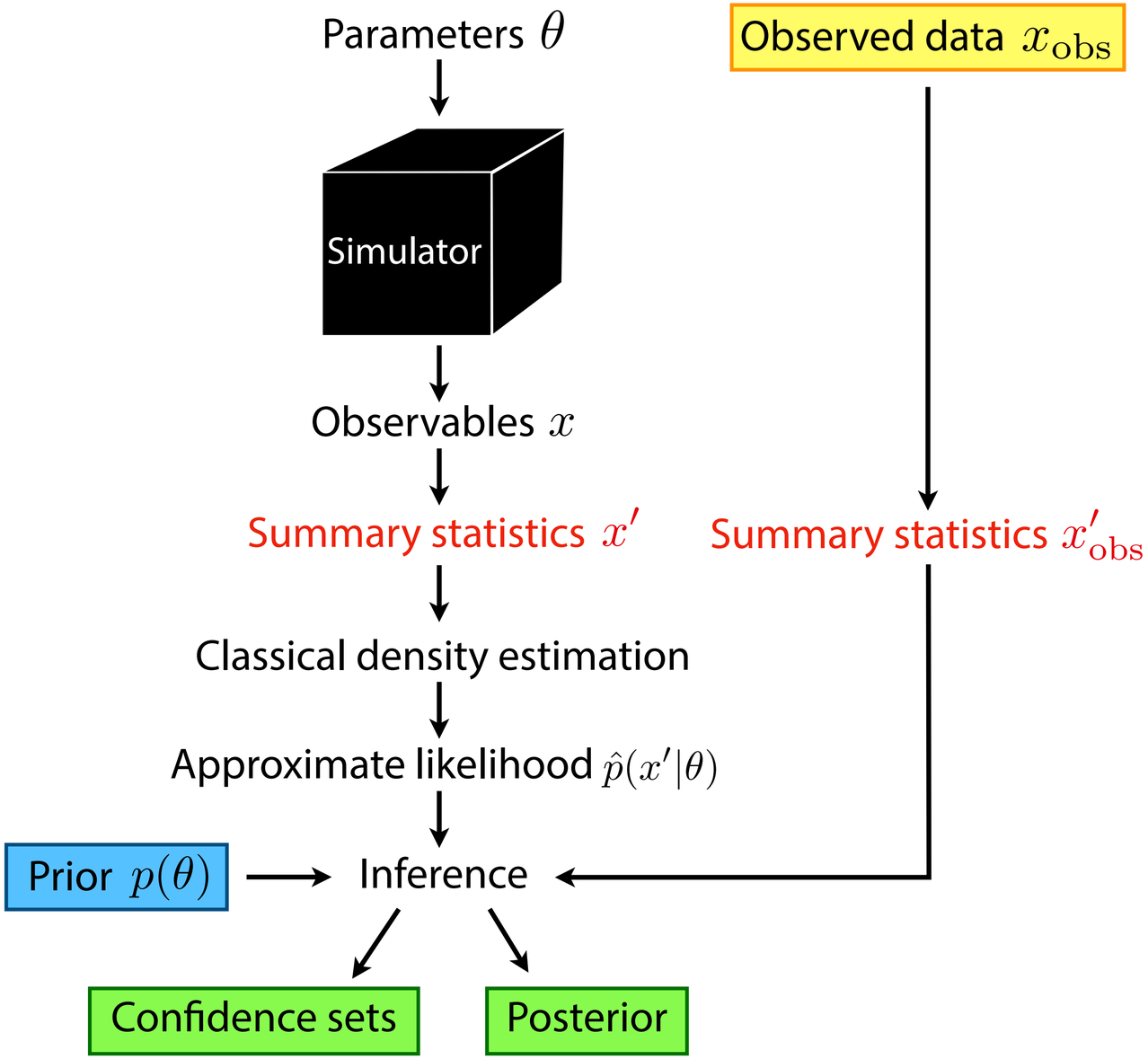
Inference by estimating the likelihood

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



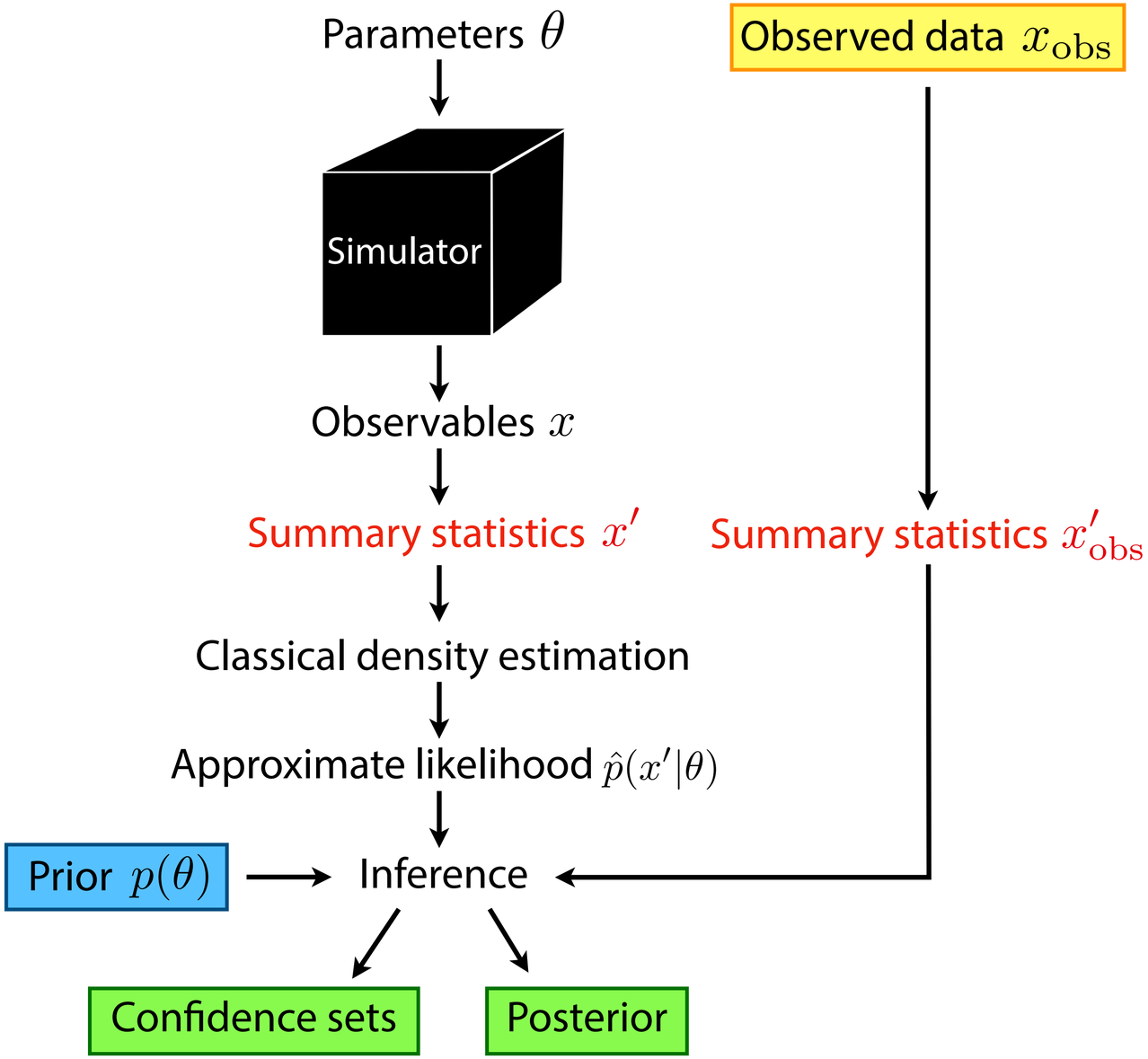
Inference by estimating the likelihood

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



Inference by estimating the likelihood

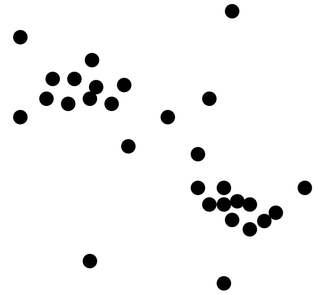
[e.g. P. Diggle, R. Gratton 1984]



- Compression to summary statistics reduces quality of inference
- Curse of dimensionality: classical density estimation methods do not scale to more than a few summary statistics
- Popular, similar alternative: Approximate Bayesian Computation (ABC) [D. Rubin 1984]

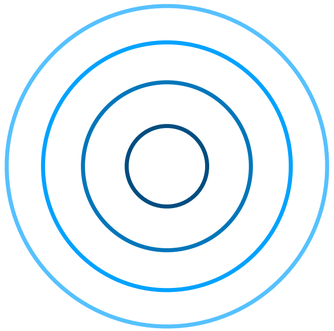
We can do inference
by using machine learning.

High-dimensional density estimation with normalizing flows



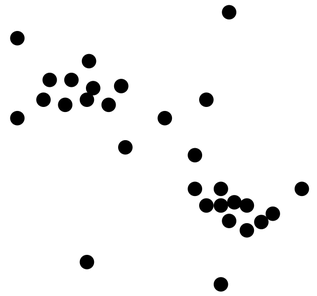
[lots of recent great work including by J. Behrmann; see G. Papamakarios et al 1912.02762 for a review]

High-dimensional density estimation with normalizing flows



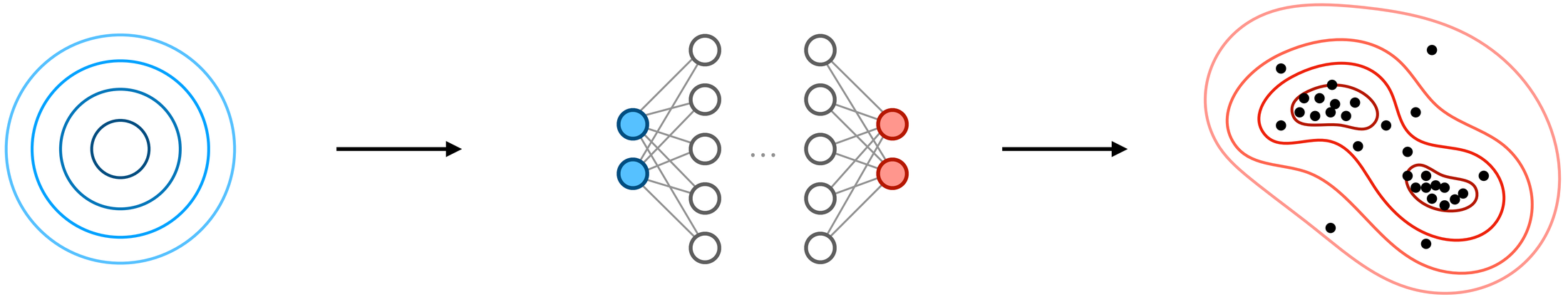
Simple base density

$u \sim \pi(u)$



[lots of recent great work including by J. Behrmann; 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 $x = f(u)$

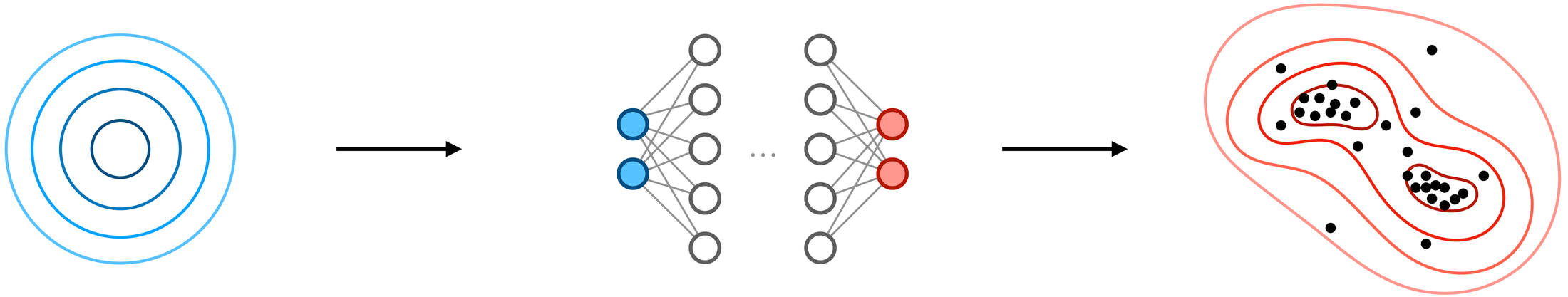
- one-to-one and invertible
- differentiable
- f^{-1} and $\det \nabla f$ are tractable

Target density is given by

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

[lots of recent great work including by J. Behrmann; 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

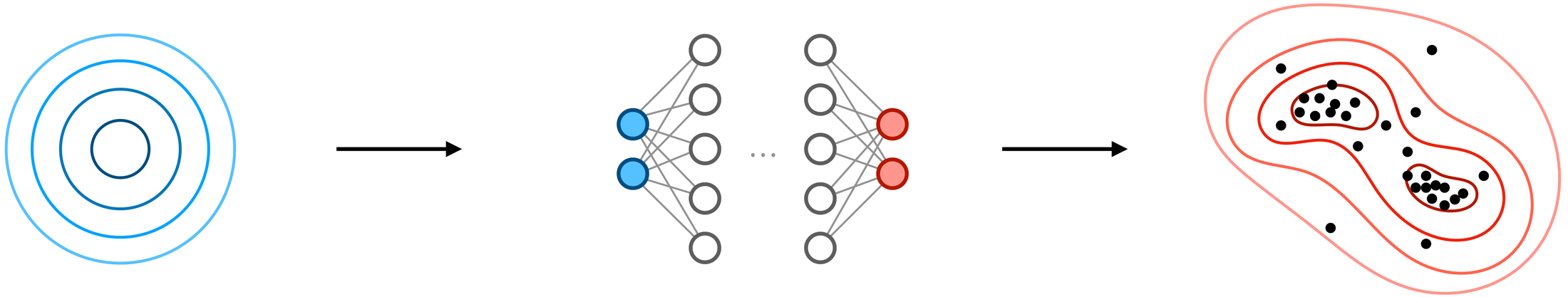
Target density is given by

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

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

[lots of recent great work including by J. Behrmann; 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

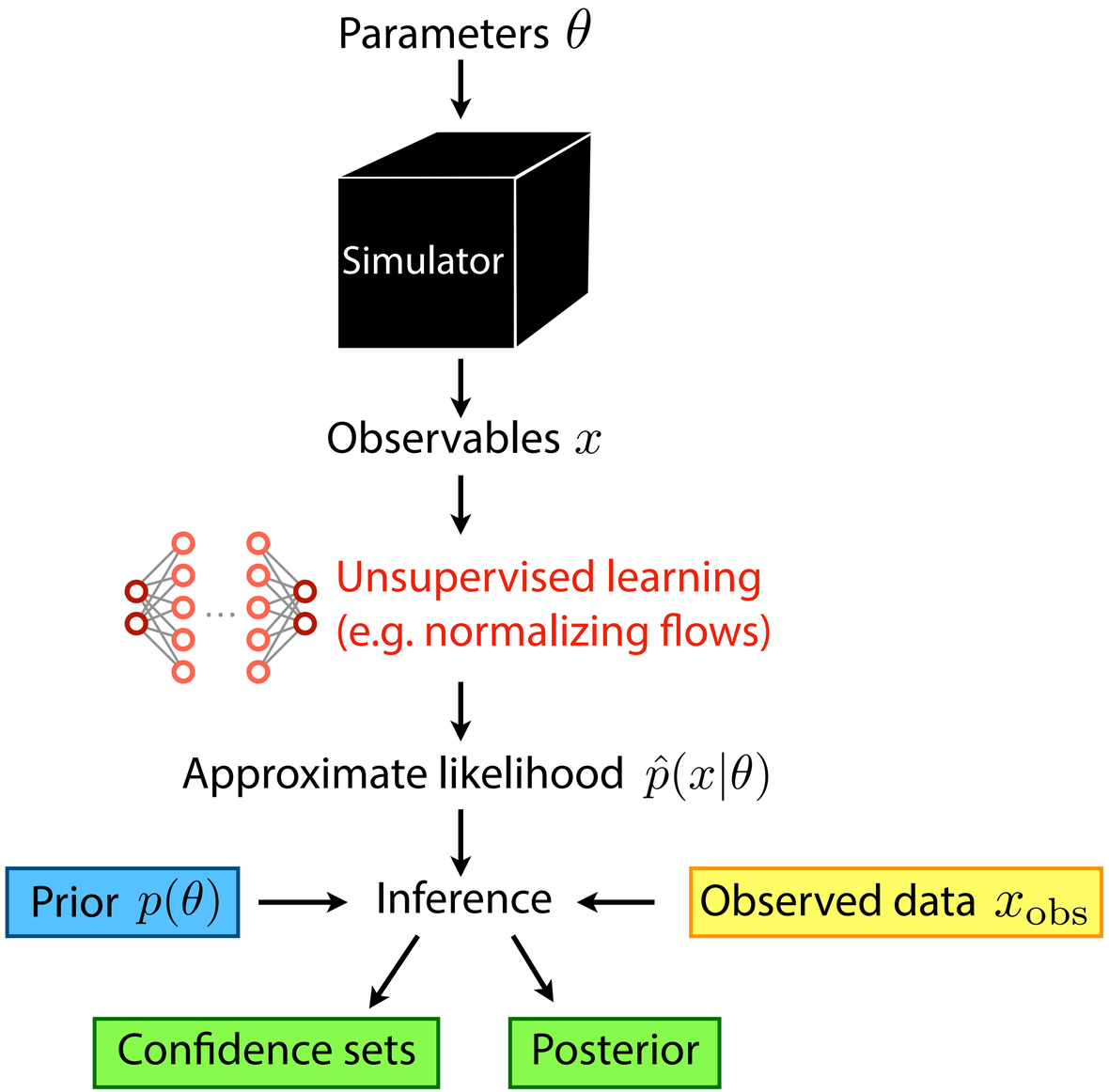
$\hat{p}(x) = \pi(f^{-1}(x)) \, |\det \nabla f|^{-1}$

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

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

[lots of recent great work including by J. Behrmann; 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]

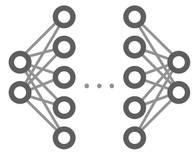
- Train conditional neural density estimator (e.g. normalizing flow) as tractable surrogate for simulator
- Scales well to high-dimensional data
  (no compression to summary stats necessary)
- Amortized: After upfront simulation + training phase, inference is efficient for new data or prior
- Alternative: learn posterior $\hat{p}(\theta|x_{\mathrm{obs}})$

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

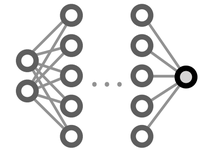
The likelihood ratio trick

- Remember GANs?

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



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

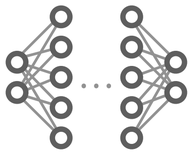


Discriminator

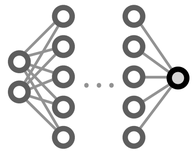
The likelihood ratio trick

- Remember GANs?

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



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



Discriminator

Discriminator learns decision function

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

The likelihood ratio trick

- Remember GAN's?

Generator
(intractable likelihood p_g

Examples
(intractable likelihood p_t

decision function

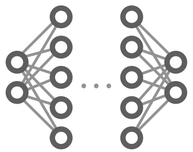
$$\text{true}(x)$$

$$) + p_{\text{true}}(x)$$

The likelihood ratio trick

- Remember GANs?

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



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

Discriminator

Discriminator learns decision function

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

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

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

θ_0 → Simulator

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

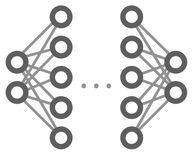
θ_1 → Simulator

Classifier

The likelihood ratio trick

- Remember GANs?

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

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

θ_0 → Simulator

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

θ_1 → Simulator

Classifier

Classifier learns decision function

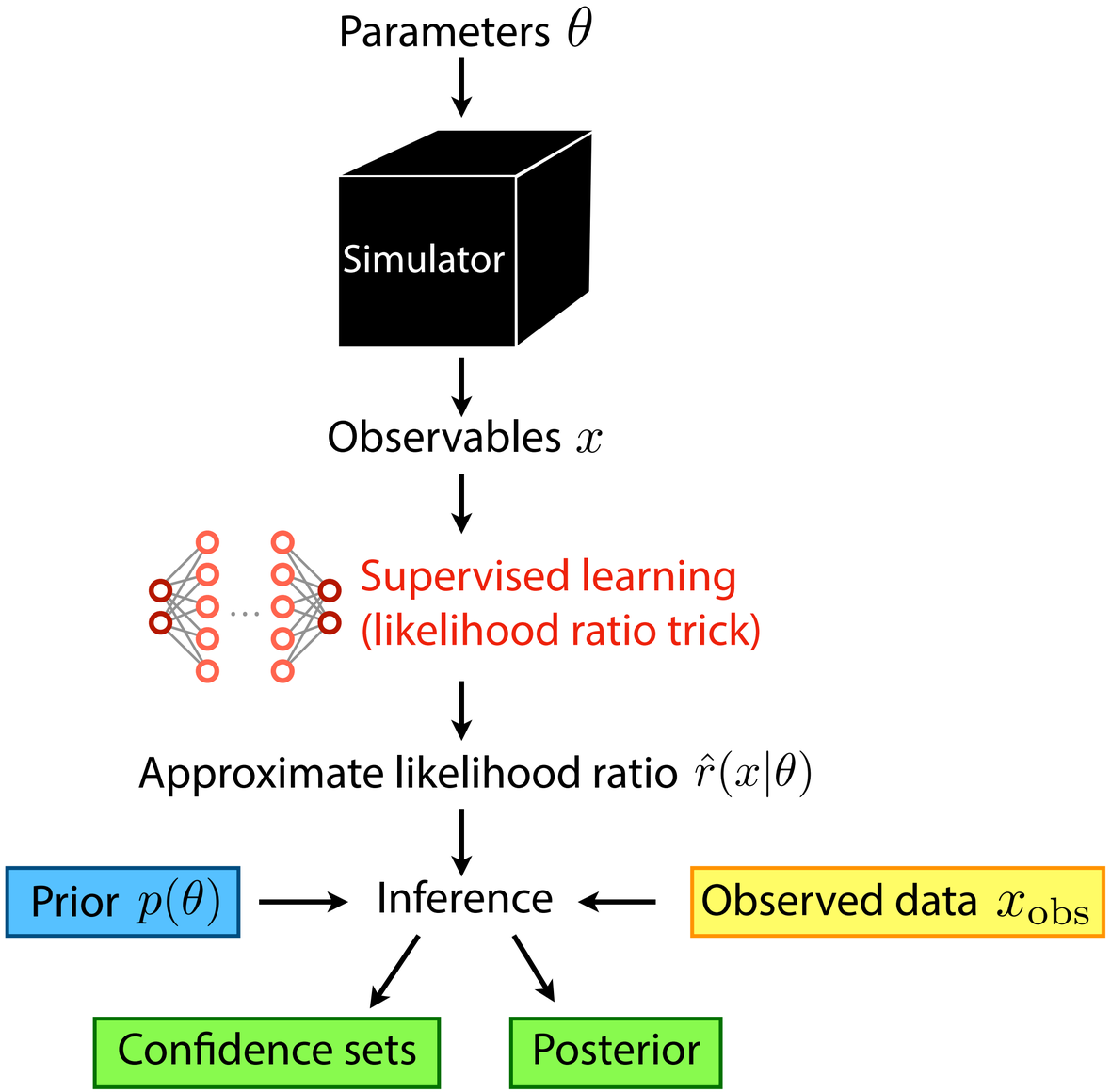
$$s(x) \to \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)}$$

⇒ Estimator for likelihood ratio

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

Inference by likelihood ratio trick

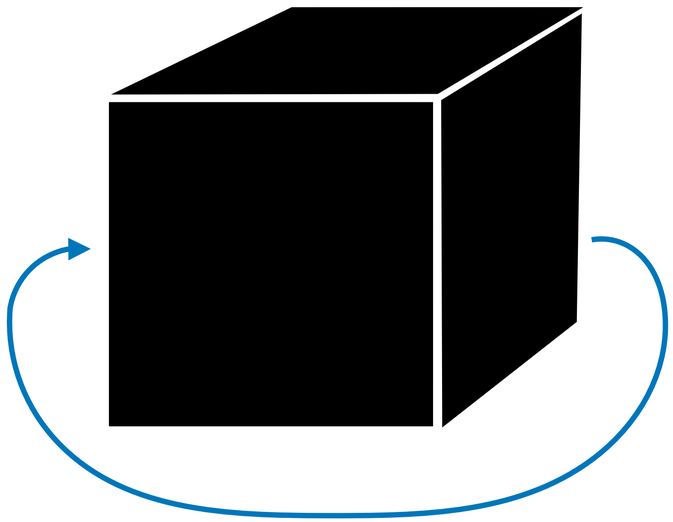
[K. Cranmer J. Pavez, G. Louppe 1506.02169]



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

Frontiers of simulation-based inference

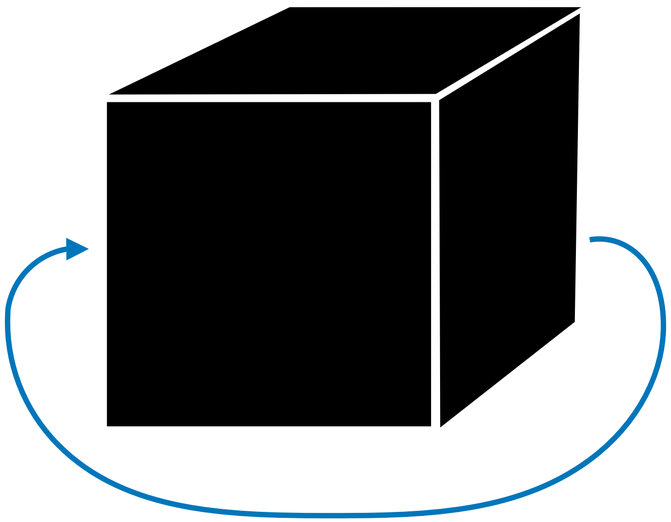
[K. Cranmer, JB, G. Louppe 1911.01429]



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

Frontiers of simulation-based inference

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



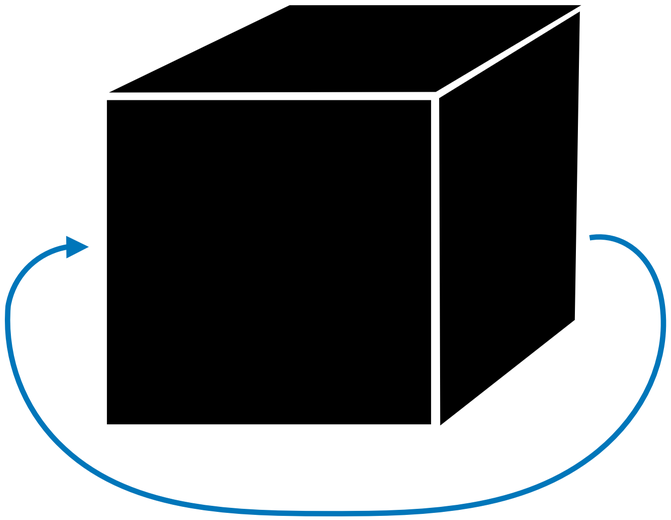
Active learning:
Iteratively guide simulator to important parameter regions based on past results

Mining gold:
Extract and leverage information from the simulator that characterizes the latent process

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

Frontiers of simulation-based inference

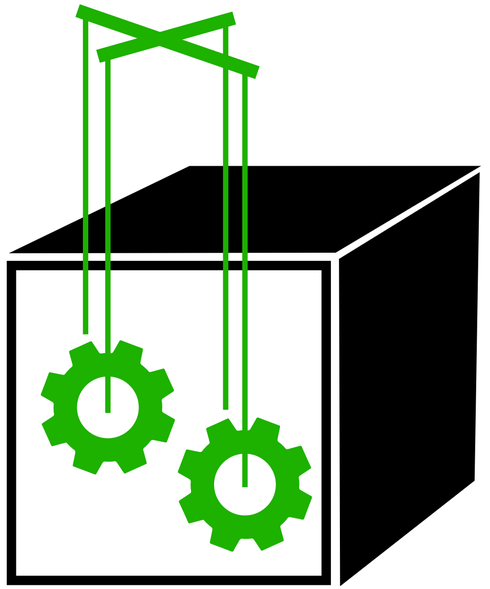
[K. Cranmer, JB, G. Louppe 1911.01429]



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

Mining gold:
Extract and leverage information from the simulator that characterizes the latent process

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

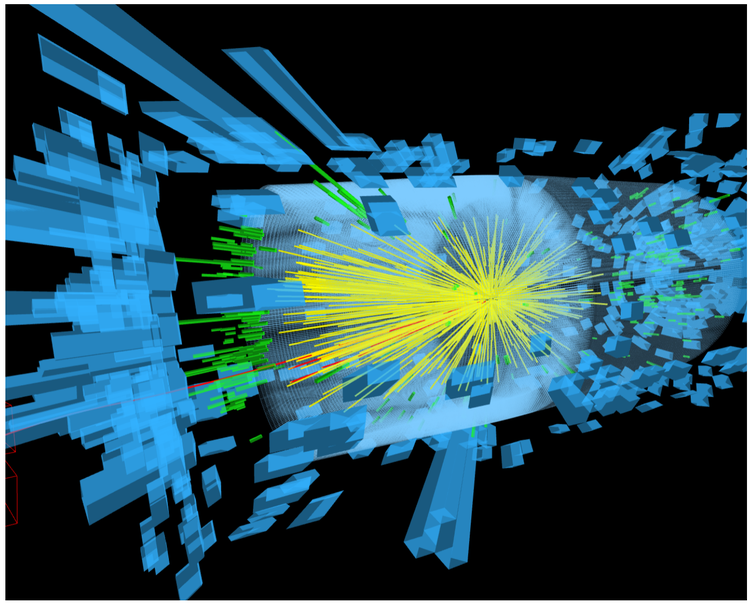


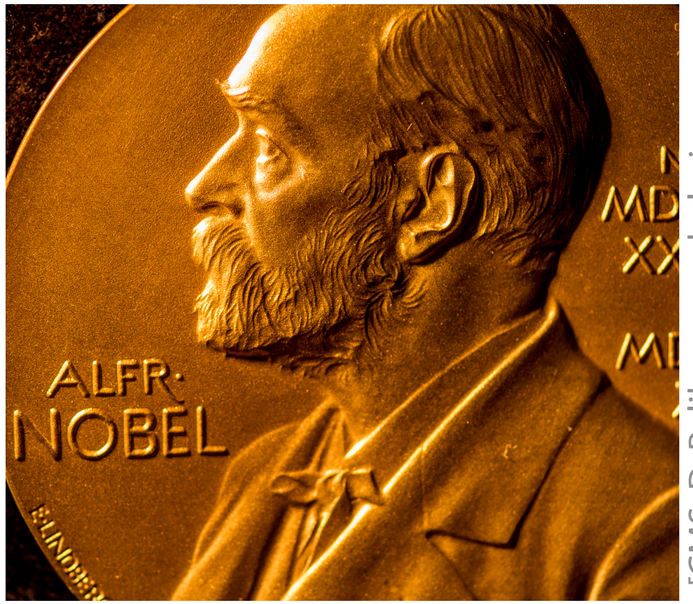
Probabilistic programming:
Explicitly write simulator as probabilistic program, with ability to condition execution trace on observations

These new methods let us extract more knowledge from particle collisions.

The Higgs boson

- Discovered 2012 as latest elementary particle, last one missing from the Standard Model



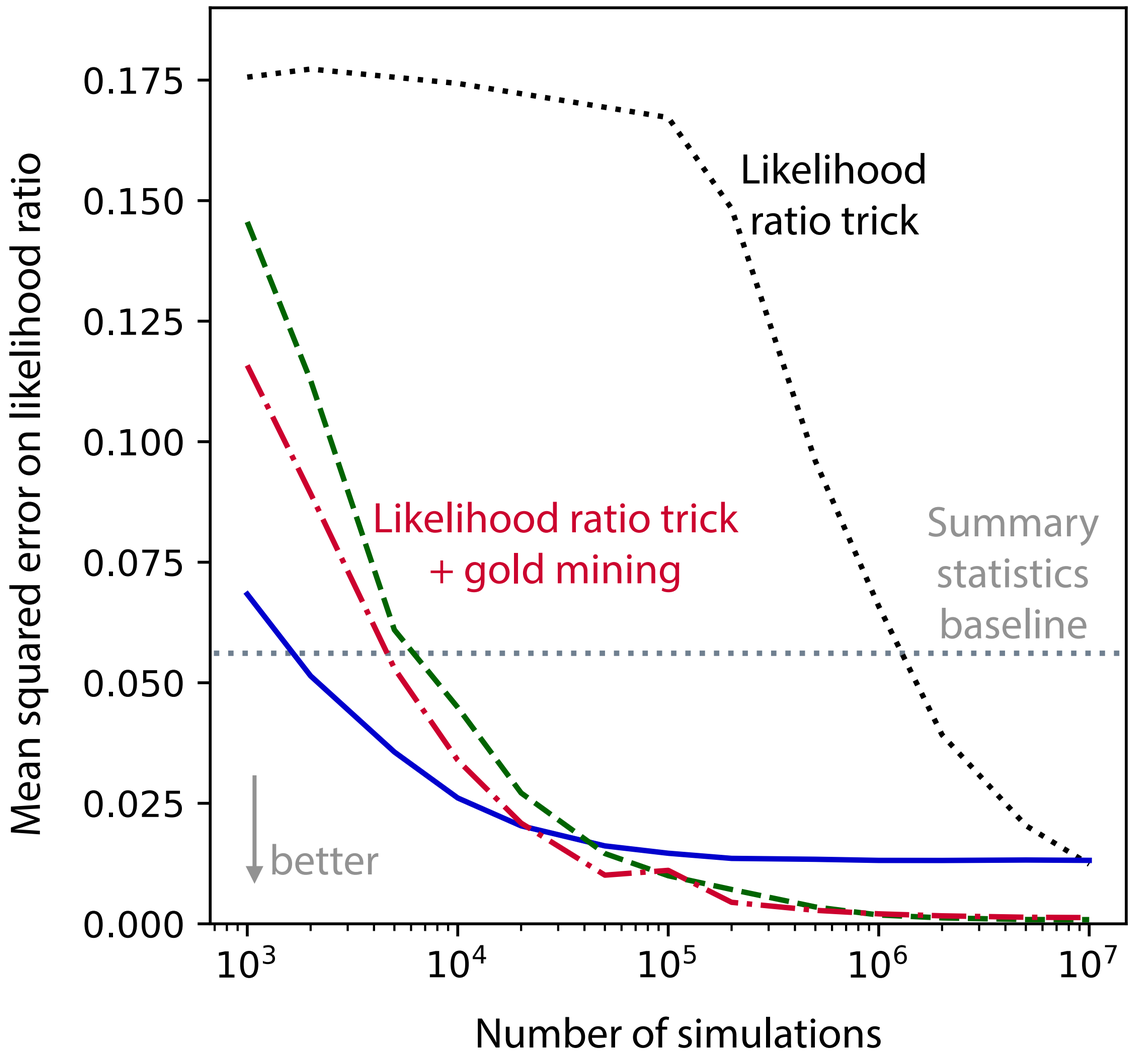


[CMS; D. Balibouse; nobelprize.org

- Measuring its properties might points us to answers to big questions
  - Why is gravity is so weak?
  - Why are some particles so light and others so heavy?
  - Why is there more matter than antimatter?
  - Is the vacuum stable?

- We analyzed Higgs properties on synthetic data sets with old and new inference techniques

Improving sample efficiency

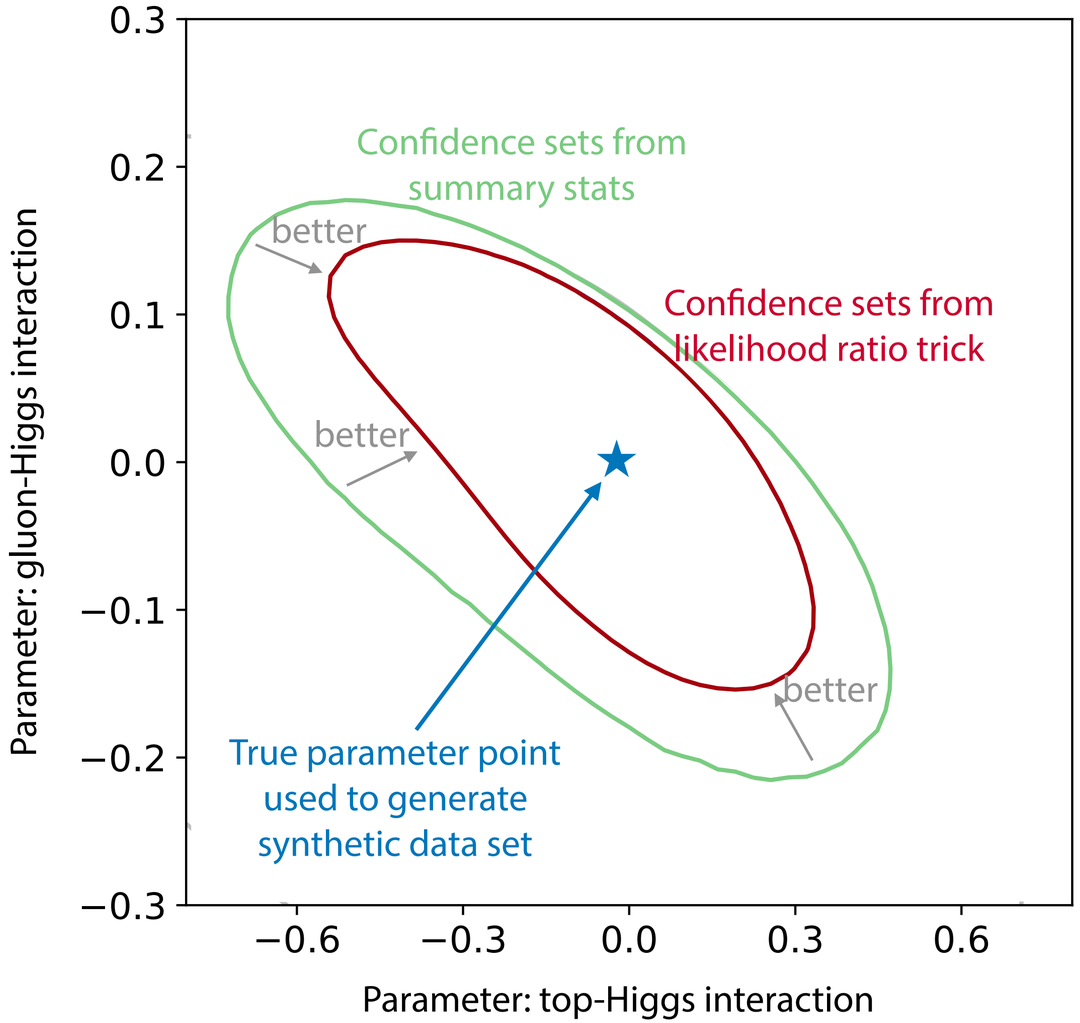


With enough training data, the ML algorithms get the likelihood function right.

Using more information from the simulator improves sample efficiency substantially.

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

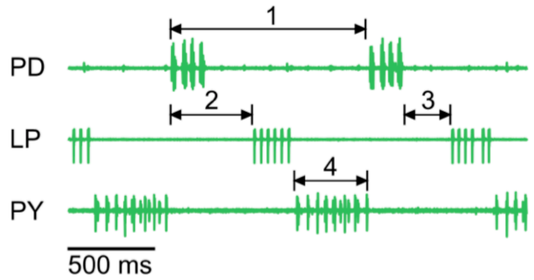
Improving quality of inference



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

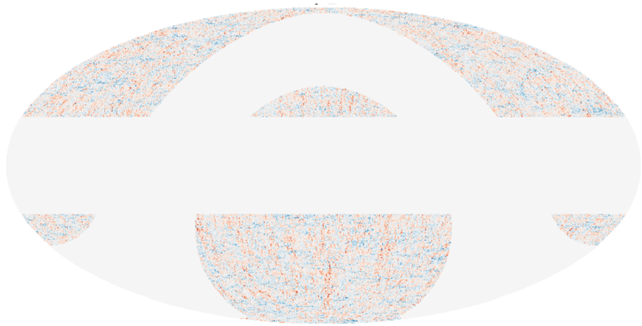
[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013; 1805.00020;
JB, F. Kling, I. Espejo, K. Cranmer 1907.10621]

Simulation-based inference is thriving



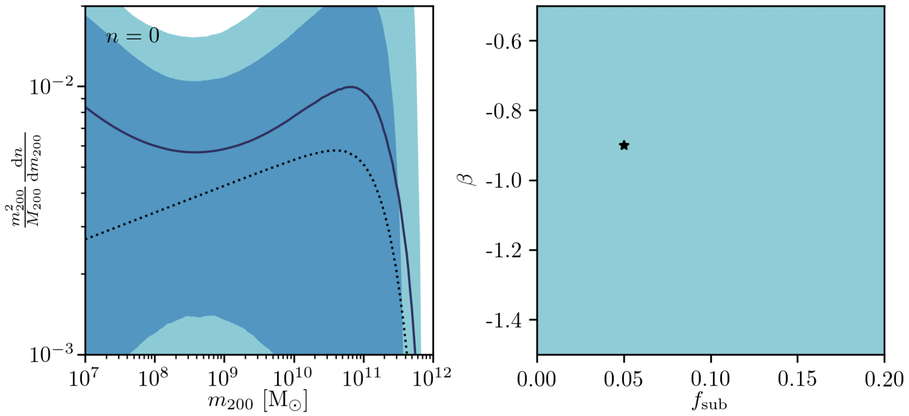
Neuroscience

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



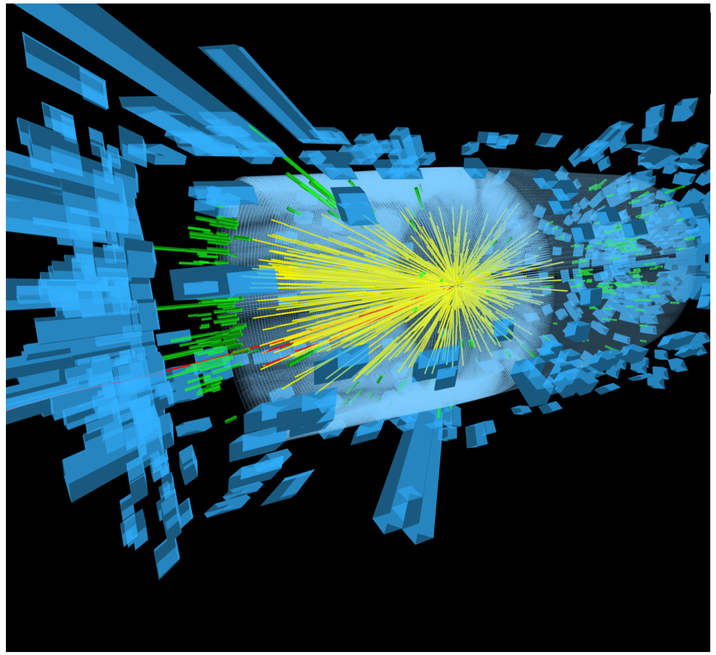
Cosmological large-scale structure

[P. Taylor et al. 1904.05364]

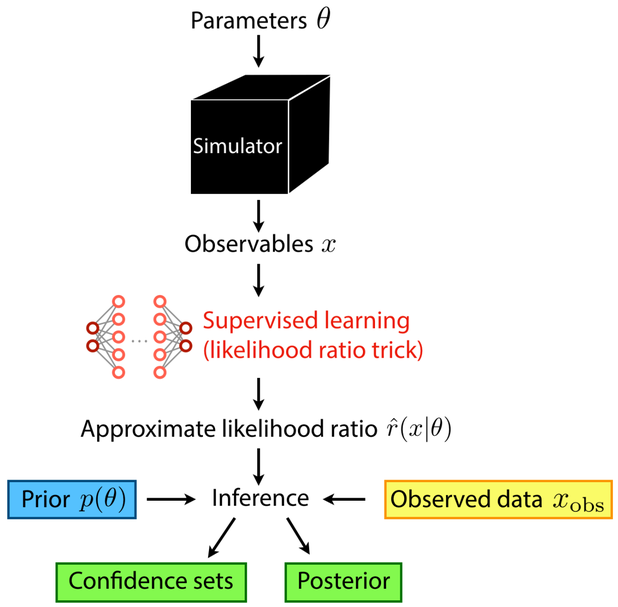


Strong gravitational lensing

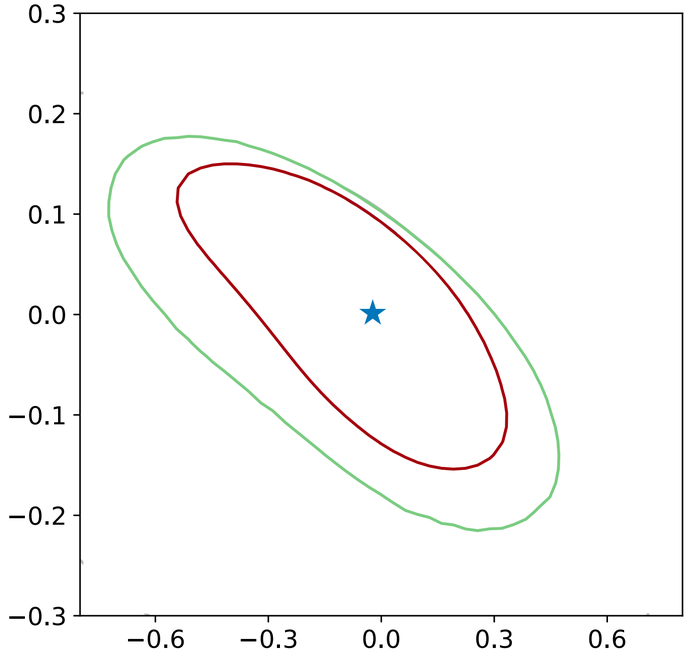
[JB, S. Mishra-Sharma et al. 1909.02005]



Scientific inference is challenging when phenomena are modeled with computer simulations



New inference algorithms are based on normalizing flows or the likelihood ratio trick



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

For a review, see K. Cranmer, JB, G. Louppe: "The frontier of simulation-based inference", arXiv:1911.01429

Bonus material

Modelling particle physics processes

Theory
parameters
θ

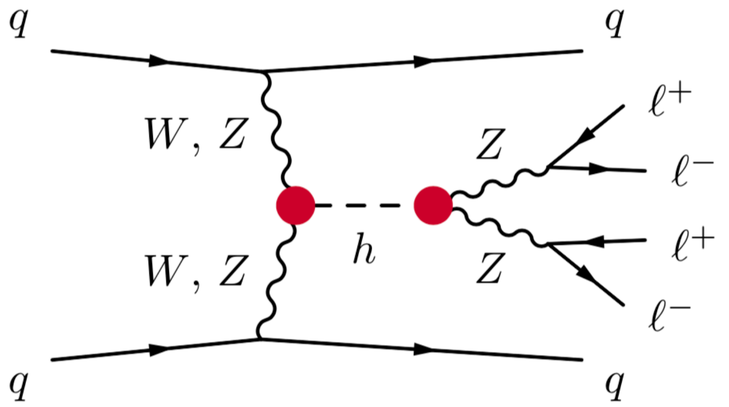
Evolution

Modelling particle physics processes

Latent variables

Parton-level momenta z_p

Theory parameters θ

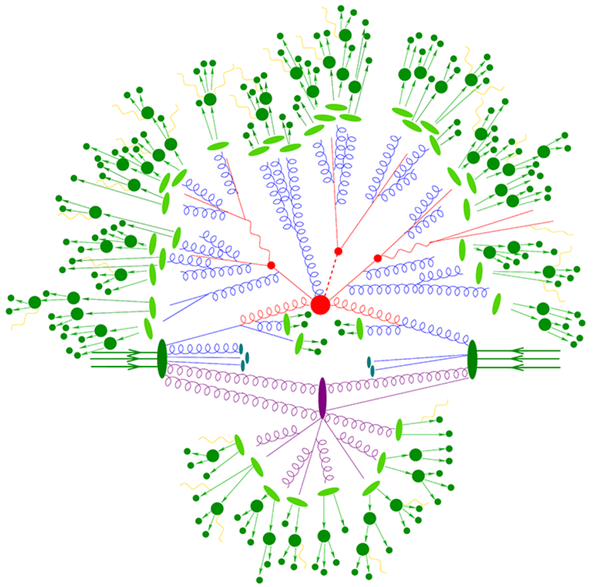


Evolution

Modelling particle physics processes

Latent variables

Shower splittings z_s ← Parton-level momenta z_p ← Theory parameters θ



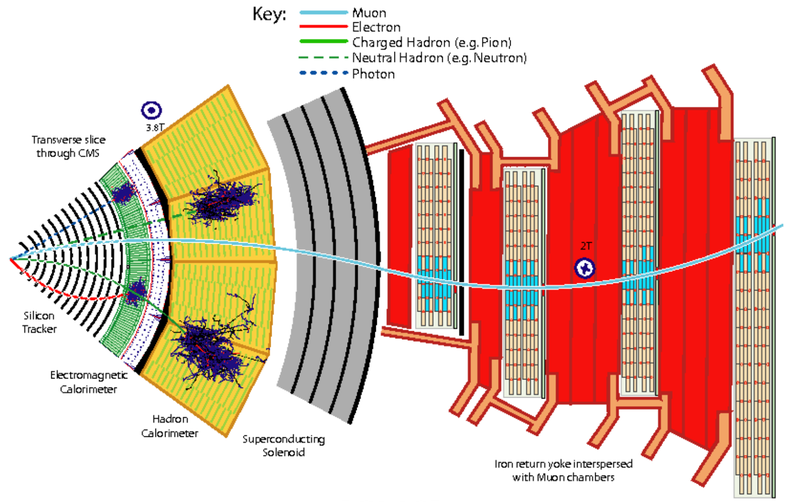
[F. Krauss]

Evolution

Modelling particle physics processes

Latent variables

Detector interactions	Shower splittings	Parton-level momenta	Theory parameters
z_d	z_s	z_p	θ



[CMS]

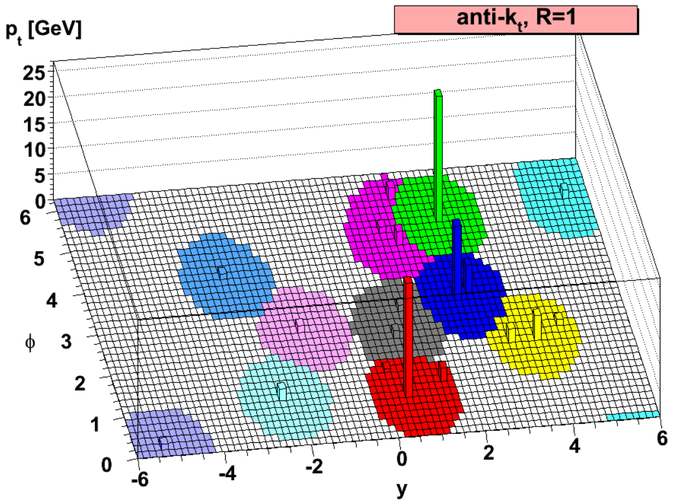
Evolution

Modelling particle physics processes

Latent variables

Observables | Detector interactions | Shower splittings | Parton-level momenta | Theory parameters

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



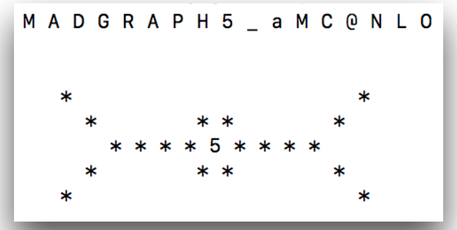
[M. Cacciari, G. Salam, G. Soyez 0802.1189]

Evolution

Modelling particle physics processes

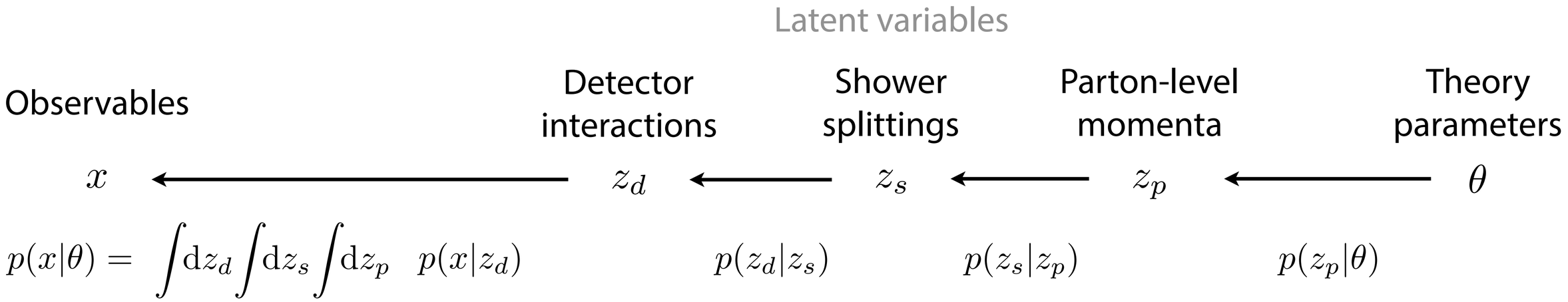
Latent variables

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



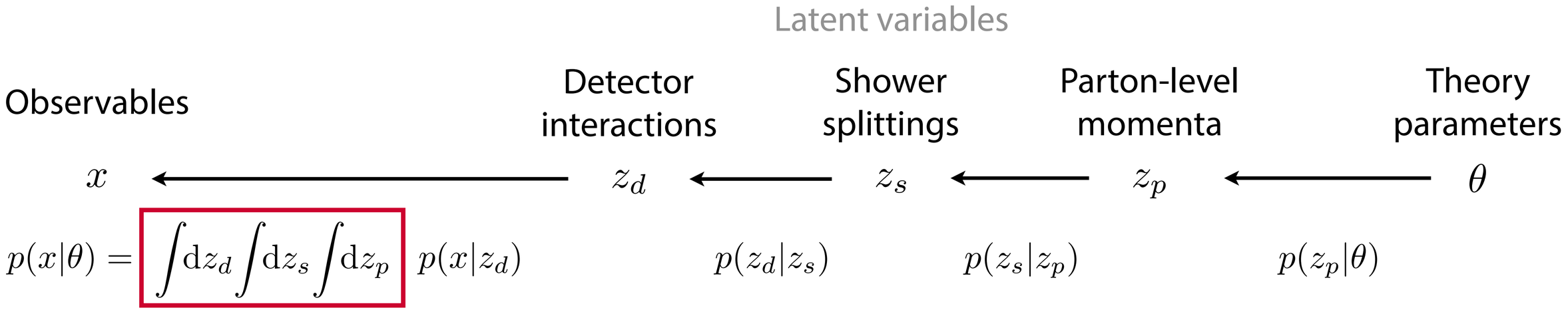
Prediction (simulation)

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.)

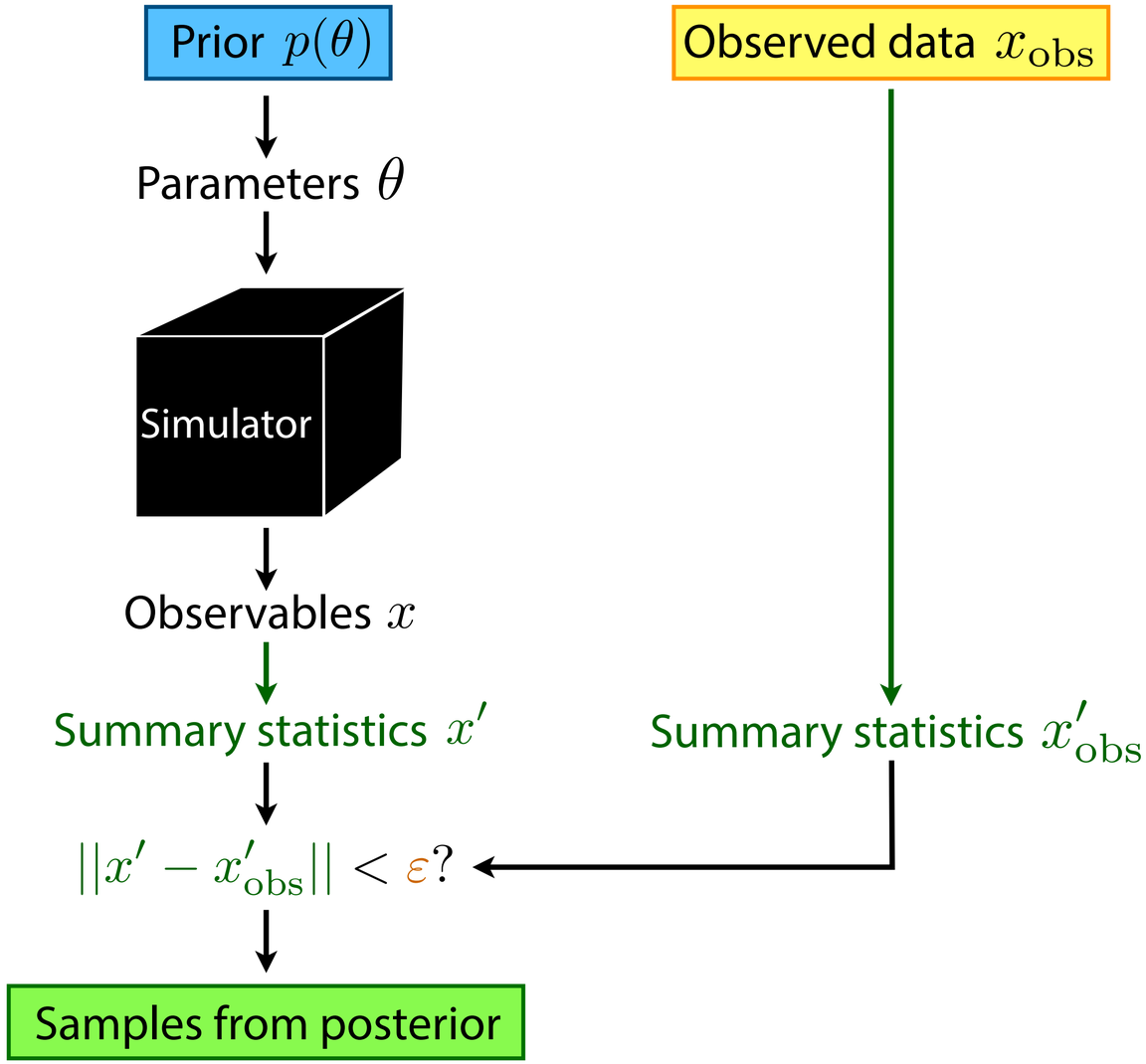
Inference

Simulation-based inference methods

	Use simulator directly during inference	Use surrogate model during inference
Any simulator	Approximate Bayesian Computation	Likelihood with summary stats Neural likelihood Neural posterior Neural likelihood ratio
Additional requirements	Probabilistic programming Exact Bayesian inference for diff. sim.	Gold mining

Approximate Bayesian Computation (ABC)

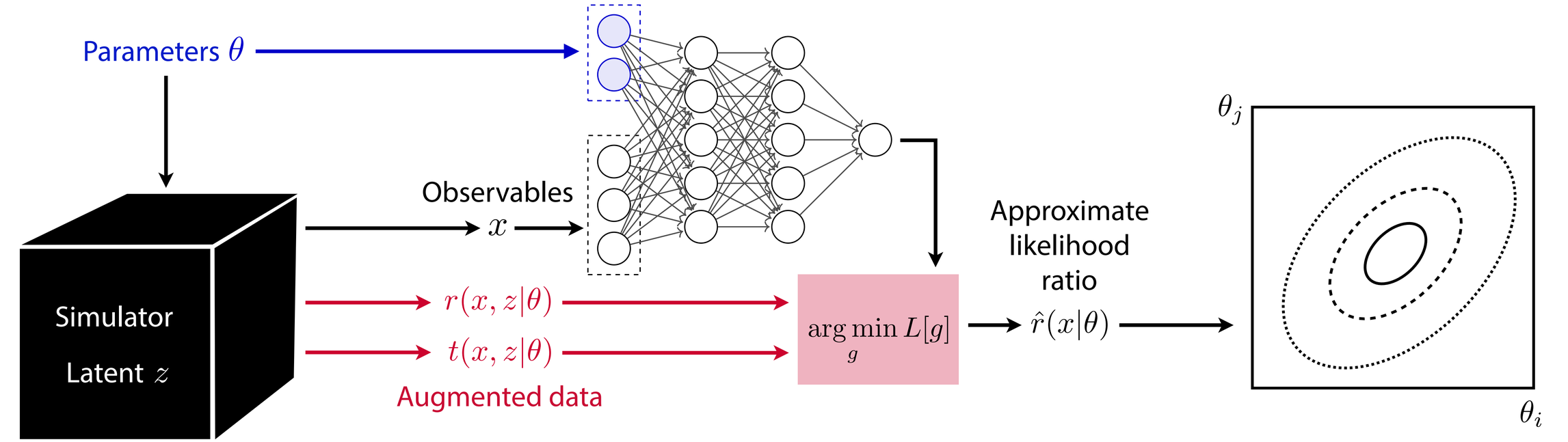
[D. Rubin 1984]



- Quality of inference:
  low-dimensional summary statistics x'
  and large bandwidths ε reduce
  statistical power

- Sample efficiency:
  high-dimensional summary statistics x'
  and small bandwidths ε require
  unfeasibly many simulations

Mining gold



1. Simulation

"Mining gold": Extract additional information from simulator

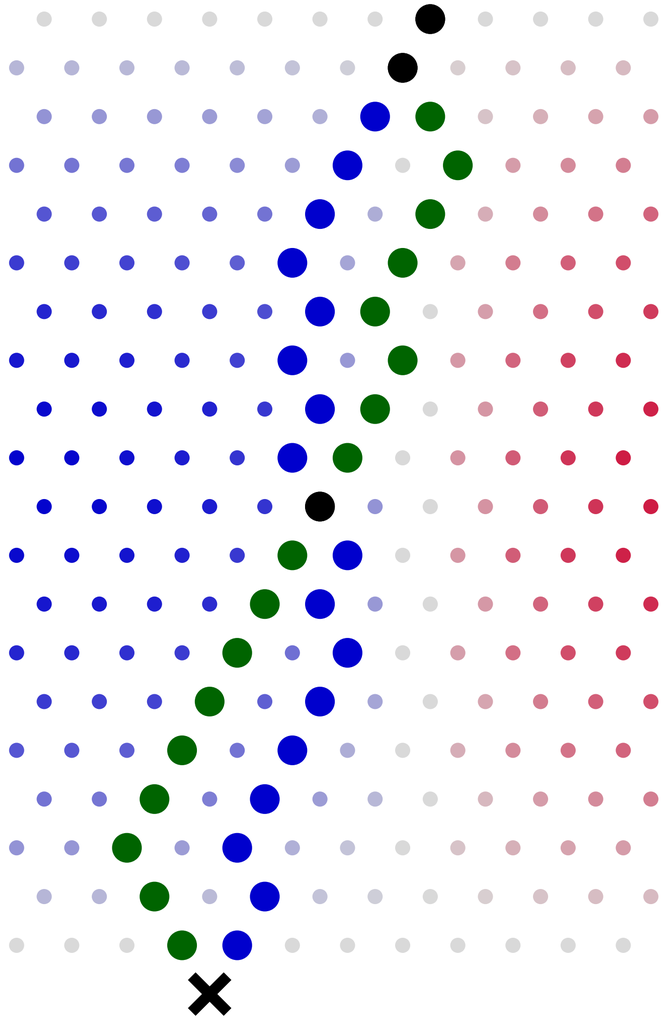
2. Machine Learning

Use this information to train estimator for likelihood ratio

3. Inference

Limit setting with standard hypothesis tests

Mining gold on the Galton board



- Remember: the likelihood

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

is intractable because of the integral over all possible paths z

- But: we can calculate the probability of each individual path

$$p(x, z|\theta) = \prod_{\text{nails } i} p_i(x, z|\theta)$$

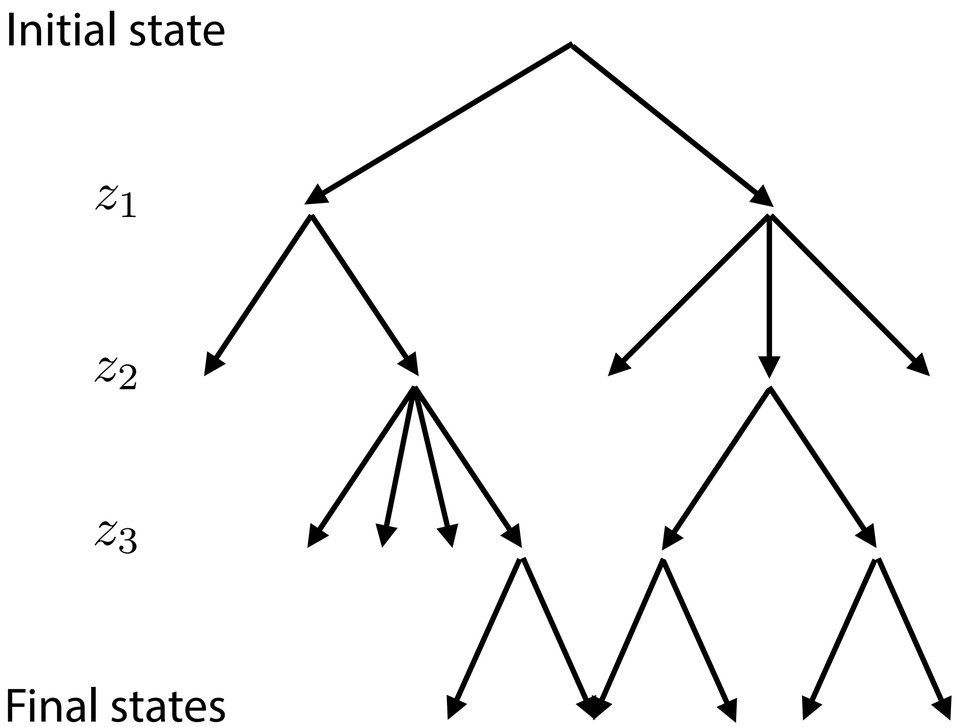
and the "joint likelihood ratio" conditional on a particular path

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

Generalizing this idea

- 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()
```



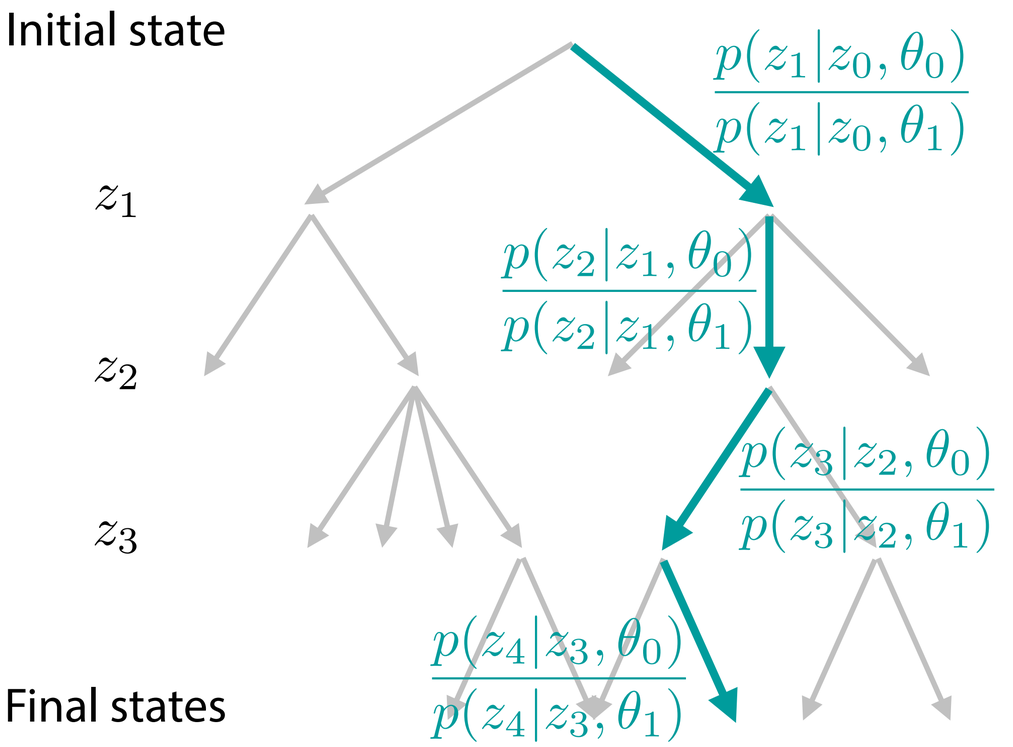
Generalizing this idea

- 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":

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



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 simulation, including all latent variables, for θ_0 compared to θ_1?")

We want the likelihood ratio function

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

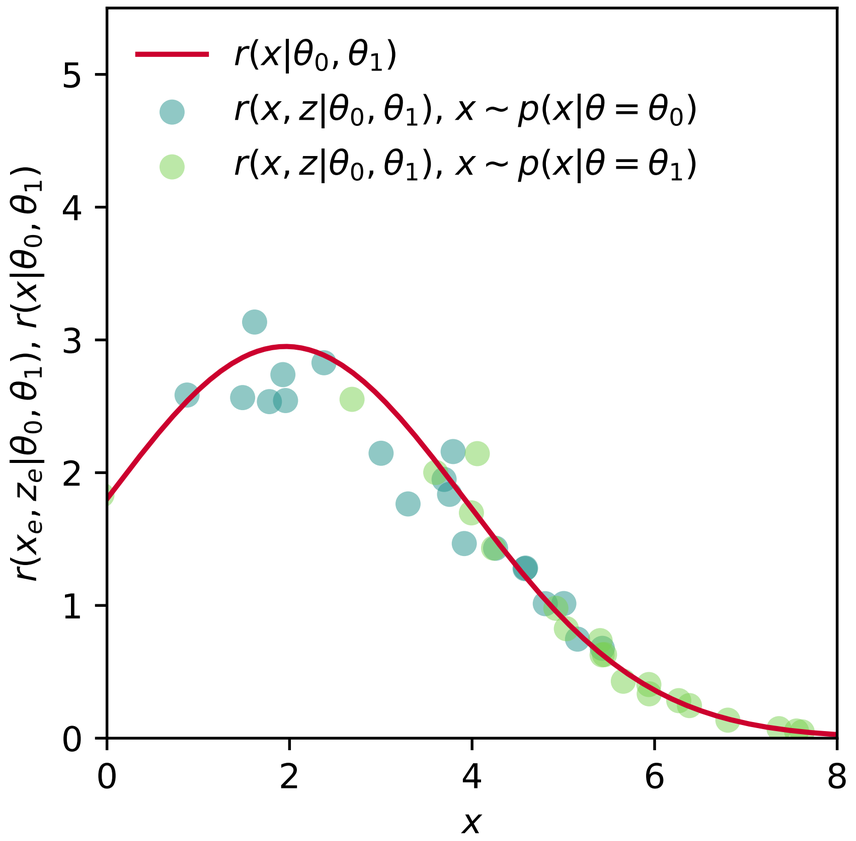
("How much more likely is the observation x for θ_0 compared to θ_1?")

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

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

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

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

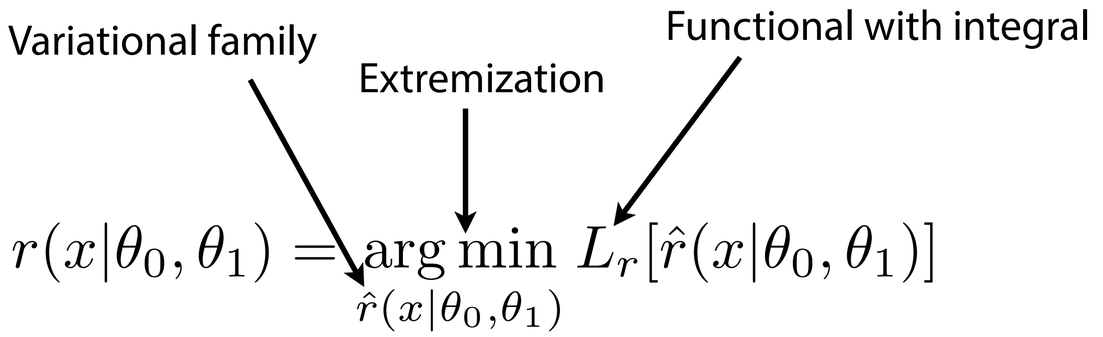
(And we can sample from $p(x, z|\theta)$ by running the simulator.)

We want the likelihood ratio function

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

Machine learning = applied calculus of variatios

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



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

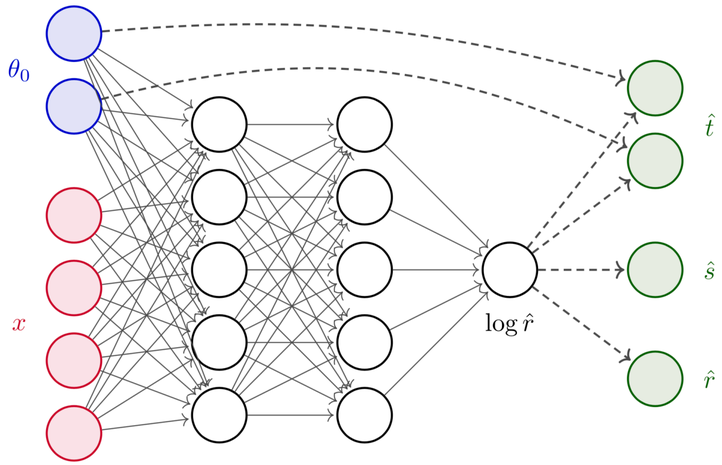
$$r(x|\theta_0,\theta_1) = \underset{\hat{r}(x|\theta_0,\theta_1)}{\arg\min}\ L_r[\hat{r}(x|\theta_0,\theta_1)]$$

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

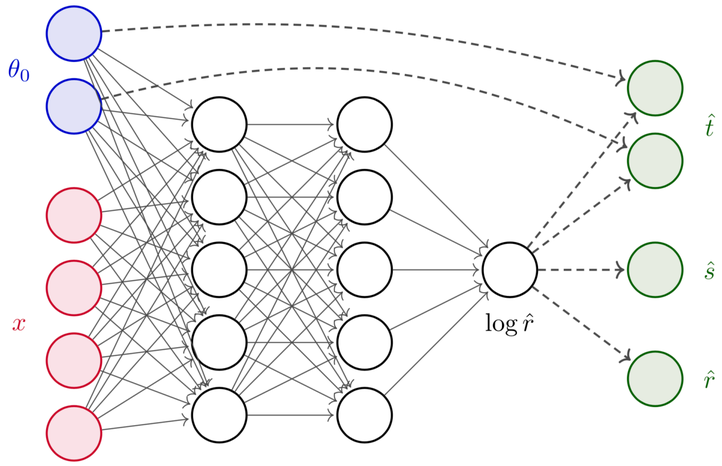
$$r(x|\theta_0,\theta_1) = \underset{\hat{r}(x|\theta_0,\theta_1)}{\arg\min}\ L_r[\hat{r}(x|\theta_0,\theta_1)]$$

This is where machine learning comes in!

Neural network

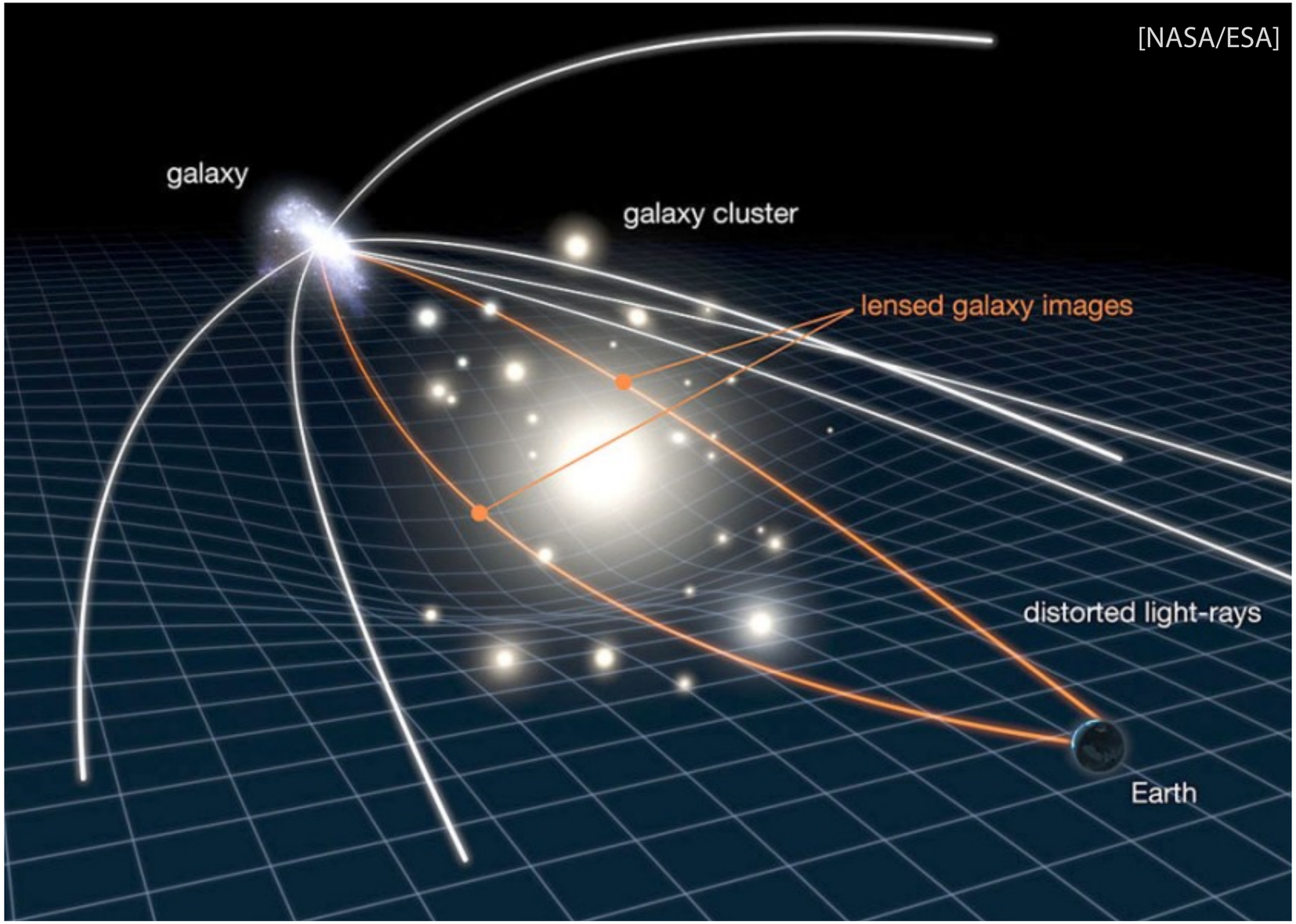
Stochastic gradient descent

Loss function with
finite sum over samples

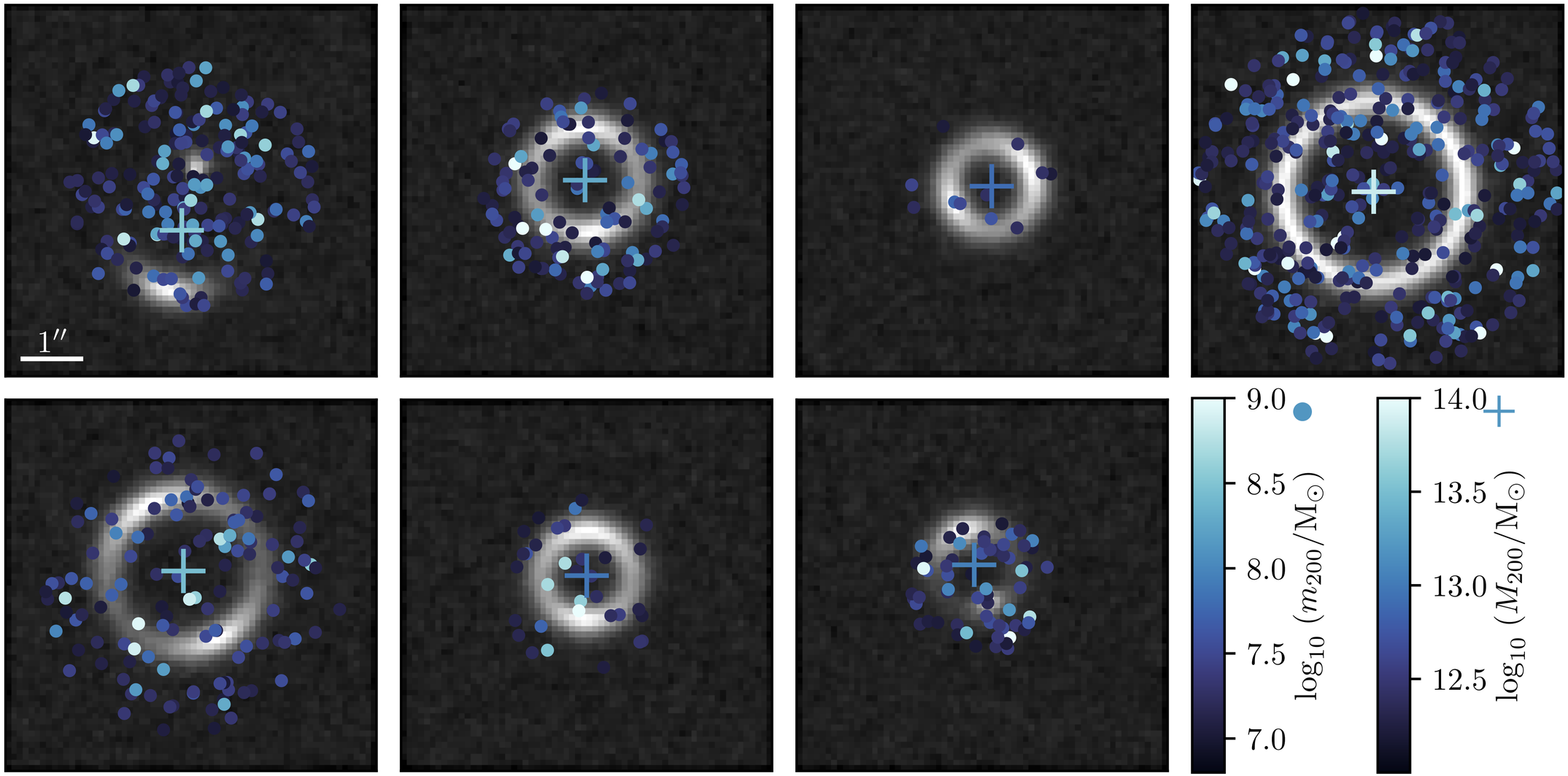


A sufficiently expressive neural network
efficiently trained in this way
with enough data will learn
the likelihood ratio function $r(x|\theta_0,\theta_1)$!

Strong gravitational lensing



Lensing simulations



Mining for Dark Matter substructure