A “Machine-learned" matrix element method

The “optimal" tool for multi-variate analysis on multi-dimensional theory space

arXiv: 1805.00020, 1805.00013, 1805.12244, 1907.10621, 2010.06439

DESY Journal Club (Feb 18, 2021)

Zhuoni Qian

Mining gold from implicit models to improve likelihood-free inference

Johann Brehmer,[1] **Gilles Louppe,**[2] **Juan Pavez,**[3] **and Kyle Cranmer**[1]
[1] New York University, [2] University of Liège, [3] Federico Santa María Technical University
johann.brehmer@nyu.edu, g.louppe@uliege.be,
juan.pavezs@alumnos.usm.cl, kyle.cranmer@nyu.edu

Abstract

Simulators often provide the best description of real-world phenomena. However, the density they implicitly define is often intractable, leading to challenging inverse problems for inference. Recently, a number of techniques have been introduced in which a surrogate for the intractable density is learned, including normalizing flows and density ratio estimators. We show that additional information that characterizes the latent process can often be extracted from simulators and used to augment the training data for these surrogate models. We introduce several new loss functions that leverage this augmented data, and demonstrate that these new techniques can improve sample efficiency and quality of inference.

Constraining Effective Field Theories with Machine Learning

Johann Brehmer,[1] Kyle Cranmer,[1] Gilles Louppe,[2] and Juan Pavez[3]

[1] *New York University, USA*
[2] *University of Liège, Belgium*
[3] *Federico Santa María Technical University, Chile*

A Guide to Constraining Effective Field Theories with Machine Learning

Johann Brehmer,[1] Kyle Cranmer,[1] Gilles Louppe,[2] and Juan Pavez[3]

[1] *New York University, USA*
[2] *University of Liège, Belgium*
[3] *Federico Santa María Technical University, Chile*

(Dated: 30th July 2018)

We present powerful new analysis techniques to constrain effective field theories at the LHC. By leveraging the structure of particle physics processes, we extract extra information from Monte-Carlo simulations, which can be used to train neural network models that estimate the likelihood ratio. These methods scale well to processes with many observables and theory parameters, do not require any approximations of the parton shower or detector response, and can be evaluated in microseconds. We show that they allow us to put significantly stronger bounds on dimension-six operators than existing methods, demonstrating their potential to improve the precision of the LHC legacy constraints.

We develop, discuss, and compare several inference techniques to constrain theory parameters in collider experiments. By harnessing the latent-space structure of particle physics processes, we extract extra information from the simulator. This augmented data can be used to train neural networks that precisely estimate the likelihood ratio. The new methods scale well to many observables and high-dimensional parameter spaces, do not require any approximations of the parton shower and detector response, and can be evaluated in microseconds. Using weak-boson-fusion Higgs production as an example process, we compare the performance of several techniques. The best results are found for likelihood ratio estimators trained with extra information about the score, the gradient of the log likelihood function with respect to the theory parameters. The score also provides sufficient statistics that contain all the information needed for inference in the neighborhood of the Standard Model. These methods enable us to put significantly stronger bounds on effective dimension-six operators than the traditional approach based on histograms. They also outperform generic machine learning methods that do not make use of the particle physics structure, demonstrating their potential to substantially improve the new physics reach of the LHC legacy results.

MadMiner: Machine learning–based inference for particle physics

Johann Brehmer,[1,*] Felix Kling,[2,3,†] Irina Espejo,[1,‡] and Kyle Cranmer[1,§]
[1] *Center for Data Science and Center for Cosmology and Particle Physics, New York University, New York, NY 10003, USA*
[2] *Department of Physics and Astronomy, University of California, Irvine, CA 92697, USA*
[3] *SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA*

Precision measurements at the LHC often require analyzing high-dimensional event data for subtle kinematic signatures, which is challenging for established analysis methods. Recently, a powerful family of multivariate inference techniques that leverage both matrix element information and machine learning has been developed. This approach neither requires the reduction of high-dimensional data to summary statistics nor any simplifications to the underlying physics or detector response. In this paper we introduce `MadMiner`, a Python module that streamlines the steps involved in this procedure. Wrapping around `MadGraph5_aMC` and `Pythia 8`, it supports almost any physics process and model. To aid phenomenological studies, the tool also wraps around `Delphes 3`, though it is extendable to a full `Geant4`-based detector simulation. We demonstrate the use of `MadMiner` in an example analysis of dimension-six operators in ttH production, finding that the new techniques substantially increase the sensitivity to new physics.

Simulation-based inference methods for particle physics

Johann Brehmer and Kyle Cranmer
New York University, New York, NY, 10003

Our predictions for particle physics processes are realized in a chain of complex simulators. They allow us to generate high-fidelty simulated data, but they are not well-suited for inference on the theory parameters with observed data. We explain why the likelihood function of high-dimensional LHC data cannot be explicitly evaluated, why this matters for data analysis, and reframe what the field has traditionally done to circumvent this problem. We then review new simulation-based inference methods that let us directly analyze high-dimensional data by combining machine learning techniques and information from the simulator. Initial studies indicate that these techniques have the potential to substantially improve the precision of LHC measurements. Finally, we discuss probabilistic programming, an emerging paradigm that lets us extend inference to the latent process of the simulator.

Theory prediction θ: Feynman diagrams and the matrix element is not the full story

Observed data x

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

Simulation chain for synthetic data (events):

Detector effects

Parton-shower, Hadroniazation

Parton-level

Difficulty: Expensive simulation w.r.t theory space and final state MVA analysis.

Goal: (efficiently) learning the likelihood $p(x, \theta)$ in the high dimensional data space.

Rapidly developing field of machine learning, map/analyze high-dimensional data efficiently

Simulation-based inference methods

Multi-variate analysis: MEM, BDT, general NN

$$\hat{p}_{MEM}(x|\theta) = \int \mathrm{d}z_p \; \hat{p}_{tf}(x|z_p)\, p(z_p|\theta) \sim \frac{1}{\sigma(\theta)} \int \mathrm{d}z_p \; \hat{p}_{tf}(x|z_p)\, |\mathcal{M}(z_p|\theta)|^2$$

Parton-level convolution:

$$\mathcal{P}(\mathbf{p}_i^{\text{vis}}|\alpha) = \frac{1}{\sigma(\alpha)} \sum_{k,l} \int dx_1 dx_2 \, \frac{f_k(x_1) f_l(x_2)}{2s x_1 x_2} \times \left[\prod_{j \in \text{inv.}} \int \frac{d^3 p_j}{(2\pi)^3 2E_j} \right] |\mathcal{M}_{kl}(p_i^{\text{vis}}, p_j; \alpha)|^2$$

Models are often implicit, meaning no explicit likelihood function $p(x \mid \theta)$ that describes how observable variables x depend on model parameters θ.

Rather, they come only in form of an architecture of simulators, that takes in parameters θ and simulates variables x.

Table 1. Dictionary of symbols that appear in this review (derived from Ref. [7]).

Symbol	Meaning	ML abstraction
θ	Theory parameters	Parameters of interest
x	All observables	Features
v	1-2 selected kinematic variables	Summary statistics
z_p	Parton-level four-momenta	Latent variables
z_s	Parton shower history	Latent variables
z_d	Detector interactions	Latent variables
$z = (z_p, z_s, z_d)$	Full simulation history of event	All latent variables
$p_{\text{full}}(\{x\}\|\theta)$	Full likelihood function, see Eq. (2)	Implicit density
$p(x\|\theta)$	Kinematic likelihood for single event (normalized fully differential xsec, Eq. (3))	Implicit density
$p_p(z_p\|\theta)$	Parton-level distribution	Tractable density
$p_s(z_s\|z_p)$	Parton-shower effects	Implicit density
$p_d(z_s\|z_p)$	Detector effects	Implicit density
$p_x(x\|z_d)$	Detector readout	Implicit density
$r(x\|\theta)$	Likelihood ratio function, see Eq. (4)	
$r(x, z\|\theta)$	Joint likelihood ratio, see Eq. (8)	Unbiased est. of $r(x\|\theta)$
$t(x)$	Score (locally optimal obs., Eq. (10))	
$t(x, z\|\theta)$	Joint score, see Eq. (9)	Unbiased est. of score
$\hat{\theta}$	Best fit for theory parameters	Estimator for θ
$\hat{p}(x\|\theta)$	Parameterized estimator for likelihood	
$\hat{r}(x\|\theta)$	Parameterized estimator for likelihood ratio	
$\hat{s}(x\|\theta)$	Parameterized classifier decision function	
$\hat{t}(x)$	Estimator for score	
$\hat{p}_{tf}(x\|z_p)$	Approximate shower and detector effects (transfer function)	

Likelihood-free inference: to emulate the intractable likelihood $p(x|\theta)$.

Example: Nomalizing-flow (neural density estimation), kernel density estimation (low-dimension only)

Supports arbitrary simulators without requiring approximations on the underlying physics and is amortized, allowing for an efficient evaluation after an upfront simulation and training cost.

"Likelihood-ratio" trick: training with reference point (e.g. EFT{θ}/SM).

easier when concerning θ with only the interested handful of NP d.o.f deviation from reference pt.

With binary cross-entropy as loss function (CARL)

Still amortized, allows for reweighing, sample efficient, but cannot generate new samples without LL.

Simulation-based inference method (improving ML with augmented data from simulator)

More is known from collider events simulation: hard process/Parton-level only is θ dependent.

Accessing collider corresponding latent variables and define:

joint likelihood-ratio

$$
\begin{aligned}
r(x_e, z_{\text{all}\,e}|\theta_0, \theta_1) &\equiv \frac{p(x_e, z_{\text{detector}\,e}, z_{\text{shower}\,e}, z_e|\theta_0)}{p(x_e, z_{\text{detector}\,e}, z_{\text{shower}\,e}, z_e|\theta_1)} \\
&= \frac{p(x_e|z_{\text{detector}\,e})}{p(x_e|z_{\text{detector}\,e})} \frac{p(z_{\text{detector}\,e}|z_{\text{shower}\,e})}{p(z_{\text{detector}\,e}|z_{\text{shower}\,e})} \frac{p(z_{\text{shower}\,e}|z_e)}{p(z_{\text{shower}\,e}|z_e)} \frac{p(z_e|\theta_0)}{p(z_e|\theta_1)} \\
&= \frac{p(z_e|\theta_0)}{p(z_e|\theta_1)} \,.
\end{aligned}
$$

joint score.

$$
\begin{aligned}
t(x_e, z_{\text{all}\,e}|\theta_0) &\equiv \nabla_\theta \log p(x_e, z_{\text{detector}\,e}, z_{\text{shower}\,e}, z_e|\theta_0) \\
&= \frac{p(x_e|z_{\text{detector}\,e})}{p(x_e|z_{\text{detector}\,e})} \frac{p(z_{\text{detector}\,e}|z_{\text{shower}\,e})}{p(z_{\text{detector}\,e}|z_{\text{shower}\,e})} \frac{p(z_{\text{shower}\,e}|z_e)}{p(z_{\text{shower}\,e}|z_e)} \frac{\nabla_\theta p(z_e|\theta)}{p(z_e|\theta)}\Big|_{\theta_0} \\
&= \frac{\nabla_\theta p(z_e|\theta)}{p(z_e|\theta)}\Big|_{\theta_0}
\end{aligned}
$$

the likelihood (ratio) dependence on θ given certain trajectory:

Or a given simulated event, the intractable likelihood $p(x \,|\, z_i)$ part cancel.

calculable from $|\mathscr{M}|^2$

Loss function of $\hat{g}(x)$ that approaches a true $g(x,z)$:

$$L[\hat{g}(x)] = \int \mathrm{d}x\,\mathrm{d}z\; p(x,z|\theta)\;|g(x,z)-\hat{g}(x)|^2$$
$$= \int \mathrm{d}x\;\underbrace{\left[\hat{g}^2(x)\int \mathrm{d}z\; p(x,z|\theta) - 2\hat{g}(x)\int \mathrm{d}z\; p(x,z|\theta)\,g(x,z) + \int \mathrm{d}z\; p(x,z|\theta)\,g^2(x,z)\right]}_{F(x)}.$$

The $g*(x)$ that extremizes $L[\hat{g}(x)]$ satisfy:

$$0 = \left.\frac{\delta F}{\delta \hat{g}}\right|_{g^*} = 2\hat{g}\underbrace{\int \mathrm{d}z\; p(x,z|\theta)}_{=p(x|\theta)} - 2\int \mathrm{d}z\; p(x,z|\theta)\,g(x,z) \quad \Rightarrow \quad g^*(x) = \frac{1}{p(x|\theta)}\int \mathrm{d}z\; p(x,z|\theta)\,g(x,z)$$

$r(x,z\,|\,\theta_0,\theta_1)$ regressing towards $r(x\,|\,\theta_0,\theta_1)$:

$$g^*(x) = \frac{1}{p(x|\theta_1)}\int \mathrm{d}z\; p(x,z|\theta_1)\,\frac{p(x,z|\theta_0)}{p(x,z|\theta_1)} = r(x|\theta_0,\theta_1)$$

By Minimizing $L[\hat{g}(x)]$

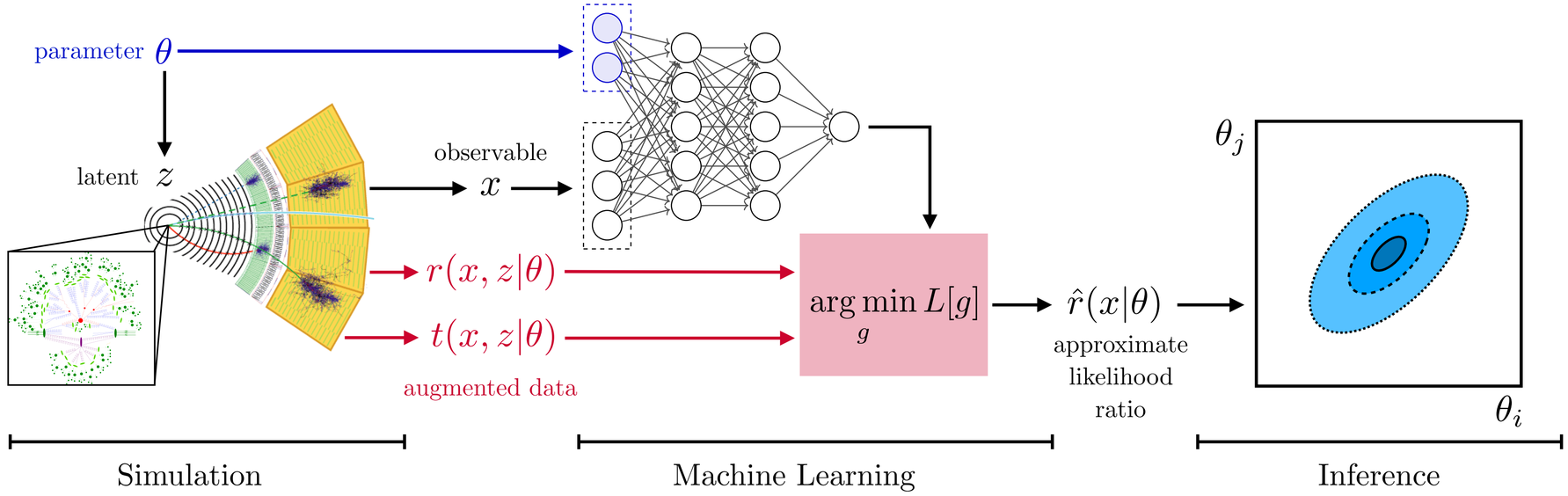
$$L[\hat{r}(x|\theta_0,\theta_1)] = \frac{1}{N}\sum_{(x_e,z_e)\sim p(x,z|\theta_1)} |r(x_e,z_{\mathrm{all},e}|\theta_0,\theta_1) - \hat{r}(x_e|\theta_0,\theta_1)|^2$$

$t(x,z\,|\,\theta_0)$ regressing towards $t(x\,|\,\theta_0)$:

$$g^*(x) = \frac{1}{p(x|\theta)}\int \mathrm{d}z\; \nabla_\theta p(x,z|\theta) = t(x|\theta)$$

$$L[\hat{t}(x|\theta)] = \frac{1}{N}\sum_{(x_e,z_e)\sim p(x,z|\theta)} |t(x_e,z_{\mathrm{all},e}|\theta) - \hat{t}(x_e|\theta)|^2$$

Abstract workflow of simulation-based ML optimiser:

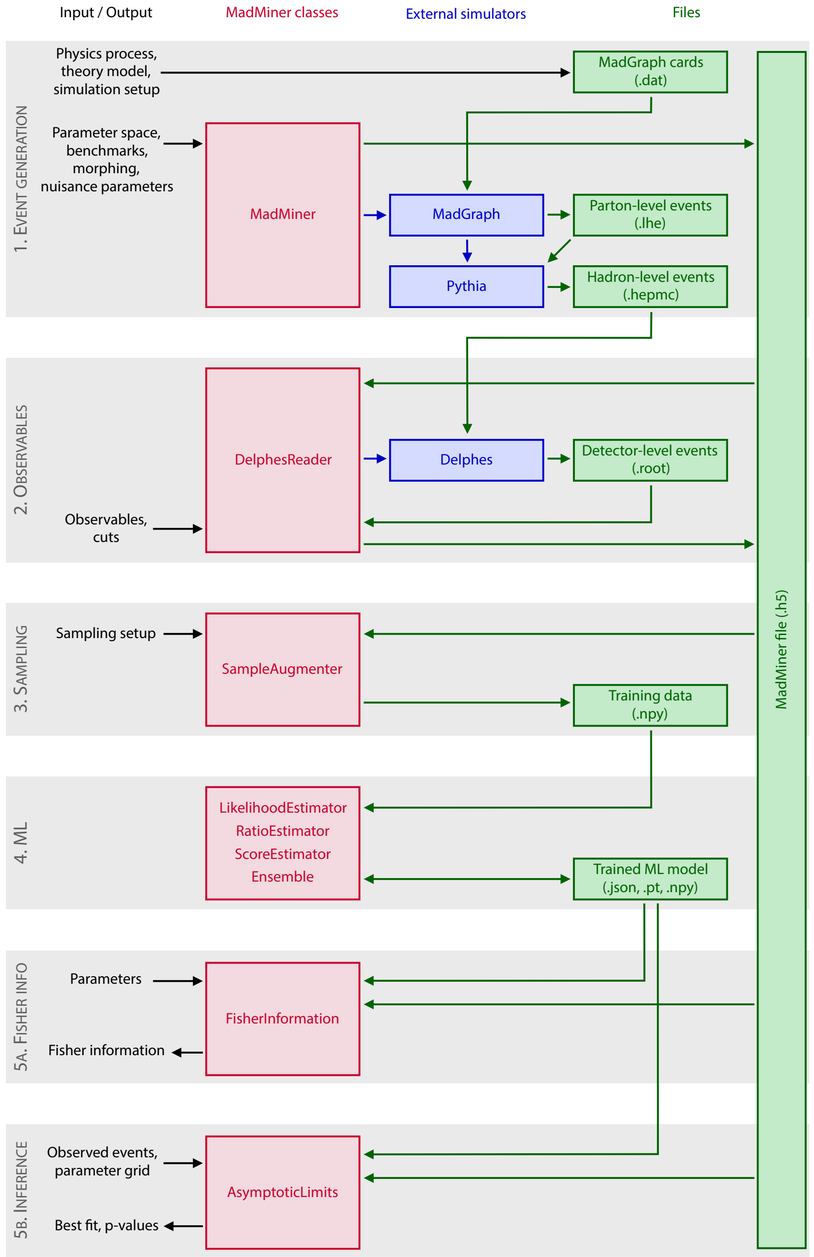


joint likelihood-ratio

$$
\begin{aligned}
r(x, z|\theta) &\equiv \frac{p(x, z|\theta)}{p_{\text{ref}}(x, z)} \\
&= \frac{p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p(z_p|\theta)}{p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p_{\text{ref}}(z_p)} \\
&= \frac{|\mathcal{M}|^2(z_p|\theta)}{|\mathcal{M}|^2_{\text{ref}}(z_p)}\, \frac{\sigma_{\text{ref}}}{\sigma(\theta)}
\end{aligned}
$$

joint score.

$$
\begin{aligned}
t(x, z|\theta) &\equiv \nabla_\theta \log p(x, z|\theta) \\
&= \frac{p_x(x|z_d)\, p_d(z_d|z_s)\, p_s(z_s|z_p)\, \nabla_\theta p_p(z_p|\theta)}{p_x(x|z_d)\, p_d(z_d|z_s)\, p_s(z_s|z_p)\, p_p(z_p|\theta)} \\
&= \frac{\nabla_\theta |\mathcal{M}|^2(z_p|\theta)}{|\mathcal{M}|^2(z_p|\theta)} - \frac{\nabla_\theta \sigma(\theta)}{\sigma(\theta)}\,.
\end{aligned}
$$



1907.10621 (MadMiner_Workflow)

In many cases the joint likelihood ratio and the joint score — quantities **conditioned** on the latent variables that characterise the trajectory through the data generation process — can be extracted from the simulator.

Substantially reduces the number of simulated events that are necessary for a good performance—in some cases by multiple orders of magnitude

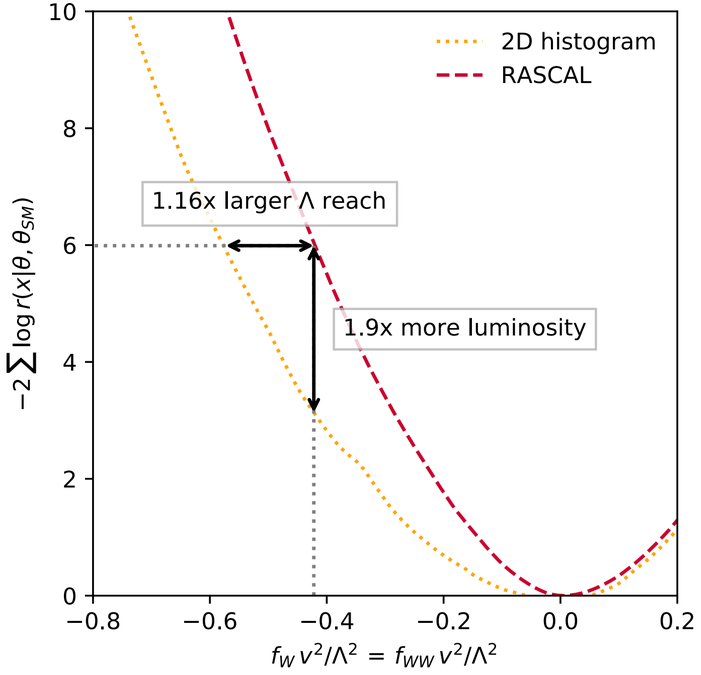
machine-learning version of the Matrix Element Method.
MEM (computationally expensive numerical integrals) -> ML regression

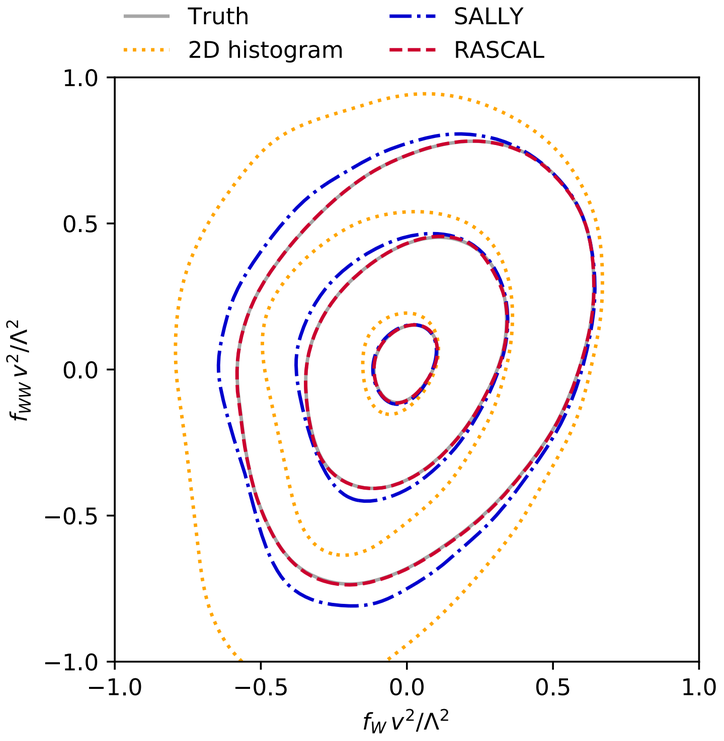
Instead of manually specifying simplified smearing functions, the effect of parton shower and detector is learned from full simulations.
the likelihood ratio can be evaluated in microseconds per event and parameter point.

An Example of constraining contour on 2D EFT space on the fly:

Study EFT operators through WBF of the Higgs in the four-lepton mode ($VV^* \to h \to 4\ell$):

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





Truth from the exact Parton-level likelihood calculation

Binning on 2D over two sensitive observables

Local score/linear approximation

Full LL-ratio & score regression

General application to machine-learn intractable Likelihood

joint likelihood ratio and joint score, that are conditional on a particular stochastic execution trace z

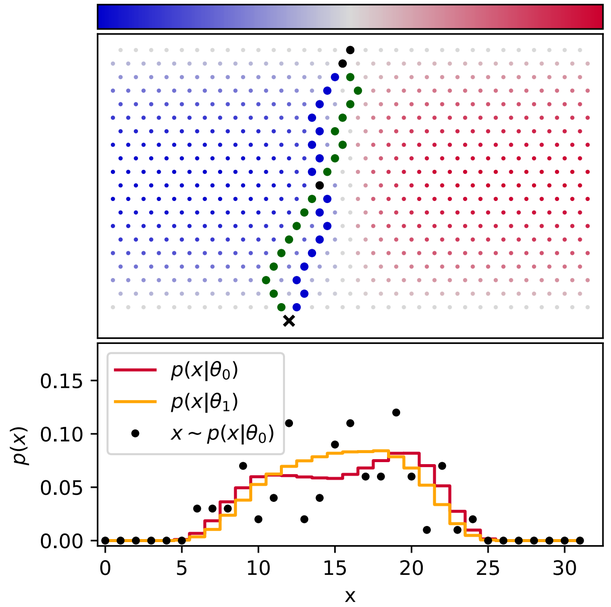


Figure 1: A toy simulation generalizing the Galton board where the transitions are biased left (blue) or right (red) depending on the nail position and the value of θ. Two example latent trajectories z are shown (blue and green), leading to the same observed value of x. Below, the distribution for $\theta_0 = -0.8$ and $\theta_1 = -0.6$. An example empirical distribution from 100 runs for θ_0 shows that the sample variance is much larger than the differences from θ_0 vs θ_1.

$$p(z_h, z_v, \theta)$$

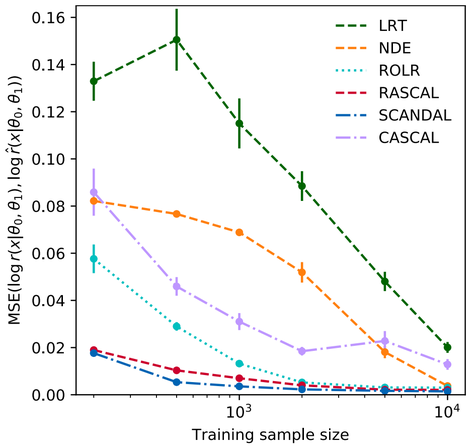
$$(1 - f(z_v))/2 + f(z_v)\sigma(5\theta(z_h - 1/2))$$

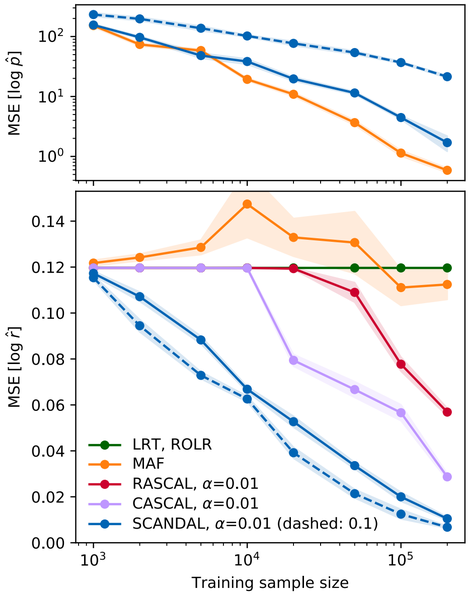
$$f(z_v) = \sin(\pi z_v)$$

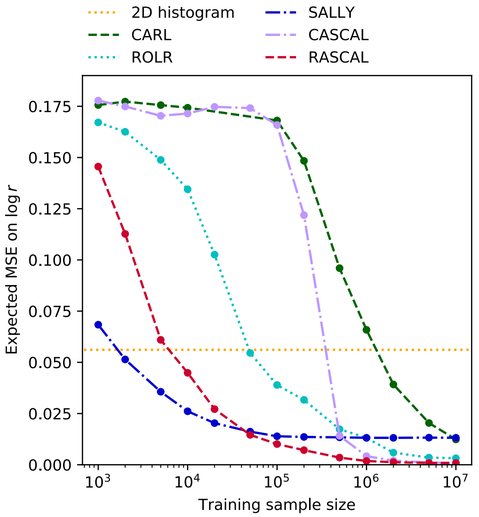
$p(x\,|\,\theta)$ can be regressed towards, by accumulating and minimising Loss function on:

$$\nabla_\theta \log p(z_h, z_v|\theta_0)$$

$$p(z_h, z_v|\theta_0)/p(z_h, z_v|\theta_1)$$







https://github.com/johannbrehmer/simulator-mining-example

Optimal summary statistics as versatile analysis tool:

Given theory parameter θ, optimal observable/score:

$$t(x) = \nabla_\theta \log p(x|\theta)\Big|_{\theta_{\text{ref}}}$$

MEM (optimal observable at parton level):

$$\hat{t}_{OO}(x) = \nabla_\theta \log \left(\int \mathrm{d}z_p \; \hat{p}_{tf}(x|z_p)\, p(z_p|\theta) \right) \Bigg|_{\theta=\theta_{\text{ref}}}$$

$$= \frac{\int \mathrm{d}z_p \; \hat{p}_{tf}(x|z_p)\, \nabla_\theta p(z_p|\theta_{\text{ref}})}{\int \mathrm{d}z_p \; \hat{p}_{tf}(x|z_p)\, p(z_p|\theta_{\text{ref}})} \,.$$

Sally Method to learn the joint score (ML minimising the mean squared error of $|\,\hat{t}(x) - t(x,z)\,|^2$):

Or more flexibly the joint score over an interested theory param. space θ (Fisher information matrix)

$$I_{ij}(\theta) = \frac{L\,\partial_i\sigma(\theta)\,\partial_j\sigma(\theta)}{\sigma(\theta)} + L\,\sigma(\theta) \int \mathrm{d}x \; p(x|\theta)\, t_i(x|\theta)\, t_j(x|\theta)$$

$$\approx \frac{L\,\partial_i\sigma(\theta)\,\partial_j\sigma(\theta)}{\sigma(\theta)} + \frac{L\,\sigma(\theta)}{n} \sum_{x\sim p(x|\theta)} t_i(x|\theta)\, t_j(x|\theta) \,,$$