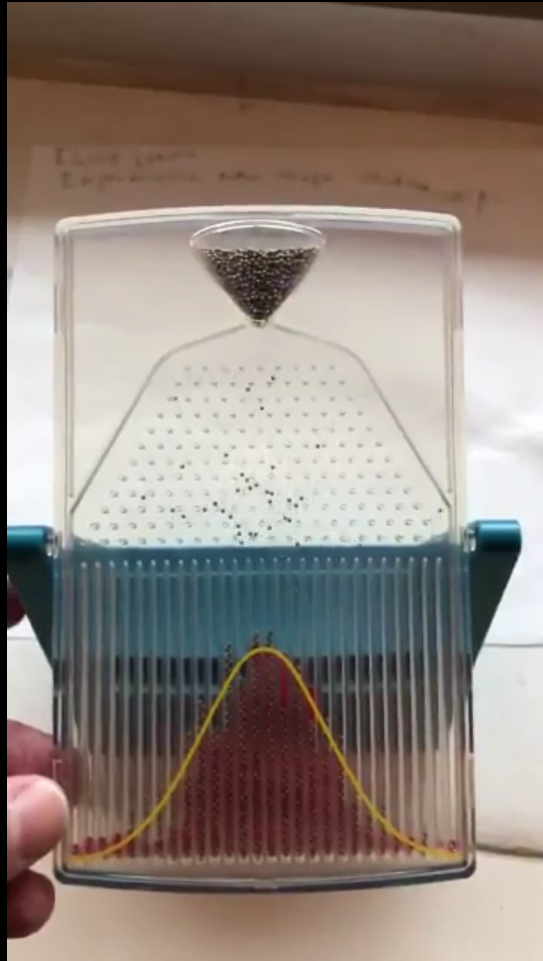
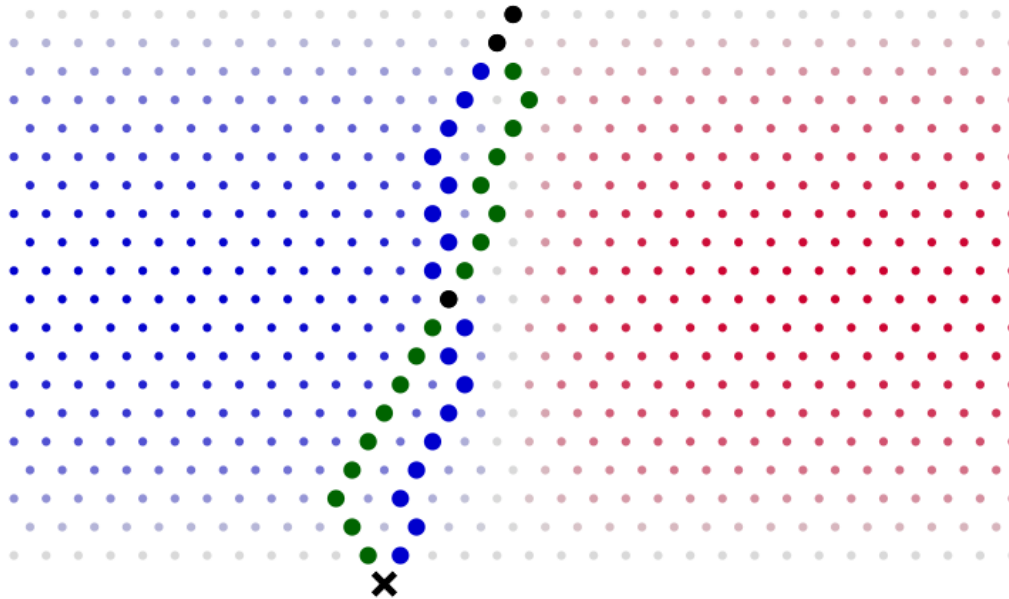


Neural Likelihood-free Inference

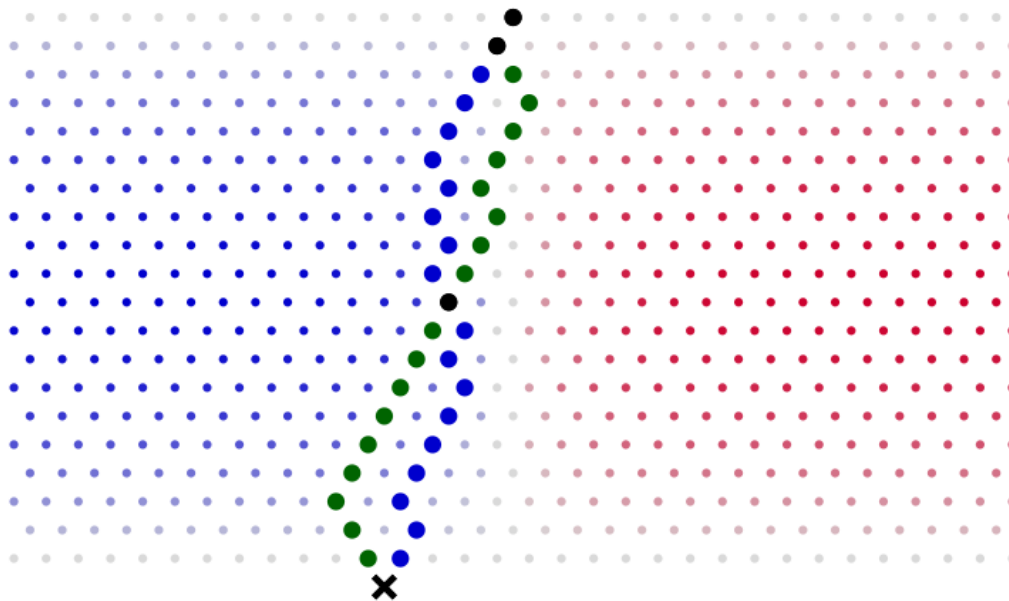
Machine Learning and Physics series
October 31, Oxford

Gilles Louppe
g.louppe@uliege.be





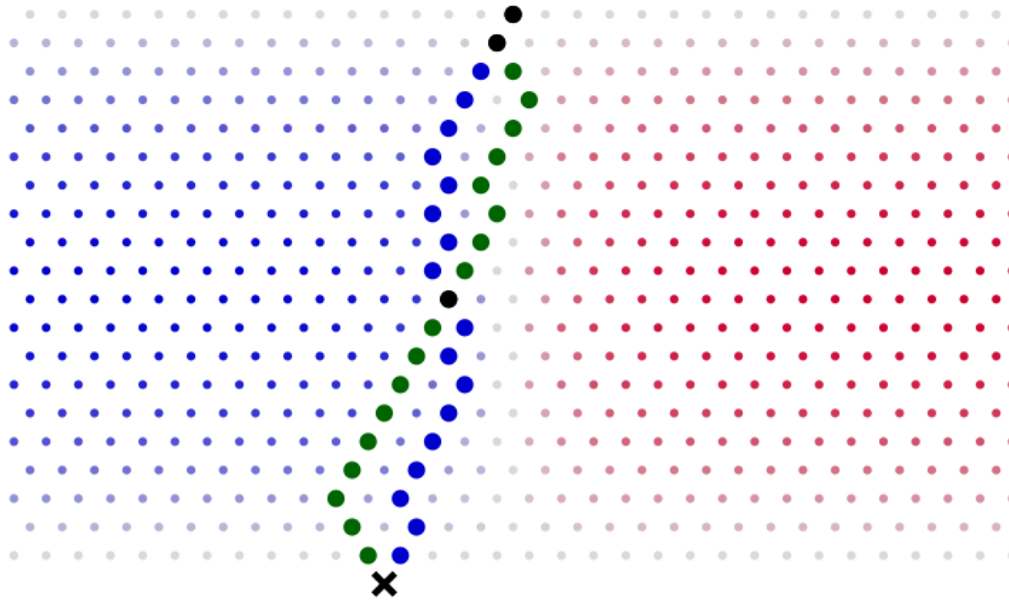
How to estimate the probability θ of going left when hitting a pin?



The probability of ending in bin x corresponds to the total probability of all the paths z from start to x ,

$$p(x|\theta) = \int p(x, z|\theta) dz = \binom{n}{x} \theta^x (1 - \theta)^{n-x}.$$

Therefore $\hat{\theta} = \arg \max \prod_{x_i} p(x_i|\theta)\pi(\theta)$.

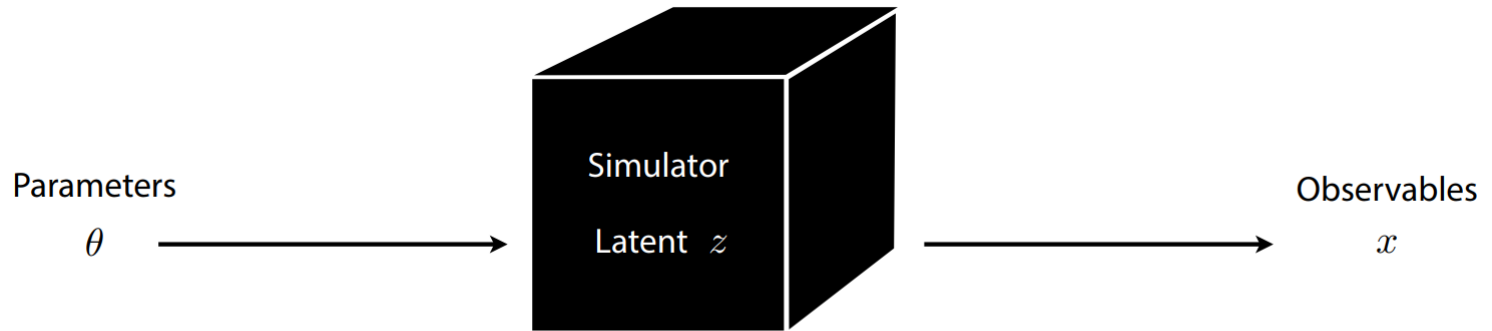


But what if we shift or remove some of the pins?

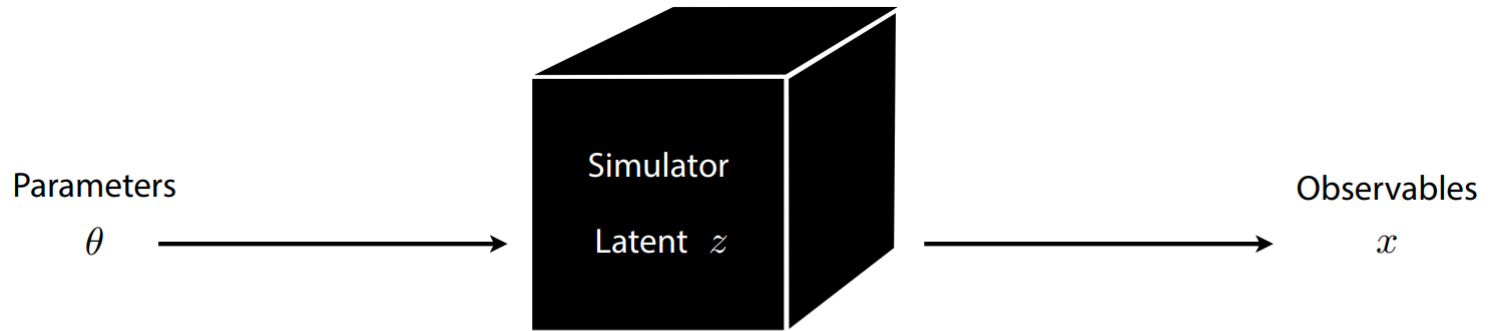
The Galton board is a **metaphore** of simulation-based science:

Galton board device	→	Computer simulation
Parameters θ	→	Model parameters θ
Buckets x	→	Observables x
Random paths z	→	Latent variables z (stochastic execution traces through simulator)

Inference in this context requires **likelihood-free algorithms**.



- Prediction:
- Well-understood mechanistic model
 - Simulator can generate samples



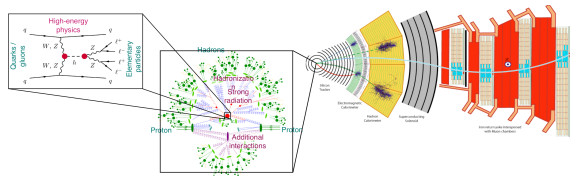
Prediction:

- Well-understood mechanistic model
- Simulator can generate samples

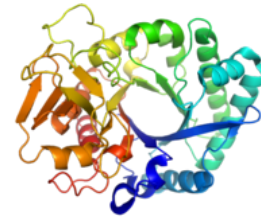
Inference:

- Likelihood function $p(x|\theta)$ is intractable
- Inference based on estimator $\hat{p}(x|\theta)$

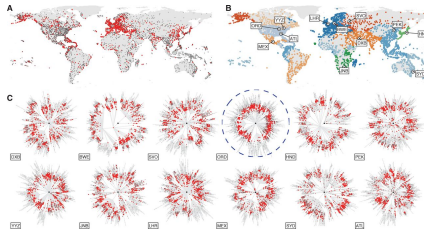
A thriving field of research



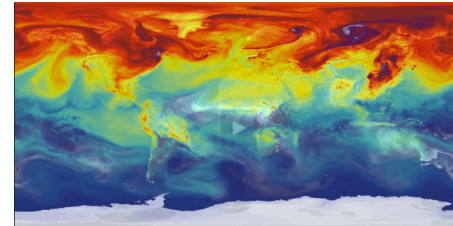
Particle physics



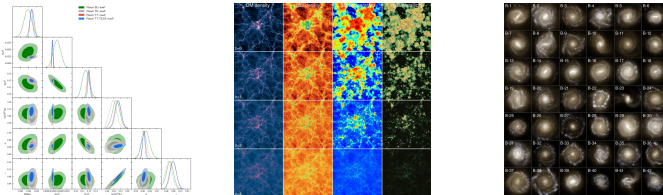
Protein folding



Epidemiology



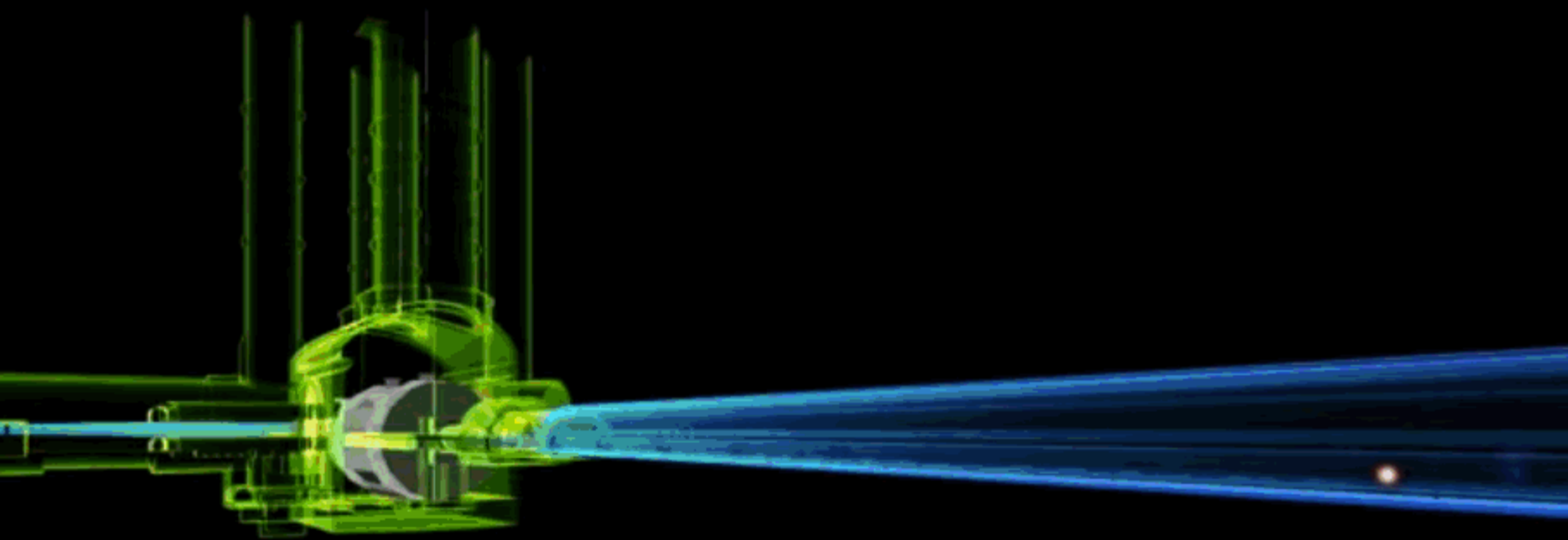
Climate science

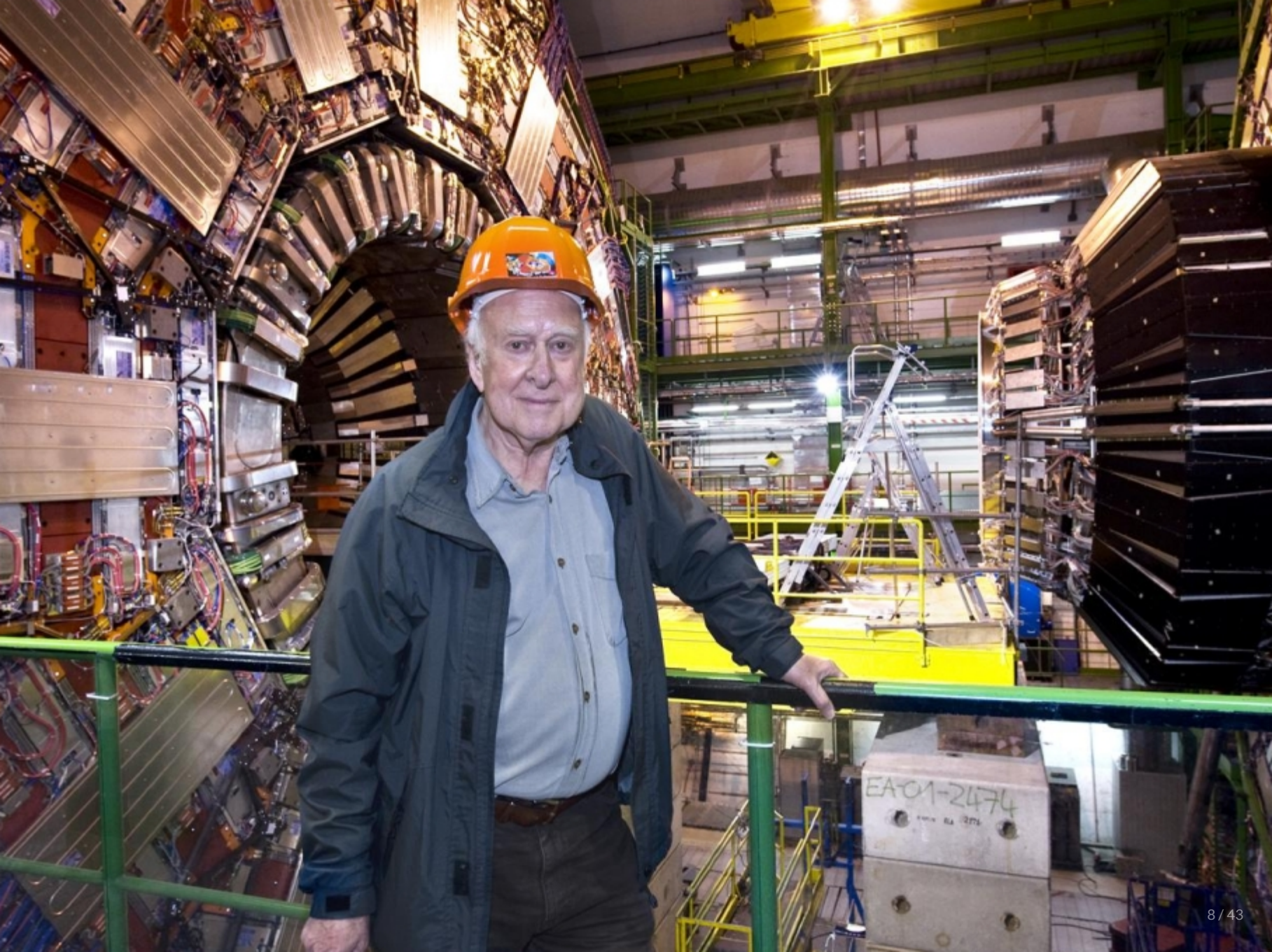


Astrophysics and cosmology

(... and many others!)

$$\begin{aligned}
\mathcal{L}_{SM} = & -\frac{1}{2}\partial_\nu g_\mu^a \partial_\nu g_\mu^a - g_s f^{abc} \partial_\mu g_\nu^a g_\mu^b g_\nu^c - \frac{1}{4}g_s^2 f^{abc} f^{ade} g_\mu^b g_\nu^c g_\mu^d g_\nu^e - \partial_\nu W_\mu^+ \partial_\nu W_\mu^- - \\
& M^2 W_\mu^+ W_\mu^- - \frac{1}{2}\partial_\nu Z_\mu^0 \partial_\nu Z_\mu^0 - \frac{1}{2c_w^2} M^2 Z_\mu^0 Z_\mu^0 - \frac{1}{2}\partial_\mu A_\nu \partial_\mu A_\nu - ig_{c_w} (\partial_\nu Z_\mu^0 (W_\mu^+ W_\nu^- - \\
& W_\nu^+ W_\mu^-) - Z_\mu^0 (W_\nu^+ \partial_\nu W_\mu^- - W_\mu^- \partial_\nu W_\nu^+) + Z_\mu^0 (W_\nu^+ \partial_\nu W_\mu^- - W_\mu^- \partial_\nu W_\nu^+)) - \\
& ig_{s_w} (\partial_\nu A_\mu (W_\mu^+ W_\nu^- - W_\nu^+ W_\mu^-) - A_\nu (W_\mu^+ \partial_\nu W_\mu^- - W_\mu^- \partial_\nu W_\mu^+) + A_\mu (W_\nu^+ \partial_\nu W_\mu^- - \\
& W_\mu^- \partial_\nu W_\nu^+)) - \frac{1}{2}g^2 W_\mu^+ W_\nu^- W_\nu^+ W_\mu^- + \frac{1}{2}g^2 W_\mu^+ W_\nu^- W_\mu^+ W_\nu^- + g^2 c_w^2 (Z_\mu^0 W_\nu^+ Z_\nu^0 W_\mu^- - \\
& Z_\mu^0 Z_\nu^0 W_\nu^+ W_\mu^-) + g^2 s_w^2 (A_\mu W_\nu^+ A_\nu W_\mu^- - A_\mu A_\nu W_\nu^+ W_\mu^-) + g^2 s_w c_w (A_\mu Z_\nu^0 (W_\mu^+ W_\nu^- - \\
& W_\nu^+ W_\mu^-) - 2A_\mu Z_\mu^0 W_\nu^+ W_\nu^-) - \frac{1}{2}\partial_\mu H \partial_\mu H - 2M^2 \alpha_h H^2 - \partial_\mu \phi^+ \partial_\mu \phi^- - \frac{1}{2}\partial_\mu \phi^0 \partial_\mu \phi^0 - \\
& \beta_h \left(\frac{2M^2}{g^2} + \frac{2M}{g} H + \frac{1}{2}(H^2 + \phi^0 \phi^0 + 2\phi^+ \phi^-) \right) + \frac{2M^4}{g^2} \alpha_h - \\
& g\alpha_h M (H^3 + H\phi^0 \phi^0 + 2H\phi^+ \phi^-) - \\
& \frac{1}{8}g^2 \alpha_h (H^4 + (\phi^0)^4 + 4(\phi^+ \phi^-)^2 + 4(\phi^0)^2 \phi^+ \phi^- + 4H^2 \phi^+ \phi^- + 2(\phi^0)^2 H^2) - \\
& gMW_\mu^+ W_\mu^- H - \frac{1}{2}g \frac{M}{c_w^2} Z_\mu^0 Z_\mu^0 H - \\
& \frac{1}{2}ig (W_\mu^+ (\phi^0 \partial_\mu \phi^- - \phi^- \partial_\mu \phi^0) - W_\mu^- (\phi^0 \partial_\mu \phi^+ - \phi^+ \partial_\mu \phi^0)) + \\
& \frac{1}{2}g (W_\mu^+ (H \partial_\mu \phi^- - \phi^- \partial_\mu H) + W_\mu^- (H \partial_\mu \phi^+ - \phi^+ \partial_\mu H)) + \frac{1}{2}g \frac{1}{c_w} (Z_\mu^0 (H \partial_\mu \phi^0 - \phi^0 \partial_\mu H) + \\
& M (\frac{1}{c_w} Z_\mu^0 \partial_\mu \phi^0 + W_\mu^+ \partial_\mu \phi^- + W_\mu^- \partial_\mu \phi^+) - ig \frac{2s_w}{c_w} M Z_\mu^0 (W_\mu^+ \phi^- - W_\mu^- \phi^+) + ig_{s_w} M A_\mu (W_\mu^+ \phi^- - \\
& W_\mu^- \phi^+) - ig \frac{1-2c_w^2}{2c_w} Z_\mu^0 (\phi^+ \partial_\mu \phi^- - \phi^- \partial_\mu \phi^+) + ig_{s_w} A_\mu (\phi^+ \partial_\mu \phi^- - \phi^- \partial_\mu \phi^+) - \\
& \frac{1}{4}g^2 W_\mu^+ W_\mu^- (H^2 + (\phi^0)^2 + 2\phi^+ \phi^-) - \frac{1}{8}g^2 \frac{1}{c_w^2} Z_\mu^0 Z_\mu^0 (H^2 + (\phi^0)^2 + 2(2s_w^2 - 1)^2 \phi^+ \phi^-) - \\
& \frac{1}{2}g^2 \frac{2s_w}{c_w} Z_\mu^0 \phi^0 (W_\mu^+ \phi^- + W_\mu^- \phi^+) - \frac{1}{2}ig^2 \frac{2s_w}{c_w} Z_\mu^0 H (W_\mu^+ \phi^- - W_\mu^- \phi^+) + \frac{1}{2}g^2 s_w A_\mu \phi^0 (W_\mu^+ \phi^- + \\
& W_\mu^- \phi^+) + \frac{1}{2}ig^2 s_w A_\mu H (W_\mu^+ \phi^- - W_\mu^- \phi^+) - g^2 \frac{2s_w}{c_w} (2c_w^2 - 1) Z_\mu^0 A_\mu \phi^+ \phi^- - \\
& g^2 s_w^2 A_\mu A_\nu \phi^+ \phi^- + \frac{1}{2}ig_s \lambda_{ij}^a (\bar{q}_i^\alpha \gamma^\mu q_j^\alpha) g_\mu^a - \bar{e}^\lambda (\gamma \partial + m_e^\lambda) e^\lambda - \bar{\nu}^\lambda (\gamma \partial + m_\nu^\lambda) \nu^\lambda - \bar{u}_j^\lambda (\gamma \partial + \\
& m_u^\lambda) u_j^\lambda - \bar{d}_j^\lambda (\gamma \partial + m_d^\lambda) d_j^\lambda + ig_{s_w} A_\mu (-\bar{e}^\lambda \gamma^\mu e^\lambda + \frac{2}{3}(\bar{u}_j^\lambda \gamma^\mu u_j^\lambda) - \frac{1}{3}(\bar{d}_j^\lambda \gamma^\mu d_j^\lambda)) + \\
& \frac{ig}{4c_w} Z_\mu^0 (\bar{\nu}^\lambda \gamma^\mu (1 + \gamma^5) \nu^\lambda) + (\bar{e}^\lambda \gamma^\mu (4s_w^2 - 1 - \gamma^5) e^\lambda) + (\bar{d}_j^\lambda \gamma^\mu (\frac{4}{3}s_w^2 - 1 - \gamma^5) d_j^\lambda) + \\
& (\bar{u}_j^\lambda \gamma^\mu (1 - \frac{8}{3}s_w^2 + \gamma^5) u_j^\lambda) \} + \frac{ig}{2\sqrt{2}} W_\mu^+ ((\bar{\nu}^\lambda \gamma^\mu (1 + \gamma^5) U^{lep}{}_{\lambda\kappa} e^\kappa) + (\bar{u}_j^\lambda \gamma^\mu (1 + \gamma^5) C_{\lambda\kappa} d_j^\kappa)) + \\
& \frac{ig}{2\sqrt{2}} W_\mu^- ((\bar{e}^\kappa U^{lep}{}_{\kappa\lambda} \gamma^\mu (1 + \gamma^5) \nu^\lambda) + (\bar{d}_j^\kappa C_{\kappa\lambda}^\dagger \gamma^\mu (1 + \gamma^5) u_j^\lambda)) + \\
& \frac{ig}{2M\sqrt{2}} \phi^+ (-m_e^\kappa (\bar{\nu}^\lambda U^{lep}{}_{\lambda\kappa} (1 - \gamma^5) e^\kappa) + m_\nu^\lambda (\bar{\nu}^\lambda U^{lep}{}_{\lambda\kappa} (1 + \gamma^5) e^\kappa) + \\
& \frac{ig}{2M\sqrt{2}} \phi^- (m_e^\lambda (\bar{e}^\lambda U^{lep}{}_{\lambda\kappa}^\dagger (1 + \gamma^5) \nu^\kappa) - m_\nu^\kappa (\bar{e}^\lambda U^{lep}{}_{\lambda\kappa}^\dagger (1 - \gamma^5) \nu^\kappa) - \frac{g}{2} \frac{m_\lambda^2}{M} H (\bar{\nu}^\lambda \nu^\lambda) - \\
& \frac{g}{2} \frac{m_\lambda^2}{M} H (\bar{e}^\lambda e^\lambda) + \frac{ig}{2} \frac{m_\lambda^2}{M} \phi^0 (\bar{\nu}^\lambda \gamma^5 \nu^\lambda) - \frac{ig}{2} \frac{m_\lambda^2}{M} \phi^0 (\bar{e}^\lambda \gamma^5 e^\lambda) - \frac{1}{4} \bar{\nu}_\lambda M_{\lambda\kappa}^R (1 - \gamma_5) \bar{\nu}_\kappa - \\
& \frac{1}{4} \bar{\nu}_\lambda M_{\lambda\kappa}^R (1 - \gamma_5) \bar{\nu}_\kappa + \frac{ig}{2M\sqrt{2}} \phi^+ (-m_d^\kappa (\bar{u}_j^\kappa C_{\lambda\kappa} (1 - \gamma^5) d_j^\kappa) + m_u^\kappa (\bar{u}_j^\kappa C_{\lambda\kappa} (1 + \gamma^5) d_j^\kappa) + \\
& \frac{ig}{2M\sqrt{2}} \phi^- (m_d^\kappa (\bar{d}_j^\kappa C_{\lambda\kappa}^\dagger (1 + \gamma^5) u_j^\kappa) - m_u^\kappa (\bar{d}_j^\kappa C_{\lambda\kappa}^\dagger (1 - \gamma^5) u_j^\kappa) - \frac{g}{2} \frac{m_\lambda^2}{M} H (\bar{u}_j^\lambda u_j^\lambda) - \\
& \frac{g}{2} \frac{m_\lambda^2}{M} H (\bar{d}_j^\lambda d_j^\lambda) + \frac{ig}{2} \frac{m_\lambda^2}{M} \phi^0 (\bar{u}_j^\lambda \gamma^5 u_j^\lambda) - \frac{ig}{2} \frac{m_\lambda^2}{M} \phi^0 (\bar{d}_j^\lambda \gamma^5 d_j^\lambda) + \bar{G}^a \partial^2 G^a + g_s f^{abc} \partial_\mu \bar{G}^a G^b g_\mu^c + \\
& \bar{X}^+ (\partial^2 - M^2) X^+ + \bar{X}^- (\partial^2 - M^2) X^- + \bar{X}^0 (\partial^2 - \frac{M^2}{c_w^2}) X^0 + \bar{Y} \partial^2 Y + ig_{c_w} W_\mu^+ (\partial_\mu \bar{X}^0 X^- - \\
& \partial_\mu \bar{X}^+ X^0) + ig_{s_w} W_\mu^+ (\partial_\mu \bar{Y} X^- - \partial_\mu \bar{X}^+ Y) + ig_{c_w} W_\mu^- (\partial_\mu \bar{X}^- X^0 - \\
& \partial_\mu \bar{X}^0 X^+) + ig_{s_w} W_\mu^- (\partial_\mu \bar{X}^- Y - \partial_\mu \bar{Y} X^+) + ig_{c_w} Z_\mu^0 (\partial_\mu \bar{X}^+ X^- - \\
& \partial_\mu \bar{X}^- X^+) + ig_{s_w} A_\mu (\partial_\mu \bar{X}^+ X^- - \\
& \partial_\mu \bar{X}^- X^+) - \frac{1}{2}gM (\bar{X}^+ X^+ H + \bar{X}^- X^- H + \frac{1}{c_w^2} \bar{X}^0 X^0 H) + \frac{1-2c_w^2}{2c_w} igM (\bar{X}^+ X^0 \phi^+ - \bar{X}^- X^0 \phi^-) + \\
& \frac{1}{2c_w} igM (\bar{X}^0 X^- \phi^+ - \bar{X}^0 X^+ \phi^-) + igM s_w (\bar{X}^0 X^- \phi^+ - \bar{X}^0 X^+ \phi^-) + \\
& \frac{1}{2}igM (\bar{X}^+ X^+ \phi^0 - \bar{X}^- X^- \phi^0) .
\end{aligned}$$





EA-01-2474

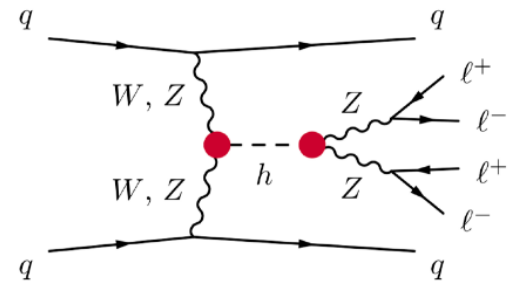
Latent variables

Parameters of interest

Parton-level momenta

Theory parameters

$$z_p \longleftarrow \theta$$



Latent variables

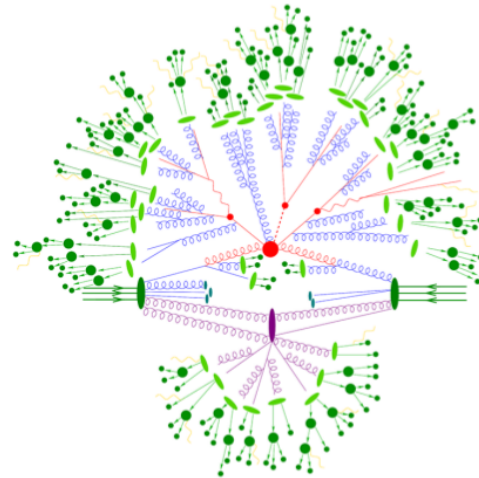
Parameters
of interest

Shower
splittings

Parton-level
momenta

Theory
parameters

z_s ← z_p ← θ



Latent variables

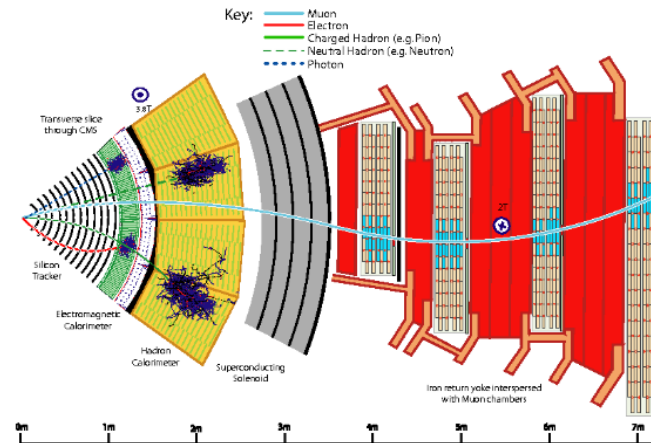
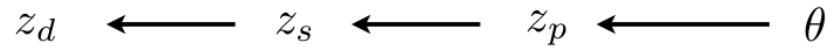
Parameters of interest

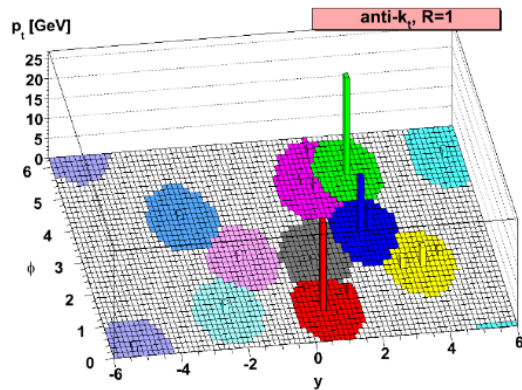
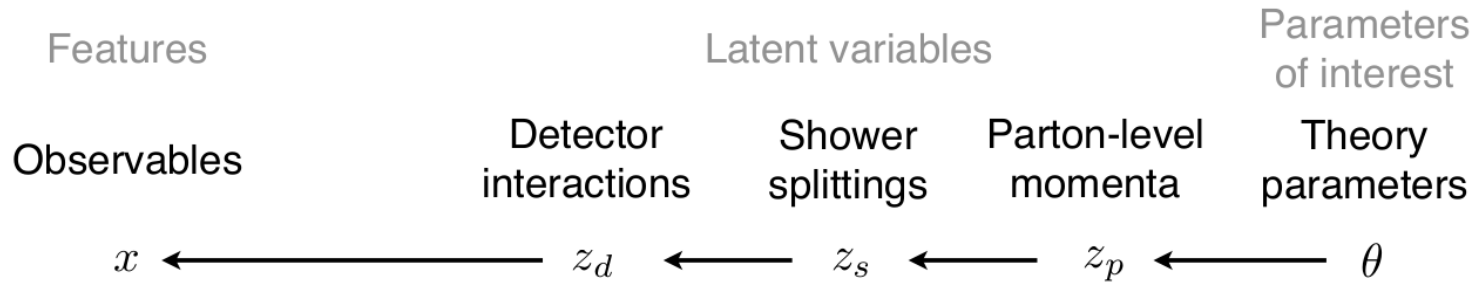
Detector interactions

Shower splittings

Parton-level momenta

Theory parameters





[Image source: M. Cacciari, G. Salam, G. Soyez 0802.1189]

$$p(x|\theta) = \underbrace{\iiint}_{\text{intractable}} p(z_p|\theta)p(z_s|z_p)p(z_d|z_s)p(x|z_d)dz_pdz_sdz_d$$

Likelihood ratio

The likelihood ratio

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

is the quantity that is **central** to many **statistical inference** procedures.

Examples

- Frequentist hypothesis testing
- Bayesian model comparison
- Bayesian posterior inference with MCMC or VI
- Supervised learning
- Generative adversarial networks
- Empirical Bayes with Adversarial Variational Optimization
- Optimal compression

When solving a problem of interest, do not solve a more general problem as an intermediate step. – Vladimir Vapnik



$$\frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)} = r(\mathbf{x}; \theta_0, \theta_1)$$

The equation is annotated with a blue curved arrow pointing from the left side to the right side, and a red curved arrow pointing from the right side to the left side, with a red diagonal slash through it.

Direct likelihood ratio estimation is simpler than density estimation.

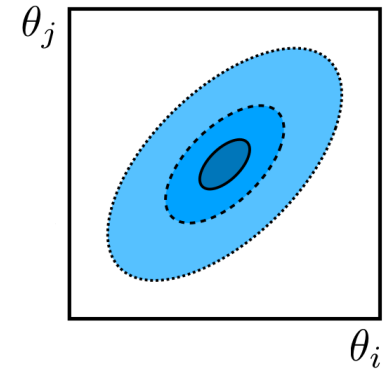
(This is fortunate, we are in the likelihood-free scenario!)

The frequentist (physicist's) way

The Neyman-Pearson lemma states that the likelihood ratio

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

is the **most powerful test statistic** to discriminate between a null hypothesis θ_0 and an alternative θ_1 .



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

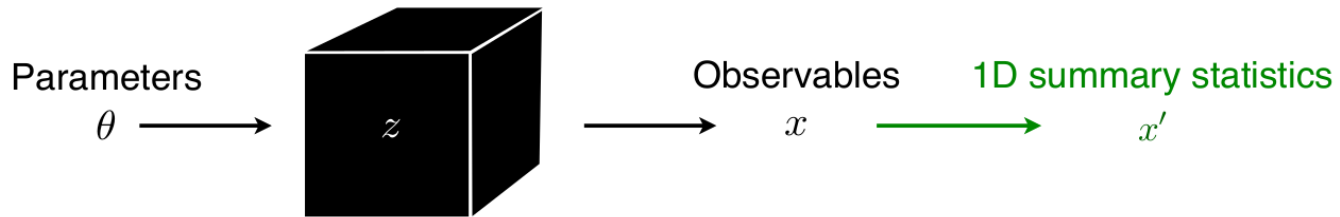
By J. NEYMAN, *Nencki Institute, Soc. Sci. Lit. Varsoviensis, and Lecturer at the Central College of Agriculture, Warsaw,* and E. S. PEARSON, *Department of Applied Statistics, University College, London.*

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

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

CONTENTS.

	PAGE.
I. Introductory	289
II. Outline of General Theory	293
III. Simple Hypotheses	



Define a projection function $s : \mathcal{X} \rightarrow \mathbb{R}$ mapping observables x to a summary statistic $x' = s(x)$.

Then, **approximate** the likelihood $p(x|\theta)$ with the surrogate $\hat{p}(x|\theta) = p(x'|\theta)$.

From this it comes

$$\frac{p(x|\theta_0)}{p(x|\theta_1)} \approx \frac{\hat{p}(x|\theta_0)}{\hat{p}(x|\theta_1)} = \hat{r}(x|\theta_0, \theta_1).$$

Wilks theorem

Consider the test statistic

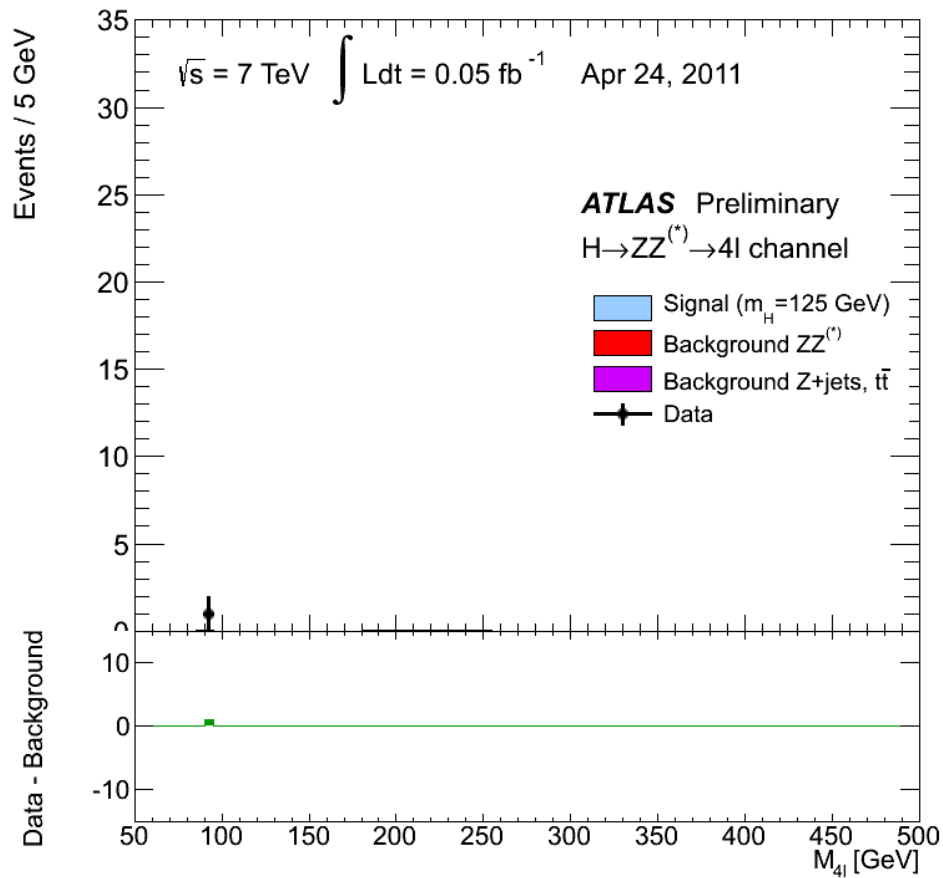
$$q(\theta) = -2 \sum_x \log \frac{p(x|\theta)}{p(x|\hat{\theta})} = -2 \sum_x \log r(x|\theta, \hat{\theta})$$

for a fixed number N of observations $\{x\}$ and where $\hat{\theta}$ is the maximum likelihood estimator.

When $N \rightarrow \infty$, $q(\theta) \sim \chi^2$.

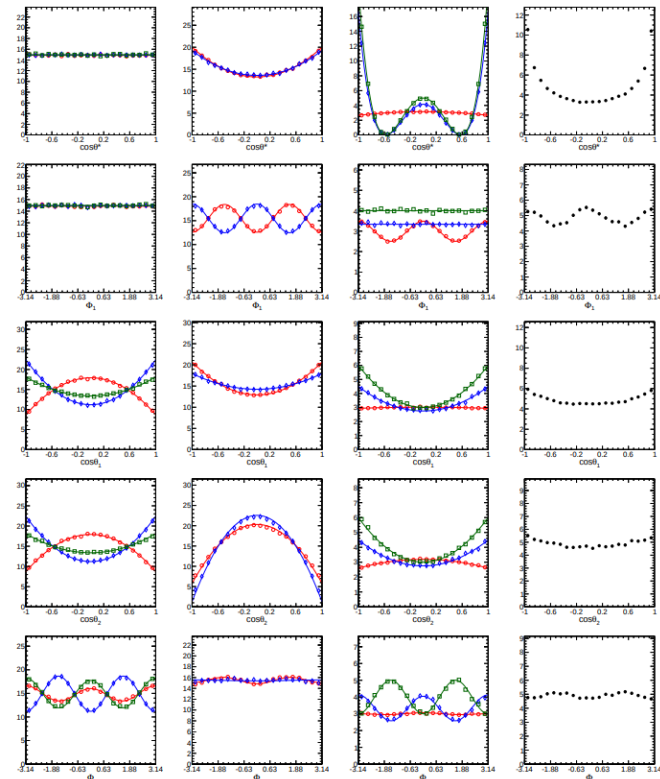
Therefore (and provided the assumptions apply!), an observed value $q_{\text{obs}}(\theta)$ translates directly to a p-value that measures the confidence with which θ can be excluded:

$$p_\theta \equiv \int_{q_{\text{obs}}(\theta)}^{\infty} dq p(q|\theta) = 1 - F_{\chi^2}(q_{\text{obs}}(\theta)).$$



Discovery of the Higgs boson at $5\text{-}\sigma$

- Choosing the projection s is difficult and problem-dependent.
- Often there is no single good variable: compressing to any x' loses information.
- Ideally, analyze **high-dimensional** x' , including all correlations.
- Unfortunately, filling high-dimensional histograms is **not tractable**.

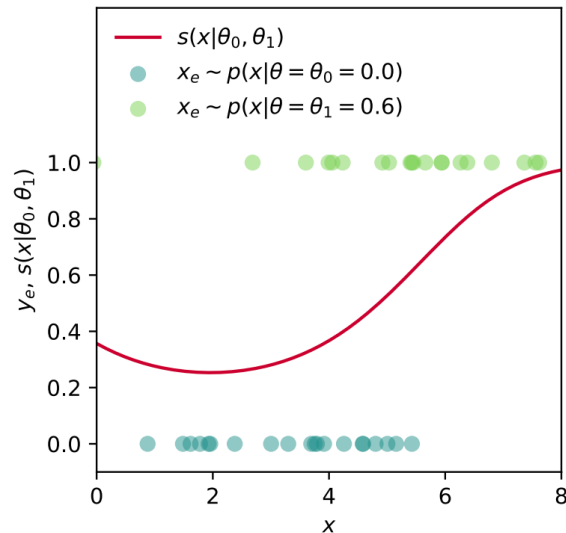


CARL

Supervised learning provides a way to **automatically** construct s :

- Let us consider a neural network classifier \hat{s} tasked to distinguish $x_i \sim p(x|\theta_0)$ labelled $y_i = 0$ from $x_i \sim p(x|\theta_1)$ labelled $y_i = 1$.
- Train \hat{s} by minimizing the cross-entropy loss

$$L_{XE}[\hat{s}] = -\frac{1}{N} \sum_{x_i, y_i} y_i \log \hat{s}(x_i) + (1 - y_i) \log(1 - \hat{s}(x_i)).$$



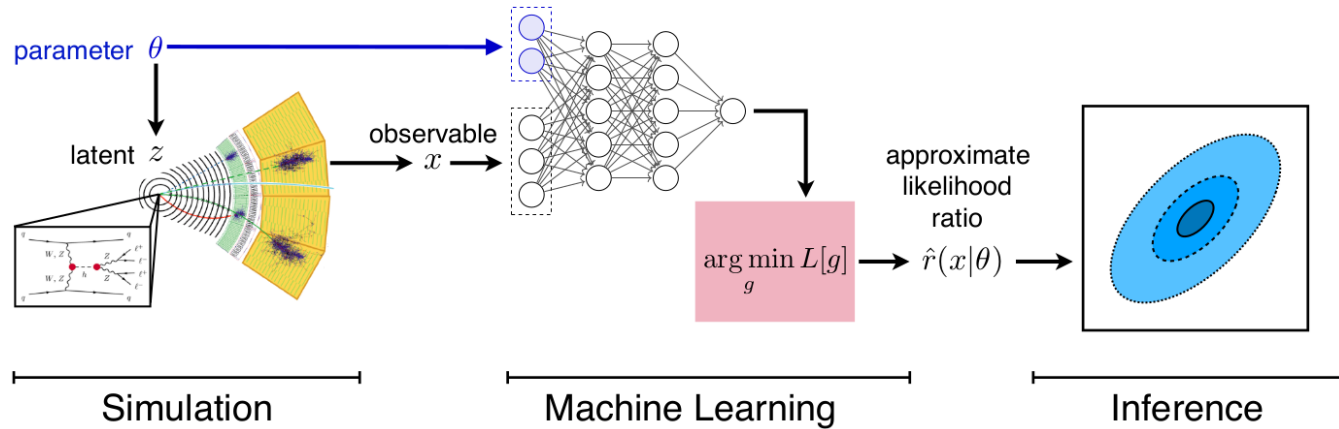
The solution \hat{s} found after training approximates the optimal classifier

$$\hat{s}(x) \approx s^*(x) = \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)}.$$

Therefore,

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

That is, **supervised classification** is equivalent to **likelihood ratio estimation**.



To avoid retraining a classifier \hat{s} for every (θ_0, θ_1) pair, fix θ_1 to θ_{ref} and train a single **parameterized** classifier $\hat{s}(x|\theta_0, \theta_{\text{ref}})$ where θ_0 is also given as input.

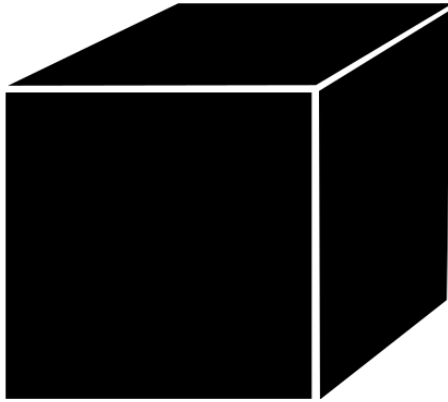
Therefore, we have

$$\hat{r}(x|\theta_0, \theta_{\text{ref}}) = \frac{1 - \hat{s}(x|\theta_0, \theta_{\text{ref}})}{\hat{s}(x|\theta_0, \theta_{\text{ref}})}$$

such that for any (θ_0, θ_1) ,

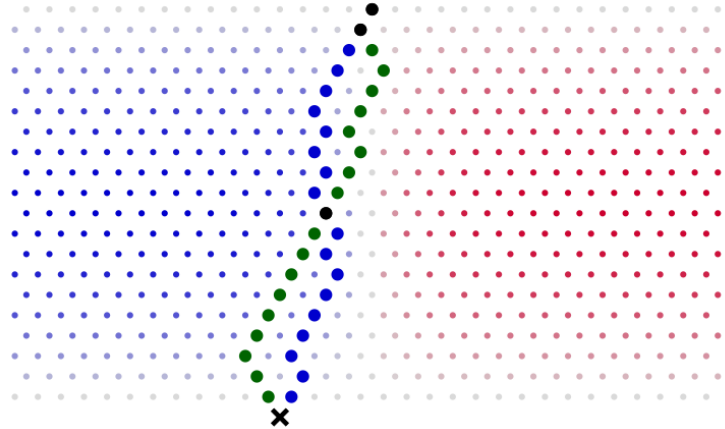
$$r(x|\theta_0, \theta_1) \approx \frac{\hat{r}(x|\theta_0, \theta_{\text{ref}})}{\hat{r}(x|\theta_1, \theta_{\text{ref}})}.$$

Opening the black box

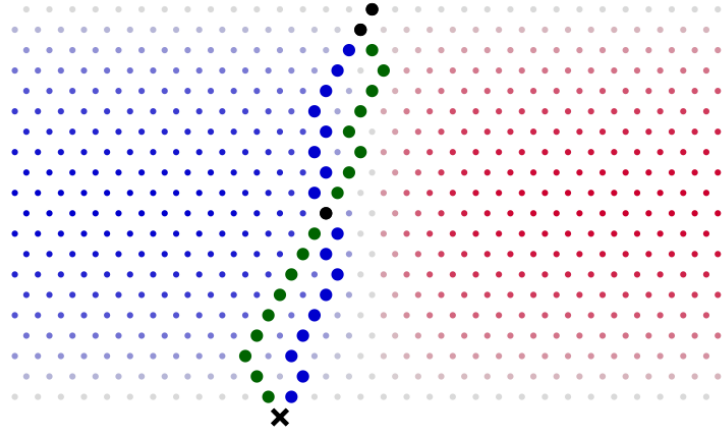


Traditional likelihood-free inference treats the simulator as a generative **black box**: parameters in, samples out.

But in most real-life problems, we have access to the simulator code and some understanding of the microscopic processes.



$p(x|\theta)$ is usually intractable. What about $p(x, z|\theta)$?



$p(x|\theta)$ is usually intractable. What about $p(x, z|\theta)$?

As the trajectory z_1, \dots, z_T and the observable x are emitted, it is often possible:

- to calculate the **joint likelihood** $p(x, z|\theta)$;
- to calculate the **joint likelihood ratio** $r(x, z|\theta_0, \theta_1)$;
- to calculate the **joint score** $t(x, z|\theta_0) = \nabla_{\theta} \log p(x, z|\theta)|_{\theta_0}$.

We call this process **mining gold** from your simulator!

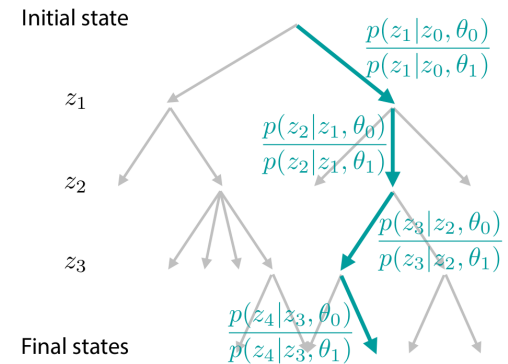
Extracting the joint likelihood ratio

- Computer simulations typically evolve along a tree-like structure of successive random branchings.
- The probabilities of each branching $p(z_i | z_{<i}, \theta)$ 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, we can calculate the probability of the chosen path for different values of the parameters and the joint likelihood-ratio:

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



RASCAL

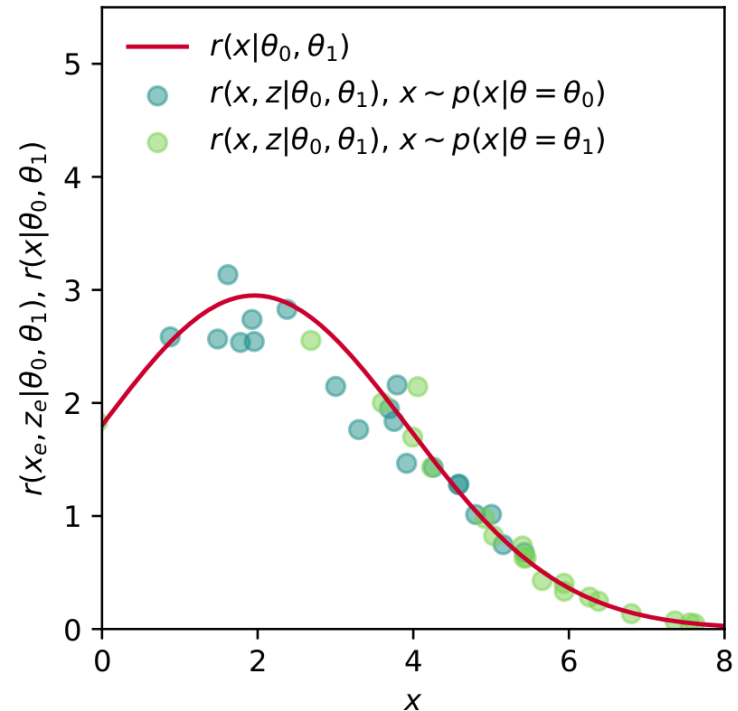
Regressing the likelihood ratio

Observe that the joint likelihood ratios

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

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

Can we use them to approximate $r(x|\theta_0, \theta_1)$?



Consider the squared error of a function $\hat{g}(x)$ that only depends on x , but is trying to approximate a function $g(x, z)$ that also depends on the latent z :

$$L_{\text{MSE}} = \mathbb{E}_{p(x, z | \theta)} [(g(x, z) - \hat{g}(x))^2].$$

Via calculus of variations, we find that the function $g^*(x)$ that extremizes $L_{\text{MSE}}[g]$ is given by

$$\begin{aligned} g^*(x) &= \frac{1}{p(x|\theta)} \int p(x, z|\theta) g(x, z) dz \\ &= \mathbb{E}_{p(z|x, \theta)} [g(x, z)] \end{aligned}$$

Therefore, by identifying the $g(x, z)$ with the joint likelihood ratio $r(x, z|\theta_0, \theta_1)$ and θ with θ_1 , we define

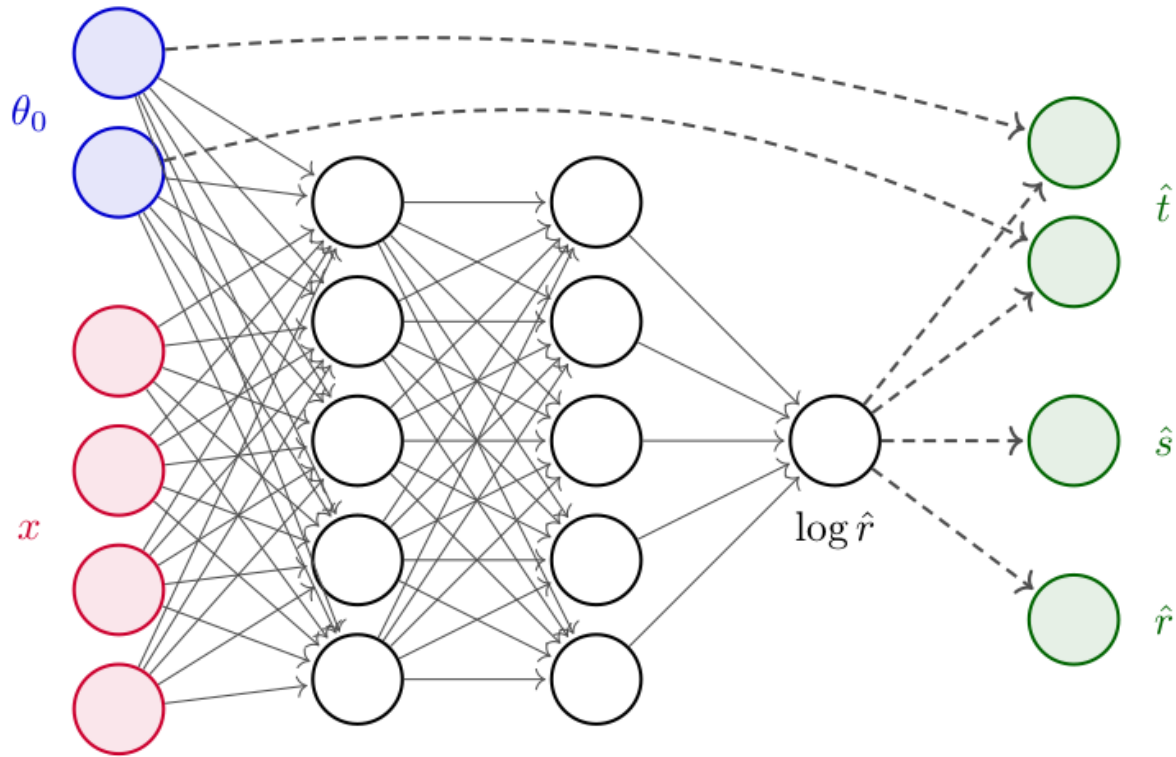
$$L_r = \mathbb{E}_{p(x, z|\theta_1)} [(r(x, z|\theta_0, \theta_1) - \hat{r}(x))^2],$$

which is minimized by

$$\begin{aligned} r^*(x) &= \frac{1}{p(x|\theta_1)} \int p(x, z|\theta_1) \frac{p(x, z|\theta_0)}{p(x, z|\theta_1)} dz \\ &= \frac{p(x|\theta_0)}{p(x|\theta_1)} \\ &= r(x|\theta_0, \theta_1). \end{aligned}$$

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

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

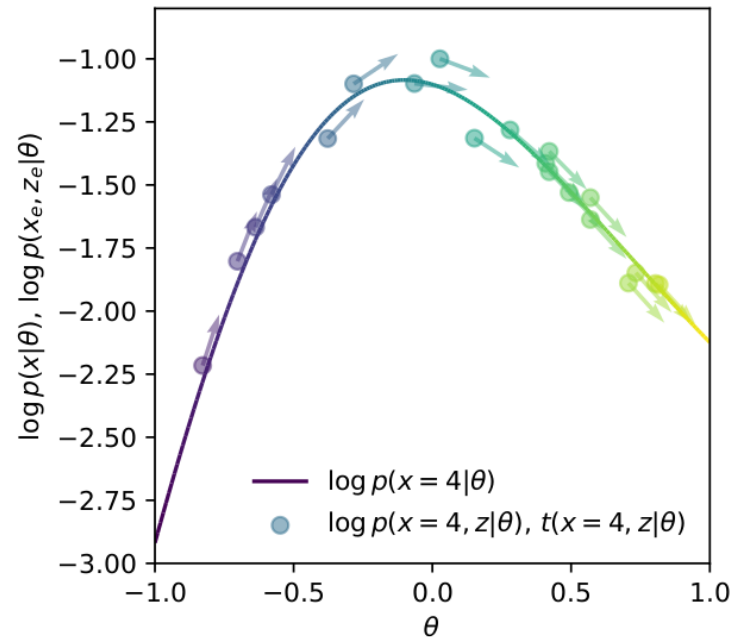


Regressing the score

Similarly, we can mine the simulator to extract the joint score

$$t(x, z|\theta_0) = \nabla_{\theta} \log p(x, z|\theta)|_{\theta_0},$$

which indicates how much more or less likely x, z would be if one changed θ_0 .



Using the same trick, by identifying $g(x, z)$ with the joint score $t(x, z|\theta_0)$ and θ with θ_0 , we define

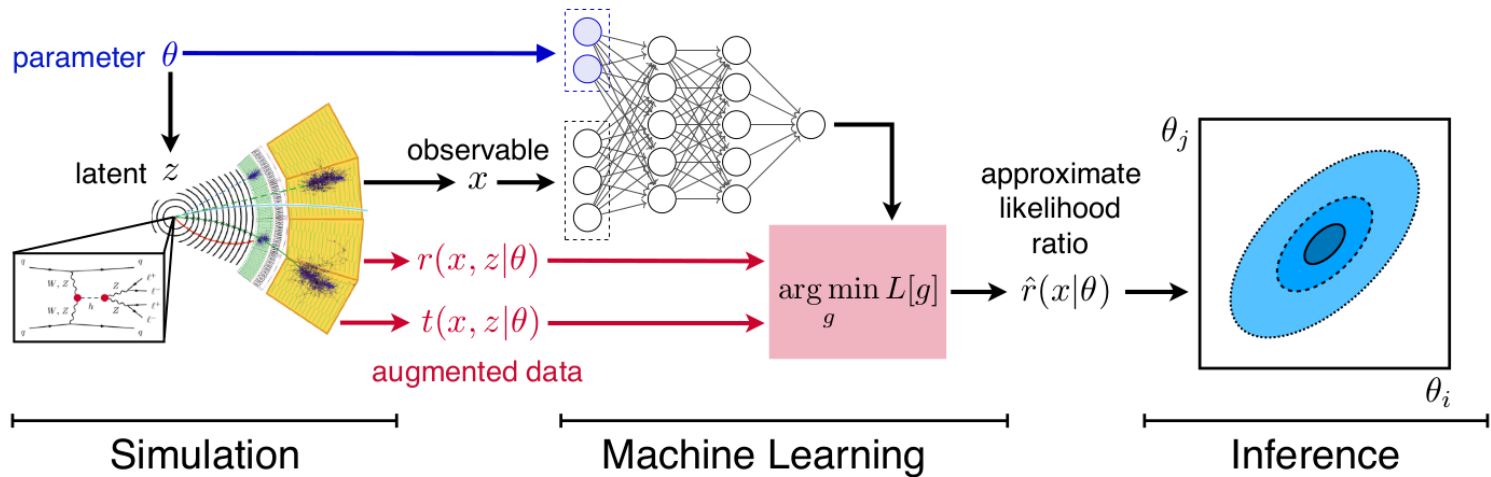
$$L_t = \mathbb{E}_{p(x, z|\theta_0)} [(t(x, z|\theta_0) - \hat{t}(x))^2],$$

which is minimized by

$$\begin{aligned} t^*(x) &= \frac{1}{p(x|\theta_0)} \int p(x, z|\theta_0) (\nabla_{\theta} \log p(x, z|\theta)|_{\theta_0}) dz \\ &= \frac{1}{p(x|\theta_0)} \int p(x, z|\theta_0) \frac{\nabla_{\theta} p(x, z|\theta)|_{\theta_0}}{p(x, z|\theta_0)} dz \\ &= \frac{\nabla_{\theta} p(x|\theta)|_{\theta_0}}{p(x|\theta_0)} \\ &= \nabla_{\theta} \log p(x|\theta)|_{\theta_0} \\ &= t(x|\theta_0). \end{aligned}$$

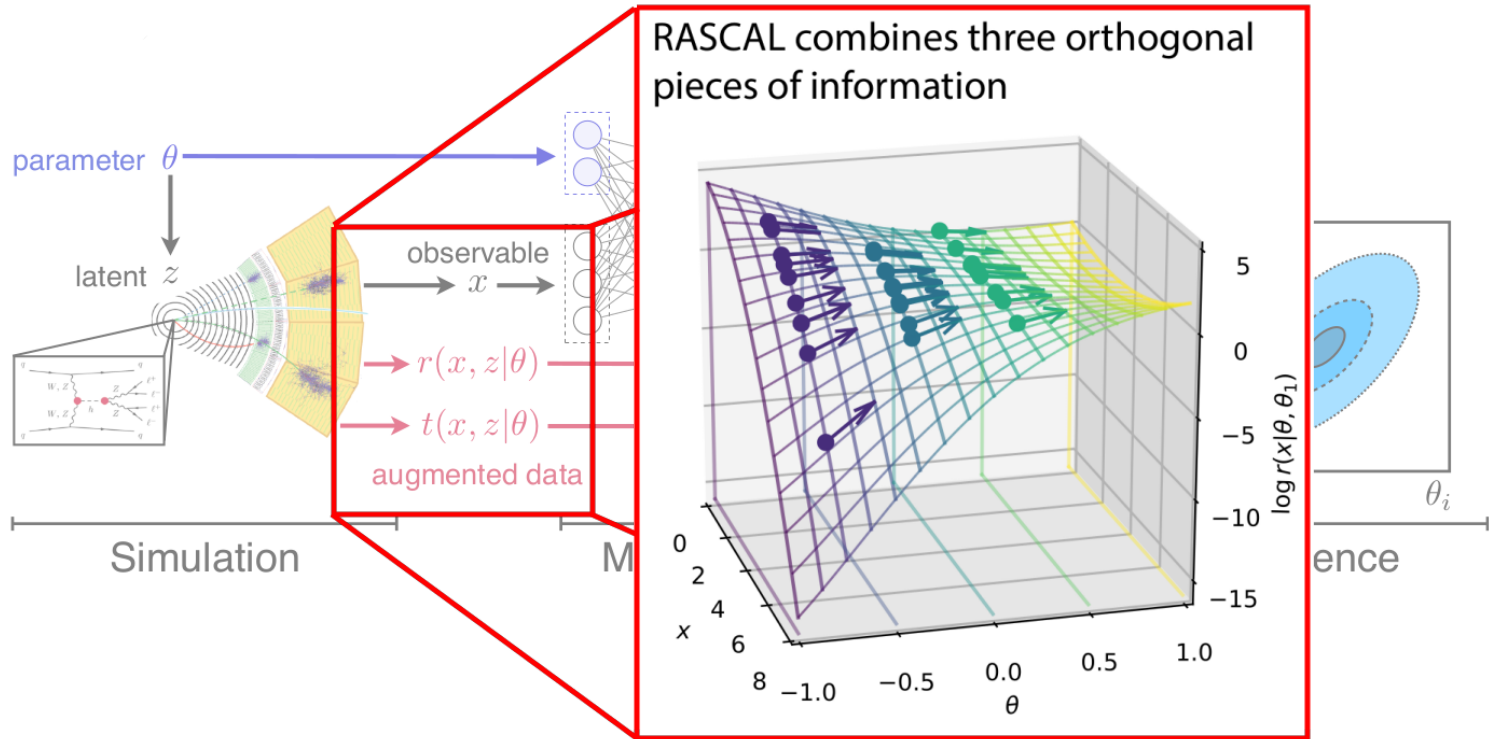
RASCAL

$$L_{\text{RASCAL}} = L_r + L_t$$



RASCAL

$$L_{\text{RASCAL}} = L_r + L_t$$

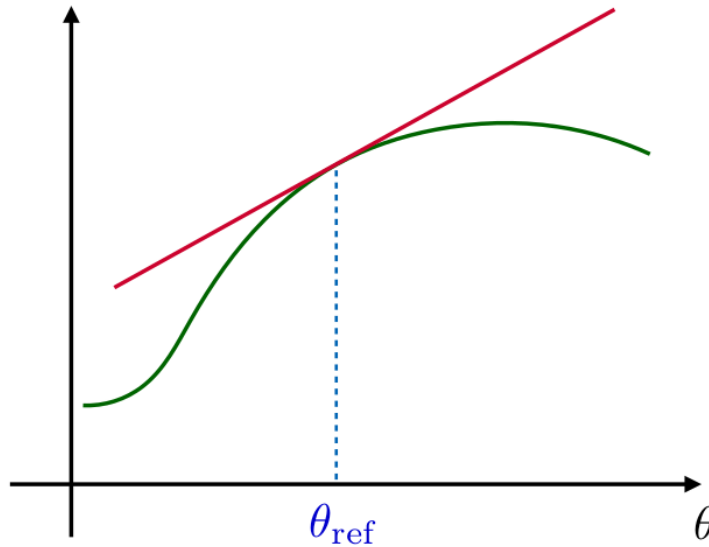


SALLY (= optimal compression)

The local model

In the neighborhood of θ_{ref} , the Taylor expansion of $\log p(x|\theta)$ is

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



This results in the exponential model

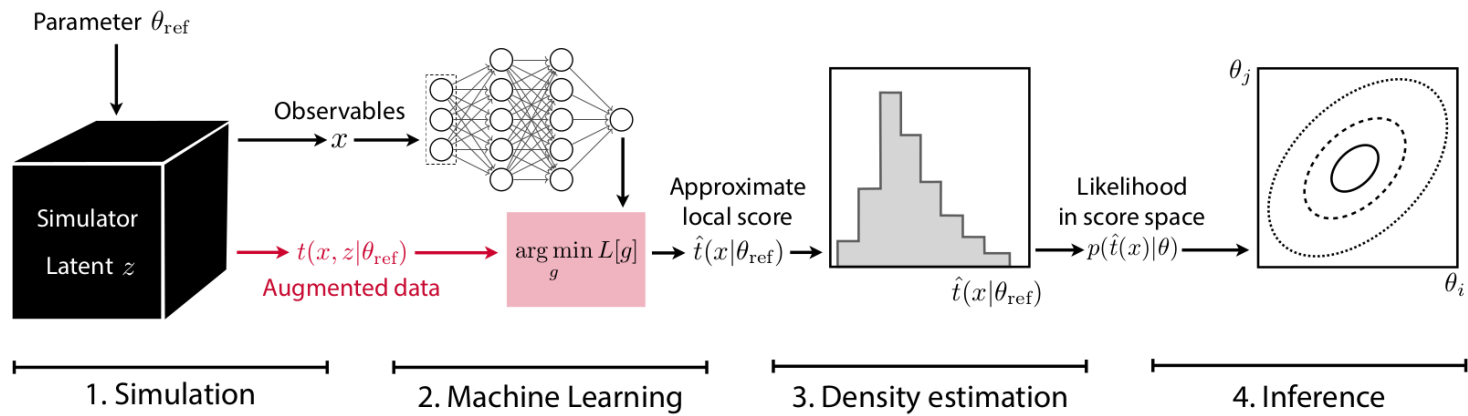
$$p_{\text{local}}(x|\theta) = \frac{1}{Z(\theta)} p(t(x|\theta_{\text{ref}})|\theta_{\text{ref}}) \exp(t(x|\theta_{\text{ref}}) \cdot (\theta - \theta_{\text{ref}}))$$

where the score $t(x|\theta_{\text{ref}})$ are its sufficient statistics.

That is,

- knowing $t(x|\theta_{\text{ref}})$ is just as powerful as knowing the full function $\log p(x|\theta)$.
- x can be compressed into a single scalar $t(x|\theta_{\text{ref}})$ without loss of power.

SALLY



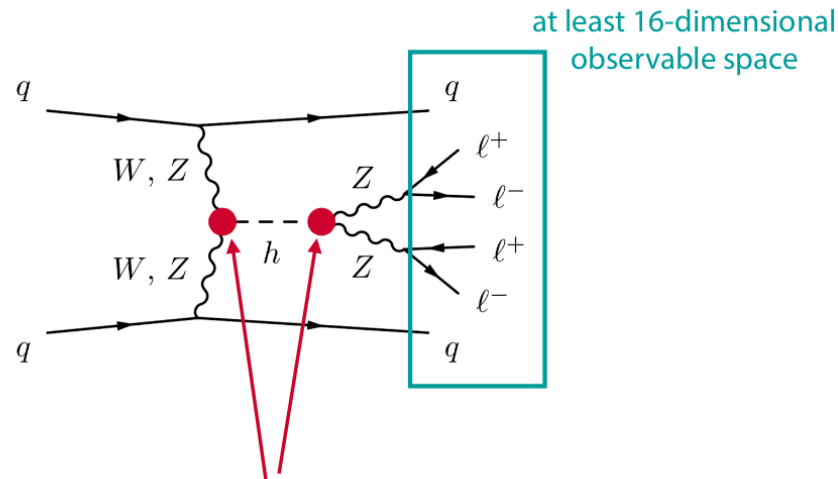
There is more...

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

Examples

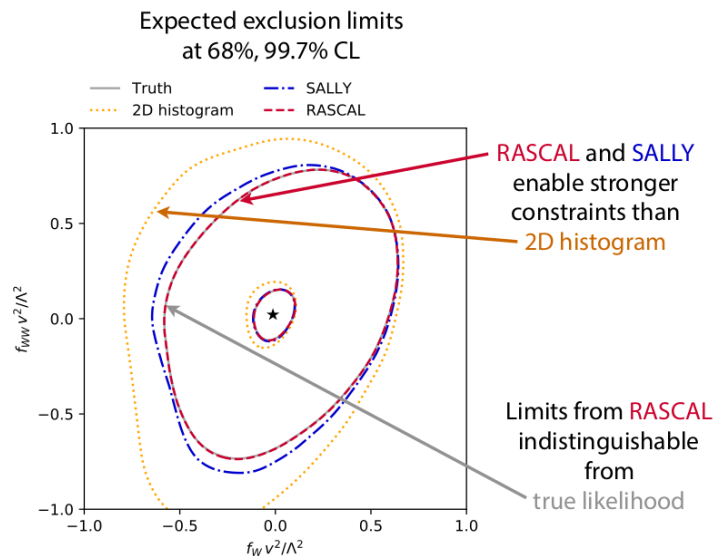
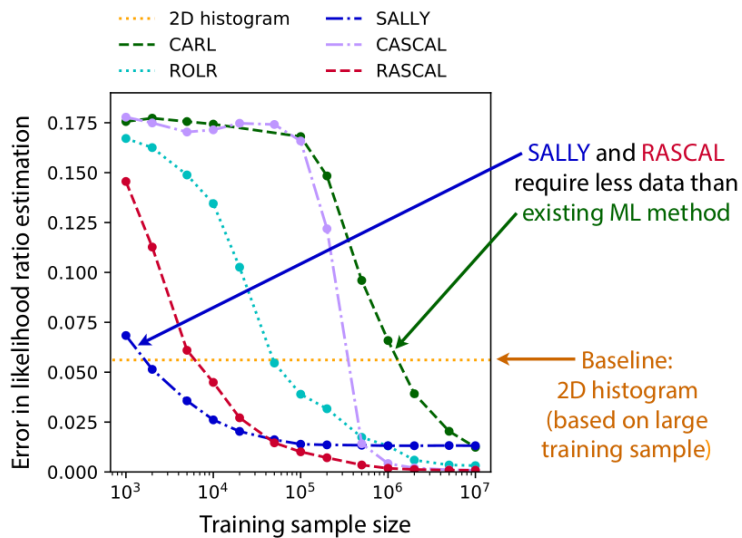
① Hunting new physics at particle colliders

The goal is to constrain two EFT parameters and compare against traditional histogram analysis.

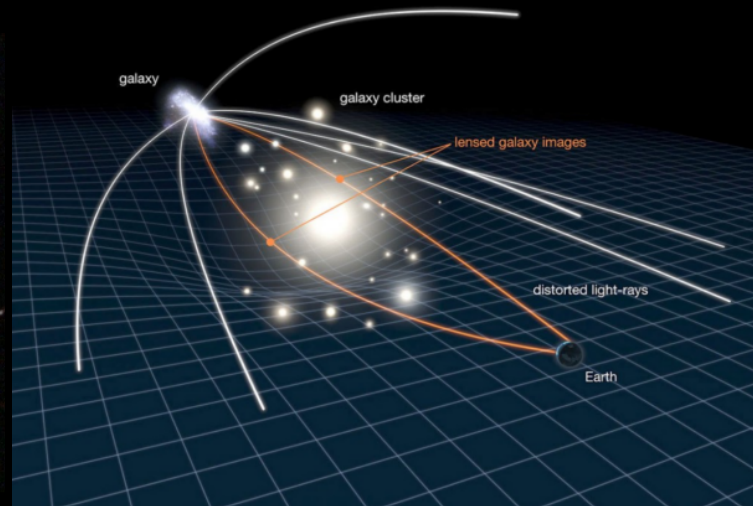
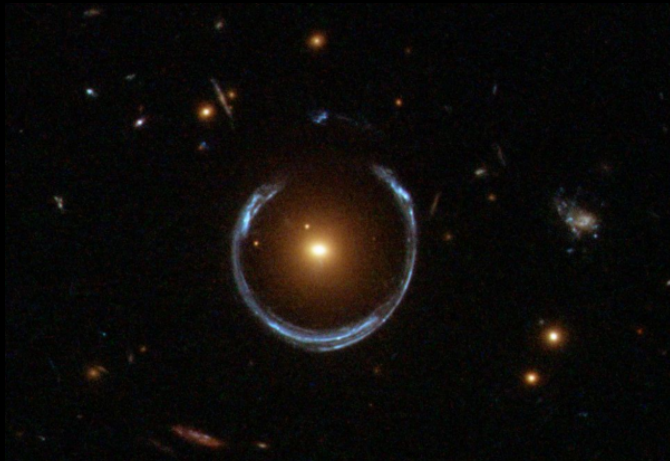


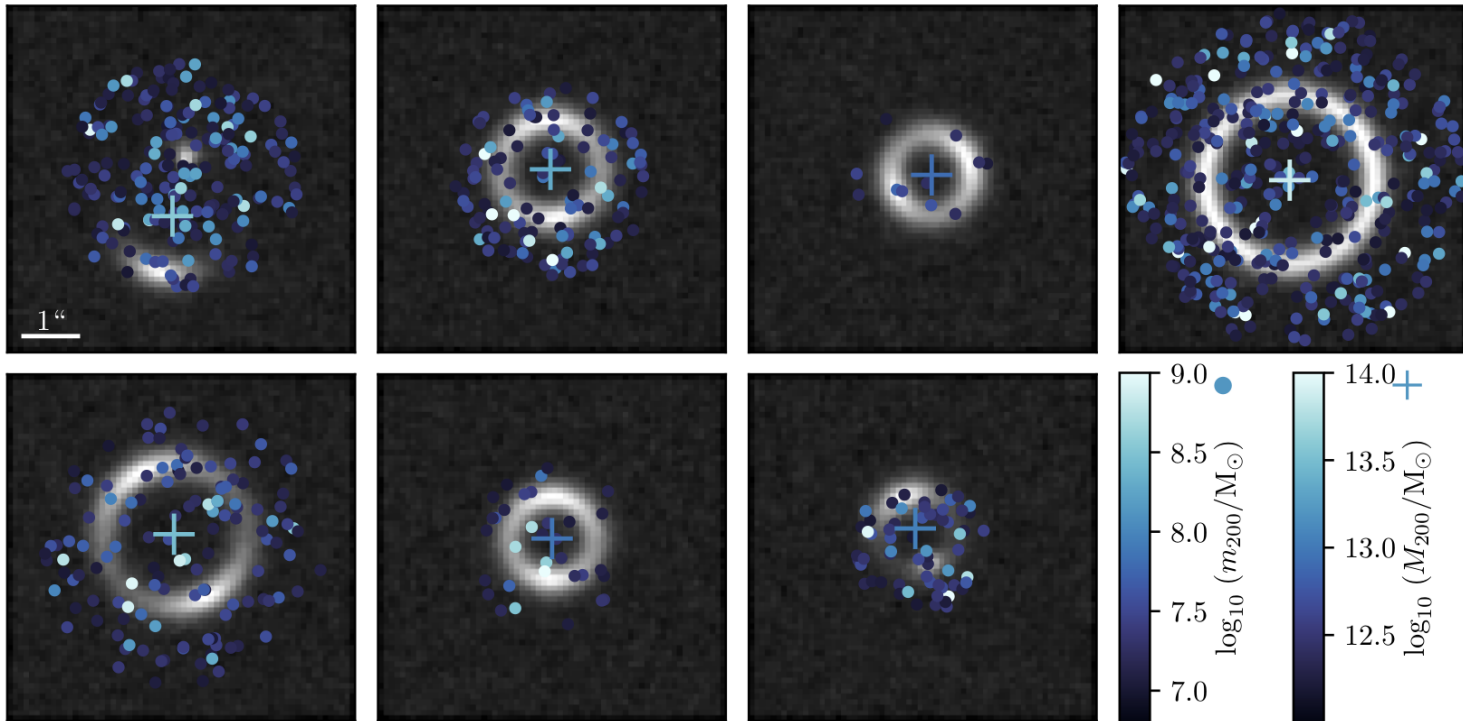
Exciting new physics might hide here!
We parameterize it with two EFT coefficients:

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



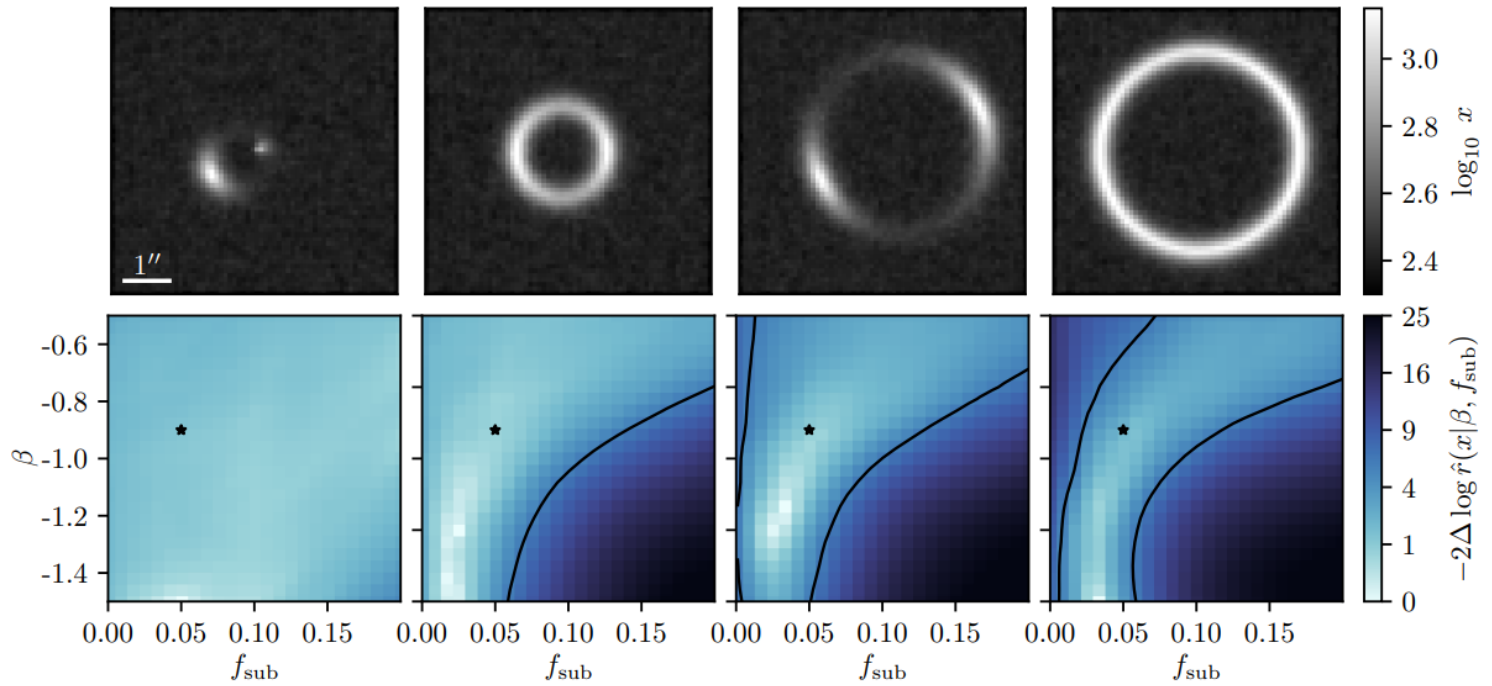
② Dark matter substructure from gravitational lensing





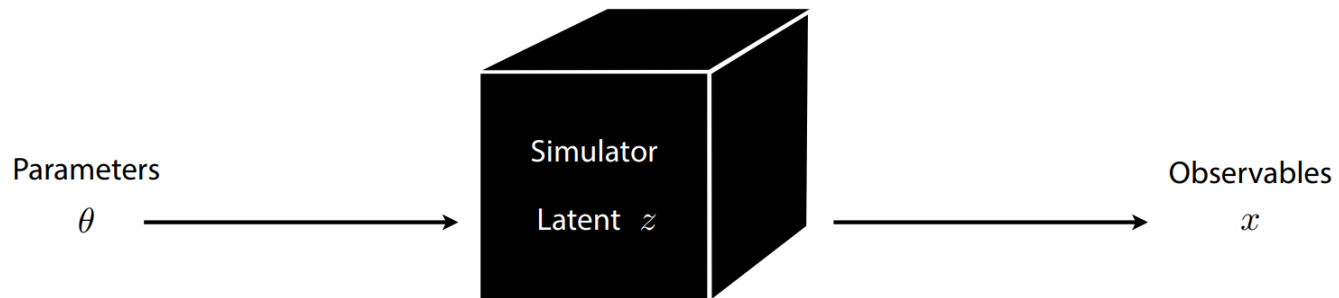
Number of dark matter subhalos and their mass and location lead to complex latent space of each image.

The goal is the **inference of population parameters** β and f_{sub} .



Summary

- Much of modern science is based on "likelihood-free" simulations.
- The likelihood-ratio is central to many statistical inference procedures, regardless of your religion.
- Supervised learning enables likelihood-ratio estimation.
- Better likelihood-ratio estimates can be achieved by mining simulators.
- (Probabilistic programming enables posterior inference in scientific simulators.)



Collaborators



Kyle Cranmer



Juan Pavez



Johann
Brehmer



Joeri
Hermans



Antoine
Wehenkel



Siddarth
Mishra-
Sharma



Lukas
Heinrich



Atılım Güneş
Baydin



Wahid Bhimji



Frank Wood

References

- Brehmer, J., Mishra-Sharma, S., Hermans, J., Louppe, G., Cranmer, K. (2019). Mining for Dark Matter Substructure: Inferring subhalo population properties from strong lenses with machine learning. arXiv preprint arXiv:1909.02005.
- Hermans, J., Begy, V., & Louppe, G. (2019). Likelihood-free MCMC with Approximate Likelihood Ratios. arXiv preprint arXiv:1903.04057.
- Baydin, A. G., Shao, L., Bhimji, W., Heinrich, L., Meadows, L., Liu, J., ... & Ma, M. (2019). Etalumis: Bringing Probabilistic Programming to Scientific Simulators at Scale. arXiv preprint arXiv:1907.03382.
- Stoye, M., Brehmer, J., Louppe, G., Pavez, J., & Cranmer, K. (2018). Likelihood-free inference with an improved cross-entropy estimator. arXiv preprint arXiv:1808.00973.
- Baydin, A. G., Heinrich, L., Bhimji, W., Gram-Hansen, B., Louppe, G., Shao, L., ... & Wood, F. (2018). Efficient Probabilistic Inference in the Quest for Physics Beyond the Standard Model. arXiv preprint arXiv:1807.07706.
- Brehmer, J., Louppe, G., Pavez, J., & Cranmer, K. (2018). Mining gold from implicit models to improve likelihood-free inference. arXiv preprint arXiv:1805.12244.
- Brehmer, J., Cranmer, K., Louppe, G., & Pavez, J. (2018). Constraining Effective Field Theories with Machine Learning. arXiv preprint arXiv:1805.00013.
- Brehmer, J., Cranmer, K., Louppe, G., & Pavez, J. (2018). A Guide to Constraining Effective Field Theories with Machine Learning. arXiv preprint arXiv:1805.00020.
- Casado, M. L., Baydin, A. G., Rubio, D. M., Le, T. A., Wood, F., Heinrich, L., ... & Bhimji, W. (2017). Improvements to Inference Compilation for Probabilistic Programming in Large-Scale Scientific Simulators. arXiv preprint arXiv:1712.07901.
- Louppe, G., Hermans, J., & Cranmer, K. (2017). Adversarial Variational Optimization of Non-Differentiable Simulators. arXiv preprint arXiv:1707.07113.
- Cranmer, K., Pavez, J., & Louppe, G. (2015). Approximating likelihood ratios with calibrated discriminative classifiers. arXiv preprint arXiv:1506.02169.

The end.

ALICE

When the joint likelihood ratio $r(x, z|\theta_0, \theta_1)$ is available from the simulator, the corresponding $s(x, z|\theta_0, \theta_1)$ are also tractable.

Therefore, the original CARL cross-entropy can be adapted to make use of the exact $s(x, z|\theta_0, \theta_1)$ instead of using labels $y \in \{0, 1\}$:

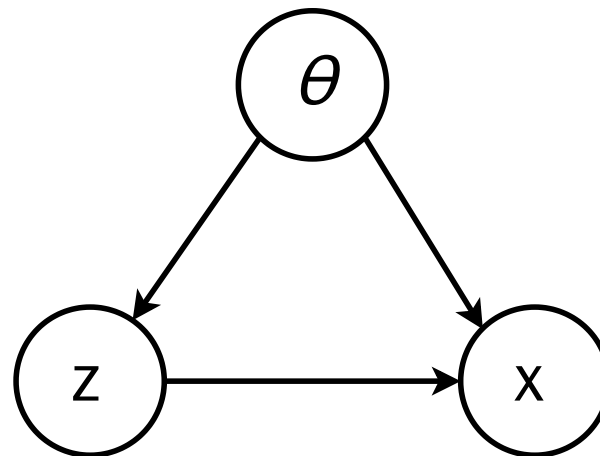
$$L_{ALICE}[\hat{s}] = -\mathbb{E}_{p(x,z)}[s(x, z|\theta_0, \theta_1) \log(\hat{s}(x)) + (1 - s(x, z|\theta_0, \theta_1)) \log(1 - \hat{s}(x))],$$

where $p(x, z) = (p(x, z|\theta_0) + p(x, z|\theta_1))/2$.

Bayesian inference

Bayesian inference = computing the posterior

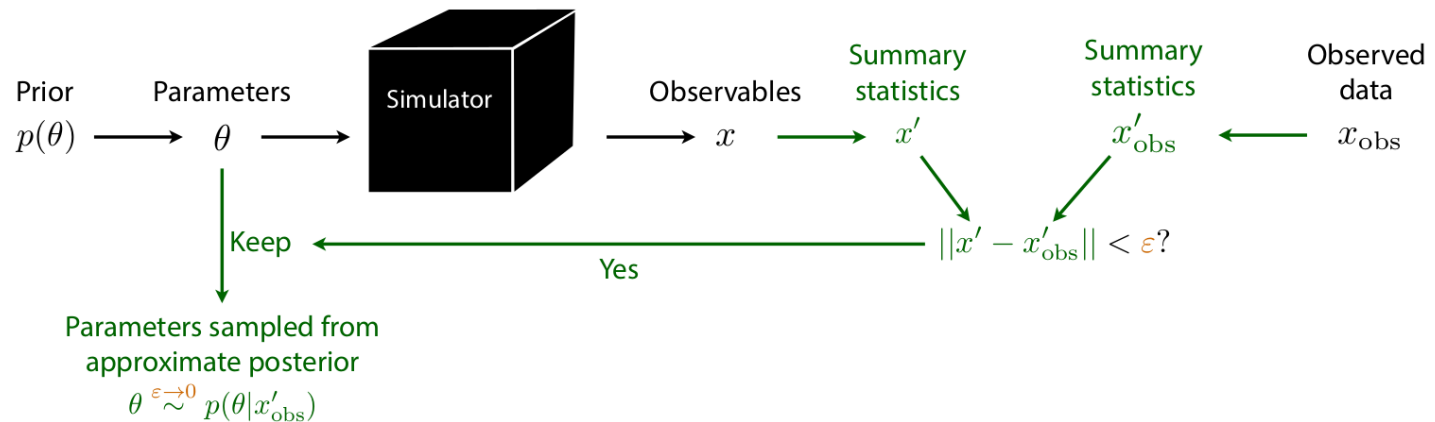
$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}.$$



Doubly **intractable** in the likelihood-free scenario:

- Cannot evaluate the likelihood $p(x|\theta) = \int p(x, z|\theta)dz$.
- Cannot evaluate the evidence $p(x) = \int p(x|\theta)p(\theta)d\theta$.

Approximate Bayesian Computation (ABC)



Issues

- How to choose x' ? ϵ ? $\| \cdot \|$?
- No tractable posterior.
- Need to run new simulations for new data or new prior.

Amortizing Bayes

The Bayes rule can be rewritten as

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} = r(x|\theta)p(\theta) \approx \hat{r}(x|\theta)p(\theta),$$

where $r(x|\theta) = \frac{p(x|\theta)}{p(x)}$ is the likelihood-to-evidence ratio.

Amortizing Bayes

The Bayes rule can be rewritten as

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} = r(x|\theta)p(\theta) \approx \hat{r}(x|\theta)p(\theta),$$

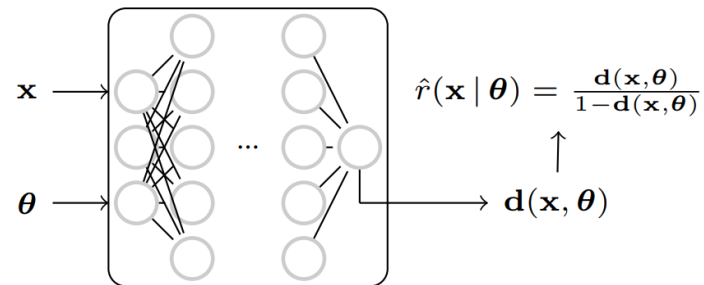
where $r(x|\theta) = \frac{p(x|\theta)}{p(x)}$ is the likelihood-to-evidence ratio.

As before, the likelihood-to-evidence ratio can be approximated e.g. from a neural network classifier trained to distinguish $x \sim p(x|\theta)$ from $x \sim p(x)$, hence enabling **direct** and **amortized** posterior evaluation.

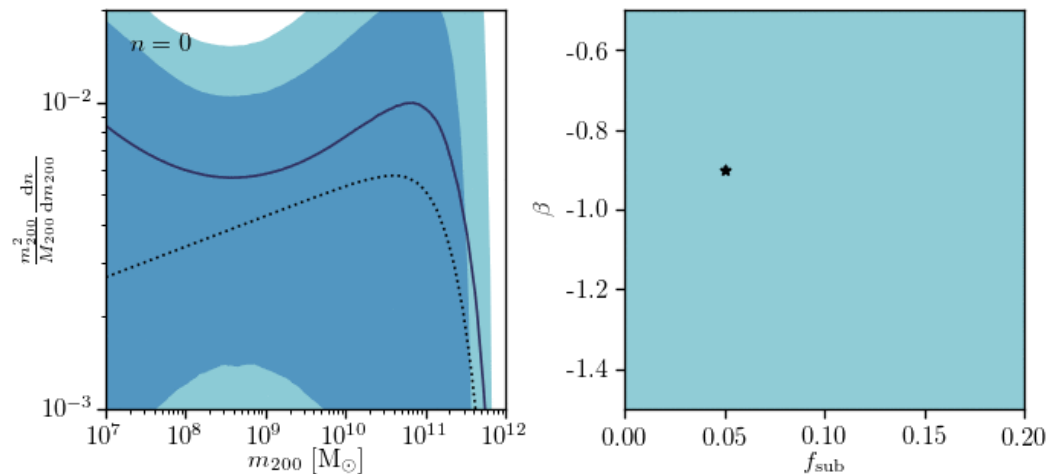
Algorithm 1 Optimization of $d(\mathbf{x}, \theta)$.

Inputs: Criterion ℓ (e.g., BCE)
Implicit generative model $p(\mathbf{x} | \theta)$
Prior $p(\theta)$
Outputs: Parameterized classifier $d_\phi(\mathbf{x}, \theta)$
Hyperparameters: Batch-size M

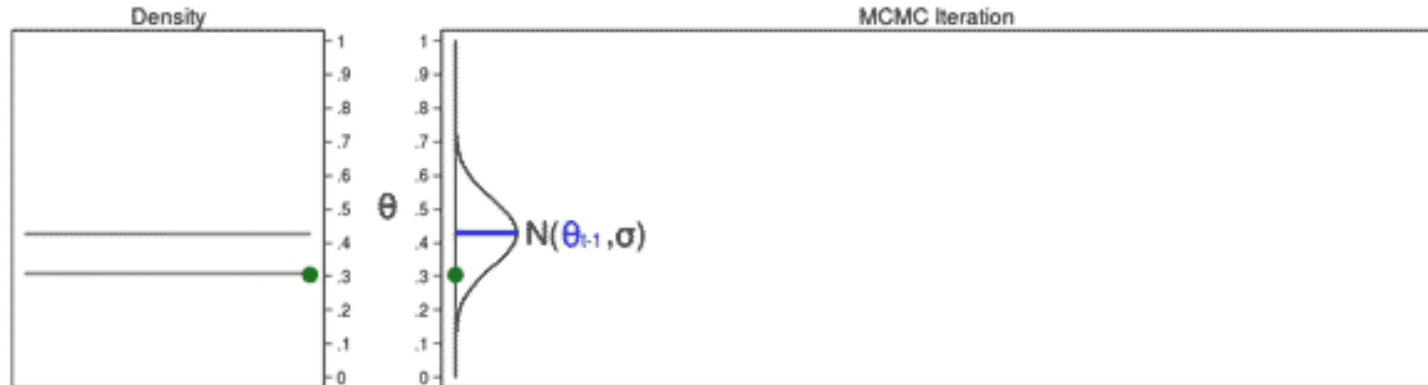
```
1: while not converged do
2:   Sample  $\theta \leftarrow \{\theta_m \sim p(\theta)\}_{m=1}^M$ 
3:   Sample  $\theta' \leftarrow \{\theta'_m \sim p(\theta)\}_{m=1}^M$ 
4:   Simulate  $\mathbf{x} \leftarrow \{\mathbf{x}_m \sim p(\mathbf{x} | \theta_m)\}_{m=1}^M$ 
5:    $\mathcal{L} \leftarrow \ell(d_\phi(\mathbf{x}, \theta), 1) + \ell(d_\phi(\mathbf{x}, \theta'), 0)$ 
6:    $\phi \leftarrow \text{OPTIMIZER}(\phi, \nabla_\phi \mathcal{L})$ 
7: end while
8: return  $d_\phi$ 
```



Bayesian inference of dark matter subhalo population parameters



MCMC posterior sampling



$$\text{Step 1: } r(\theta_{\text{new}}, \theta_{t-1}) = \frac{\text{Posterior}(\theta_{\text{new}})}{\text{Posterior}(\theta_{t-1})} = \frac{\text{Beta}(1,1,0.306) \times \text{Binomial}(10,4,0.306)}{\text{Beta}(1,1,0.429) \times \text{Binomial}(10,4,0.429)} = 0.834$$

$$\text{Step 2: Acceptance probability } \alpha(\theta_{\text{new}}, \theta_{t-1}) = \min\{r(\theta_{\text{new}}, \theta_{t-1}), 1\} = \min\{0.834, 1\} = 0.834$$

$$\text{Step 3: Draw } u \sim \text{Uniform}(0,1) = 0.617$$

$$\text{Step 4: If } u < \alpha(\theta_{\text{new}}, \theta_{t-1}) \rightarrow \text{If } 0.617 < 0.834 \quad \text{Then } \theta_t = \theta_{\text{new}} = 0.306 \\ \text{Otherwise } \theta_t = \theta_{t-1} = 0.429$$

Likelihood-free MCMC

MCMC samplers require the evaluation of the posterior ratios:

$$\begin{aligned}\frac{p(\theta_{\text{new}}|x)}{p(\theta_{t-1}|x)} &= \frac{p(x|\theta_{\text{new}})p(\theta_{\text{new}})/p(x)}{p(x|\theta_{t-1})p(\theta_{t-1})/p(x)} \\ &= \frac{p(x|\theta_{\text{new}})p(\theta_{\text{new}})}{p(x|\theta_{t-1})p(\theta_{t-1})} \\ &= r(x|\theta_{\text{new}}, \theta_{t-1}) \frac{p(\theta_{\text{new}})}{p(\theta_{t-1})}\end{aligned}$$

Again, MCMC samplers can be made **likelihood-free** by plugging a **learned approximation** $\hat{r}(x|\theta_{\text{new}}, \theta_{t-1})$ of the likelihood ratio.

For MCMC, best results are obtained when using ratios of likelihood-to-evidence ratios:

$$\hat{r}(x|\theta_{\text{new}}, \theta_{t-1}) = \frac{\hat{r}(x|\theta_{\text{new}})}{\hat{r}(x|\theta_{t-1})}$$

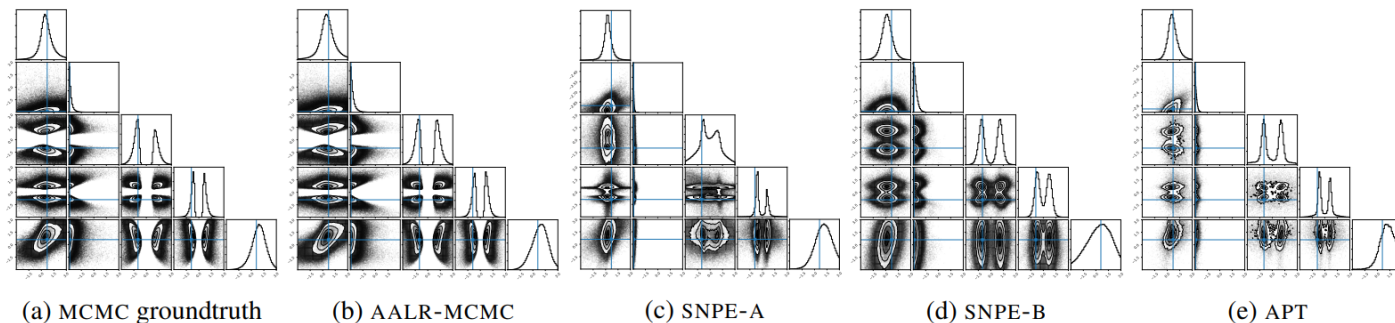
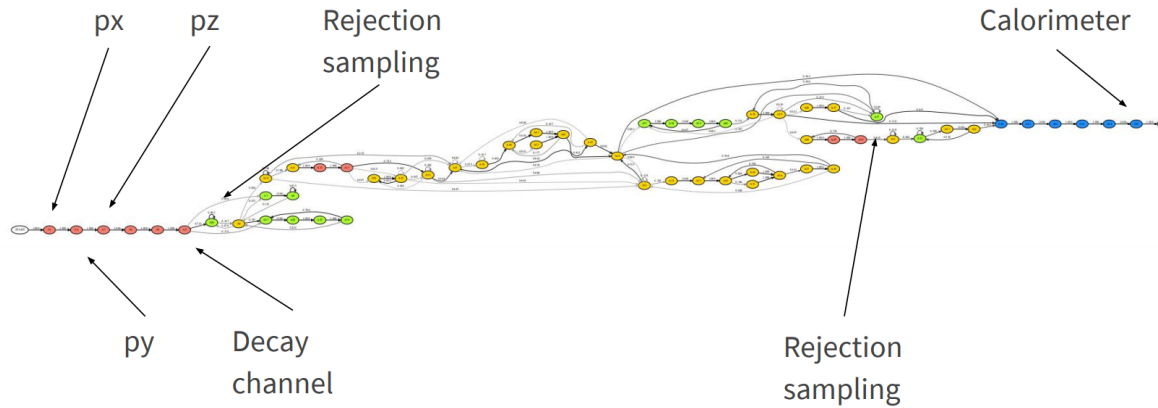


Figure 3: Posteriors from the tractable benchmark. The experiments are repeated 25 times and the approximate posteriors are subsampled from those runs. AALR-MCMC shares the same structure with the MCMC truth, demonstrating its accuracy. Some runs of the other methods were not consistent, contributing to the variance observed in Table 2.

Algorithm	MMD	ROC AUC
AALR-MCMC (ours)	0.05 ± 0.005	0.59 ± 0.0010
ABC ($\epsilon = 32$)	0.51 ± 0.001	0.99 ± 0.0001
ABC ($\epsilon = 16$)	0.50 ± 0.003	0.99 ± 0.0002
ABC ($\epsilon = 8$)	0.39 ± 0.001	0.99 ± 0.0003
ABC ($\epsilon = 4$)	0.29 ± 0.004	0.98 ± 0.0007
APT	0.17 ± 0.036	0.86 ± 0.0008
AALR-MCMC (LRT)	0.53 ± 0.004	0.99 ± 0.0001
SNPE-A	0.21 ± 0.070	0.97 ± 0.0098
SNPE-B	0.20 ± 0.061	0.92 ± 0.0181

Table 2: AALR-MCMC outperforms all other methods. Numerical errors introduced by MCMC might have contributed to these results. A comparison of the PDFs between the true posterior and our ratio estimator are shown in Figure 11 (Appendix D.2). The MMD scores are in agreement with [41].

Probabilistic programming



53

A probabilistic program defines a joint distribution of **unobserved** x and **observed** y variables $p(x, y)$.

Probabilistic programming extends ordinary programming with two added constructs:

- Sampling from distributions
- Conditioning random variables by specifying observed values

Inference engines give us distributions over unobserved variables, given observed variables (data)

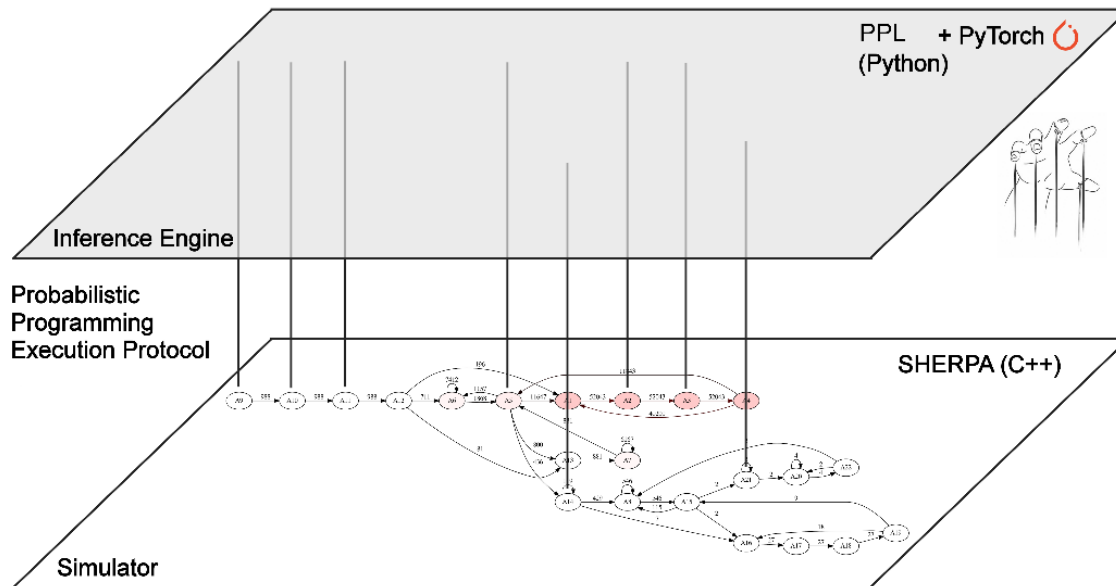
$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

Probabilistic programming languages

- Anglican (Clojure)
- Church (Scheme)
- Edward, TensorFlow Probability (Python, TensorFlow)
- Pyro (Python, PyTorch)

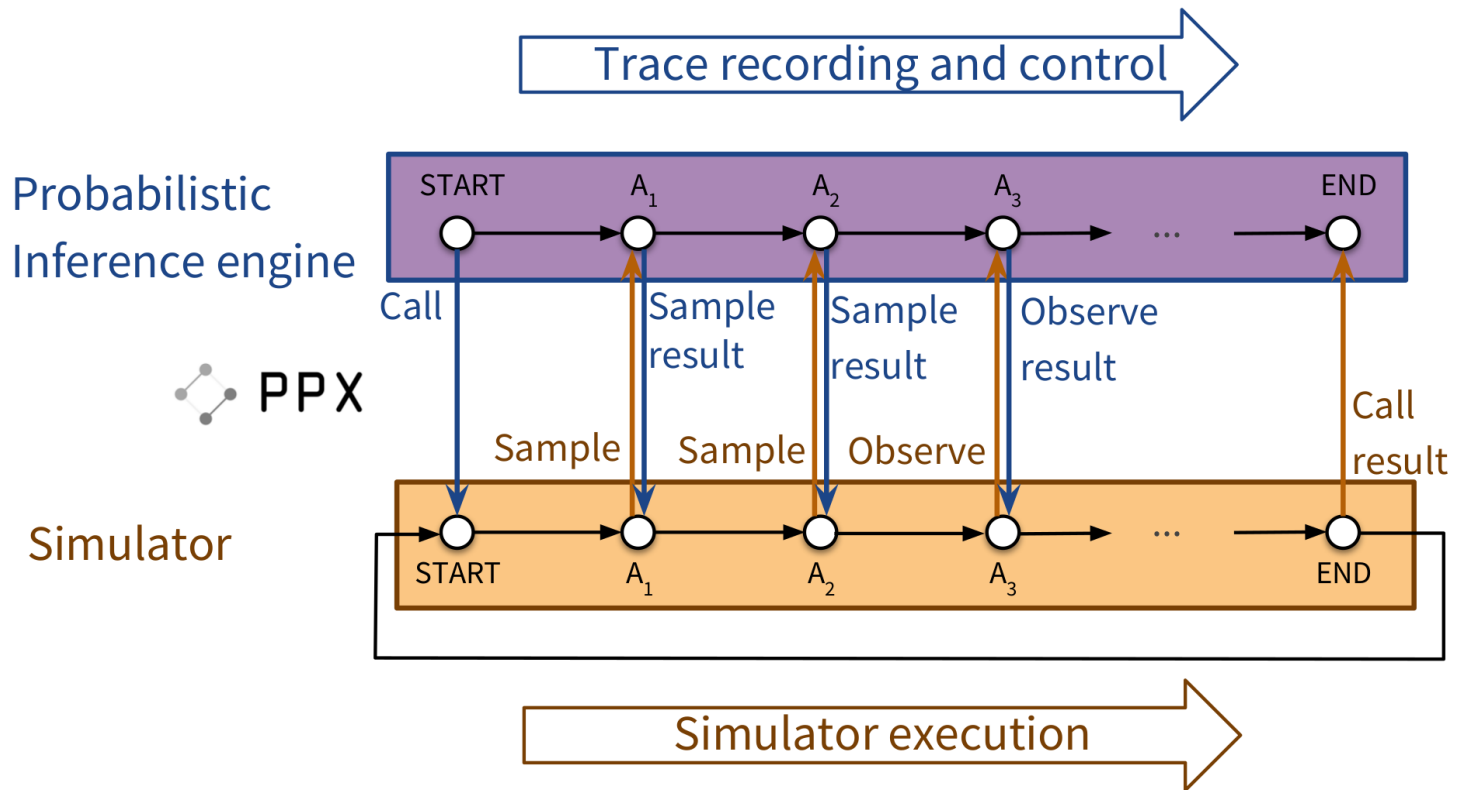
A stochastic simulator implicitly defines a probability distribution by sampling pseudo-random numbers.

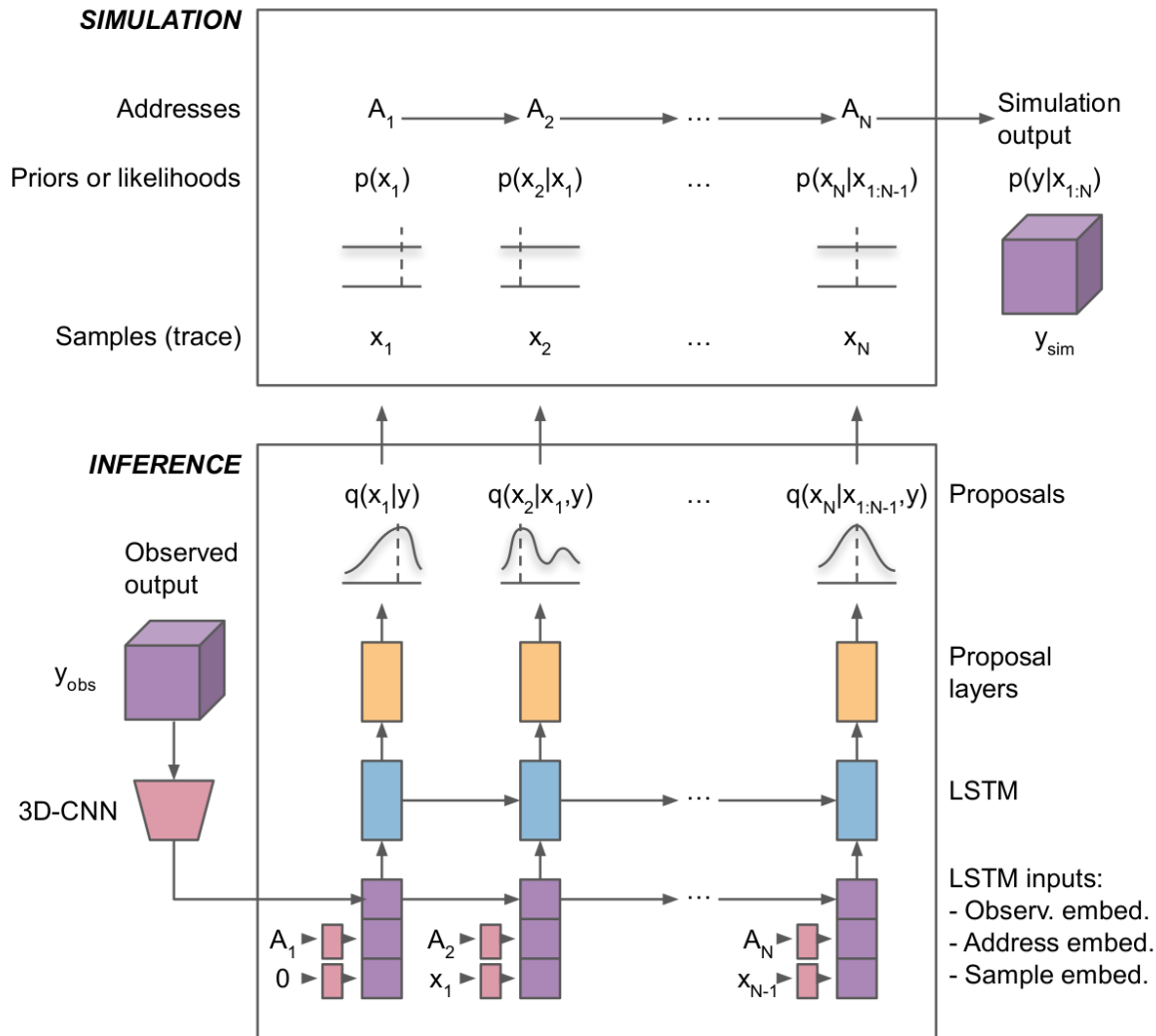
Scientific simulators are probabilistic programs!



Key idea

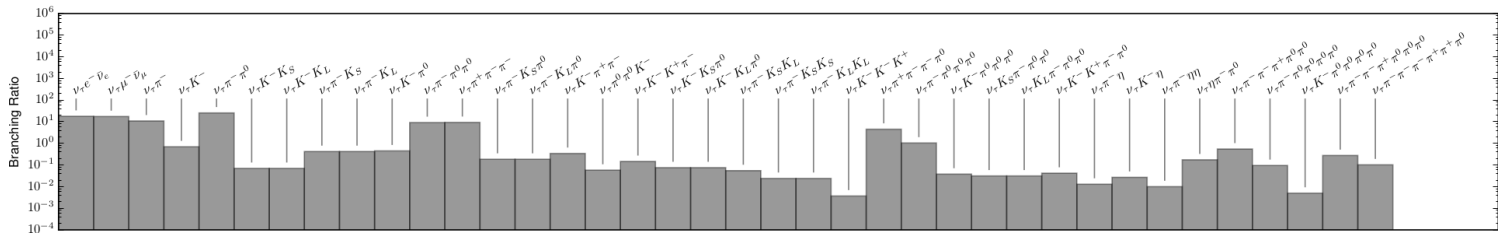
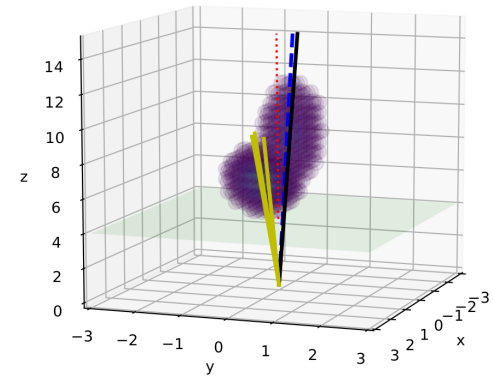
Let a neural network take full control of the internals of the simulation program by hijacking all calls to the random number generator.



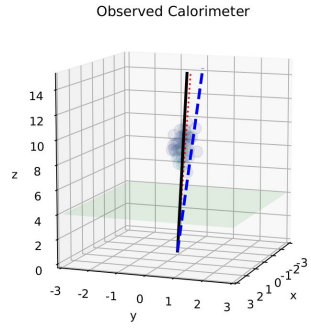


③ Taking control of Sherpa (particle physics simulator)

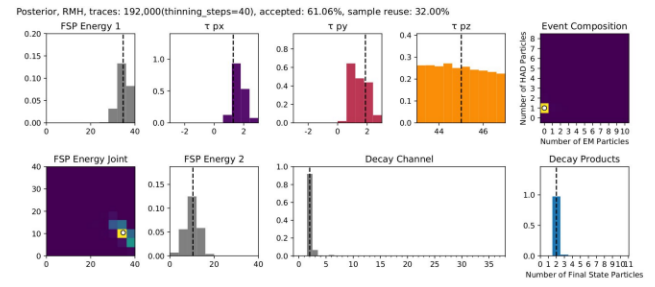
- τ decay in Sherpa, 38 decay channels, coupled with an approximate calorimeter simulation in C++.
- Observations are 3D calorimeter depositions.
- Latent variables (Monte Carlo truth) of interest: decay channel, px, py, pz momenta, final state momenta and IDs.



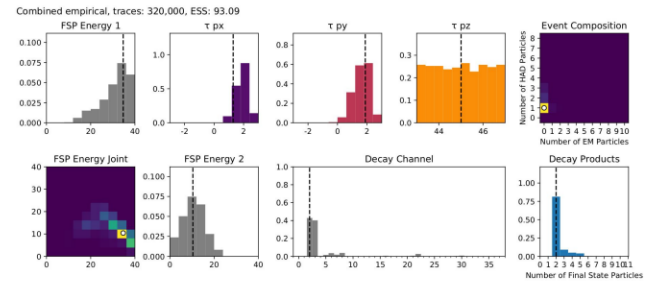
Inference results



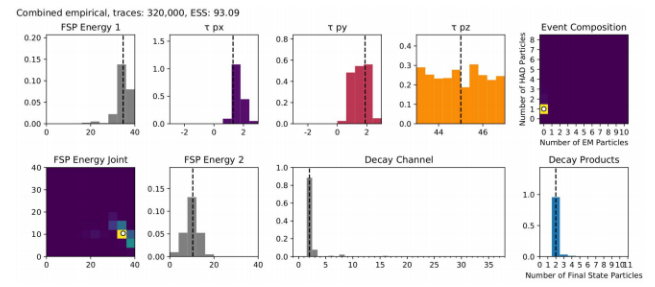
MCMC true posterior
(7.7M single node)

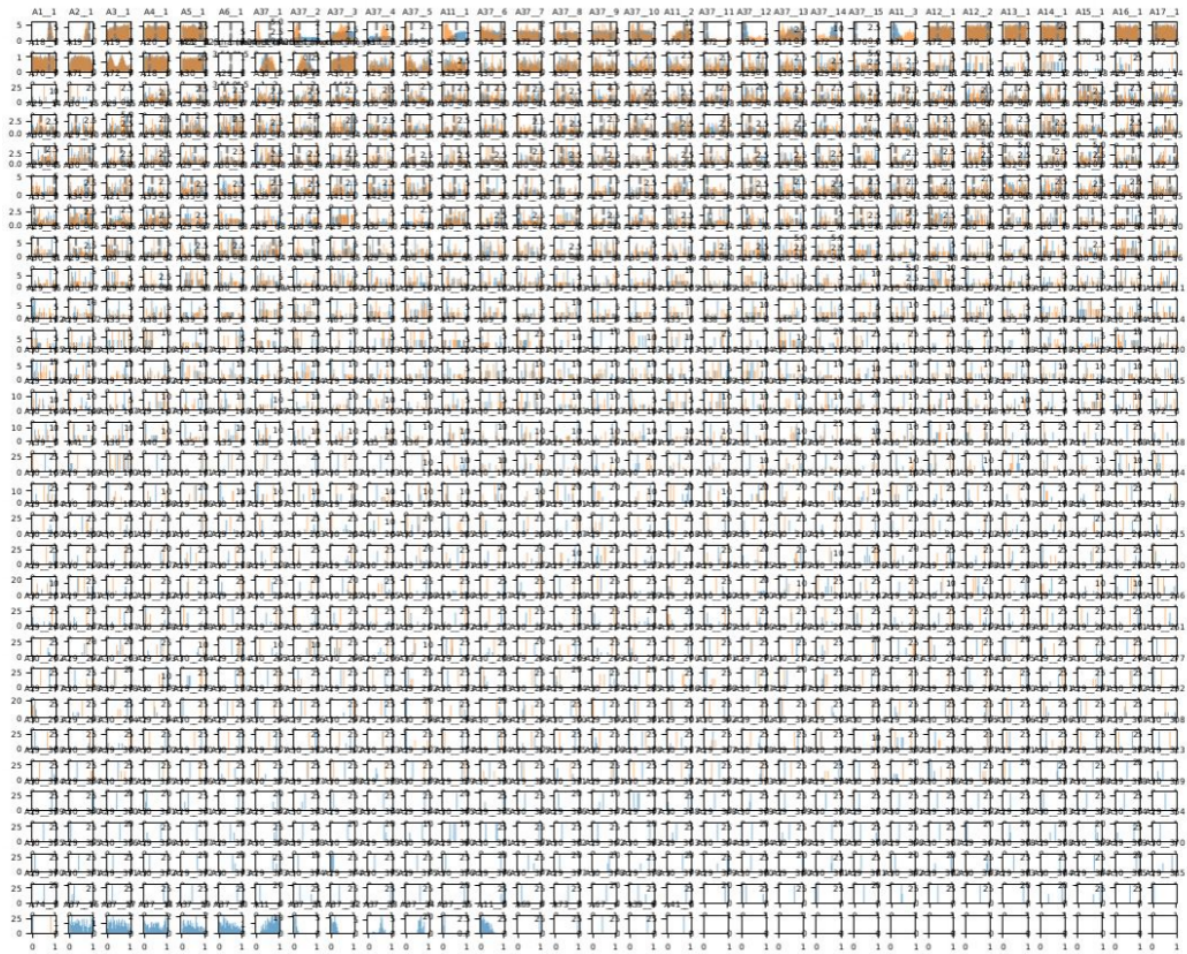


IC proposal
from trained NN



IC posterior
after importance
weighting

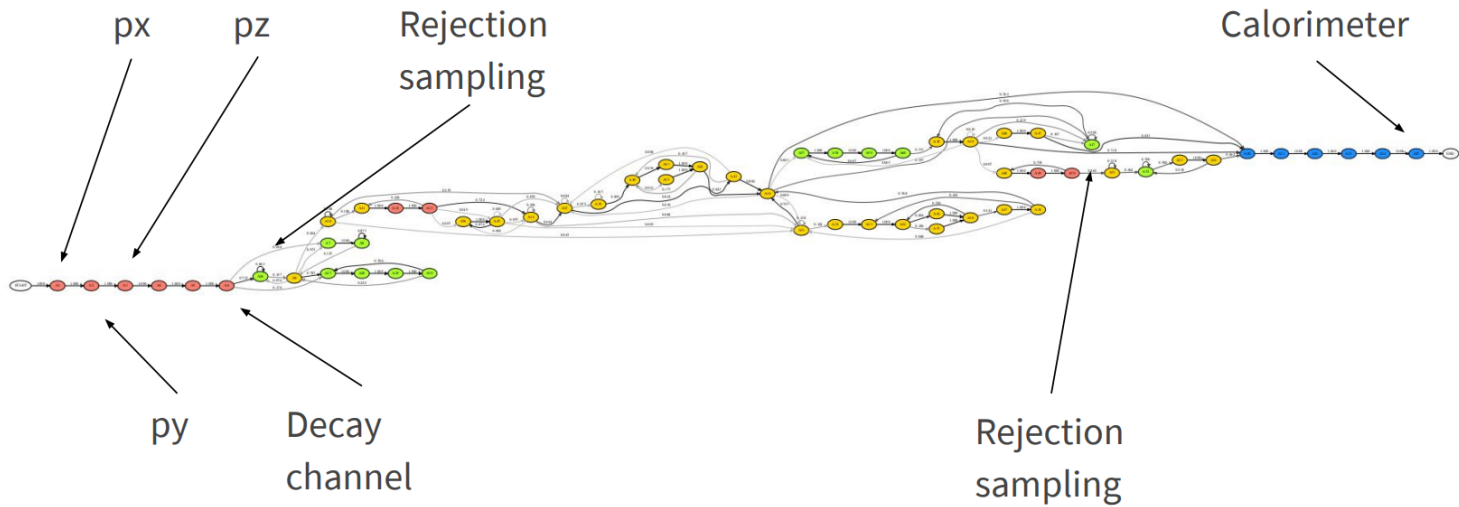


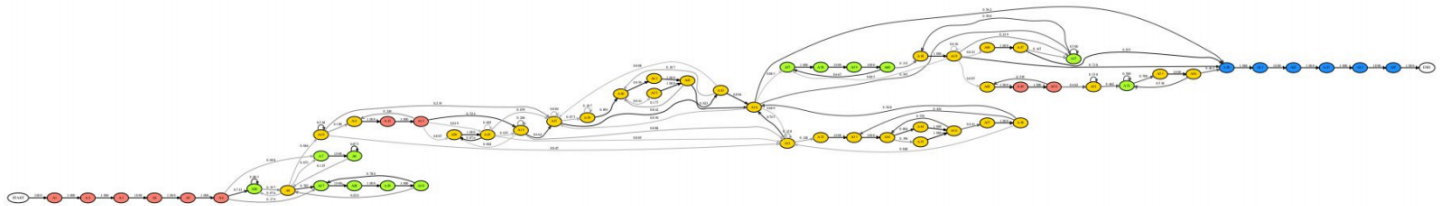


We obtain posteriors over the whole Sherpa address space, 1000s of addresses.

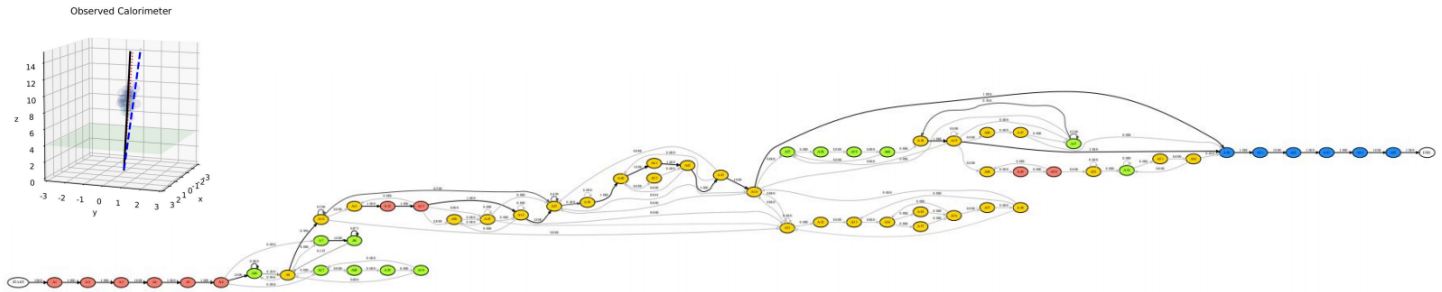
Interpretability

Latent probabilistic structure of the 250 most frequent trace types:





(a) Prior execution $p(\mathbf{x})$.



(b) Posterior execution $p(\mathbf{x}|\mathbf{y})$ conditioned on a given calorimeter observation \mathbf{y} .