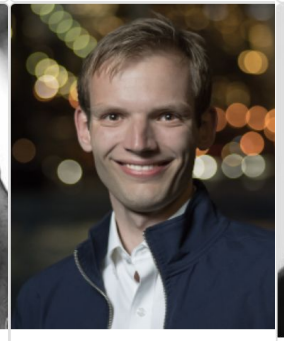
IRIS/HEP Steering Board: Analysis Systems

Kyle Cranmer (NYU)

Analysis Systems Team

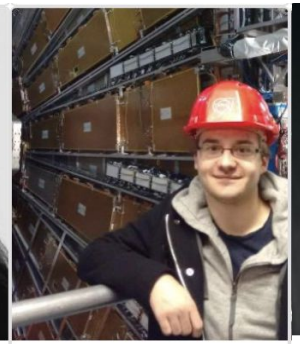
Institutions: NYU, Washington, Princeton, Cincinnati, Illinois

Kyle Cranmer
New York University



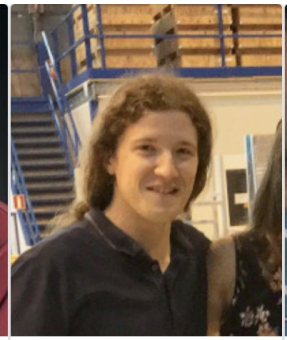
Johann Brehmer
New York University

Irina Espejo
New York University



Alexander Held
New York University

Gordon Watts
University of Washington

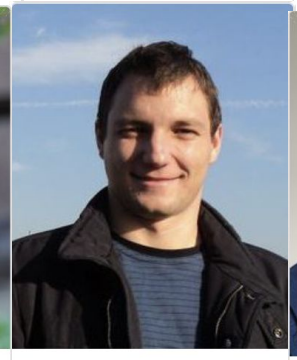


Mason Proffitt
University of Washington

Emma Torro
University of Washington

Ianna Osborne
Princeton University

Jim Pivarski
Princeton University



Vassil Vassilev
Princeton University

Henry Schreiner
Princeton University

Mike Sokoloff
University of Cincinnati

Ben Galwesky
National Center for Supercomputing Applications

Mark Neubauer
University of Illinois at Urbana-Champaign

Daniel S. Katz
University of Illinois at Urbana-Champaign

Matthew Feickert
University of Illinois at Urbana-Champaign

Prior to IRIS-HEP

Bulk Data Processing

Reconstruction Algorithms

Analysis Code

Analysis code in HEP is often more free-form with less organized development:

- one-off approach limits functionality
- slow iteration cycle
- slow on-boarding and lack of interoperability
- difficult to reproduce and reuse

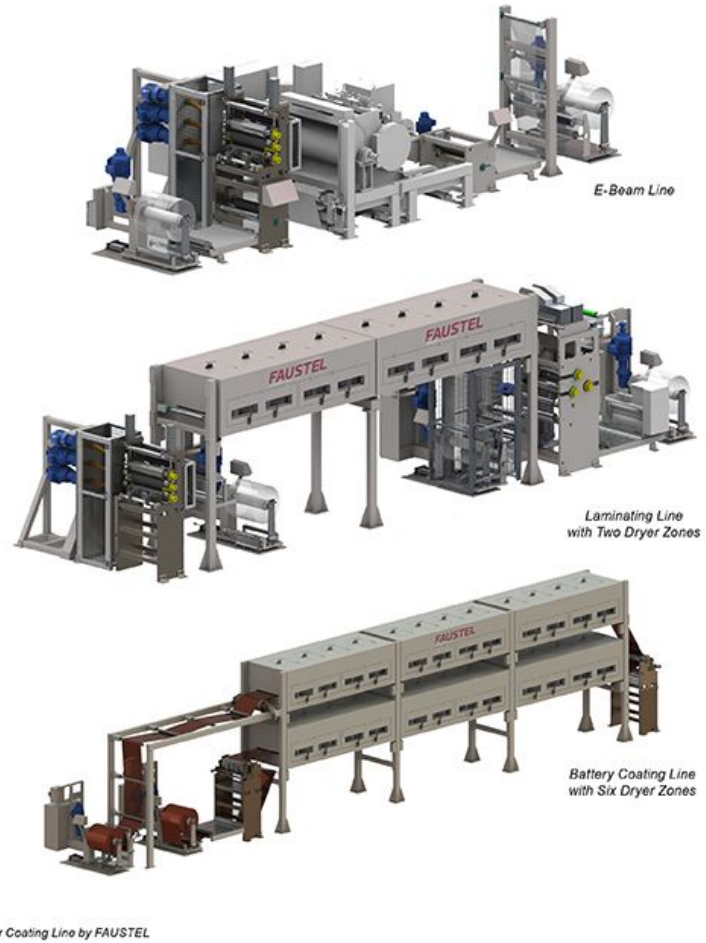
- primarily ROOT & C++
- lack of developer community
- overlapping solutions
- data redundancy

Analysis Systems

ad hoc analysis code

IRIS-HEP

Analysis Systems

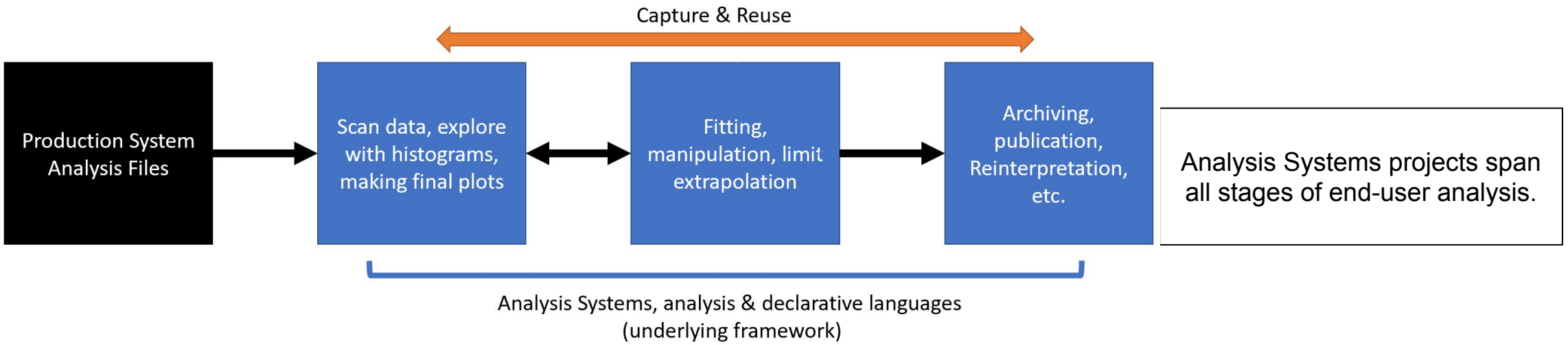


Analysis Systems strategies:

- improve functionality & interoperability
- more modular, less dependence on ROOT
- declarative: focus on what to do not how to do it
- align with modern data science practices

Analysis Systems

- Develop sustainable analysis tools to extend the physics reach of the HL-LHC experiments
  - *create greater functionality to enable new techniques,*
  - *reducing time-to-insight and physics,*
  - *lowering the barriers for smaller teams, and*
  - *streamlining analysis preservation, reproducibility, and reuse.*



Projects

- Analysis systems are connected to analysis use cases
- Systems are composed of components
- Most of these projects refer to those components
  - *many projects include people beyond IRIS-HEP*
- Milestones and activities mainly oriented towards integration, evaluation, with a global overview of the vertical slice

ADL Benchmarks

Functionality benchmarks for analysis description languages

More information

AmpGen

Generation and fitting for multibody hadron decays

More information

Awkward Array

Manipulate arrays of complex data structures

More information

DecayLanguage

Describe and convert particle decays

More information

Functional ADL

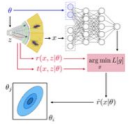
Functional Analysis Description Language

More information

Histogram projects

Histogramming efforts

More information



MadMiner

Likelihood-free Inference

More information

Particle

Pythonic particle information

More information

ROOT on Conda Forge

Use ROOT in Conda through Conda-Forge

More information

Scikit-HEP

pythonic analysis tools

More information

awesome-hep

A curated list of awesome high energy and particle physics software

More information

exploratory-ml

Analysis Reinterpretation

More information

ppx

cross-platform Probabilistic Programming eXecution protocol

More information

pyhf

Differentiable likelihoods

More information

recast

Analysis Reinterpretation

More information

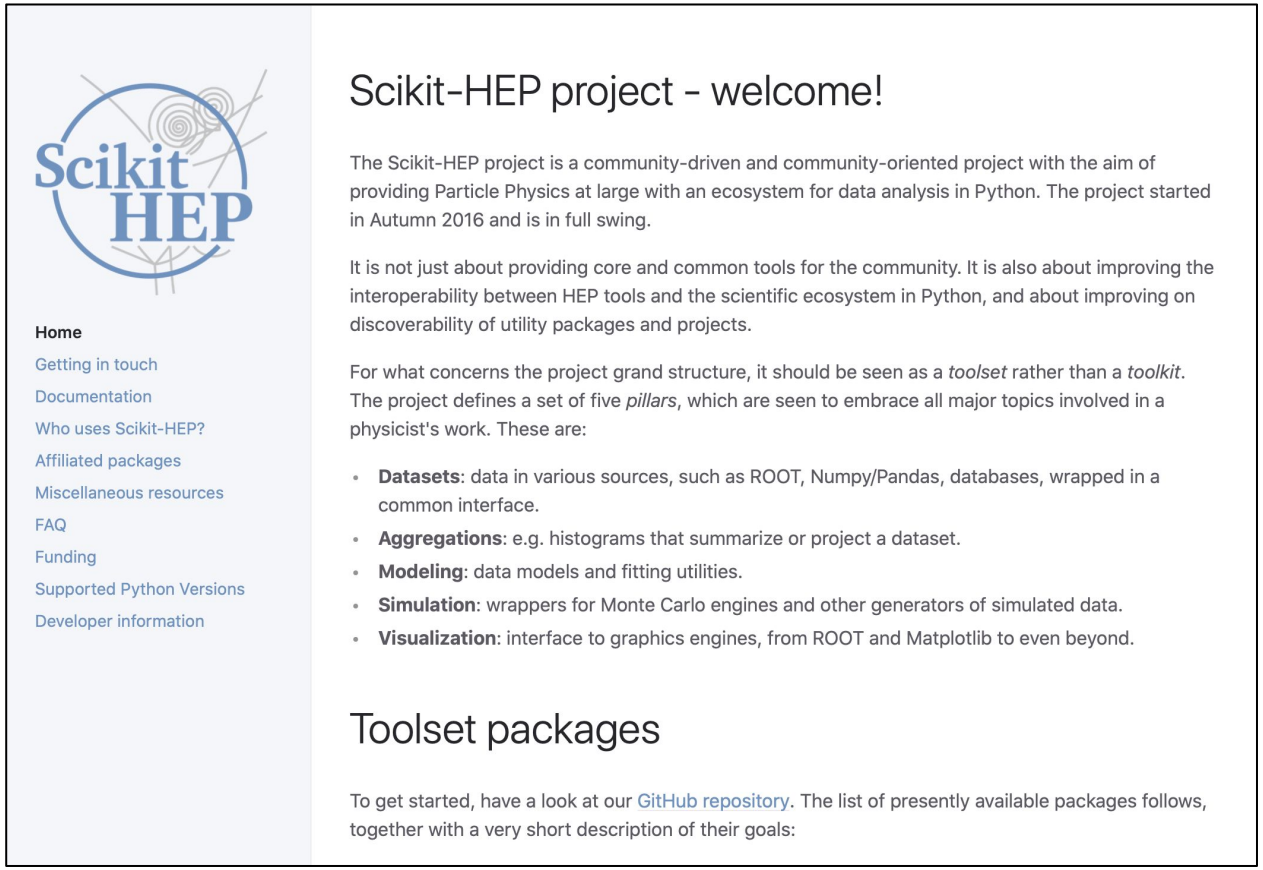
uproot

Read and write ROOT files in Python

More information

Scikit-HEP

A broad community project with heavy IRIS-HEP involvement.



Scikit HEP

Home
Getting in touch
Documentation
Who uses Scikit-HEP?
Affiliated packages
Miscellaneous resources
FAQ
Funding
Supported Python Versions
Developer information

Scikit-HEP project - welcome!

The Scikit-HEP project is a community-driven and community-oriented project with the aim of providing Particle Physics at large with an ecosystem for data analysis in Python. The project started in Autumn 2016 and is in full swing.

It is not just about providing core and common tools for the community. It is also about improving the interoperability between HEP tools and the scientific ecosystem in Python, and about improving on discoverability of utility packages and projects.

For what concerns the project grand structure, it should be seen as a *toolset* rather than a *toolkit*. The project defines a set of five *pillars*, which are seen to embrace all major topics involved in a physicist's work. These are:

- **Datasets**: data in various sources, such as ROOT, Numpy/Pandas, databases, wrapped in a common interface.
- **Aggregations**: e.g. histograms that summarize or project a dataset.
- **Modeling**: data models and fitting utilities.
- **Simulation**: wrappers for Monte Carlo engines and other generators of simulated data.
- **Visualization**: interface to graphics engines, from ROOT and Matplotlib to even beyond.

Toolset packages

To get started, have a look at our GitHub repository. The list of presently available packages follows, together with a very short description of their goals:

Basics:

Awkward Array

awkward-array : Manipulate arrays of complex data structures as easily as Numpy.
pypi v0.12.20 conda-forge v0.12.20

hepunits : Units and constants in the HEP system of units.
pypi v1.1.1

Data manipulation and interoperability:

formulate : Easy conversions between different styles of expressions.
pypi v0.0.8

root_numpy : Interface between ROOT and NumPy.
pypi v4.8.0 conda-forge v4.8.0

root_pandas : Module for conveniently loading/saving ROOT files as pandas DataFrames.
pypi v0.7.0 conda-forge v0.7.0

uproot

uproot : Minimalist ROOT I/O in pure Python and Numpy.
pypi v3.11.3 conda-forge v3.11.3

uproot-methods : Pythonic behaviours for non-I/O related ROOT classes.
pypi v0.7.3 conda-forge v0.7.3

Histogramming:

aghast

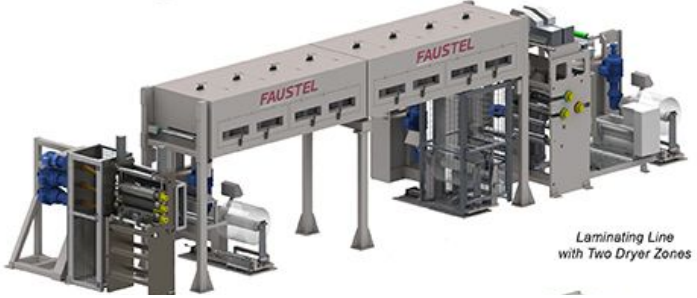
aghast : Convert between histogram representations
pypi v0.2.1 conda-forge v0.2.1

Boost Histogram

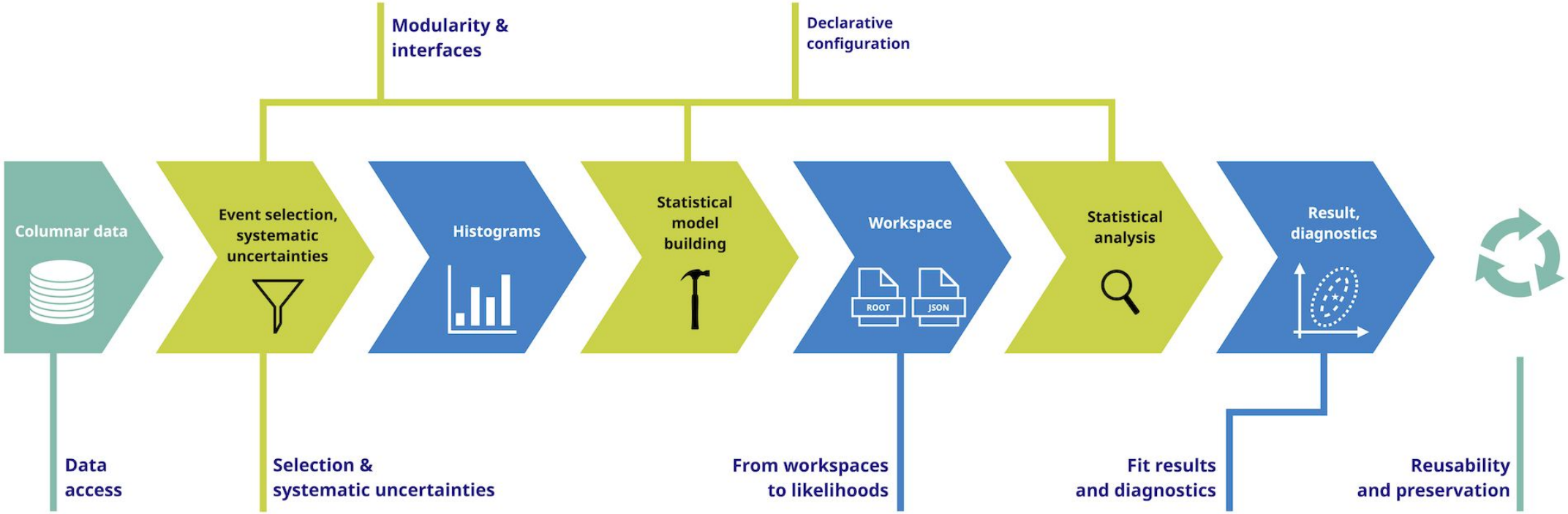
boost-histogram : Python bindings for the C++14 Boost::Histogram library.
pypi v0.6.2 conda-forge v0.6.2

NSF

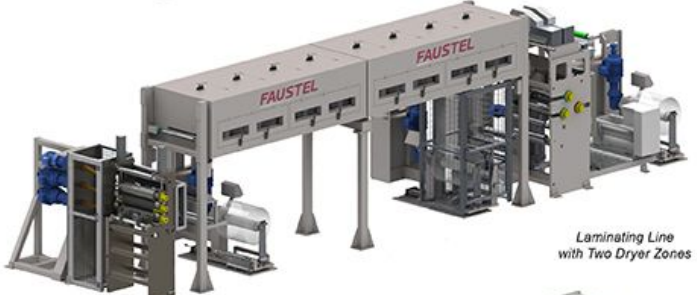
A coherent ecosystem

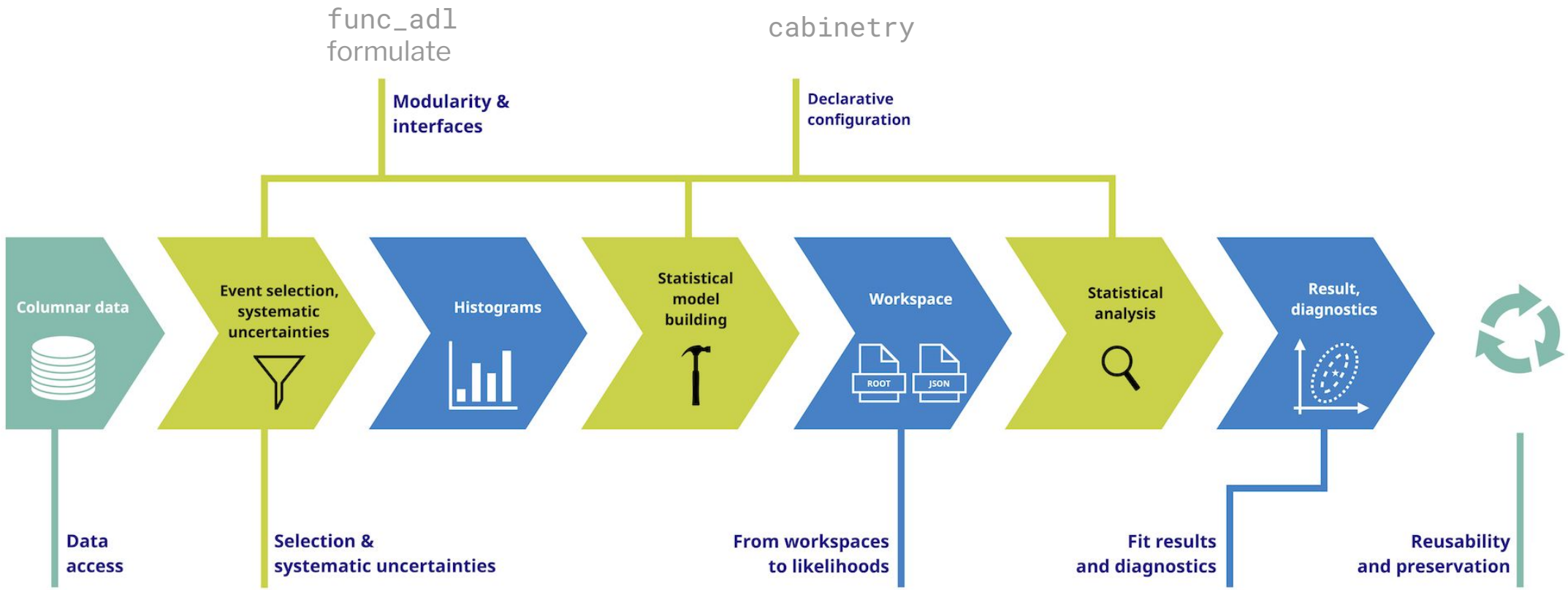


One of our analysis use cases involves a vertical slice from ServiceX to final limits for a real-world ATLAS Higgs analysis. See Alex Held's poster.



A coherent ecosystem





ServiceX
uproot
Awkward Array
coffea
Boost Histogram
pyhf differentiable likelihoods
yadage
recast

Y3 Themes

Integration:

- Many of the individual tools are at beta stage or better.
- Increase our efforts towards integrating tools into systems (vertical slices)
  - *this is expected to expose areas where tools can be improved, modified, etc.*
- More explicit coordination and planning with Analysis Facility / DOMA / SSL
- An important step towards almost any Grand Challenge involving Analysis Systems

Adoption:

- Some of our tools and projects are at “tipping point,” rapidly gaining traction within experiments and in user communities.
  - *Example: pyhf adoption is rapid (papers, likelihood publishing, etc.)*
  - *Example: ATLAS is ramping up RECAST efforts (papers to come near end Y3)*
  - *Example: Scikit-hep as an example of community-driven software effort*
- Good to invest effort in these areas for results and to build IRIS-HEP reputation
  - *Development, Training, Documentation*
  - *Misc. experiment specific contributions also valuable for ”delivery to experiments”*

Fellows & GSOC

IRIS-HEP Fellows and Google Summer of Code student have made significant progress on a GPU backend for Awkward1.

Anish Biswas · 3rd

GSoC'20 with CERN | Planning Head at Project MANAS

Udupi, Karnataka, India · 500+ connections · **Contact info**

Pratyush (Reik) Das

Institute of Engineering & Management (Kolkata)

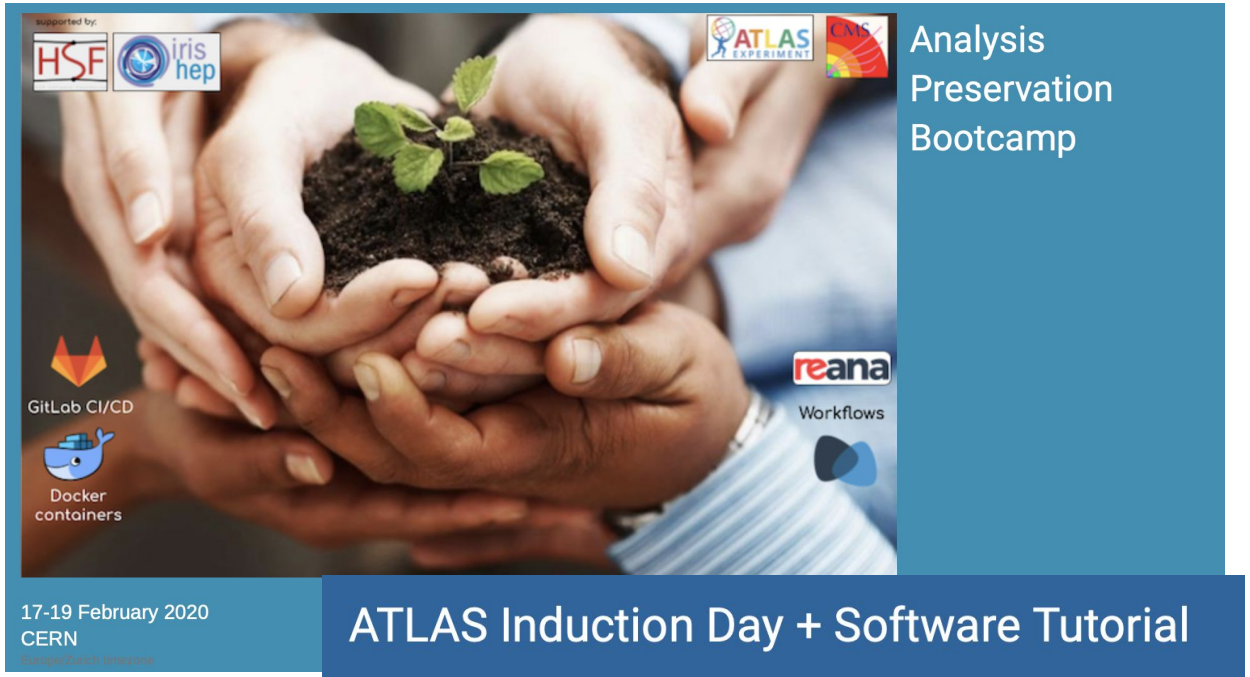
Jun - Sep 2020

Jun - Sep 2019

Bo Zheng

Rice University

Training



Analysis Preservation Bootcamp

17-19 February 2020
CERN
Europe/Zurich timezone

ATLAS Induction Day + Software Tutorial

21-25 October 2019
CERN
Europe/Zurich timezone

Introduction to pyhf	*Giordon Holtsberg Stark et al.*	
222/R-001, CERN		14:00 - 14:30
Hands-on with pyhf	*Giordon Holtsberg Stark et al.*	
Docker Analysis Release Containers	*Lukas Alexander Heinrich*	
222/R-001, CERN		16:30 - 16:50
Using GitLab for Analysis Code Management	*Giordon Holtsberg Stark*	
222/R-001, CERN		17:00 - 17:20

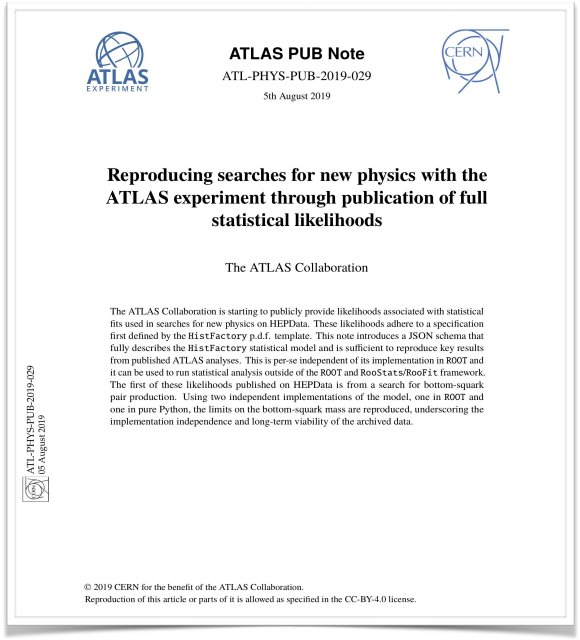
Recent Highlights

- 16 Jul 2020 - "pyhf Tutorial: Accelerating analyses and preserving likelihoods", Matthew Feickert, PyHEP 2020 Workshop
- 15 Jul 2020 - "Uproot and Awkward Array tutorial", Jim Pivarski, PyHEP 2020
- 7 Jul 2020 - "pyhf: a pure Python statistical fitting library with tensors and autograd", Matthew Feickert, 19th Python in Science Conference (SciPy 2020)
- 7 Jul 2020 - "Boost-histogram: High-Performance Histograms as Objects", Henry Schreiner, Python in Science Conference (SciPy) 2020
- 5 Jul 2020 - "Awkward Array: Manipulating JSON like Data with NumPy like Idioms", Jim Pivarski, SciPy 2020

Highlight

- The field is at a tipping point, DIANA/DASPOS/IRIS-HEP contributions have been transformational.
- First results using the RECAST reinterpretation framework and publishing full statistical likelihoods (using pyhf)

ROOT: 10+ hours
pyhf: < 30 minutes



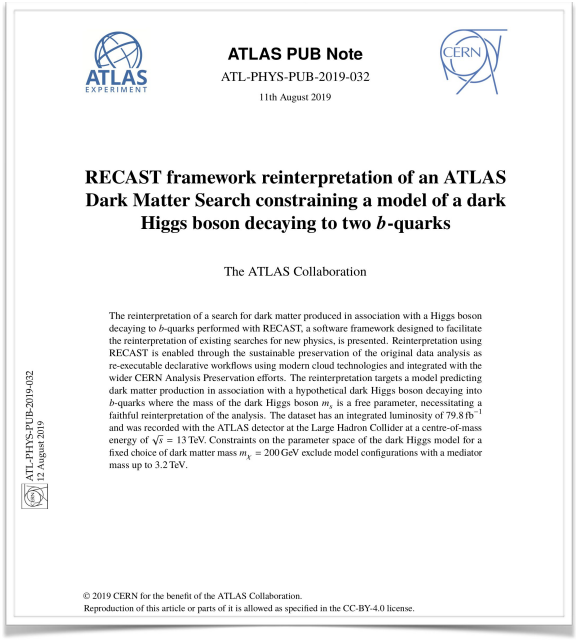
ATLAS PUB Note
ATL-PHYS-PUB-2019-029
5th August 2019

Reproducing searches for new physics with the ATLAS experiment through publication of full statistical likelihoods

The ATLAS Collaboration

The ATLAS Collaboration is starting to publicly provide likelihoods associated with statistical fits used in searches for new physics on HEPData. These likelihoods adhere to a specification first defined by the `HistFactory` p.d.f. template. This note introduces a JSON schema that fully describes the `HistFactory` statistical model and is sufficient to reproduce key results from published ATLAS analyses. This is per-se independent of its implementation in `ROOT` and it can be used to run statistical analysis outside of the `ROOT` and `RooStats`/`RooFit` framework. The first of these likelihoods published on HEPData is from a search for bottom-squark pair production. Using two independent implementations of the model, one in `ROOT` and one in pure Python, the limits on the bottom-squark mass are reproduced, underscoring the implementation independence and long-term viability of the archived data.

ATL-PHYS-PUB-2019-029
05 August 2019



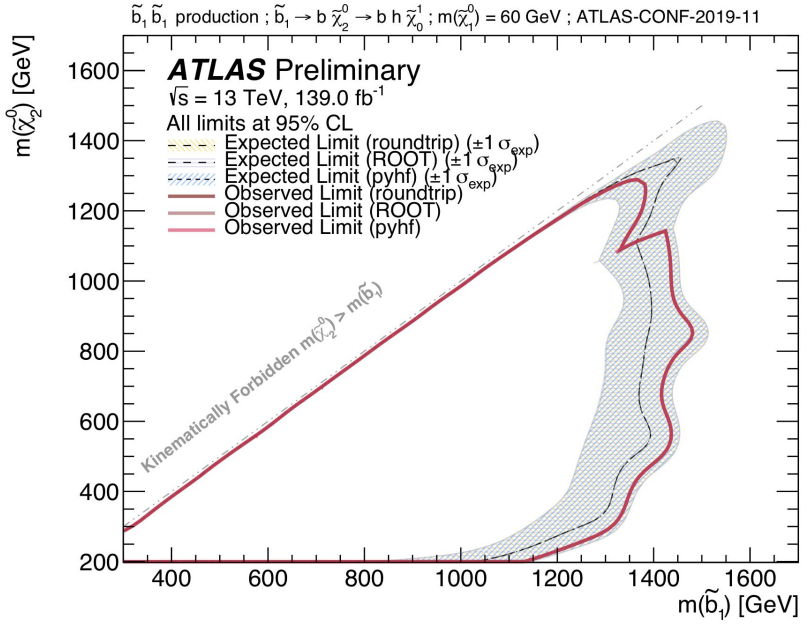
ATLAS PUB Note
ATL-PHYS-PUB-2019-032
11th August 2019

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_χ = 200 GeV exclude model configurations with a mediator mass up to 3.2 TeV.

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



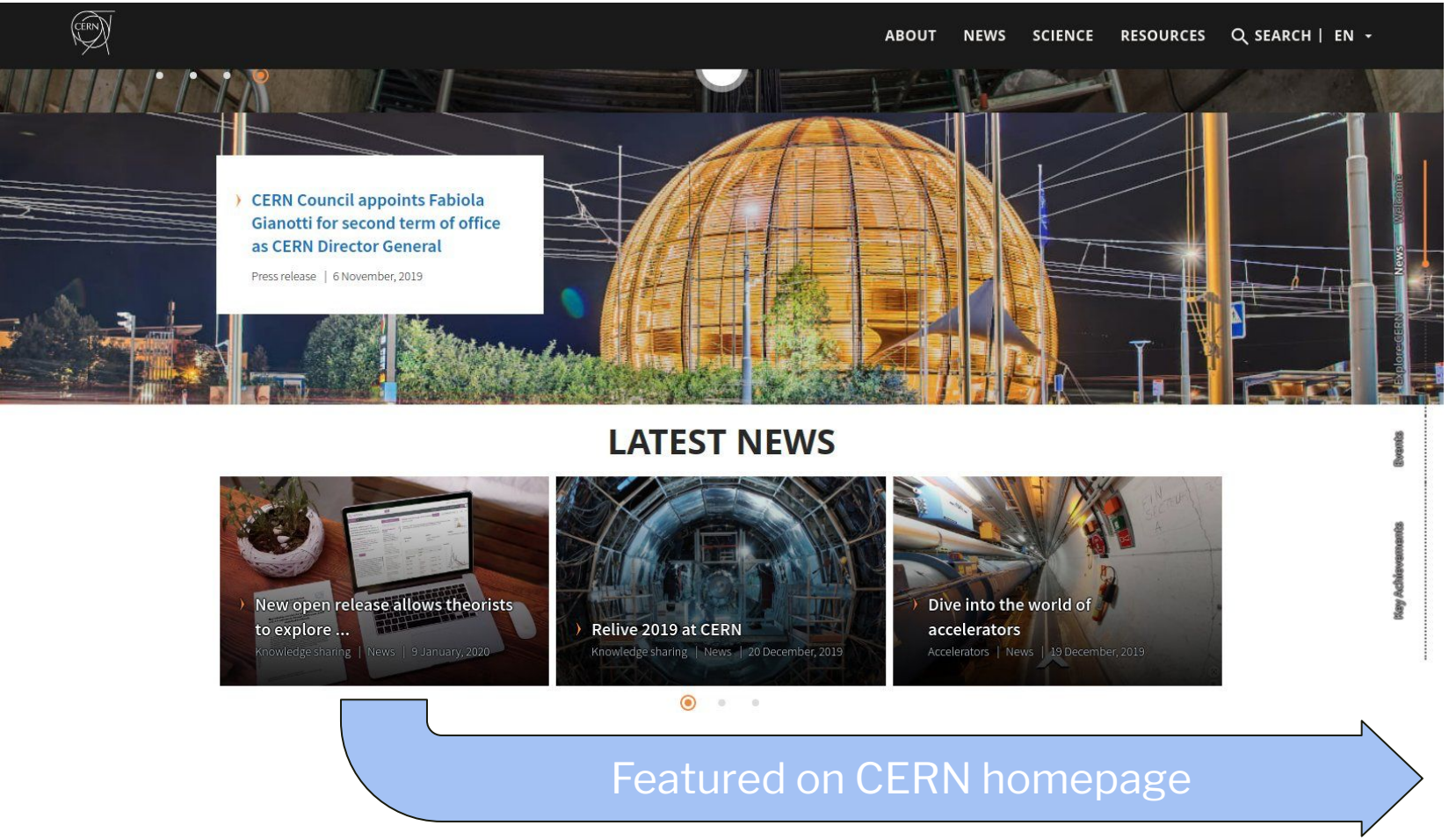
Highlight

HEPData
@HEPData

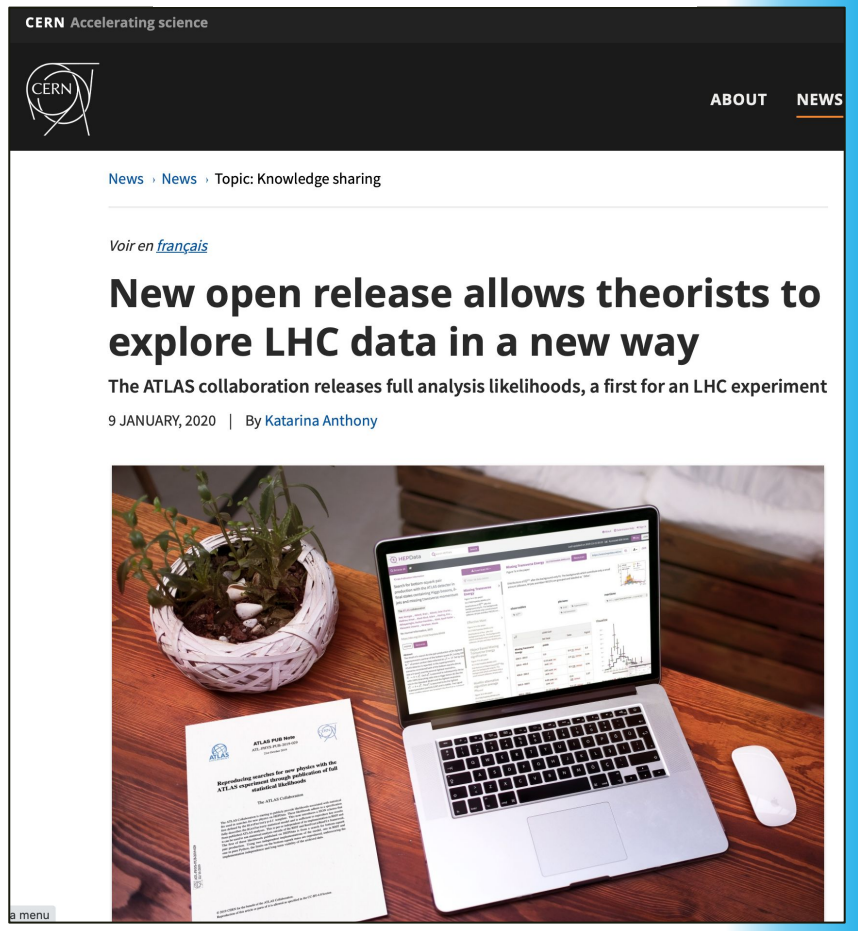
Thanks @KyleCranmer for your support and promotion of @HEPData over several years. Looking forward to future collaboration with @iris_hep on #pyhf likelihoods and more.

> **Kyle Cranmer** @KyleCranmer · Jan 29
> I would like to applaud @STFC_Matters for funding @HEPData, a vital piece of cyberinfrastructure for HEP. The @NSF has been supporting HEP software and cyberinfrastructure with DASPOS, @diana_hep and @iris_hep. @iris_hep looks forward to collaborating with you! twitter.com/HEPData/status...

1:15 PM · Jan 30, 2020 · Twitter Web App



Featured on CERN homepage



[https://arxiv.org/abs/2009.01809]

Recent Highlights

A SModelS interface for pyhf likelihoods

Gaël Alguero[a], Sabine Kraml[a], Wolfgang Waltenberger[b,c]

Abstract

SModelS is an automatized tool enabling the fast interpretation of simplified model results from the LHC within any model of new physics respecting a $\mathbb{Z}_2$ symmetry. We here present a new version of SModelS, which can use the full likelihoods now provided by ATLAS in the form of pyhf JSON files. This much improves the statistical evaluation and therefore also the limit setting on new physics scenarios.

Keywords: LHC; physics beyond the standard model; reinterpretation; simplified models; likelihoods

The new version, SModelS v1.2.4, is publicly available from `https://smodels.github.io/` and can readily be employed for physics studies. We congratulate ATLAS to the important move of making full likelihood information available in digital format and are looking forward to including more such data in future updates of SModelS.

Recent Highlights

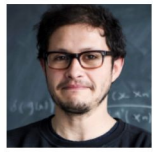
Machine Learning and the Physical Sciences

Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS)

December 11, 2020

Organizers

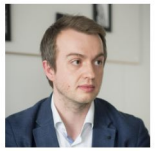
Atılım Güneş Baydin
University of Oxford



Juan Felipe Carrasquilla
Vector Institute / University of Waterloo

Adji Bousso Dieng
Columbia University

Karthik Kashinath
NERSC, Berkeley Lab



Gilles Louppe
University of Liège

Brian Nord
Fermilab

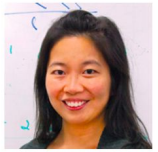
Michela Paganini
Facebook AI Research

Savannah Thais
Princeton University / IRIS-HEP

Steering Committee

Anima Anandkumar
Caltech / NVIDIA

Kyle Cranmer
New York University



Shirley Ho
Flatiron / Princeton / Carnegie Mellon

Prabhat
NERSC, Berkeley Lab

Lenka Zdeborova
Institut de Physique Théorique

Analysis Grand Challenge (from Retreat)

Some considerations

Looking for a challenge that:

- involves multiple IRIS-HEP products and can serve as an effort to help unifying and connect projects / efforts
- Needs to be clearly relevant to HL-LHC and IRIS-HEP goals
- Would like it to span scope of Analysis Systems
- Improve “light-house” and intellectual hub aspect of IRIS-HEP
- Align with other goals, like training and workforce development

Analysis Grand Challenge

User Story: "The analyzer wants to optimize an analysis end-to-end for a targeted signal hypothesis (including systematics) on an HL-LHC sized dataset so that they can obtain sensitive observed results for that signal while still being able to later reinterpret the analysis for various signal hypotheses."

Assumptions:

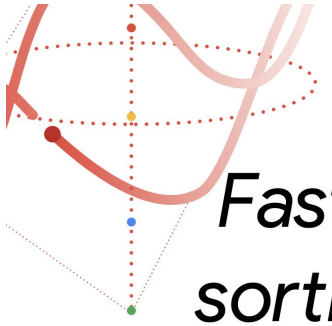
- The analyzer has 200 TB for background MC samples, a specific signal hypothesis to target for optimization, a placeholder for "observed data", and multiple signal scenarios suitable for reinterpretation.
- The analyzer has a moderately complex LHC analysis with multiple selection regions, cuts, and variables to be histogrammed for input to a template analysis tool like that of `pyhf`.
- The analyzer has access to an analysis facility with ServiceX and SkyHook and 1500 cores. The number of cores estimate is based on the requirement of being able to do a full iteration of the analysis in 25 minutes: if each core can process @ 50 kHz each this gives 75MHz which would process 100B events @ 2kB/event (=200 TB) in 25 min. To optimize the result multiple iterations will have to be performed.

Acceptance Criteria:

- End-to-end analysis optimization including systematics on a realistically sized HL-LHC (~200 TB) end-user analysis dataset + observed limit & reinterpretation afterburner
- End-to-end starts with the hand-off from DOMA via ServiceX and SkyHook. Specifications of regions, variables, and systematic variations declared using `cabinetry` and `func_adl`. Use of ServiceX, SkyHook, Coffea to perform event selection and deliver histograms for the `pyhf` model.
- Optimize analysis by using automatic differentiation to compute the gradient of the optimization target (e.g. analysis sensitivity) with respect to the analysis parameters, which are back-propagated from from output of statistics tool, through `pyhf` running in fitting service, back to ServiceX running at analysis facility, and through the event selection & histogramming code.
- Once optimized and final analysis parameters are set, apply the analysis to "observed data" (may also be synthetic in reality) to obtain "observed limits".
- End-to-end analysis optimization and results can be achieved in 24-72 hours with an analysis facility that has the anticipated HL-LHC capabilities
- Analysis Preservation & RECASTing: Preserve the optimized analysis (in git repositories, docker images, workflow components, etc.) and reproduce results and reinterpret the analysis with a new signal hypothesis.

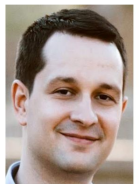
Why Differentiable?

Forward looking, game changing functionality



Fast differentiable sorting and ranking



M.Blondel O. Teboul Q. Berthet J. Djolonga

March 12th, 2020

Google Research

DL as Differentiable Programming

Deep learning increasingly synonymous with differentiable programming

Yann LeCun, 2018

"People are now building a **new kind of software** by assembling networks of parameterized **functional blocks** (including loops and conditionals) and by **training** them from examples using some form of gradient-based optimization."

Wikipedia on Differentiable programming

Why focus on differentiable programming?

- **Intellectual Leadership:** It is a modern paradigm growing and abstracting from success of deep learning, and a more natural fit to HEP than replacing everything with machine learning.
- **Increased Functionality:** We will have more sensitive analyses. Differentiable analysis systems would dramatically accelerate and improve essentially all fitting / tuning / optimization tasks. It also facilitates propagation of uncertainty in a more powerful way. Paves way to hybrid systems that fuse traditional approaches and machine learning more seamlessly.
- **Connection with Industry:** This has been an effective conduit to connections with Google (Jax and Tensorflow teams) and pytorch community.
- **Foster Innovation:** there are a ton of ideas around use of differentiable programming
- **Training & Workforce development:** These are very valuable skills, young people will do much better on job market if they are familiar with diff prob.

In our community

http://gradhep.github.io

gradHEP About Search Tags

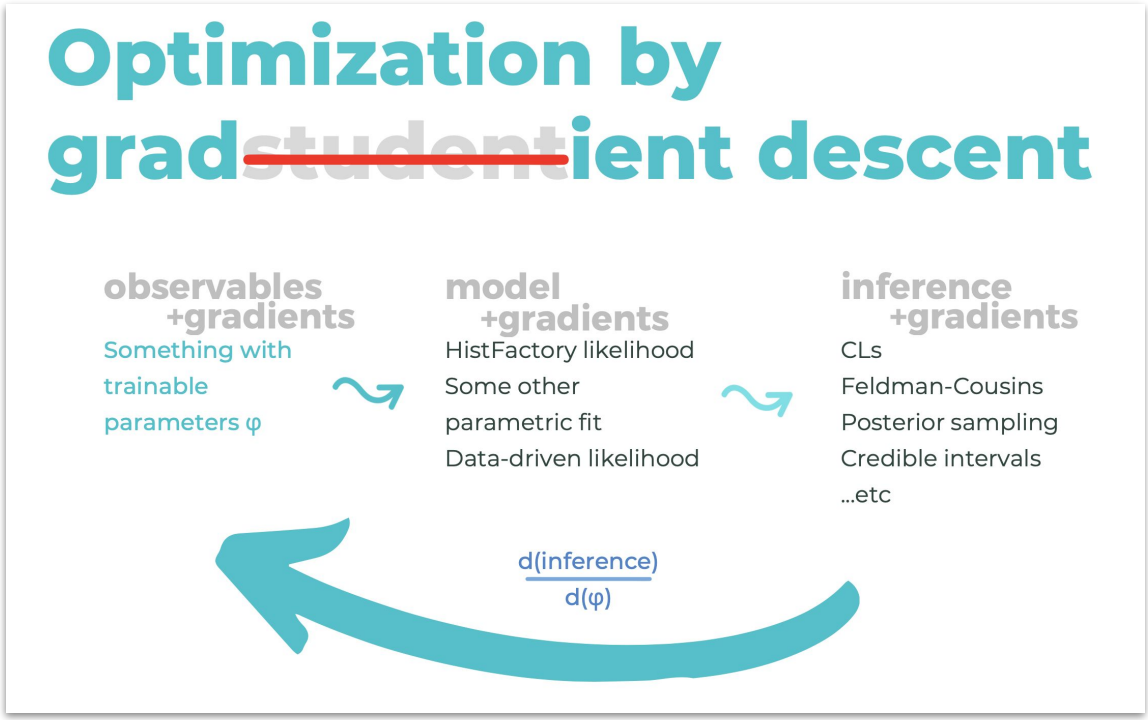
gradHEP

Welcome!

gradHEP is a group of people who are interested in high-energy physics (HEP) analysis that can be done in a *differentiable* way. This is just jargon for wanting to optimize the analysis directly with respect to the physics goals of interest using gradient-based methods, such as gradient descent.

We can make this possible if we keep track of the derivatives of each step (i.e. line of code) of the analysis with respect to its inputs. While that may sound like a harsh requirement, this is made pretty simple thanks to the magic of *automatic differentiation*, or 'autodiff' for short. If you code up a program using a library that supports autodiff, the library will keep track of the gradients throughout your program by stepping through each elementary operation – e.g. addition, multiplication, log, etc., which have known differentiation rules – and calculate the gradients using the chain rule.

Of course, this is not without its caveats, as not all lines of code are necessarily differentiable. In particular, common operations in HEP like binning a set of data or making a cut do not vary smoothly with respect to their inputs. That's why there's an ongoing effort by this group to provide drop-in replacements for these operations that are differentiable, as well as entire differentiable analysis 'blocks', such as statistical model building using HistFactory, or inference using the profile likelihood as a test statistic.

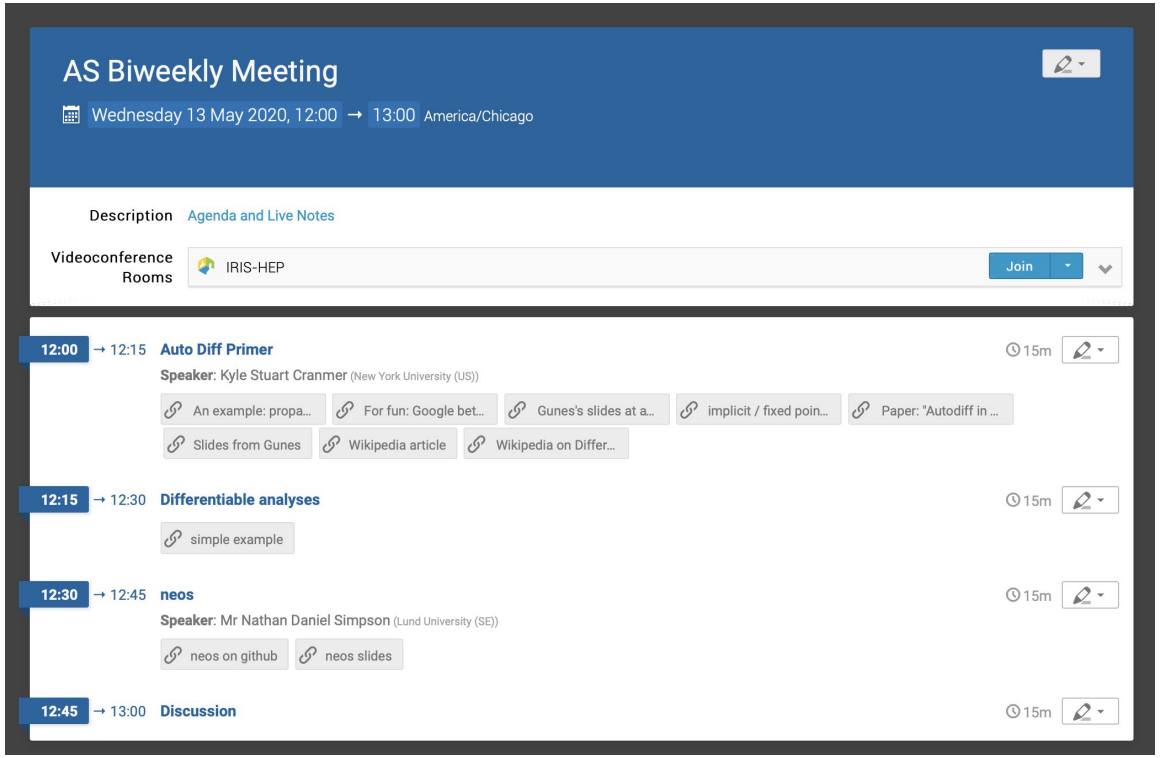


slide from Nathan Simpson: [link to talk]

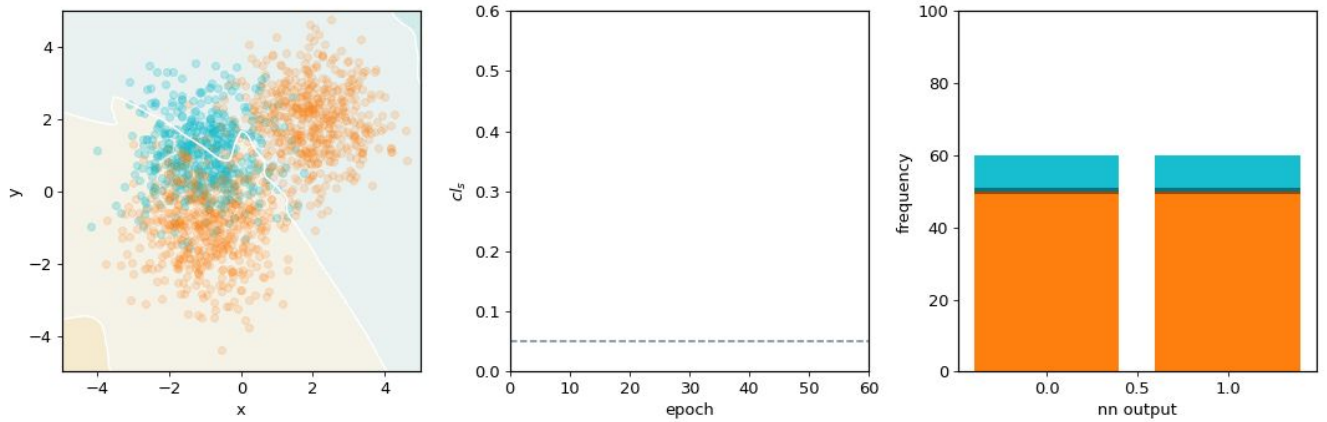
Wikipedia on Differentiable programming

Prototypes

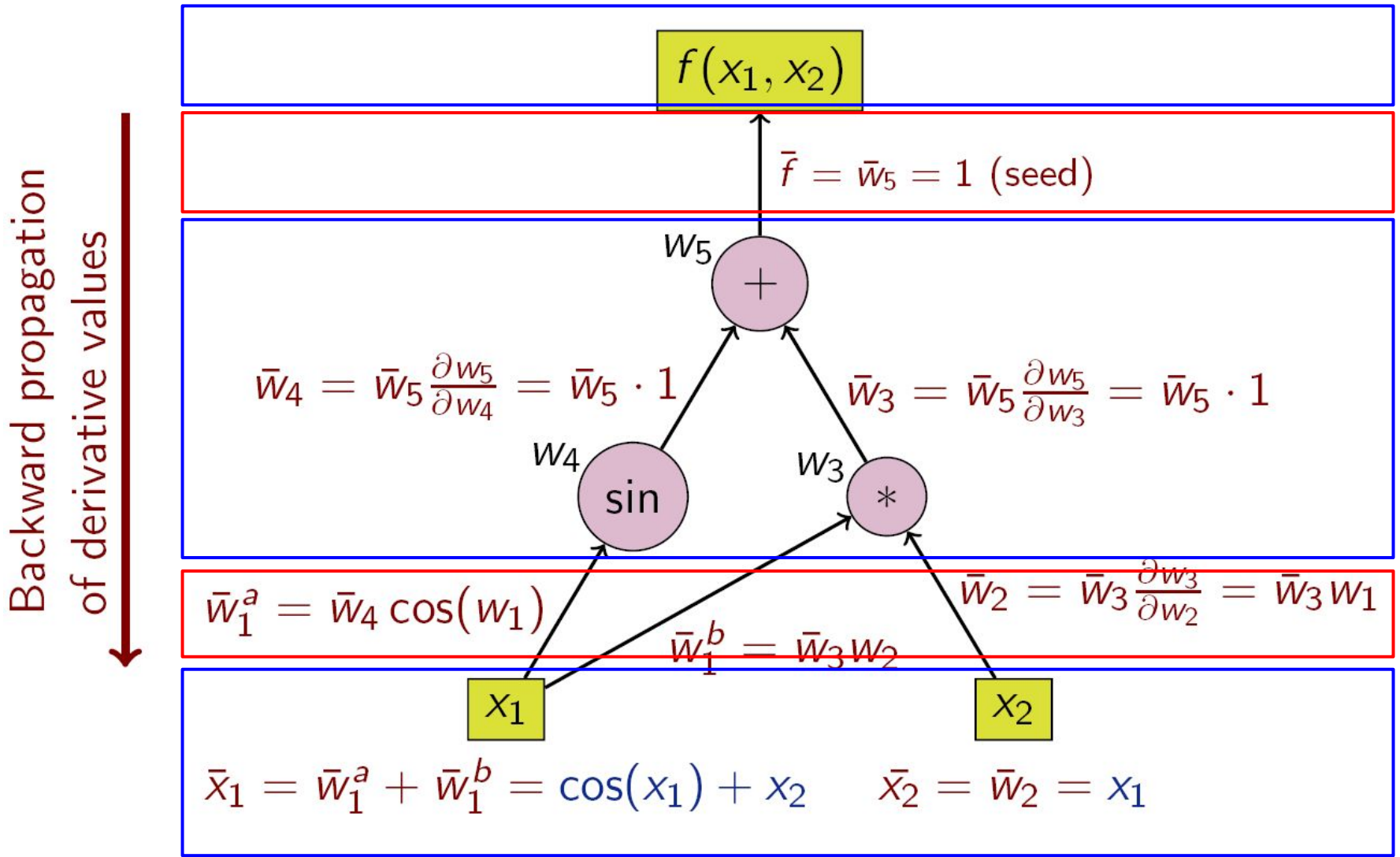
https://indico.cern.ch/event/915053/



neos



Challenge: Auto-diff across systems



fitting service calculates expected significance or limit

pass gradients back

Final event selection, filling of histograms and building of statistical model

pass gradients back ?

Initial selection of events and columns needed

Snowmass LOI

Differentiable Programming in High-Energy Physics

Atılım Güneş Baydin (Oxford), Kyle Cranmer (NYU), Matthew Feickert (UIUC),
Lindsey Gray (FermiLab), Lukas Heinrich (CERN), Alexander Held (NYU)
Andrew Melo (Vanderbilt) Mark Neubauer (UIUC), Jannicke Pearkes (Stanford),
Nathan Simpson (Lund), Nick Smith (FermiLab), Giordon Stark (UCSC),
Savannah Thais (Princeton), Vassil Vassilev (Princeton), Gordon Watts (U. Washington)

August 31, 2020

Abstract

A key component to the success of deep learning is the use of gradient-based optimization. Deep learning practitioners compose a variety of modules together to build a complex computational pipeline that may depend on millions or billions of parameters. Differentiating such functions is enabled through a computational technique known as automatic differentiation. The success of deep learning has led to an abstraction known as **differentiable programming**, which is being promoted to a first-class citizen in many programming languages and data analysis frameworks. This often involves replacing some common non-differentiable operations (eg. binning, sorting) with relaxed, differentiable analogues. The result is a system that can be optimized from end-to-end using efficient gradient-based optimization algorithms. A *differentiable analysis* could be optimized in this way — basic cuts to final fits all taking into account full systematic errors and automatically analyzed. This Snowmass LOI outlines the potential advantages and challenges of adopting a differentiable programming paradigm in high-energy physics.

Looking forward

Much (not all) of our future work is framed in the context of the Analysis Grand Challenge

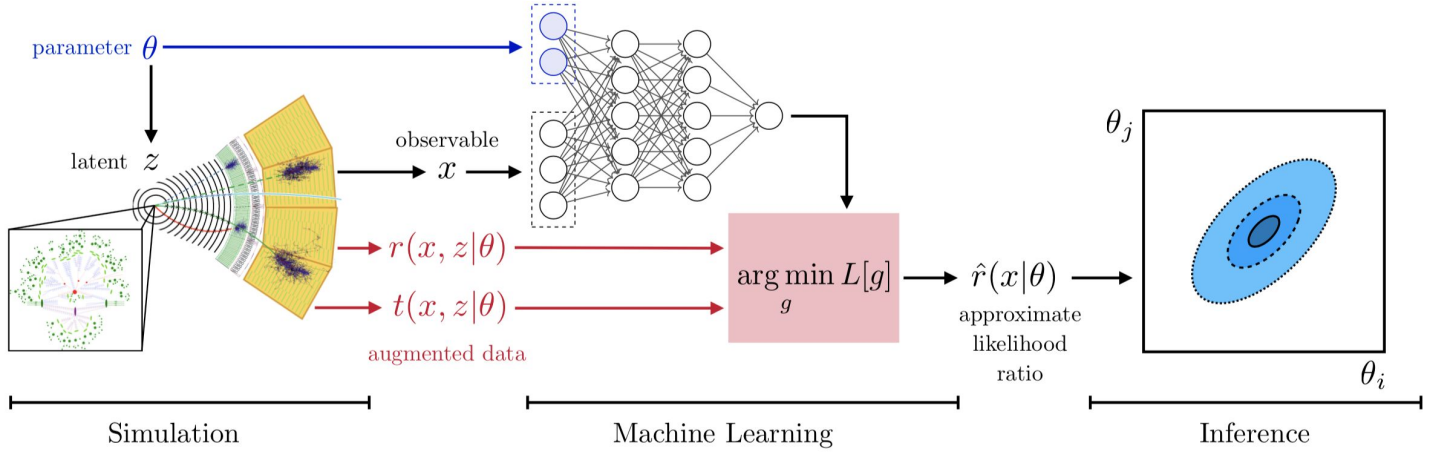
- A blueprint to bring together DOMA, Analysis Systems, SSL + Ops Programs to define one or more named scenarios for the capabilities of an analysis facility
- Scoping and specification of the target analysis to be used in the Analysis Grand Challenge.
  - *Will we have one, or one per experiment?*
- Baseline programming Interfaces between components like ServiceX, func ADL, HEP tables, Coffea, cabinetry, and pyhf.
- Differentiable programming roadmap across services needed for analysis challenge.
- Coordinate with DOMA, SSL, and operations programs to
  - *benchmark performance of prototype system components to be used for Analysis Grand Challenge.*
  - *execute the Analysis Grand Challenge*

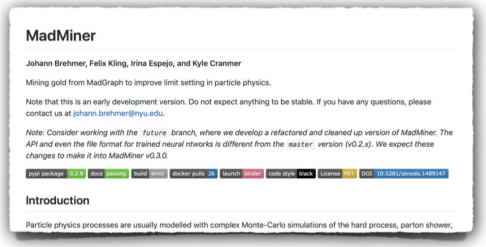
Backup

The Future

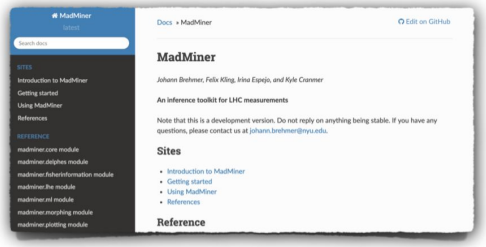
Tight integration of

- Simulation
- Machine Learning
- Statistical Inference

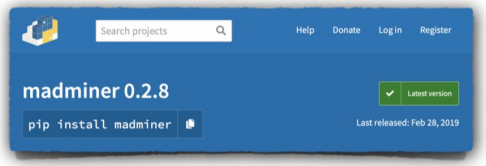




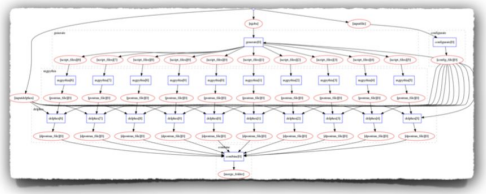
Repository and tutorials:
github.com/johannbrehmer/madminer



Documentation:
madminer.readthedocs.io

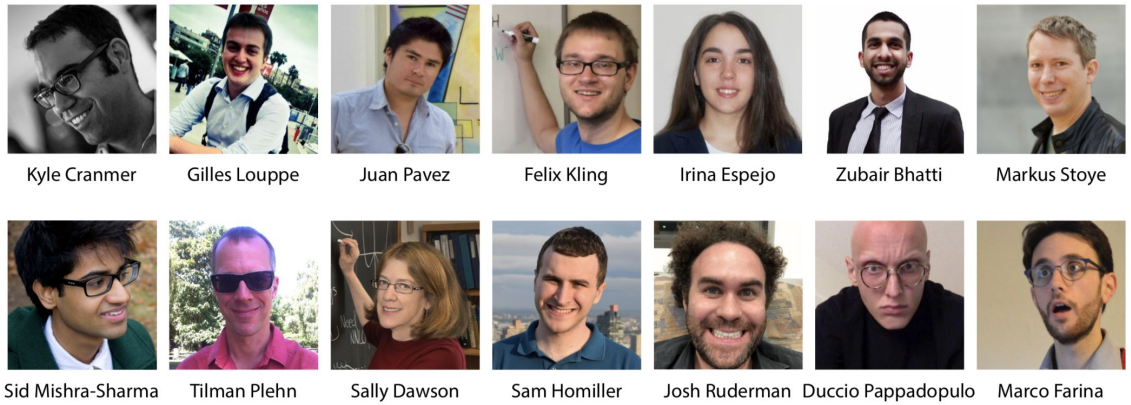


Installation:
`pip install madminer`



Deployment with Docker, yadage, REANA:
github.com/irinaespejo/workflow-madminer

34/40



Thanks to Kyle, Gilles, Felix, Irina, and Sam for material and inspiration for slides!

NYU MSDSE dianahep iris hep Institute for Research and Innovation in Software for High Energy Physics

The SCAILFIN Project
scailfin.github.io

Slides from Johann Brehmer's Keynote talk at ACAT on Constraining Effective Field Theories with Machine Learning

Major Activities

- Development of declarative specifications for different stages of analysis
- Identification and benchmarking of traditional implementations for benchmark example use-cases that span the scope of AS
- Implementation of prototype components & integration
  - *connection with DOMA (particularly ServiceX)*
- Benchmarking and assessment of prototype implementations and declarative specifications for the same example use cases
  - *connection with SSL (dedicated Blueprint Activity)*
- Exploratory research in machine learning that may impact how analysis is performed
- Engagement with community of early adopters and developers

Are there internal or external collaborations associated with each project or activity? For external collaborations, is IRIS-HEP leading, contributing or simply “connecting/liaising”?

Internal:

- **SSL**: benchmarking and scaling, REANA testbeds, etc.
- **SSL** & **DOMA**: ServiceX

External:

- **DIANA/HEP:** last bits of funding on NCE supporting various items very aligned
- **SCAILFIN**: developing products, good synergy w/ IRIS-HEP. **REANA** dev team
- **INSPIRE-HEP, HEPData, CAP, Invenio:** Advisory boards, join in development
- **ATLAS** stats effort: docker containers for RooFit-based statistical analysis & combinations and development of pyhf tools. IRIS-HEP (Matthew, Kyle, Alex) & Lukas & Giordon are leading
- HEP Statistics Serialization Standard (HS3) similar cast of characters
- **scikit-hep:** useful umbrella (not seen as US, ATLAS/CMS, or HSF) IRIS-HEP leading by example
  - *Awkward:*
    - formal collaboration with Amy Roberts at UC Denver on **Kaitai Structs**
    - frequent collaboration with **LPC/Coffea** (Lindsey Gray)
    - close liaisons with **Anaconda.com**: Numba and Dask developers
    - intermittent contact with **Oxford Big Data Institute** (genetics, developers of **Zarr**)

Link to Google doc

User Story

As an analyzer, I want to optimize an analysis end-to-end for a targeted signal hypothesis (including systematics) on an HL-LHC sized dataset so that I can obtain sensitive observed results for that signal while still being able to reinterpret the analysis for various signal hypotheses.

Link to Google doc

Assumptions

1. We have ~200TB for background MC samples, a specific signal hypothesis to target for optimization, and a placeholder for “observed data”
2. We have a typical SUSY search with multiple selection regions, cuts, and variables to be histogrammed for input to a template analysis like that of pyhf. The analysis strategy has some sensitivity to the target signal hypothesis.
3. We have an analysis facility with ServiceX and SkyHook (hand off from DOMA to AS) and 1500 cores. If each core can process @ 50 kHz each this gives 75MHz which would process 100B events @ 2kB/event (=200TB) in 25 min. Eg. 25 min per optimization iteration.
4. We have necessary ingredients to compute systematic variations.
   Either:
   a. Pre-computed event weights, scale factors, varied kinematics, etc. that needs to be processed for input to statistical model, or
   b. code to compute those ingredients on the fly
   c. Could also have (b) triggered in first pass, and then use those cached values for those ingredients cached for the later optimization passes.
5. “End-to-end” starts with necessary ingredients described above and ends with limits on signal strength and background-only p-values as objective of optimization
   a. Assuming here the beginning of the analysis chain is already in a format compatible with columnar analysis tools (eg. Arrow, Awkward, Coffea) and no conversion from xAOD etc. is needed -- this has already been demonstrated and such a conversion shouldn’t be part of our eventual analysis model)
6. We have multiple signal scenarios suitable for reinterpretation.

Link to Google doc

Acceptance Criteria

1. End-to-end analysis optimization including systematics on a realistically sized HL-LHC (~200TB) end-user analysis dataset + observed limit & reinterpretation afterburner
2. Specifications of regions, variables, and systematic variations declared using cabinetry and func_adl
3. Corresponding input datasets should be identified using in a way that is, abstracted from traditional file-based interface
4. Use of ServiceX, SkyHook, Coffea to perform event selection and deliver histograms for pyhf model
5. Optimize analysis by using automatic differentiation to compute d(Expected limit)/d(analysis parameters), which are back-propagated from from output of stats tool, through pyhf running in fitting service, back to ServiceX running at analysis facility, and through the event selection & histogramming code.
    - Notes:
        - we have prototypes for this running on a single machine [see grad-hep, neos]
        - We could make some aspects of this challenge be stretch goals, but let's don't water it down so that this goal is not taken seriously.
        - We will need to do some more quick investigation before we are are able to
        - it is also possible to use non-gradient based approaches for optimization (eg. Bayesian Optimization) that wouldn't require passing gradients back to ServiceX. And some of these approaches can use partial gradient information.
6. Once optimized: apply optimized analysis to "observed data" (may also be synthetic in reality) to obtain "observed limits".
7. Analysis Preservation & RECASTing:
    - Given a record pointing to preserved analysis (may be more than a git repository, could also include docker images, workflow components, etc.) and that I have access to compatible compute resources, I can reproduce results and reinterpret the analysis
8. Stretch: using active learning [excursion] to reinterpret efficiently

Slide for closeout session

- Generally a positive reaction to the proposed Grand Challenge
- Need to clarify framing of challenge so that it is highlights both the facilities aspect and the techniques aspect (in response to Brian's comment)
  - *NB: I consider the end-to-end optimization element to be a 'systems-level' view in terms of both the computing facilities aspect and the technique*
- We can include GPU/accelerators for bulk data processing as well as a stretch goal
- Need to start working backwards into milestones (Peter & Ben)
  - *Some aspects of challenge can be firmed up very soon, but it will take some more time to decide if some aspects of the challenge are core or stretch goals*
- Challenge motivates one or more blueprint meetings
- One milestone is to identify and settle on the facilities to carry out the challenge
  - *Options: Ops programs, Expanse SDSC, GKE?*
  - *Mike H.: Ops program can contribute by helping test and integrate the infrastructure components like SkyHook, ServiceX, etc.*