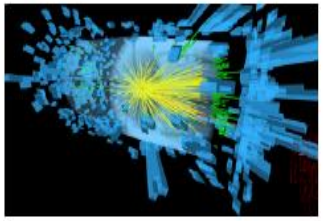
Simulation-based inference

- **Based on K. Cranmer presentation from PhyStat25 workshop**

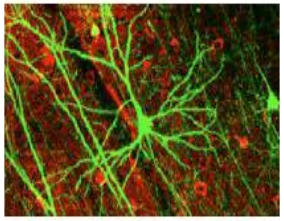
Simulators and inference

- **Many domains of science have developed complex simulations to describe phenomena of interest.**
- **While these simulators provide higly sophisticated models, they are poorly suited for inference and lead to challenging inverse problems.**
- **The source of the challenge is that the probability density (or likelihood) for a given observation is typically intractable. Such models are often refered to as „implicit" models.**
- **The often used solution is to construct poweful summary statistics (observable) and compare observed data to simulated data. This approach have been used for Higgs discovery in a frequentist paradigm.**
- **Alternative technique known as Approximate Bayesian Computation (ABC) compares observed and simulated data based on some distance measure involving the summary statistics. ABS ,ethods is widely used in population biology, computational neuroscience, cosmology.**

High-fidelity simulators in Science



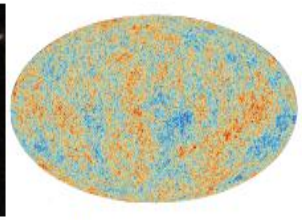
Particle colliders



Neuron activity

Epidemics

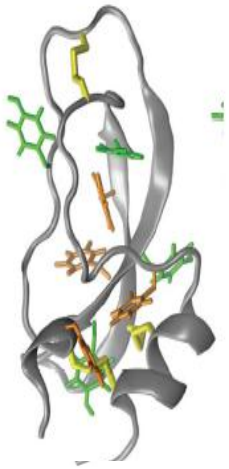
Gravitational lensing

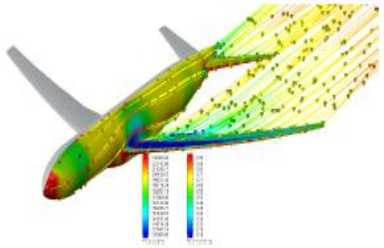


Evolution of the Universe

10^{-18} 10^{-15} 10^{-12} 10^{-9} 10^{-6} 10^{-3} 10^{0} 10^{3} 10^{6} 10^{9} 10^{12} 10^{15} 10^{18} 10^{21} 10^{24} 10^{27}

Length scale [m]

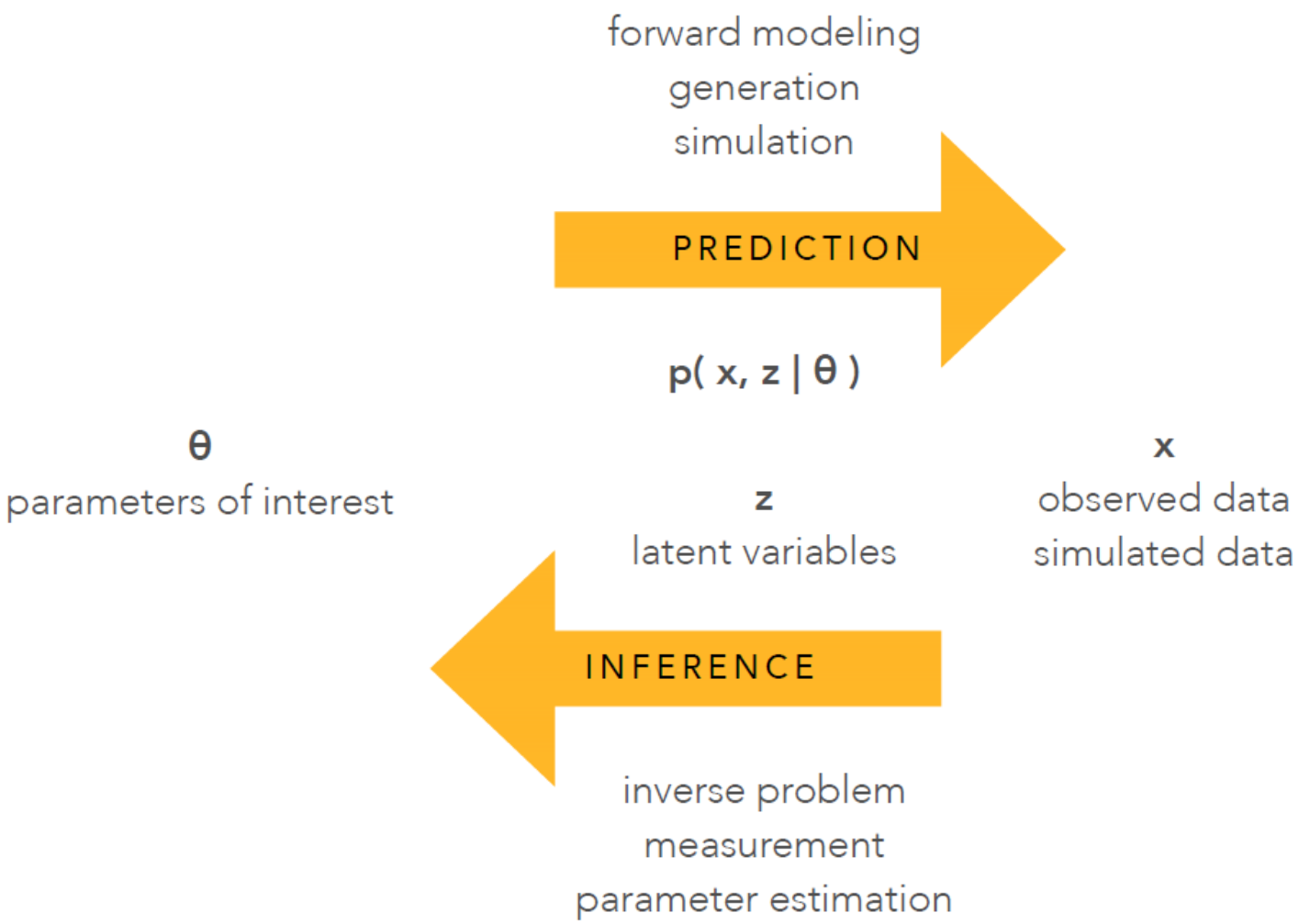




The expressiveness of programming languages facilitates the development of complex, high-fidelity simulations, and the power of modern computing provides the ability to generate synthetic data from them.

Unfortunately, these simulators are poorly suited for statistical inference.

Statistical framing



Simulation based inference

- **The class of inference methods for stochastic simulator where**
  - **evaluating the likelihood is intractable**
  - **it is possible to sample synthetic data $x \sim p(x|\theta)$**

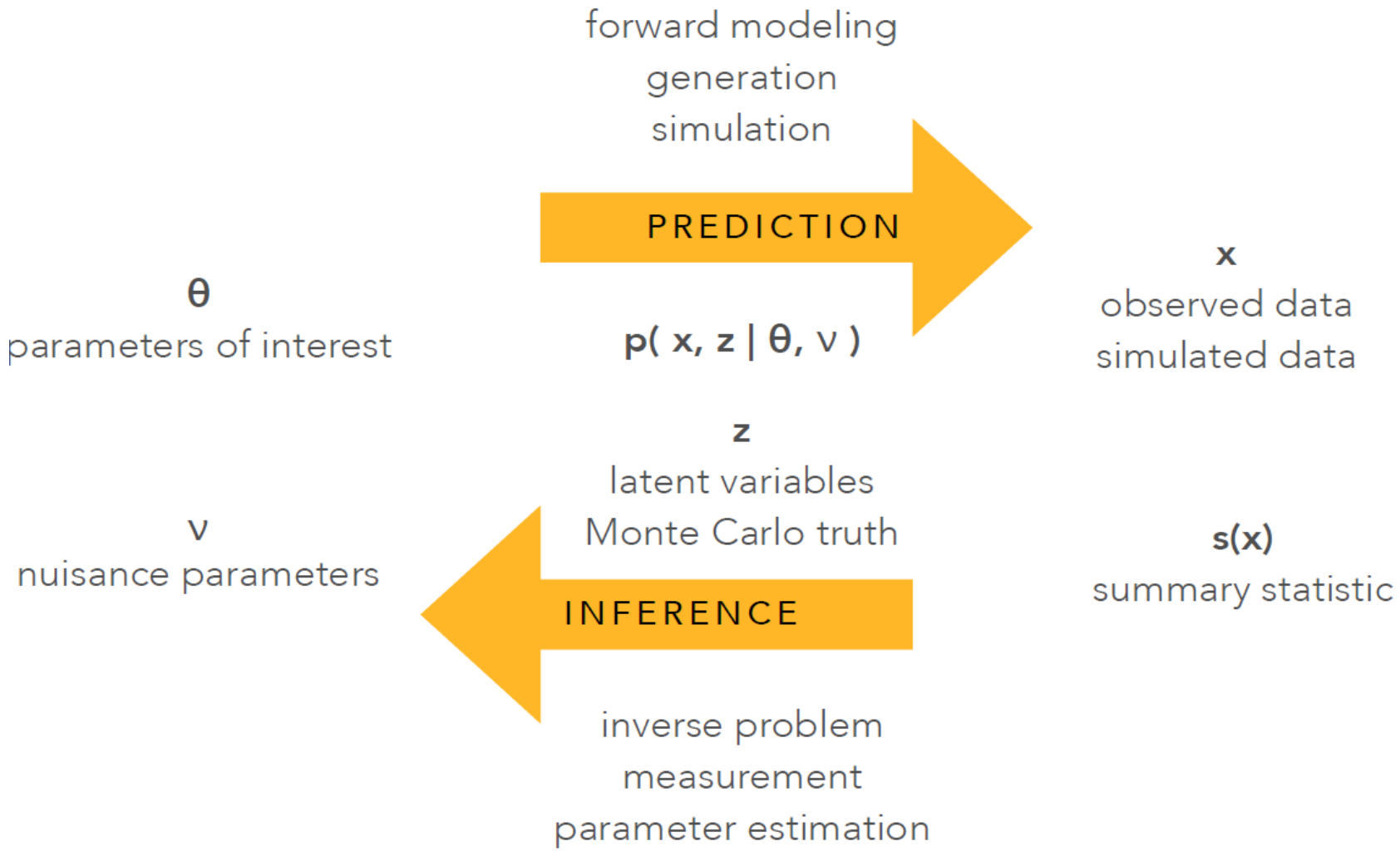
- **One usually approximate likelihood or likelihood ratio and then uses established inference techniques.**

Model mis-specification

- **Inference is always done within cotext of a model**
  - **If the model is mis-specified it wil affect inference**
  - **Here the model is the simulator**
    - **The simulator may not be perfect**
    - **Simulators usually include more effects than traditionally prescibed models**

- **To account for mis-modeling, simulators are often extended in numerous ways**
  - **Often these extensions are not based on first principles but ad-hoc/practical parameters**
  - **The simulator now also depends on nuisance parameters ν**

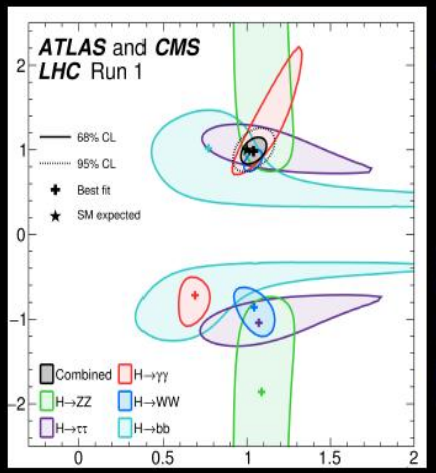
Statistical framing



The particle physics context

High-dimensional event data

[CMS]



[ATLAS, CMS 1606.02266]

Precision constraints on new physics

Parameterization e.g. in Effective Field Theory:

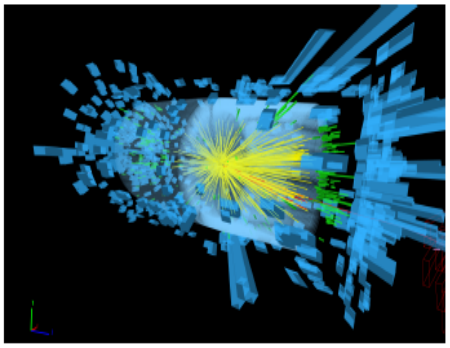
systematic expansion of new physics around Standard Model

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

10s to 100s "universal" parameters to measure

The likelihood is a key object

Let θ denote the coefficients of higher dimensional operators in the Lagrangian, x be high-dimensional data associated to an event, and $p(x \mid \theta) = \frac{1}{\sigma(\theta)} \frac{d\sigma}{dx}$ be the distribution for the data

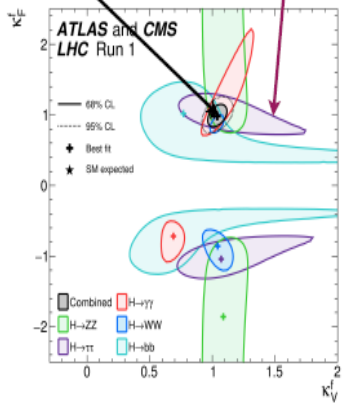


High-dimensional event data x

Likelihood function $p(x|\theta)$

Maximum-likelihood estimator

Confidence limits based on likelihood ratio tests



Constraints on parameters θ

Modeling particle physics processes

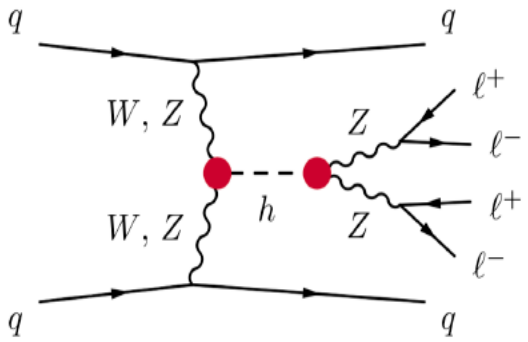
Theory
parameters
θ

Evolution

Modeling particle physics processes

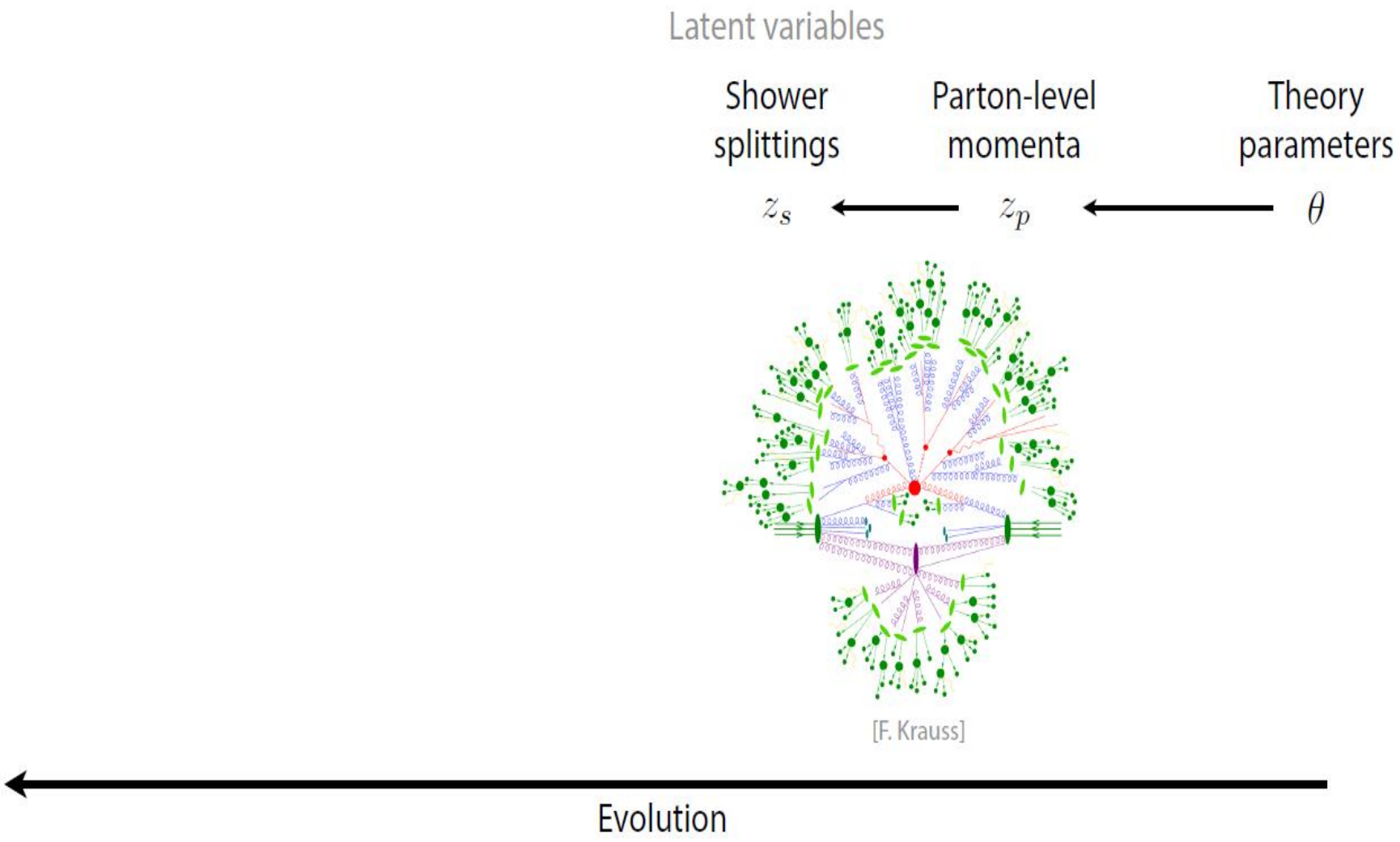
Latent variables

Parton-level momenta z_p ← Theory parameters θ



Evolution

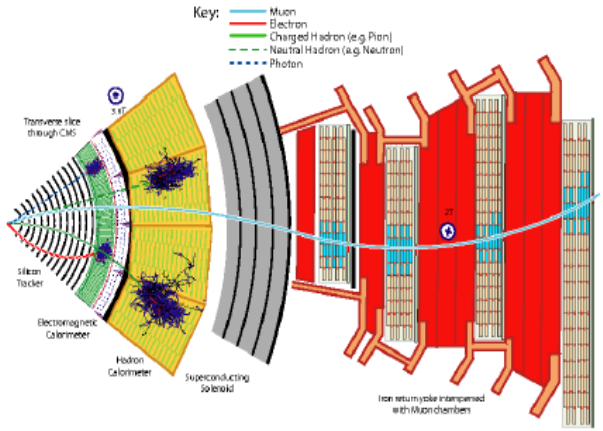
Modeling particle physics processes



Modeling particle physics processes

Latent variables

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



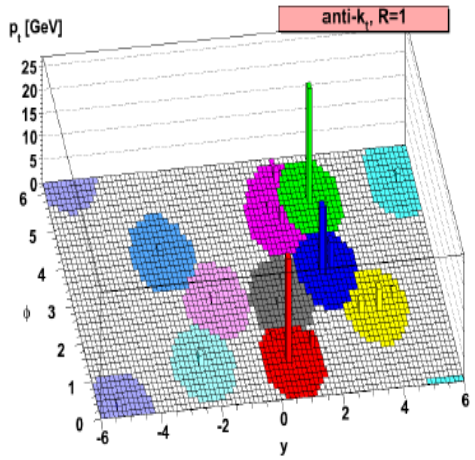
[CMS]

Evolution

Modeling particle physics processes

Latent variables

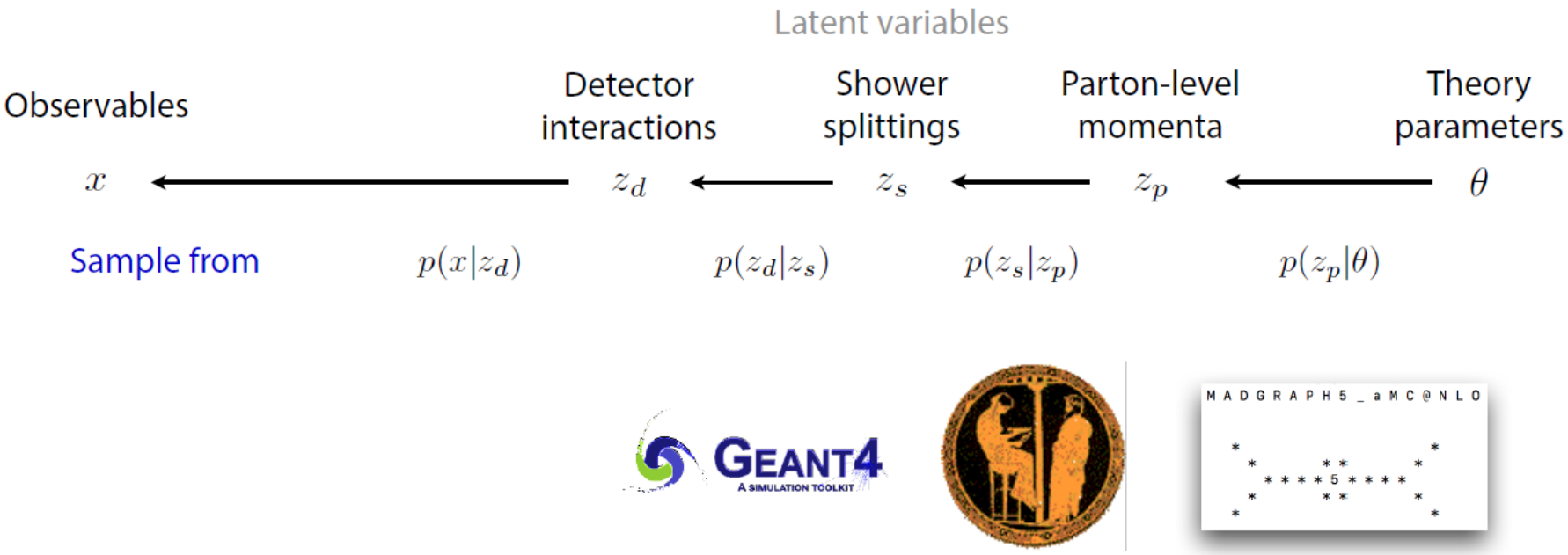
Observables	Detector interactions	Shower splittings	Parton-level momenta	Theory parameters
x	z_d	z_s	z_p	θ



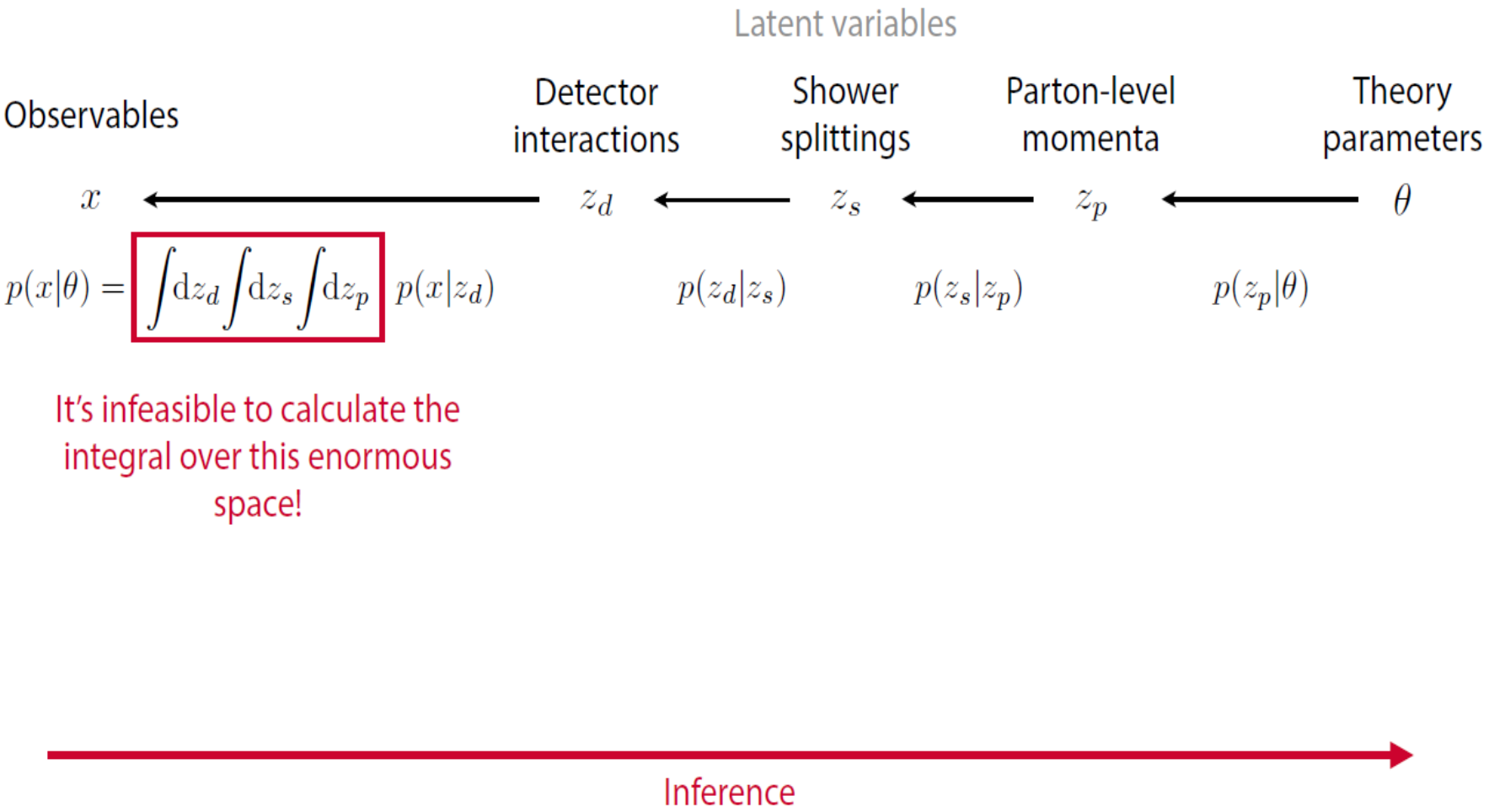
[M. Cacciari, G. Salam, G. Soyez 0802.1189]

Evolution

Modeling particle physics processes



Modeling particle physics processes

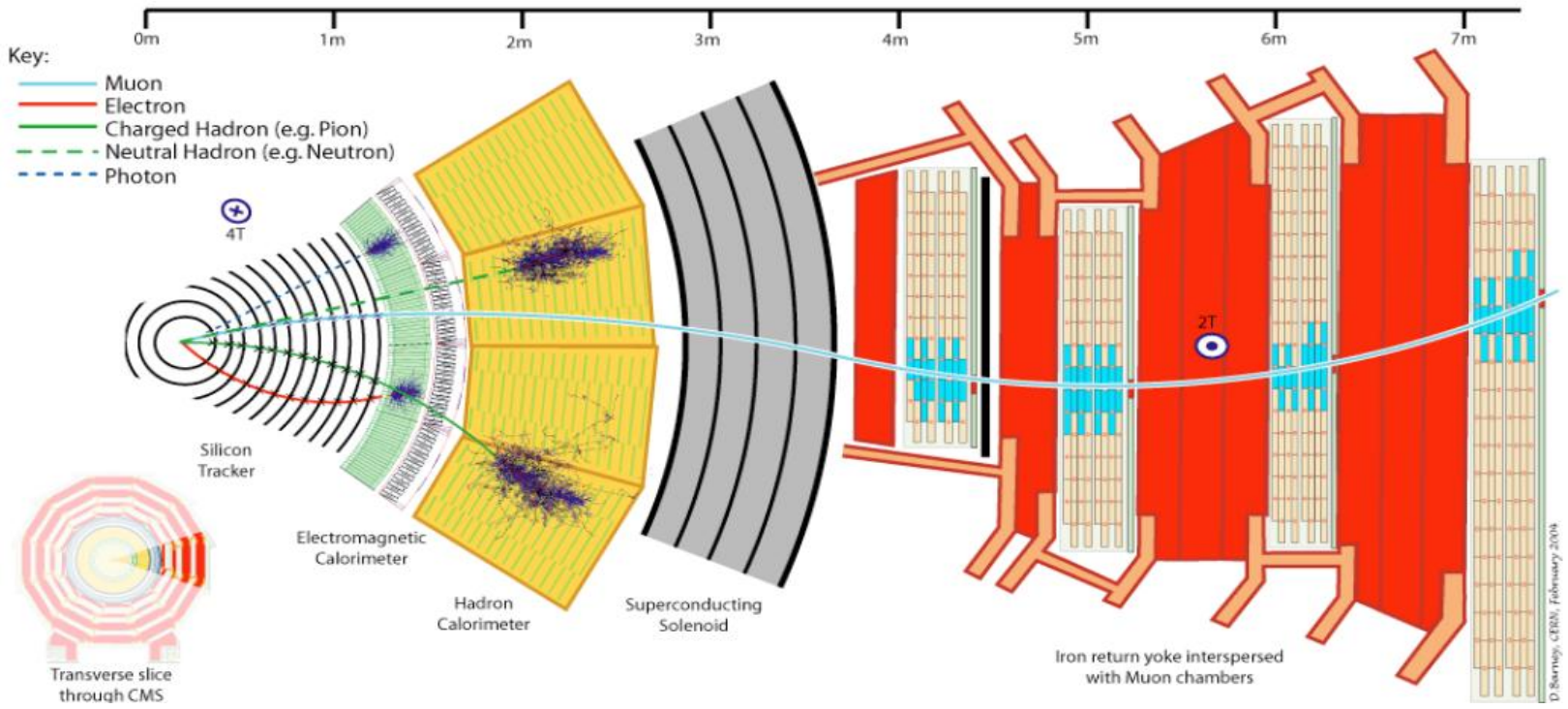


Detector simulation

Conceptually: Prob(detector response | particles)

Implementation: Monte Carlo integration over micro-physics

Consequence: evaluation of the likelihood is intractable



Detector simulation

Conceptually: Prob(detector response | particles)

Implementation: Monte Carlo integration over micro-physics

Consequence: evaluation of the likelihood is intractable

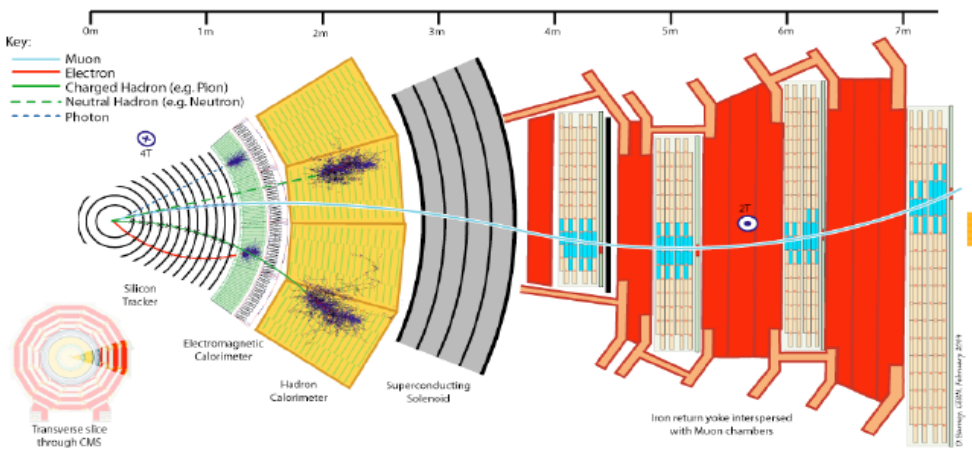
This motivates a new class of algorithms for what is called **likelihood-free inference (or simulation-based inference)**, which only require ability to generate samples from the simulation in the "forward mode"

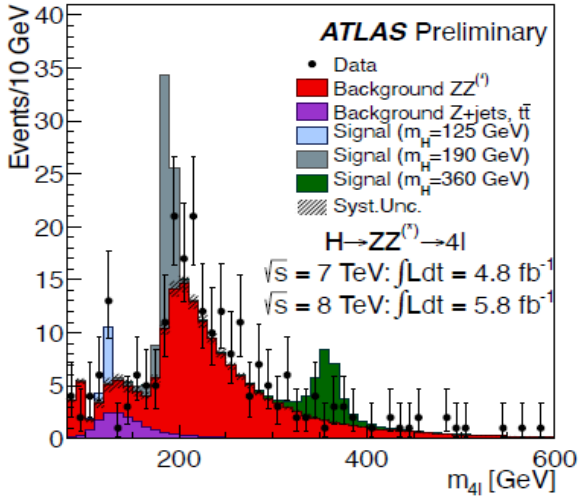
Detector simulation

10^8 SENSORS → 1 REAL-VALUED QUANTITY

Most measurements and searches for new particles at the LHC are based on the distribution of a single variable / observable / feature / summary statistic **s(x)**

- designing a good observable / summary statistic **s(x)** is a task for a skilled physicist and tailored to the goal of measurement or new particle search
- likelihood p(s|θ) **approximated** using histograms (univariate density estimation)





This doesn't scale if s is high dimensional!

An intractable integral

Monte Carlo Sampling

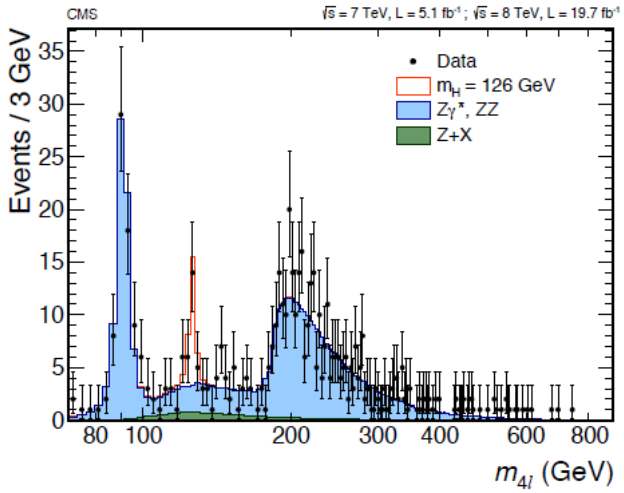
observed

what happened inside simulation

$$p(s \mid \theta) = \int dz dx \, p(s(x), z \mid \theta)$$

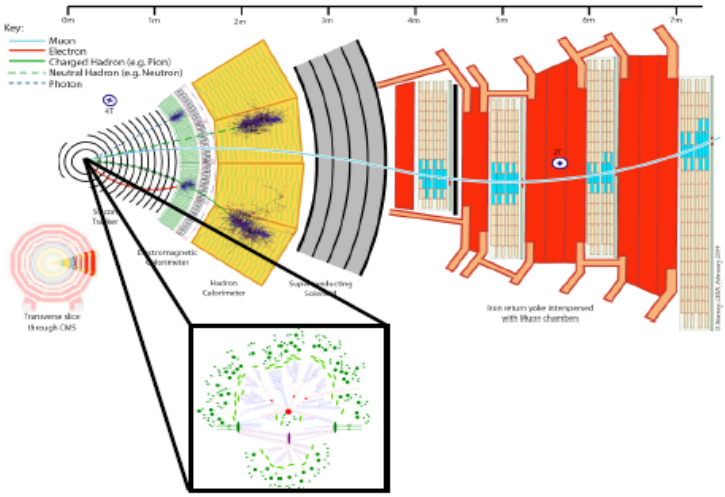
$$\hat{p}(s \mid \theta)$$

histogram approximation



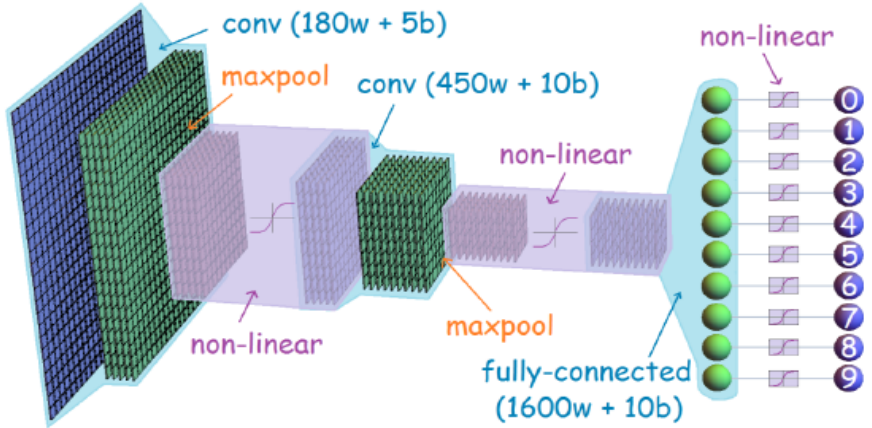
Two approaches for simulation based inference

Use simulator
(much more efficiently)



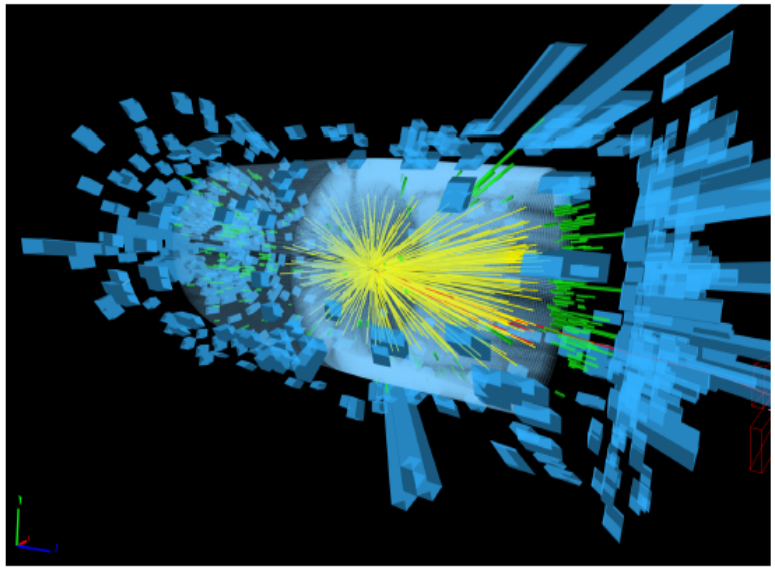
- Approximate Bayesian Computation (ABC)
- Probabilistic Programming
- Adversarial Variational Optimization (AVO)

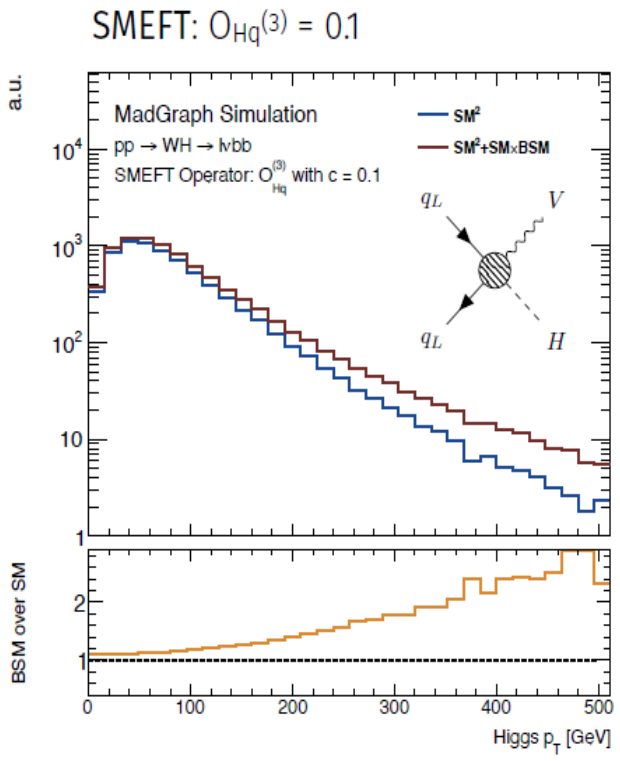
Learn simulator
(with deep learning)



- Generative Adversarial Networks (GANs), Variational Auto-Encoders (VAE)
- Likelihood ratio from classifiers (CARL)
- Autogregressive models, Normalizing Flows

What we usually do: number counting or single differential





High-dimensional event data x

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

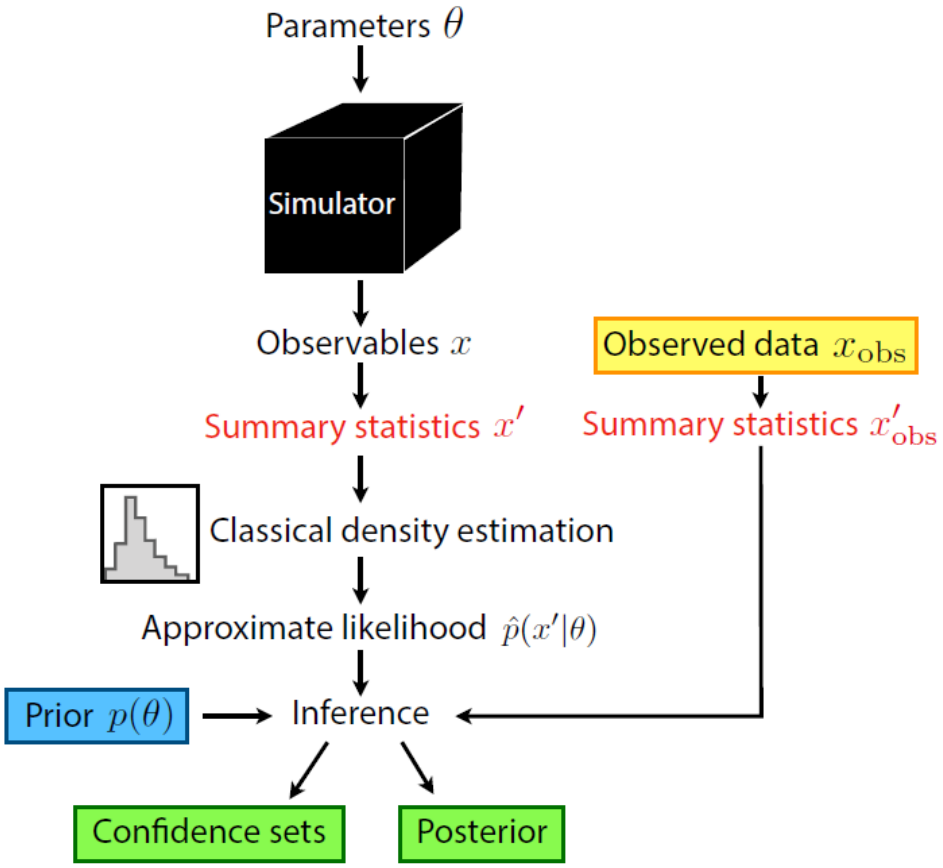
One or two summary statistics x'

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

n.b. "summary statistic" = a sensitive observable

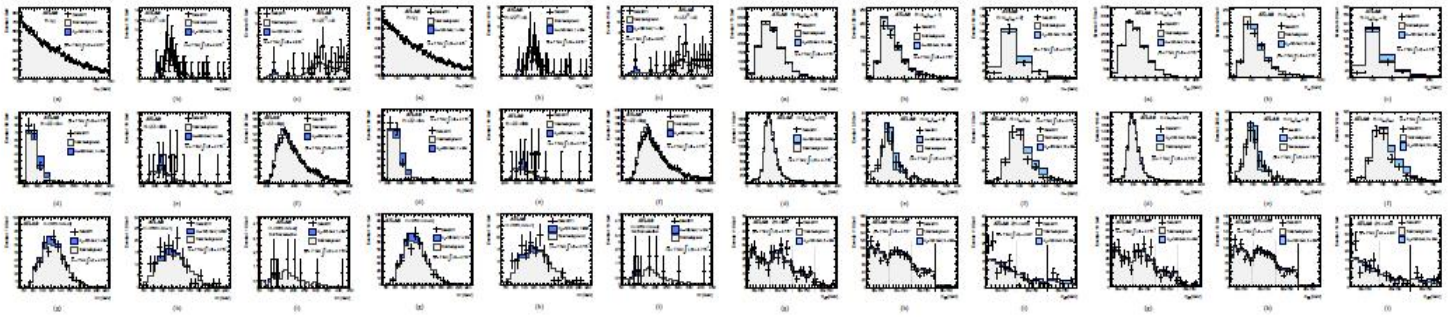
Inference by estimating the likelihood

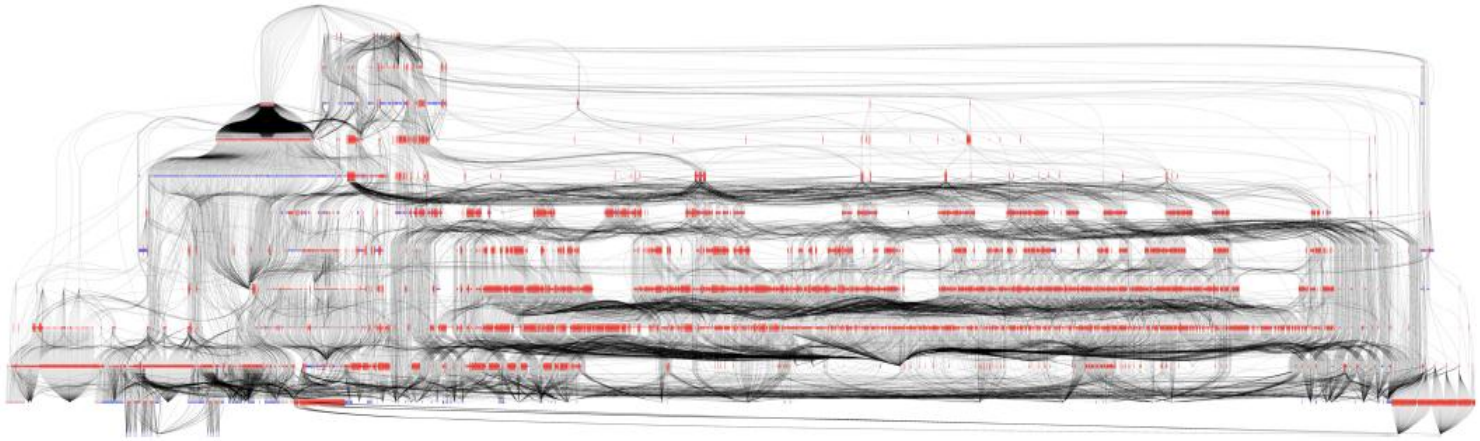
[e.g. P. Diggle, R. Gratton 1984]



- Compression to summary statistics loses information & reduces quality of inference
- Curse of dimensionality: does not scale to more than a few summary statistics
- Related alternative: Approximate Bayesian Computation (ABC) [D. Rubin 1984]

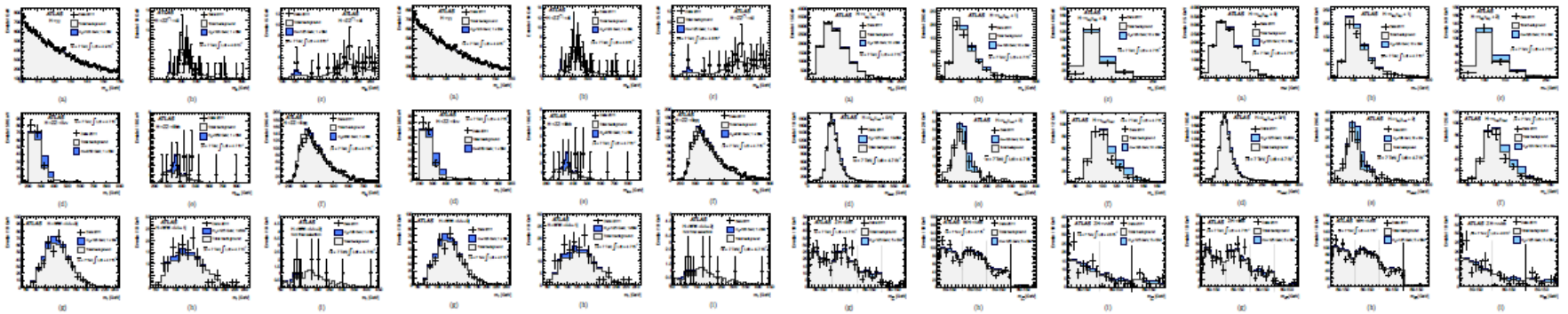
Combined fits for Higgs discovery

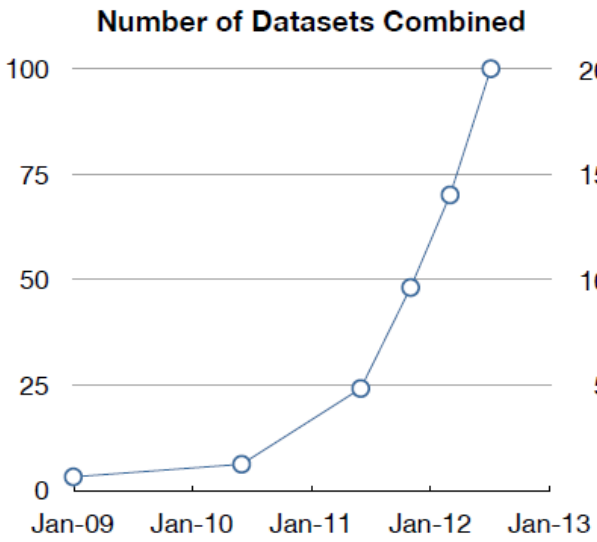


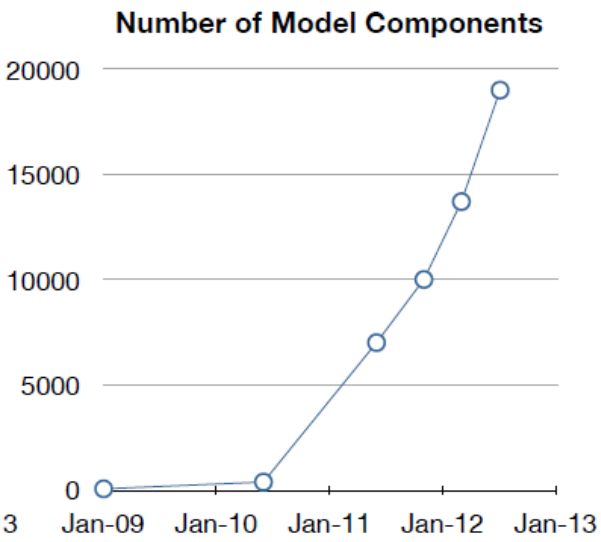


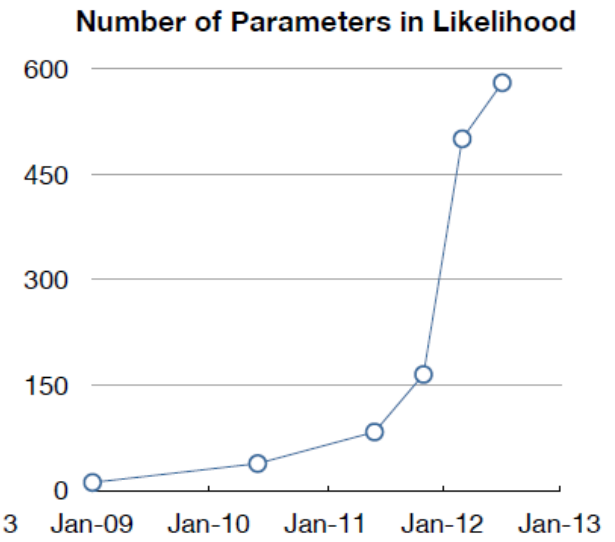
$$\mathrm{f}_{\mathrm{tot}}(\mathcal{D}_{\mathrm{sim}}, \mathcal{G}|\boldsymbol{\alpha}) = \prod_{c \in \mathrm{channels}} \left[\mathrm{Pois}(n_c|\nu_c(\boldsymbol{\alpha})) \prod_{e=1}^{n_c} f_c(x_{ce}|\boldsymbol{\alpha}) \right] \cdot \prod_{p \in \mathbb{S}} f_p(a_p|\alpha_p)$$

Colaborative statistial modeling





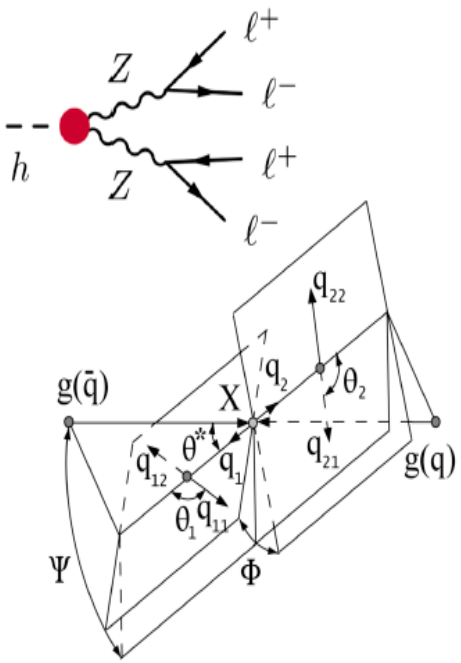


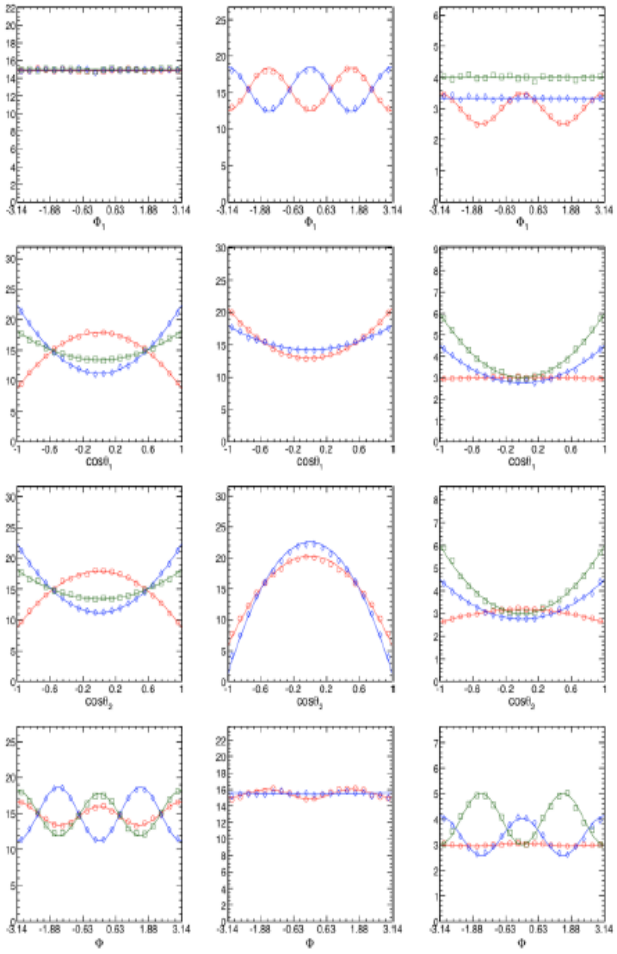


Summary statistics for LHC?

- In many LHC problems (eg. EFTs) there is no single good summary statistic: compressing to any x' loses information!
  [JB, K. Cranmer, F. Kling, T. Plehn 1612.05261; JB, F. Kling, T. Plehn, T. Tait 1712.02350]

- Ideally: analyze all trustworthy high-level features (reconstructed four-momenta…), or some form of low-level features, including correlations
  ("fully differential cross section")





[Bolognesi et al. 1208.4018]

Solve by approximating the integral

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

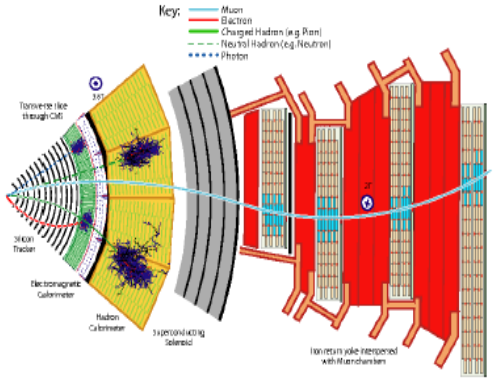
$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \; p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p(z_p|\theta)$$

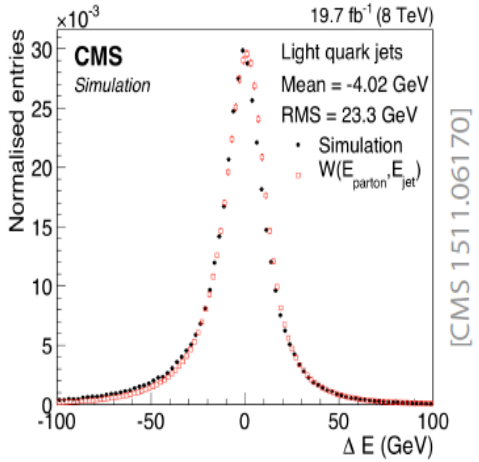
- Matrix Element Method (and similarly Optimal Observables): [K. Kondo 1988]
  - approximate shower + detector effects into transfer function $\hat{p}(x|z_p)$
  - explicitly calculate remaining integral

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

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

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





What if we could estimate the likelihood…

- for high-dimensional observables, including correlations?

  like MEM: no need to pick summary statistics

- including state-of-the-art shower and detector models?

  allowing for extra radiation, no need for transfer functions

- in microseconds?

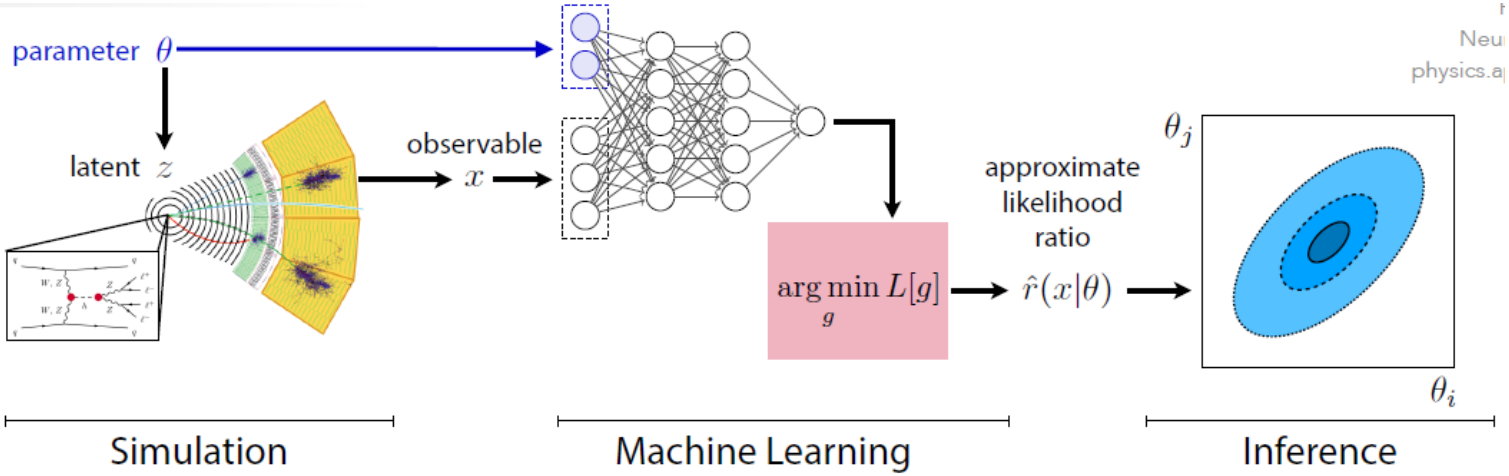
  amortized inference: train once, then always evaluate fast

- requiring less training examples than established machine learning methods?

  using matrix element information: "ML version of MEM"

Learning the likelihood ratio

Cranmer, Louppe, Pavez, arXiv:1506.02169
PNAS, arXiv:1805.12244
PRL, arXiv:1805.00013
PRD, arXiv:1805.00020
NeurIPS, arXiv:1808.00973
physics.aps.org/articles/v11/90



The **surrogate for the likelihood ratio** used for inference

A 2-stage process:

1. learning surrogate (**amortized**)
2. Inference on parameters of simulator (frequentist or Bayesian)

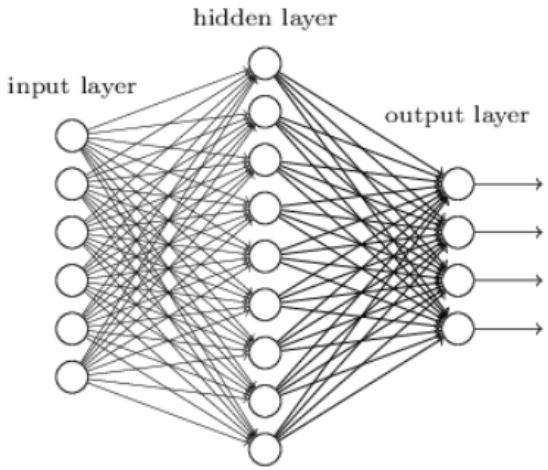
No Bayesian prior used for training, but one can use prior for inference.

NN = A highly flexible family of functions

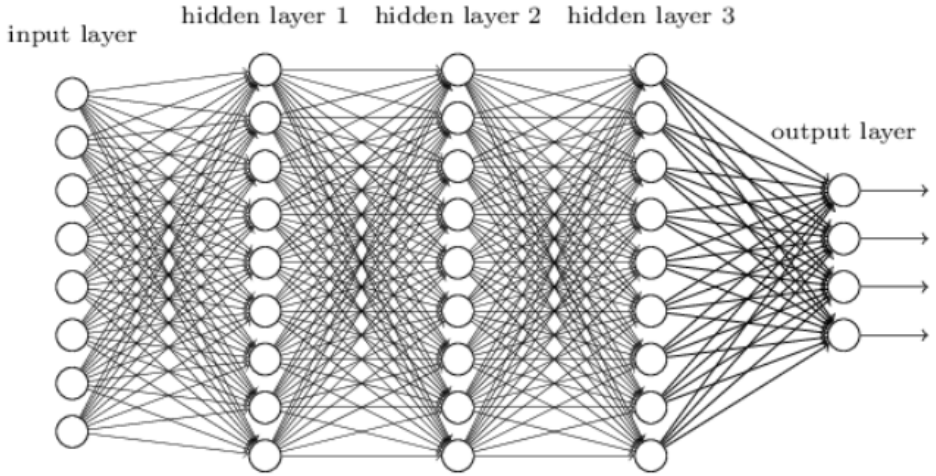
In calculus of variations, the optimization is over all functions: $\hat{s} = \underset{s}{\operatorname{argmin}} L[s]$

- In applied calculus of variations, we consider a highly flexible family of functions s_ϕ and optimize
- Think of neural networks as a highly flexible family of functions
- Machine learning also includes non-convex optimization algorithms that are affective even with millions of parameters!

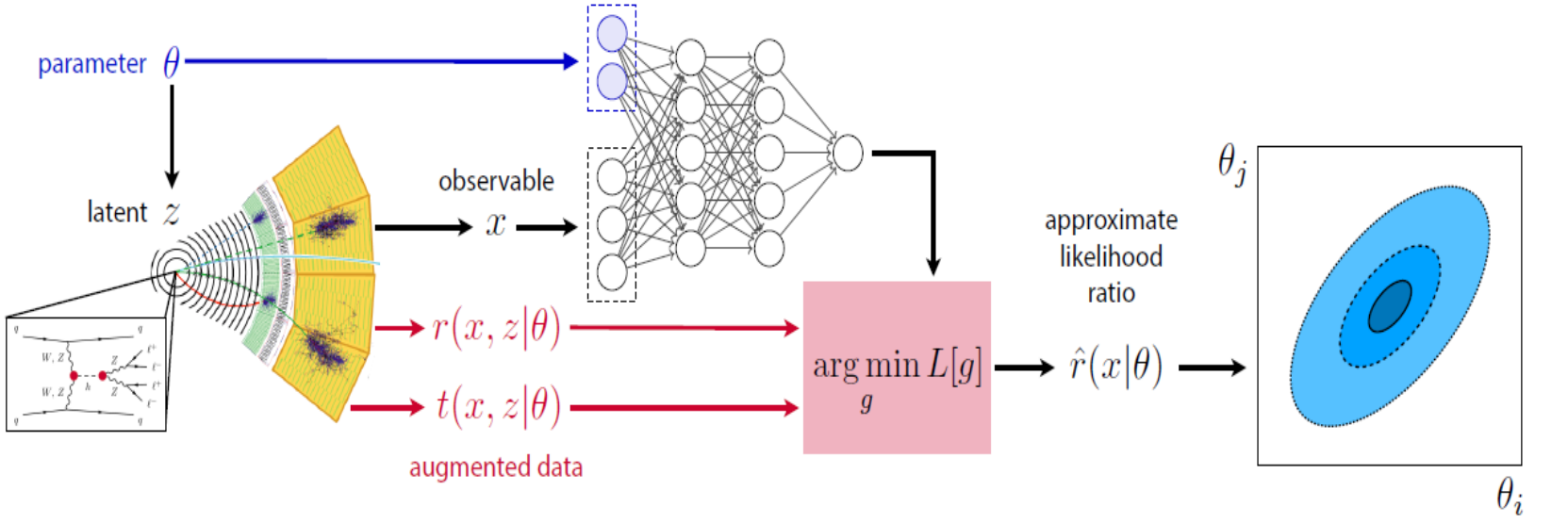
Shallow neural network



Deep neural network



Learning with Augmented Data



Simulation

"Mining gold": Extract additional information from simulator

Machine Learning

Use this information to train estimator for likelihood ratio

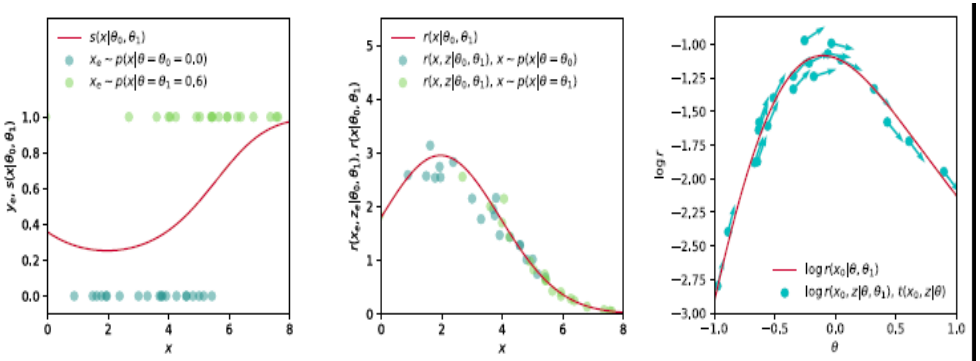
Inference

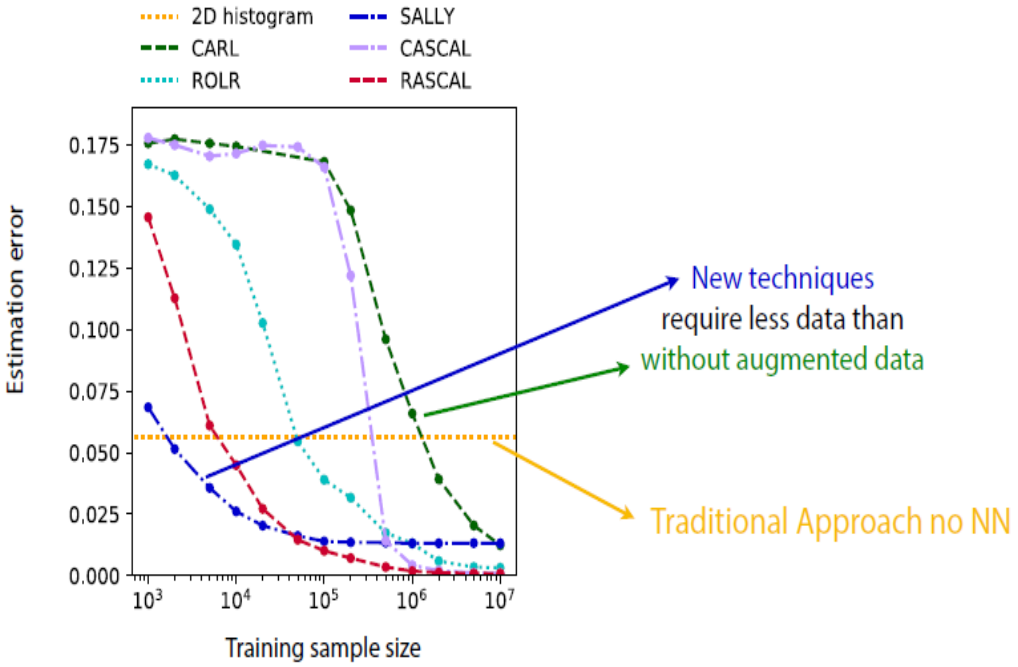
Limit setting with standard hypothesis tests

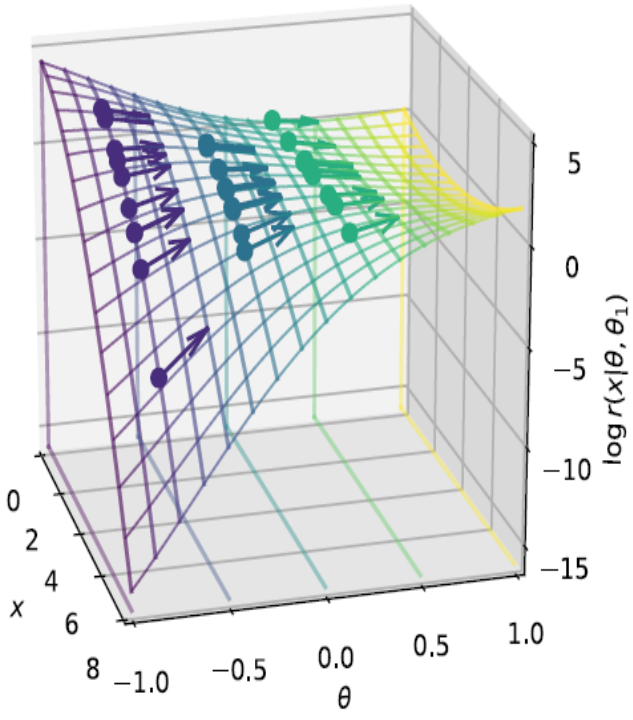
Gold mining: augmenting the training data

The augmented training data converts supervised classification into supervised regression with lower variance

- improvement in training efficiency

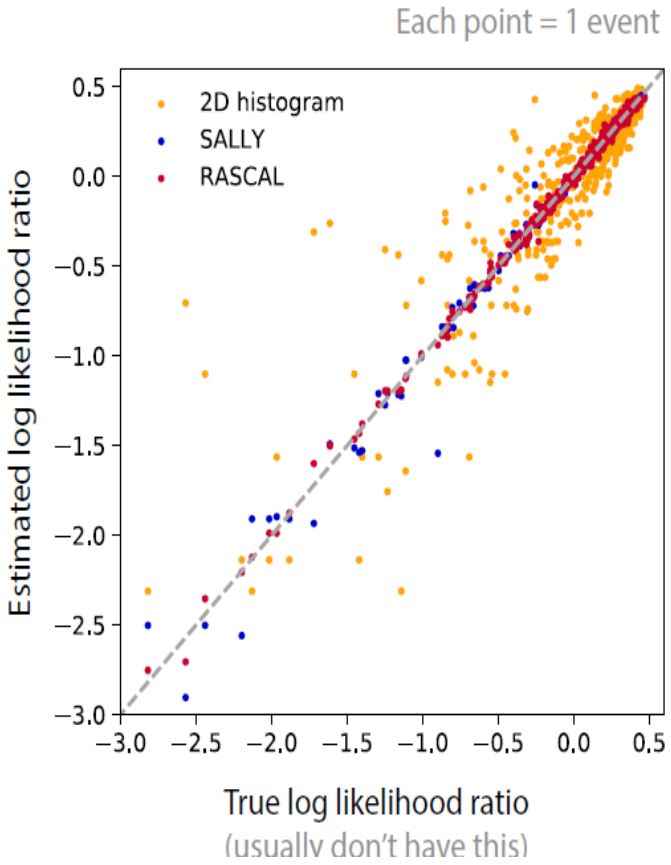


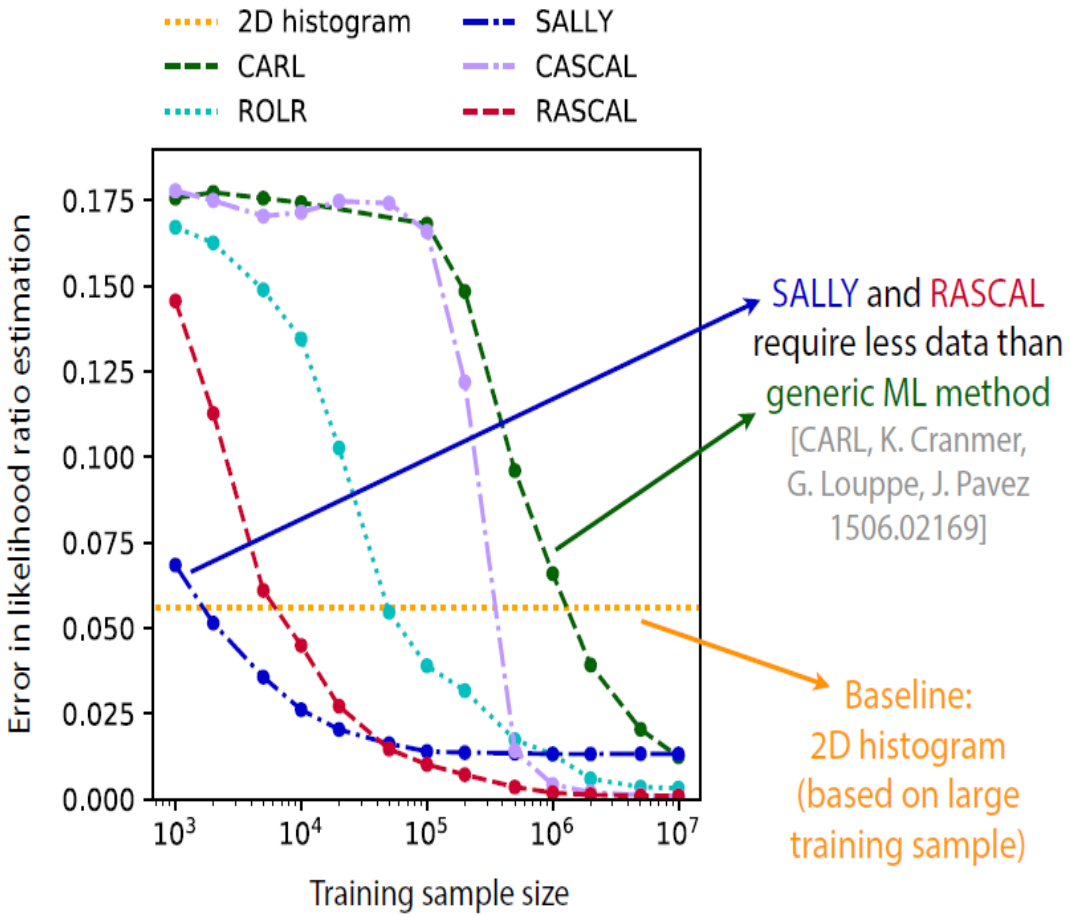




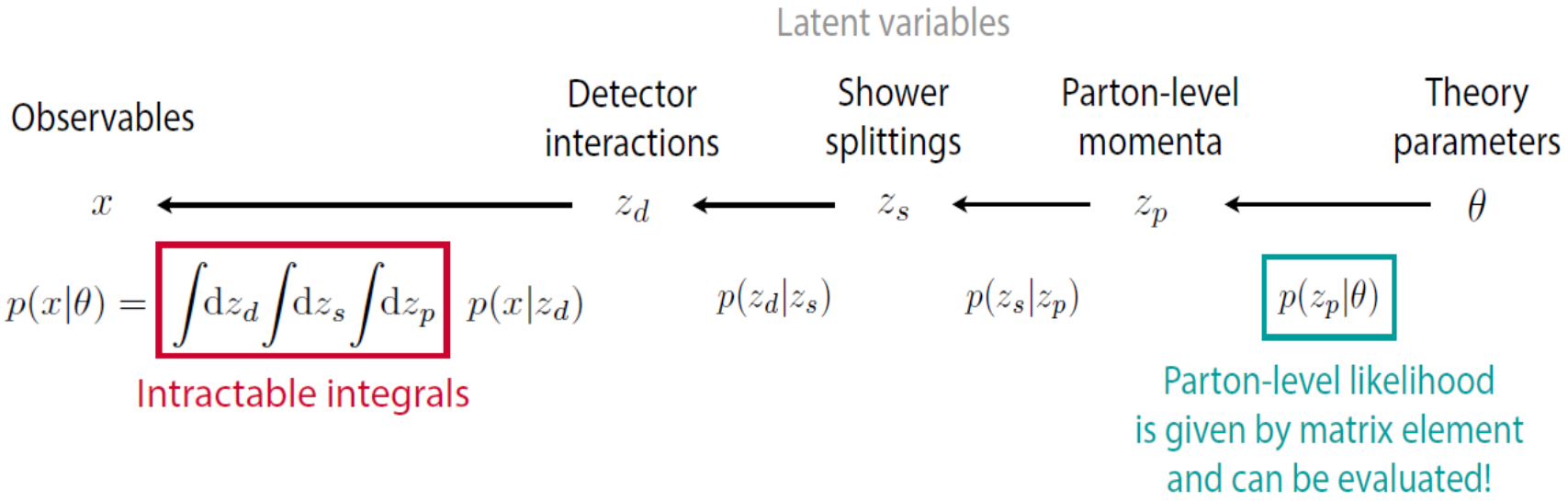
ehmer, Louppe, Pavez, KC, PNAS (2019), arXiv:1805.12244

More precise likelihood ratio estimates with less training data





Mining gold from the simulator



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

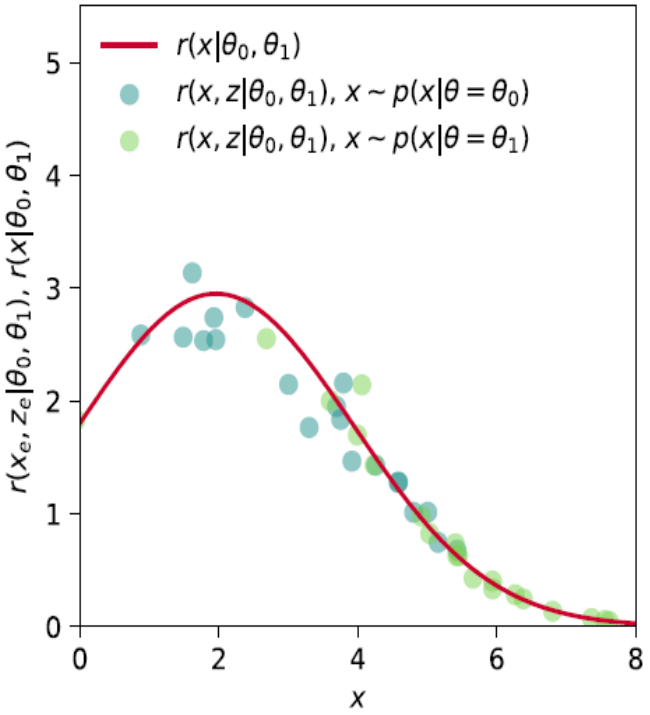
$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p|\theta_0)}{p(x, z_d, z_s, z_p|\theta_1)} = \frac{p(x|z_d)}{p(x|z_d)} \frac{p(z_d|z_s)}{p(z_d|z_s)} \frac{p(z_s|z_p)}{p(z_s|z_p)} \frac{p(z_p|\theta_0)}{p(z_p|\theta_1)} \sim \frac{|\mathcal{M}(z_p|\theta_0)|^2}{|\mathcal{M}(z_p|\theta_1)|^2}$$

The value of gold

We can calculate the joint likelihood ratio

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

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



We want the likelihood ratio function

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

The value of gold

We can calculate the joint likelihood ratio

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

We want the likelihood ratio function

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

With $r(x, z|\theta_0, \theta_1)$, we define a functional like

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

It is minimized by

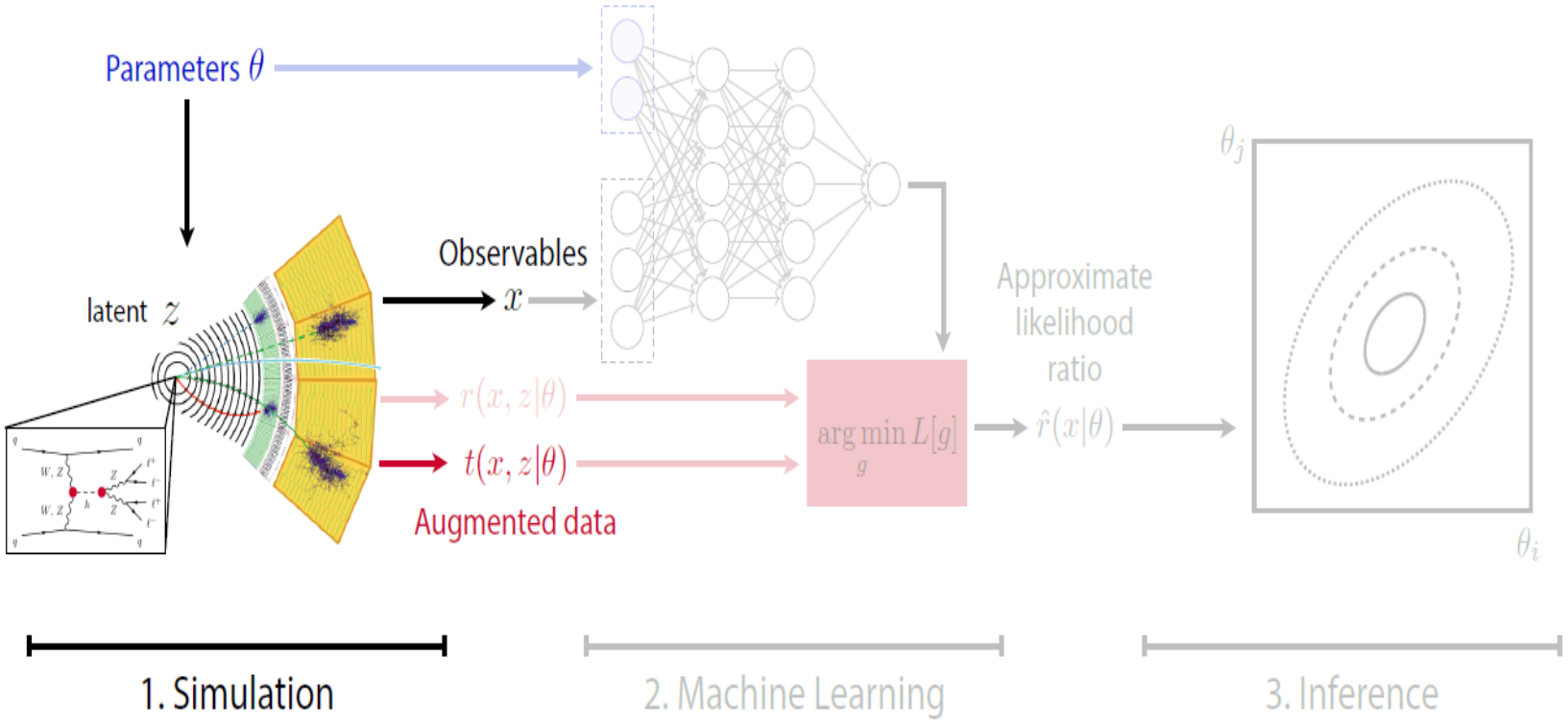
$$\mathbb{E}_{z\sim p(z|x,\theta_1)}\left[r(x, z|\theta_0, \theta_1)\right] = \underset{\hat{r}(x|\theta_0,\theta_1)}{\arg\min}\, L_r[\hat{r}(x|\theta_0, \theta_1)]!$$

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

.... and then magic ...

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

Learning with augmented data



Learning the score (related to optimal observables)

Similar to the joint likelihood ratio, from the simulator we can extract the joint score

$$t(x, z|\theta_0) \equiv \nabla_\theta \log p(x, z_d, z_s, z_p|\theta)\Big|_{\theta_0}$$

We want the score

$$t(x|\theta_0) \equiv \nabla_\theta \log p(x|\theta)\Big|_{\theta_0}$$

Given $t(x, z|\theta_0)$, we define the functional

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

One can show it is minimized by

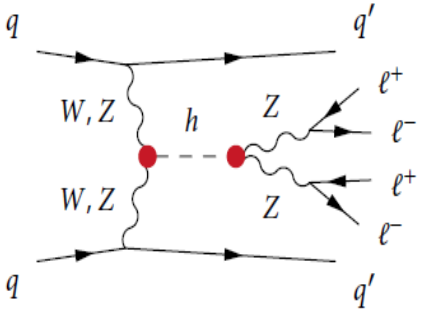
$$t(x|\theta_0) = \arg\min_{\hat{t}(x|\theta_0)} L_t[\hat{t}(x|\theta_0)]\,.$$

Again, we implement this minimization through machine learning.

Challenge for EFT

Let θ denote the coefficients of higher dimensional operators in the Lagrangian, x be high-dimensional data associated to an event, and $p(x \mid \theta) = \frac{1}{\sigma(\theta)} \frac{d\sigma}{d\theta}$ be the distribution for the data

- we want to compare any two points in EFT parameter space
- evaluate the **likelihood ratio** $r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$



Difficulty is that one changes the parameters of the EFT, the distributions p(x|θ) change due to interference.

- It would be very computationally expensive (infeasible) to generate samples for every value of θ and estimate $p(x|\theta)$ with histograms. Small changes mean we need a lot of MC events!
- Ideally we could directly estimate the **score** $t(x|\theta_0) \equiv \nabla_\theta \log p(x|\theta)\Big|_{\theta_0}$

EFT Embeded in a vector space

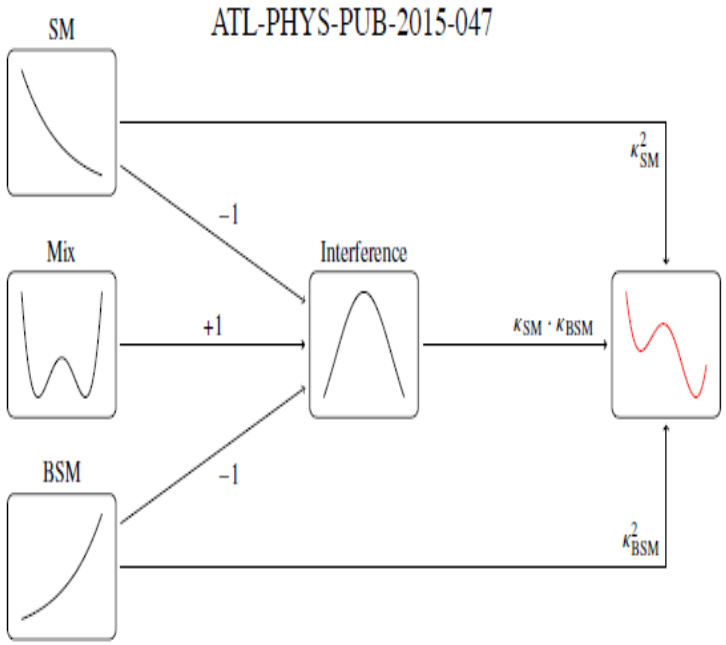
EFT Embedded in a vector space

Difficulty is that one changes the parameters of the EFT, the distributions p(x|θ) change due to interference.

But there is a trick:

Simple example:

$$|g_1 M_{SM} + g_2 M_{BSM}|^2 = g_1^2 |M_{SM}|^2 + 2 g_1 g_2 Re\left[M_{SM}^* M_{BSM}\right] + g_2^2 |M_{BSM}|^2$$



3-d vector space, distribution for any point in this space is linear mixture of distribution for 3 basis samples!

EFT Embeded in a vector space

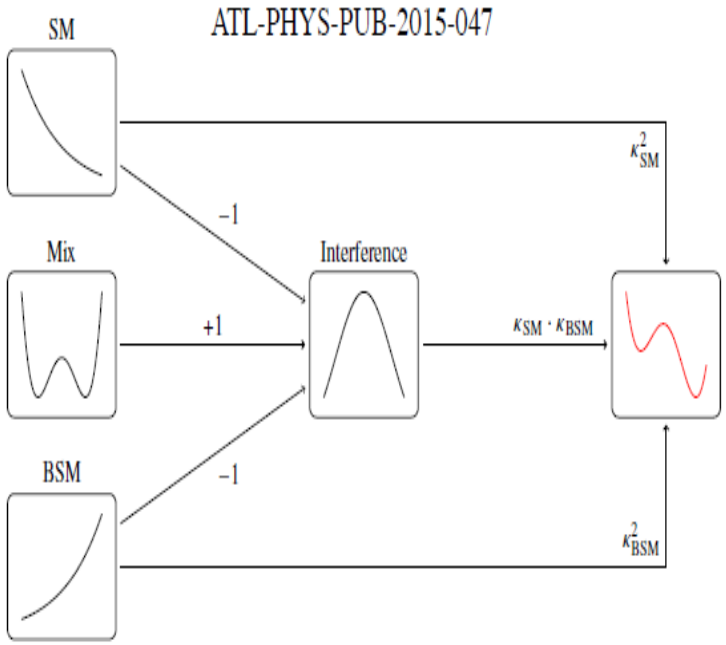
EFT Embedded in a vector space

Difficulty is that one changes the parameters of the EFT, the distributions p(x|θ) change due to interference.

But there is a trick:

Simple example:

$$|g_1 M_{SM} + g_2 M_{BSM}|^2 = g_1^2 |M_{SM}|^2 + 2 g_1 g_2 Re\left[M_{SM}^* M_{BSM}\right] + g_2^2 |M_{BSM}|^2$$



3-d vector space, distribution for any point in this space is linear mixture of distribution for 3 basis samples!

EFT decomposition

$$d\sigma \propto \left| \left(\mathcal{M}^p_{\mathrm{SM}} + \sum_i \frac{f_i}{\Lambda^2} \mathcal{M}^p_i \right) \left(\mathcal{M}^d_{\mathrm{SM}} + \sum_j \frac{f_j}{\Lambda^2} \mathcal{M}^d_j \right) \right|^2$$

production decay

Express EFT as a mixture:

$$p(x|\theta) = \sum_c w_c(\theta) p_c(x)$$

$w_c(\theta)$ are polynomials

$\nabla_\theta \log p(x|\theta)$ is now possible!

Process	Number of components for n operators					
	$\mathcal{O}(\Lambda^0)$	$\mathcal{O}(\Lambda^{-2})$	$\mathcal{O}(\Lambda^{-4})$	$\mathcal{O}(\Lambda^{-6})$	$\mathcal{O}(\Lambda^{-8})$	Σ
hV / WBF production	1	n	$\frac{n(n+1)}{2}$			$\frac{(n+1)(n+2)}{2}$
$h \to VV$ decay	1	n	$\frac{n(n+1)}{2}$			$\frac{(n+1)(n+2)}{2}$
Production + decay	1	n	$\frac{n(n+1)}{2}$	$\binom{n+2}{3}$	$\binom{n+3}{4}$	$\binom{n+4}{4}$

Table 1: Number of components c as given in Eq. (6) for different processes, sorted by their suppression by the EFT cutoff scale Λ.

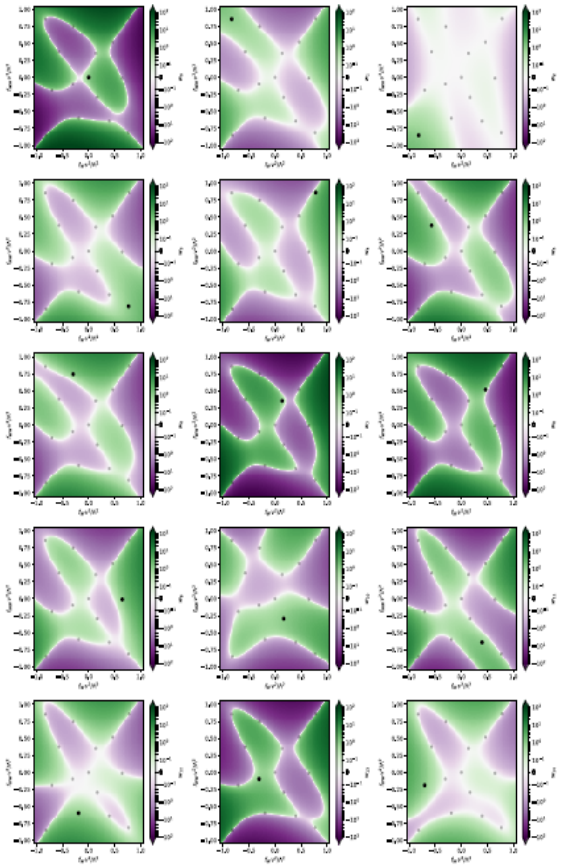


Figure 13: Morphing weights $w_1(\boldsymbol{\theta})$ for basis points distributed over the full relevant parameter space.

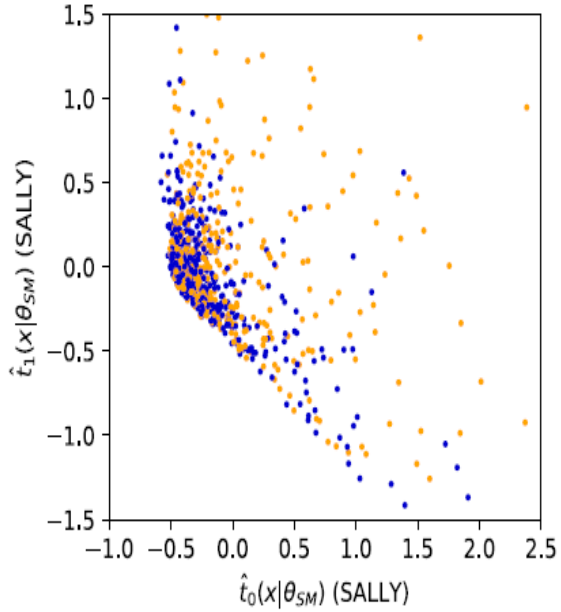
For 2 BSM operators affecting VBF Higgs production and decay, we need a 15-D vector space

For 5 BSM operators we need 126-D vector space

Locally sufficient statistics

One of the initial motivations for using ML to approximate the likelihood is that most summary statistics loose information.

However, the score provides "locally sufficient statistics" that capture all the information in the region of neighborhood of θ_0 (aka the standard model)

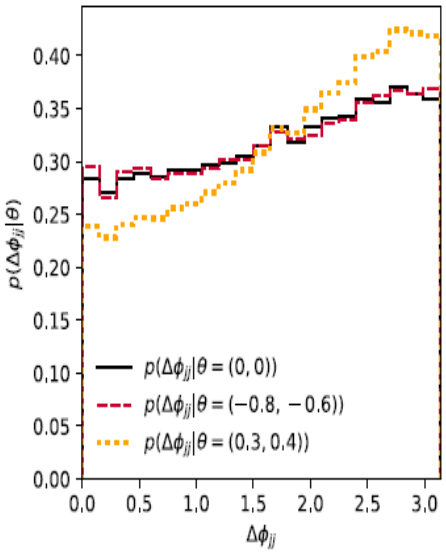


One summary statistic per parameter

$$p(x|\theta) \sim e^{t(x|\theta_{SM})\cdot(\theta-\theta_{SM})}$$

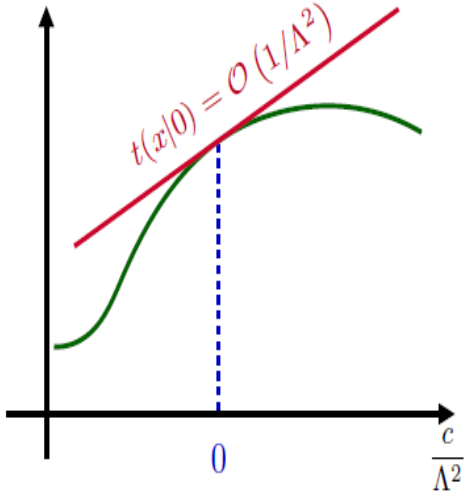
$$t(x|\theta_0) \equiv \nabla_\theta \log p(x|\theta)\Big|_{\theta_0}$$

Perfect match for EFT measurement

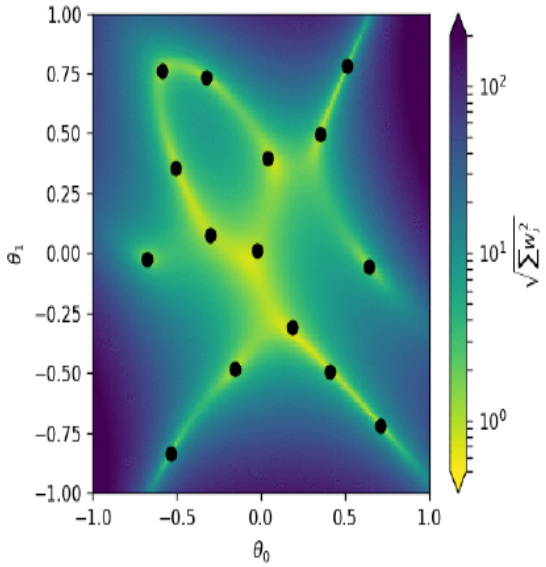


- Good for subtle kinematic effects

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



- Interference effects can be isolated using SALLY at the SM



- Morphing techniques allow fast reweighting to any parameter points

  [e.g. ATL-PHYS-PUB-2015-047]

Wrap-up of simulation-based inference method

Method	Approximations	Upfront cost	Eval
Summary statistics:			
Likelihood for summary stats (standard histograms)	Reduction to summary stats	Fast	Fast
Approximate Bayesian Computation	Reduction to summary stats	Depends	Depends
Matrix elements:			
Matrix Element Method	Transfer fns	Fast	Slow
Optimal Observables	Transfer fns, optimal only locally	Fast	Slow
Neural networks:			
Neural likelihood	NN	Needs many samples	Fast
Neural posterior	NN	Needs many samples	Fast
Neural likelihood ratio	NN	Needs many samples	Fast
Neural networks + matrix elements:			
Neural likelihood (ratio) + gold mining (RASCAL etc)	NN	Needs less samples	Fast
Neural optimal observables (SALLY)	NN, optimal only locally	Needs less samples	Fast

Discussion

Inference is always done within the context of a model

- If the model is mis-specified it will affect inference
- Here the model is the simulator, or the surrogate for the simulator
  - **One hand:** simulators usually include more effects than traditional approaches
  - **Other hand:** more chances for method to focus on aspects that are poorly modeled

Humans are good at designing robust summary statistics that are not sensitive to mis-modeled features in the data

- Now there are numerous approaches to build this into the training of ML models (related to domain adaptation, algorithmic fairness, pivotal quantities, profiling, etc.) eg. uBoost by J. Stevens, M. Williams, "learning to pivot" by KC, Louppe, Kagan

These methods **do not address hypothesis generation**.

- They are not designed to discover new laws of nature.

Conclusions

- **Machine learning can help us get more out of our simulators**
  - **It can provide effective statistical models**
- **Our understanding how to leverage our prio physics knowledge while letting machine learning do what it's good at is maturing.**
  - **build in robustness to systematics uncertainty**
  - **ability to inject and extract physics knowledge from models**
  - **exploit symmetries, hierarchical structure of data**
- **Harnessing full potential of theses techniques requires augmenting existing simulators.**

References

Opinionated review
K. Cranmer, J. Brehmer, G. Louppe:
"The frontier of simulation-based inference"
[1911.01429]

Review focusing on particle physics use-case
K. Cranmer and Johann Brehmer
"Simulation-based inference methods for particle physics"
[Artificial Intelligence for Particle Physics [book] and arXiv:2010.06439]

Do It Yourself (for LHC physics)
J. Brehmer, F. Kling, I. Espejo, K. Cranmer:
"MadMiner: Machine learning—based inference for particle physics"
[CSBS, 1907.10621, https://github.com/diana-hep/madminer]