INFERNO: INFERENCE-AWARE NEURAL OPTIMISATION

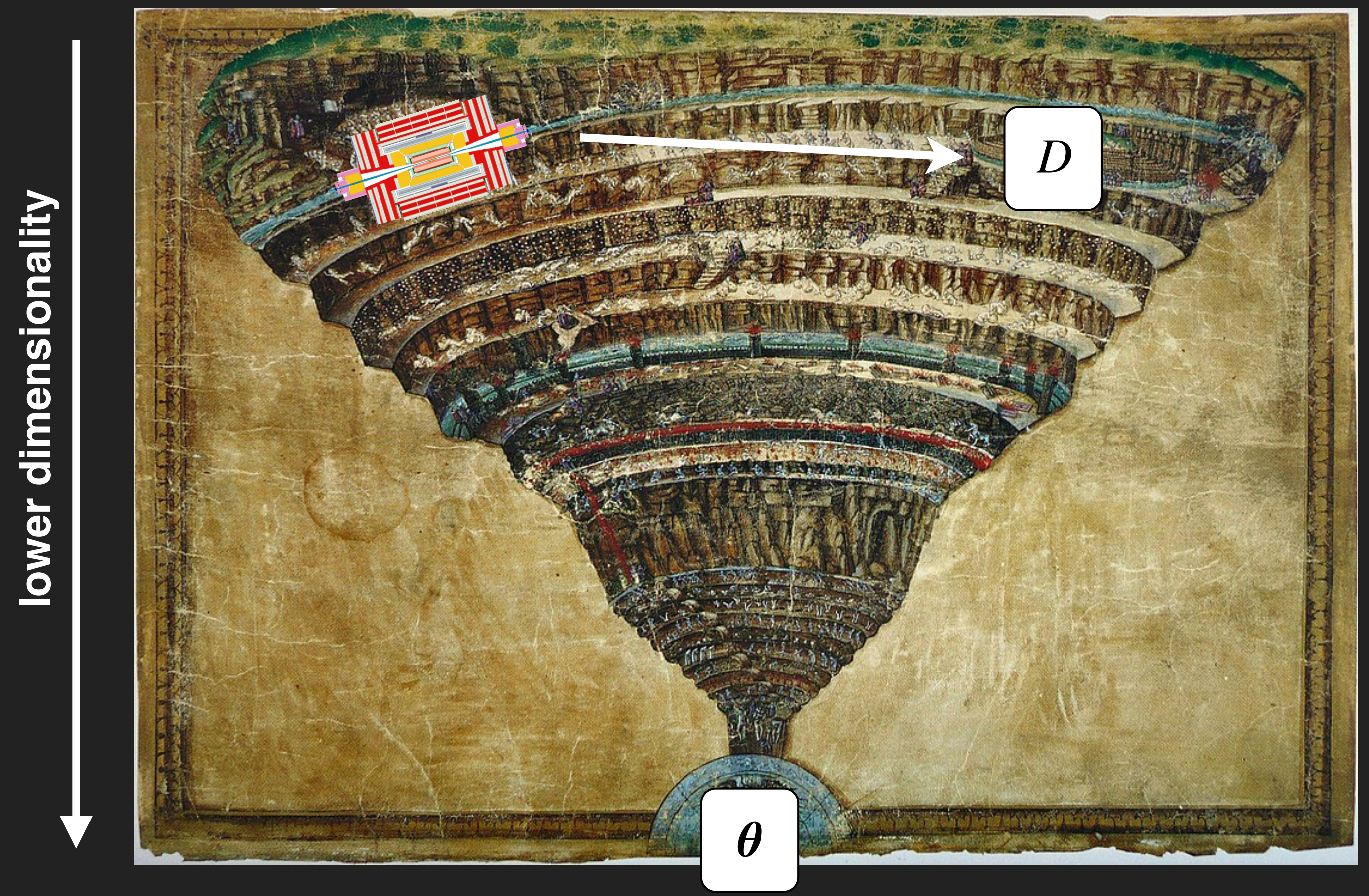
Pablo de Castro (@pablodecm) and Tommaso Dorigo (@dorigo)

13th March 2018 @ ACAT 2019 (Saas Fee, Switzerland)

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

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

STATISTICAL INFERENCE IN PARTICLE COLLIDER EXPERIMENTS

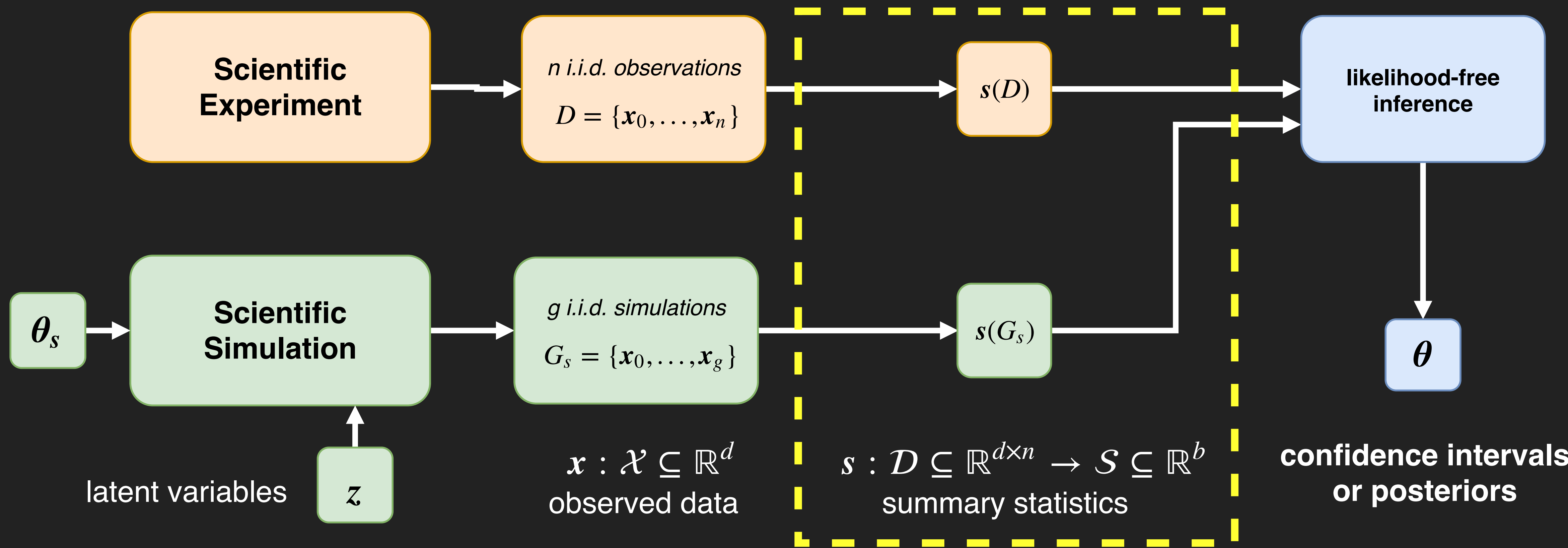


Main challenges:

- High dimensionality, both # observation and # dimensions of each observation
- Link between model parameters $\boldsymbol{\theta}$ and data only implicitly defined via forward simulation

SIMULATION-BASED STATISTICAL INFERENCE

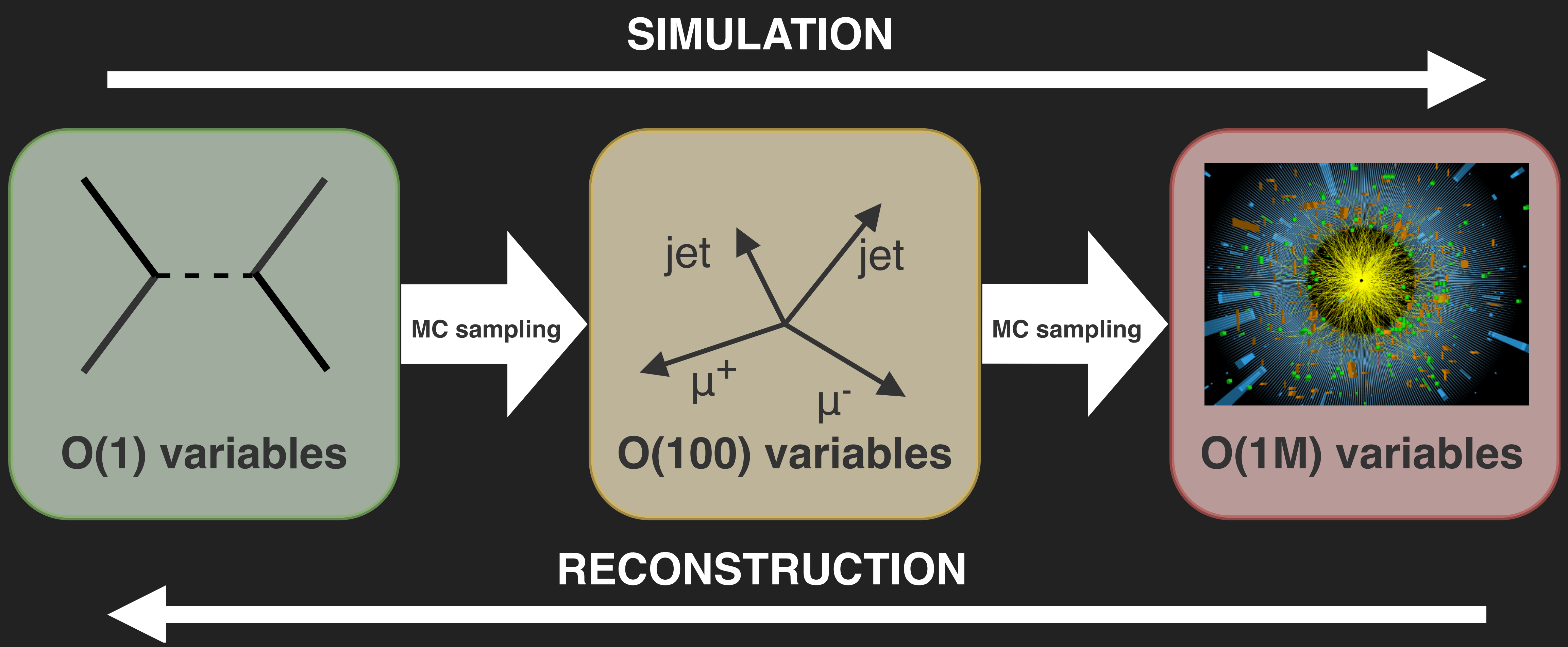
Realistic experimental modelling can only be carried out through forward sampling via complex computer programs in many scientific disciplines



Thus $p(\mathbf{x}|\boldsymbol{\theta})$ is not tractable and we resort to **likelihood-free inference** techniques or **non-parametric likelihoods**, often requiring using low-dim summaries $s(D)$

$\mathbf{p(x|model)}$ IS NOT KNOWN AT THE LHC EXPERIMENTS

Simulated data samples can be obtained via complex physics-based Monte Carlo programs but $p(\mathbf{x}|\boldsymbol{\theta})$ cannot be directly evaluated



Good approximations of $p(\mathbf{x}|\boldsymbol{\theta})$ are effectively unachievable due to curse of dimensionality. So a dim. reduction $\mathrm{R}^n \rightarrow \mathrm{R}^{O(1)}$ step is used to build a summary statistic $s(D)$ keeping as much information for inference as possible.

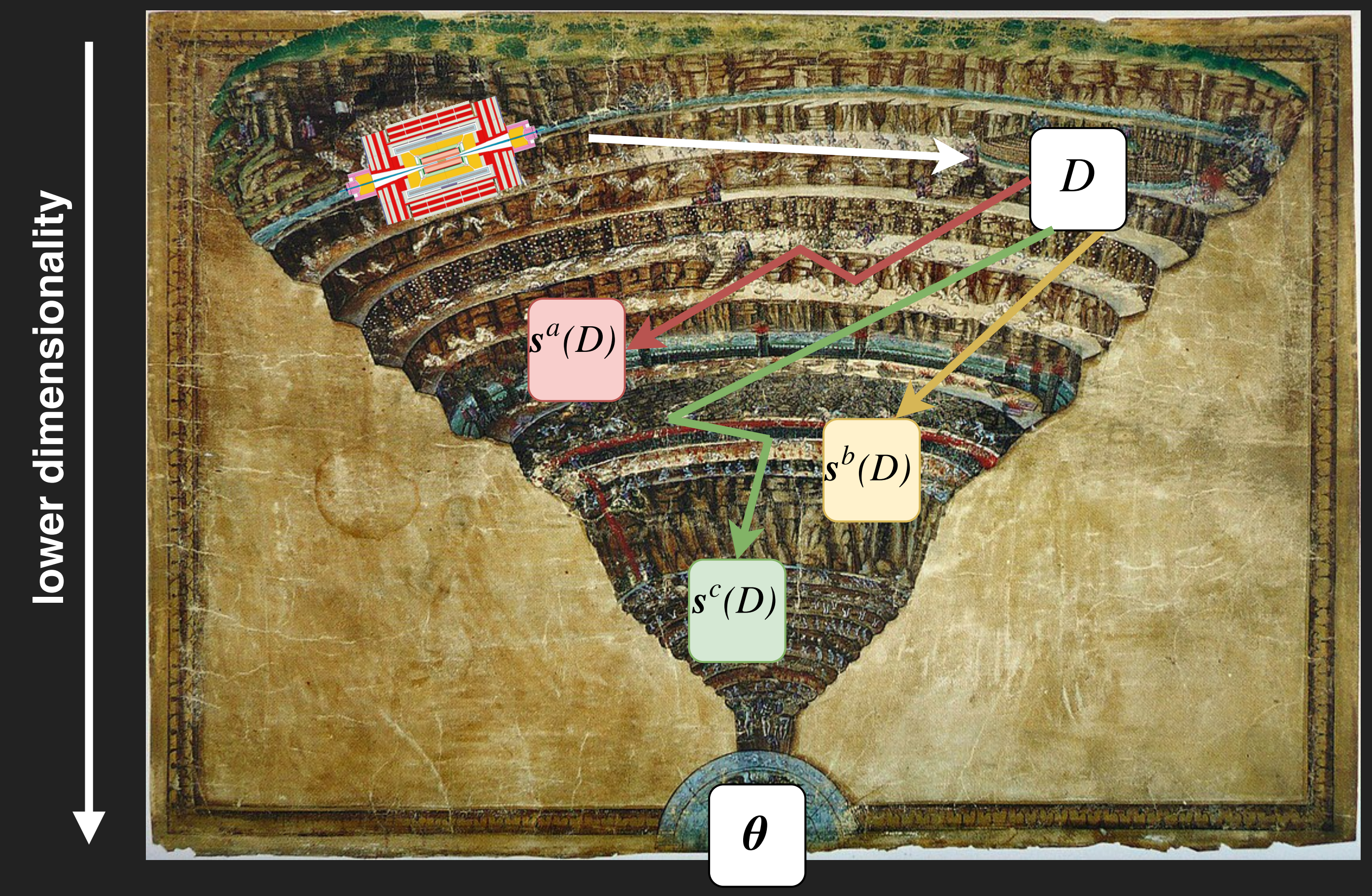
A QUEST FOR POWERFUL SUMMARY STATISTICS

How to choose a good summary statistic $\boldsymbol{s}(D)$ for statistical inference of the parameters of interest $\boldsymbol{\theta}$?

Ideally we want a **sufficient summary statistic**, defined as:

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

Such statistic is problem specific and might not exist or could not be obtained because $p(D|\boldsymbol{\theta})$ is not known



WHY SIGNAL VS BACKGROUND CLASSIFICATION IS OFTEN USED?

IF

A two-component mixture problem, e.g. signal $f_s(\boldsymbol{x}|\boldsymbol{\theta})$ and background $f_b(\boldsymbol{x}|\boldsymbol{\theta})$, which is common in many disciplines:

$$p(x|\mu,\boldsymbol{\theta}) = \mu\, f_s(\boldsymbol{x}|\boldsymbol{\theta}) + (1-\mu)\, f_b(\boldsymbol{x}|\boldsymbol{\theta})$$
$$\text{2-component mixture}$$

and the mixture coefficient μ **is the only unknown parameter**, i.e. all other parameters $\boldsymbol{\theta}$ are known and fixed

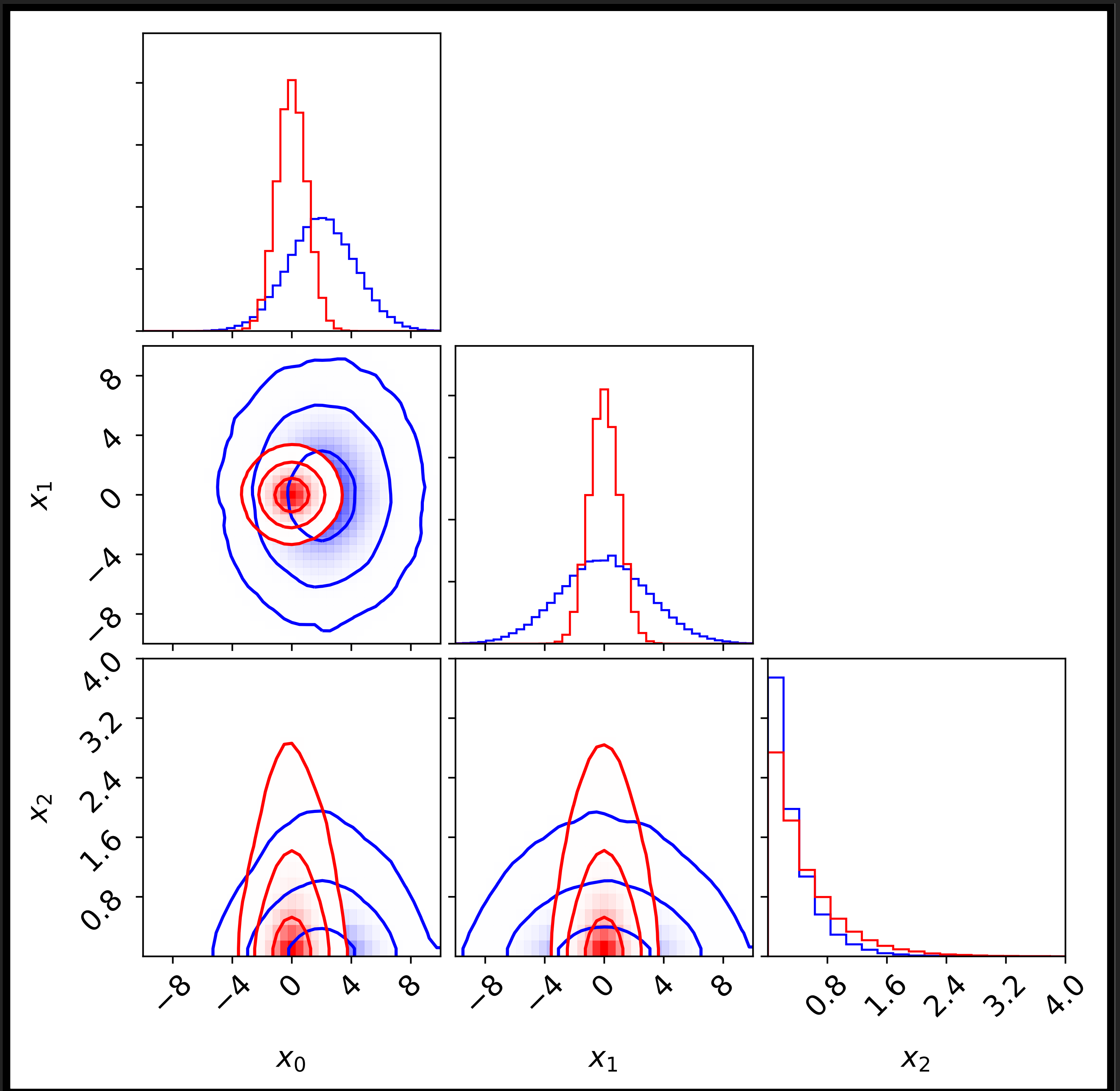
THEN

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

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

Can be approximated from simulated $\mathbf{s}$ and $\mathbf{b}$ samples using probabilistic classification models (e.g. neural network minimizing cross entropy)

A PRACTICAL EXAMPLE: 3D SYNTHETIC MIXTURE



A two-component mixture model:

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

where each component is:

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

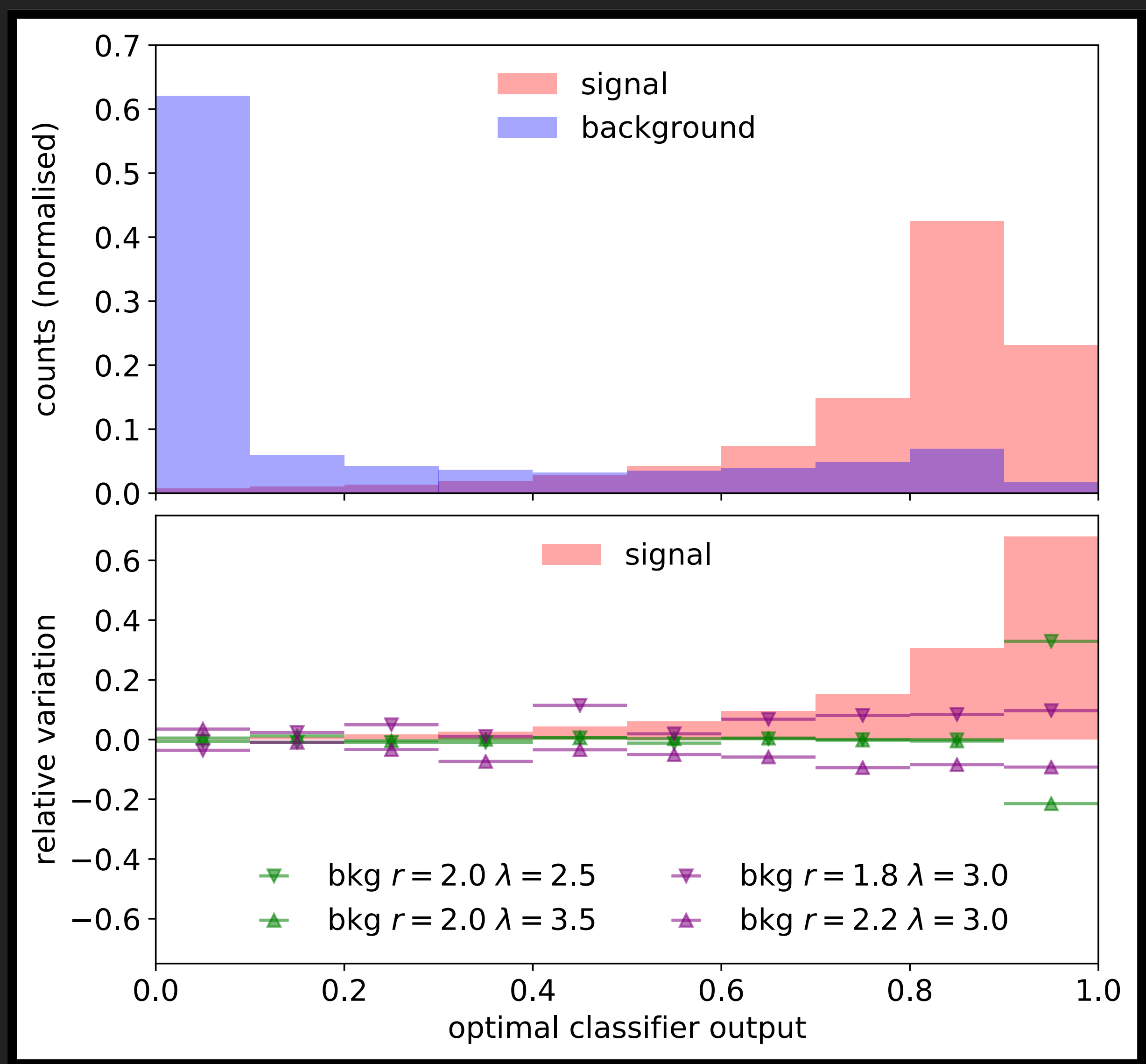
$$f_s(\boldsymbol{x}) = \mathcal{N}\left((x_0, x_1) \middle| (1, 1), \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) Exp(x_2|2)$$

can also be parametrised:

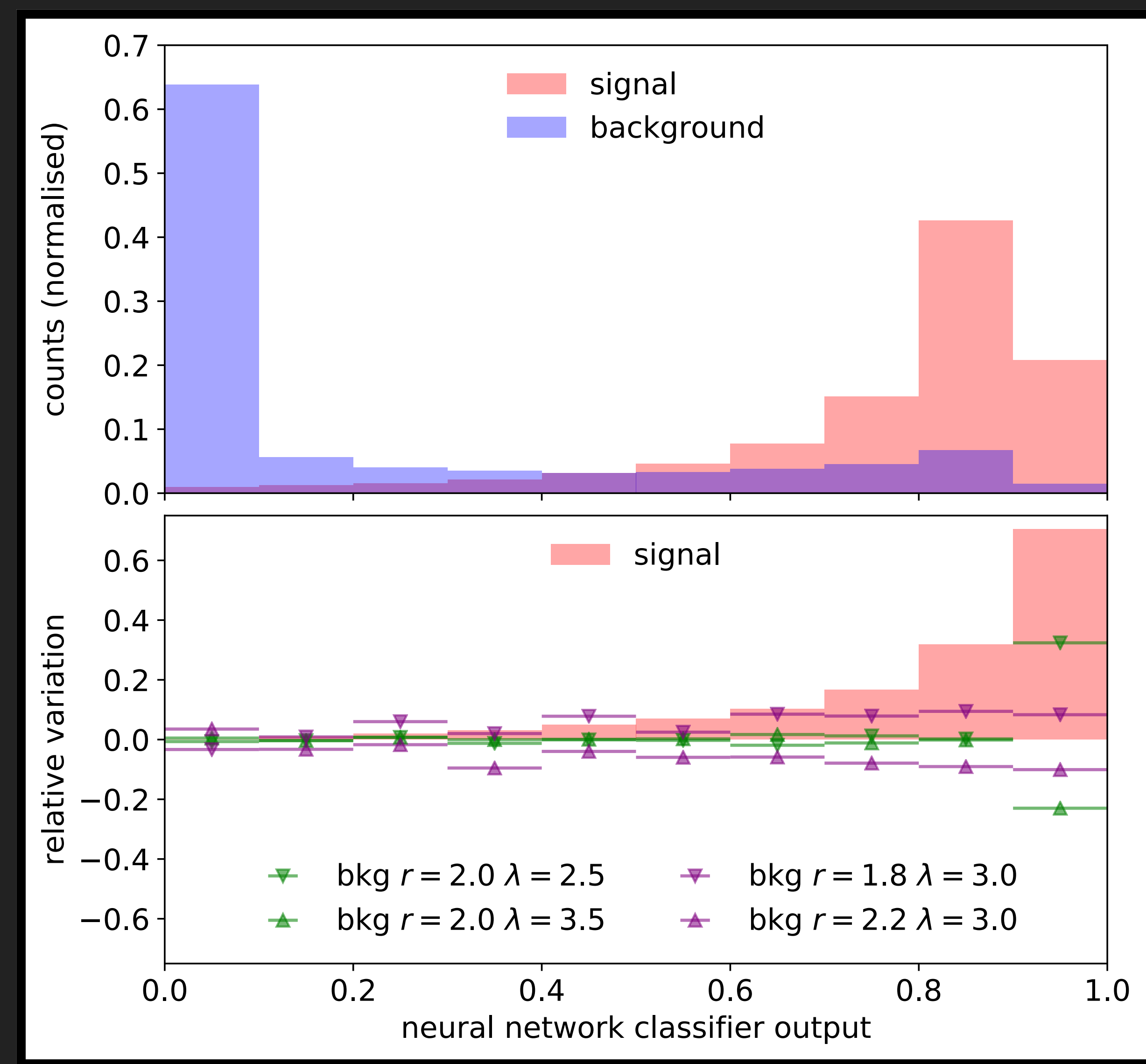
$$p(\boldsymbol{x}|s, r, \lambda, b) = \frac{b}{s+b} f_b(\boldsymbol{x}|r,\lambda) + \frac{s}{s+b} f_s(\boldsymbol{x})$$

CLASSIFICATION AS SURROGATE TASK TO OBTAIN SUMMARIES

Classification machine learning techniques (e.g. binary cross entropy neural network) can be used to obtain really good approximations of $s_{\mathrm{clf}}(\boldsymbol{x}|\boldsymbol{\theta})$ with enough simulated samples



bayes optimal classifier $s_{\mathrm{clf}}(\boldsymbol{x}|\boldsymbol{\theta})$ using 3D synthetic example analytical density

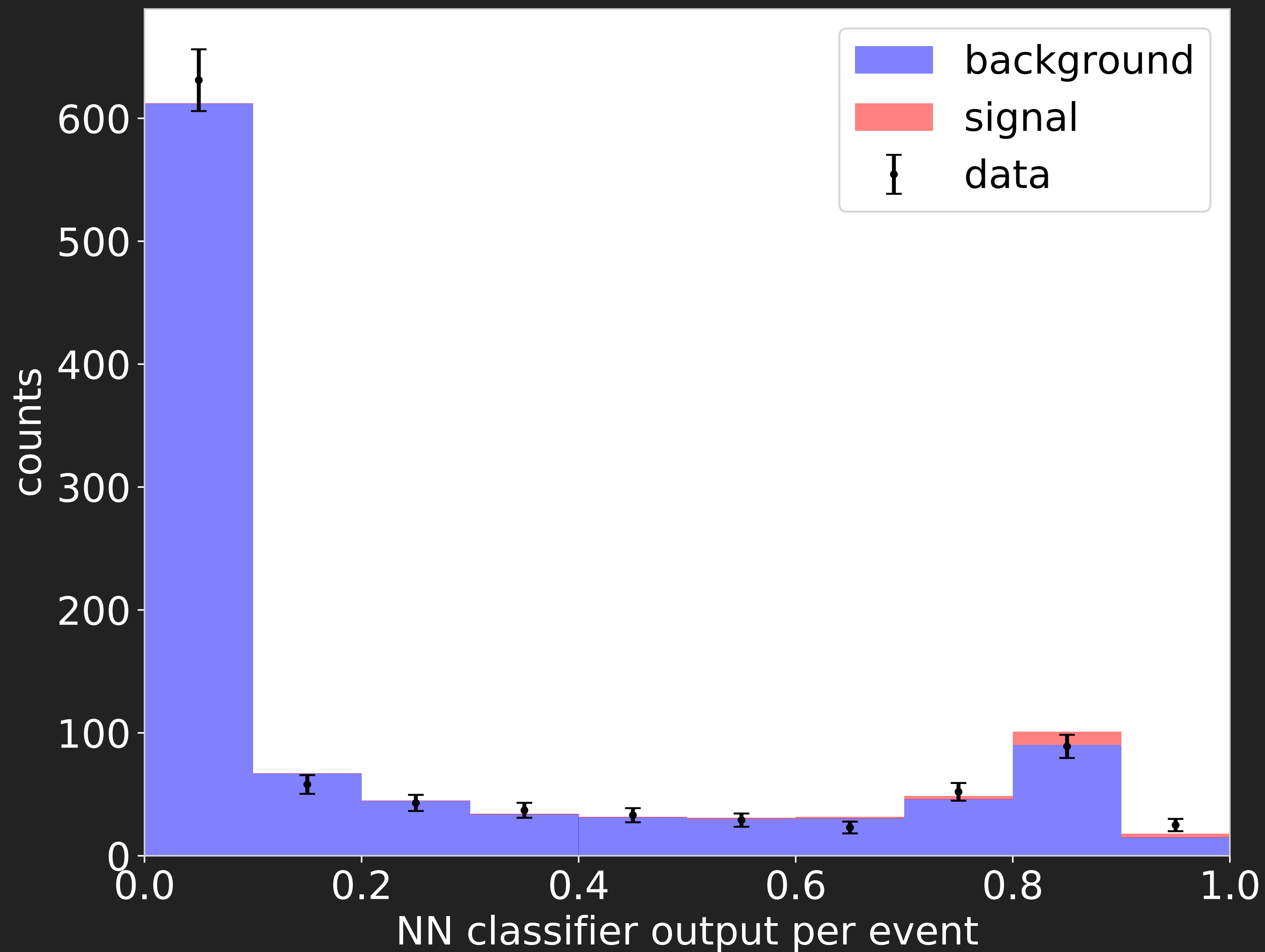


output of neural network trained using using sig and bkg classification

CLASSIFIER-BASED INFERENCE

A trained probabilistic classifer $d(\mathbf{x})$ provides a fixed approximation of $s_{\mathrm{clf}}(\boldsymbol{x}|\boldsymbol{\theta})$.

How can it be used for statistical inference given data $\mathcal{D}$?



1-D → cut or histogram to build a Poisson counts non-parametric likelihood

$$\mathcal{L}(\mu|\boldsymbol{\theta}) = \prod_{\mathrm{i}\in\mathrm{bins}} \mathrm{Pois}(n_{\mathrm{i}}|\mu \cdot s_{\mathrm{i}}(\boldsymbol{\theta}) + b_{\mathrm{i}}(\boldsymbol{\theta}))$$

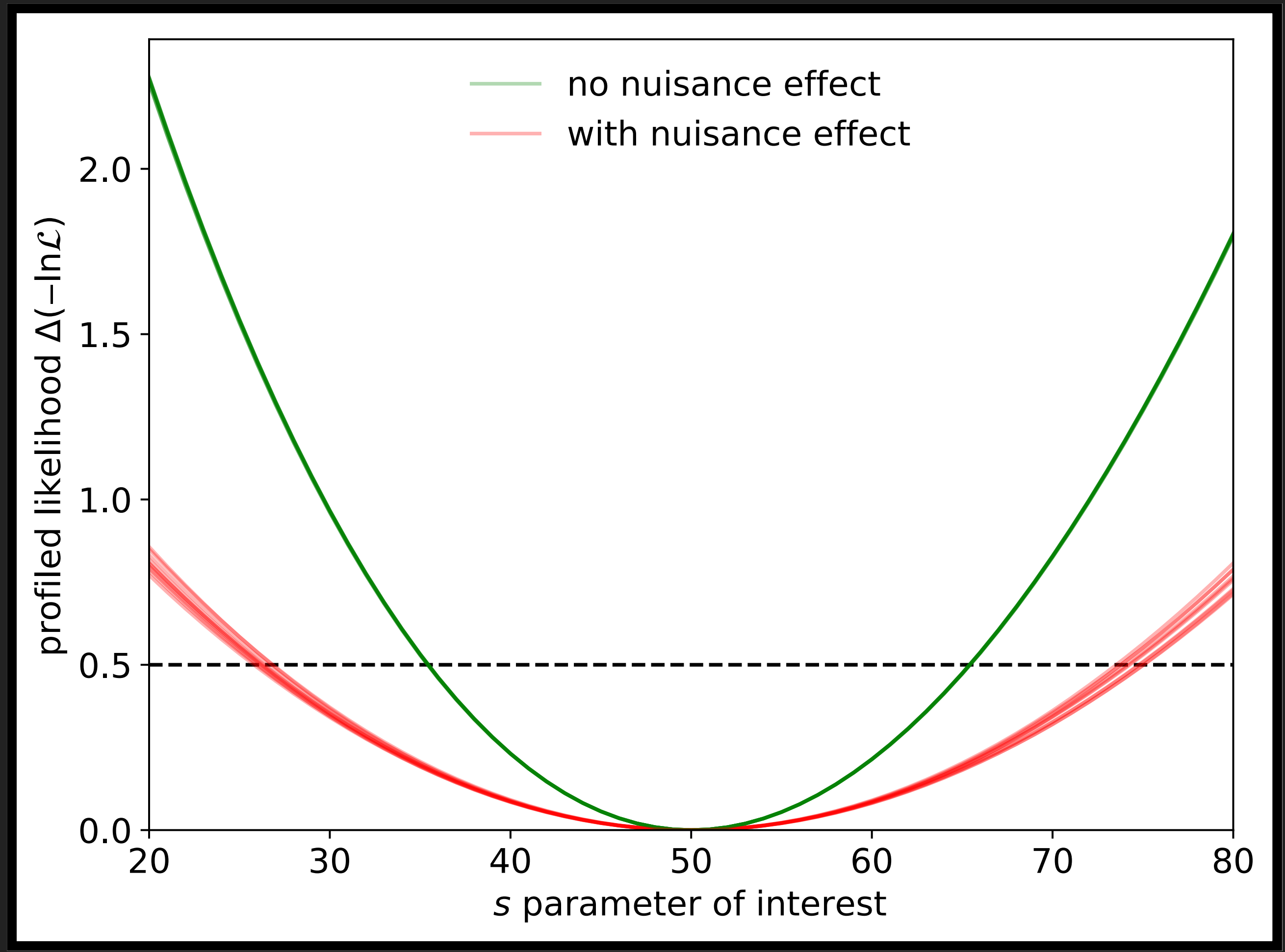
which can be used for statistical inference, such as measuring μ given $\mathcal{D}$

REAL WORLD: MODELLING UNCERTAINTIES DEGRADE INFERENCE

Simulations are imperfect, mainly due to the limited information of the system being modelled

Lack of knowledge for inference accounted by additional unknown parameters (nuisance parameters η)

Causes a degradation of classifier-based inference, leading to larger measurement uncertainties

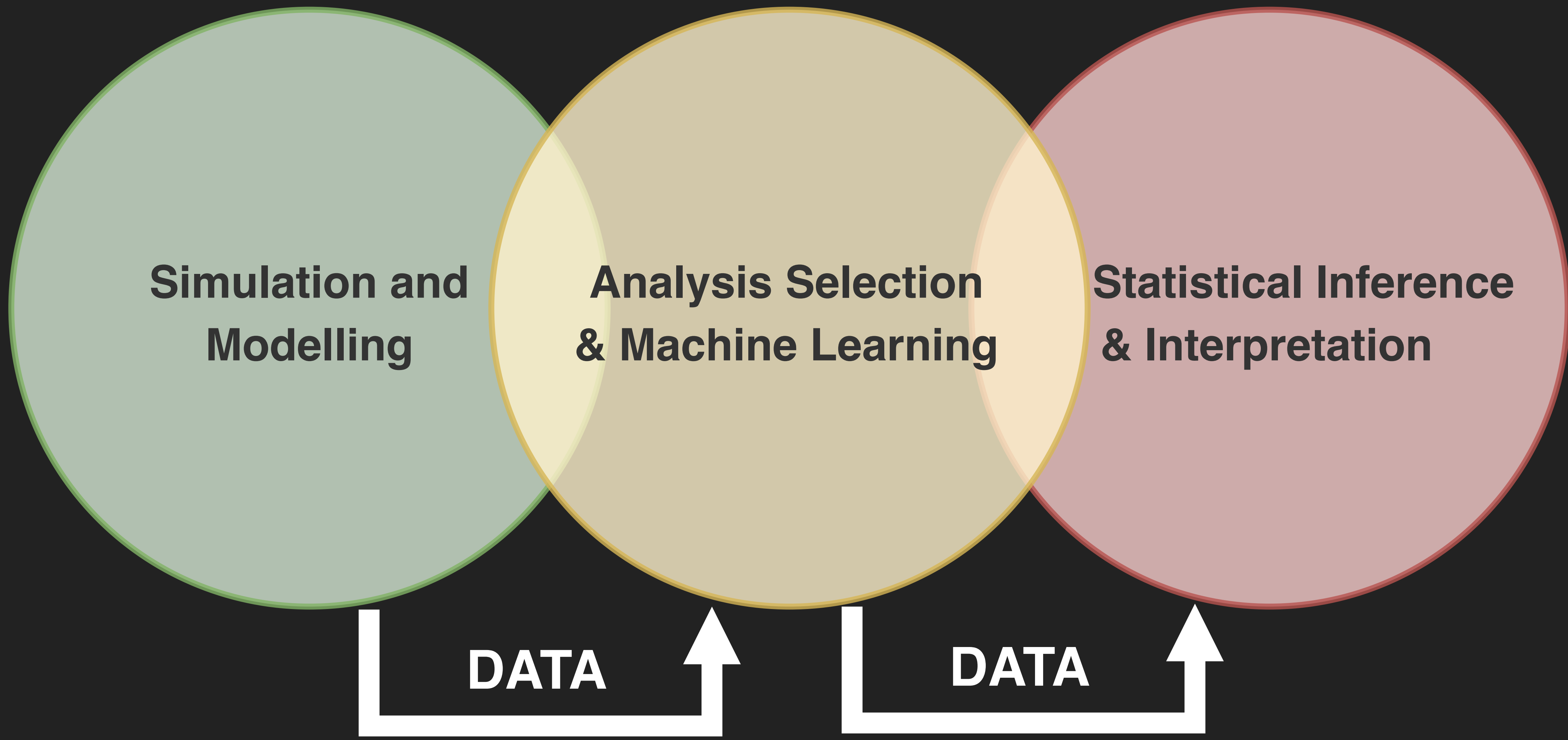


UPPER LIMIT OF ML USEFULNESS IN LHC ANALYSES

Classifiers can be made pivotal as described in "Learning to Pivot" by G. Louppe et al. A review/benchmarks on how to deal with systematics when using machine learning can be found in Adversarial learning to eliminate systematic errors: a case study in High Energy Physics by Victor Estrade et al NIPS2017.

EXPLORING WITH NEW MACHINE LEARNING PATHS

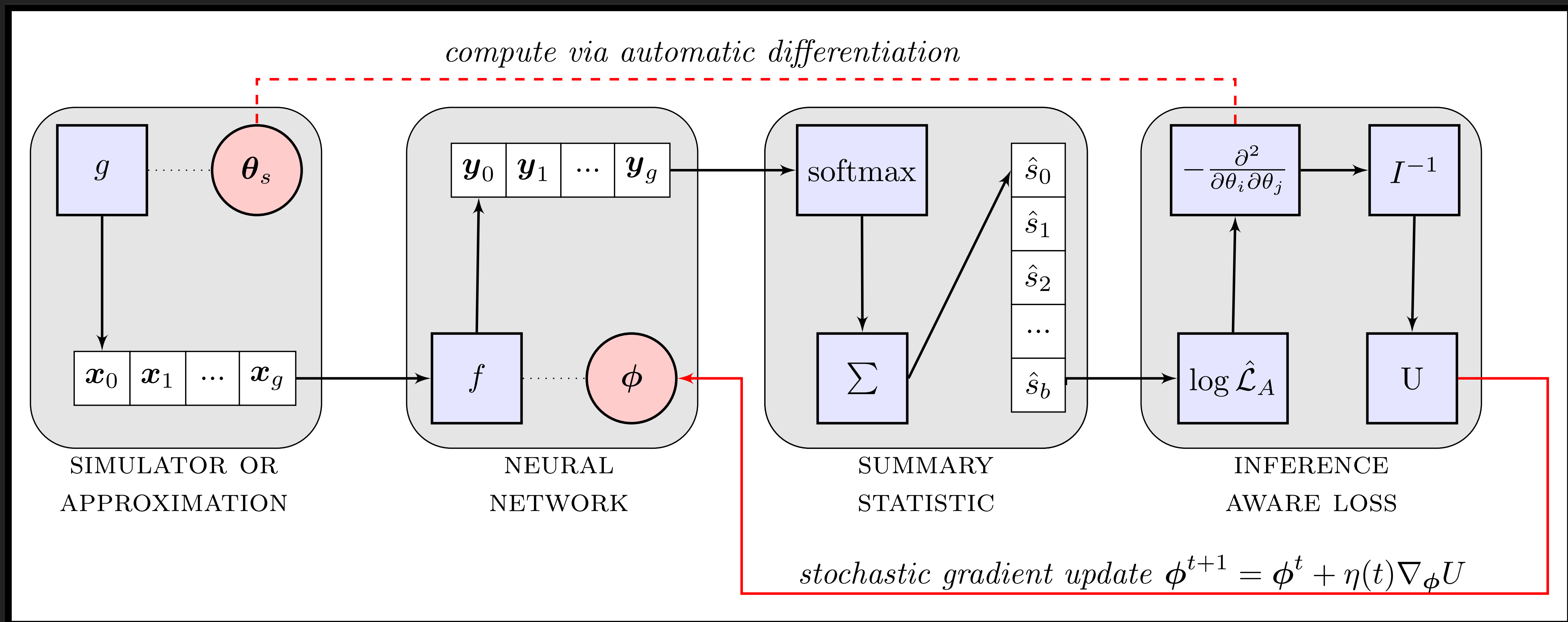
Embed some of the knowledge about modelling and statistical inference such as the uncertainty due to nuisance parameters in the dimensionality-reduction step



GLUE → AUTODIFF GRAPH FRAMEWORKS

INFERENCE-AWARE NEURAL OPTIMISATION

An approach to learn non-linear summary statistics by directly minimizing 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.

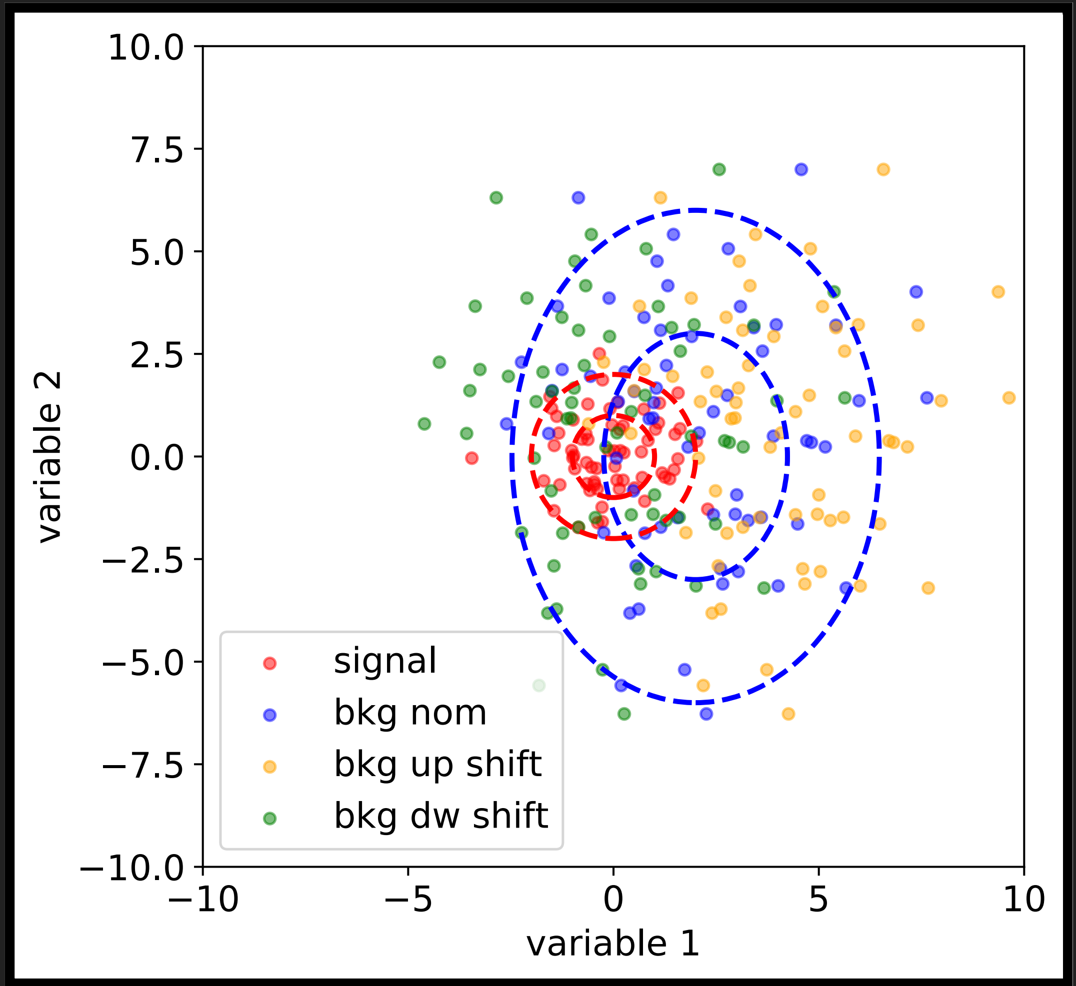
EFFECT OF NUISANCE PARAMETERS

Differentiable approximation of the effect of parameters of interest θ and nuisance parameters η over a given simulated event/observation $(\boldsymbol{x}_i, z_i, \boldsymbol{w}_i)$

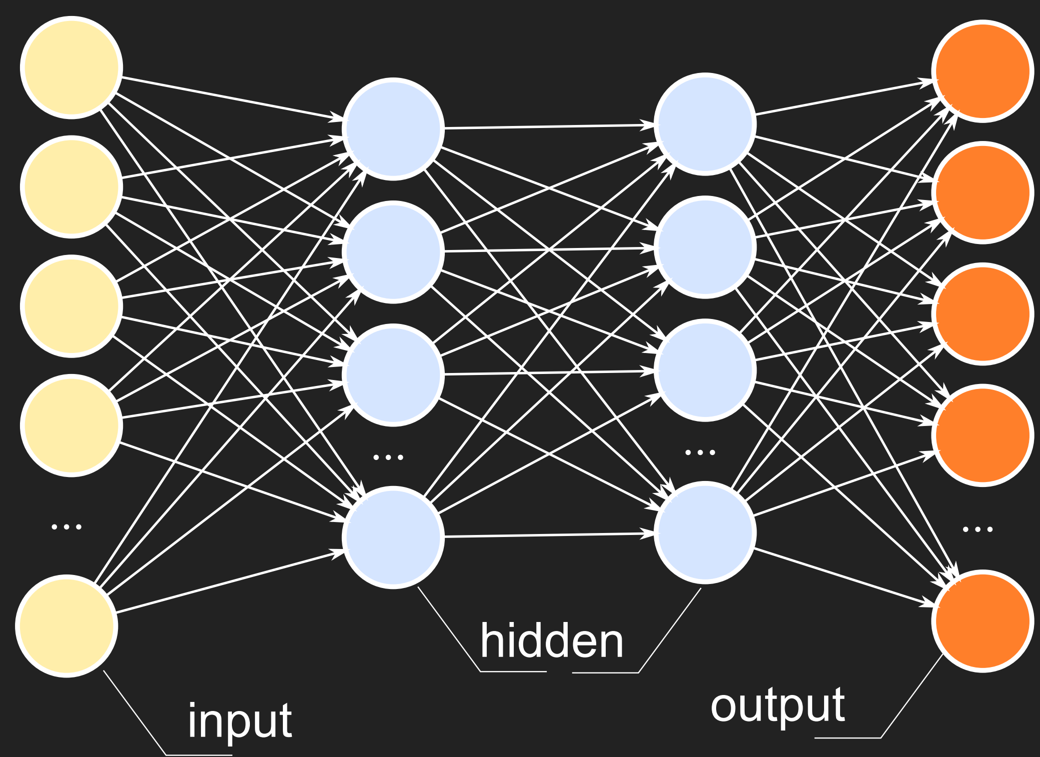
In general is a non-linear function that depends on the problem details, transforming features $\boldsymbol{x}_i$ or weights $\boldsymbol{w}_i$

Implemented in autodiff frameworks such as TensorFlow or PyTorch

Simple graphical example: shift for background in 2D, i.e $\mathbf{x}' = \mathbf{x} + \eta \cdot \mathbf{v}$



TRAINABLE PARAMETRIZED MODEL

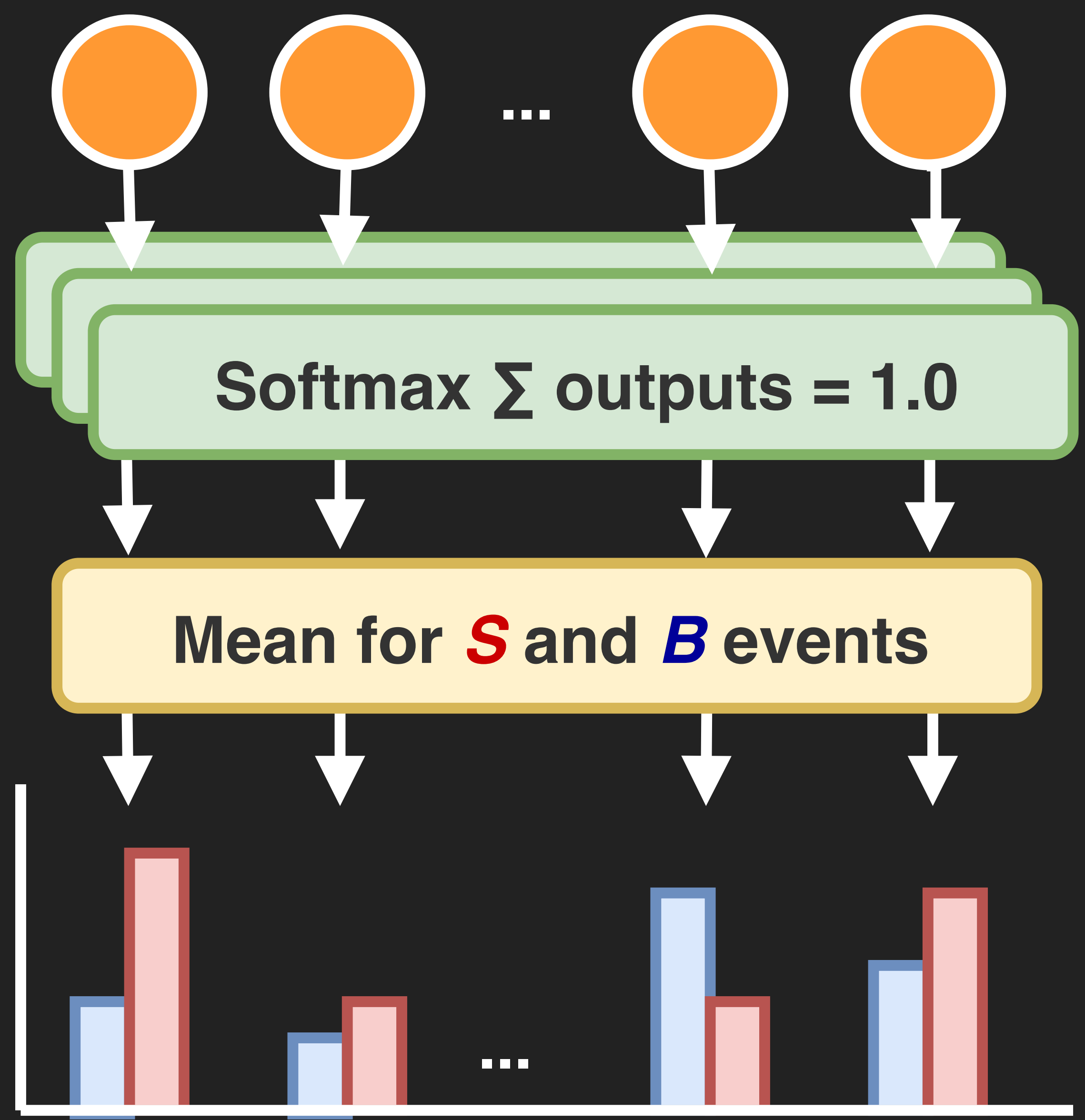


Parameters ϕ will be learnt during the optimisation process, defining summary statistic transformation $\mathbf{s}(x|\phi)$

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

NEURAL NETWORK OUTPUT → SUMMARY STATISTIC



We can approximate a histogram-like summary statistic from the NN output applying softmax for each event and summing over each dataset

$$\mathcal{L}(\theta, \eta; \phi) = \prod_{\mathrm{i} \in \mathrm{bins}} \mathrm{Pois}(n_{\mathrm{i}} | \alpha_s s_{\mathrm{i}} + \alpha_b b_{\mathrm{i}})$$

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

INFERENCE-MOTIVATED LOSS FUNCTION

If we expand negative log-likelihood around known minimum (e.g. Asimov $n_{\mathrm{i}} = \alpha_s s_{\mathrm{i}} + \alpha_b b_{\mathrm{i}}$):

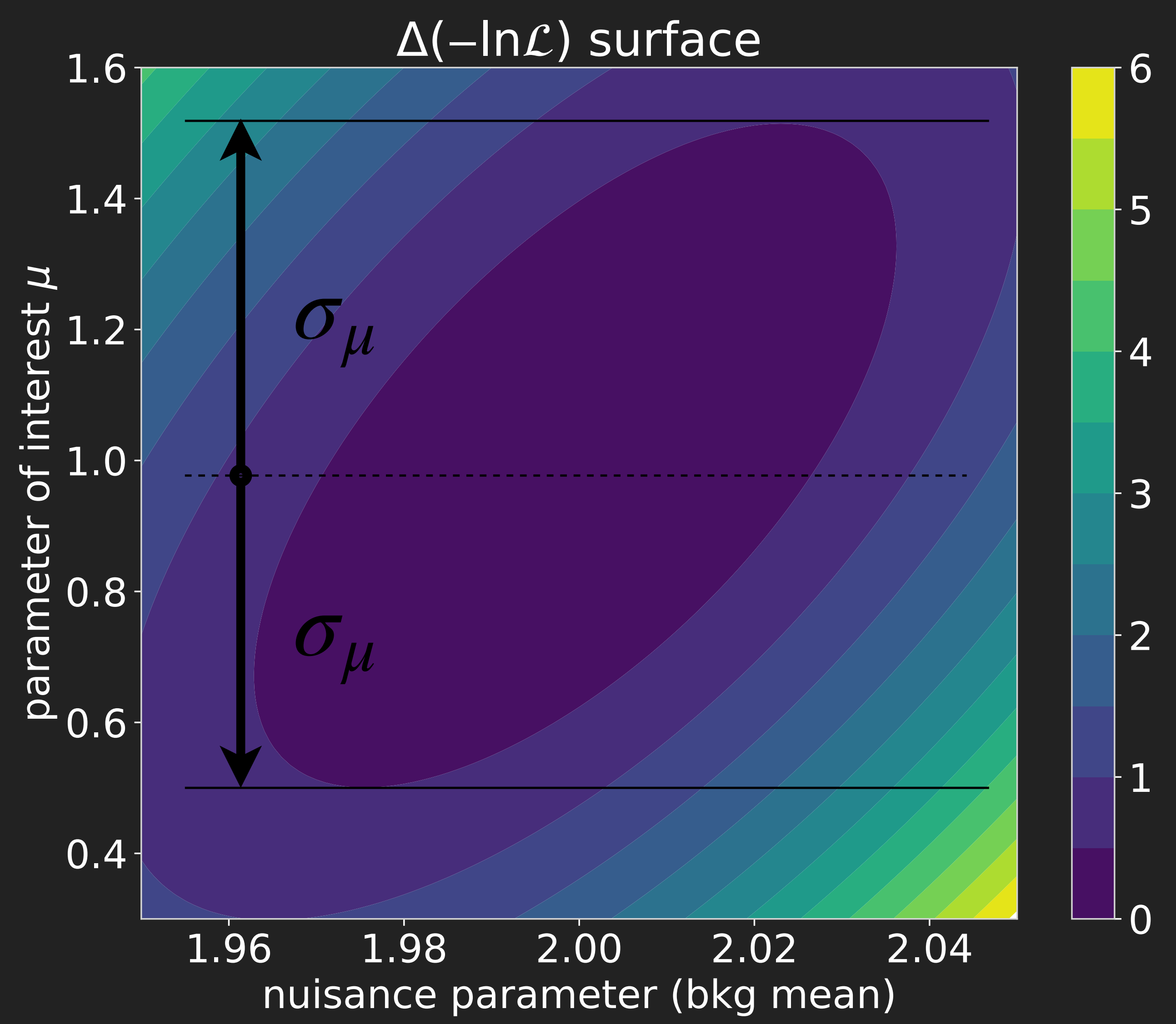
$$\text{covariance} \approx \mathbf{H^{-1}}(-\ln \mathcal{L})$$

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

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

that accounts for the effect of unknowns nuisance parameters

Training will thus vary the $\boldsymbol{s}(D|\boldsymbol{\phi})$ so it minimises the p.o.i. uncertainty σ_μ



EQUIVALENT TO THE LAPLACE APPROXIMATION IN BAYESIAN INFERENCE

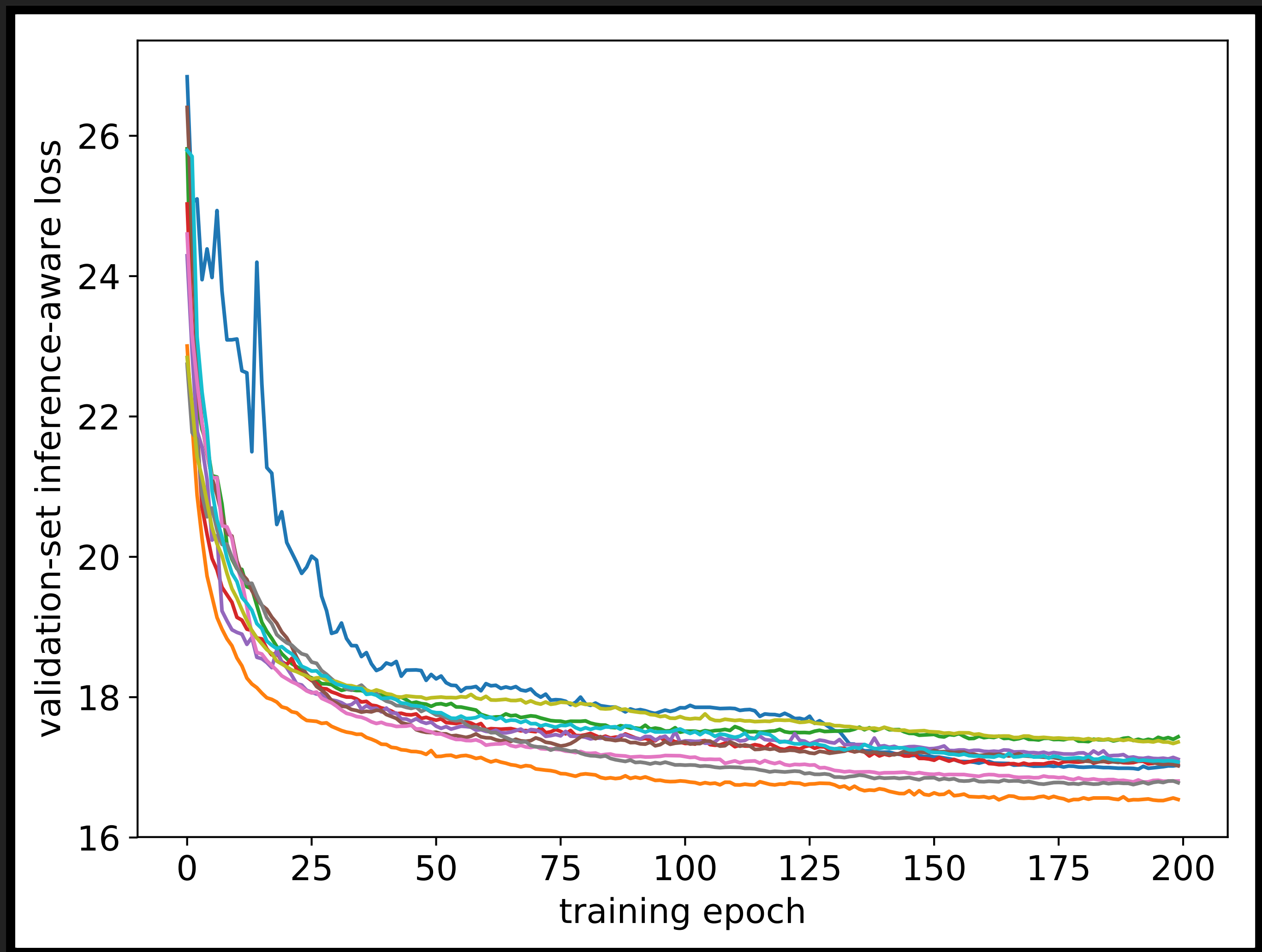
SYNTHETIC INFERENCE BENCHMARKS

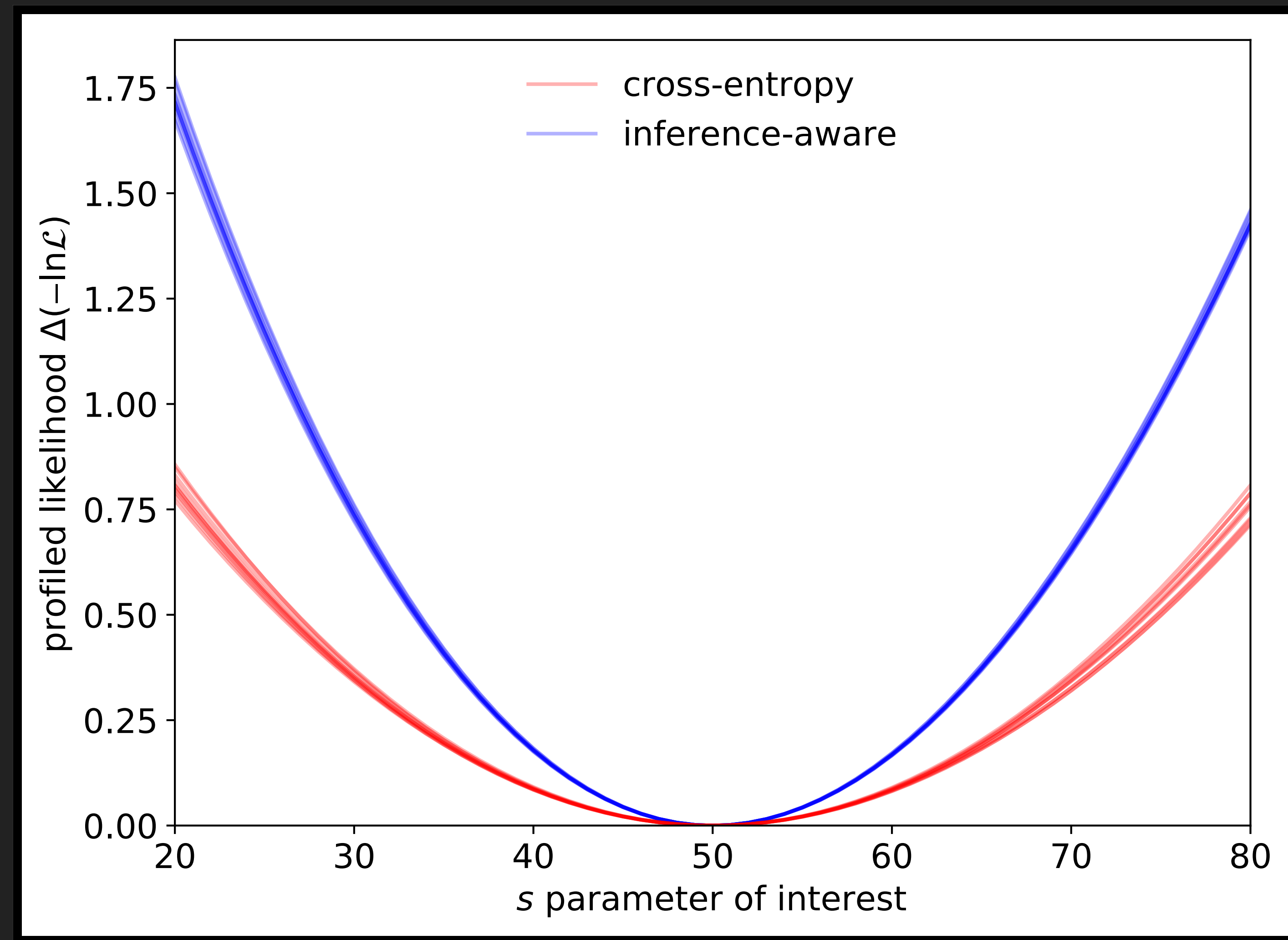
Several inference problems regarding $s = \mu b/(1-\mu)$ are considered based on the 3D benchmark mentioned before

	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)	$\mathcal{N}(\lambda\|3.0, 1.0)$	$\mathcal{N}(\lambda\|3.0, 1.0)$
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 can be used within INFERNO but not with probabilistic classifers

3D SYNTHETIC MIXTURE RESULTS (BENCHMARK 2)





INFERNO consistently converges to low-variance summary statistics

Clearly outperforms classifers in the presence of nuisance parameters

COMPARISON WITH CLASSIFICATION-BASED APPROACH

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

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

CONCLUSIONS AND PROSPECTS

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 but studies applied to more complex problems (see talk by Victor Estrade later today) and comparisons with alternative techniques [1] are needed to shed more light on real-world usefulness

[1] Particularly "Learning to Pivot" and the promising techniques by J. Brehmer et al "Mining gold from implicit models to improve likelihood-free inference" (2018) available at http://arxiv.org/abs/1805.12244

ON ANSWERING THE RIGHT QUESTIONS...

Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise

John Wilder Tukey (1915-2000)
in *The future of data analysis* (1962)

Image Source and Biography

MORE DETAILS ON ARXIV PREPRINT

Cornell University

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

export.arXiv.org > stat > arXiv:1806.04743

Search or Article ID | All papers

(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)

Submission history

From: Pablo de Castro [view email]
[v1] Tue, 12 Jun 2018 20:08:53 GMT (852kb,D)
[v2] Thu, 11 Oct 2018 12:41:56 GMT (339kb,D)

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

Bookmark (what is this?)

published on a stat.ML preprint arxiv.org/abs/1806.04743
feedback and comments: DM @pablodecm or pablo.decastro@cern.ch