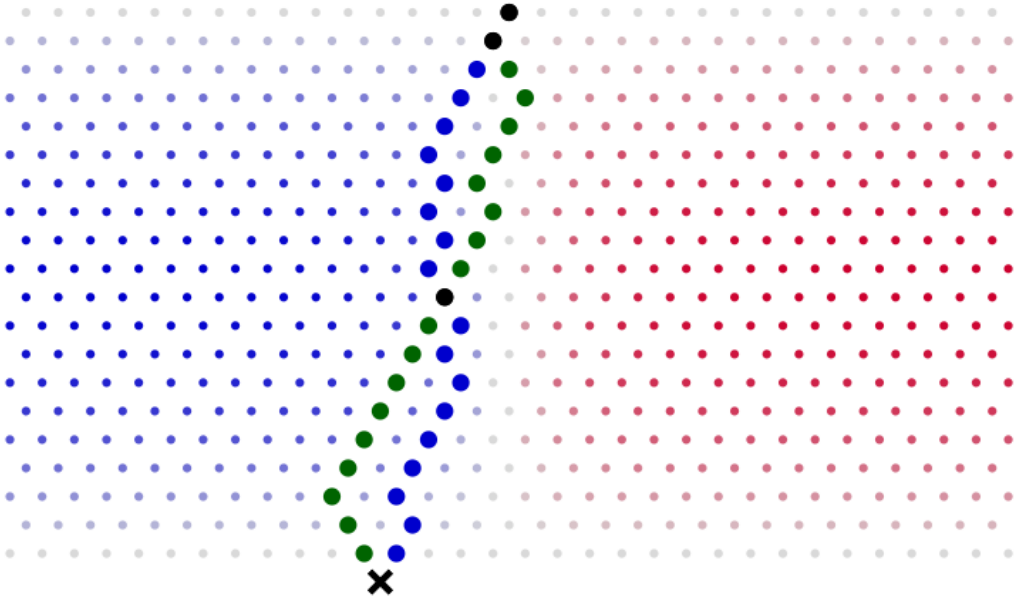
Advanced Machine Learning

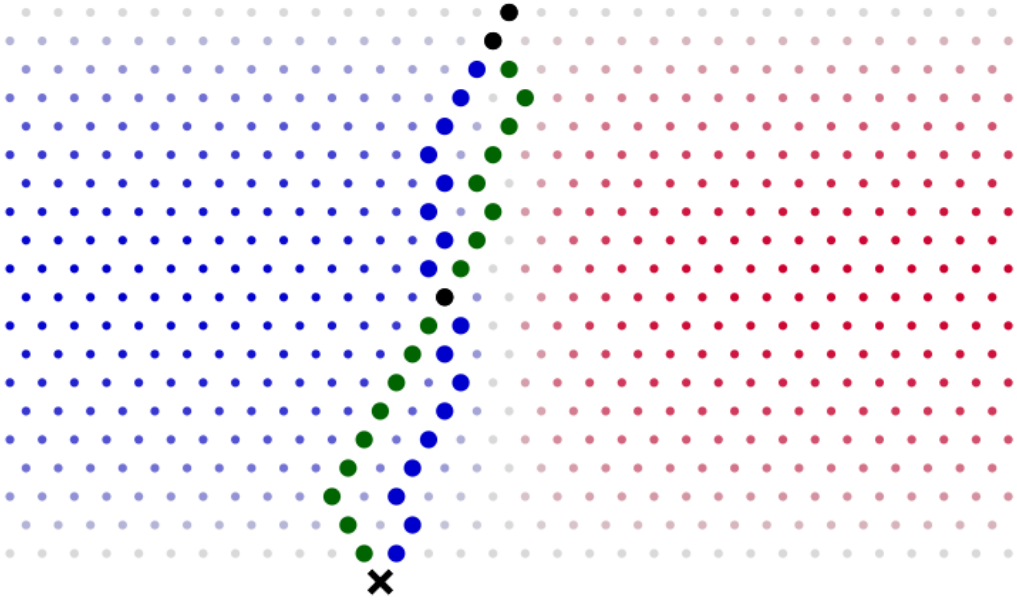
Neural Likelihood-free Inference

Prof. Gilles Louppe
g.louppe@uliege.be

LIÈGE université



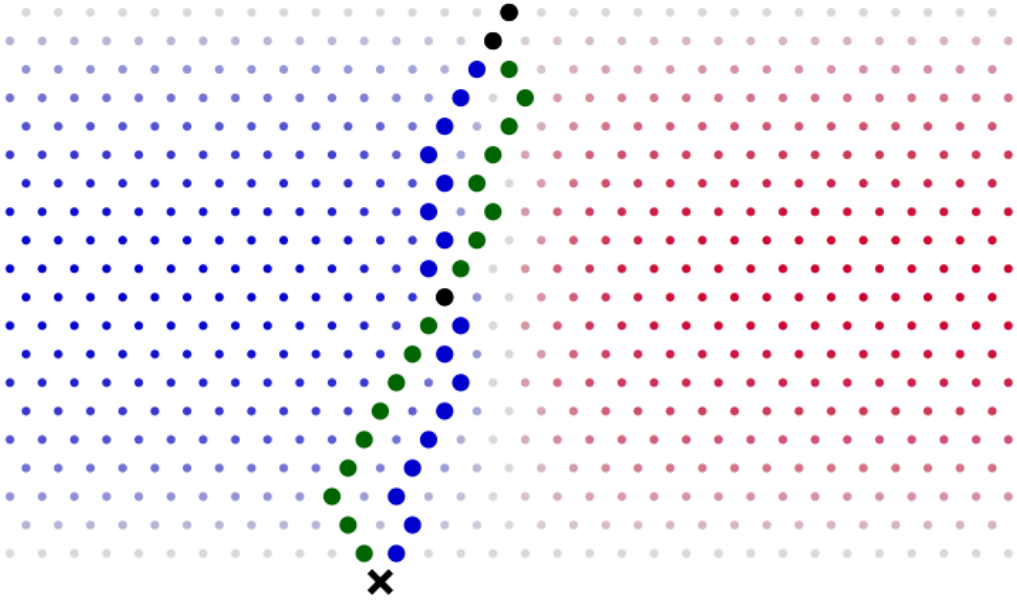
How to estimate the probability θ of going left when hitting a pin?



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

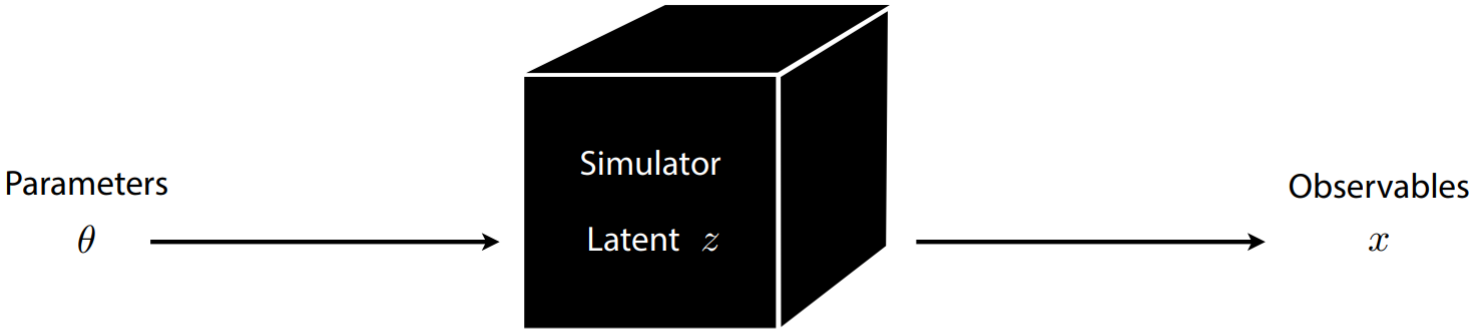


But what if we shift or remove some of the pins?

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

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

Inference in this context requires likelihood-free algorithms.

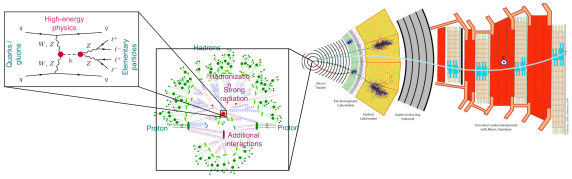


Prediction:
- Well-understood mechanistic model
- Simulator can generate samples

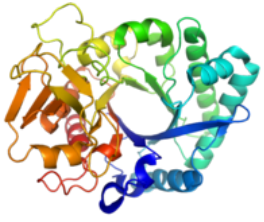
Inference:
- Likelihood function $p(x|\theta)$ is intractable
- Inference based on estimator $\hat{p}(x|\theta)$

Credits: Johann Brehmer.

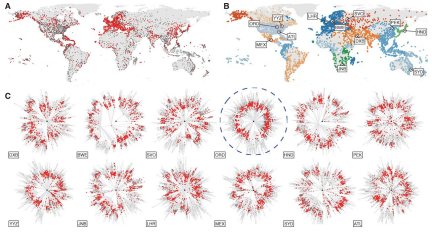
A thriving field of research



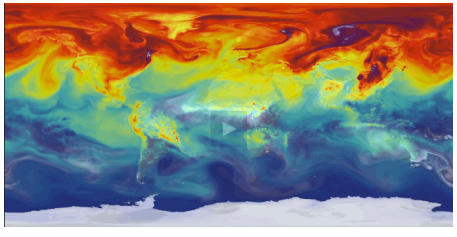
Particle physics



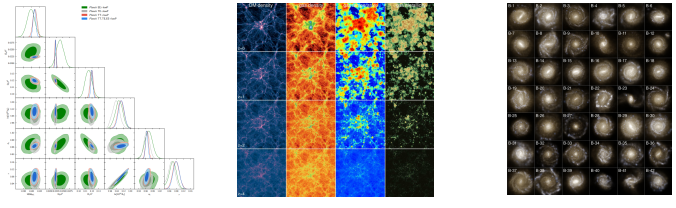
Protein folding



Epidemiology



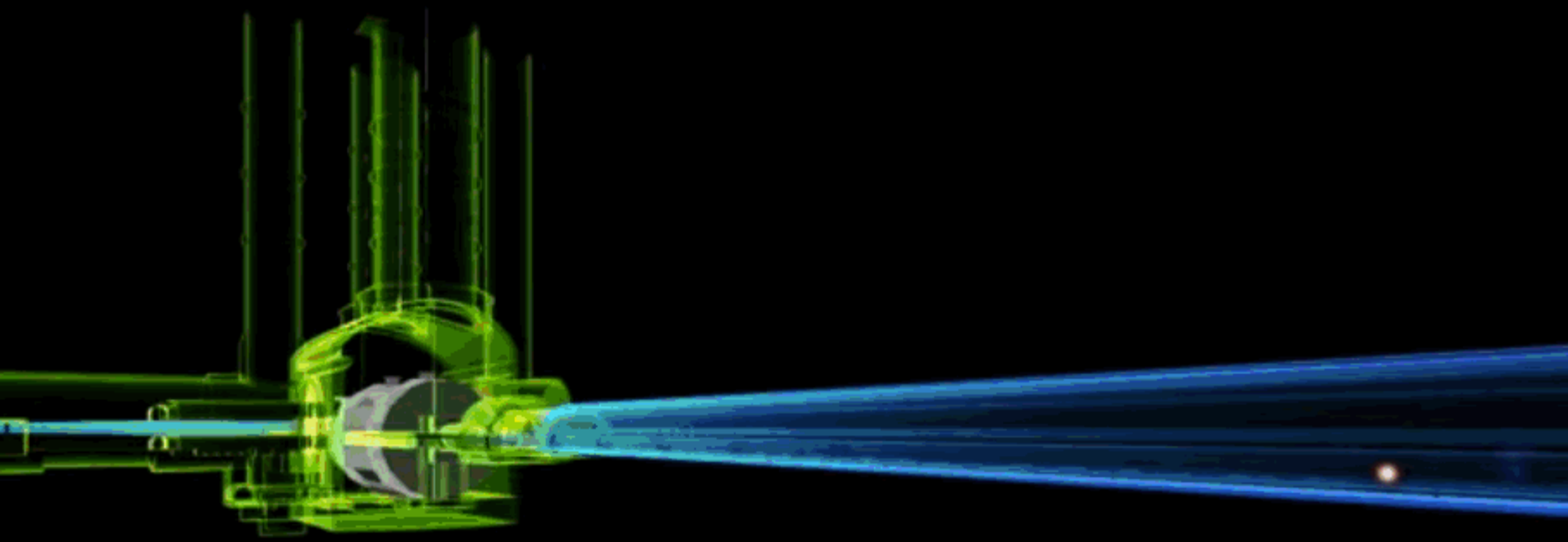
Climate science

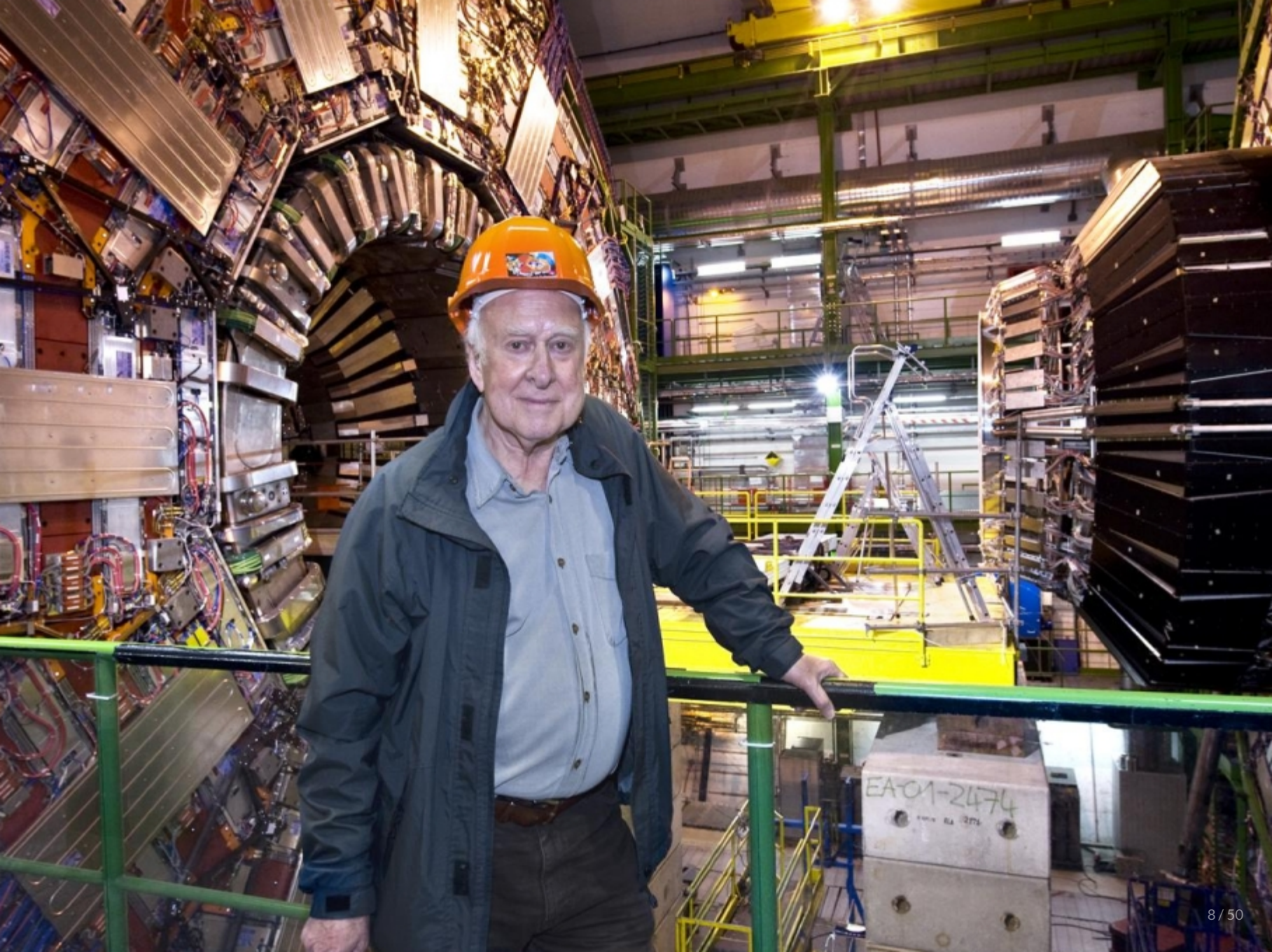


Astrophysics and cosmology

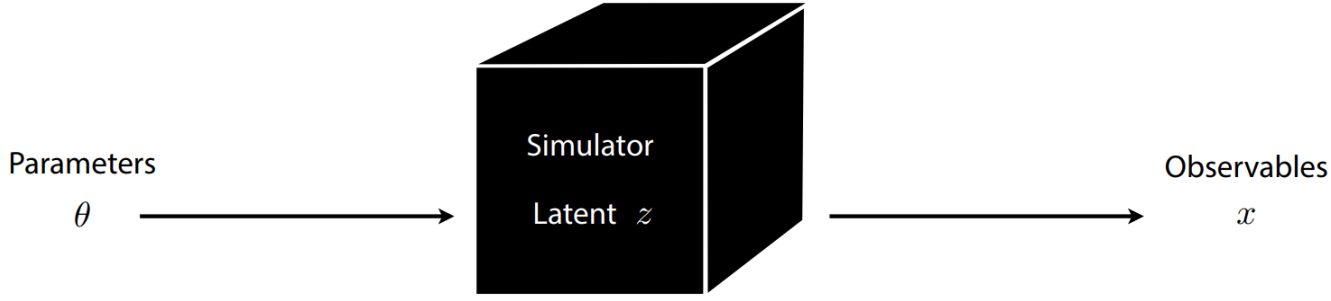
(... and many others!)

$$\mathcal{L}_{SM} = -\tfrac{1}{2}\partial_\nu g^a_\mu \partial_\nu g^a_\mu - g_s f^{abc}\partial_\mu g^a_\nu g^b_\mu g^c_\nu - \tfrac{1}{4}g_s^2 f^{abc}f^{ade}g^b_\mu g^c_\nu g^d_\mu g^e_\nu - \partial_\nu W^+_\mu \partial_\nu W^-_\mu - M^2 W^+_\mu W^-_\mu - \tfrac{1}{2}\partial_\nu Z^0_\mu \partial_\nu Z^0_\mu - \tfrac{1}{2c_w^2}M^2 Z^0_\mu Z^0_\mu - \tfrac{1}{2}\partial_\mu A_\nu \partial_\mu A_\nu - igc_w(\partial_\nu Z^0_\mu (W^+_\mu W^-_\nu - W^+_\nu W^-_\mu) - Z^0_\nu (W^+_\mu \partial_\nu W^-_\mu - W^-_\mu \partial_\nu W^+_\mu) + Z^0_\mu (W^+_\nu \partial_\nu W^-_\mu - W^-_\nu \partial_\nu W^+_\mu)) - igs_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^-_\nu \partial_\nu W^+_\mu)) - \tfrac{1}{2}g^2 W^+_\mu W^-_\mu W^+_\nu W^-_\nu + \tfrac{1}{2}g^2 W^+_\mu W^-_\nu W^+_\mu W^-_\nu + g^2 c_w^2 (Z^0_\mu W^+_\mu Z^0_\nu W^-_\nu - Z^0_\mu Z^0_\mu W^+_\nu W^-_\nu) + g^2 s_w^2 (A_\mu W^+_\mu A_\nu W^-_\nu - A_\mu A_\mu W^+_\nu W^-_\nu) + g^2 s_w c_w (A_\mu Z^0_\nu (W^+_\mu W^-_\nu - W^+_\nu W^-_\mu) - 2A_\mu Z^0_\mu W^+_\nu W^-_\nu) - \tfrac{1}{2}\partial_\mu H \partial_\mu H - 2M^2 \alpha_h H^2 - \partial_\mu \phi^+ \partial_\mu \phi^- - \tfrac{1}{2}\partial_\mu \phi^0 \partial_\mu \phi^0 - \beta_h \left(\tfrac{2M^2}{g^2} + \tfrac{2M}{g}H + \tfrac{1}{2}(H^2 + \phi^0\phi^0 + 2\phi^+\phi^-) \right) + \tfrac{2M^4}{g^2}\alpha_h - g\alpha_h M (H^3 + H\phi^0\phi^0 + 2H\phi^+\phi^-) - \tfrac{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 - \tfrac{1}{2}g\tfrac{M}{c_w^2}Z^0_\mu Z^0_\mu H - \tfrac{1}{2}ig\left(W^+_\mu(\phi^0\partial_\mu\phi^- - \phi^-\partial_\mu\phi^0) - W^-_\mu(\phi^0\partial_\mu\phi^+ - \phi^+\partial_\mu\phi^0)\right) + \tfrac{1}{2}g\left(W^+_\mu(H\partial_\mu\phi^- - \phi^-\partial_\mu H) + W^-_\mu(H\partial_\mu\phi^+ - \phi^+\partial_\mu H)\right) + \tfrac{1}{2}g\tfrac{1}{c_w}(Z^0_\mu(H\partial_\mu\phi^0 - \phi^0\partial_\mu H) + M\left(\tfrac{1}{c_w}Z^0_\mu\partial_\mu\phi^0 + W^+_\mu\partial_\mu\phi^- + W^-_\mu\partial_\mu\phi^+\right) - ig\tfrac{s_w^2}{c_w}MZ^0_\mu(W^+_\mu\phi^- - W^-_\mu\phi^+) + igs_wMA_\mu(W^+_\mu\phi^- - W^-_\mu\phi^+) - ig\tfrac{1-2c_w^2}{2c_w}Z^0_\mu(\phi^+\partial_\mu\phi^- - \phi^-\partial_\mu\phi^+) + igs_wA_\mu(\phi^+\partial_\mu\phi^- - \phi^-\partial_\mu\phi^+) - \tfrac{1}{4}g^2W^+_\mu W^-_\mu(H^2 + (\phi^0)^2 + 2\phi^+\phi^-) - \tfrac{1}{8}g^2\tfrac{1}{c_w^2}Z^0_\mu Z^0_\mu(H^2 + (\phi^0)^2 + 2(2s_w^2 - 1)^2\phi^+\phi^-) - \tfrac{1}{2}g^2\tfrac{s_w^2}{c_w}Z^0_\mu\phi^0(W^+_\mu\phi^- + W^-_\mu\phi^+) - \tfrac{1}{2}ig^2\tfrac{s_w^2}{c_w}Z^0_\mu H(W^+_\mu\phi^- - W^-_\mu\phi^+) + \tfrac{1}{2}g^2 s_w A_\mu\phi^0(W^+_\mu\phi^- + W^-_\mu\phi^+) + \tfrac{1}{2}ig^2 s_w A_\mu H(W^+_\mu\phi^- - W^-_\mu\phi^+) - g^2\tfrac{s_w}{c_w}(2c_w^2 - 1)Z^0_\mu A_\mu\phi^+\phi^- - g^2 s_w^2 A_\mu A_\mu\phi^+\phi^- + \tfrac{1}{2}ig_s\lambda^a_{ij}(\bar{q}^\sigma_i\gamma^\mu q^\sigma_j)g^a_\mu - \bar{e}^\lambda(\gamma\partial + m^\lambda_e)e^\lambda - \bar{\nu}^\lambda(\gamma\partial + m^\lambda_\nu)\nu^\lambda - \bar{u}^\lambda_j(\gamma\partial + m^\lambda_u)u^\lambda_j - \bar{d}^\lambda_j(\gamma\partial + m^\lambda_d)d^\lambda_j + igs_wA_\mu\left(-(\bar{e}^\lambda\gamma^\mu e^\lambda) + \tfrac{2}{3}(\bar{u}^\lambda_j\gamma^\mu u^\lambda_j) - \tfrac{1}{3}(\bar{d}^\lambda_j\gamma^\mu d^\lambda_j)\right) + \tfrac{ig}{4c_w}Z^0_\mu\{(\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}^\lambda_j\gamma^\mu(\tfrac{4}{3}s_w^2 - 1 - \gamma^5)d^\lambda_j) + (\bar{u}^\lambda_j\gamma^\mu(1 - \tfrac{8}{3}s_w^2 + \gamma^5)u^\lambda_j)\} + \tfrac{ig}{2\sqrt{2}}W^+_\mu\left((\bar{\nu}^\lambda\gamma^\mu(1+\gamma^5)U^{lep}_{\lambda\kappa}e^\kappa) + (\bar{u}^\lambda_j\gamma^\mu(1+\gamma^5)C_{\lambda\kappa}d^\kappa_j)\right) + \tfrac{ig}{2\sqrt{2}}W^-_\mu\left((\bar{e}^\kappa U^{lep\dagger}_{\kappa\lambda}\gamma^\mu(1+\gamma^5)\nu^\lambda) + (\bar{d}^\kappa_j C^\dagger_{\kappa\lambda}\gamma^\mu(1+\gamma^5)u^\lambda_j)\right) + \tfrac{ig}{2M\sqrt{2}}\phi^+\left(-m^\kappa_e(\bar{\nu}^\lambda U^{lep}_{\lambda\kappa}(1-\gamma^5)e^\kappa) + m^\lambda_\nu(\bar{\nu}^\lambda U^{lep}_{\lambda\kappa}(1+\gamma^5)e^\kappa)\right) + \tfrac{ig}{2M\sqrt{2}}\phi^-\left(m^\lambda_e(\bar{e}^\lambda U^{lep\dagger}_{\lambda\kappa}(1+\gamma^5)\nu^\kappa) - m^\kappa_\nu(\bar{e}^\lambda U^{lep\dagger}_{\lambda\kappa}(1-\gamma^5)\nu^\kappa)\right) - \tfrac{g}{2}\tfrac{m^\lambda_e}{M}H(\bar{\nu}^\lambda\nu^\lambda) - \tfrac{g}{2}\tfrac{m^\lambda_e}{M}H(\bar{e}^\lambda e^\lambda) + \tfrac{ig}{2}\tfrac{m^\lambda_e}{M}\phi^0(\bar{\nu}^\lambda\gamma^5\nu^\lambda) - \tfrac{ig}{2}\tfrac{m^\lambda_e}{M}\phi^0(\bar{e}^\lambda\gamma^5 e^\lambda) - \tfrac{1}{4}\bar{\nu}_\lambda M^R_{\lambda\kappa}(1-\gamma_5)\hat{\nu}_\kappa - \tfrac{1}{4}\bar{\nu}_\lambda M^R_{\lambda\kappa}(1-\gamma_5)\hat{\nu}_\kappa + \tfrac{ig}{2M\sqrt{2}}\phi^+\left(-m^\kappa_d(\bar{u}^\lambda_j C_{\lambda\kappa}(1-\gamma^5)d^\kappa_j) + m^\lambda_u(\bar{u}^\lambda_j C_{\lambda\kappa}(1+\gamma^5)d^\kappa_j\right) + \tfrac{ig}{2M\sqrt{2}}\phi^-\left(m^\lambda_d(\bar{d}^\lambda_j C^\dagger_{\lambda\kappa}(1+\gamma^5)u^\kappa_j) - m^\kappa_u(\bar{d}^\lambda_j C^\dagger_{\lambda\kappa}(1-\gamma^5)u^\kappa_j\right) - \tfrac{g}{2}\tfrac{m^\lambda_u}{M}H(\bar{u}^\lambda_j u^\lambda_j) - \tfrac{g}{2}\tfrac{m^\lambda_u}{M}H(\bar{d}^\lambda_j d^\lambda_j) + \tfrac{ig}{2}\tfrac{m^\lambda_u}{M}\phi^0(\bar{u}^\lambda_j\gamma^5 u^\lambda_j) - \tfrac{ig}{2}\tfrac{m^\lambda_d}{M}\phi^0(\bar{d}^\lambda_j\gamma^5 d^\lambda_j) + \bar{G}^a\partial^2 G^a + g_s f^{abc}\partial_\mu\bar{G}^a G^b g^c_\mu + \bar{X}^+(\partial^2 - M^2)X^+ + \bar{X}^-(\partial^2 - M^2)X^- + \bar{X}^0(\partial^2 - \tfrac{M^2}{c_w^2})X^0 + \bar{Y}\partial^2 Y + igc_w W^+_\mu(\partial_\mu\bar{X}^0X^- - \partial_\mu\bar{X}^+X^0) + igs_w W^+_\mu(\partial_\mu\bar{Y}X^- - \partial_\mu\bar{X}^+Y) + igc_w W^-_\mu(\partial_\mu\bar{X}^-X^0 - \partial_\mu\bar{X}^0X^+) + igs_w W^-_\mu(\partial_\mu\bar{X}^-Y - \partial_\mu\bar{Y}X^+) + igc_w Z^0_\mu(\partial_\mu\bar{X}^+X^+ - \partial_\mu\bar{X}^-X^-) + igs_w A_\mu(\partial_\mu\bar{X}^+X^+ - \partial_\mu\bar{X}^-X^-) - \tfrac{1}{2}gM\left(\bar{X}^+X^+H + \bar{X}^-X^-H + \tfrac{1}{c_w^2}\bar{X}^0X^0H\right) + \tfrac{1-2c_w^2}{2c_w}igM\left(\bar{X}^+X^0\phi^+ - \bar{X}^-X^0\phi^-\right) + \tfrac{1}{2c_w}igM\left(\bar{X}^0X^-\phi^+ - \bar{X}^0X^+\phi^-\right) + igMs_w\left(\bar{X}^0X^-\phi^+ - \bar{X}^0X^+\phi^-\right) + \tfrac{1}{2}igM\left(\bar{X}^+X^+\phi^0 - \bar{X}^-X^-\phi^0\right).$$



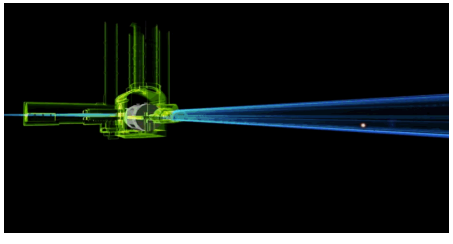

EA01-2474

Particle physics

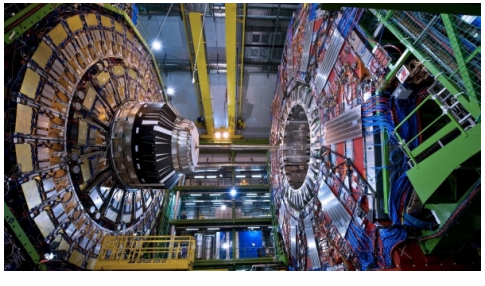


SM with parameters θ

Simulated observables x



Real observations x_{obs}



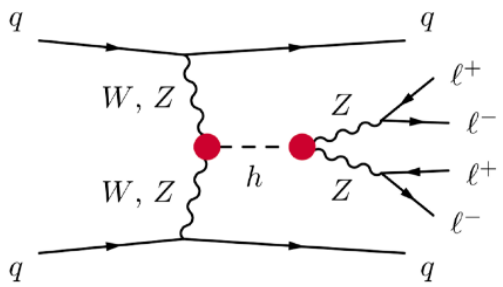
Latent variables

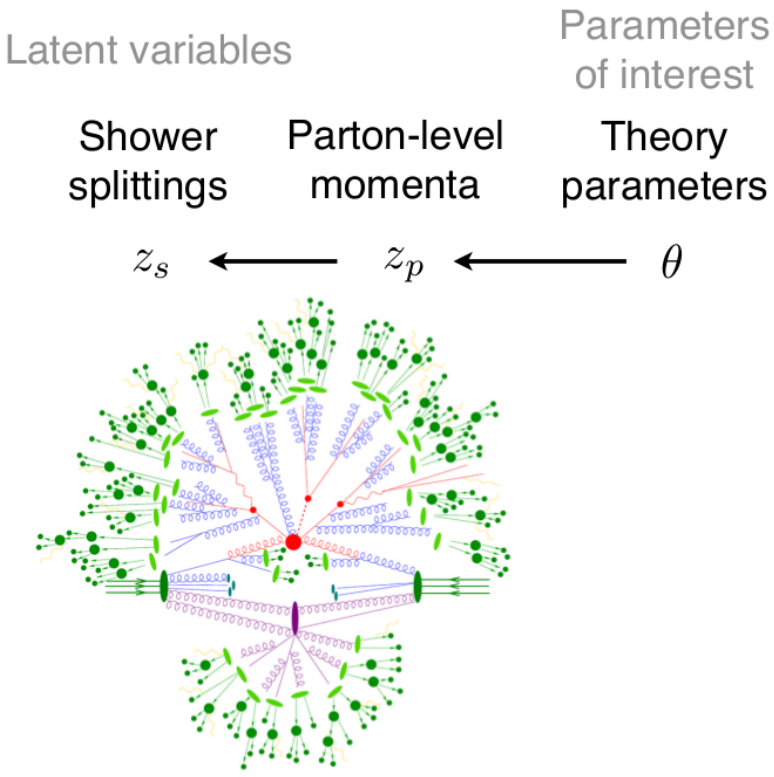
Parameters of interest

Parton-level momenta

Theory parameters

$z_p \longleftarrow \theta$





Credits: Johann Brehmer.

Latent variables

Parameters of interest

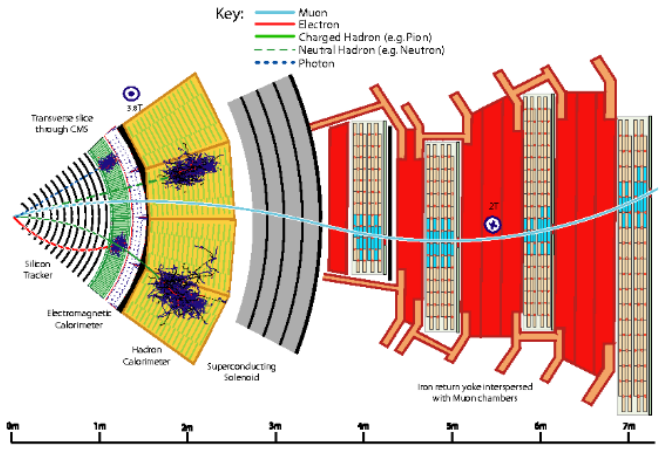
Detector interactions

Shower splittings

Parton-level momenta

Theory parameters

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



Credits: Johann Brehmer.

Features

Latent variables

Parameters of interest

Observables

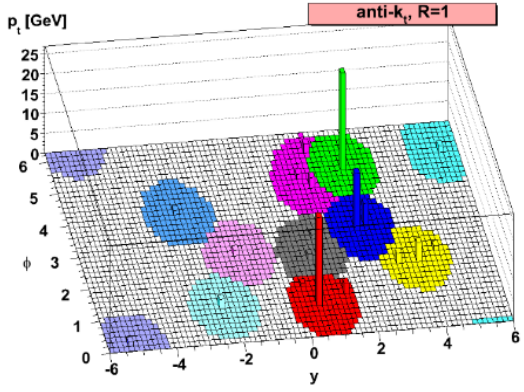
Detector interactions

Shower splittings

Parton-level momenta

Theory parameters

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



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

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

Ingredients

Statistical inference requires the computation of key ingredients, such as

- the likelihood $p(x|\theta)$,
- the likelihood ratio $r(x|\theta_0, \theta_1) = \frac{p(x|\theta_0)}{p(x|\theta_1)}$,
- or the posterior $p(\theta|x)$.

In the simulator-based scenario, each of these ingredients can be approximated with modern machine learning techniques, even if none are tractable during training!

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

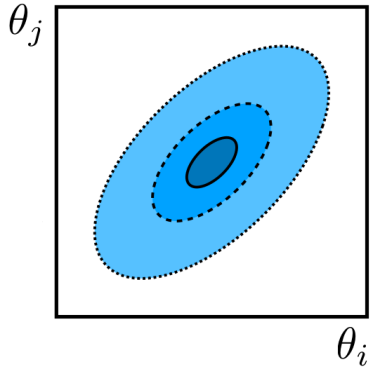
Frequentist inference

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.*

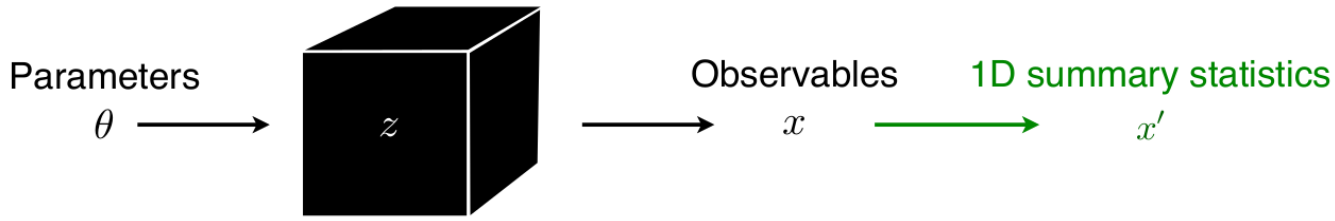
(*Communicated by* K. PEARSON, *F.R.S.*)

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

CONTENTS.

PAGE.

I. Introductory 289
II. Outline of General Theory 293



Define a projection function $s : \mathcal{X} \to \mathbb{R}$ 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

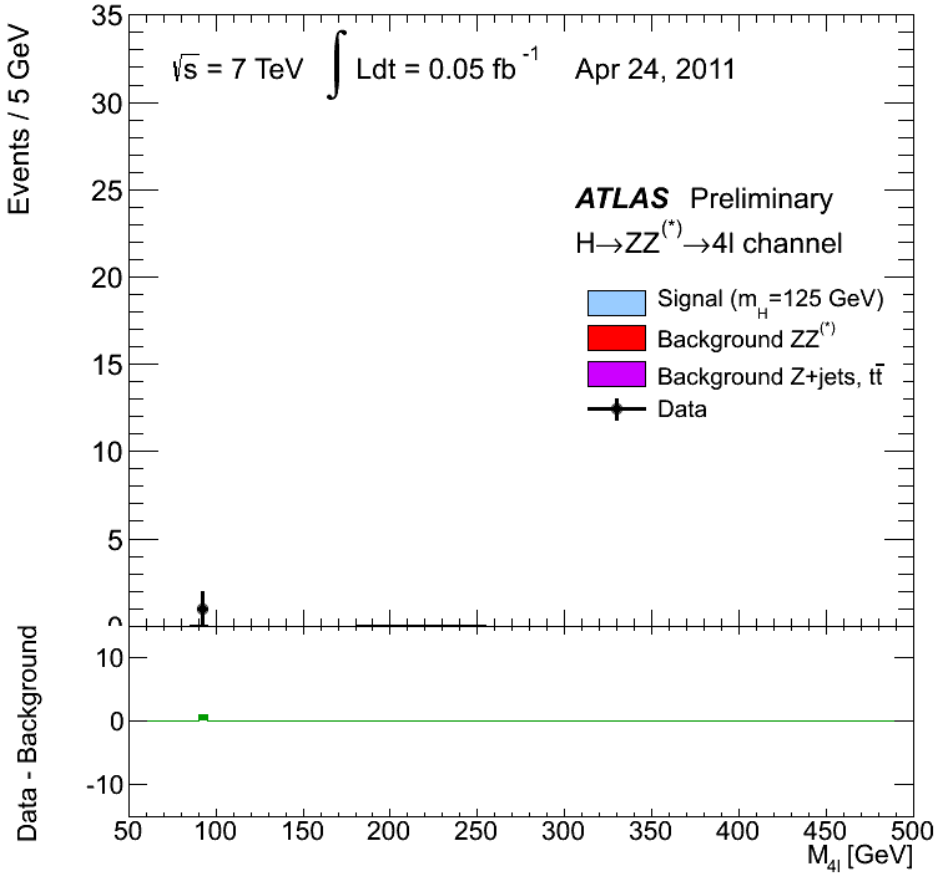
Consider the test statistic

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

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

When $N \to \infty, q(\theta) \sim \chi_2$.

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.

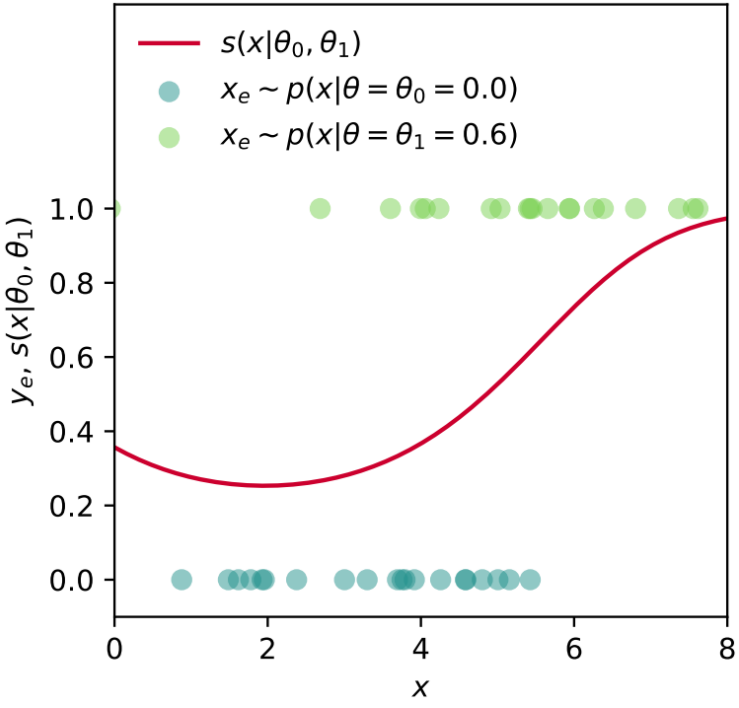


Discovery of the Higgs boson at 5-σ

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.



Cranmer, Pavez and Louppe, 2015 [arXiv:1506.02169].

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

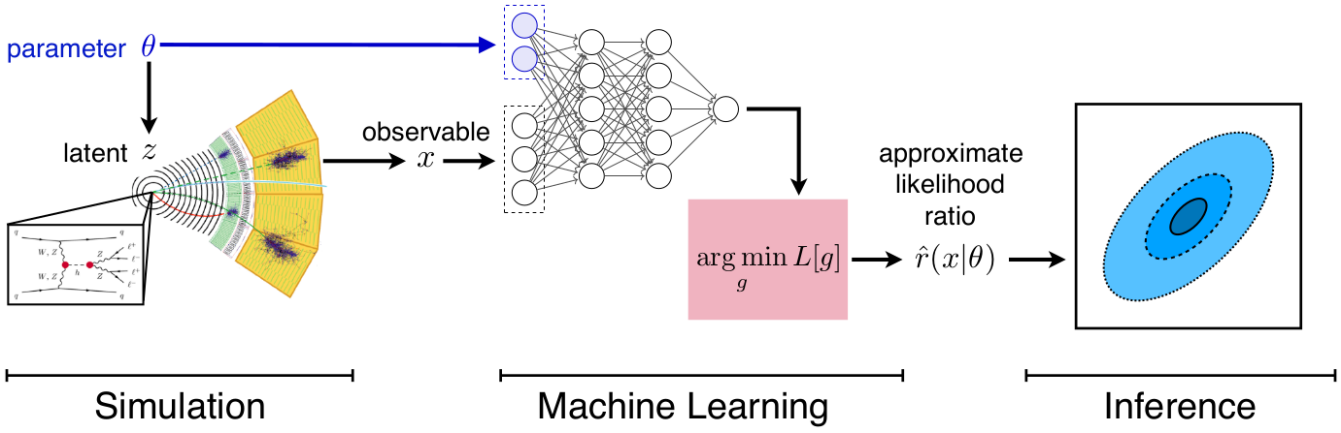
$$\hat{s}(x) \approx s^*(x) = \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)}.$$

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.

Cranmer, Pavez and Louppe, 2015 [arXiv:1506.02169].



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

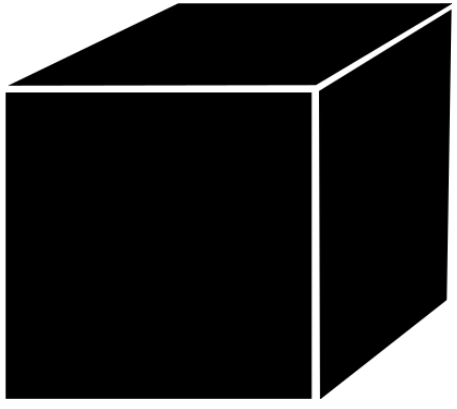
$$\hat{r}(x|\theta_0, \theta_{\text{ref}}) = \frac{1 - \hat{s}(x|\theta_0, \theta_{\text{ref}})}{\hat{s}(x|\theta_0, \theta_{\text{ref}})}$$

such that for any (θ_0, θ_1),

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

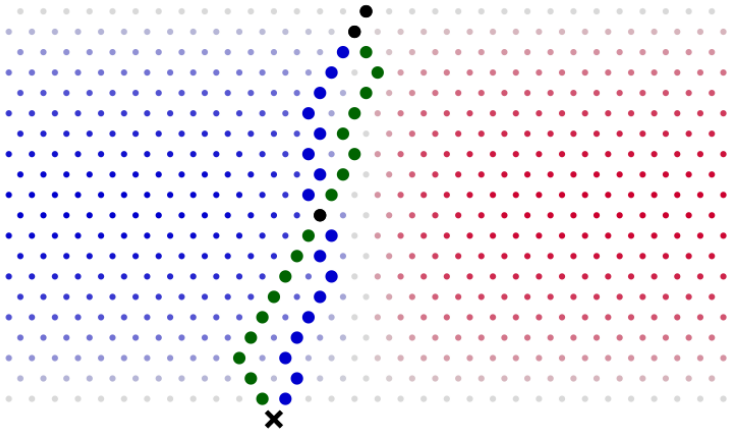
Cranmer, Pavez and Louppe, 2015 [arXiv:1506.02169].

Opening the black box



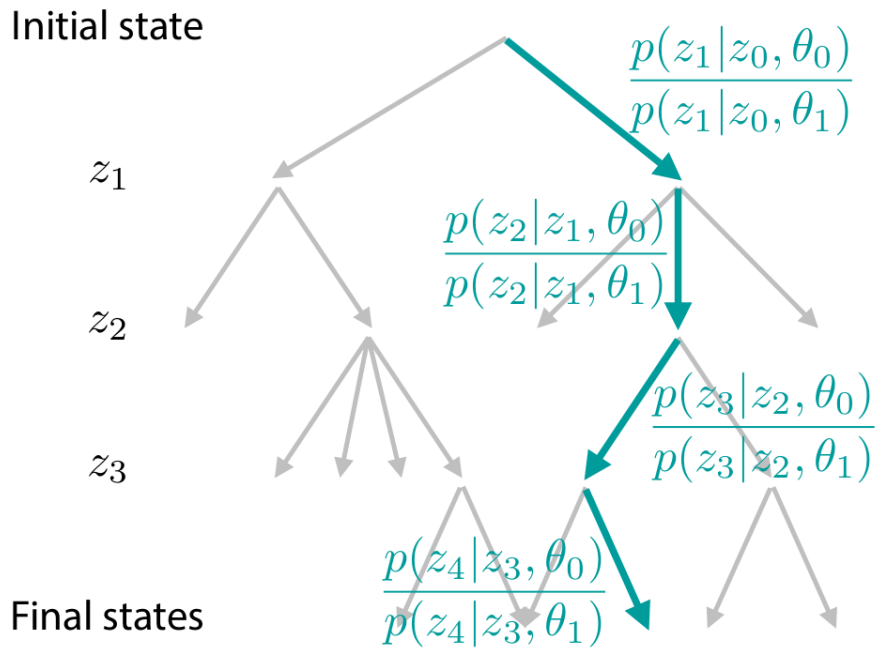
Traditional likelihood-free inference treats the simulator as a generative black box: parameters in, samples out.

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 $p(x, z|\theta)$?

Extracting the joint likelihood ratio



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

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

Credits: Johann Brehmer.

RASCAL

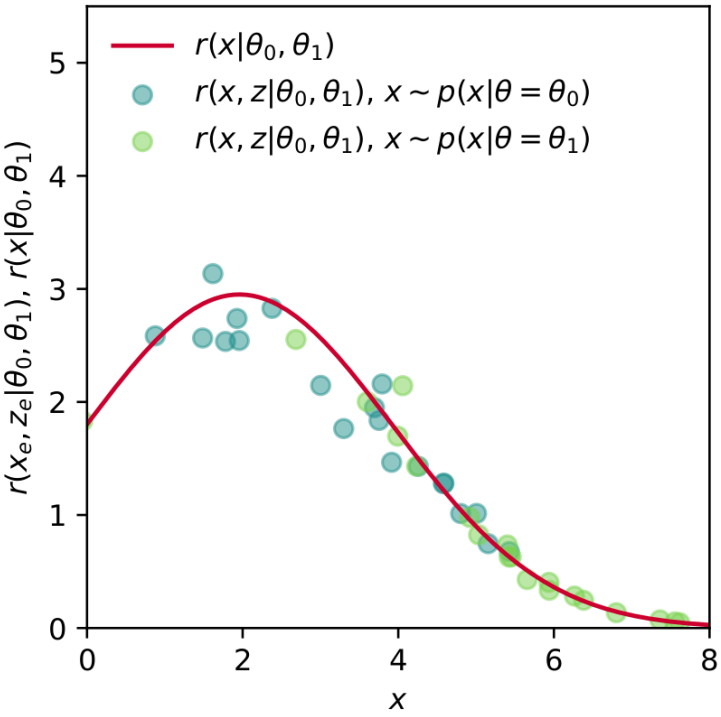
Regressing the likelihood ratio

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:

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

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

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

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

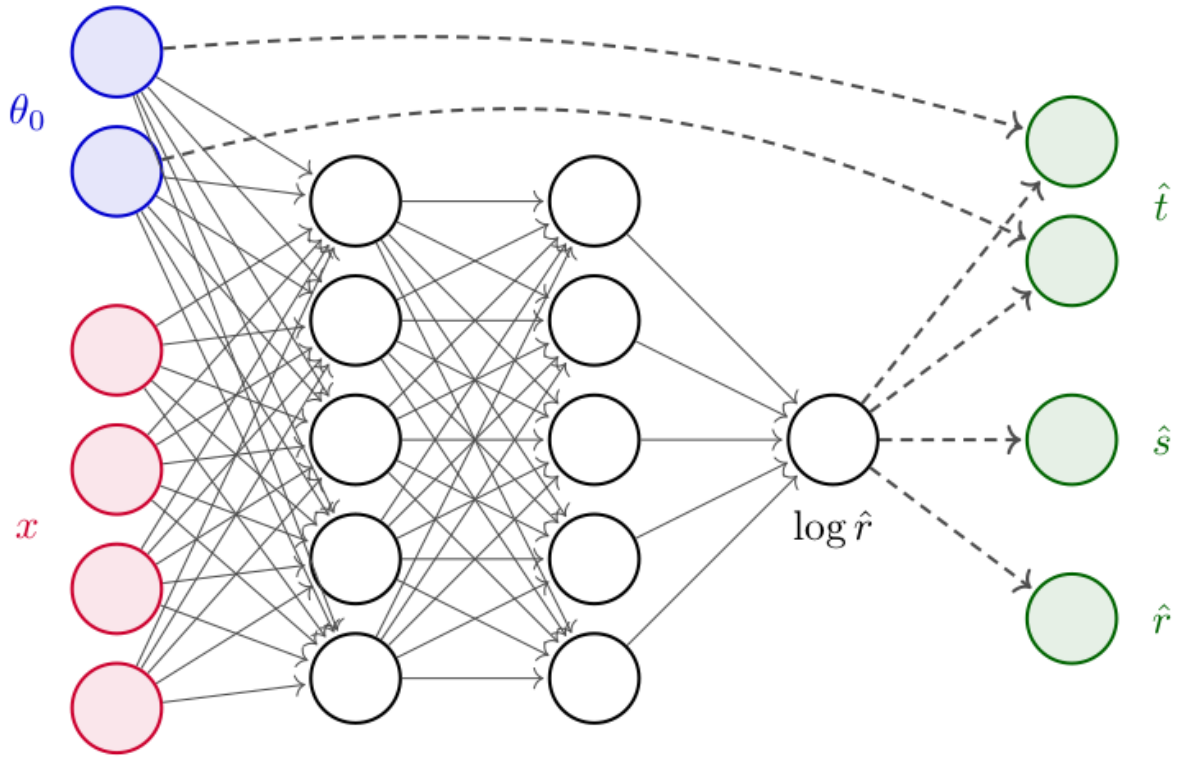
$$L_r = \mathbb{E}_{p(x,z|\theta_1)} \left[(r(x, z|\theta_0, \theta_1) - \hat{r}(x))^2 \right],$$

which is minimized by

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

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

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

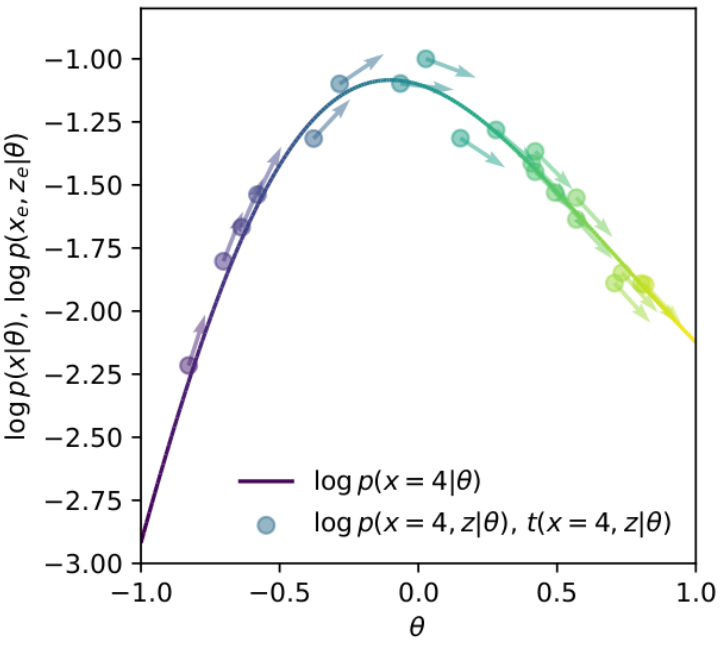


Regressing the score

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

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

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

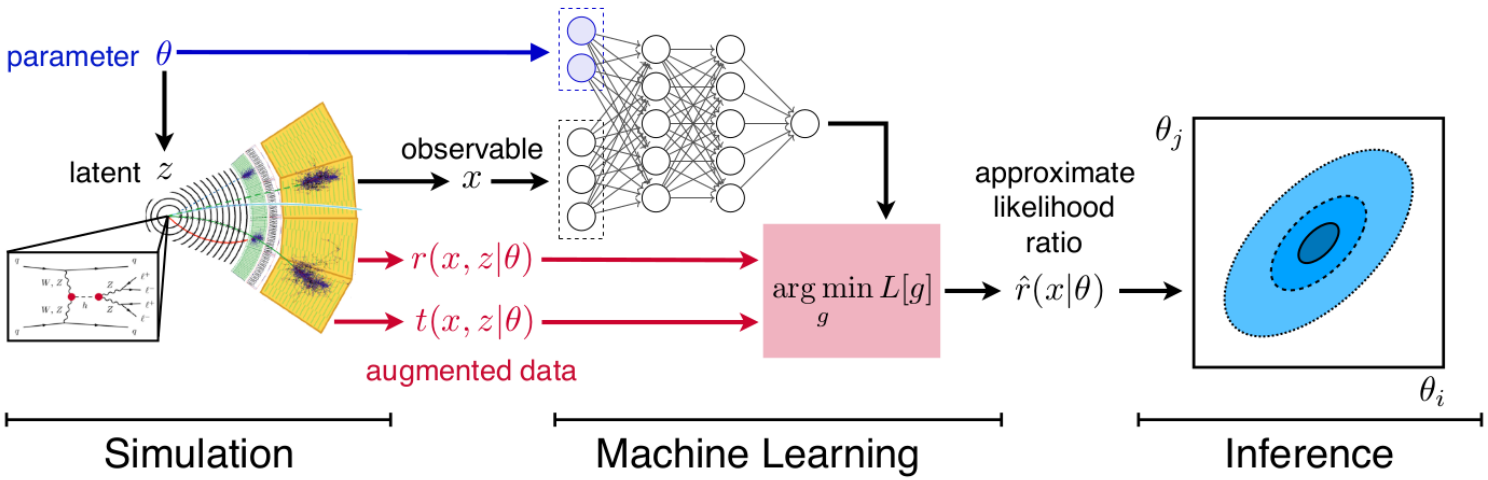
$$L_t = \mathbb{E}_{p(x,z|\theta_0)} \left[(t(x, z|\theta_0) - \hat{t}\,(x))^2 \right],$$

which is minimized by

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

RASCAL

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



There is more...

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

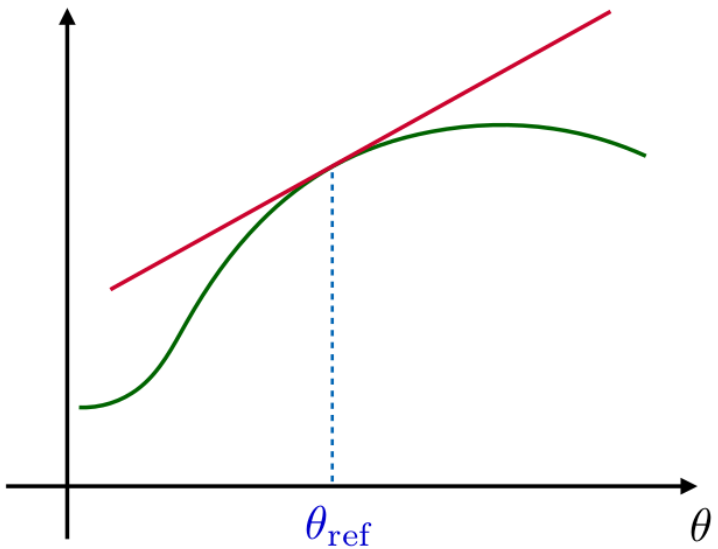
Brehmer, Louppe, Pavez and Cranmer, 2018 [arXiv:1805.12244].

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

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

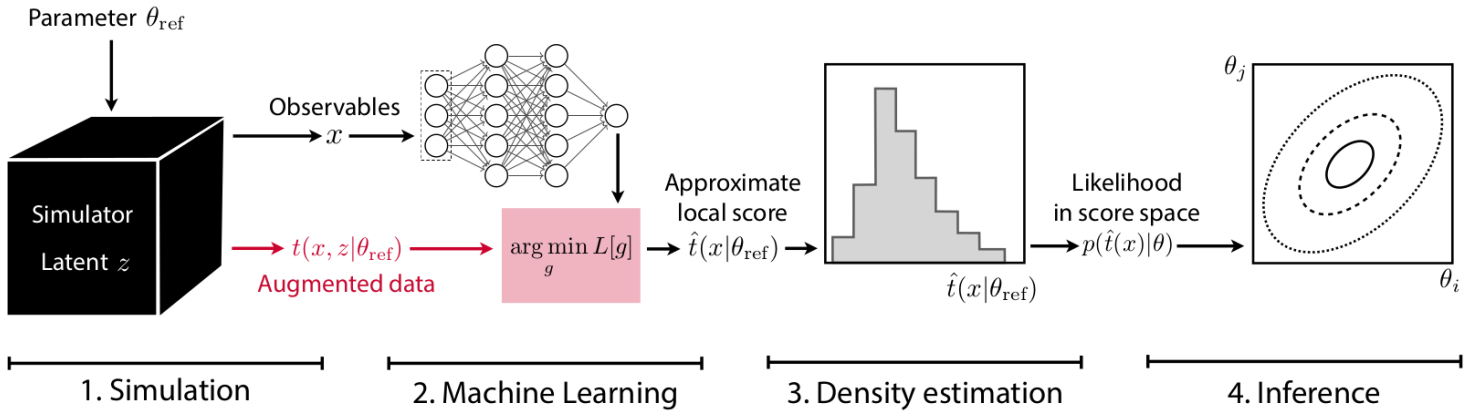
where the score $t(x|\theta_{\text{ref}})$ are its sufficient statistics.

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.

Brehmer, Louppe, Pavez and Cranmer, 2018 [arXiv:1805.12244].

SALLY

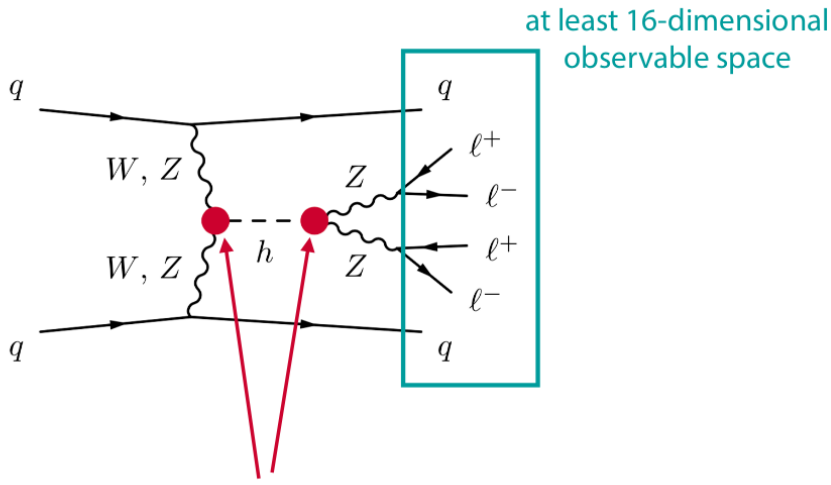


Brehmer, Louppe, Pavez and Cranmer, 2018 [arXiv:1805.12244].

Examples

① Hunting new physics at particle colliders

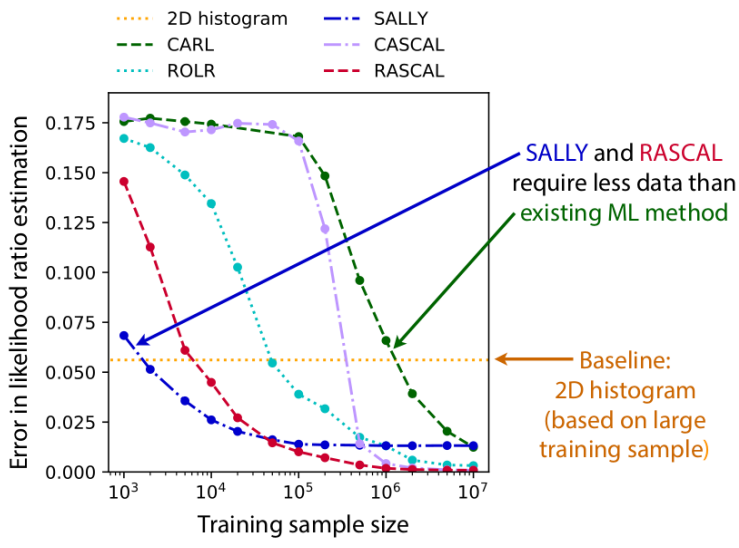
The goal is to constrain two EFT parameters and compare against traditional histogram analysis.

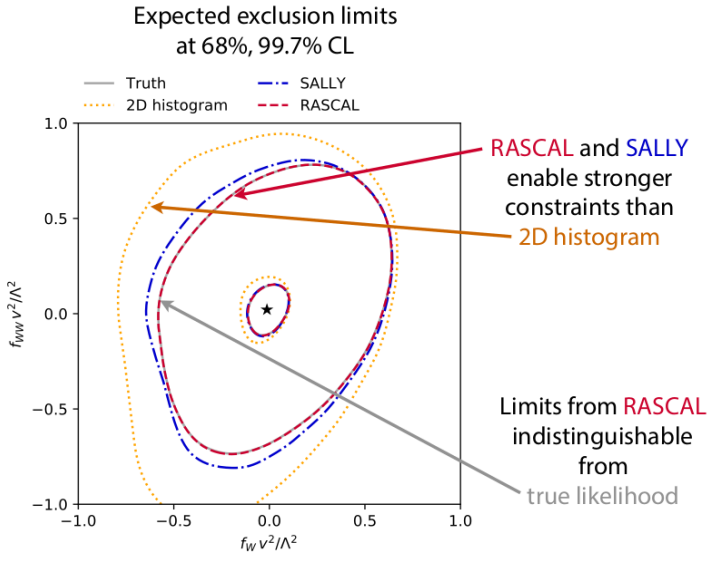


Exciting new physics might hide here!
We parameterize it with two EFT coefficients:

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

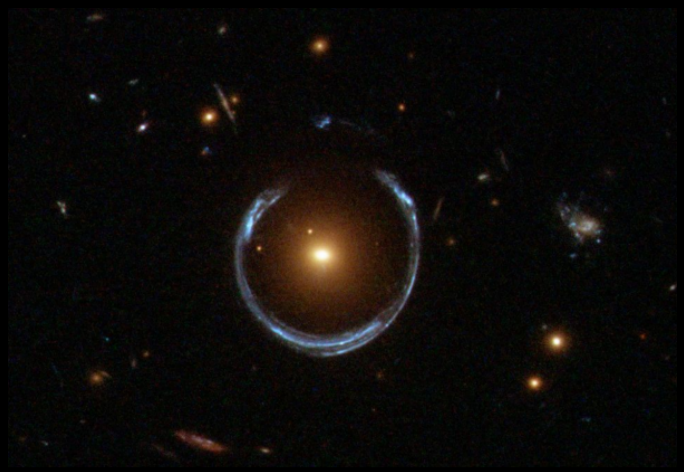
Brehmer, Cranmer, Louppe, and Pavez, 2018a [arXiv:1805.00020], 2018b [arXiv:1805.00013]; Brehmer, Louppe, Pavez and Cranmer, 2018 [arXiv:1805.12244].

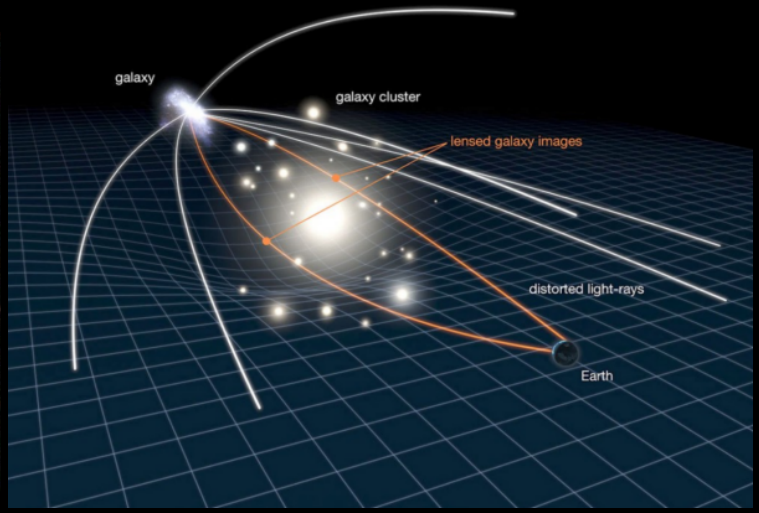




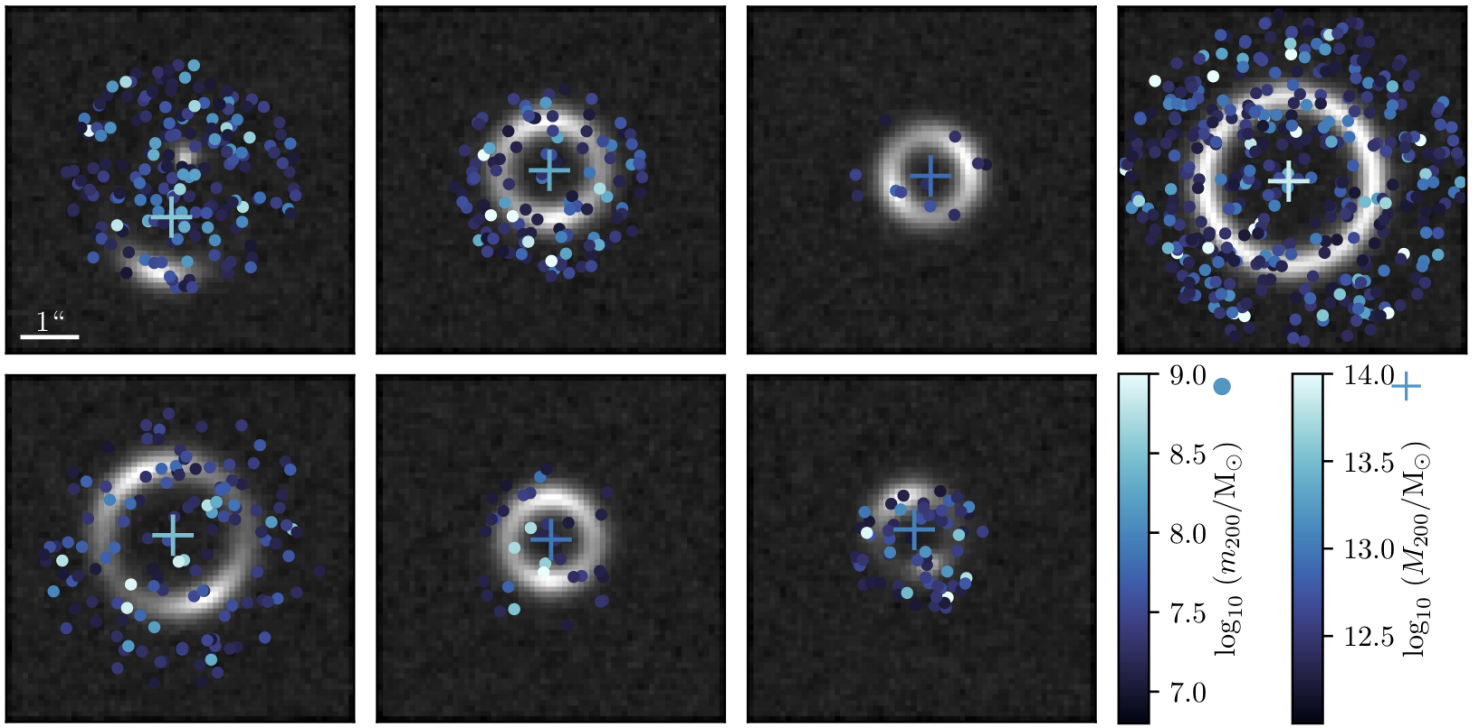
Brehmer, Cranmer, Louppe, and Pavez, 2018a [arXiv:1805.00020], 2018b [arXiv:1805.00013]; Brehmer, Louppe, Pavez and Cranmer, 2018 [arXiv:1805.12244].

② Dark matter substructure from gravitational lensing





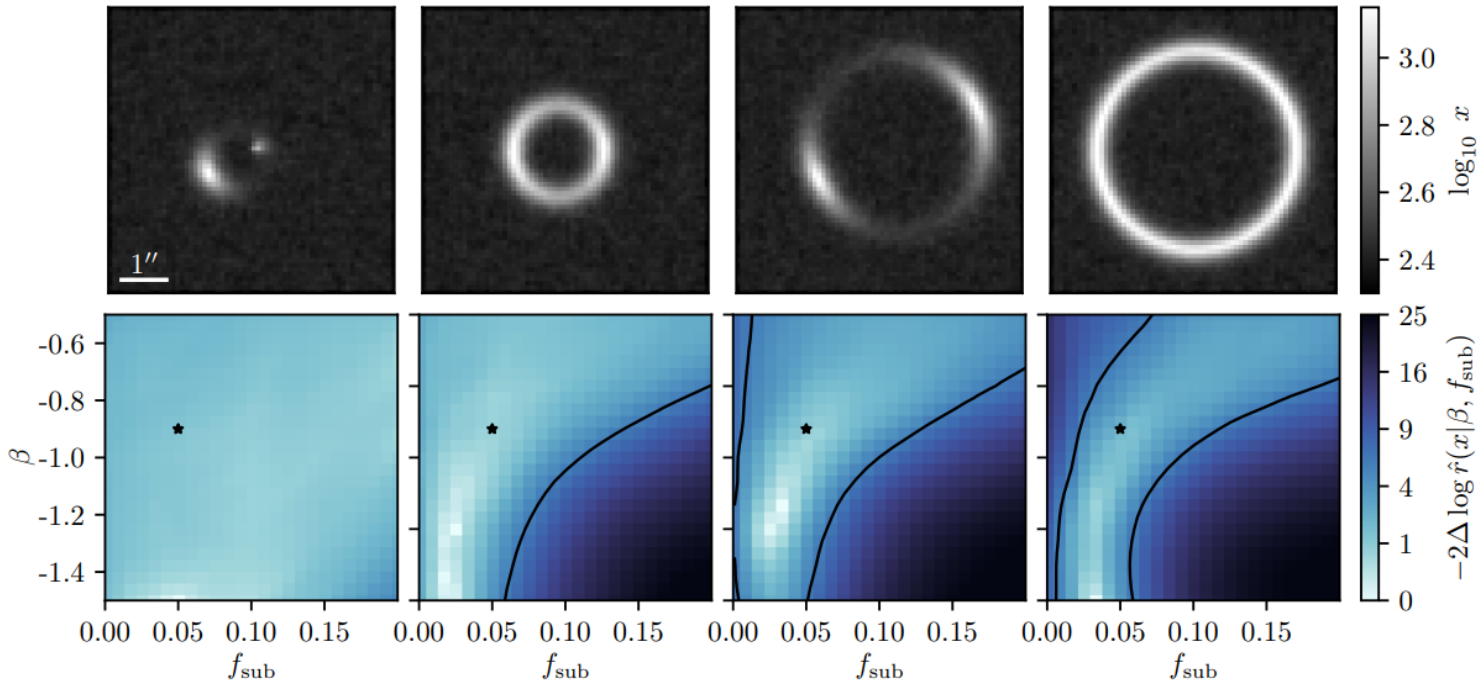
Brehmer, Mishra-Sharma, Hermans, Louppe, and Cranmer, 2019 [arXiv:1909.02005].



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

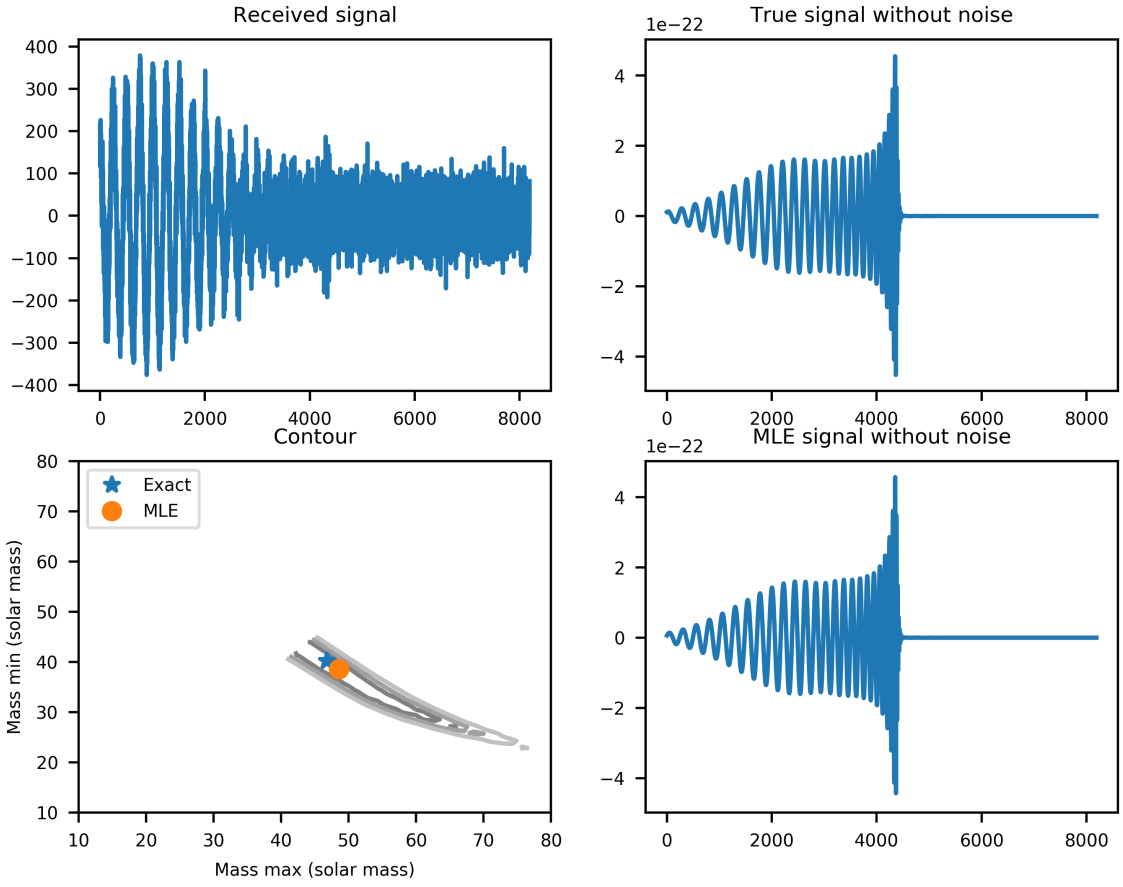
The goal is the **inference of population parameters β and $f_{\rm sub}$.**

Brehmer, Mishra-Sharma, Hermans, Louppe, and Cranmer, 2019 [arXiv:1909.02005].



Brehmer, Mishra-Sharma, Hermans, Louppe, and Cranmer, 2019 [arXiv:1909.02005].

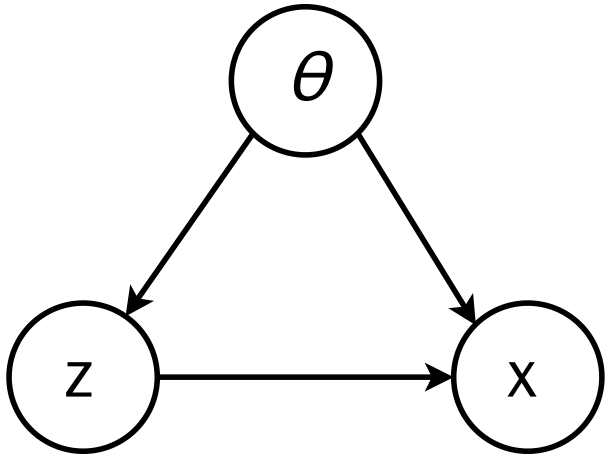
③ Gravitational waves (work in progress with GRAPPA!)



Bayesian inference

Bayesian inference = computing the posterior

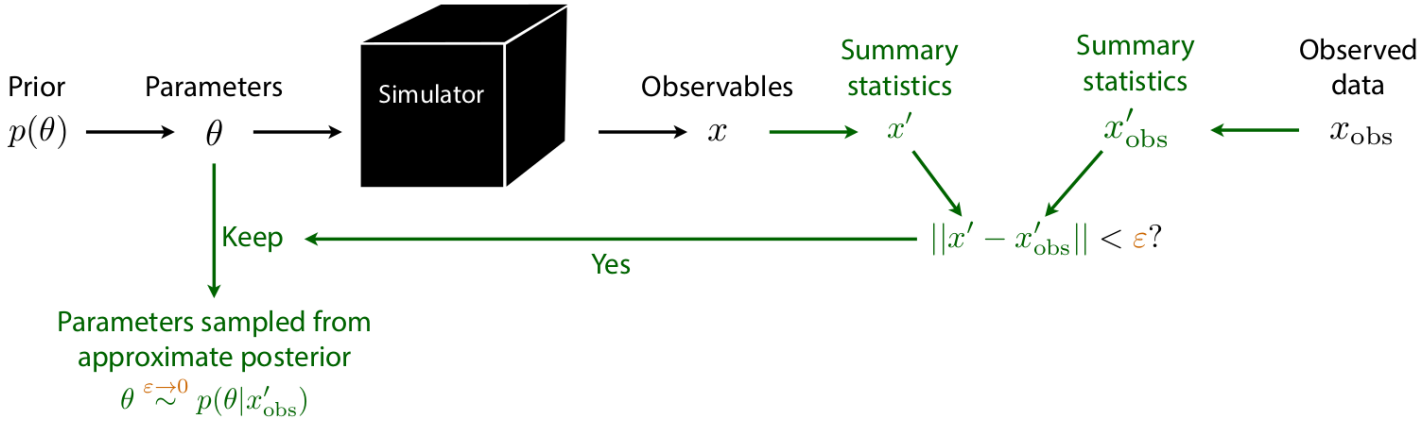
$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}.$$



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.

Credits: Johann Brehmer.

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

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

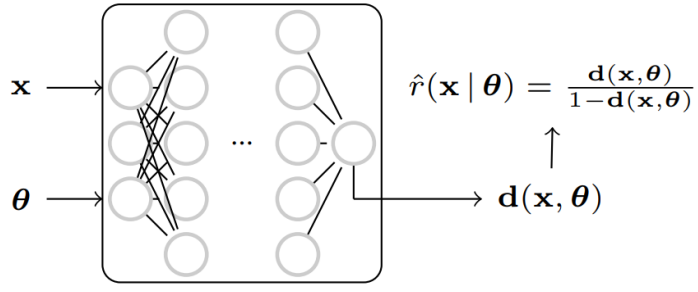
where $r(x|\theta) = \frac{p(x|\theta)}{p(x)}$ is the likelihood-to-evidence ratio.

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 $\mathbf{d}(\mathbf{x}, \boldsymbol{\theta})$.

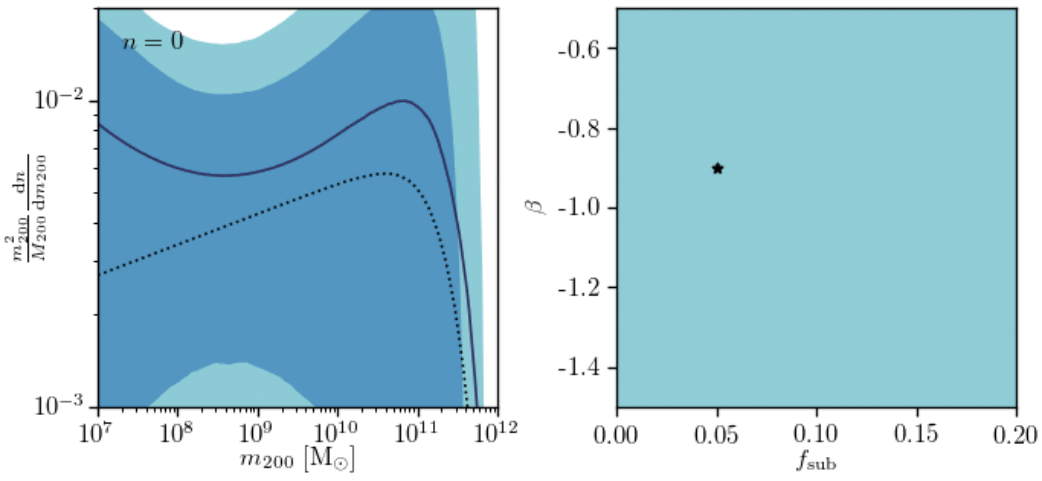
Inputs:	Criterion ℓ (e.g., BCE)
	Implicit generative model $p(\mathbf{x} \mid \boldsymbol{\theta})$
	Prior $p(\boldsymbol{\theta})$
Outputs:	Parameterized classifier $\mathbf{d}_\phi(\mathbf{x}, \boldsymbol{\theta})$
Hyperparameters:	Batch-size M

while not converged do
  Sample $\boldsymbol{\theta} \leftarrow \{\boldsymbol{\theta}_m \sim p(\boldsymbol{\theta})\}_{m=1}^{M}$
  Sample $\boldsymbol{\theta}' \leftarrow \{\boldsymbol{\theta}'_m \sim p(\boldsymbol{\theta})\}_{m=1}^{M}$
  Simulate $\mathbf{x} \leftarrow \{\mathbf{x}_m \sim p(\mathbf{x} \mid \boldsymbol{\theta}_m)\}_{m=1}^{M}$
  $\mathcal{L} \leftarrow \ell(\mathbf{d}_\phi(\mathbf{x}, \boldsymbol{\theta}),\ 1) + \ell(\mathbf{d}_\phi(\mathbf{x}, \boldsymbol{\theta}'),\ 0)$
  $\phi \leftarrow$ **OPTIMIZER**$(\phi,\ \nabla_\phi \mathcal{L})$
end while
return $\mathbf{d}_\phi$



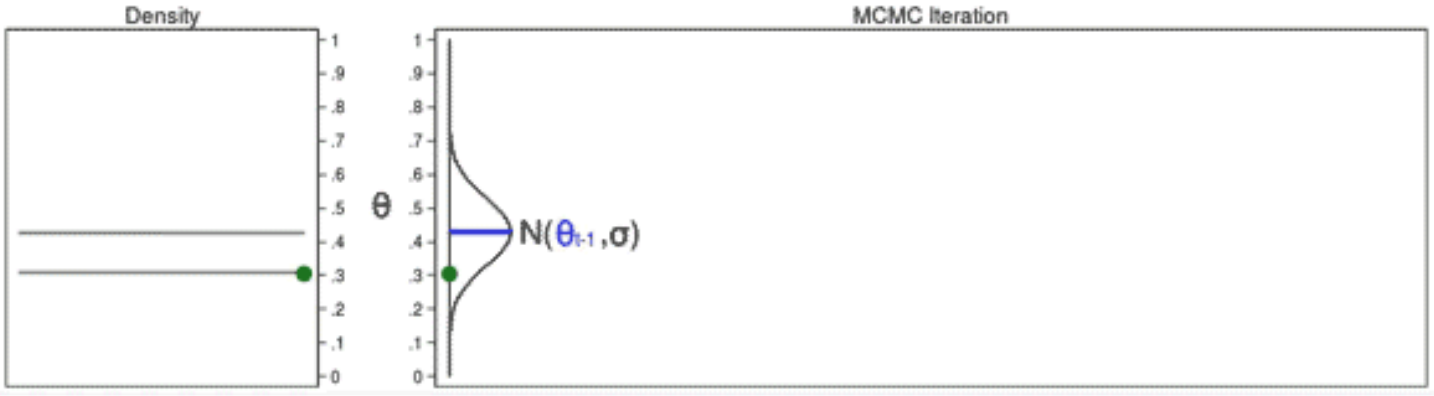
Hermans, Begy and Louppe, 2019 [arXiv:1903.04057]; Brehmer, Mishra-Sharma, Hermans, Louppe, and Cranmer, 2019 [arXiv:1909.02005].

Bayesian inference of dark matter subhalo population parameters



Brehmer, Mishra-Sharma, Hermans, Louppe, and Cranmer, 2019 [arXiv:1909.02005].

MCMC posterior sampling



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

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

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

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

Otherwise $\theta_t = \theta_{t-1} = 0.429$

Credits: Chuck Huber, 2016.

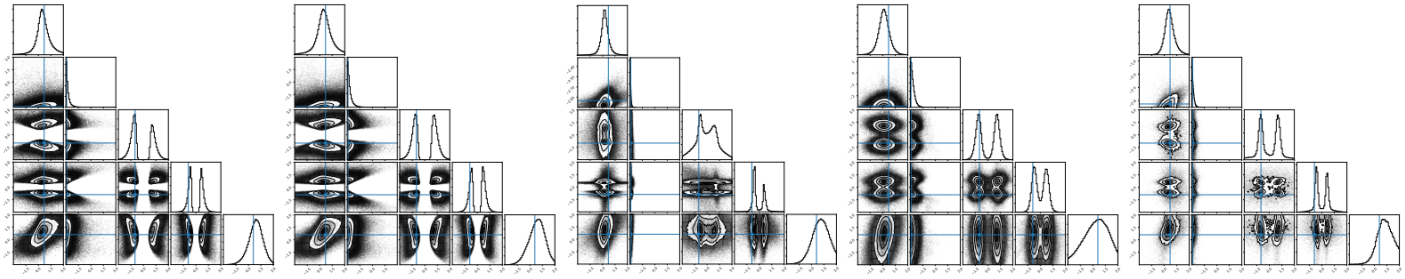
Likelihood-free MCMC

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 $\hat{r}\left(x|\theta_{\text{new}}, \theta_{t-1}\right)$ of the likelihood ratio.

Hermans, Begy and Louppe, 2019 [arXiv:1903.04057].



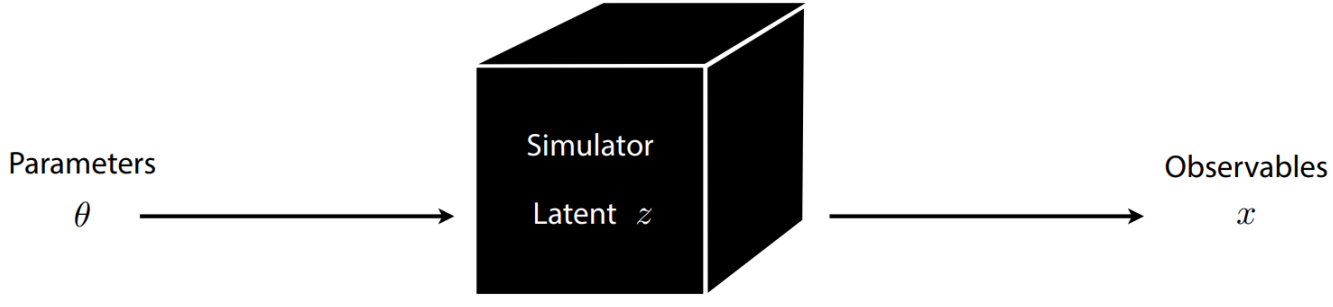
(a) MCMC groundtruth (b) AALR-MCMC (c) SNPE-A (d) SNPE-B (e) APT

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.

Hermans, Begy and Louppe, 2019 [arXiv:1903.04057].

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

Arnaud Delaunoy

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

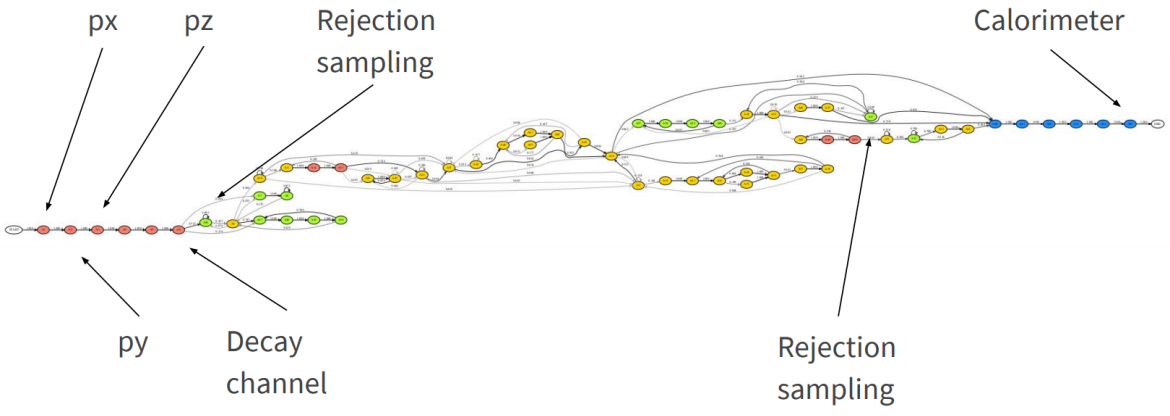
When the joint likelihood ratio $r(x, z|\theta_0, \theta_1)$ is available from the simulator, the corresponding $s(x, z|\theta_0, \theta_1)$ are also tractable.

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

Probabilistic programming



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)

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

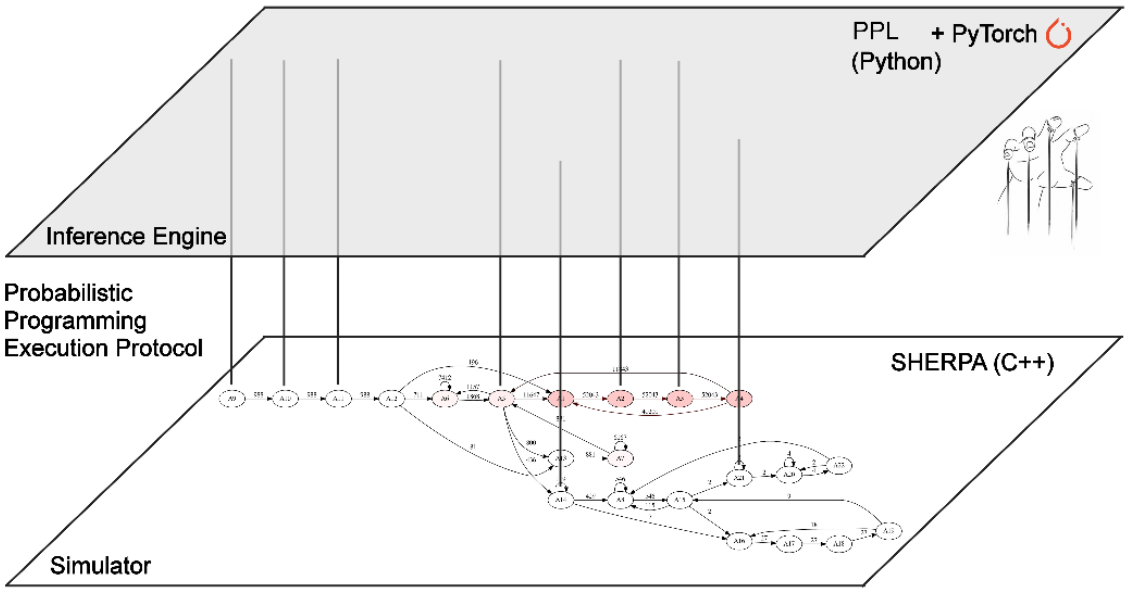
Le et al, 2016 [arXiv:1610.09900]; Baydin et al, 2018 [arXiv:1807.07706]; Baydin et al, 2019 [arXiv:1907.03382].

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.

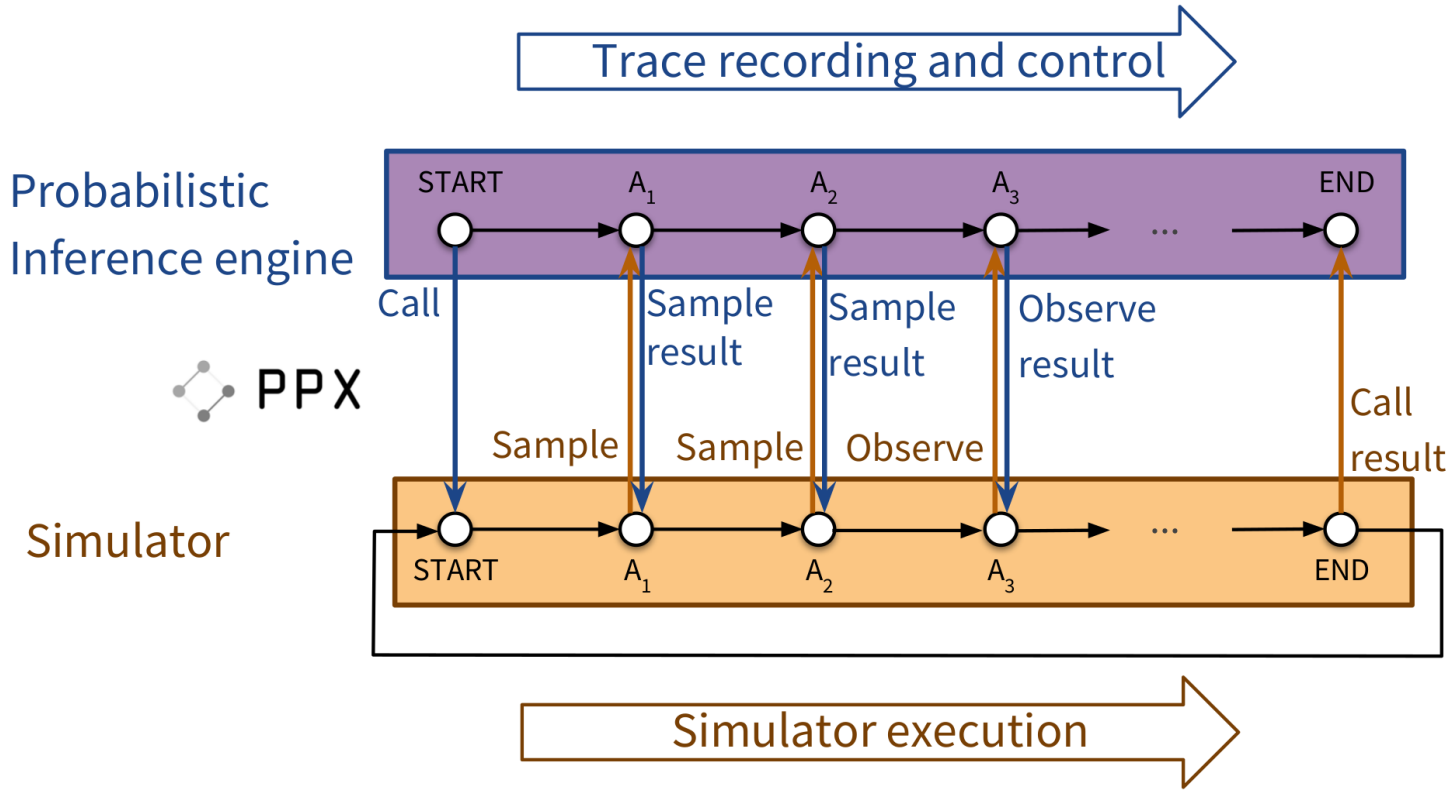
Scientific simulators are probabilistic programs!



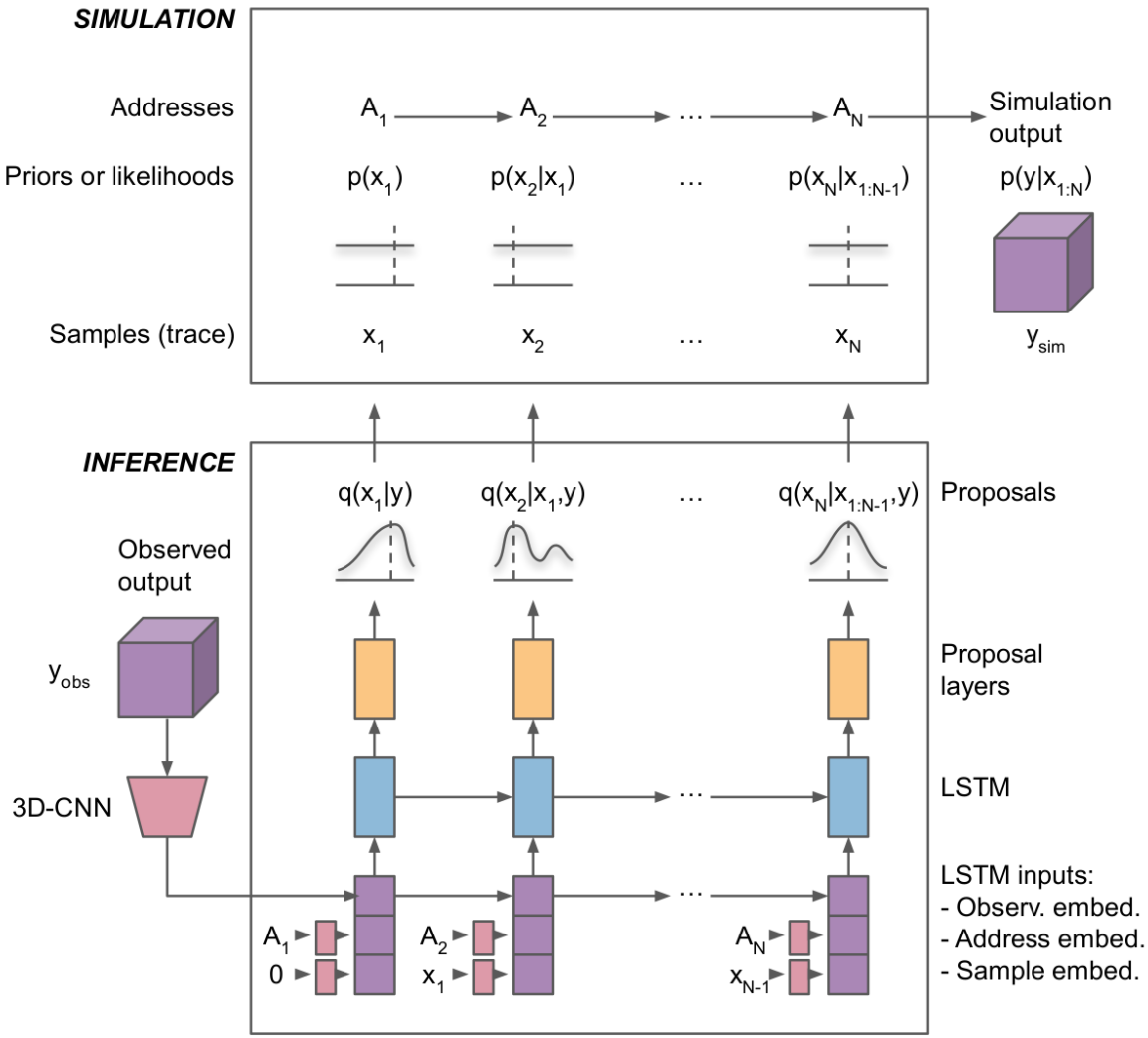
Key idea

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

Le et al, 2016 [arXiv:1610.09900]; Baydin et al, 2018 [arXiv:1807.07706]; Baydin et al, 2019 [arXiv:1907.03382].



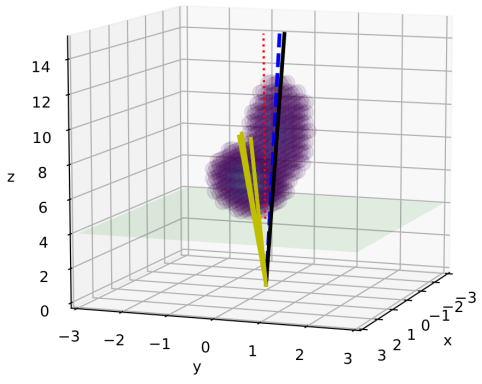
Le et al, 2016 [arXiv:1610.09900]; Baydin et al, 2018 [arXiv:1807.07706]; Baydin et al, 2019 [arXiv:1907.03382].

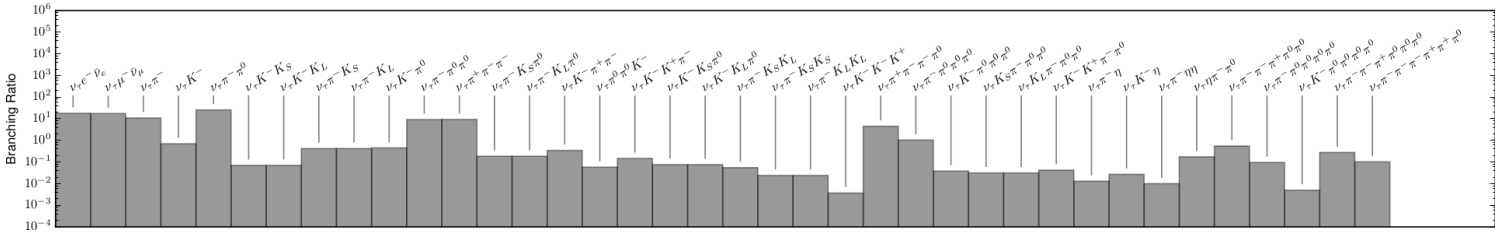


Le et al, 2016 [arXiv:1610.09900]; Baydin et al, 2018 [arXiv:1807.07706]; Baydin et al, 2019 [arXiv:1907.03382].

③ Taking control of Sherpa (particle physics simulator)

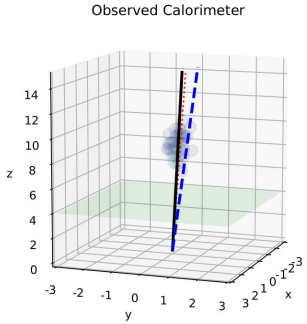
- τ decay in Sherpa, 38 decay channels, coupled with an approximate calorimeter simulation in C++.
- Observations are 3D calorimeter depositions.
- Latent variables (Monte Carlo truth) of interest: decay channel, px, py, pz momenta, final state momenta and IDs.





Le et al, 2016 [arXiv:1610.09900]; Baydin et al, 2018 [arXiv:1807.07706]; Baydin et al, 2019 [arXiv:1907.03382].

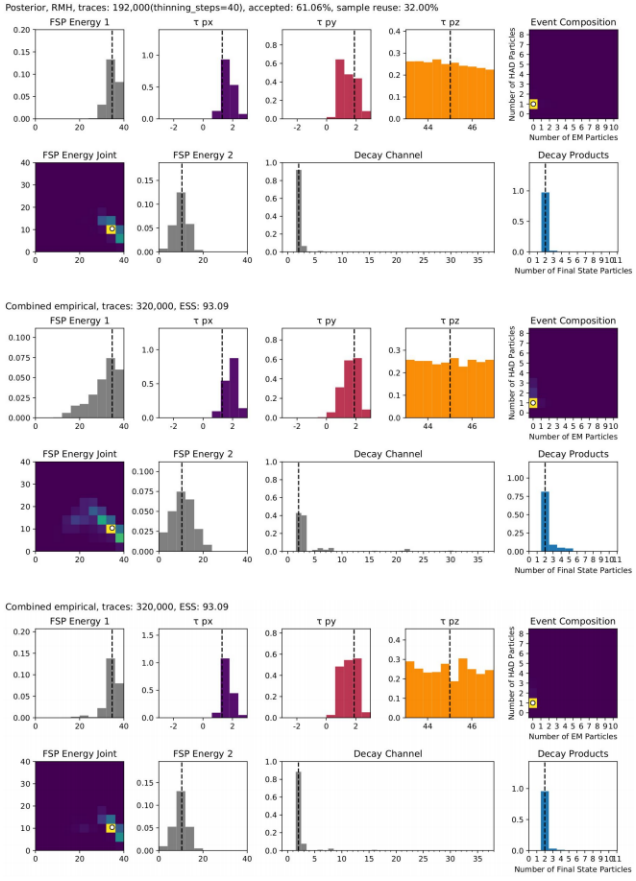
Inference results

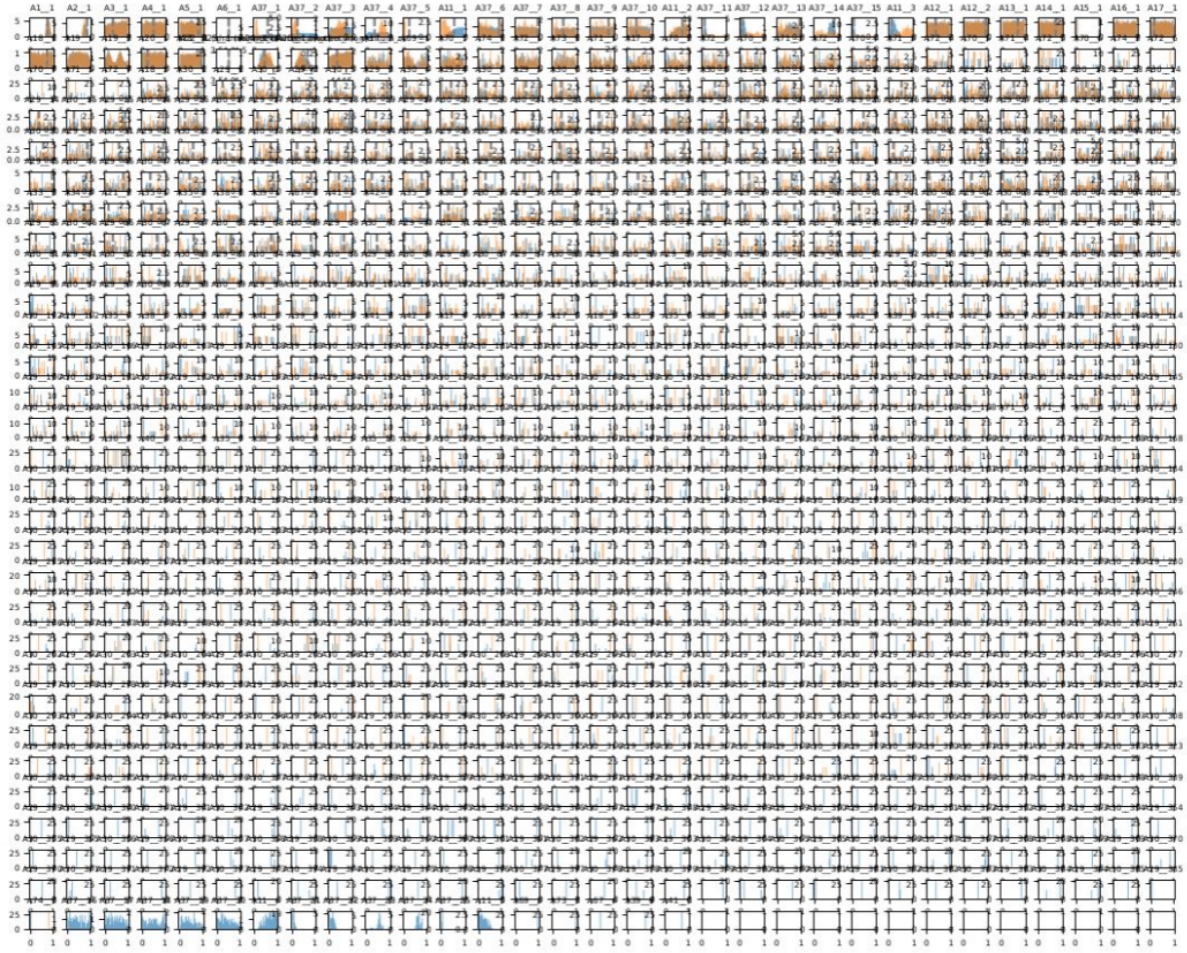


MCMC true posterior (7.7M single node)

IC proposal from trained NN

IC posterior after importance weighting



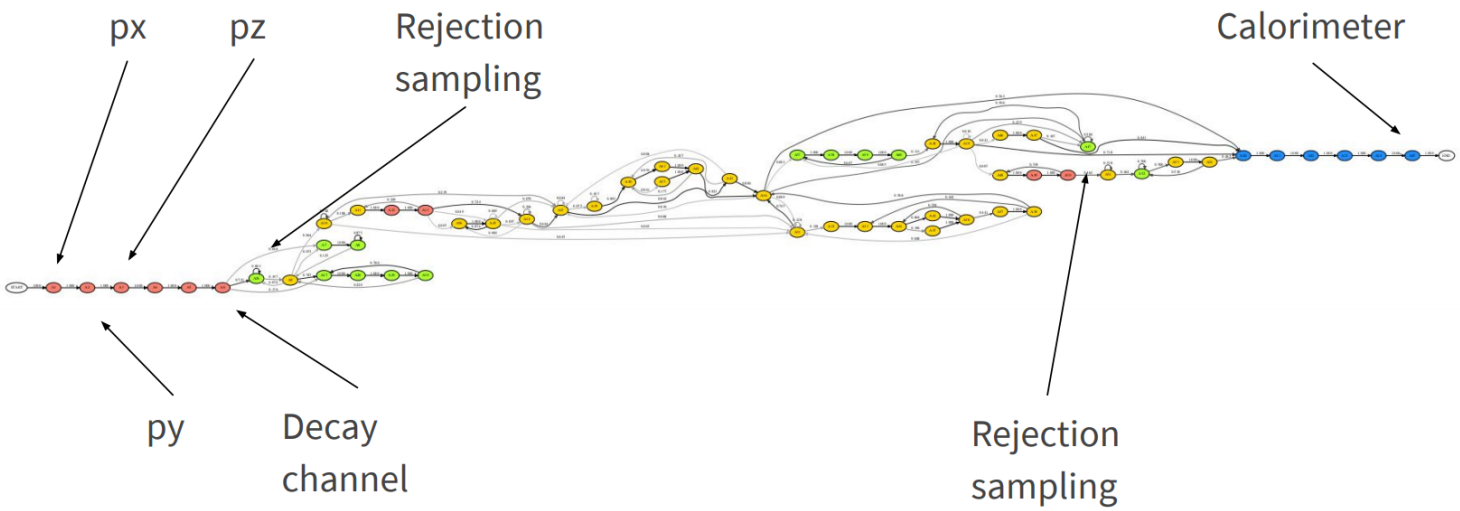


We obtain posteriors over the whole Sherpa address space, 1000s of addresses.

Le et al, 2016 [arXiv:1610.09900]; Baydin et al, 2018 [arXiv:1807.07706]; Baydin et al, 2019 [arXiv:1907.03382].

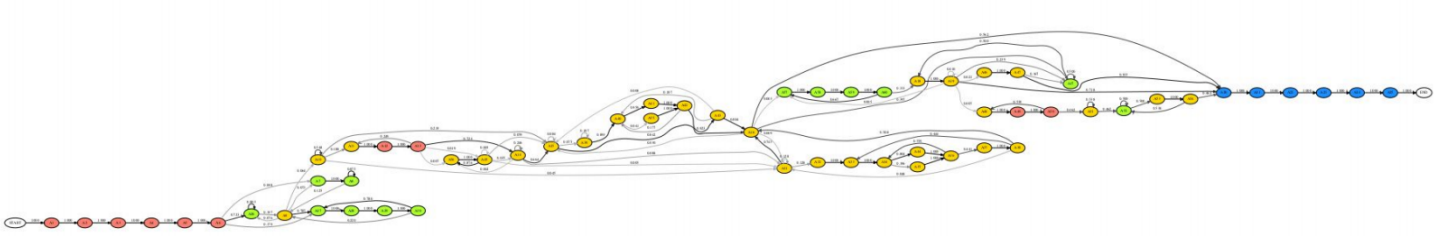
Interpretability

Latent probabilistic structure of the 250 most frequent trace types:

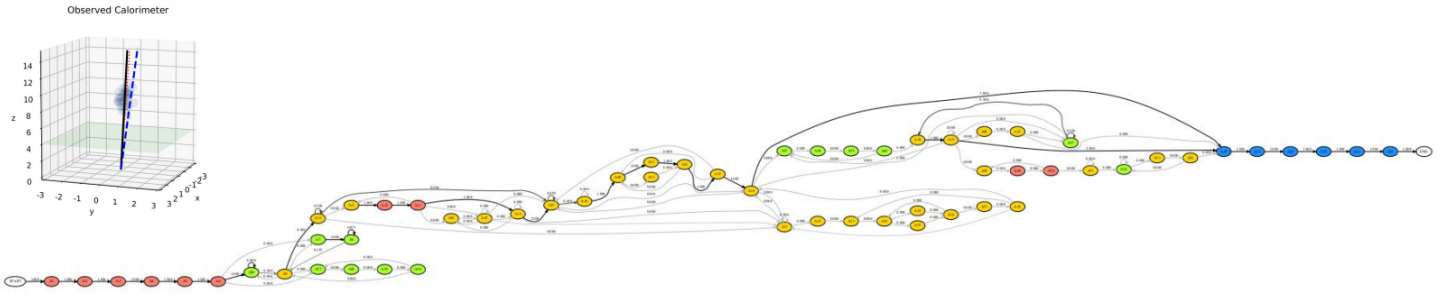


53

Le et al, 2016 [arXiv:1610.09900]; Baydin et al, 2018 [arXiv:1807.07706]; Baydin et al, 2019 [arXiv:1907.03382].



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



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

Le et al, 2016 [arXiv:1610.09900]; Baydin et al, 2018 [arXiv:1807.07706]; Baydin et al, 2019 [arXiv:1907.03382].