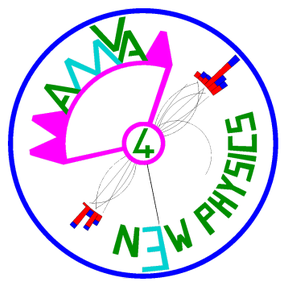
INFERNO: Inference-Aware Neural Optimisation

Pablo de Castro *(@pablodecm) and Tommaso Dorigo (@dorigo)*

19th March 2018 @ Likelihood-Free Inference Workshop (Flatiron Institute, NYC)

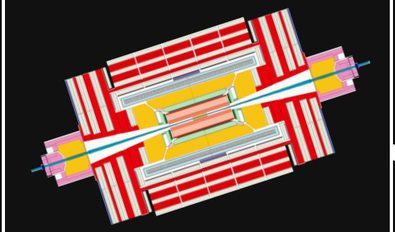
More details available on our preprint arxiv:1806.04743
as well as our GitHub code repository

online version of this slides available at
https://pdecastr.web.cern.ch/pdecastr/lfi_flatiron_march_2019

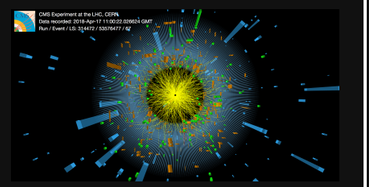


AMVA4NewPhysics has received funding from European Union's Horizon 2020 Programme under Grant Agreement number 675440

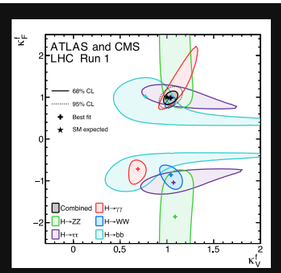
Statistical Inference in Particle Colliders



particle collider experiment (e.g. CMS)

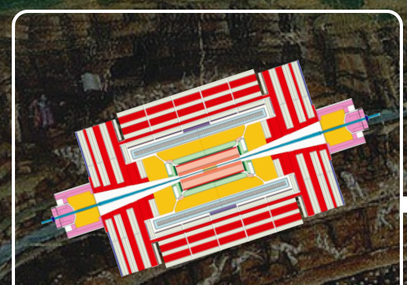


high-dim event data
$D = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_n\}$

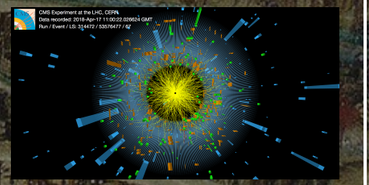


constraints on model parameters θ

Statistical Inference in Particle Colliders

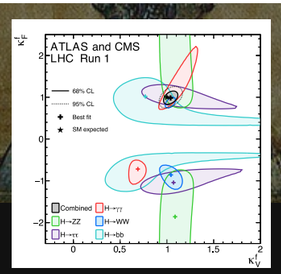


particle collider experiment (e.g. CMS)



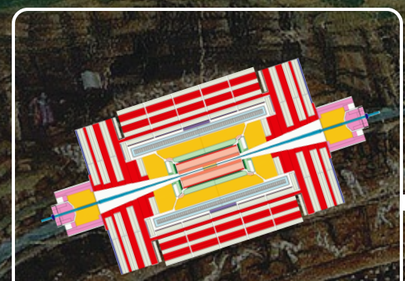
high-dim event data

$D = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_n\}$

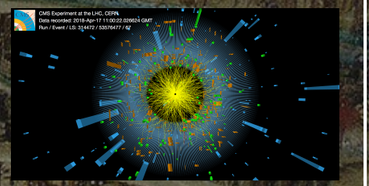


constraints on model parameters θ

Statistical Inference in Particle Colliders



particle collider experiment (e.g. CMS)

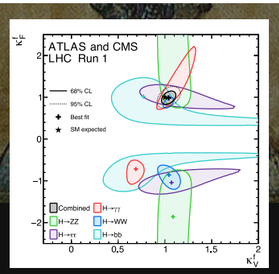


high-dim event data

$D = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_n\}$

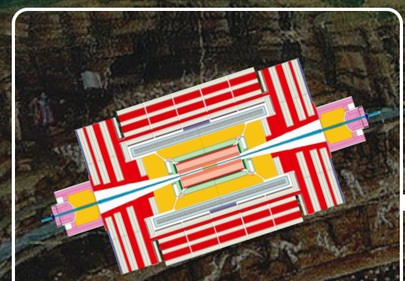
$p(\boldsymbol{x}|\boldsymbol{\theta})$

is intractable

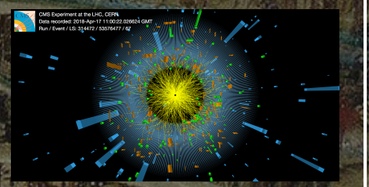


constraints on model parameters θ

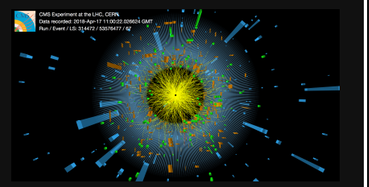
Statistical Inference in Particle Colliders



particle collider experiment (e.g. CMS)

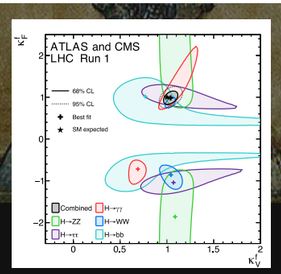


high-dim event data
$D = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_n\}$



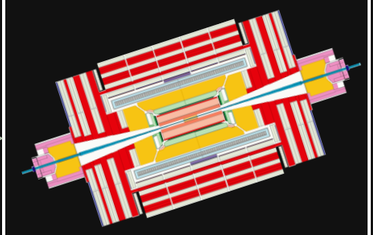
high-dim sim data
$G_s = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_g\}$

$p(\boldsymbol{x}|\boldsymbol{\theta})$
is intractable



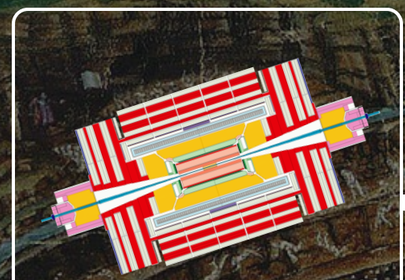
constraints on model parameters θ

nuisance parameters

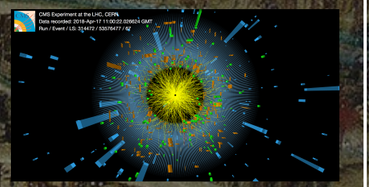


realistic simulator

Statistical Inference in Particle Colliders

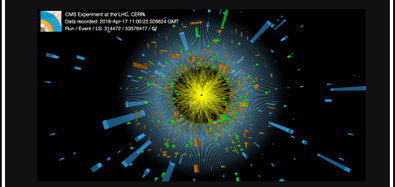


particle collider experiment (e.g. CMS)



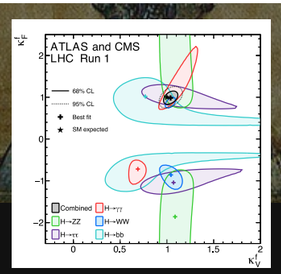
high-dim event data
$D = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_n\}$

curse of dimensionality



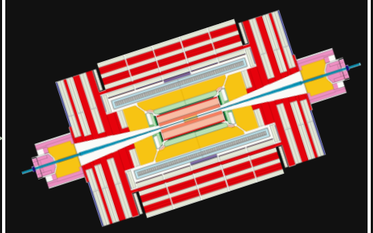
high-dim sim data
$G_s = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_g\}$

$p(x|\theta)$
is intractable



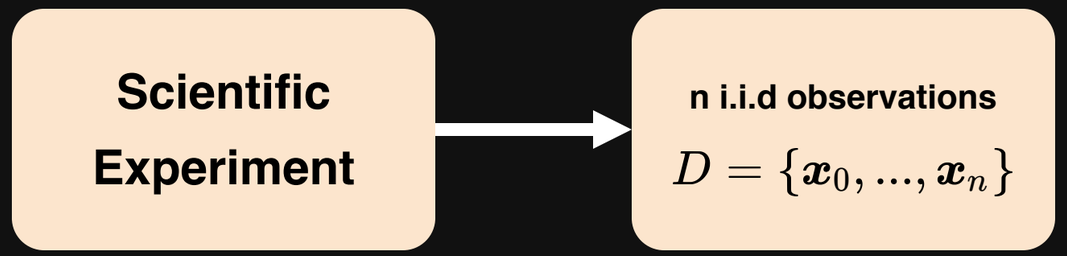
constraints on model parameters θ

nuisance parameters



realistic simulator

Simulation-based Inference



$\boldsymbol{\theta}$

Simulation-based Inference

Scientific Experiment

n i.i.d observations

$D = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_n\}$

Simulator

$\boldsymbol{x} \sim \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta_g})$

g i.i.d simulations

$G_s = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_g\}$

observations $\boldsymbol{x}$: $\mathcal{X} \subseteq \mathcal{R}^d$

$\boldsymbol{\theta_g}$

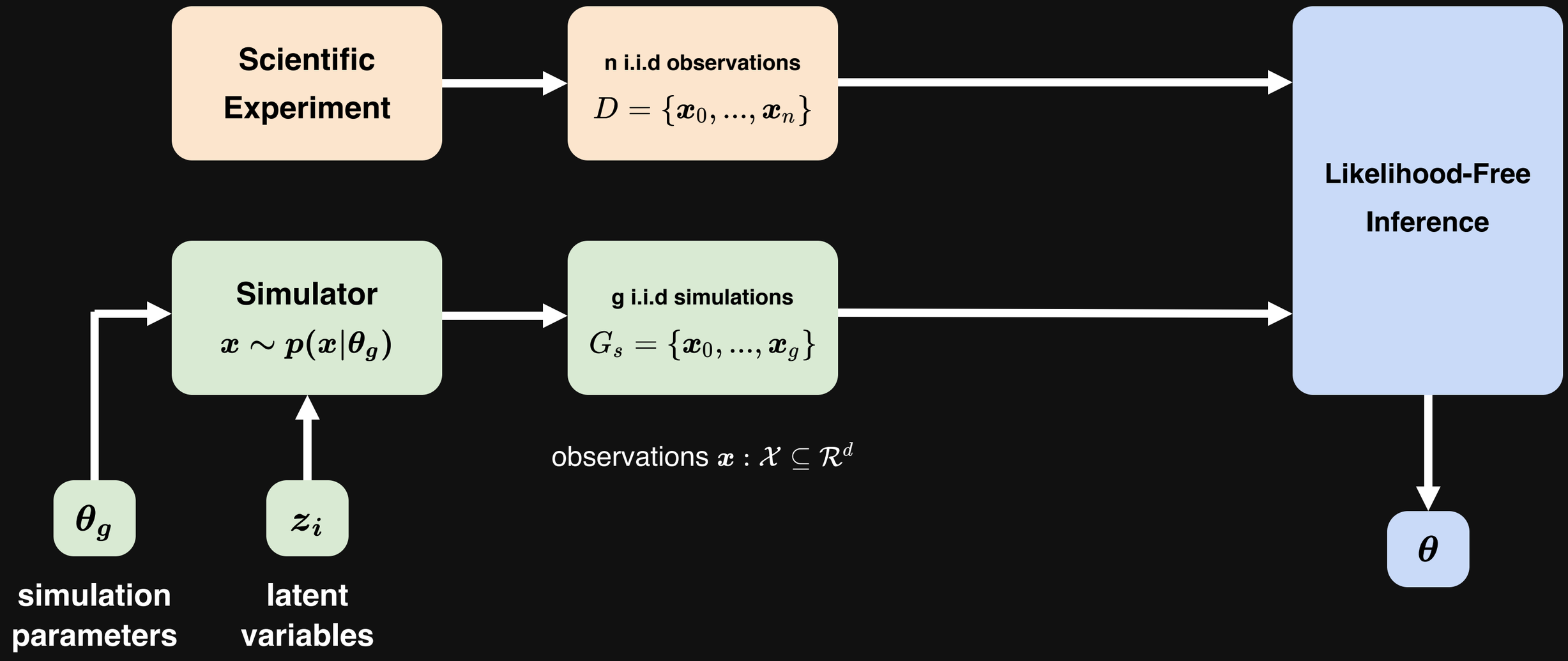
simulation parameters

$\boldsymbol{z_i}$

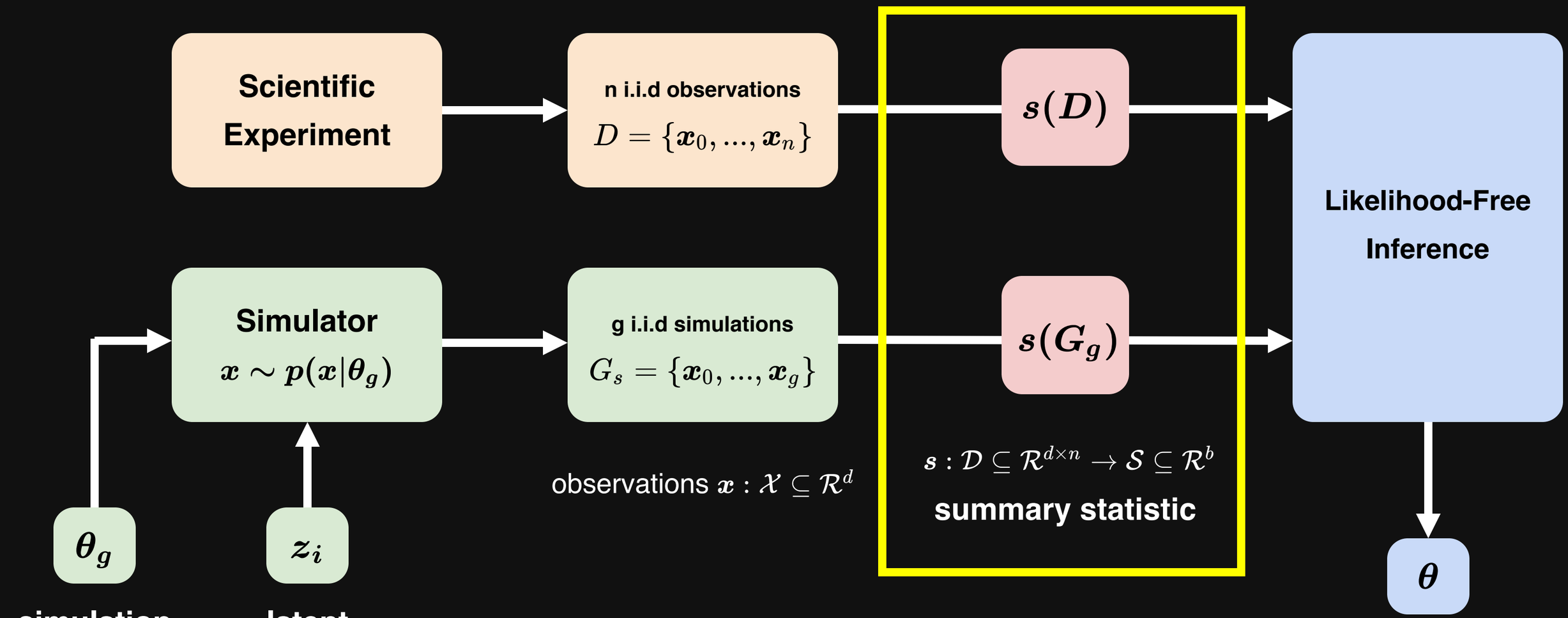
latent variables

$\boldsymbol{\theta}$

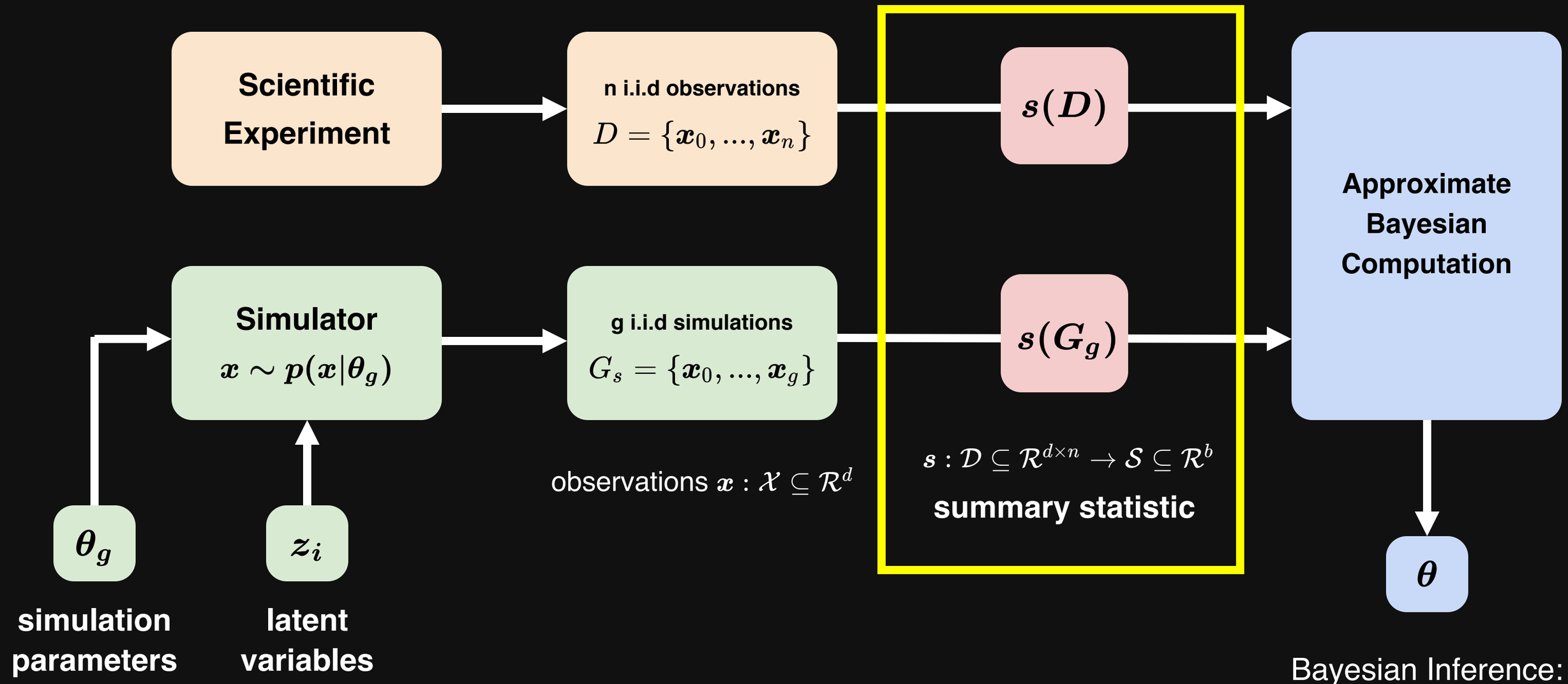
Simulation-based Inference



Simulation-based Inference



Simulation-based Inference



Simulation-based Inference

Scientific Experiment

n i.i.d observations

$D = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_n\}$

$\boldsymbol{s}(\boldsymbol{D})$

Simulator

$\boldsymbol{x} \sim \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta_g})$

g i.i.d simulations

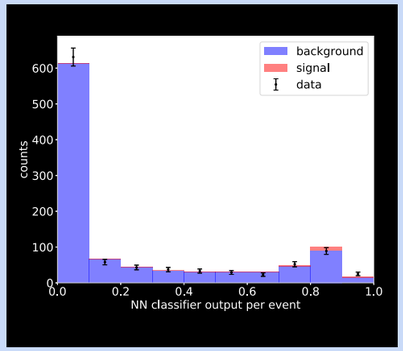
$G_s = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_g\}$

observations $\boldsymbol{x} : \mathcal{X} \subseteq \mathcal{R}^d$

$\boldsymbol{s}(\boldsymbol{G_g})$

$\boldsymbol{s} : \mathcal{D} \subseteq \mathcal{R}^{d \times n} \rightarrow \mathcal{S} \subseteq \mathcal{R}^b$

summary statistic



non-parametric likelihood

$\mathcal{L}(\theta) = \prod Poisson$

$\boldsymbol{\theta}$

Frequentist Inference: Confidence Interval or Hypothesis Testing

$\boldsymbol{\theta_g}$

simulation parameters

$\boldsymbol{z_i}$

latent variables

$p(\boldsymbol{x}|\mathrm{model})$ structure at particle colliders

simulation process $\sim$ sampling

observation

$\boldsymbol{x}$

theory parameters

θ

statistical inference

*Slide heavily inspired by those by J. Brehmer and K. Cranmer

$p(\boldsymbol{x}|\mathrm{model})$ structure at particle colliders

simulation process $\sim$ sampling

observation

$\boldsymbol{x}$

parton momenta

$\boldsymbol{z}_\mathrm{p}$

theory parameters

$\boldsymbol{\theta}$

$p(\boldsymbol{z}_\mathrm{p}|\boldsymbol{\theta})$

statistical inference

*Slide heavily inspired by those by J. Brehmer and K. Cranmer

$p(\boldsymbol{x}|\mathrm{model})$ structure at particle colliders

simulation process $\sim$ sampling

observation

$\boldsymbol{x}$

shower splittings

$\boldsymbol{z}_\mathrm{s}$

parton momenta

$\boldsymbol{z}_\mathrm{p}$

theory parameters

$\boldsymbol{\theta}$

$p(\boldsymbol{z}_\mathrm{s}|\boldsymbol{z}_\mathrm{p})$

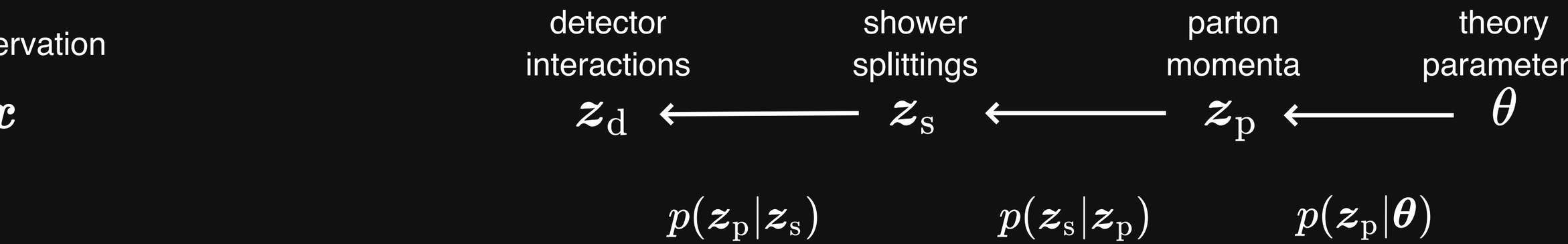
$p(\boldsymbol{z}_\mathrm{p}|\boldsymbol{\theta})$

statistical inference

*Slide heavily inspired by those by J. Brehmer and K. Cranmer

$p(\boldsymbol{x}|\text{model})$ structure at particle colliders

simulation process $\sim$ sampling



statistical inference

*Slide heavily inspired by those by J. Brehmer and K. Cranmer

$p(\boldsymbol{x}|\text{model})$ structure at particle colliders

simulation process $\sim$ sampling

observation $\boldsymbol{x}$ ← detector interactions $\boldsymbol{z}_\text{d}$ ← shower splittings $\boldsymbol{z}_\text{s}$ ← parton momenta $\boldsymbol{z}_\text{p}$ ← theory parameters θ

$p(x|\boldsymbol{z}_\text{d})$ $p(\boldsymbol{z}_\text{p}|\boldsymbol{z}_\text{s})$ $p(\boldsymbol{z}_\text{s}|\boldsymbol{z}_\text{p})$ $p(\boldsymbol{z}_\text{p}|\boldsymbol{\theta})$

statistical inference

*Slide heavily inspired by those by J. Brehmer and K. Cranmer

$p(\boldsymbol{x}|\mathrm{model})$ structure at particle colliders

simulation process $\sim$ sampling

observation | detector interactions | shower splittings | parton momenta | theory parameters

$\boldsymbol{x} \leftarrow \boldsymbol{z}_\mathrm{d} \leftarrow \boldsymbol{z}_\mathrm{s} \leftarrow \boldsymbol{z}_\mathrm{p} \leftarrow \theta$

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

intractable integral

statistical inference

*Slide heavily inspired by those by J. Brehmer and K. Cranmer

$p(\boldsymbol{x}|\text{model})$ structure at particle colliders

simulation process $\sim$ sampling

observation | detector interactions | shower splittings | parton momenta | theory parameters

$\boldsymbol{x} \leftarrow \boldsymbol{z}_\text{d} \leftarrow \boldsymbol{z}_\text{s} \leftarrow \boldsymbol{z}_\text{p} \leftarrow \boldsymbol{\theta}$

$p(x|\theta) = \int d\boldsymbol{z}_\text{p} \int d\boldsymbol{z}_\text{s} \int d\boldsymbol{z}_\text{p} \;\; p(x|\boldsymbol{z}_\text{d}) \;\; p(\boldsymbol{z}_\text{p}|\boldsymbol{z}_\text{s}) \;\; p(\boldsymbol{z}_\text{s}|\boldsymbol{z}_\text{p}) \;\; p(\boldsymbol{z}_\text{p}|\boldsymbol{\theta})$

intractable integral

$\sum_{j=0}^{K-1} p(z_i = j|\boldsymbol{\theta}) p(\boldsymbol{z}_\text{p}|\boldsymbol{\theta}, z_i = j)$

mixture model structure

statistical inference

*Slide heavily inspired by those by J. Brehmer and K. Cranmer

$p(\boldsymbol{x}|\text{model})$ structure at particle colliders

simulation process $\sim$ sampling

reconstruction $\sim$ regression of latent variables

observation | detector interactions | shower splittings | parton momenta | theory parameters

$\boldsymbol{x} \leftarrow \boldsymbol{z}_\text{d} \leftarrow \boldsymbol{z}_\text{s} \leftarrow \boldsymbol{z}_\text{p} \leftarrow \boldsymbol{\theta}$

$p(x|\theta) = \int d\boldsymbol{z}_\text{p} \int d\boldsymbol{z}_\text{s} \int d\boldsymbol{z}_\text{p} \;\; p(x|\boldsymbol{z}_\text{d}) \;\; p(\boldsymbol{z}_\text{p}|\boldsymbol{z}_\text{s}) \;\; p(\boldsymbol{z}_\text{s}|\boldsymbol{z}_\text{p}) \;\; p(\boldsymbol{z}_\text{p}|\boldsymbol{\theta})$

intractable integral

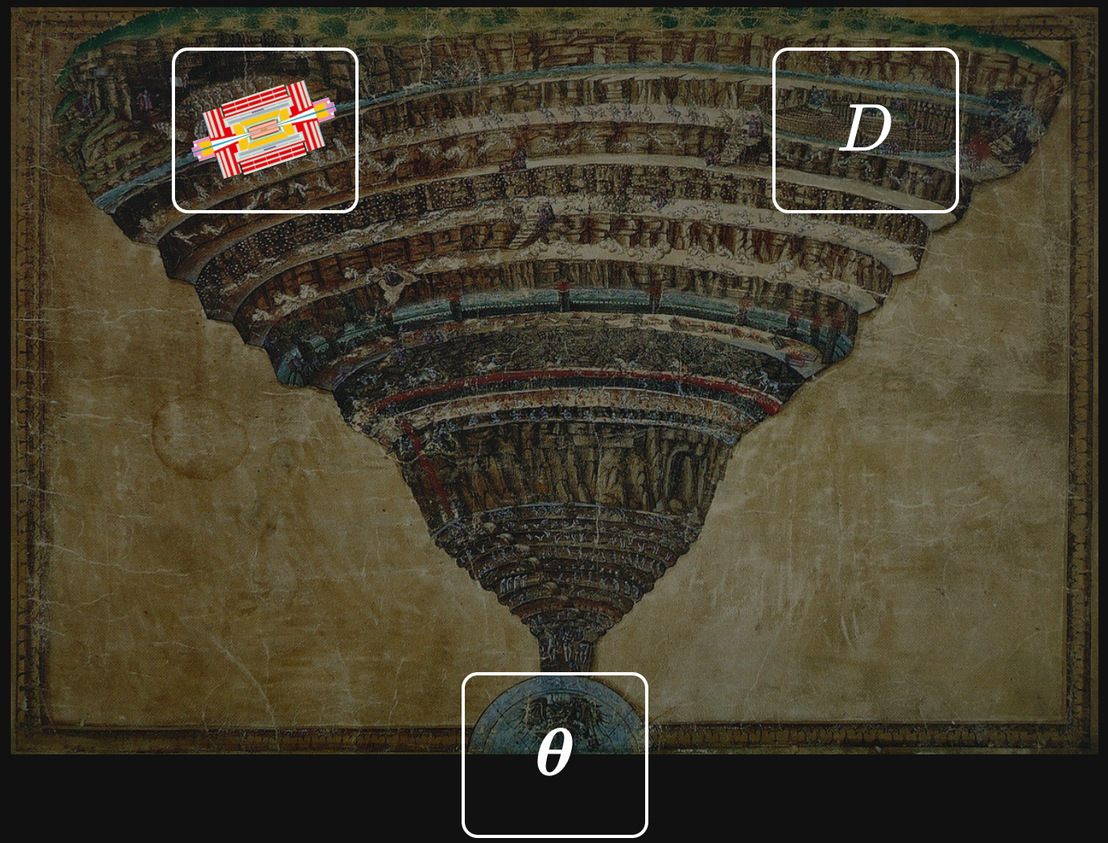
$\sum_{j=0}^{K-1} p(z_i = j|\boldsymbol{\theta}) p(\boldsymbol{z}_\text{p}|\boldsymbol{\theta}, z_i = j)$

mixture model structure

statistical inference

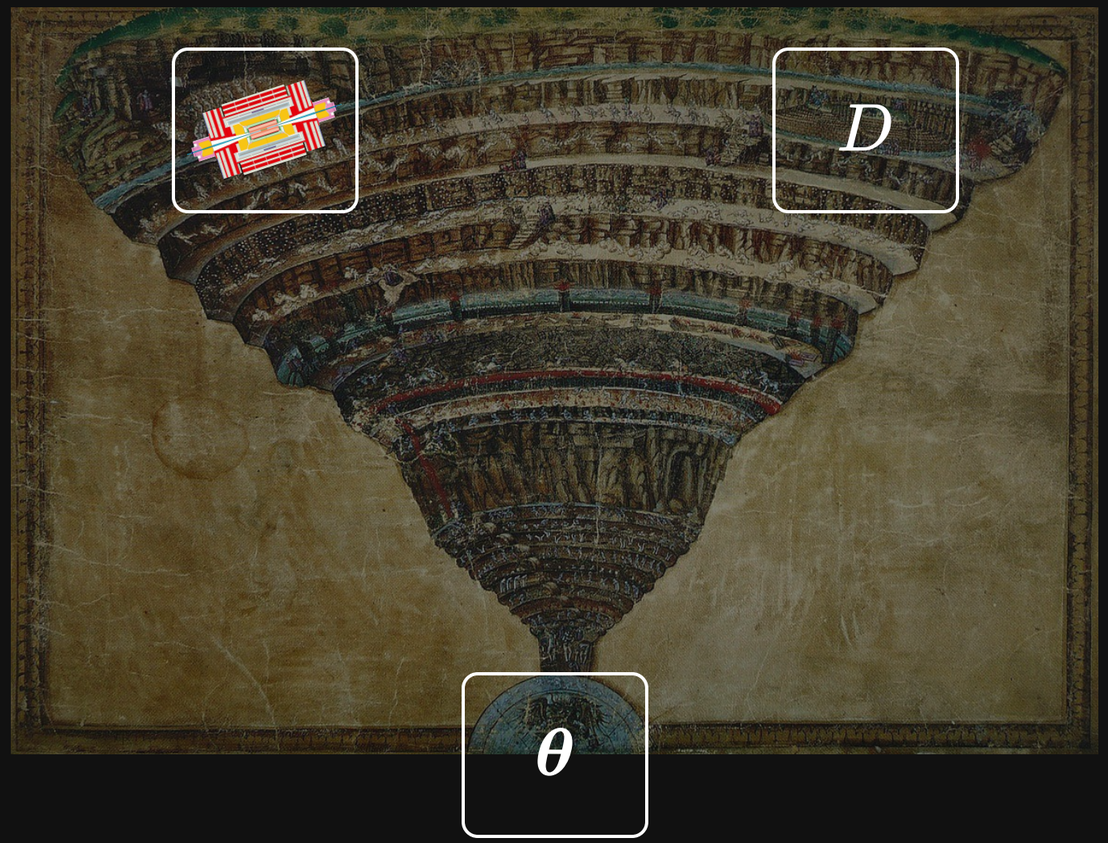
*Slide heavily inspired by those by J. Brehmer and K. Cranmer

The *hard problem* of choosing a $s(D)$



A low-dimensional summary statistics is required for inference, yet a naive choice will result on an important information loss

The *hard problem* of choosing a $s(D)$



A low-dimensional summary statistics is required for inference, yet a naive choice will result on an important information loss

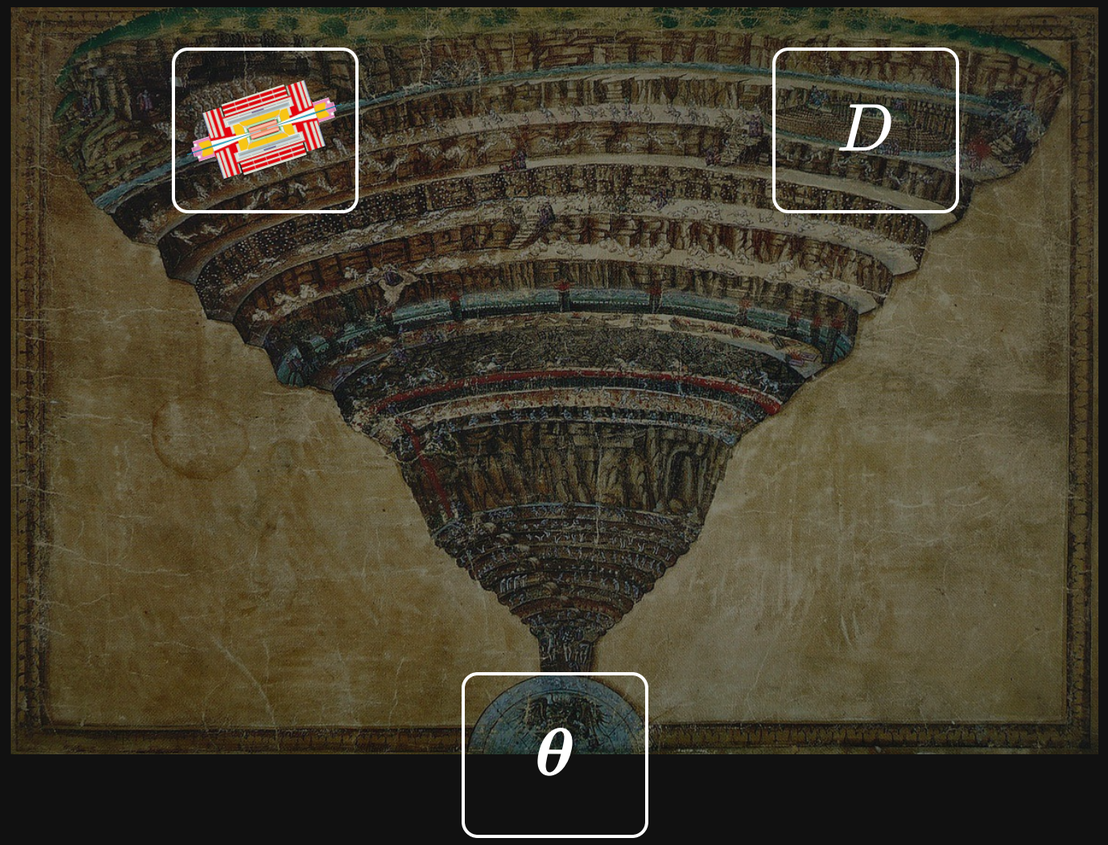
Ideally we want a sufficient summary statistic:

$$p(D|\boldsymbol{\theta}) = h(D)g\left(\boldsymbol{s}(D)|\boldsymbol{\theta}\right)$$

classical sufficiency

cannot be obtained in general and might not exist

The *hard problem* of choosing a $s(D)$



A low-dimensional summary statistics is required for inference, yet a naive choice will result on an important information loss

Ideally we want a sufficient summary statistic:

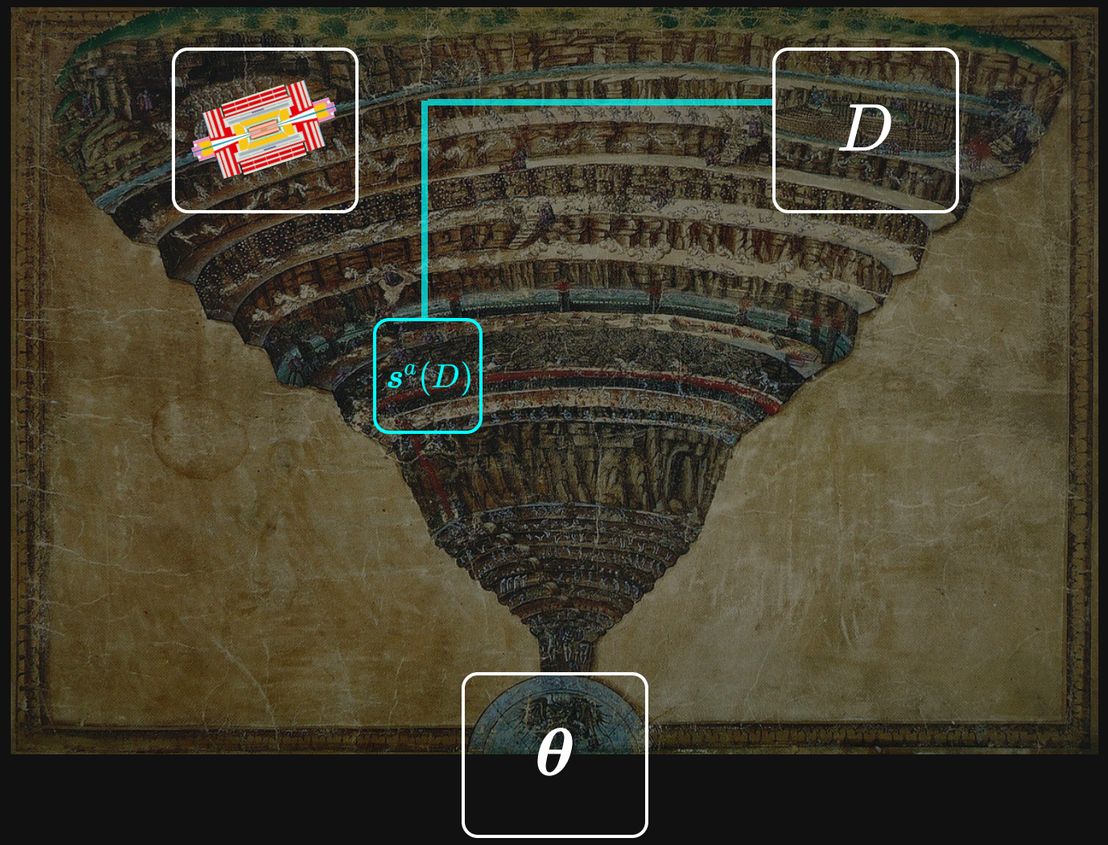
$$p(D|\boldsymbol{\theta}) = h(D)g\left(\boldsymbol{s}(D)|\boldsymbol{\theta}\right)$$

classical sufficiency

cannot be obtained in general
and might not exist

How to obtain good a summary statistic $s(D)$?

The *hard problem* of choosing a $s(D)$



A low-dimensional summary statistics is required for inference, yet a naive choice will result on an important information loss

Ideally we want a sufficient summary statistic:

$$p(D|\boldsymbol{\theta}) = h(D)g\left(\boldsymbol{s}(D)|\boldsymbol{\theta}\right)$$

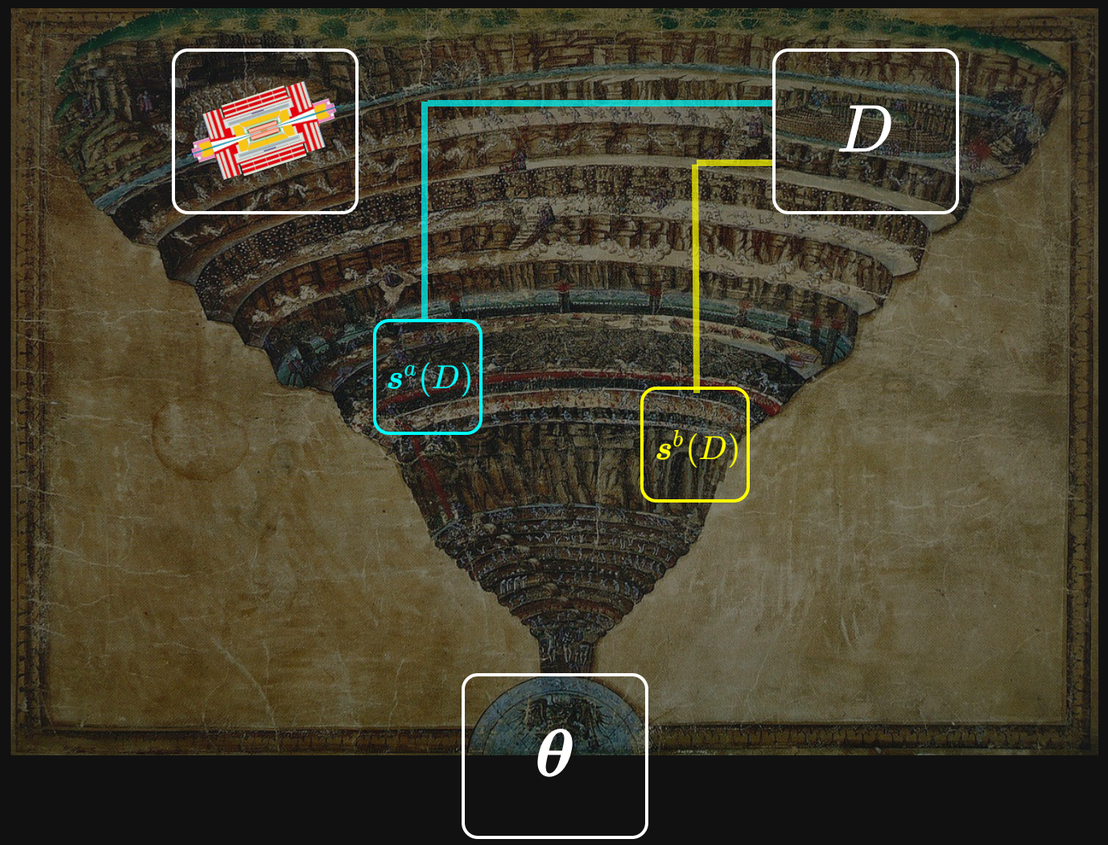
classical sufficiency

cannot be obtained in general and might not exist

How to obtain good a summary statistic $s(D)$?

a) Use domain inspired (i.e. physics) variables based on a combination of reconstructed variables

The *hard problem* of choosing a $s(D)$



A low-dimensional summary statistics is required for inference, yet a naive choice will result on an important information loss

Ideally we want a sufficient summary statistic:

$$p(D|\boldsymbol{\theta}) = h(D)g\left(\boldsymbol{s}(D)|\boldsymbol{\theta}\right)$$

classical sufficiency

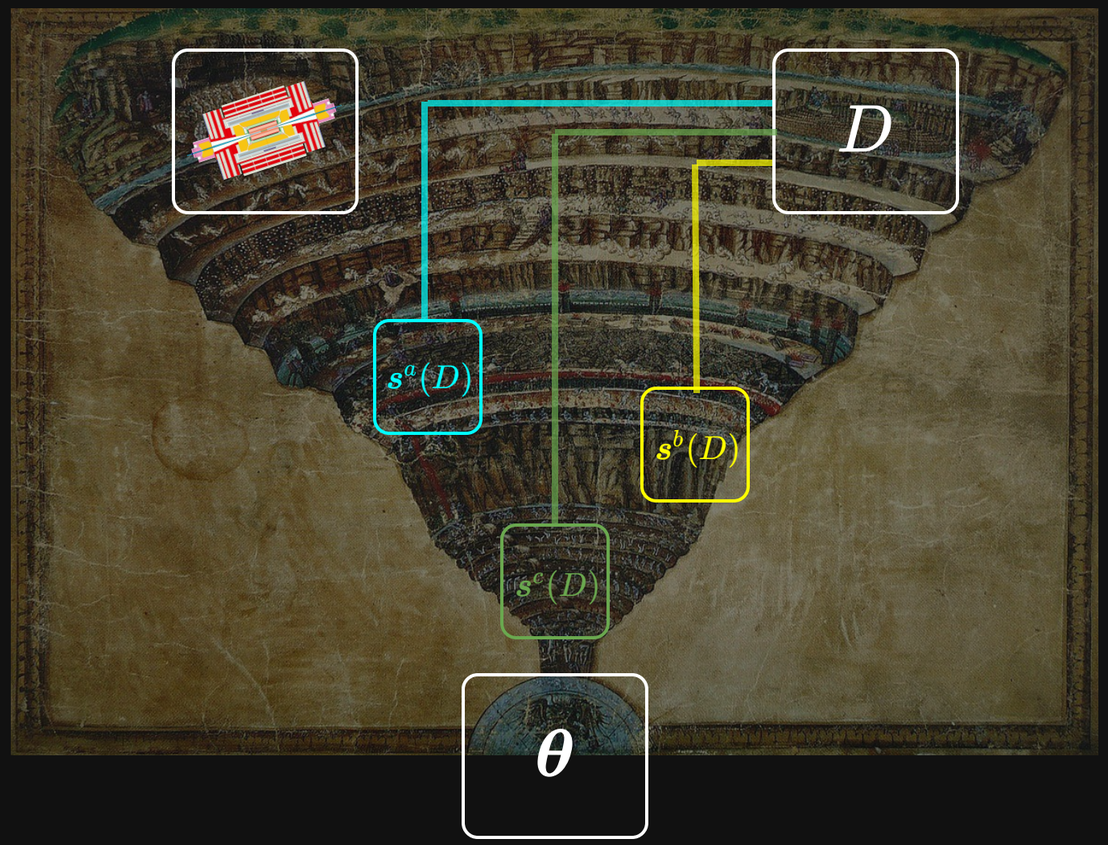
cannot be obtained in general
and might not exist

How to obtain good a summary statistic $s(D)$?

a) Use domain inspired (i.e. physics) variables based on a combination of reconstructed variables

b) Use the prediction of a machine learning classification or regression model trained on simulated observations

The *hard problem* of choosing a $s(D)$



A low-dimensional summary statistics is required for inference, yet a naive choice will result on an important information loss

Ideally we want a sufficient summary statistic:

$$p(D|\boldsymbol{\theta}) = h(D)g\left(\boldsymbol{s}(D)|\boldsymbol{\theta}\right)$$

classical sufficiency

cannot be obtained in general
and might not exist

How to obtain good a summary statistic $s(D)$?

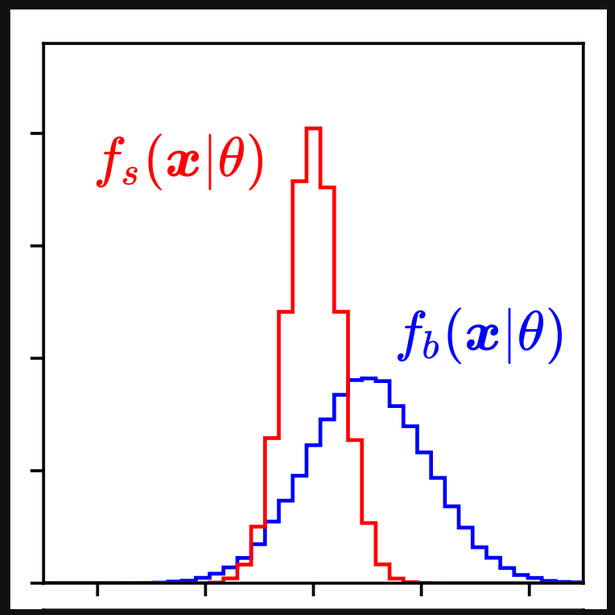
a) Use domain inspired (i.e. physics) variables based on a combination of reconstructed variables

b) Use the prediction of a machine learning classification or regression model trained on simulated observations

c) Use the technique presented in this talk or one of the alternatives for obtaining powerful $s(x)$ proposed in this workshop

An overused trick to obtain a good $s(x)$

IF



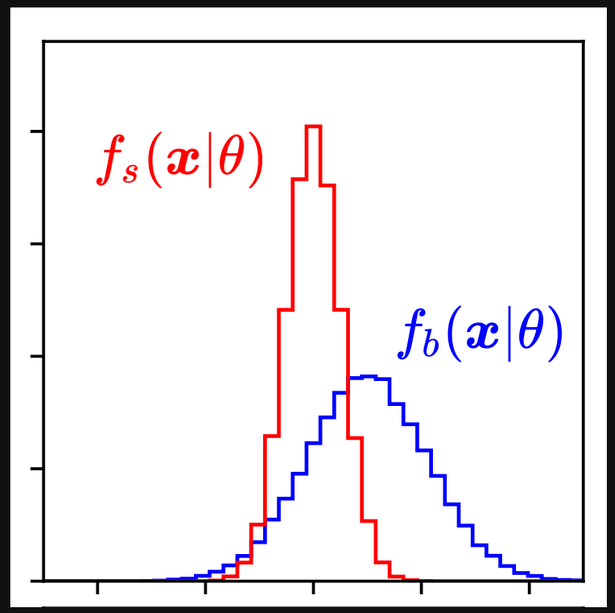
$$p(\boldsymbol{x}|\mu, \boldsymbol{\theta}) = \mu\, f_s(\boldsymbol{x}|\boldsymbol{\theta}) + (1-\mu)\, f_b(\boldsymbol{x}|\boldsymbol{\theta})$$

2-component mixture

and we only want to infer about the mixture fraction μ, **other θ are fixed and known**

An overused trick to obtain a good $s(x)$

IF



$$p(\boldsymbol{x}|\mu, \boldsymbol{\theta}) = \mu\, f_s(\boldsymbol{x}|\boldsymbol{\theta}) + (1-\mu)\, f_b(\boldsymbol{x}|\boldsymbol{\theta})$$

2-component mixture

and we only want to infer about the mixture fraction μ, **other θ are fixed and known**

THEN

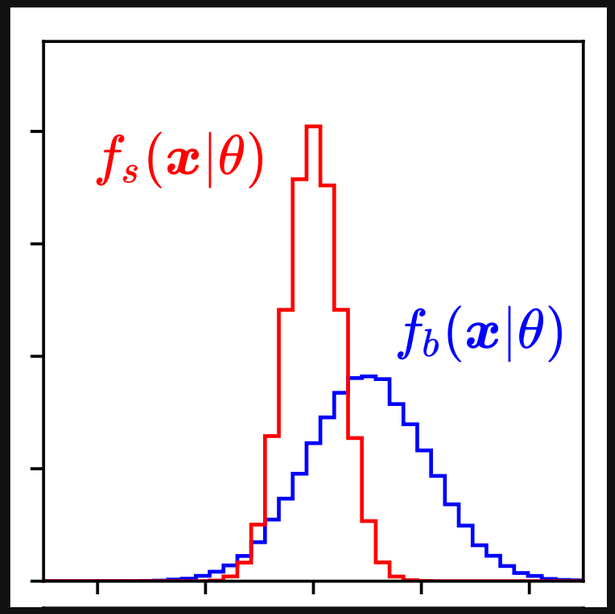
$$s_{\text{clf}}(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{f_s(\boldsymbol{x}|\boldsymbol{\theta})}{f_s(\boldsymbol{x}|\boldsymbol{\theta}) + f_b(\boldsymbol{x}|\boldsymbol{\theta})}$$

bayes optimal classifier

is a **one-dimensional summary statistic** that is **sufficient for inference about μ**

An overused trick to obtain a good $s(x)$

IF



$$p(\boldsymbol{x}|\mu, \boldsymbol{\theta}) = \mu\, f_s(\boldsymbol{x}|\boldsymbol{\theta}) + (1-\mu)\, f_b(\boldsymbol{x}|\boldsymbol{\theta})$$

2-component mixture

and we only want to infer about the mixture fraction μ, **other θ are fixed and known**

THEN

$$s_{\text{clf}}(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{f_s(\boldsymbol{x}|\boldsymbol{\theta})}{f_s(\boldsymbol{x}|\boldsymbol{\theta}) + f_b(\boldsymbol{x}|\boldsymbol{\theta})}$$

bayes optimal classifier

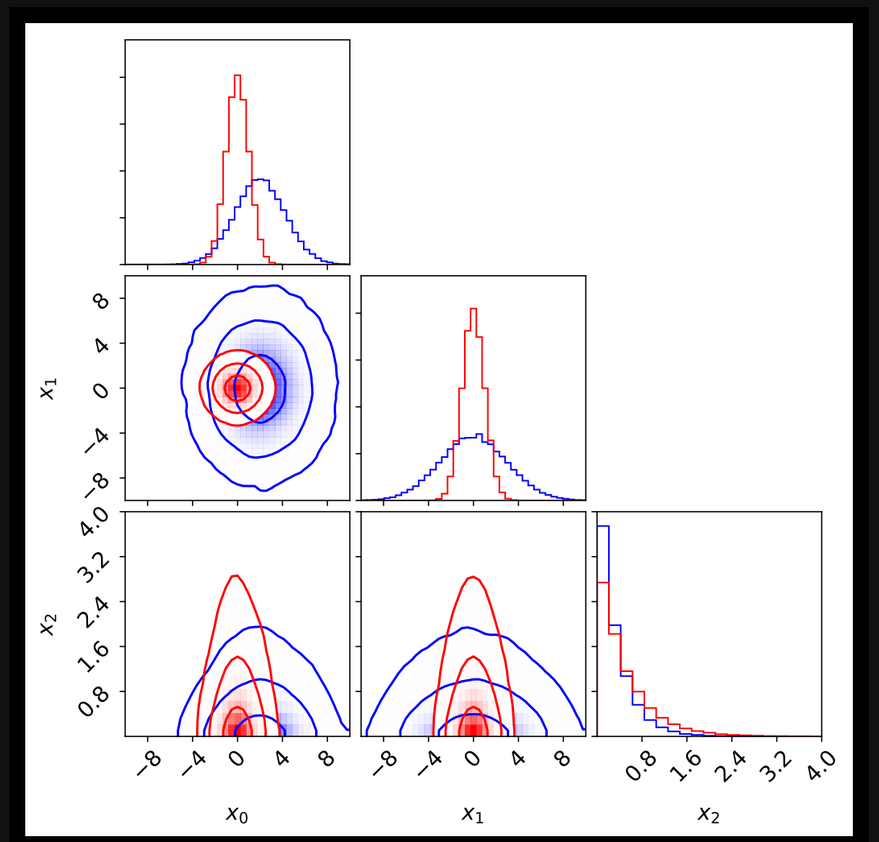
is a **one-dimensional summary statistic** that is **sufficient for inference about μ**

can be approximated very efficiently by training a probabilistic classifier to distinguish signal and background observations

$$L_{\text{BCE}} = -y \log(\hat{y}) - (1-y) \log(1-\hat{y})$$

e.g. neural net minimising cross entropy

An example: 3D Synthetic Problem



$s = 50 \quad b = 1000 \quad \lambda = 3.0 \quad r = 0.0$

A 3D (i.e. $\boldsymbol{x} \in \mathcal{X} \subseteq \mathcal{R}^3$) mixture component model:

$$p(\boldsymbol{x}|\mu, r, \lambda) = \mu f_s(\boldsymbol{x}) + (1-\mu) f_b(\boldsymbol{x}|r, \lambda)$$

where signal and background distributions are

$$f_s(\boldsymbol{x}) = \mathcal{N}\left((x_0, x_1) \middle| \ (1,1) \quad , \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) Exp(x_2|2)$$

$$f_b(\boldsymbol{x}|r, \lambda) = \mathcal{N}\left((x_0, x_1) \middle| (2+r, 0), \begin{bmatrix} 5 & 0 \\ 0 & 9 \end{bmatrix}\right) Exp(x_2|\lambda)$$

alternative parametrisation that will be used

$$p(\boldsymbol{x}|s, r, \lambda, b) = \frac{s}{s+b} f_s(\boldsymbol{x}) + \frac{b}{s+b} f_b(\boldsymbol{x}|r, \lambda)$$

parameter of interest

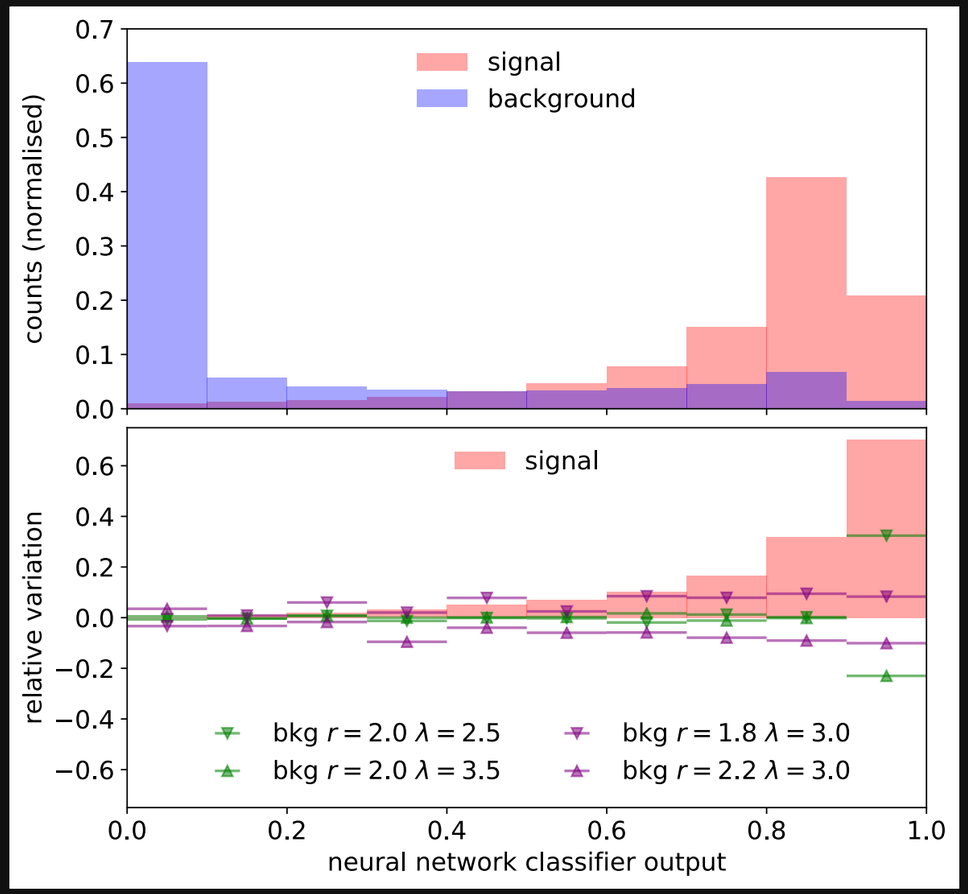
nuisance parameters

Classification as a Surrogate Task

Probabilistic classification has been used for many years in particle colliders to obtain powerful summary statistics by discriminating signal and background simulated observations:

neural net minimising cross entropy

$$L_{\mathrm{BCE}} = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$$

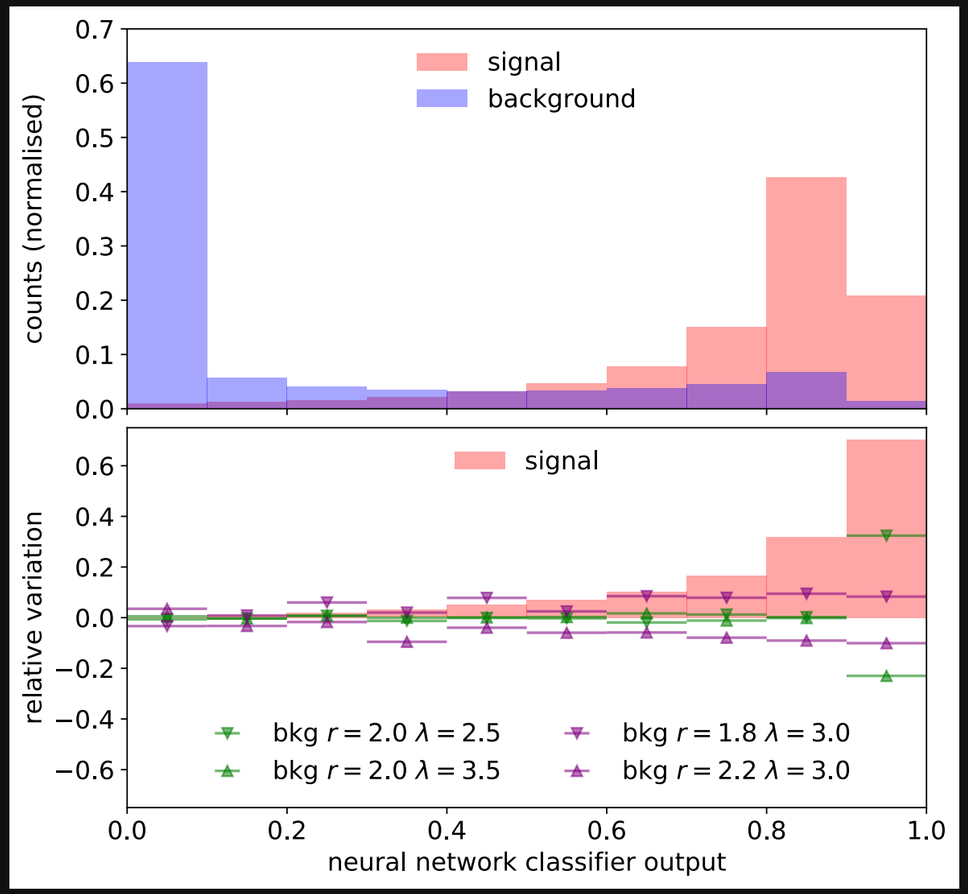


Classification as a Surrogate Task

Probabilistic classification has been used for many years in particle colliders to obtain powerful summary statistics by discriminating signal and background simulated observations:

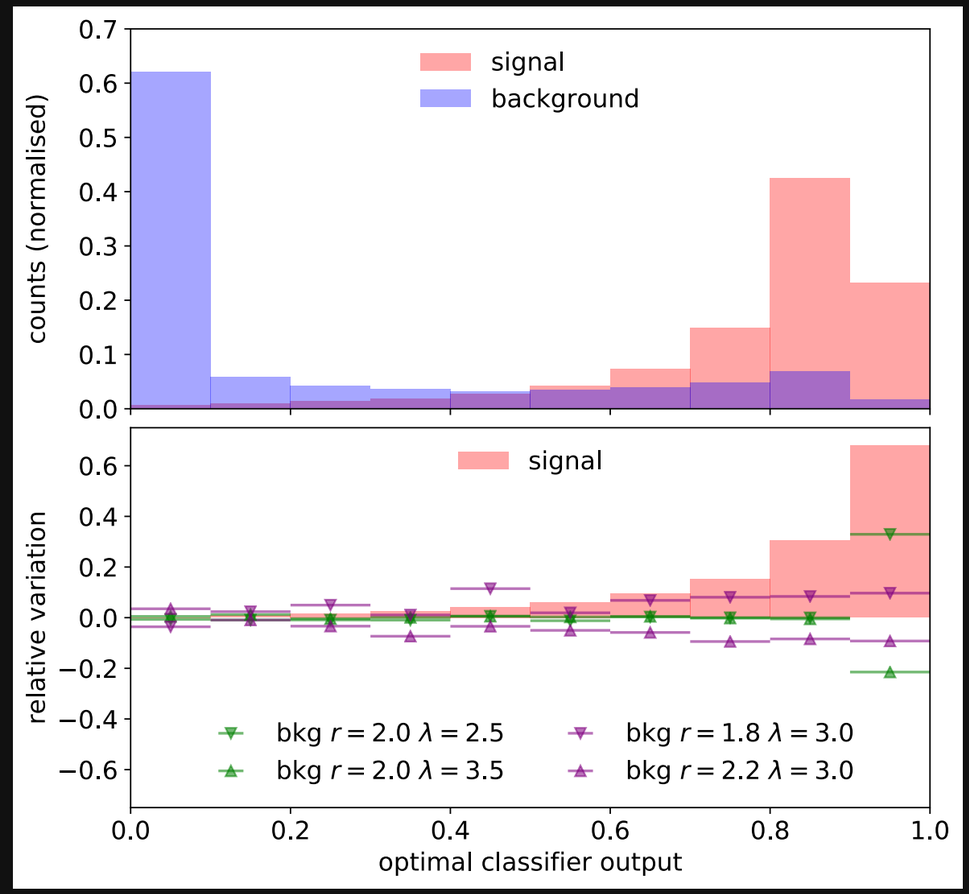
neural net minimising cross entropy

$$L_{\text{BCE}} = -y \log(\hat{y}) - (1-y)\log(1-\hat{y})$$



very good approximation

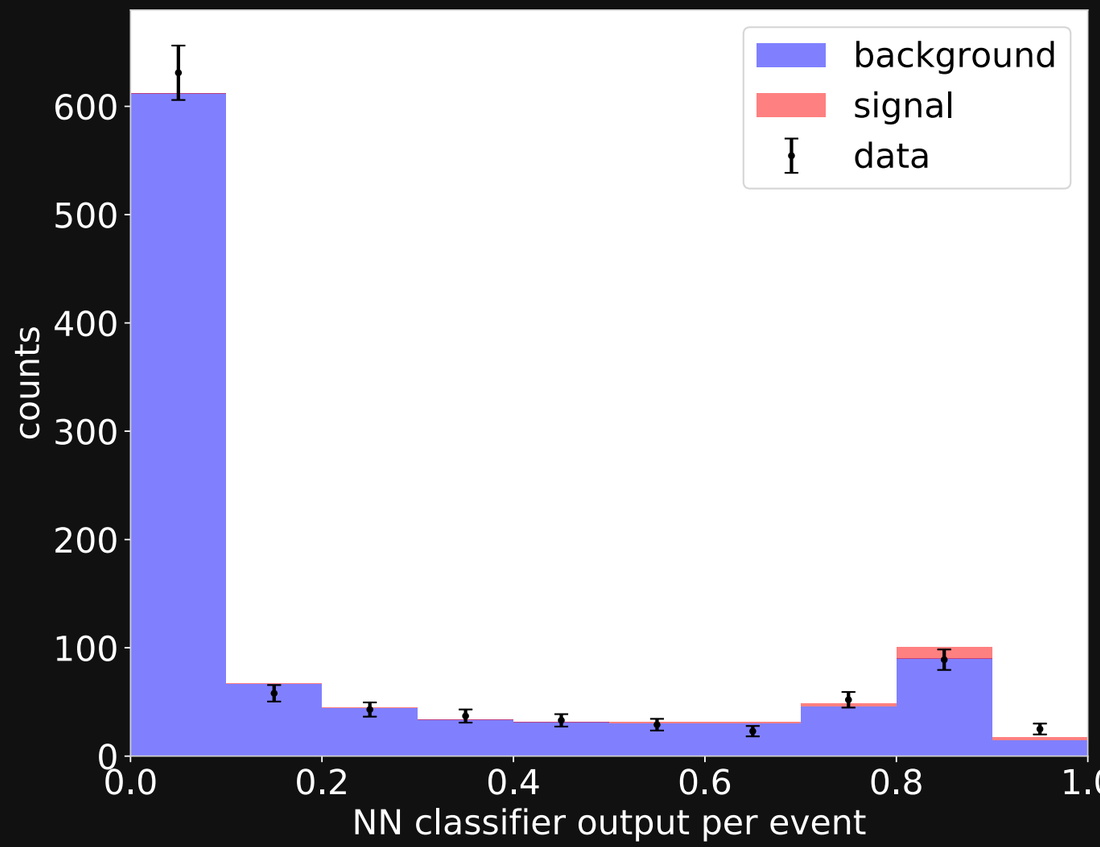
analytical $s_{\text{clf}}(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{f_s(\boldsymbol{x}|\boldsymbol{\theta})}{f_s(\boldsymbol{x}|\boldsymbol{\theta}) + f_b(\boldsymbol{x}|\boldsymbol{\theta})}$



Given enough data and a powerful enough model and learning rule, the probabilisitic classifier output can **approximate** very closely the optimal classifier $s_{\text{clf}}(\boldsymbol{x})$, which is a sufficient statistic for s

Inference based on a low-dim $s(x)$

In most particle collider analyses, statistical inference is based on a single one-dimensional summary statistic $s(x)$ such the output of probabilistic classifier or a clever combination of reconstructed variables based on physics



$$x \longrightarrow s(x)$$

dim $\mathcal{O}(10^8)$ dim $\mathcal{O}(1)$

Inference is then carried out by building a non-parametric likelihood, typically a histogram associated with a Poisson count likelihood:

$$\mathcal{L}(\boldsymbol{\theta}|s(D)) = \prod_{\mathrm{i}\in\mathrm{bins}} \mathrm{Pois}(n_\mathrm{i}|n_\mathrm{i}^s(\boldsymbol{\theta}) + n_\mathrm{i}^b(\boldsymbol{\theta}))$$

which can in turn be used for arbitrary statistical inference given some data, such as obtaining CI on μ or arbitrary hypothesis testing

In the case of classifier-based $s(x)$, this approach should be near optimal when μ (or s) are the only unknown parameters and the histogram binning is small enough

Real world problems → nuisance pars

Simulations are often imperfect and often depend on additional modelling parameters:

$$\boldsymbol{\theta} = (\boldsymbol{\eta}, \boldsymbol{\nu})$$

nuisance parameters (e.g. λ and r in 3D problem)

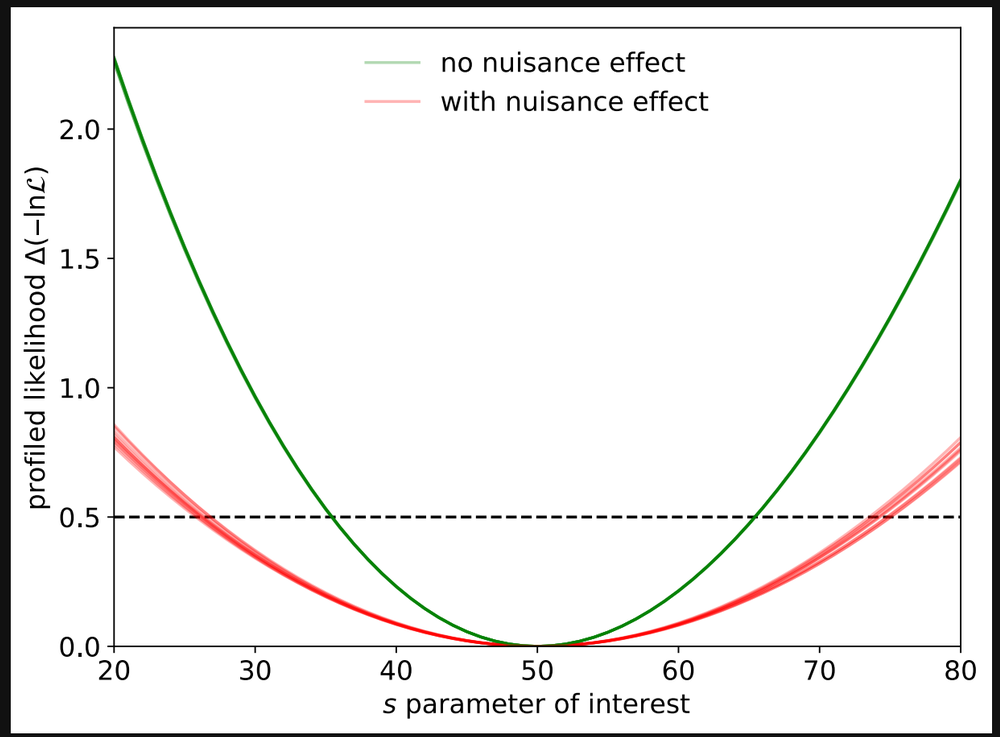
parameters of interest (e.g. s in 3D problem)

Cause a degradation of classifier-based inference, leading to larger uncertainties in the p.o.i.

Profile likelihood of s when:

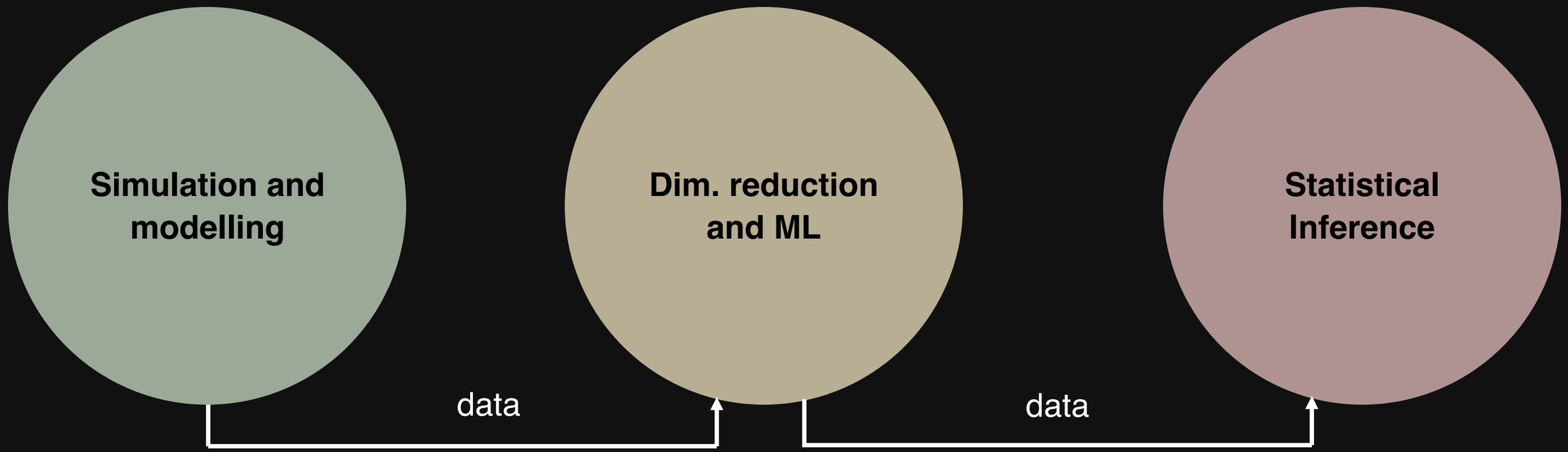
s unknown - λ and r fixed

s unknown - λ and r unknown

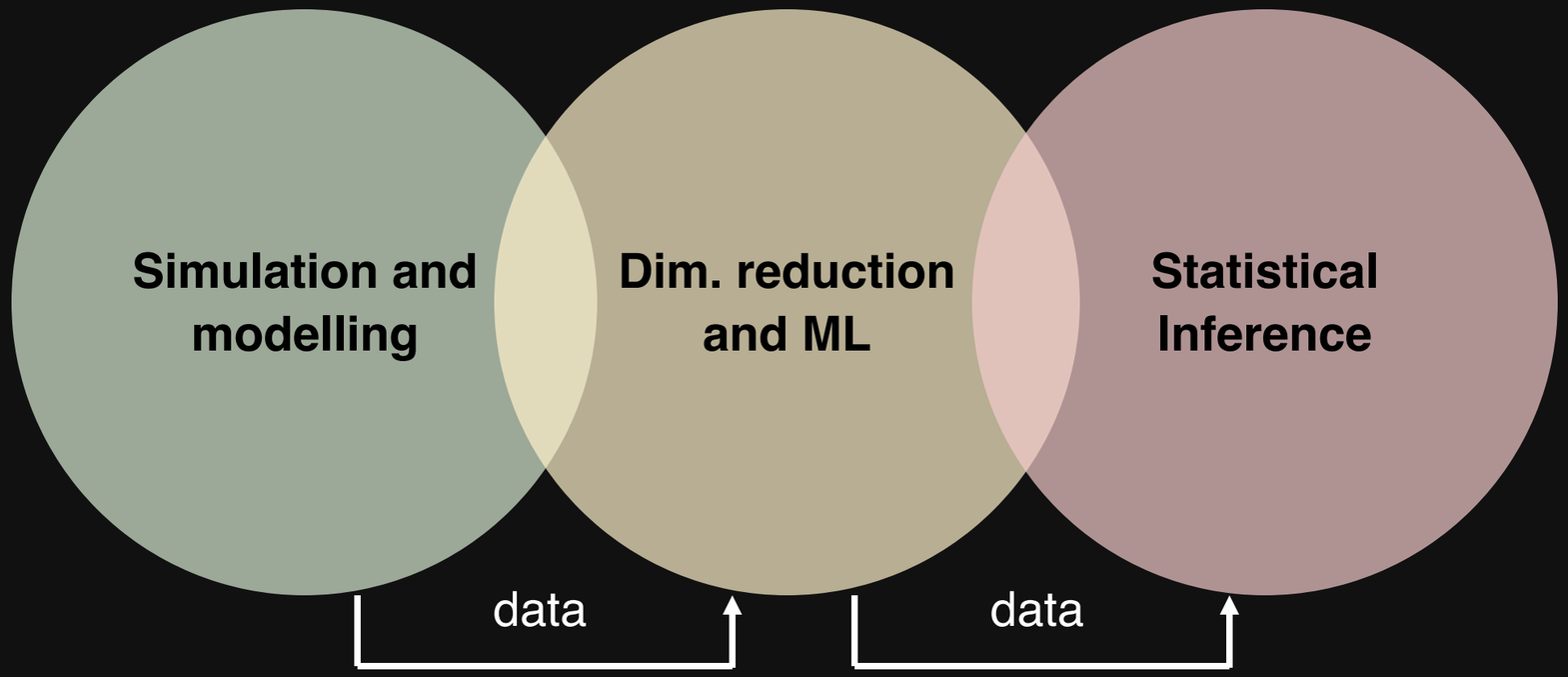


Upper limit of the usefulness of probabilitistic classifiers as $s(\boldsymbol{x})$

A more holistic approach?

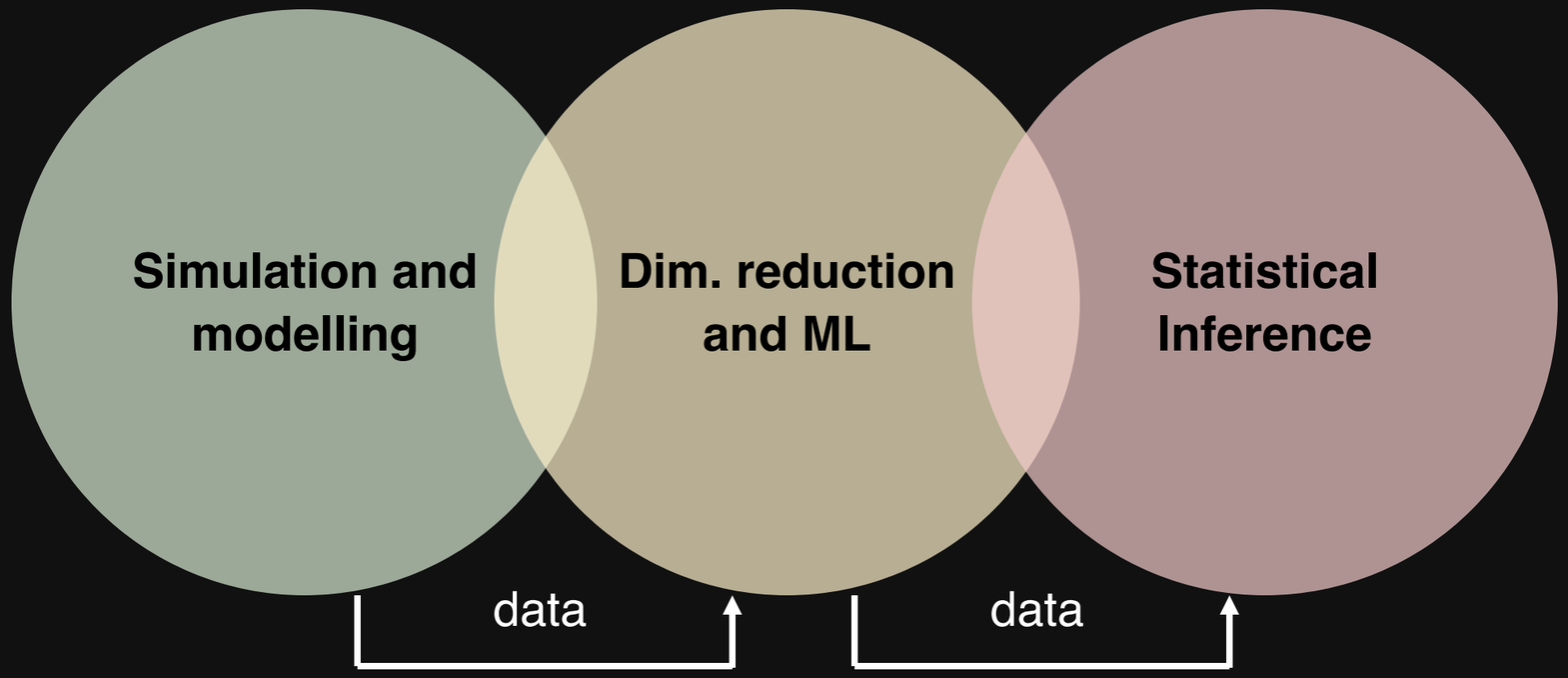


A more holistic approach?



Optimise directly some **approximation of the expected interval width or significance**, accounting for the effect of nuisance parameters, taking advantage of modern auto-diff (e.g. TensorFlow or PyTorch)

A more holistic approach?



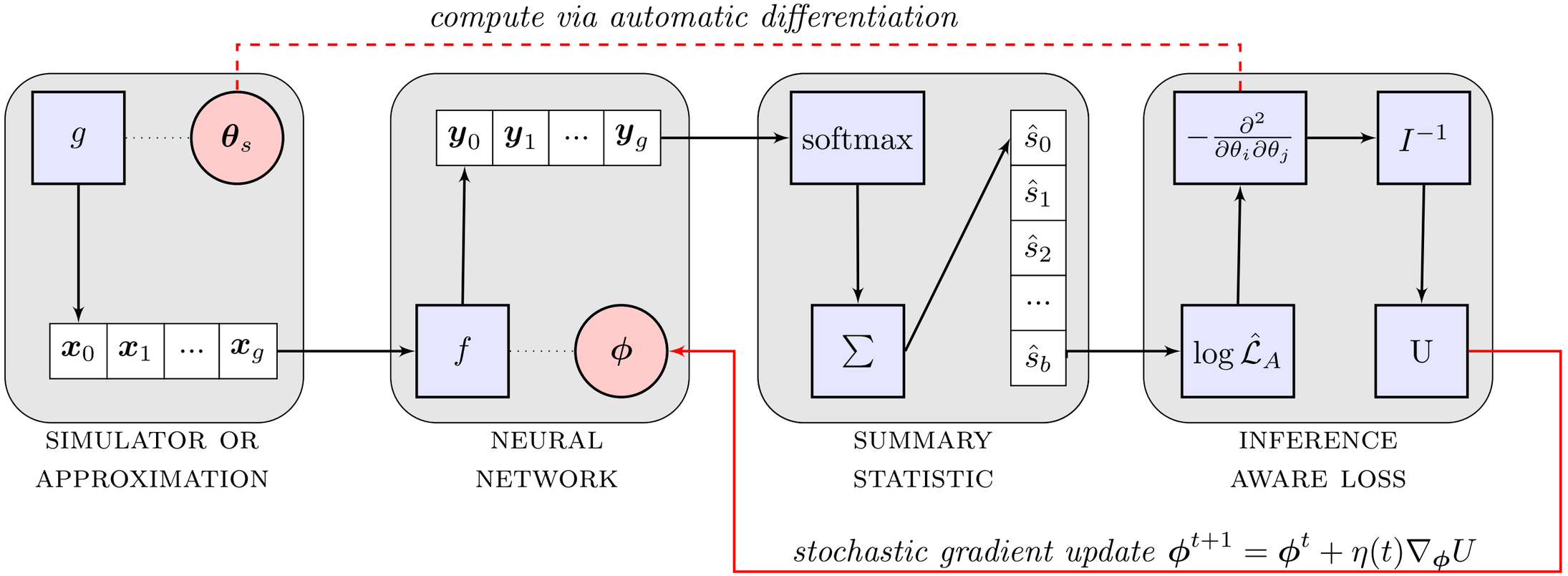
Optimise directly some **approximation of the expected interval width or significance**, accounting for the effect of nuisance parameters, taking advantage of modern auto-diff (e.g. TensorFlow or PyTorch)

Related alternative techniques (not an exhaustive list):

- *Learning to Pivot with Adversarial Neural Networks* [G. Louppe et al arxiv:1611.01046]
- *Mining gold from implicit models to improve likelihood-free inference* [J. Brehmer et al arxiv:1805.12244]
- *Nuisance hardened data compression for fast likelihood-free inference* [J. Alsing et al arxiv:1903.01473]
- *Automatic physical inference with information maximising neural networks* [T. Charnock et al arxiv:1802.03537]

INFERence-Aware Neural Optimisation

An approach to learn non-linear summary statistics $\boldsymbol{s}(D)$ by **directly minimising an approximation the expected profiled (or marginalised) interval width** accounting for the effect of nuisance parameters



check arxiv.org/abs/1806.04743 for a more detailed description

Variational Effect of Parameters $\boldsymbol{\theta} = (\boldsymbol{\eta}, \boldsymbol{\nu})$

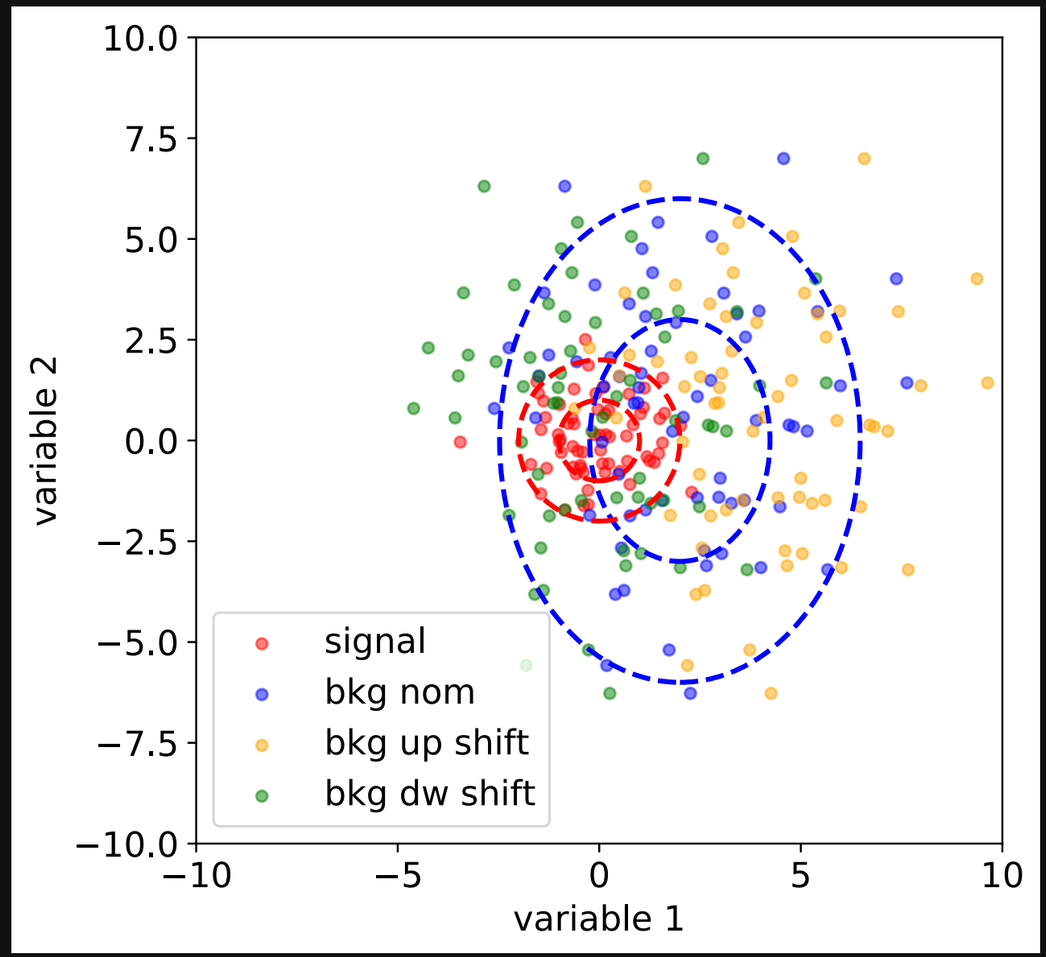
Differentiable approximation of the effect of parameters $\boldsymbol{\theta} = (\boldsymbol{\eta}, \boldsymbol{\nu})$ over a given simulated observation $(\boldsymbol{x}_i, \boldsymbol{z}_i, w_i)$

$$g(\boldsymbol{\theta}_s)$$
$$(\boldsymbol{x}_i, \boldsymbol{z}_i, w_i) \longrightarrow (\boldsymbol{x}'_i, \boldsymbol{z}'_i, w'_i)$$

a non-linear function that depends on the particularities of the problem

Implemented in autodiff framework (e.g. TensorFlow) to easily obtain derivatives

Alternatively could use a generative surrogate conditioned on the parameters $\boldsymbol{\theta}$

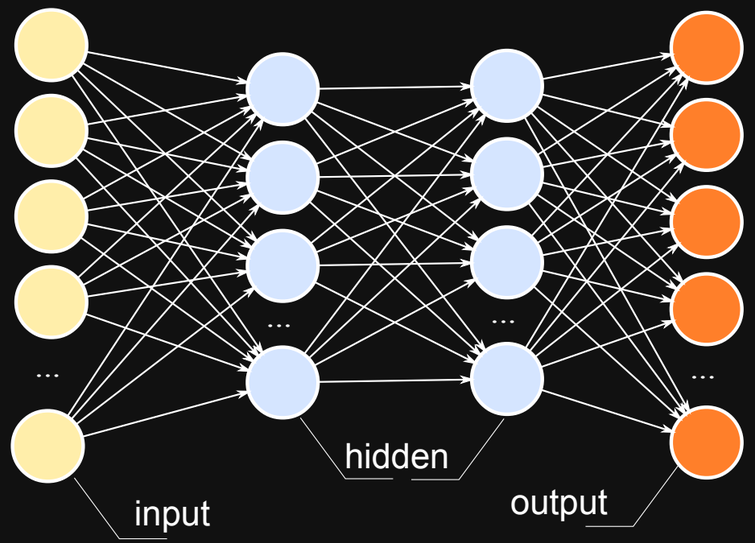


Illustrative example based on 3D example

$$(x'_0, x'_1) = (x_0, x_1) + r(1, 0)$$

Neural net: a learnable non-linear $s(x)$

$g(\boldsymbol{x}|\boldsymbol{\theta})$



loss L

$\dim(\mathcal{S})$ → **number of neural net outputs**

$$\boldsymbol{s}(\boldsymbol{x}') = \boldsymbol{f}(\boldsymbol{x}'|\boldsymbol{\phi})$$

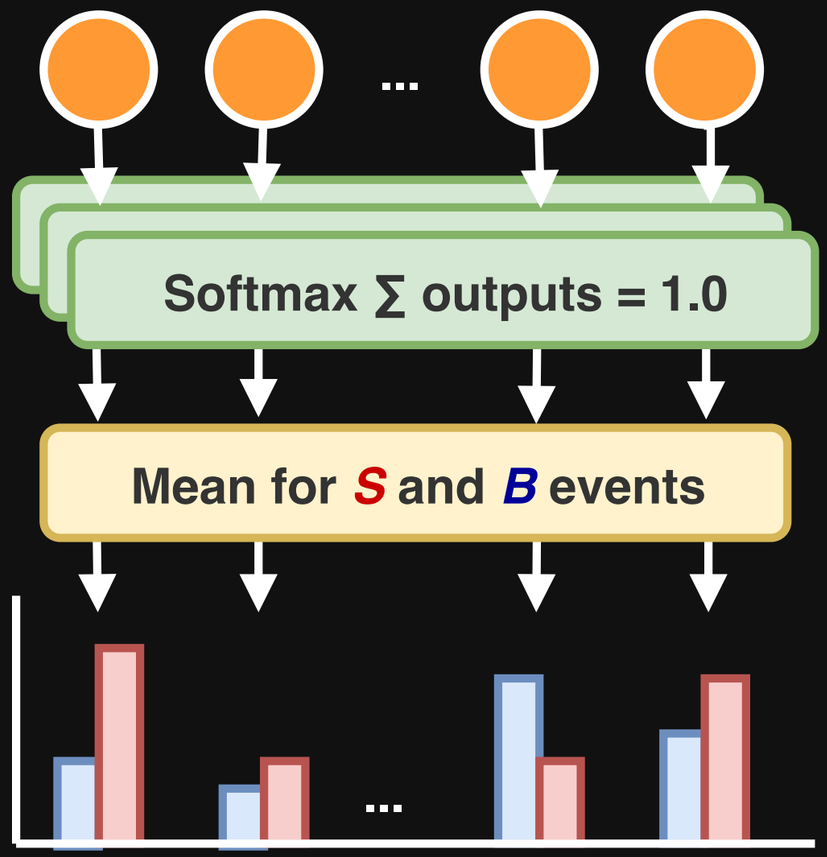
neural net parameters are trained during optimisation process

Could re-use the same techniques and architectures as for standard supervised deep learning

A two-hidden layer MLP (100 units each, ReLU activation, He normal init) used for synthetic examples in this work

Building a non-parametric likelihood

$$\boldsymbol{x}'_i \longrightarrow s(\boldsymbol{x}'_i) \longrightarrow s(\boldsymbol{D}) \longrightarrow \mathcal{L}(\theta, \eta; \phi)$$



current choice → approximate a histogram-like summary statistic $\boldsymbol{s}(D)$

applying softmax to each observation output from the neural net and summing over each dataset

$$\mathcal{L}(\theta; s(G_s|\phi)) = \prod_{\mathrm{i}\in\mathrm{bins}} \mathrm{Pois}(n_\mathrm{i}|n_\mathrm{i}^s + n_\mathrm{i}^b)$$

The likelihood depends both on the neural network parameters ϕ and the statistical model parameters (η, ν)

Alternative approaches to build $\mathcal{L}(\theta, \eta; \phi)$ are possible (e.g. non-parametric likelihood of $\boldsymbol{s}(\boldsymbol{x})$ using kernel density estimation)

Inference-motivated Loss Function

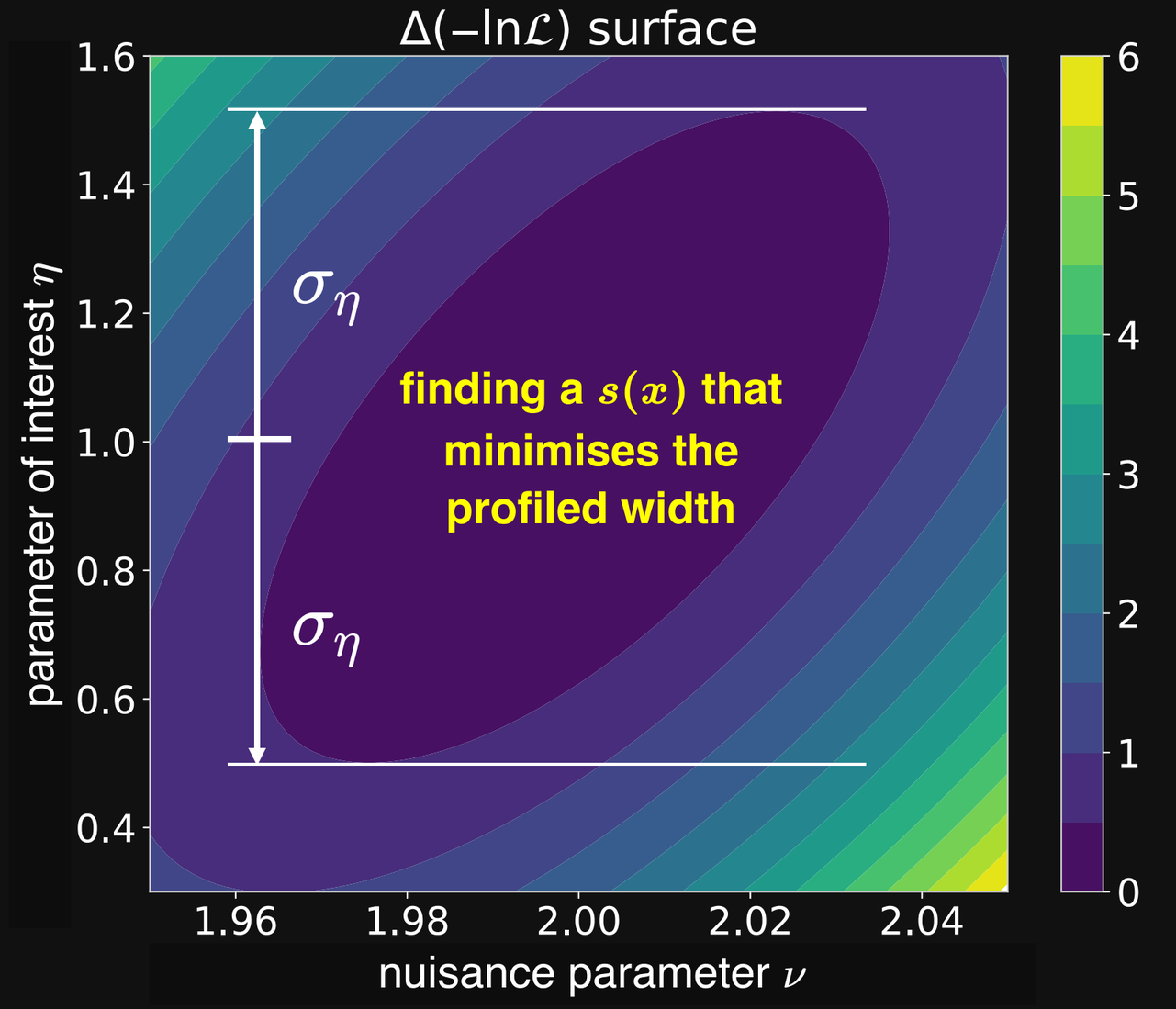
If we expand negative log-likelihood around known minimum (i.e. assuming data equals expectation $n_i = n_i^s + n_i^b$) :

$$\text{covariance} \approx \mathbf{H}^{-1}(-\ln \mathcal{L})$$

can use as loss function directly the approximate variance estimator on the pars of interest:

$$\text{loss} \approx \text{Var}(\eta) \quad (\text{expected})$$

that accounts for the effect of unknowns nuisance parameters. Could be Fisher sub-determinant for multiple p.o.i.



Equivalent to Laplace approx. in Bayesian Inference (i.e. loss ~ marginalised posterior width)

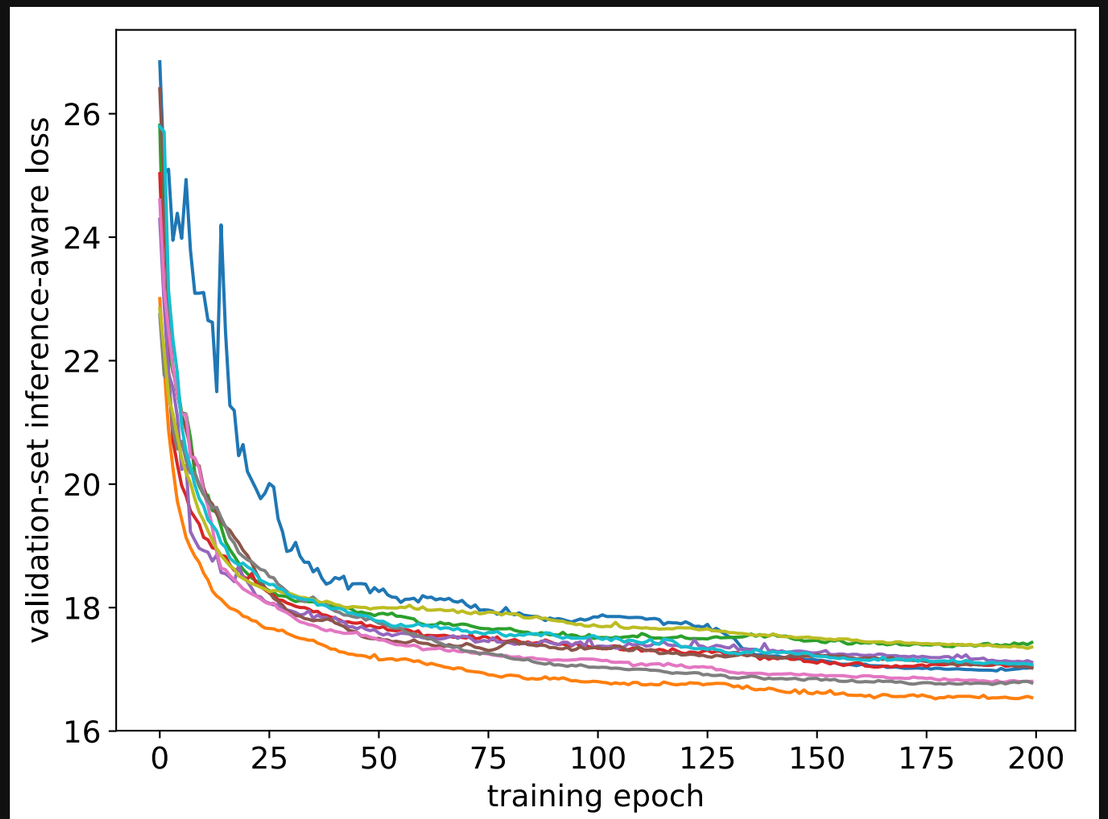
Synthetic Inference Benchmarks

Several inference problems about $s = \mu b/(1-\mu)$, with different parameter constraints, are considered based on the 3D problem mentioned before

	Benchmark 0	Benchmark 1	Benchmark 2	Benchmark 3	Benchmark 4
interest pars	1 (s)	1 (s)	1 (s)	1 (s)	1 (s)
nuisance pars	0 (all fixed)	1 (r)	2 (r and λ)	2 (r and λ)	3 (r, λ and b)
r (bkg shift)	0.0 (fixed)	free (init 0.0)	free (init 0.0)	$\mathcal{N}(\lambda\|0.0, 0.4)$	$\mathcal{N}(\lambda\|0.0, 4.0)$
λ (bkg exp rate)	3.0 (fixed)	3.0 (fixed)	free (init 3.0)	$\mathcal{N}(\lambda\|3.0, 1.0)$	$\mathcal{N}(\lambda\|3.0, 1.0)$
b (bkg normalisation)	1000 (fixed)	1000 (fixed)	1000 (fixed)	1000 (fixed)	$\mathcal{N}(b\|1000, 100)$

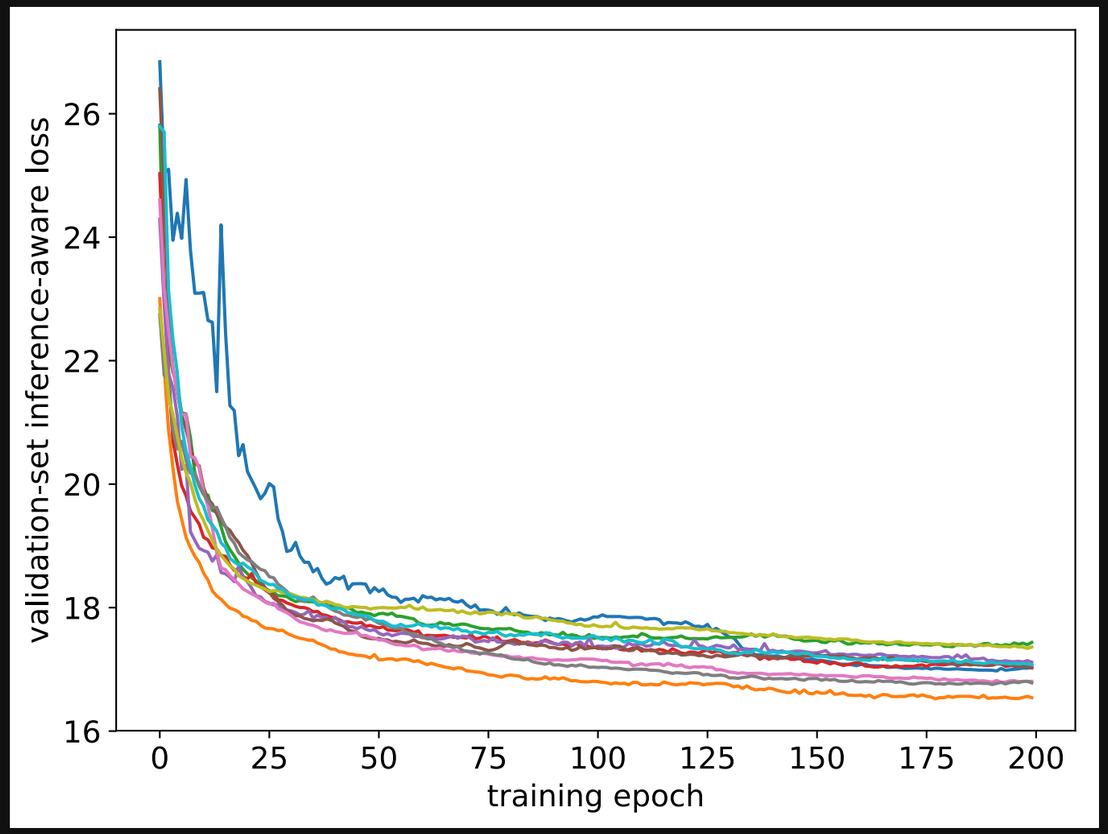
Information about the inference problem at hand can be used within INFERNO but not with standard probabilistic classifiers

Technique works! (results for Benchmark 2)

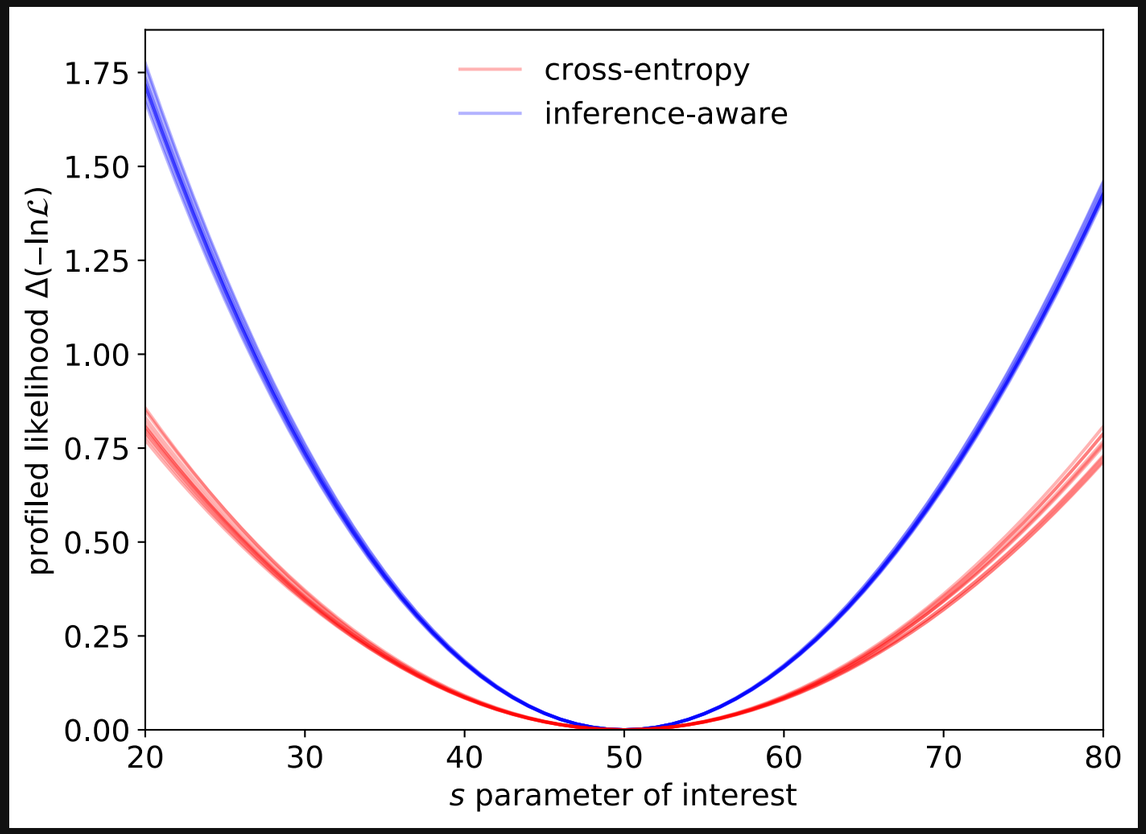


During training, INFERNO consistently converges to low-variance summary statistics

Technique works! (results for Benchmark 2)



During training, INFERNO consistently converges to low-variance summary statistics



Clearly outperforms classifiers when nuisance parameters are relevant

Comparison w/ Classification Approach

A more systematic comparison, shows that INFERNO clearly outperforms any classifier (even optimal Bayes) when nuisance parameters are relevant

(less is better)

Table 1: Expected uncertainty on the parameter of interest s for each of the inference benchmarks considered using a cross-entropy trained neural network model, INFERNO customised for each problem and the optimal classifier and likelihood based resuls.

	Benchmark 0	Benchmark 1	Benchmark 2	Benchmark 3	Benchmark 4
NN classifier	$14.99^{+0.02}_{-0.00}$	$18.94^{+0.11}_{-0.05}$	$23.94^{+0.52}_{-0.17}$	$21.54^{+0.27}_{-0.05}$	$26.71^{+0.56}_{-0.11}$
INFERNO 0	$\mathbf{15.51^{+0.09}_{-0.02}}$	$18.34^{+5.17}_{-0.51}$	$23.24^{+6.54}_{-1.22}$	$21.38^{+3.15}_{-0.69}$	$26.38^{+7.63}_{-1.36}$
INFERNO 1	$15.80^{+0.14}_{-0.04}$	$\mathbf{16.79^{+0.17}_{-0.05}}$	$21.41^{+2.00}_{-0.53}$	$20.29^{+1.20}_{-0.39}$	$24.26^{+2.35}_{-0.71}$
INFERNO 2	$15.71^{+0.15}_{-0.04}$	$16.87^{+0.19}_{-0.06}$	$\mathbf{16.95^{+0.18}_{-0.04}}$	$16.88^{+0.17}_{-0.03}$	$18.67^{+0.25}_{-0.05}$
INFERNO 3	$15.70^{+0.21}_{-0.04}$	$16.91^{+0.20}_{-0.05}$	$16.97^{+0.21}_{-0.04}$	$\mathbf{16.89^{+0.18}_{-0.03}}$	$18.69^{+0.27}_{-0.04}$
INFERNO 4	$15.71^{+0.32}_{-0.06}$	$16.89^{+0.30}_{-0.07}$	$16.95^{+0.38}_{-0.05}$	$16.88^{+0.40}_{-0.05}$	$\mathbf{18.68^{+0.58}_{-0.07}}$
Optimal classifier	14.97	19.12	24.93	22.13	27.98
Analytical likelihood	14.71	15.52	15.65	15.62	16.89

$s_{\text{clf}}(x)$

upper bound if we knew $p(x|\theta)$

classifier is expected to be near optimal **sufficient** $s(x)$

INFERNO adapted for each inference problem is considerable better than classifiers

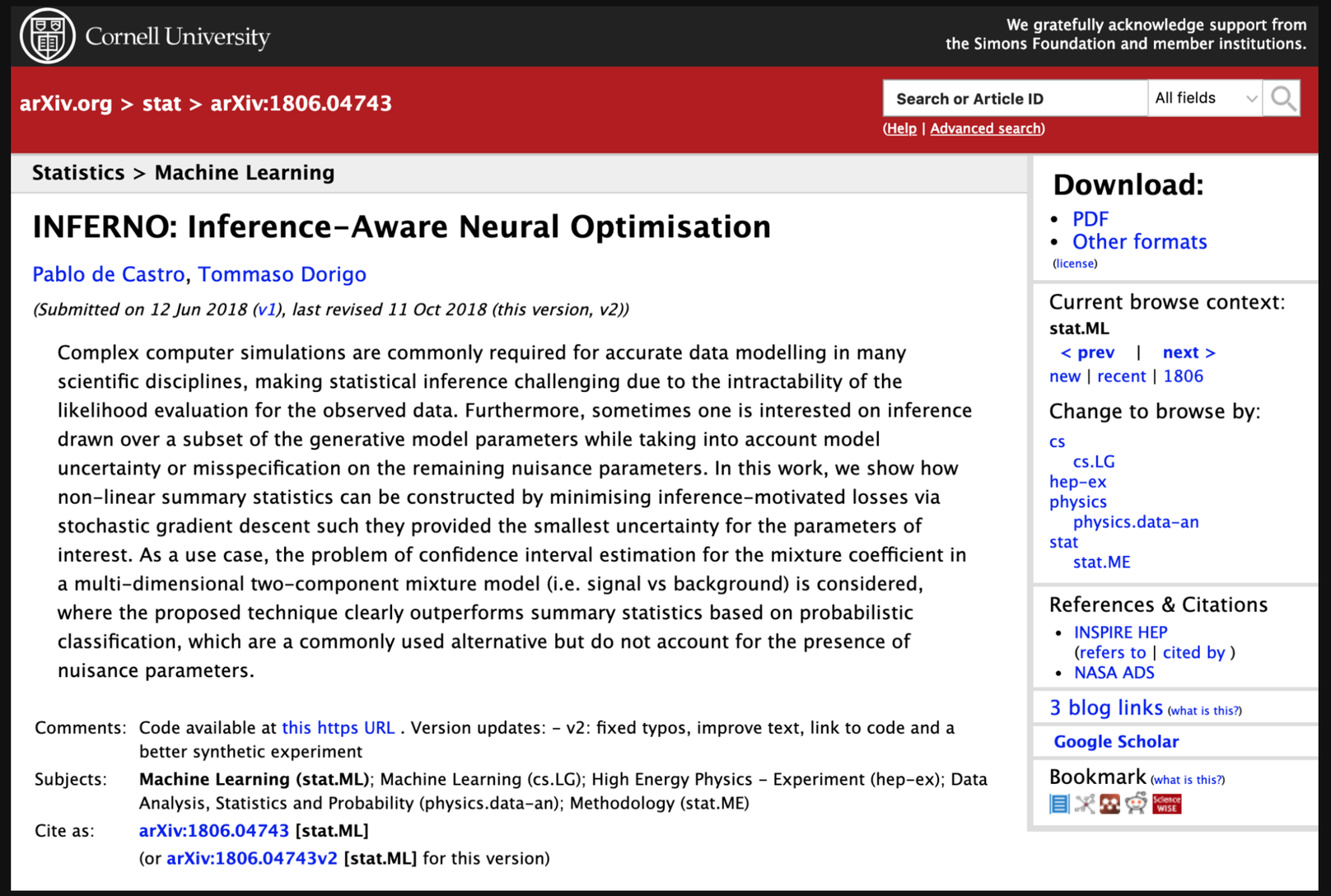
Final Remarks

Alternative ways to construct summary statistics in cases where nuisance parameters are important could greatly increase the discovery reach of scientific experiments based on simulation-based inference

The proposed INFERNO technique obtains non-linear summary statistics by minimising the expected uncertainty accounting for the effect of nuisance parameters

Early results are really promising, yet benchmarking in more complex problems and comparisons with alternative techniques are needed to shed more light on its real-world usefulness

More details in our ArXiv preprint



Cornell University

We gratefully acknowledge support from the Simons Foundation and member institutions.

arXiv.org > stat > arXiv:1806.04743

Search or Article ID | All fields

(Help | Advanced search)

Statistics > Machine Learning

INFERNO: Inference-Aware Neural Optimisation

Pablo de Castro, Tommaso Dorigo

(Submitted on 12 Jun 2018 (v1), last revised 11 Oct 2018 (this version, v2))

Complex computer simulations are commonly required for accurate data modelling in many scientific disciplines, making statistical inference challenging due to the intractability of the likelihood evaluation for the observed data. Furthermore, sometimes one is interested on inference drawn over a subset of the generative model parameters while taking into account model uncertainty or misspecification on the remaining nuisance parameters. In this work, we show how non-linear summary statistics can be constructed by minimising inference-motivated losses via stochastic gradient descent such they provided the smallest uncertainty for the parameters of interest. As a use case, the problem of confidence interval estimation for the mixture coefficient in a multi-dimensional two-component mixture model (i.e. signal vs background) is considered, where the proposed technique clearly outperforms summary statistics based on probabilistic classification, which are a commonly used alternative but do not account for the presence of nuisance parameters.

Comments: Code available at this https URL . Version updates: - v2: fixed typos, improve text, link to code and a better synthetic experiment

Subjects: **Machine Learning (stat.ML)**; Machine Learning (cs.LG); High Energy Physics - Experiment (hep-ex); Data Analysis, Statistics and Probability (physics.data-an); Methodology (stat.ME)

Cite as: **arXiv:1806.04743 [stat.ML]**

(or **arXiv:1806.04743v2 [stat.ML]** for this version)

Download:

- PDF
- Other formats

(license)

Current browse context: **stat.ML**

< prev | next >

new | recent | 1806

Change to browse by:

cs
cs.LG
hep-ex
physics
physics.data-an
stat
stat.ME

References & Citations

- INSPIRE HEP (refers to | cited by)
- NASA ADS

3 blog links (what is this?)

Google Scholar

Bookmark (what is this?)

many more details on our stat.ML preprint arxiv.org/abs/1806.04743
feedback, comments, discussions are welcomed via

Twitter DM @pablodecm, pablo.decastro@cern.ch or workshop Gitter