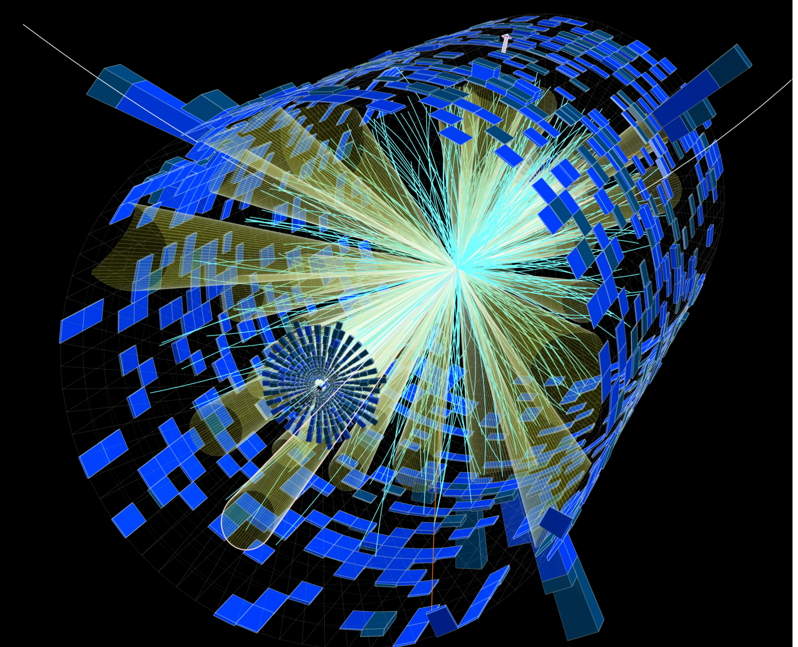
NYU Center for Data Science

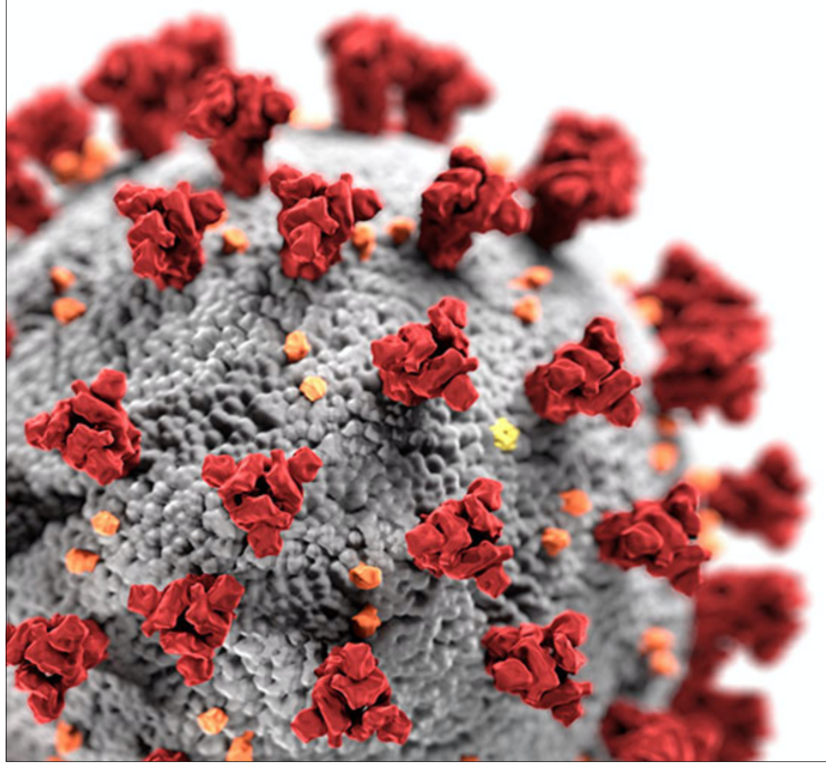
Center for Cosmology and Particle Physics

FUTURE ANALYSIS SYSTEMS



@KyleCranmer
New York University
Department of Physics
Center for Data Science
CILVR Lab

Sorry if this is a little disorganized, COVID has complicated work life



SUPPORT

NSF

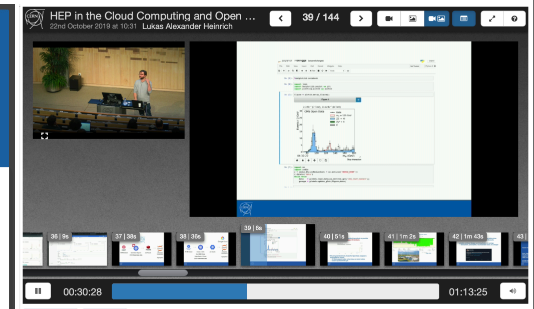
MSDSE
Moore - Sloan
Data Science Environments

dianahep

iris
hep

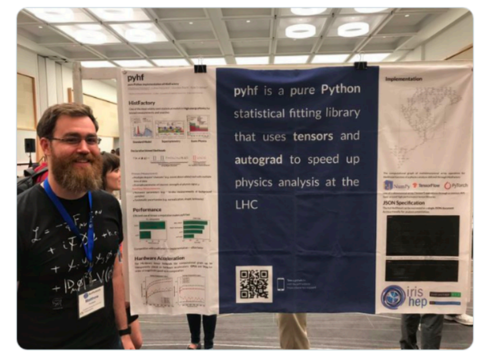
The SCAILFIN Project
scailfin.github.io

ACKNOWLEDGEMENTS

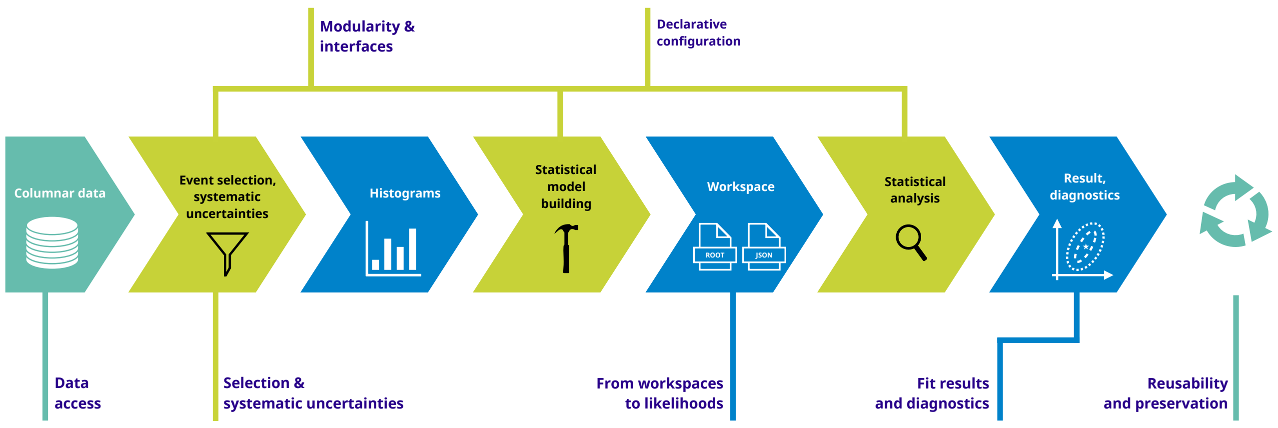


Matthew Feickert
@HEPfeickert

Thanks to everyone at #SciPy2019 who came and asked me great questions about pyhf!



ANALYSIS SYSTEMS



TOPICS

Accelerating analysis design

- more powerful observables
- end-to-end optimization
- benchmarking of algorithms

Accelerating fitting

- pyhf and a fitting service

More efficient simulation

- excursion
- Probabilistic programming

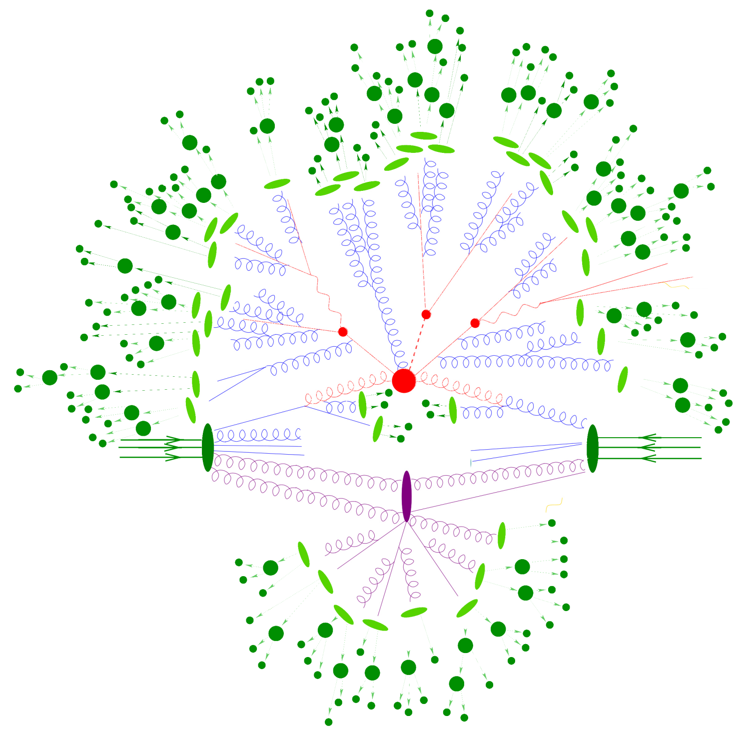
Extending impact of results

- RECAST

Core technologies:

- automatic differentiation
- GPUs & TPUs
- Cloud-native : docker, kubernetes
- Workflows & REANA
- Functions as a service Accelerating analysis design

PREDICTIONS IN PARTICLE PHYSICS

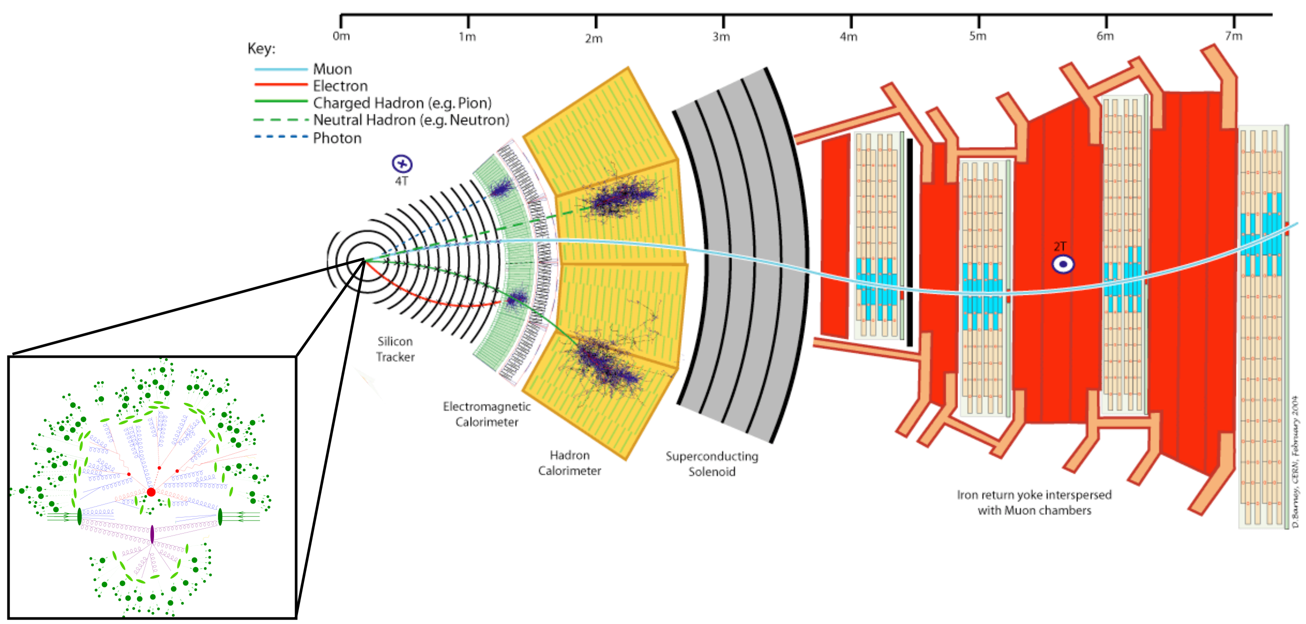


$$
\begin{aligned}
\mathcal{L}_{SM} = \; & \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^{a}_{\mu\nu}G^{\mu\nu}_{a}}_{\text{kinetic energies and self-interactions of the gauge bosons}} \\
& + \underbrace{\bar{L}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})L + \bar{R}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g'YB_{\mu})R}_{\text{kinetic energies and electroweak interactions of fermions}} \\
& + \underbrace{\frac{1}{2}\left|(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})\,\phi\right|^{2} - V(\phi)}_{W^{\pm},Z,\gamma,\text{and Higgs masses and couplings}} \\
& + \underbrace{g''(\bar{q}\gamma^{\mu}T_{a}q)\,G^{a}_{\mu}}_{\text{interactions between quarks and gluons}} + \underbrace{(G_{1}\bar{L}\phi R + G_{2}\bar{L}\phi_{c}R + h.c.)}_{\text{fermion masses and couplings to Higgs}}
\end{aligned}
$$

Conceptually: Prob(detector response | particles)

Implementation: Monte Carlo integration over micro-physics

Consequence: evaluation of the likelihood is intractable

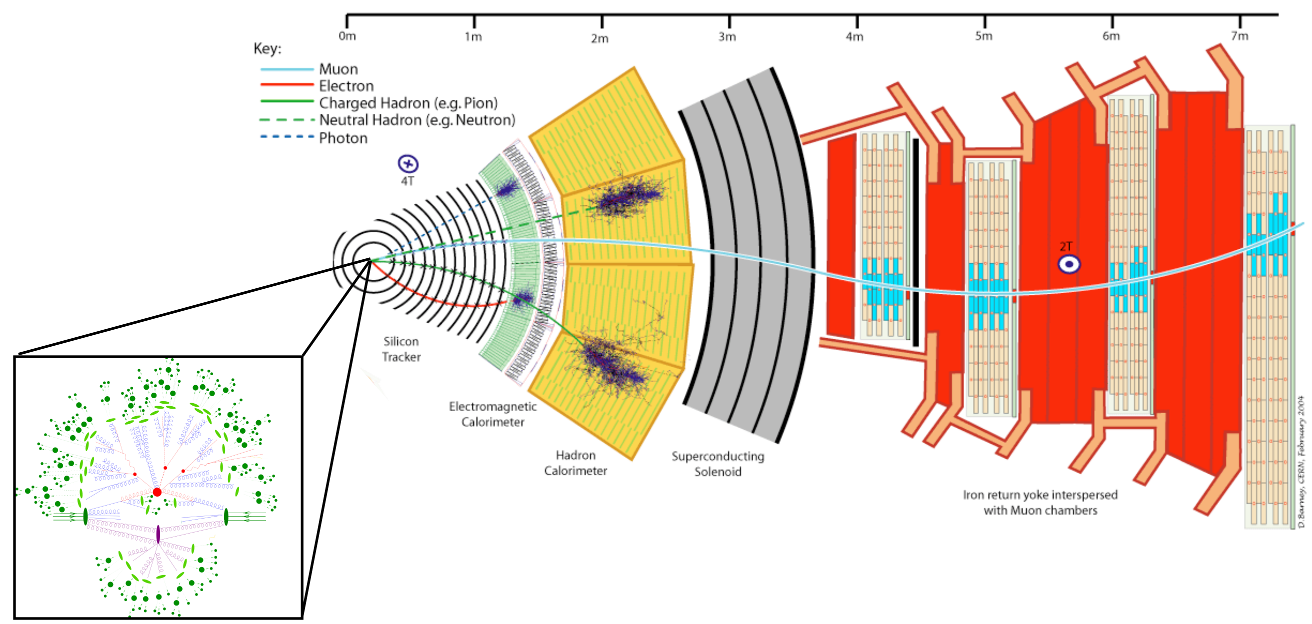


THE CAUSAL, GENERATIVE MODEL

Observables	Detector interactions	Shower splittings	Parton-level momenta	Theory parameters

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

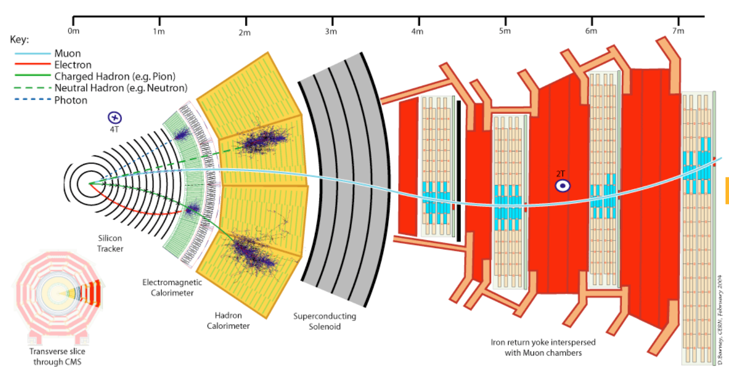
$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \; p(x|z_d) \quad p(z_d|z_s) \quad p(z_s|z_p) \quad p(z_p|\theta)$$

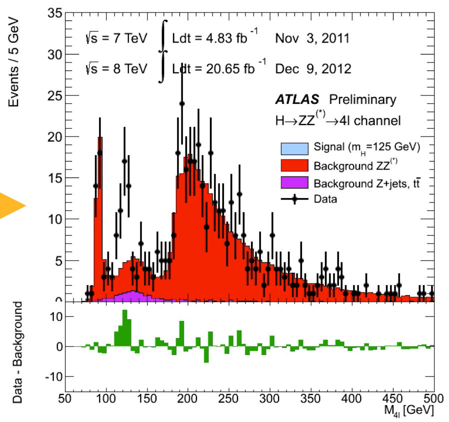


10^8 SENSORS → 1 REAL-VALUED QUANTITY

Most measurements and searches for new particles at the LHC are based on the distribution of a single summary statistic

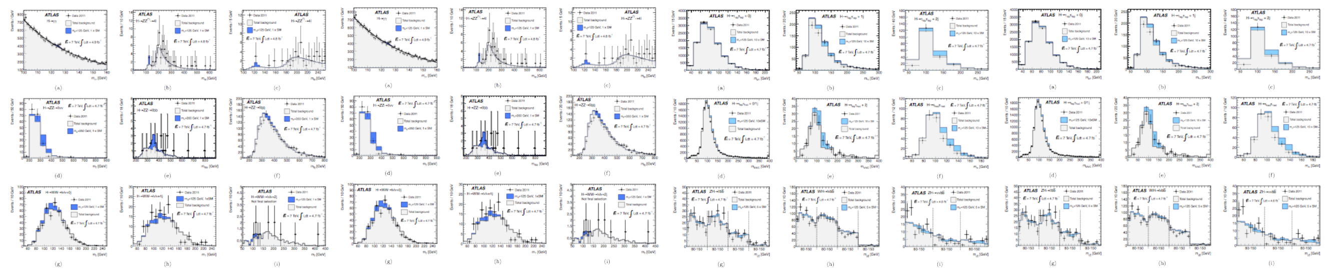
- choosing a good summary statistic (feature engineering) is a task for a skilled physicist and tailored to the goal of measurement or new particle search
- likelihood $p(x|\theta)$ **approximated** using histograms (univariate density estimation)

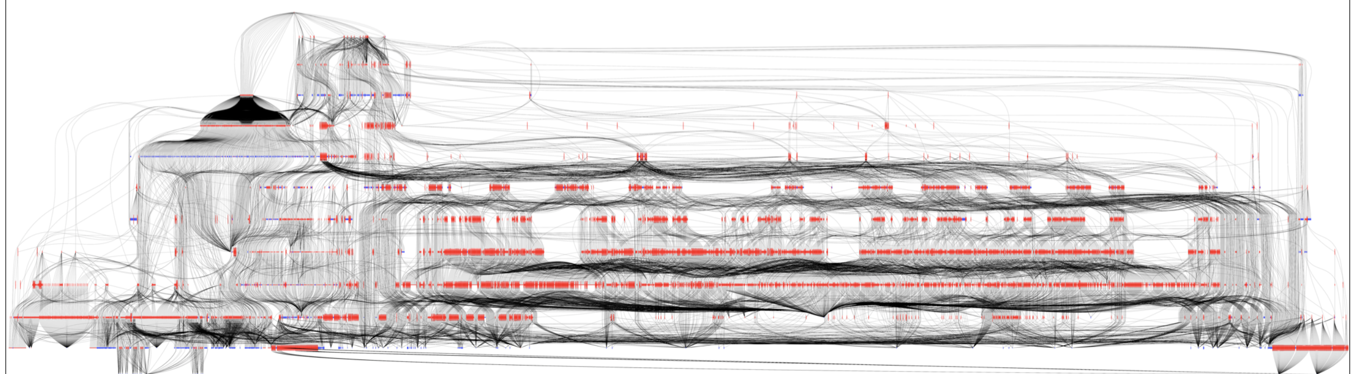




This doesn't scale if x is high dimensional!

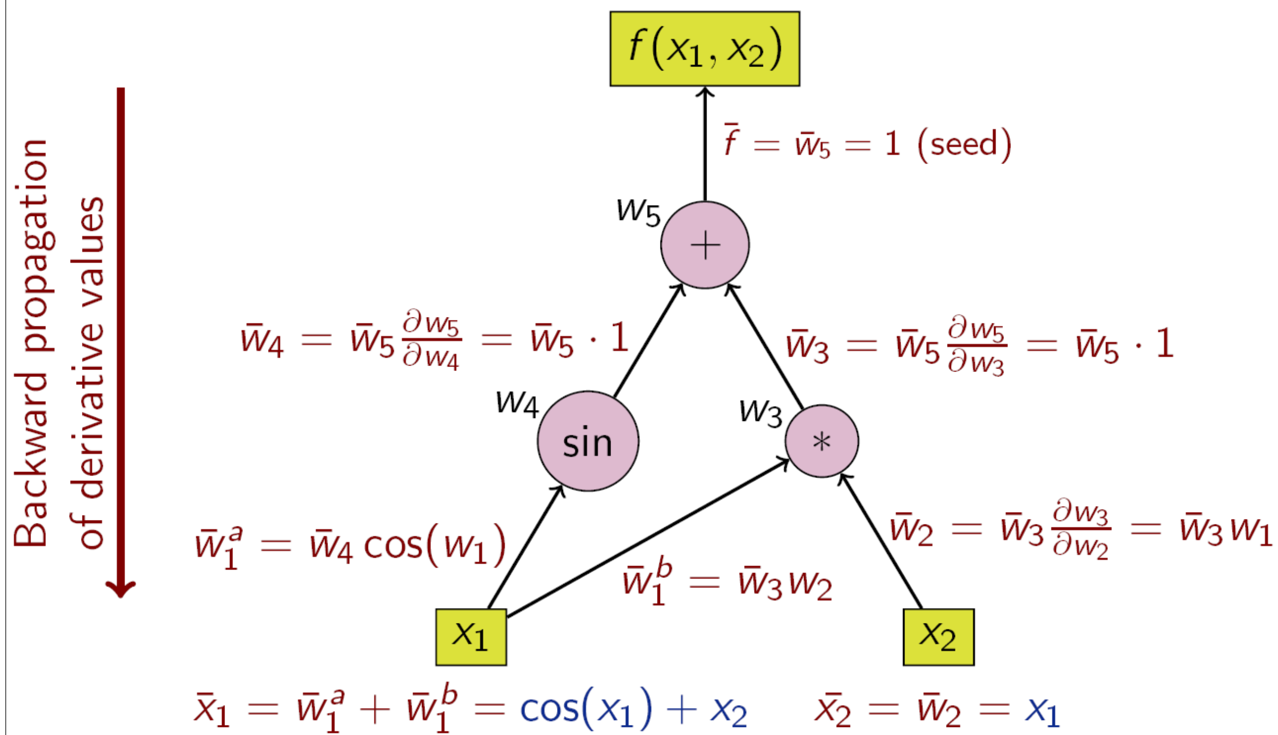
THE DISCOVERY OF THE HIGGS BOSON





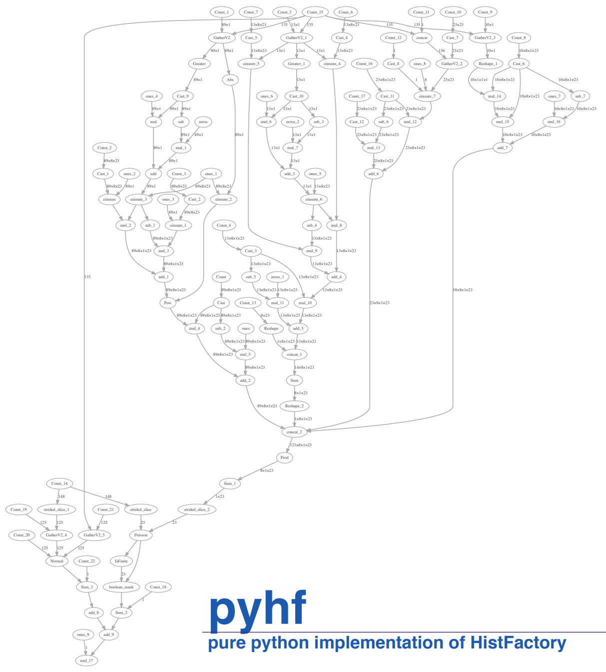
$$\mathbf{f}_{\mathrm{tot}}(\mathcal{D}_{\mathrm{sim}},\mathcal{G}|\boldsymbol{\alpha}) = \prod_{c\in\mathrm{channels}}\left[\mathrm{Pois}(n_c|\nu_c(\boldsymbol{\alpha}))\prod_{e=1}^{n_c} f_c(x_{ce}|\boldsymbol{\alpha})\right]\cdot\prod_{p\in\mathbb{S}} f_p(a_p|\alpha_p)$$

2. **reverse accumulation** computes the recursive relation: $\frac{dy}{dw_i} = \frac{dy}{dw_{i+1}} \frac{dw_{i+1}}{dw_i}$ with $w_0 = x$.



pyhf: auto-differentiable binned HEP likelihoods

Kyle Cranmer (NYU) Matthew Feickert (SMU)
Lukas Heinrich (CERN), Giordon Stark (UCSC)



pyhf

pure python implementation of HistFactory

implementation of HistFactory likelihood (1) as a **computational graph of multi-dimensional array operations.**

Use of array ("tensor") operations through a common API layer around high-performance tensor libraries: e.g.

NumPy TensorFlow PyTorch

Installation:

```
$> pip install pyhf
```

Example: simple number-counting experiment

```python
import pyhf
pdf = pyhf.simplemodels.hepdata_like(
  signal_data=[12.0, 11.0], bkg_data=[50.0, 52.0], bkg_uncerts=[3.0, 7.0]
)
CLs_obs, CLs_exp = pyhf.utils.hypotest(1.0, [51, 48] + pdf.config.auxdata, pdf, return_expected=True)


print('Observed: {}, Expected: {}'.format(CLs_obs, CLs_exp))
Observed: [0.05290116], Expected: [0.06445521]
```

Auto-Differentiation:

Tensor libraries from ML communty provide **exact gradients** for use in minimization.

$$\frac{\partial \mathcal{L}}{\partial \mu}, \frac{\partial \mathcal{L}}{\partial \theta_i}$$

Optimizers

pyhf likelihood are simple tensor-value python functions. Can use multiple minimization algorithms, such as `scipy.minimize` or MINUIT

OTHER ADVANTAGES OF USING TENSOR BACKENDS

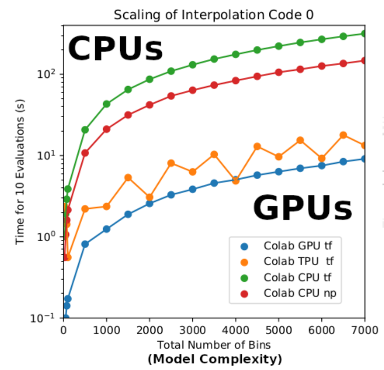
Slide from Matthew Feickert

- All numerical operations implemented in **tensor backends** through an API of n-dimensional array operations
- Using deep learning frameworks as computational backends allows for **exploitation of auto differentiation (autograd) and GPU acceleration**
- As huge buy in from industry we benefit for free as these frameworks are **continually improved** by professional software engineers

NumPy

PyTorch

TensorFlow

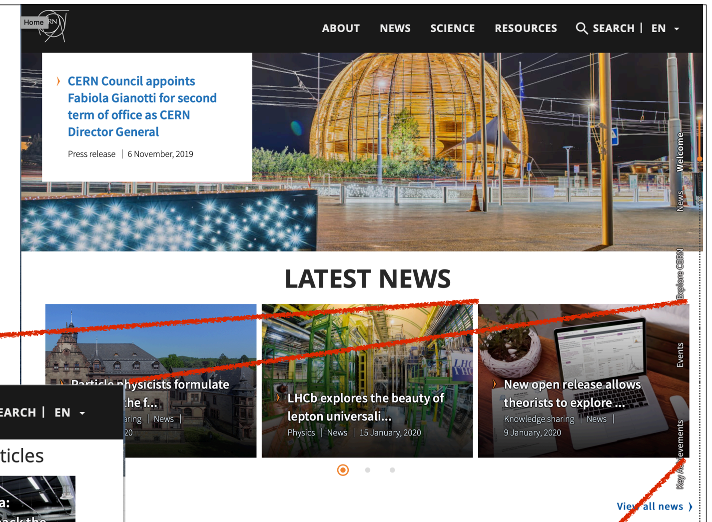


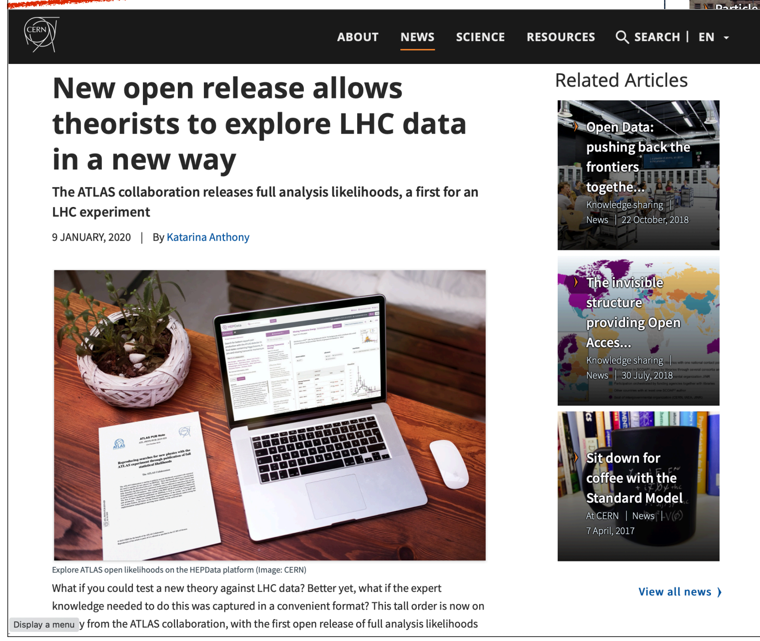
- Preliminary results
- Show hardware acceleration giving **order of magnitude speedup** for some models!
- Hardware acceleration benchmarking planned
- Improvements over traditional
  - 10 hrs to 30 min; 20 min to 10 sec

MAKING IT STANDARD

10 years later: community embraces publishing likelihoods as a standard

- Moved to JSON schema





pyhf
differentiable
Likelihoods

FITTING SERVICE

With JSON format, it is much easier to stream necessary data to a fitting service.

- Lukas prototyped this using functions-as-a-service

Ideally have a machine with a big GPU or TPU for this

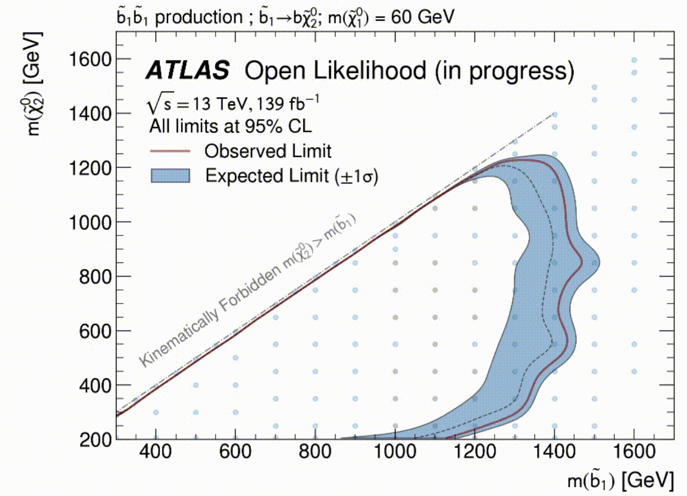
- From 10 min to a few seconds!

Accessing Fitting Service

```python
def func(data):
    filename = data['filename']
    region = data['region']
    m = re.compile("sbottom_(\d+)_(\d+)_(\d+)").search(filename).group(0)
    outname = 'results/region{}.result.{}.json'.format(region,m)
    for i in range(10):
        try:
            d = requests.post(
                '{}/region{}'.format(FITTING_SVC,region),
                data = open(filename), headers = {'Content-Type': 'application/json'}
            ).json()
            json.dump(d,open(outname,'w'))
            break
        except:
            pass#retry
```

```python
%matplotlib notebook
fig,ax = plt.subplots(1,1)
fig.set_size_inches(9.33,7)
apply_decorations(ax,label = 'Open Likelihood (in progress)')
```

Figure 1




```
No handles with labels found to put in legend.
```

```python
for x in glob.glob('results/*.json'):
    os.unlink(x)
cA = [{'region': 'A', 'filename': f} for f in glob.glob('RegionA/patch*_60.json')]
cC = [{'region': 'C', 'filename': f} for f in glob.glob('RegionC/patch*_60.json')]
configs = cA[:] + cC[:]
# np.random.shuffle(configs)

import time
import concurrent.futures
fig.canvas.draw()

with concurrent.futures.ThreadPoolExecutor(max_workers=MAX_WORKERS) as executor:
    for i,_ in enumerate(tqdm(executor.map(func, configs),total = len(configs))):
        if i > 5 and i % 5 == 0:
            make_plot(ax,label = 'Open Likelihood (in progress)', color = 'steelblue', showPoints = True)
            fig.canvas.draw()
        time.sleep(.005)
make_plot(ax, label = 'Open Likelihood', color = 'gold', showPoints = False)
```

35% 91/259 [01:55<01:54, 1.46it/s]

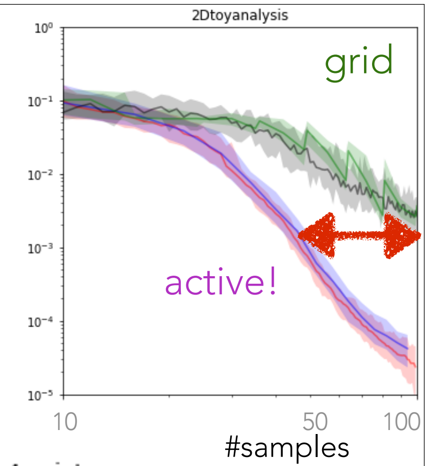
```
/home/jovyan/interpolate.py:365: UserWarning: No contour levels were found within the data range.
  c = ax.contour(xi,yi,zi, [level])
```

ACTIVE LEARNING

Instead of generating Monte Carlo a priori, generate it on demand where it is relevant!

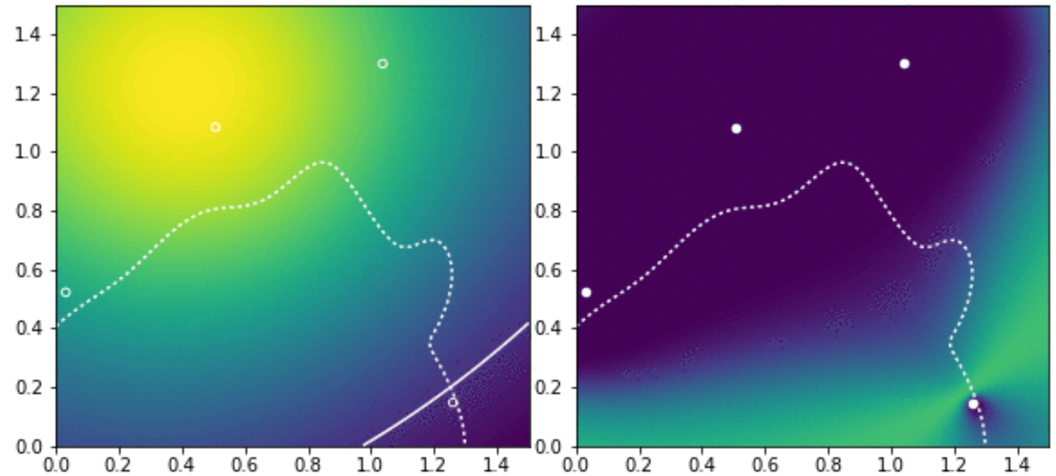
↓ An algorithm for finding exclusion contours

Drastically more efficient use of computing resources →



K.C., Lukas Heinrich, Gilles Louppe, ACAT2019

4 points



Scalable cyberinfrastructure applications

Team: J. Brehmer[1 2], K. Cranmer[1 2], **Irina Espejo**[1], S. Macaluso[1 2] and H. Müller[1]
Institutions: [1] Center for Data Science, New York University
[2] Department of Physics, New York University

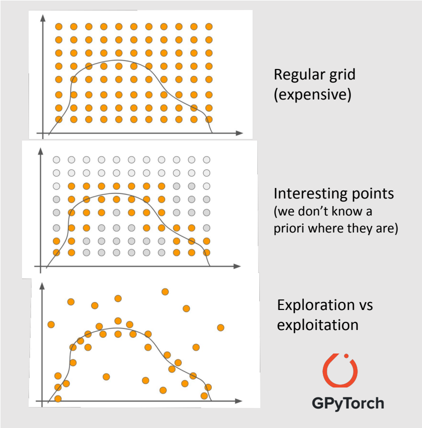
The SCAILFIN Project

Excursion

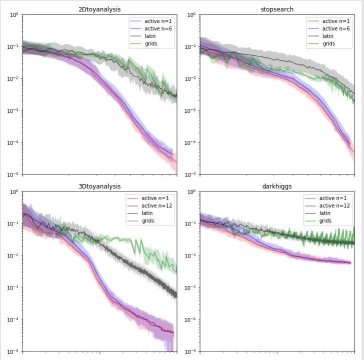
in collaboration with G. Louppe[2] and L. Heinrich[3]
[2]University of Liège, [3]CERN

diana-hep/excursion
irinaespejo/excursion

- Goal is to find *level sets of black-box functions* that are expensive to evaluate. Examples: test statistics from complex simulations.
- Evaluate the black blox function at *interesting points only* instead of evaluating at whole regular grid. We use a *Gaussian process* to: interpolate between samples and model uncertainty in the knowledge of the black box function.
The *acquisition function* regulates the exploration vs exploitation tradeoff. Select one that *minimizes global uncertainty* of the location of the excursion set.
- Future: efforts will focus on *scalability* wrt the dimensionality of the function domain space. Example, likelihood ratio as function of mass, charge, spin,...



… Is it all worth it? Yes! [3]



[3] L. Heinrich, G. Louppe, K. Cranmer, *Excursion Set Estimation using Sequential Entropy Reduction for Efficient Searches for New Physics at the LHC*, ACAT 2019

README.md

https://indico.cern.ch/event/708041/contributions/3269754/

`excursion` — Efficient Excursion Set Estimation

DOI 10.5281/zenodo.1634427 launch binder build passing

This package implements a Bayesian Optimization procedure based on Gaussian Processes to efficiently determine excursion sets (or equivalently iso-surfaces) of one or many expensive black-box functions.

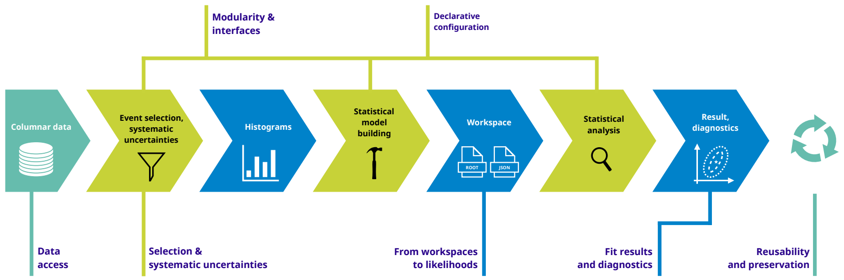
Installation and Example

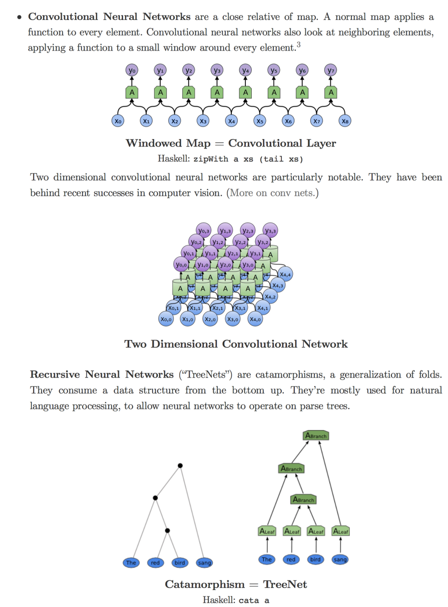
Install via `pip install excursion==0.0.1a0`.

DIFFERENTIABLE PROGRAMING

Automatic differentiation is not just for Machine Learning!

- Differentiable Programming
- **Attitude**: we can auto-diff through analysis and reconstruction
- End-to-end optimization





- **Convolutional Neural Networks** are a close relative of map. A normal map applies a function to every element. Convolutional neural networks also look at neighboring elements, applying a function to a small window around every element.[3]

Windowed Map = Convolutional Layer

Haskell: `zipWith a xs (tail xs)`

Two dimensional convolutional neural networks are particularly notable. They have been behind recent successes in computer vision. (More on conv nets.)

Two Dimensional Convolutional Network

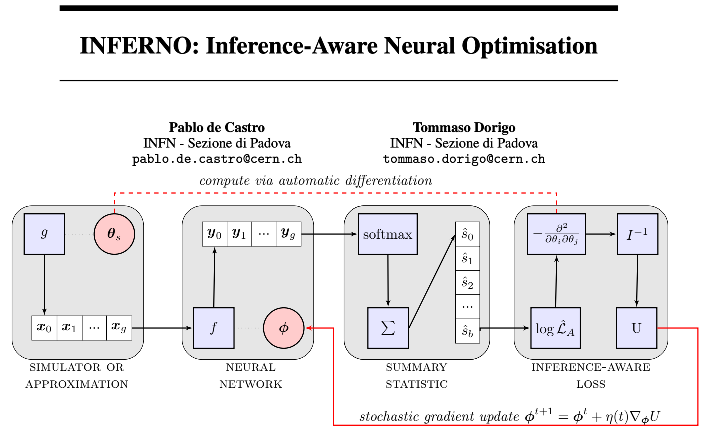
Recursive Neural Networks ("TreeNets") are catamorphisms, a generalization of folds. They consume a data structure from the bottom up. They're mostly used for natural language processing, to allow neural networks to operate on parse trees.

Catamorphism = TreeNet

Haskell: `cata a`

http://colah.github.io/posts/2015-09-NN-Types-FP/

END-TO-END OPTIMIZATION WITH AUTODIFF



Kyle Cranmer @KyleCranmer · 19h

Take note! Here is a nice example of differentiable programming. It shows end-to-end optimization of a NN for event categorization wrt. final statistical analysis (using pyhf). Requires running gradients through results of maximum likelihood with fixed-point differentiation 👏

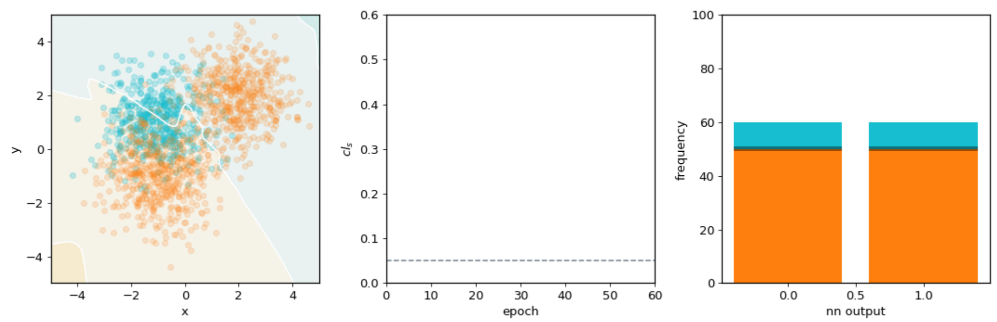
Nathan Simpson @ CERN @phi_nate

I'm *very* excited to share with you what I've been working on recently in collaboration with @lukasheinrich_ !

We've developed a module that performs end-to-end learning with respect to statistical inference in particle physics.

try it yourself at github.com/pyhf/neos! :)

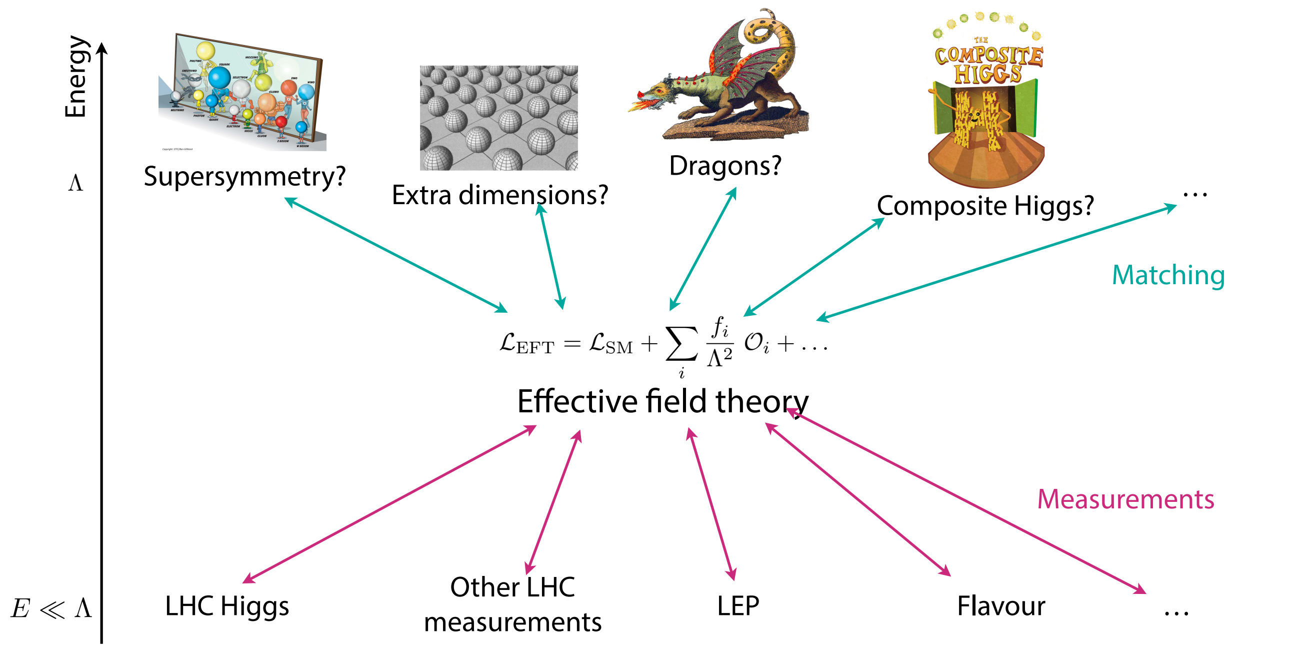
Backpropagate: dLimit /dSelection including full statistical treatment with systematics.



https://github.com/pyhf/neos

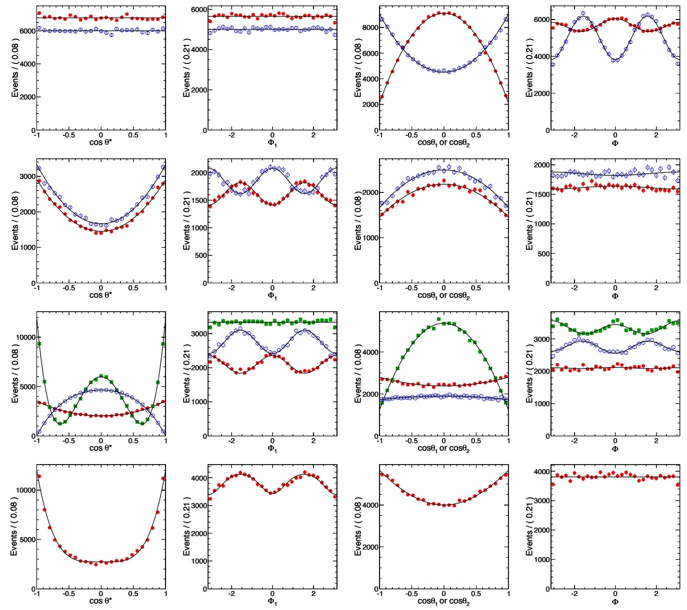
Effective field theory

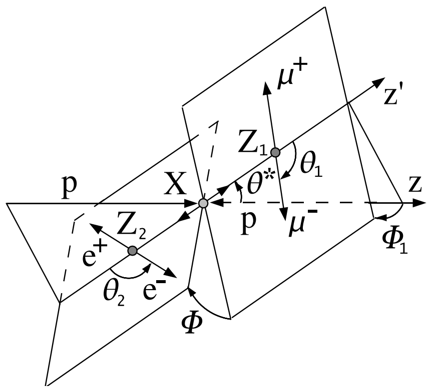
[STFC / Ben Gililand, Sean Carroll, Friedrich Justin Bertuch 1806, symmetry]

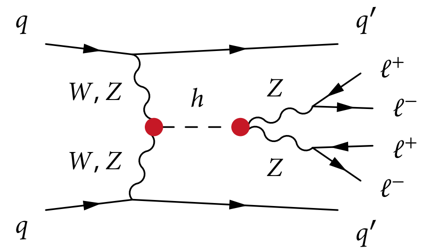


When looking for deviations from the standard model Higgs, we would like to look at all sorts of kinematic correlations

- thus each observation **x** is high-dimensional

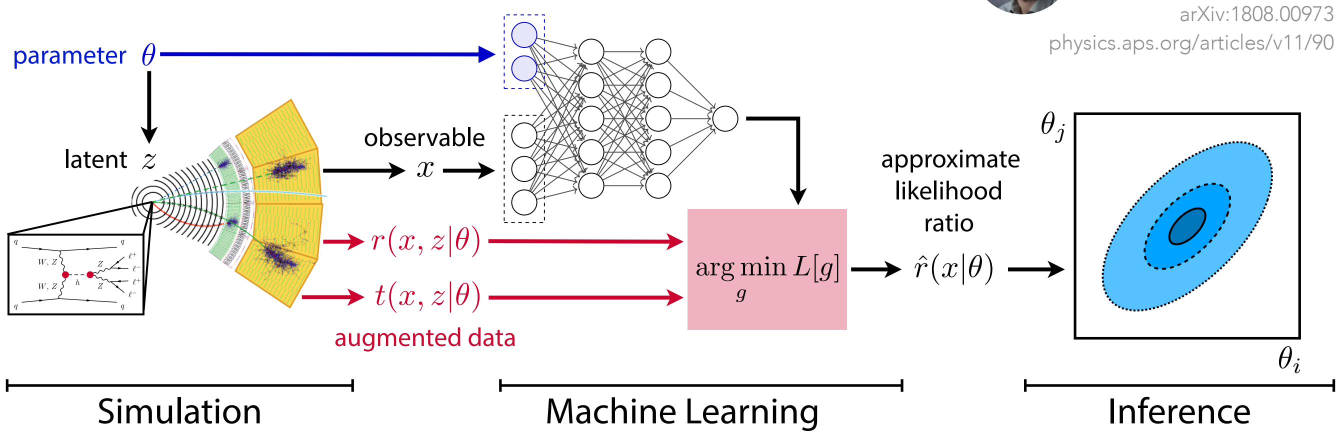


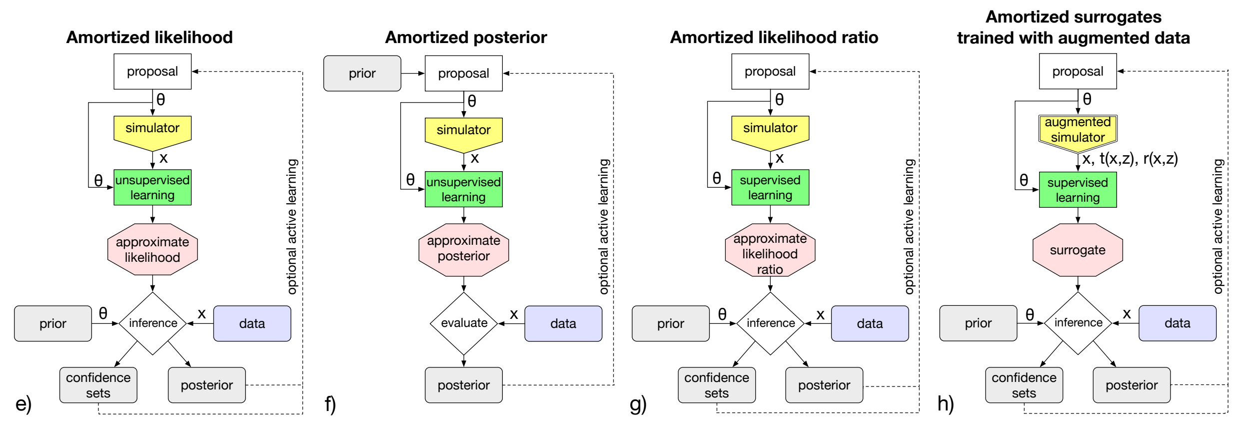




LEARNING THE LIKELIHOOD RATIO

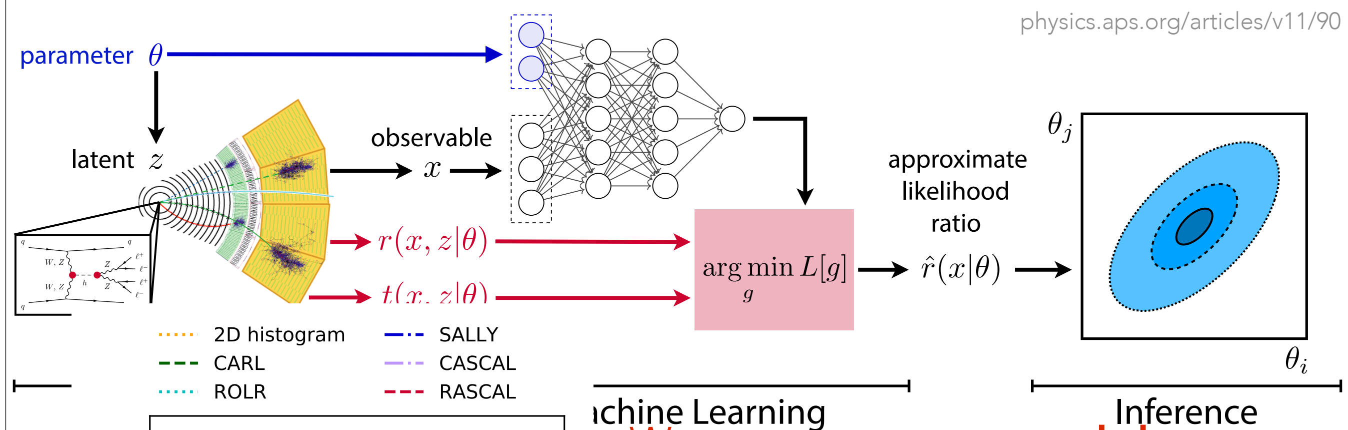
arXiv:1805.12244
PRL, arXiv:1805.00013
PRD, arXiv:1805.00020
arXiv:1808.00973
physics.aps.org/articles/v11/90



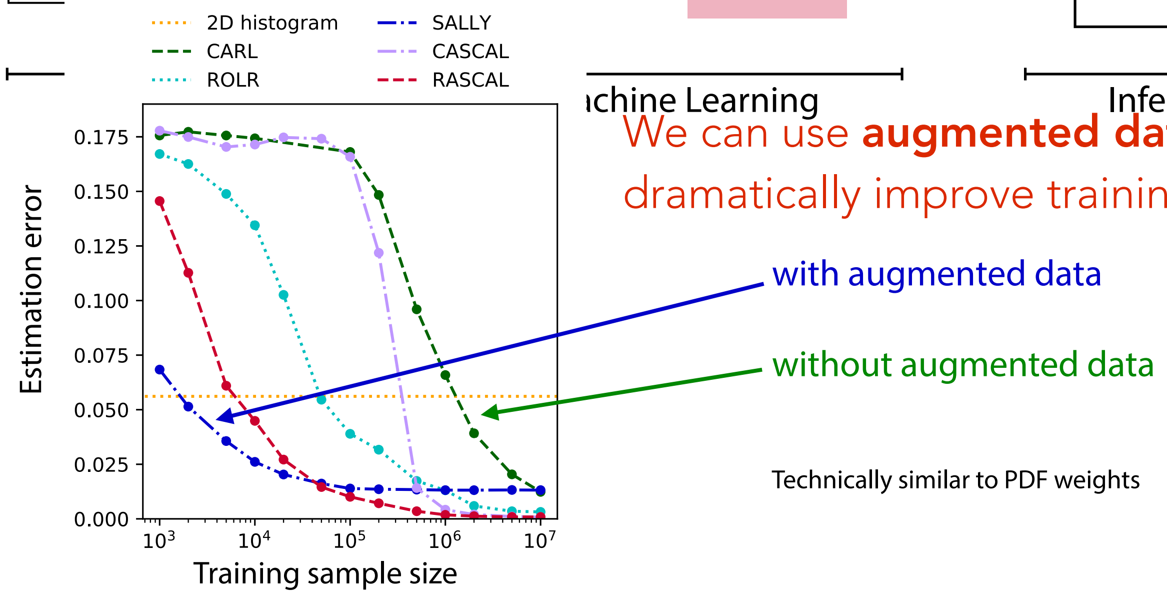


LIKELIHOOD-FREE INFERENCE

arXiv:1805.12244
PRL, arXiv:1805.00013
PRD, arXiv:1805.00020
physics.aps.org/articles/v11/90



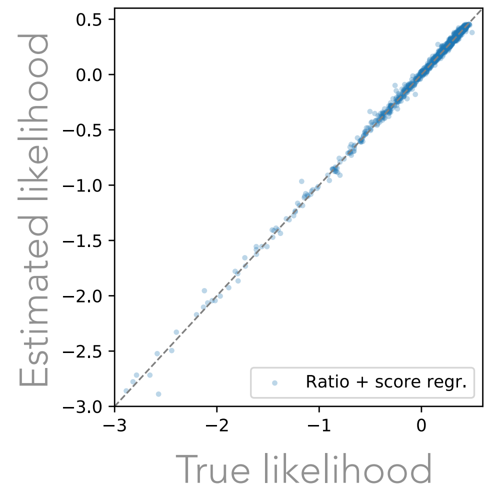
We can use **augmented data** to dramatically improve training

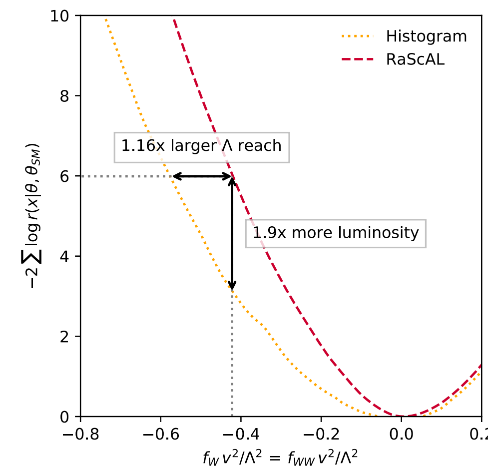


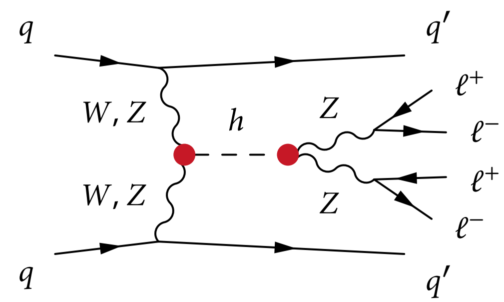
Technically similar to PDF weights

IMPACT ON STUDIES OF THE HIGGS BOSON

(based on a 42-Dim observation **x**)





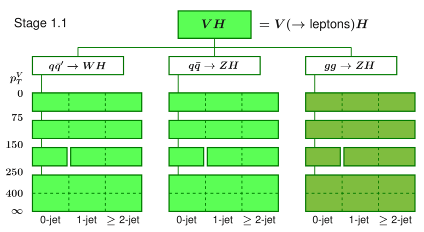


J Brehmer, J Pavez, G Louppe, K.C. PRL & PRD 2018 [arXiv:1805.00013 & arXiv:1805.00020]
"Better Higgs Measurements Through Information Geometry" [arXiv:1612.05261] & CARL [arxiv:1506.02169]

BENCHMARKING STXS IN WH

[JB, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

- Simplified Template Cross-Sections (STXS) define observable bins that are supposed to capture as much information on NP as possible
  [N. Berger et al. 1906.02754; HXSWG YR4]

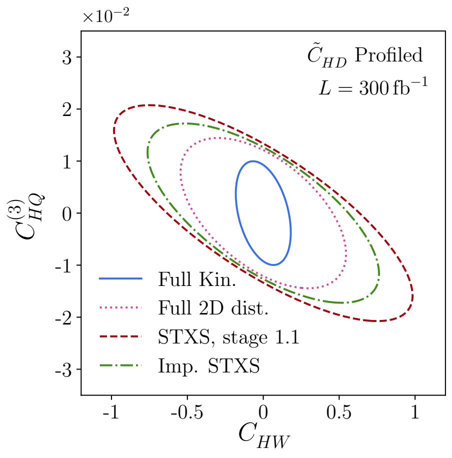


- Let's check! How much information on

$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\Box} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger\phi)\Box(\phi^\dagger\phi) - \frac{1}{4}(\phi^\dagger D^\mu\phi)^*(\phi^\dagger D_\mu\phi)$$
$$\mathcal{O}_{HW} = \phi^\dagger\phi W^a_{\mu\nu}W^{\mu\nu a}$$
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu\phi)(\overline{Q}_L\sigma^a\gamma^\mu Q_L)\,,$$

can we extract from $pp \to WH \to \ell\nu\, b\bar{b}$?

- Results: STXS are indeed sensitive to operators, adding a few more bins improve them, but a multivariate analysis is still stronger



HANDS-ON TUTORIAL AT DESY LAST WEEK

We accomplished a lot!

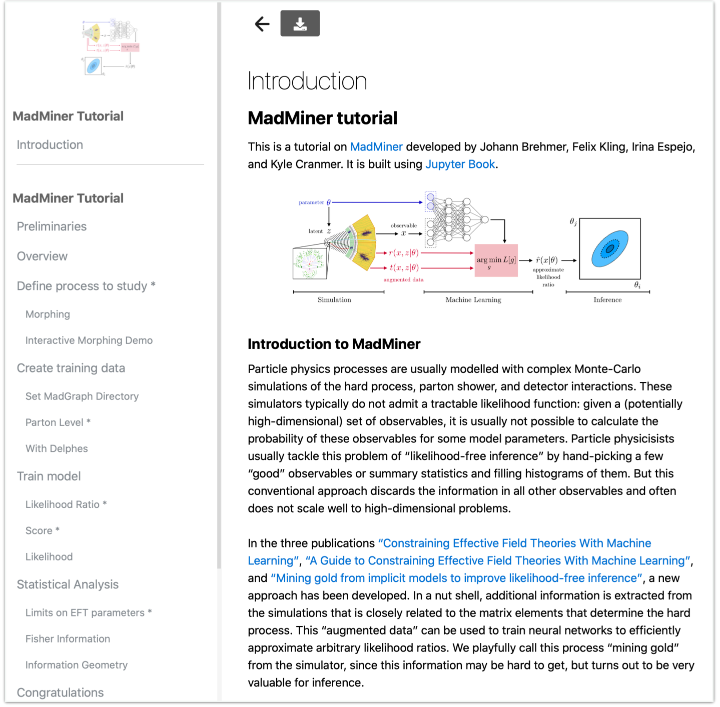
From scratch:

- Generate simulated data for EFT with MadGraph
- Fast detector simulation
- Trained neural network to learn likelihood ratio
- Trained neural network to learn Score (Optimal Observable)
- Calculated expected limit for both approaches and compared to simple 1-d histogram approach
- Calculated Fisher information matrix

This is workflow for several published papers

- To speed this up, working to streamline MadMiner with REANA

https://cranmer.github.io/madminer-tutorial/


MadMiner Tutorial
Introduction

MadMiner Tutorial
Preliminaries
Overview
Define process to study *
Morphing
Interactive Morphing Demo
Create training data
Set MadGraph Directory
Parton Level *
With Delphes
Train model
Likelihood Ratio *
Score *
Likelihood
Statistical Analysis
Limits on EFT parameters *
Fisher Information
Information Geometry
Congratulations

Introduction

MadMiner tutorial

This is a tutorial on MadMiner developed by Johann Brehmer, Felix Kling, Irina Espejo, and Kyle Cranmer. It is built using Jupyter Book.

Introduction to MadMiner

Particle physics processes are usually modelled with complex Monte-Carlo simulations of the hard process, parton shower, and detector interactions. These simulators typically do not admit a tractable likelihood function: given a (potentially high-dimensional) set of observables, it is usually not possible to calculate the probability of these observables for some model parameters. Particle physicisists usually tackle this problem of "likelihood-free inference" by hand-picking a few "good" observables or summary statistics and filling histograms of them. But this conventional approach discards the information in all other observables and often does not scale well to high-dimensional problems.

In the three publications "Constraining Effective Field Theories With Machine Learning", "A Guide to Constraining Effective Field Theories With Machine Learning", and "Mining gold from implicit models to improve likelihood-free inference", a new approach has been developed. In a nut shell, additional information is extracted from the simulations that is closely related to the matrix elements that determine the hard process. This "augmented data" can be used to train neural networks to efficiently approximate arbitrary likelihood ratios. We playfully call this process "mining gold" from the simulator, since this information may be hard to get, but turns out to be very valuable for inference.

Scalable cyberinfrastructure applications

Team: J. Brehmer[1 2], K. Cranmer[1 2], **Irina Espejo**[1], S. Macaluso[1 2] and H. Müller[1]
Institutions: [1] Center for Data Science, New York University
[2] Department of Physics, New York University

The SCAILFIN Project

MadMiner on REANA

Simulators provide a *high-fidelity description* of phenomena. If they are too complex the likelihood is intractable leading to *poor statistical analysis*

⇩

MadMiner [1], [2] uses: advances in *Neural Networks* + latent simulator information to estimate the likelihood ratio

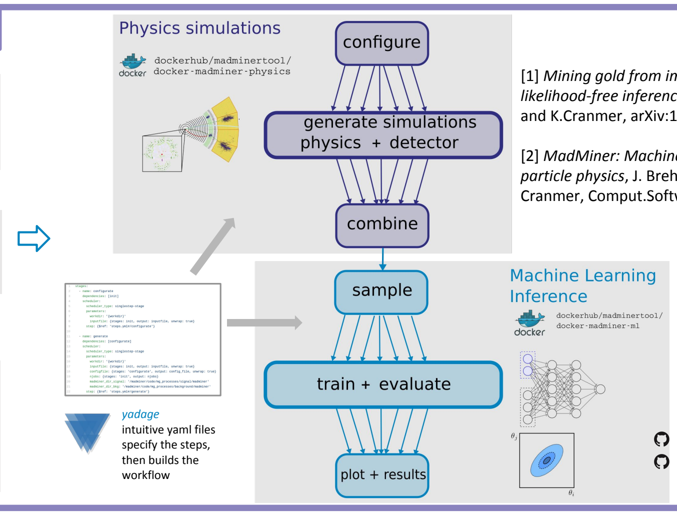
⇧

To deploy MadMiner in a scalable way we need:

- ✓ containerization
- ✓ parallelization
- ✓ reusability

reana

⇨



[1] *Mining gold from implicit models to improve likelihood-free inference*, J.Brehmer, G.Louppe, J.Pavez and K.Cranmer, arXiv:1805.12244 (2018)

[2] *MadMiner: Machine learning-based inference for particle physics*, J. Brehmer, F. Kling, I. Espejo and K. Cranmer, Comput.Softw.Big Sci. 4 3 (2020)

scailfin/workflow-madminer
dianahep/madminer

70 TB Dataset

OpenStack Magnum

25000 **Kubernetes Jobs**

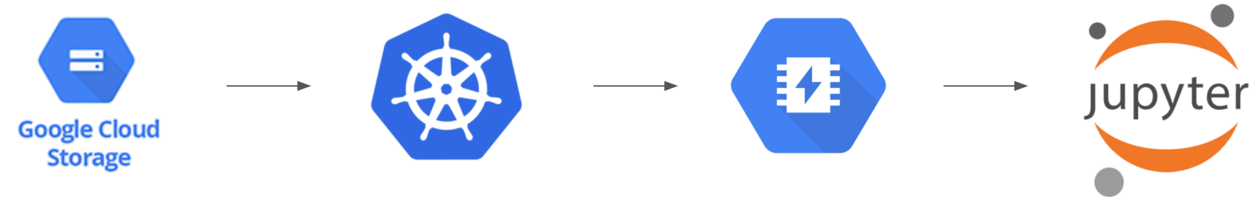
Job Results

Interactive Visualization

Aggregation

CERN openlab

Google Cloud



70 TB Dataset

Cluster on GKE

Max **25000 Cores**

Single Region, 3 Zones

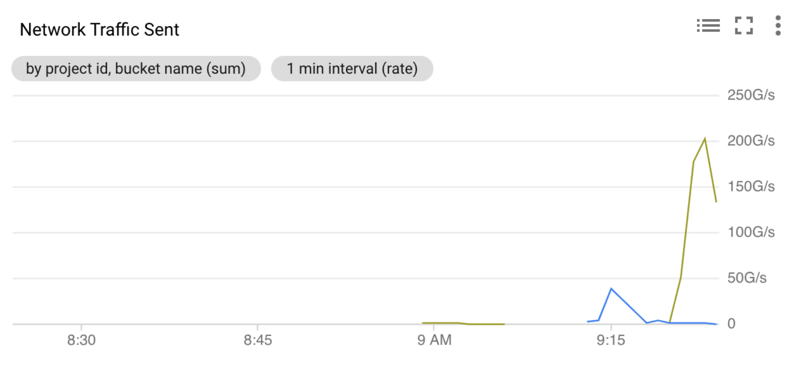
25000 **Kubernetes Jobs**

Job Results

Interactive Visualization

Aggregation

- **HEP analysis on Cloud w/ standard ingredients is possible**
  - **Terabit per second throughput for analysis.**



- **All using s is achievable for anyone**
  - **O(10k)/analysis nodes not an issue on short notice**
  - **haven't pushed limits yet.**

SHIFTING FROM REPRODUCIBILITY TO REUSE

nature physics
PERSPECTIVE
https://doi.org/10.1038/s41567-018-0342-2
Corrected: Publisher Correction
OPEN

Open is not enough

Xiaoli Chen[1,2], Sünje Dallmeier-Tiessen[1*], Robin Dasler[1,11], Sebastian Feger[1,3], Pamfilos Fokianos[1], Jose Benito Gonzalez[1], Harri Hirvonsalo[1,4,12], Dinos Kousidis[1], Artemis Lavasa[1], Salvatore Mele[1], Diego Rodriguez Rodriguez[1], Tibor Šimko[1*], Tim Smith[1], Ana Trisovic[1,5*], Anna Trzcinska[1], Ioannis Tsanaktsidis[1], Markus Zimmermann[1], Kyle Cranmer[6], Lukas Heinrich[6], Gordon Watts[7], Michael Hildreth[8], Lara Lloret Iglesias[9], Kati Lassila-Perini[4] and Sebastian Neubert[10]

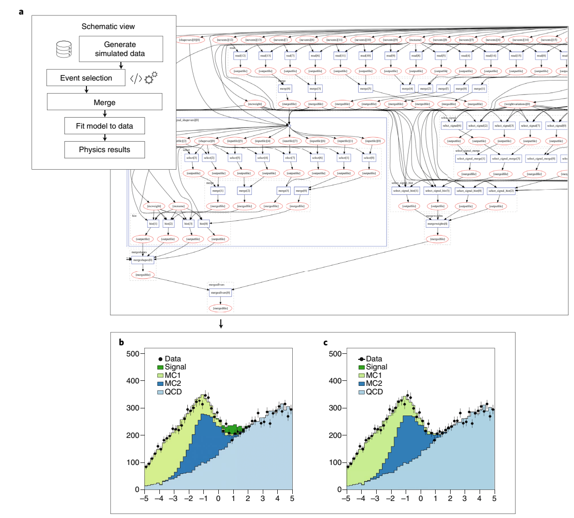
The solutions adopted by the high-energy physics community to foster reproducible research are examples of best practices that could be embraced more widely. This first experience suggests that reproducibility requires going beyond openness.

reana

Reproducible research data analysis platform

Flexible	Scalable	Reusable	Free
Run many computational workflow engines.	Support for remote compute clouds.	Containerise once, reuse elsewhere. Cloud-native.	Free Software. MIT licence. Made with ❤ at CERN.

http://reanahub.io



2 | Example of a complex computational workflow on REANA mimicking a beyond the standard model (BSM) analysis . This figure shows an ıple where the experimental data is compared to the predictions of the standard model with an additional hypothesized signal component. The ıple permits one to study the complex computational workflows used in typical particle physics analyses. **a–c**, The computational workflow (**a**) may ist of several tens of thousands of computational steps that are massively parallelizable and run in a cascading 'map-reduce' style of computations stributed compute clusters. The workflow definition is modelled using the Yadage workflow specification and produces an upper limit on the ıl strength of the BSM process. A typical search for BSM physics consists of simulating a hypothetical signal process (**c**), as well as the background esses predicted by the standard model with properties consistent with the hypothetical signal (marked dark green in (**b**)). The background often ists of simulated background estimates (dark blue and light green histograms) and data-driven background estimates (light blue histogram). ıtistical model involving both signal (dark green histogram) and background components is built and fit to the observed experimental data (black ers). **b**, Results of the model in its pre-fit configuration at nominal signal strength. We can see the excess of the signal over data, meaning that the inal setting does not describe the data well. The post-fit distribution would scale down the signal in order to fit the data. This REANA example is cly available at ref. [35]. For icon credits, see Fig. 1.

https://doi.org/10.1038/s41567-018-0342-2

BUILD IT AND THEY WILL COME

In 2010 we identified a use-case with high scientific value for community

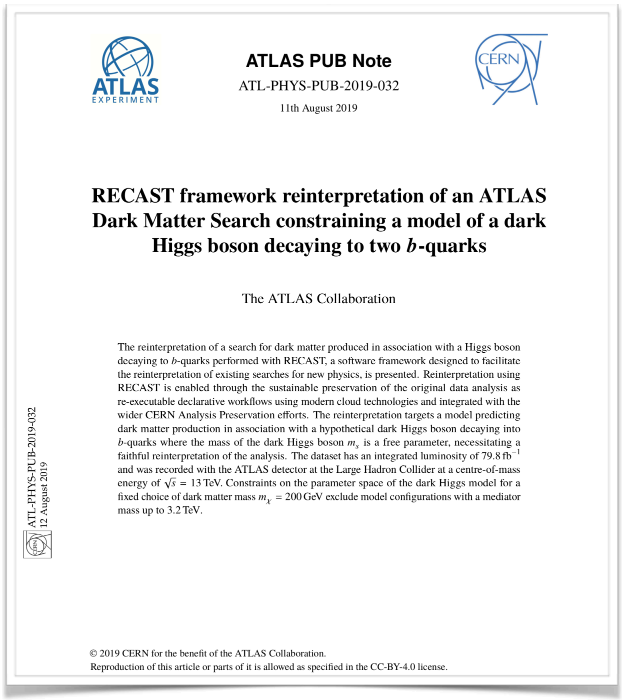
- Conservative narrative compared to "open data"
- Not conservative enough for many. Lots of resistance
- People said it couldn't be done, our workflows are too complicated
- Hard to get effort to work on it.

Got lucky with an amazing student that took a risk and just built it.

- Containers & Cloud technology
- 9 years later …

recast

Lukas Heinrich



ATLAS PUB Note
ATL-PHYS-PUB-2019-032
11th August 2019

RECAST framework reinterpretation of an ATLAS Dark Matter Search constraining a model of a dark Higgs boson decaying to two *b*-quarks

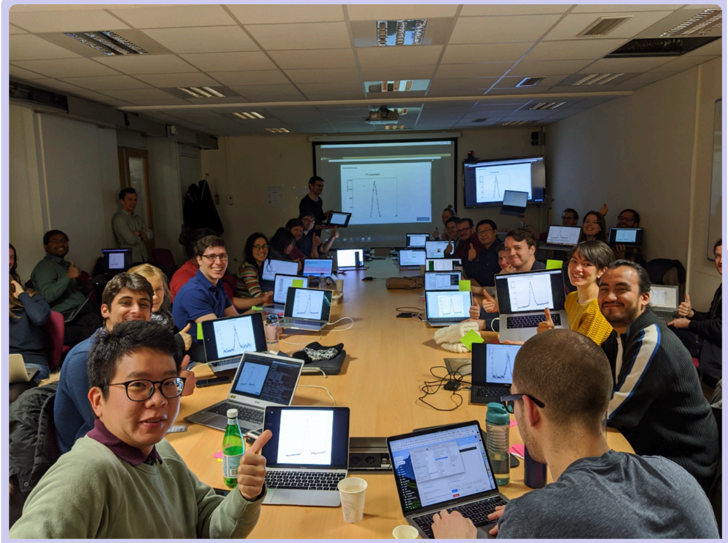
The ATLAS Collaboration

The reinterpretation of a search for dark matter produced in association with a Higgs boson decaying to *b*-quarks performed with RECAST, a software framework designed to facilitate the reinterpretation of existing searches for new physics, is presented. Reinterpretation using RECAST is enabled through the sustainable preservation of the original data analysis as re-executable declarative workflows using modern cloud technologies and integrated with the wider CERN Analysis Preservation efforts. The reinterpretation targets a model predicting dark matter production in association with a hypothetical dark Higgs boson decaying into *b*-quarks where the mass of the dark Higgs boson m_s is a free parameter, necessitating a faithful reinterpretation of the analysis. The dataset has an integrated luminosity of 79.8 fb^{-1} and was recorded with the ATLAS detector at the Large Hadron Collider at a centre-of-mass energy of $\sqrt{s} = 13$ TeV. Constraints on the parameter space of the dark Higgs model for a fixed choice of dark matter mass $m_\chi = 200$ GeV exclude model configurations with a mediator mass up to 3.2 TeV.

ATL-PHYS-PUB-2019-032
12 August 2019

Orig Proposal in 2010: [arXiv.org:1010.2506]

Analysis preservation bootcamp

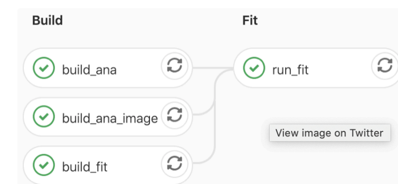


Participants in Analysis Preservation Bootcamp showing off their ability to reproduce an LHC analysis. Photo Credit: Samuel Meehan

Josh McFayden
@JoshMcFayden

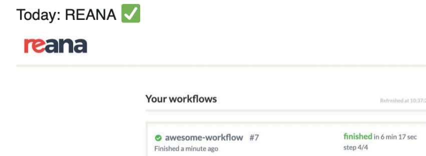
Thoroughly enjoying myself at an @iris_hep/@diana_hep analysis preservation bootcamp @CERN today!
indico.cern.ch/event/854880/o…

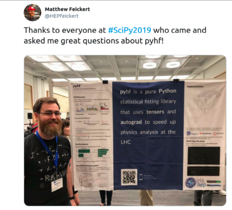
Josh McFayden @JoshMcFayden · Feb 18, 2020
Replying to @JoshMcFayden and 3 others
PROGRESS ✅



Josh McFayden
@JoshMcFayden

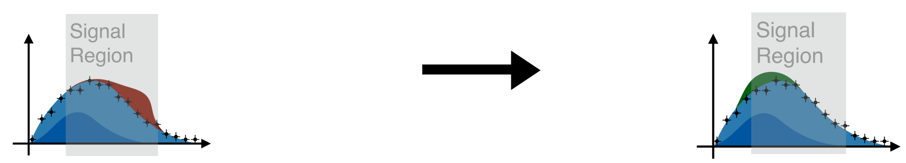
Today: REANA ✅





JSON Patch for signal model (reinterpretation)

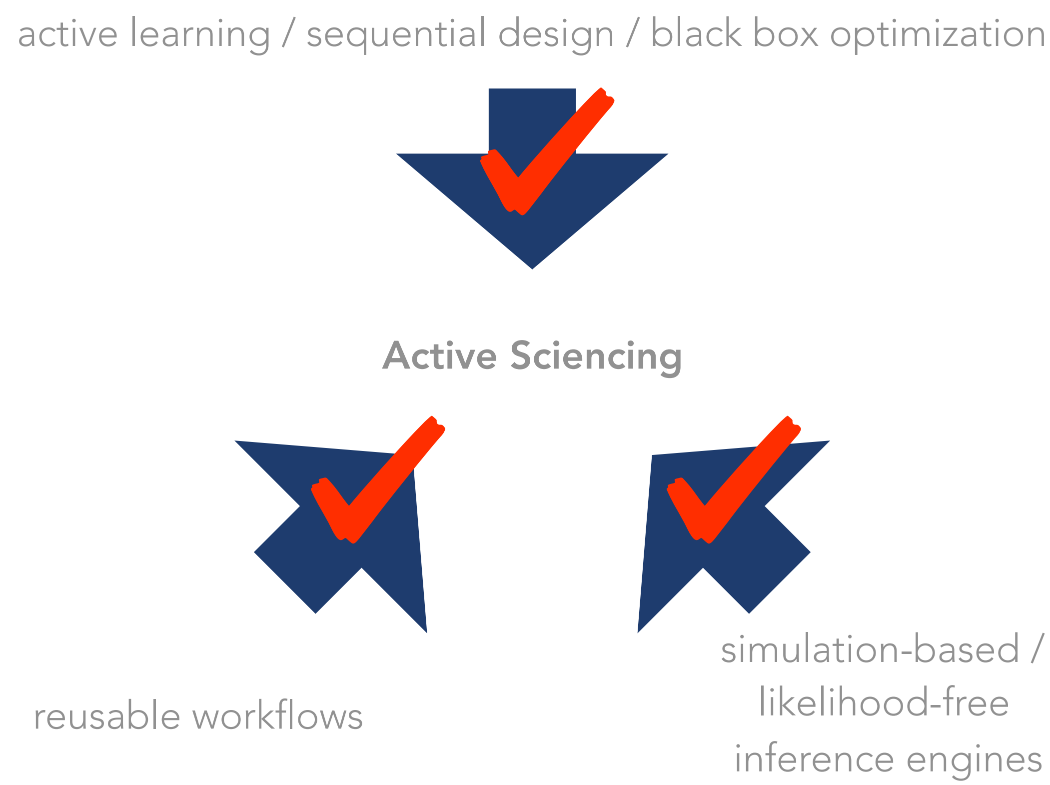
```
$ pyhf cls example.json | jq .CLs_obs
0.053994246621274014

$ cat new_signal.json
[{
    "op": "replace",
    "path": "/channels/0/samples/0/data",
    "value": [10.0, 6.0]
}]

$ pyhf cls example.json --patch new_signal.json | jq .CLs_obs
0.3536906623262466
```



Original analysis (model A)

Recast analysis (model B)

SYNTHESIS


active learning / sequential design / black box optimization
Active Sciencing
reusable workflows
simulation-based /
likelihood-free
inference engines

Scalable cyberinfrastructure applications

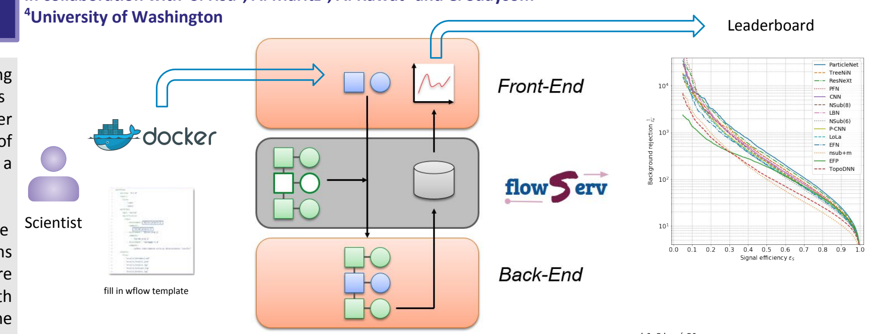
Team: J. Brehmer[1 2], K. Cranmer[1 2], **Irina Espejo**[1], S. Macaluso[1 2] and H. Müller[1]
Institutions: [1] Center for Data Science, New York University
[2] Department of Physics, New York University

The SCAILFIN Project

ROB
Reproducible Open Benchmark Platform

in collaboration with S. Hsu[4], A. Maritz[4], A. Rawat[4] and C. Suaysom[4]
[4]University of Washington

- ROB is an experimental prototype for enabling *community benchmarks* of data analysis algorithms. The goal of ROB is to allow user communities to evaluate the performance of their different data analysis algorithms in a *reproducible competition-style* format.
- The *workflow template* and input data are defined by a coordinator. The template contains placeholders for workflow steps that are implemented by the participants (e.g., with Docker containers). The backend processes the submission workflows. The user interface allows participants to *submit new runs* and to view the results.



Architecture of ROB

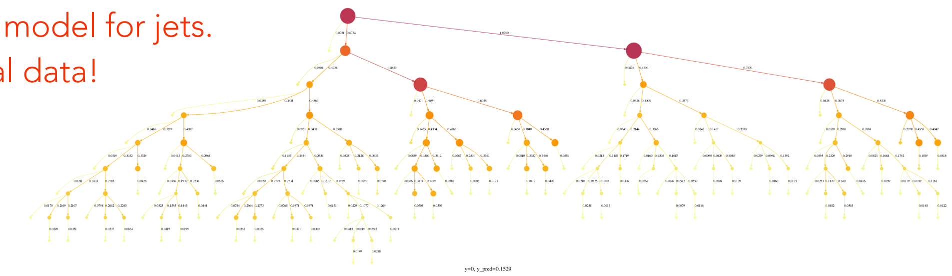
- scailfin/flowserv-core
- scailfin/rob-webapi-flask
- scailfin/rob-ui
- SebastianMacaluso/TopTagComparison

We can use the same technology to streamline comparison of up-stream tools like ML-based jet taggers.

- Building ROB & FlowServe on top of REANA

NN-BASED SIMULATION TRAINED ON DATA

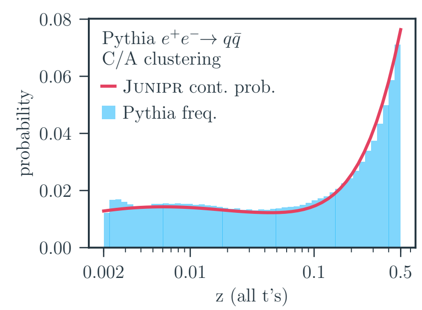
JUNIPR is a generative model for jets.
Can train on real data!

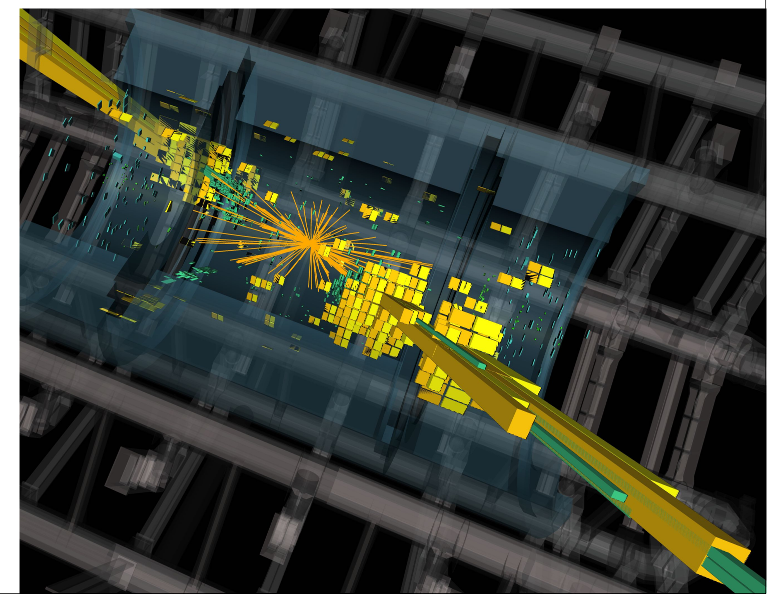


tractable likelihood

$$P_{\text{jet}}(\{p_1, \ldots p_n\}) = \left[\prod_{t=1}^{n-1} P_t\big(k_1^{(t+1)}, \ldots, k_{t+1}^{(t+1)} \big| k_1^{(t)}, \ldots, k_t^{(t)}\big)\right] \times P_n\big(\text{end}\big|k_1^{(n)}, \ldots, k_n^{(n)}\big).$$

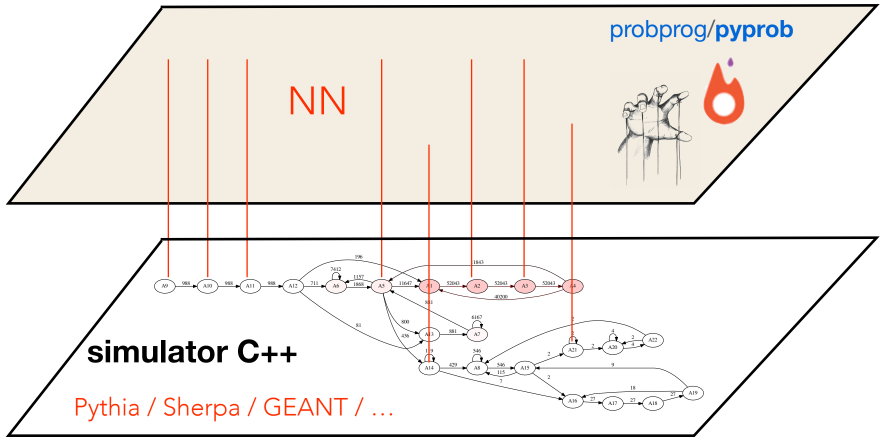
… and it is interpretable



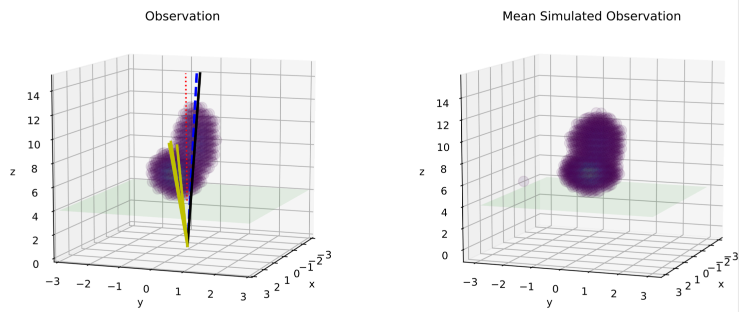


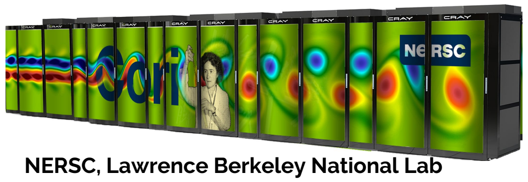
Andreassen, Feige, Frye, Schwartz arXiv:1804.09720

Idea: hijack the random number generators and use Neural Network to perform a *very* fancy type of importance sampling



- Neural Network powered inference engine (python)
- real-world scientific simulator (C++)



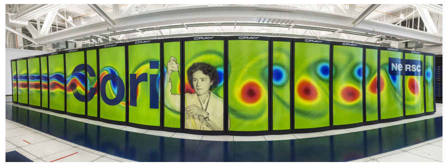


NERSC, Lawrence Berkeley National Lab

arXiv:1807.07706

Highlight

https://arxiv.org/abs/1907.03382

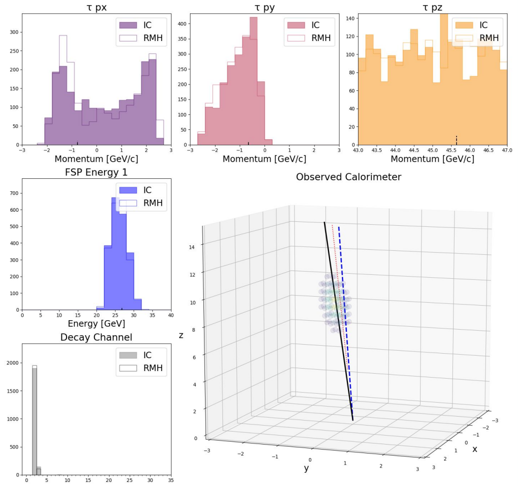


Finalist for best paper award at SC19 (Super Computing)

- Largest scale Bayesian inference ever using in a universal probabilistic programming language
  - ***Applied to complex LHC Physics use case: Sherpa code base of ~1M lines of code in C++***

- 230x speedup for synchronous data parallel training of a 3DCNN-LSTM neural network
  - ***1,024 nodes (32,768 CPU cores)***
  - ***128k minibatch size, largest for this NN architecture***
  - ***One of the largest-scale use of PyTorch built-in MPI***

- Novel protocol (PPX) to execute & control existing, large-scale, scientific simulator code bases



SUMMARY

Accelerating analysis design

- more powerful observables
- end-to-end optimization
- benchmarking of algorithms

Accelerating fitting

- pyhf and a fitting service

More efficient simulation

- excursion
- Probabilistic programming

Extending impact of results

- RECAST

Core technologies:

- automatic differentiation
- GPUs & TPUs
- Cloud-native : docker, kubernetes
- Workflows & REANA
- Functions as a service Accelerating analysis design

Sorry if this is a little disorganized, COVID has complicated work life